JBoss Operations Network 3.2

Using JBoss Operations Network for Monitoring, Deploying, and Managing Resources

for maintaining an efficient JBoss and IT infrastructure

Edition 3.2

Last Updated: 2017-10-25

JBoss Operations Network 3.2 Using JBoss Operations Network for Monitoring, Deploying, and Managing Resources

for maintaining an efficient JBoss and IT infrastructure
Edition 3.2

Ella Deon Ballard
dlackey@redhat.com

Legal Notice

Copyright © 2013 Red Hat, Inc..

This document is licensed by Red Hat under the Creative Commons Attribution-ShareAlike 3.0 Unported License. If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

The primary function of JBoss Operations Network is monitoring the status of your resources. The core of monitoring includes critical availability monitoring, collecting metrics on platform and server performance, and tracking events. JBoss ON also provides a way to define alerts and then notify administrators whenever a resource is performing poorly. This guide provides GUI-based procedures to view monitoring information, to track events, to define alerts and notifications, and to initiate operations.

Table of Contents

DOCUMENT INFORMATION

This guide is part of the overall set of guides for users and administrators of JBoss ON. Our goal is clarity, completeness, and ease of use.

1. DOCUMENT HISTORY

Revision 3.2-18	**May 24, 2014**	**Ella Deon Ballard**
Fixing typos.		
Revision 3.2-15	**December 11, 2013**	**Ella Deon Ballard**
Initial release for JON 3.2.		

CHAPTER 1. USING THE JBOSS ON WEB INTERFACE

JBoss Operations Network has a rich, layered UI which covers a broad range of functionality. This chapters gives a brief summary of the major sections of the UI so that users can more effectively perform management tasks.

1.1. SUPPORTED WEB BROWSERS

JBoss ON supports these browser releases for accessing the server GUI:

- Firefox 17 or later
- Internet Explorer 9

1.2. LOGGING INTO THE JBOSS ON WEB UI

Aside from some minor configuration in its **`rhq-server.properties`** file, JBoss ON is completely administered through its web interface.

By default, the JBoss ON server listens over port 7080. (A different port can be configured when the server is installed, and the port number can be changed in the server configuration.) To connect to the server, then, simply open a standard HTTP page with a URL in the format *hostname:port*. For example:

```
http://server.example.com:7080
```

Then, log in using any valid username/password combination. The default administrative user has the name and password **`rhqadmin`**.

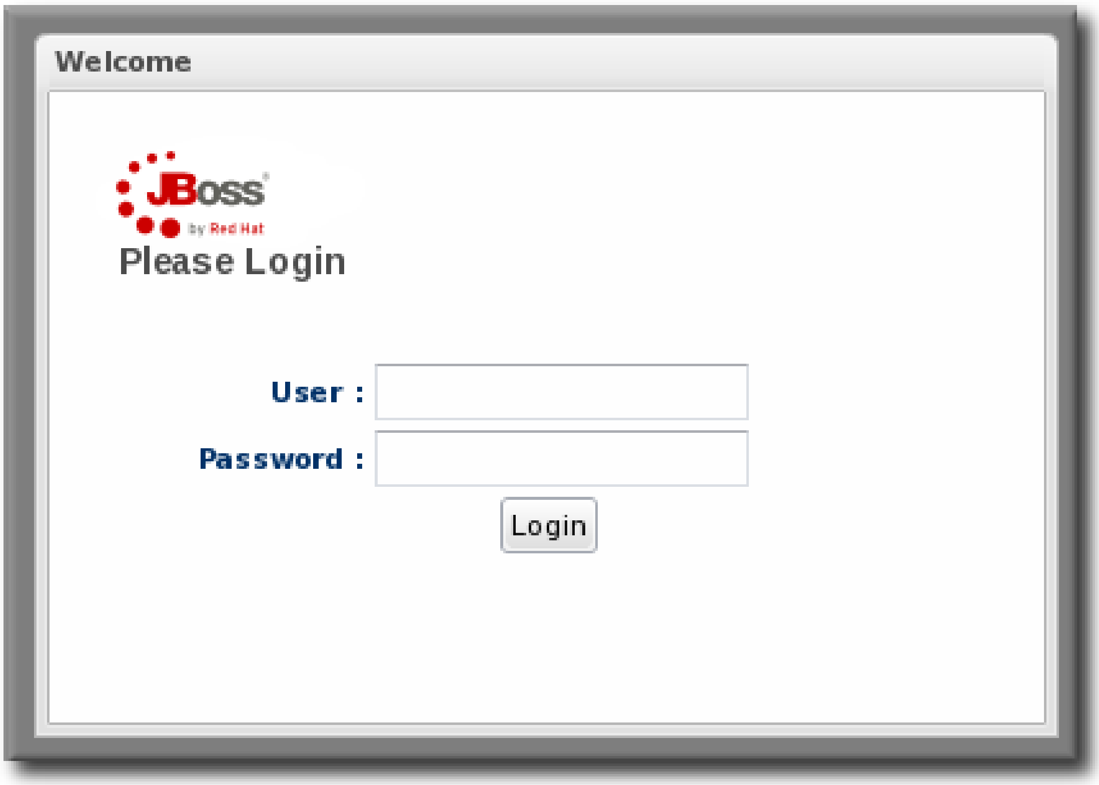
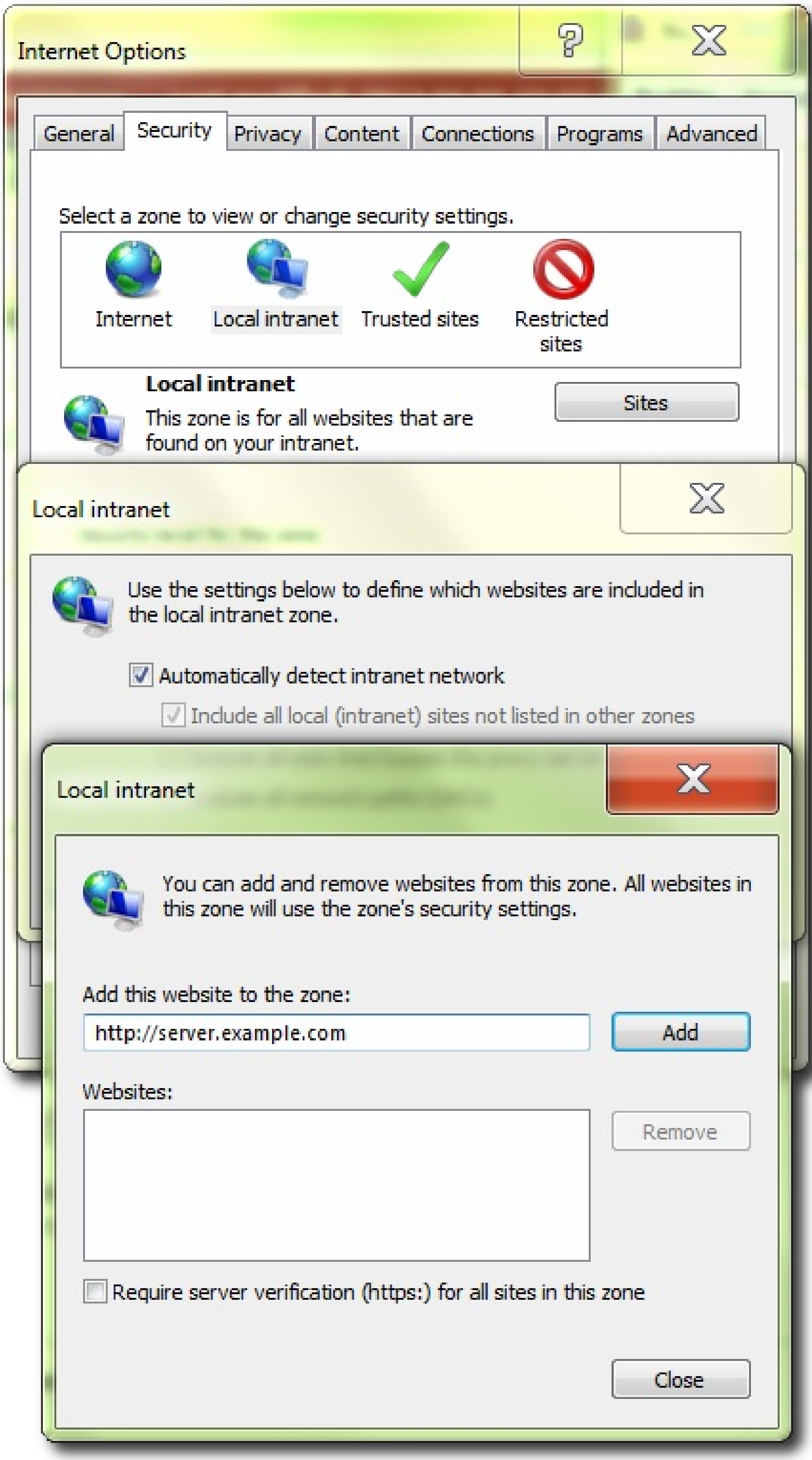


Figure 1.1. Logging into JBoss ON

1.3. CONFIGURING INTERNET EXPLORER

Some Internet Explorer settings can prevent the JBoss ON login page from loading properly. By default, Internet Explorer is in *stealth* mode, which disables some JavaScript access for websites. To allow the login page to load, add the IP address of the JBoss ON server to the whitelist for Internet Explorer.

1. In Internet Explorer, click the gear icon in the upper right corner and select **Internet options**.
2. Open the **`Security`** tab, and select the **`Local intranet`** icon.
3. Click the **`Sites`** button.
4. Click the **`Advanced`** button at the bottom of the pop-up window.
5. Enter the JBoss ON server hostname or IP address in the **`Add this webiste to the zone:`** field, and click the **`Add`**.



Select a zone to view or change security settings.

Local intranet

This zone is for all websites that are found on your intranet.

Use the settings below to define which websites are included in the local intranet zone.

☑ Automatically detect intranet network

☑ Include all local (intranet) sites not listed in other zones

You can add and remove websites from this zone. All websites in this zone will use the zone's security settings.

Add this website to the zone:

http://server.example.com

Websites:

☐ Require server verification (https:) for all sites in this zone

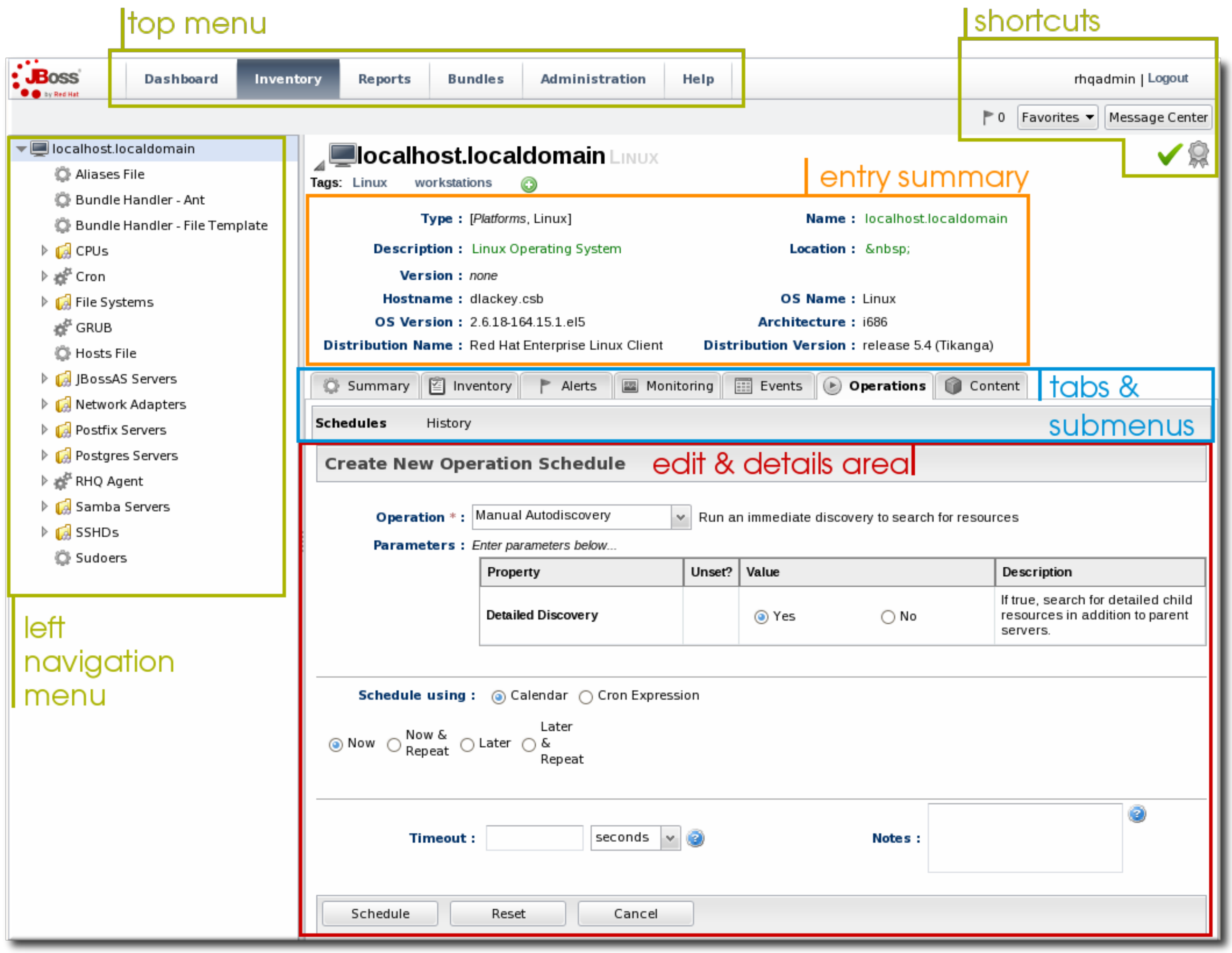
6. Close out the options windows.

1.4. A HIGH LEVEL WALK-THROUGH

The JBoss ON UI is very rich — there are a lot of small elements that are all layered together to provide a very detailed and flexible interface for interacting with the JBoss ON servers and resources. To maximize its use of space, JBoss ON uses top navigation menus, tabbed browsing with subtabs, active links, and navigation trees to establish relationships between JBoss ON resources and JBoss ON functionality. In a very general view, several types of visual elements that work together to comprise the UI:

- The top menu
- The left menu tables
- The dashboard
- Resource-based tables, which can be for the resource inventory, a summary report, or the results of a search
- Configuration pages which both provide details for and access to elements in JBoss ON, including resources, groups, plug-ins, and JBoss ON server settings

All of these elements fit together in a repeated and reliable pattern.



Tags: Linux workstations

Type : [Platforms, Linux] Name : localhost.localdomain
Description : Linux Operating System Location :
Version : none
Hostname : dlackey.csb OS Name : Linux
OS Version : 2.6.18-164.15.1.el5 Architecture : i686
Distribution Name : Red Hat Enterprise Linux Client Distribution Version : release 5.4 (Tikanga)

Create New Operation Schedule

Operation * : Manual Autodiscovery Run an immediate discovery to search for resources
Parameters : Enter parameters below...

Property	Unset?	Value	Description
Detailed Discovery		[✓ Yes] No	If true, search for detailed child resources in addition to parent servers.

Schedule using : [✓ Calendar] Cron Expression

[✓ Now] Now & Repeat Later Later & Repeat

Timeout : seconds

Notes :

Figure 1.2. UI Elements All Together

Understanding the type of page that you're on can make it easier to navigate through the JBoss ON UI and can help you more completely understand what you can accomplish in JBoss ON.

1.4.1. The Top Menu

At the very top of the JBoss ON UI is a menu bar with, with five tabs that go to the major configuration areas of JBoss ON.

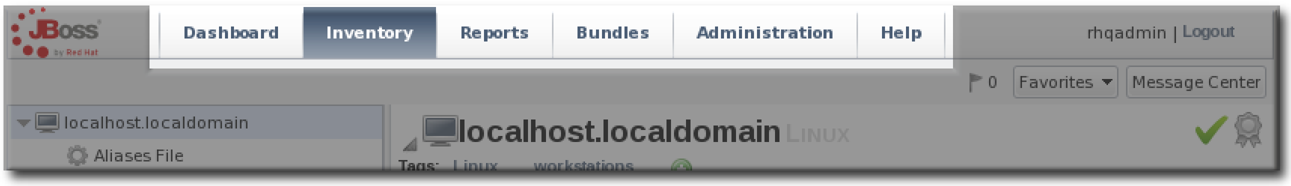


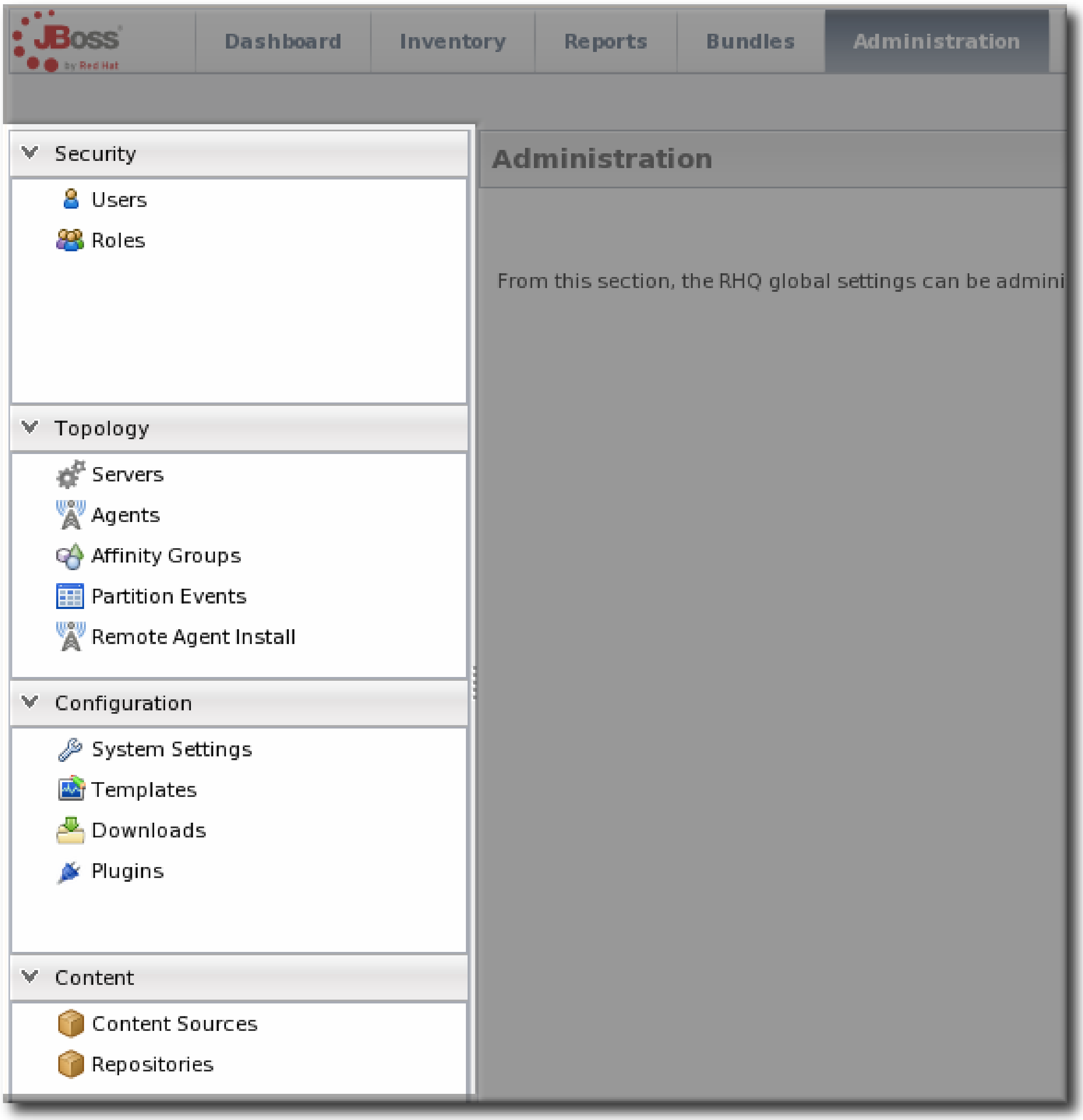
Figure 1.3. The Top Menu

Each menu item relates to a different functional aspect of JBoss ON.

- The Dashboard contains a global overview of JBoss ON and its resources. Different, configurable snapshot summaries (called *portlets*) show different aspects of the resources and server, such as the discovery queue, recent alerts, recent operations, and resource counts.
- The `Inventory` tab shows both resources and groups.
- The Reports tab shows pre-defined reports. These are slightly different than the Dashboard, which focuses exclusively on resource information: the reports look at the current actions of the different subsystem (or major functional areas) of JBoss ON, such as alerts, operations, metric collection, and configuration history.
- The Bundles tab opens the provisioning and content functional area. This is for uploading and deploying content bundles that are used to provision new applications.
- Administration goes to all areas related to configuring the JBoss ON server itself. This includes server settings, plug-ins, users and security, and agent settings.

1.4.2. The Left Menu

Rather than using drop-down or tabbed options, much of the configuration for JBoss ON is accessed through the left menu. There are individual tables that contain related areas of configuration, like users and groups, server configuration, server/agent connections, and content for the `Administration` area in Figure 1.4, “The Left Menu”.



From this section, the RHQ global settings can be admini

Figure 1.4. The Left Menu

Clicking the up or down arrows at the left of the menu tables collapses and expands the tables. This can make it easier to navigate the left menu.

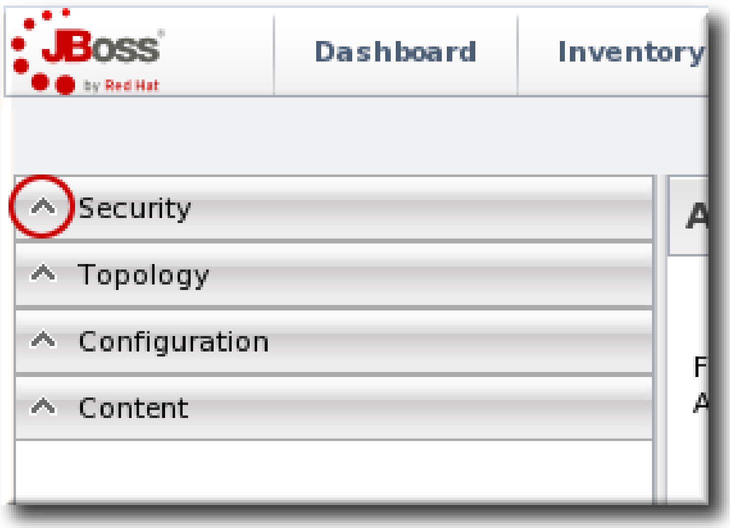
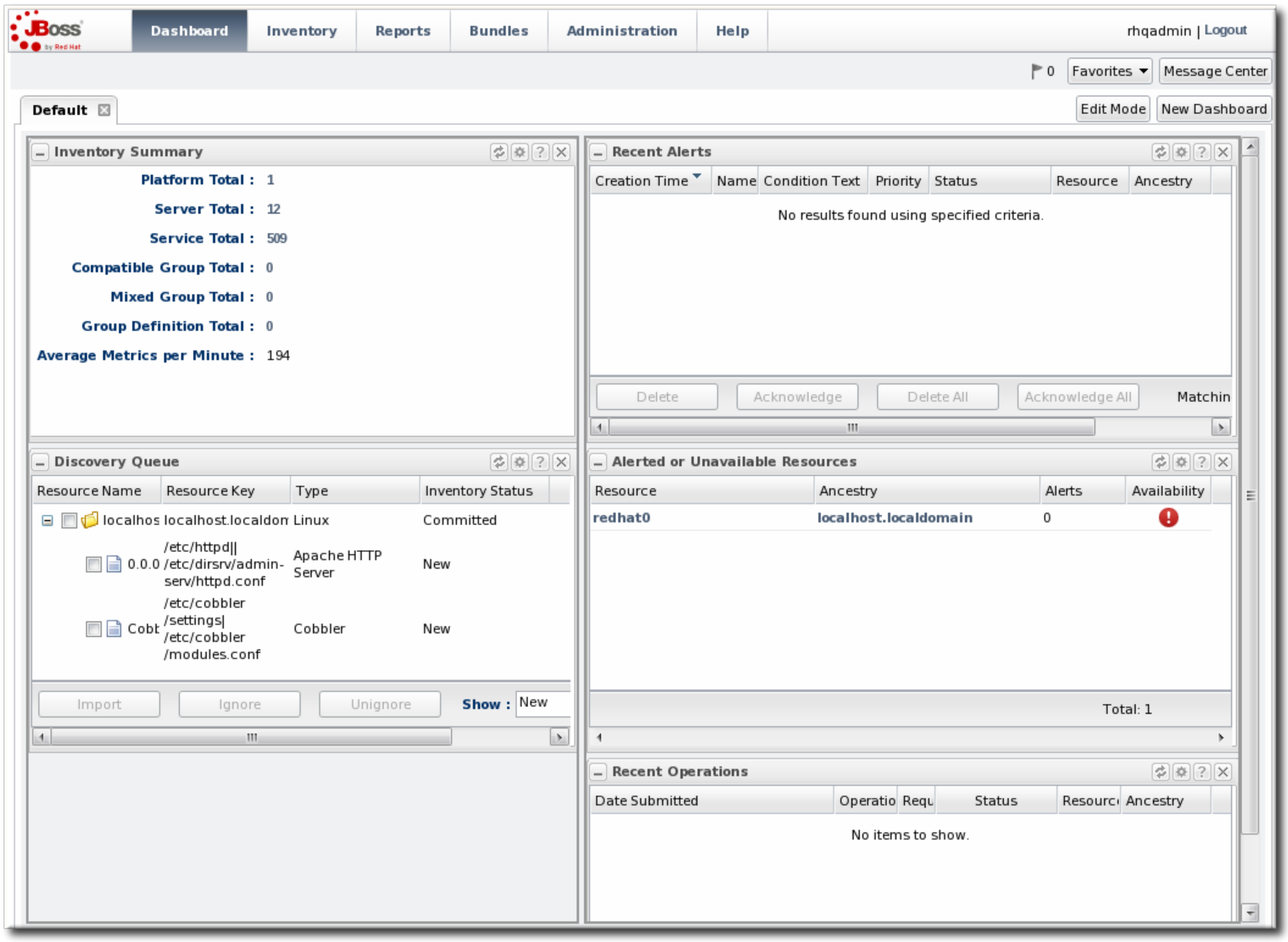


Figure 1.5. Collapsing the Left Menu

1.4.3. Dashboard

The Dashboard is an overview of everything in JBoss ON, from recent actions (fired alerts and operations) to availability reports to newly discovered and imported resources. This page, unlike any other area of JBoss ON, is customizable so it can be used to display only the collection of information that you want to see. Each table of information is called a *portlet*, a mini-portal into a view of the JBoss ON server or resources. There can be multiple Dashboard views configured, with different portlets or different layouts; these are accessed by tabs at the top of the Dashboard page.



Inventory Summary

Platform Total :	1
Server Total :	12
Service Total :	509
Compatible Group Total :	0
Mixed Group Total :	0
Group Definition Total :	0
Average Metrics per Minute :	194

Recent Alerts

Creation Time	Name	Condition Text	Priority	Status	Resource	Ancestry

No results found using specified criteria.

Delete | Acknowledge | Delete All | Acknowledge All | Matchin

Discovery Queue

Resource Name	Resource Key	Type	Inventory Status
localhos	localhost.localdon	Linux	Committed
0.0.0	/etc/httpd||/etc/dirsrv/admin-serv/httpd.conf	Apache HTTP Server	New
Cobt	/etc/cobbler/settings|/etc/cobbler/modules.conf	Cobbler	New

Import | Ignore | Unignore | Show : New

Alerted or Unavailable Resources

Resource	Ancestry	Alerts	Availability
redhat0	localhost.localdomain	0	●

Total: 1

Recent Operations

Date Submitted	Operatio	Requ	Status	Resourc	Ancestry

No items to show.

Figure 1.6. Dashboard View

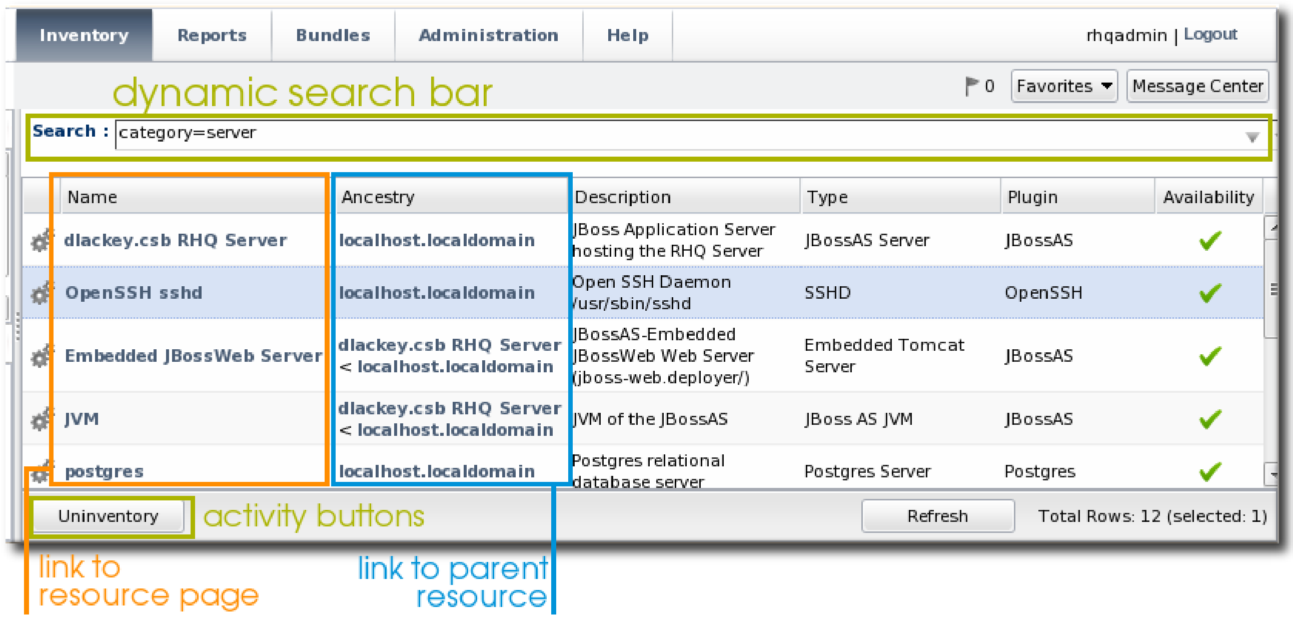
The Dashboard is the landing page for JBoss ON, the first page that comes up after login.

1.4.4. Inventory Browsers and Summaries

Some pages are essentially long tables of information, presented in basically the same way:

- Tabs for different areas, with subtabs that further break down information
- A table of results
- Icons that open a configuration or task option for that specific entity
- Buttons that perform actions (create, delete, or some other specific action) on the entries; some of these buttons aren't active unless an entry is selected

The inventory interface in Figure 1.7, “Inventory Browser” is rich with functionality. The search bar for resources and groups uses a specialized syntax and flexible dynamic search. Hovering over any resource name gives a small popup message with more information about that resource. Clicking the name of the entry itself opens its default entry configuration page, while clicking the name of its parent opens up that parent resource's configuration page.



Name	Ancestry	Description	Type	Plugin	Availability
dlackey.csb RHQ Server	localhost.localdomain	JBoss Application Server hosting the RHQ Server	JBossAS Server	JBossAS	✓
OpenSSH sshd	localhost.localdomain	Open SSH Daemon /usr/sbin/sshd	SSHD	OpenSSH	✓
Embedded JBossWeb Server	dlackey.csb RHQ Server < localhost.localdomain	JBossAS-Embedded JBossWeb Web Server (jboss-web.deployer/)	Embedded Tomcat Server	JBossAS	✓
JVM	dlackey.csb RHQ Server < localhost.localdomain	JVM of the JBossAS	JBoss AS JVM	JBossAS	✓
postgres	localhost.localdomain	Postgres relational database server	Postgres Server	Postgres	✓

Total Rows: 12 (selected: 1)

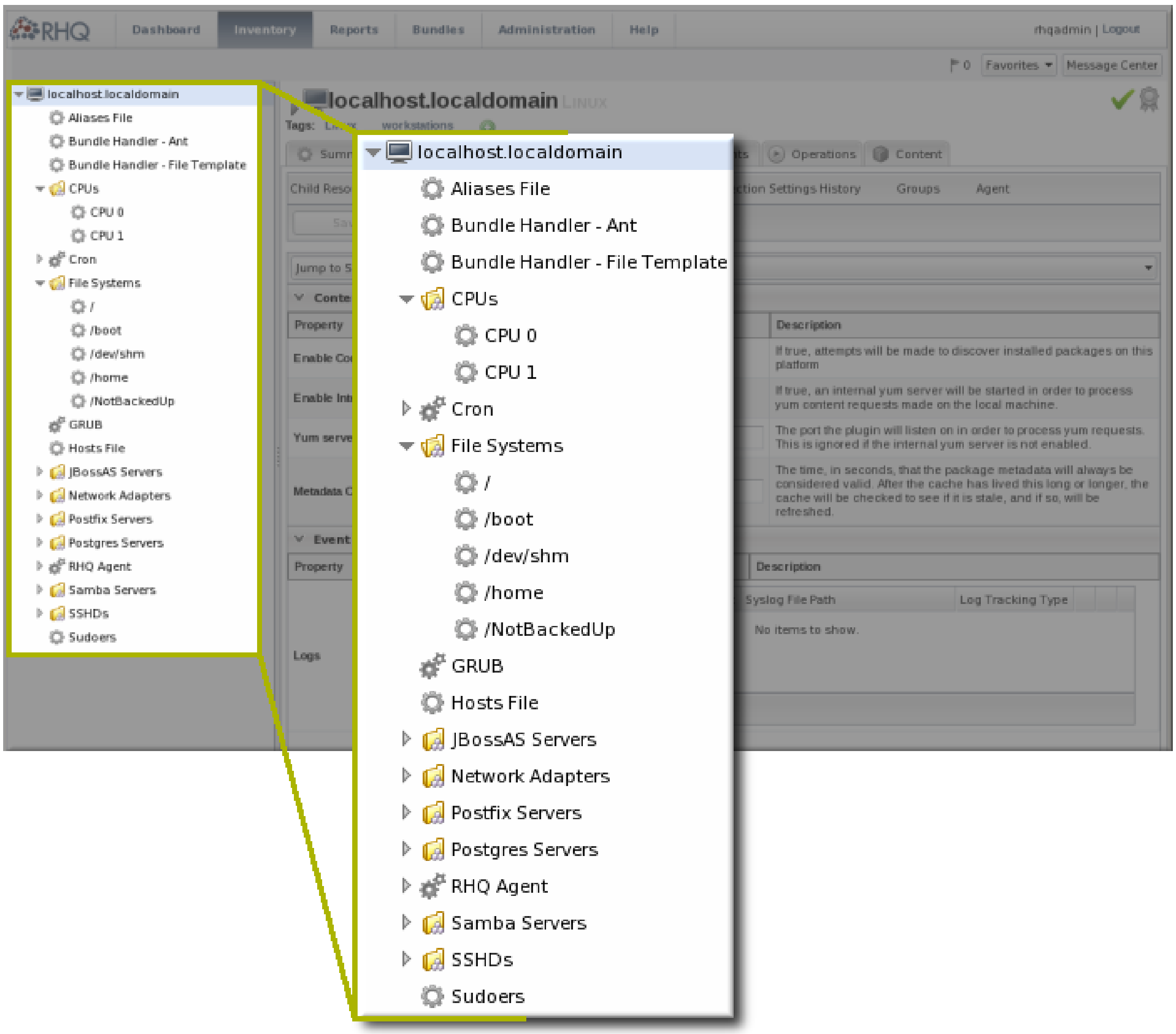
Figure 1.7. Inventory Browser

In Figure 1.7, “Inventory Browser”, the **`UNINVENTORY SELECTED`** button is active because a resource is selected. If no entries are actively selected, activity buttons are grayed out.

1.4.5. Entry Details Pages

Possibly the most functionality-saturated area in JBoss ON is an entry's details page.

The left navigation area shows the hierarchy, both parents and children, of the selected resource. This makes it very easy to navigate among all of the different services and servers that affect a resource.



If true, attempts will be made to discover installed packages on this platform

If true, an internal yum server will be started in order to process yum content requests made on the local machine.

The port the plugin will listen on in order to process yum requests. This is ignored if the internal yum server is not enabled.

The time, in seconds, that the package metadata will always be considered valid. After the cache has lived this long or longer, the cache will be checked to see if it is stale, and if so, will be refreshed.

Figure 1.8. Resource Tree

Right-clicking any of the resources in the left navigation opens shortcuts to that entry's configuration.

NOTE

Resources have short names that are automatically assigned based on their type, instance or system name, or IP address. These names are used in the inventory and in the tree navigation.

The configuration area of a resource entry page (and other JBoss ON entities, like plug-ins and templates) has three information layers that provide all of the possible functionality and tasks available for that entry.

The entry's configuration page is tabbed according to each area that can be configured, and frequently has subtabs for additional configuration options and to show the history of that area (like fired alerts, previous content updates, or monitoring data).

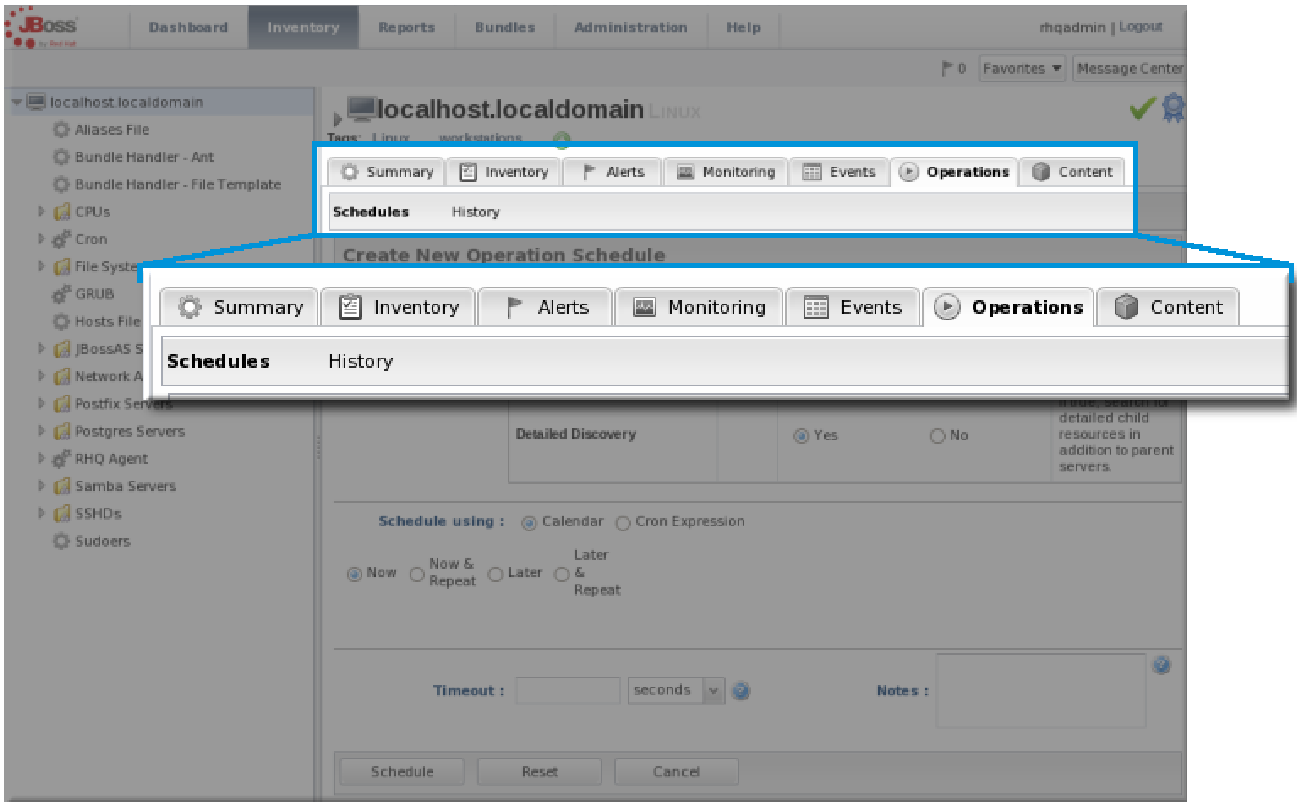
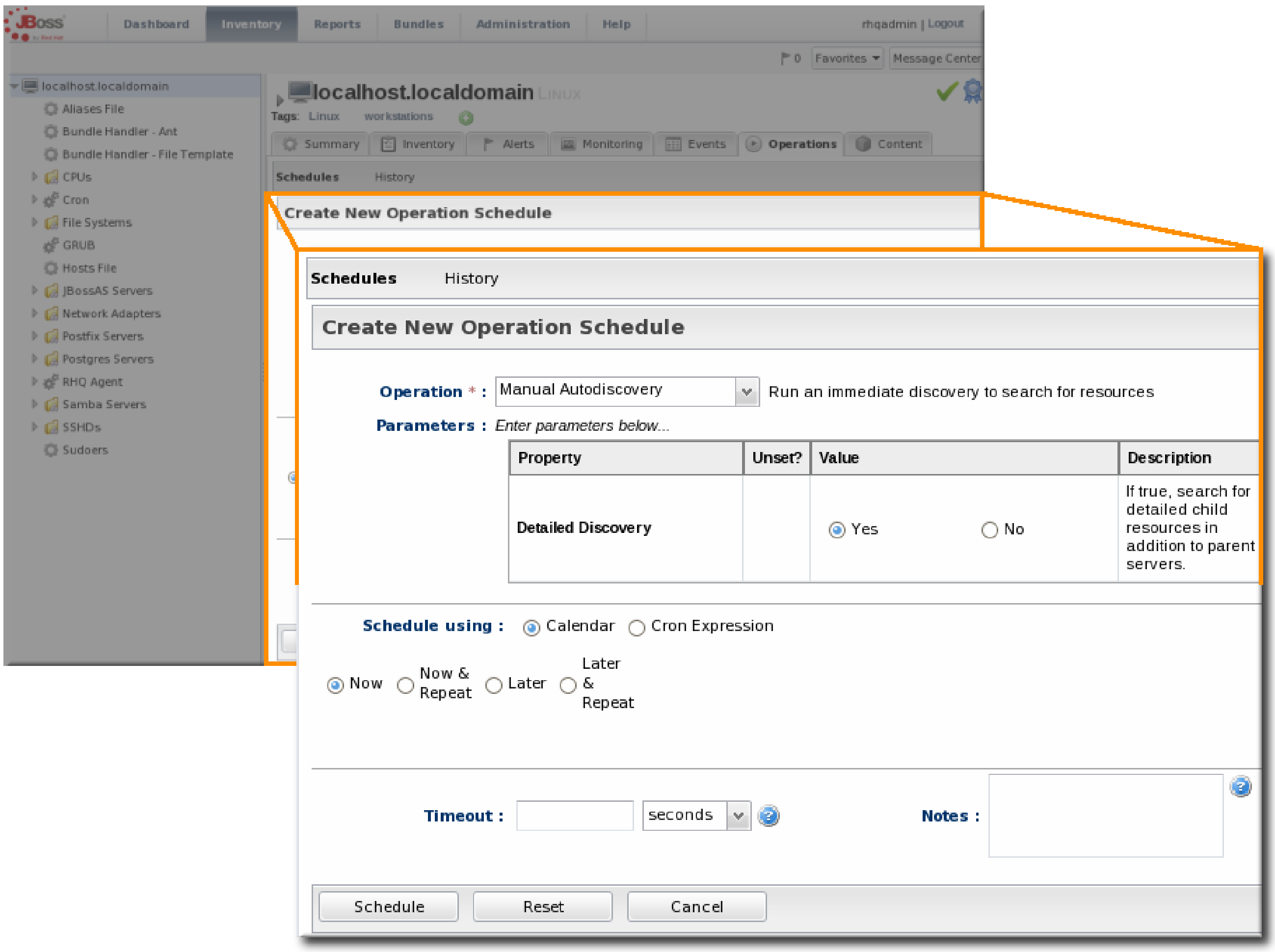


Figure 1.9. Tabs for a Resource Entry

The next section in the entry area shows lists of related configured entries for that task. For example, an `Operations` area will have a list of available operations in a table below the tabs. For `Inventory`, there is a list of configured child resources, while `Alerts` shows all of the configured alerts for that resource. All of those entries are listed in a table similar to the search results available in other parts of the JBoss ON UI.

Many elements are both a *details* page and an *edit* page. meaning that many fields are active automatically. This makes it possible to perform management tasks directly, without opening a separate page.



Create New Operation Schedule

Operation * : Manual Autodiscovery — Run an immediate discovery to search for resources

Parameters : *Enter parameters below...*

Property	Unset?	Value	Description
Detailed Discovery		[✓ Yes] No	If true, search for detailed child resources in addition to parent servers.

Schedule using : [✓ Calendar] Cron Expression

[✓ Now] Now & Repeat Later Later & Repeat

Timeout : seconds

Notes :

Schedule Reset Cancel

Figure 1.10. Editable Areas for a Resource Entry

1.4.6. Shortcuts in the UI

To the far right of the top menu is a small cluster of icons that provide very quick, targeted insight into JBoss ON.

- The Message Center shows all notifications that have been sent by the JBoss ON server. This includes alerts, configuration changes, changes to the inventory, or error messages for the server or UI.
- The Favorites button can be used to navigate to selected resources and groups quickly, while the little blue ribbon on resource pages can be used to add that resource to the favorites list.
- The resource available is shown as a green check mark if the resource is available and a red X if the resource is down.

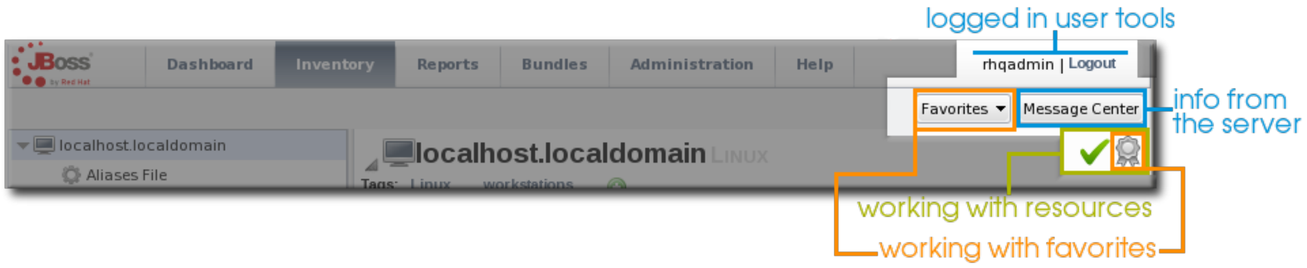
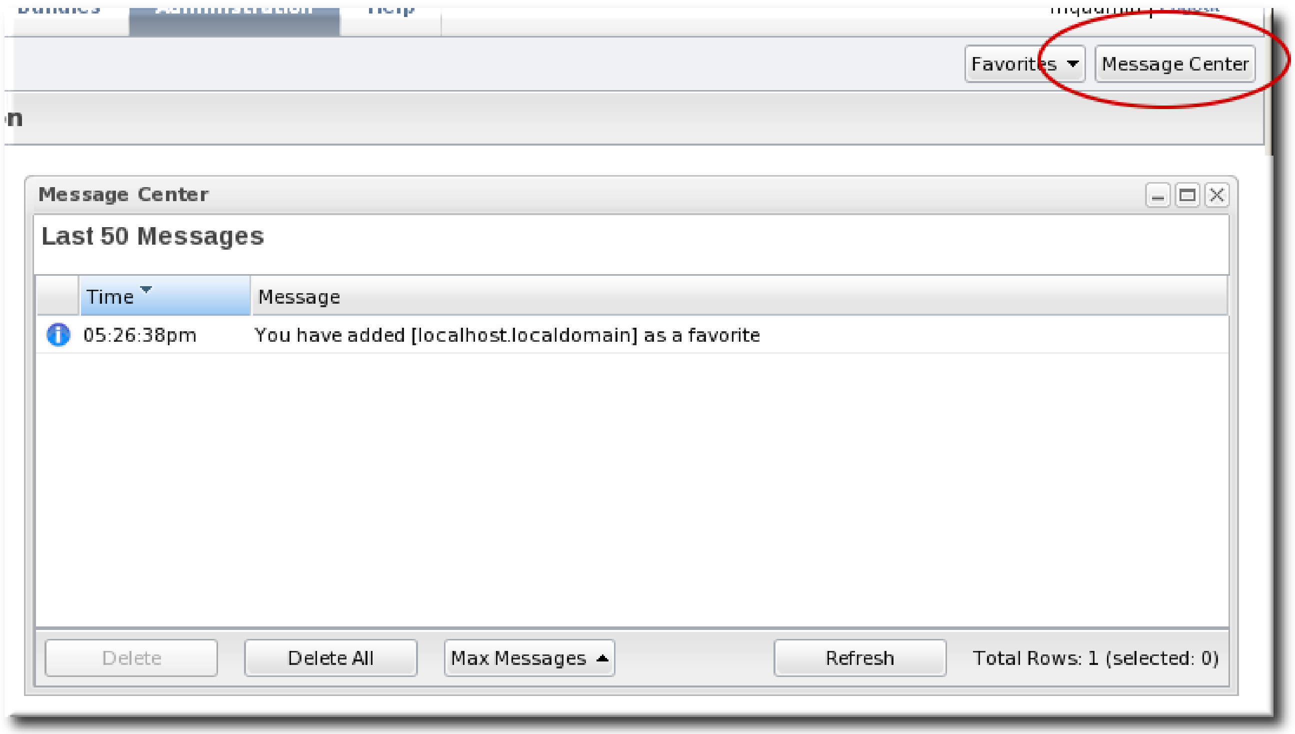


Figure 1.11. Shortcuts

1.5. GETTING NOTIFICATIONS IN THE MESSAGE CENTER

The Message Center shows all of the messages that have been returned by the GUI for the current browser session. This includes any actions taken in the UI — like adding resources to the inventory, configuring resources, or uploading content — and it also includes any error messages that may have been returned during the session.



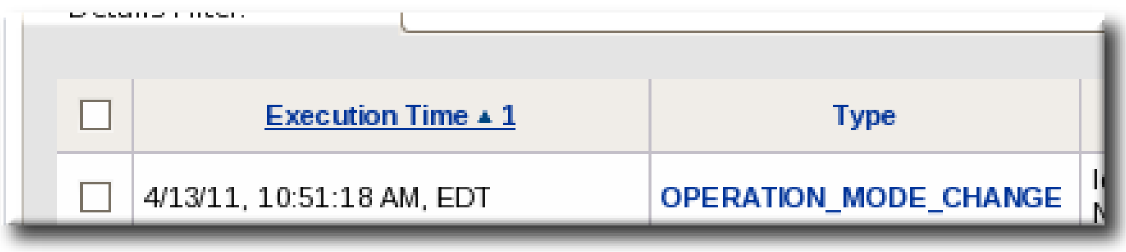
```
05:26:38pm    You have added [localhost.localdomain] as a favorite
```

Figure 1.12. Message Center

1.6. SORTING AND CHANGING TABLE DISPLAYS

Almost all of the information in JBoss ON is displayed in tables, from the resource inventory to the list of plug-ins for the agent. The SmartGWT UI has some versatility in how that table information is sorted and displayed.

A few tables use a very simple ascending/descending order based on the column being sorted, either numerically or alphabetically.



	Execution Time ▲ 1	Type
☐	4/13/11, 10:51:18 AM, EDT	OPERATION_MODE_CHANGE

Figure 1.13. Basic Table Sorting on the Partition Events List

Most areas in the UI allow a more complex method of displaying information. As with basic tables, simply clicking a column name will sort that column in ascending/descending order. However, advanced GWT tables also have an option to change the table layout and sort options, by clicking a menu arrow at the right of the column.

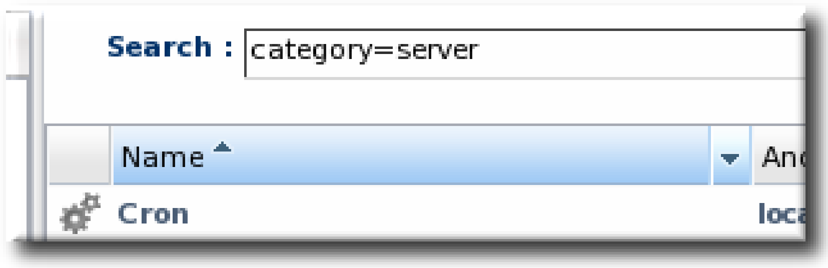


Figure 1.14. Basic Table Sorting on the Server Resources List

When a menu arrow is selected, the sort order for that column can be changed, or any other column. You can also change the column sizing and even the types of columns displayed. The options are generated dynamically, depending on what kind of entry is contained in the table.

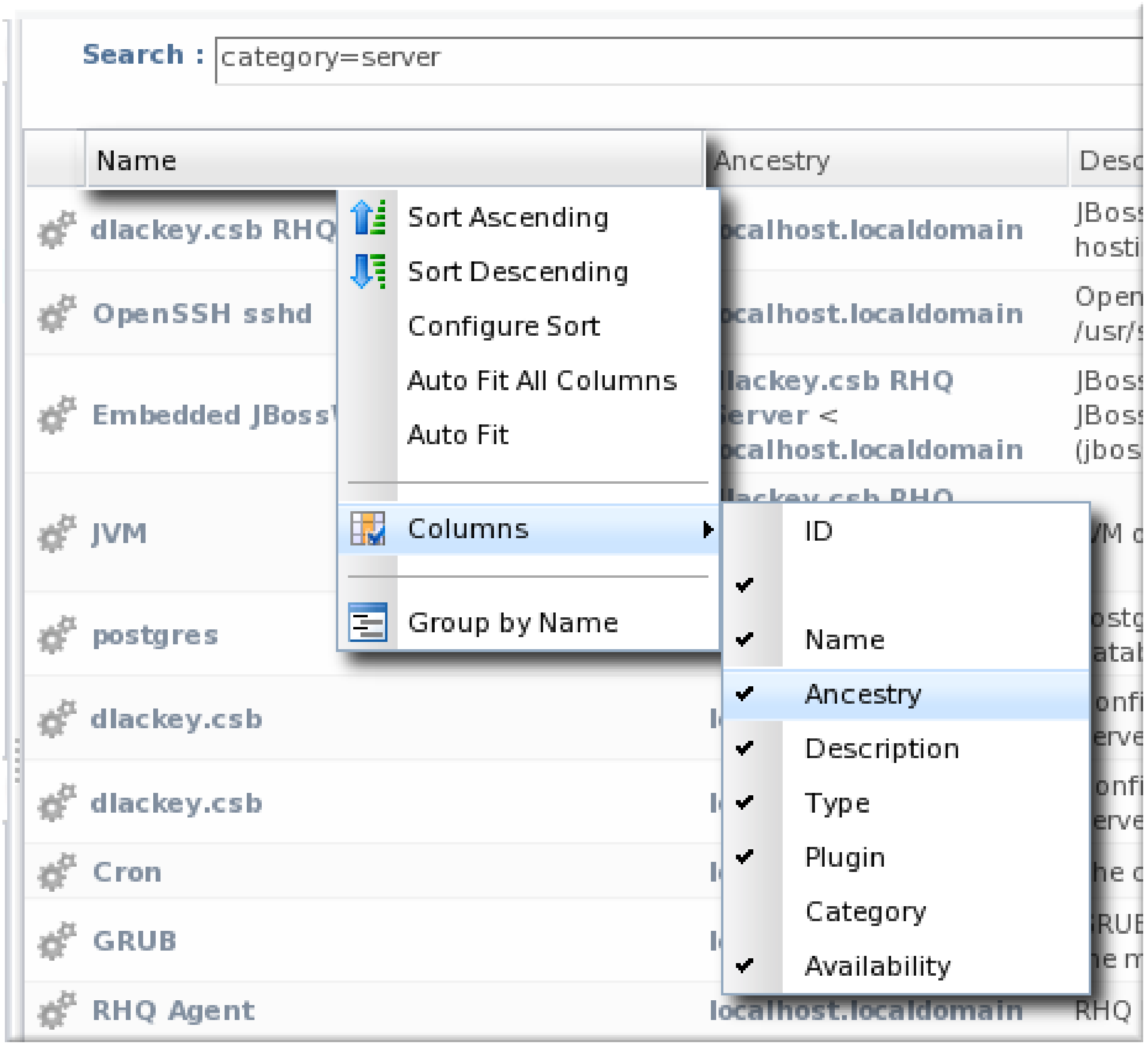
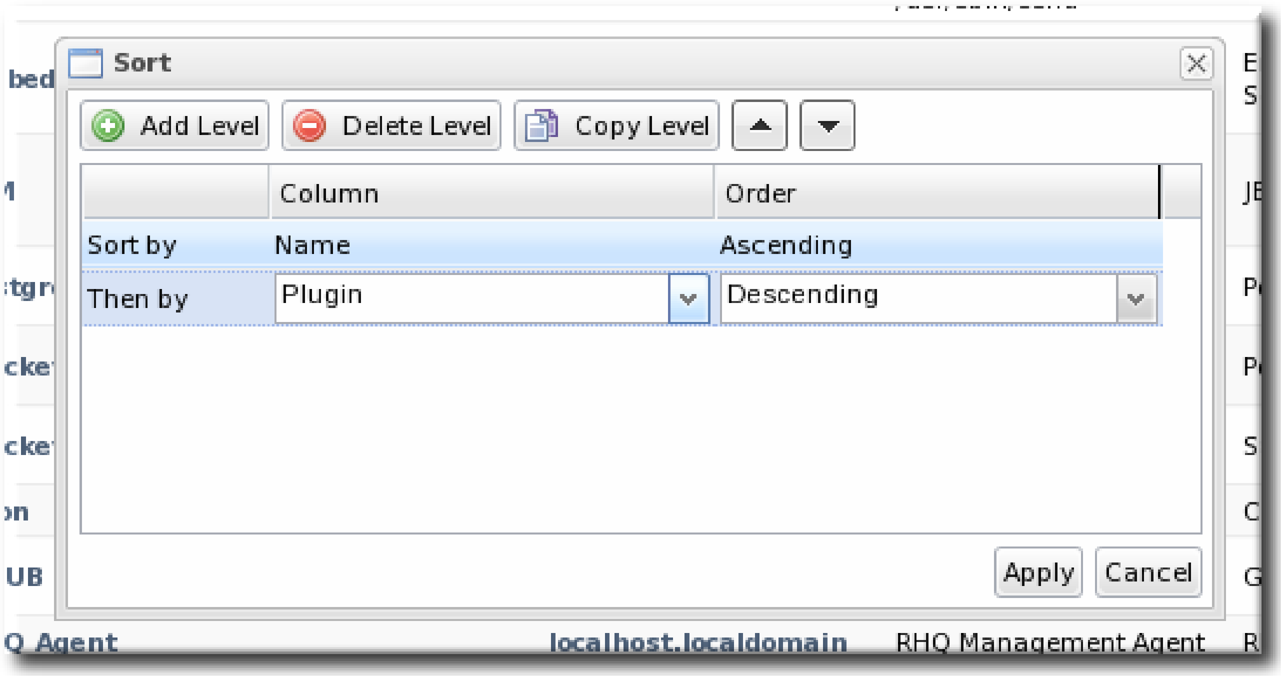


Figure 1.15. Advanced Table Sorting on the Server Resources List

The sort order can even be prioritized by specifying multiple criteria. For example, resources can be sorted by name, then by plug-in, then by ID. Since resources have standardized names, sorting by name or parent alone may not be specific enough to give a meaningful order to the entries; providing multiple, prioritized criteria can make the table display more accurate.



	Column	Order
Sort by	Name	Ascending
Then by	Plugin	Descending

Figure 1.16. Changing the Sort Method

1.7. CUSTOMIZING THE DASHBOARD

The Dashboard is configurable. It is composed of individual portlets, and these portlets can be rearranged or independently refreshed through the icon menu displayed on each portlet. There can even be multiple Dashboards, with different portlets, which can be used to give different and specific views into JBoss ON and its resources.

NOTE

Dashboards are configured per user, not globally.

1.7.1. Editing Portlets

To move portlets within the Dashboard layout, use the arrows in the portlet tool bar. To get rid of a portlet in a current, click the minimize icon on the far left to collapse it or click the X icon on the far right to delete the portlet from the Dashboard entirely. Some types of portlet allow customization, which can be accessed by clicking the wrench icon.

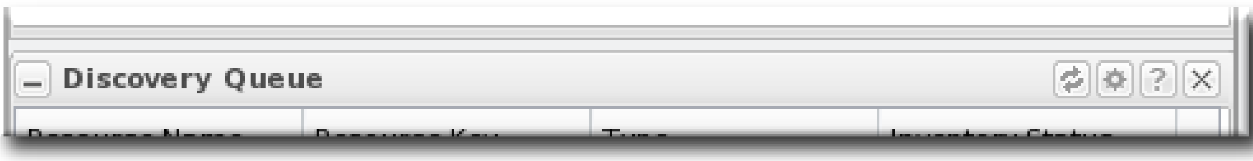
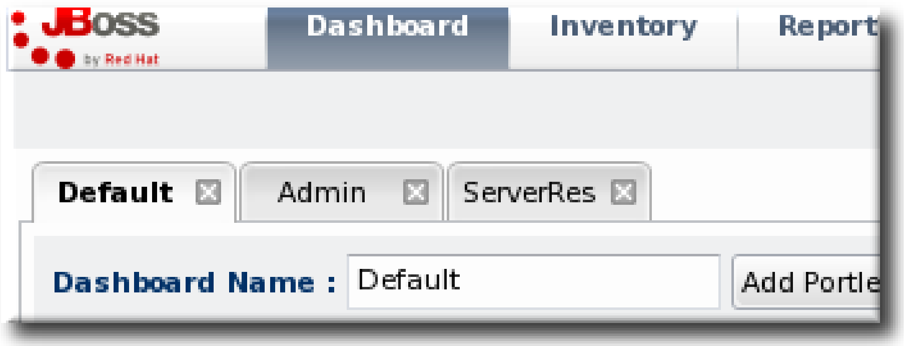


Figure 1.17. Portlet Icons

1.7.2. Adding and Editing Dashboards

The Dashboard page can actually contain multiple Dashboards, each with different portlets, column layouts, and refresh intervals. This makes it possible to get a logical grouping of information for a very fast assessment of the state of resources in JBoss ON. When multiple Dashboards are configured, they are displayed as tabs in the UI.

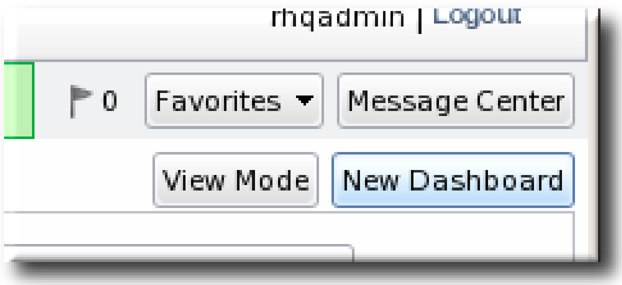


Dashboard Name : Default

Figure 1.18. Tabbed Dashboards

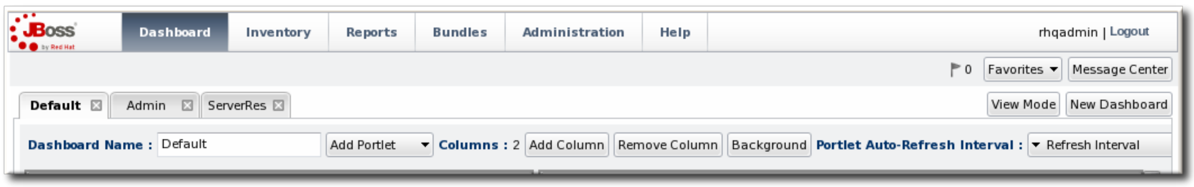
To add a new Dashboard:

1. Click the **`New Dashboard`** button in the far right of the main Dashboard.



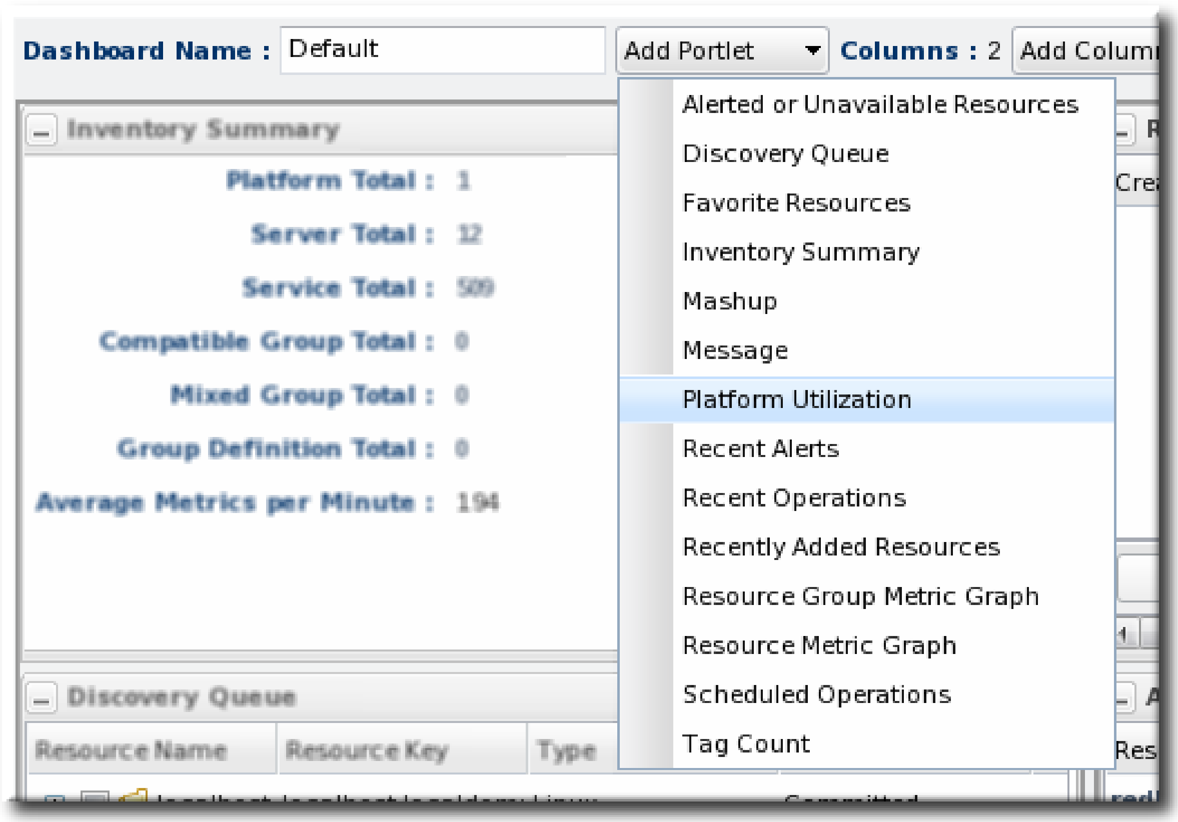
> **NOTE**
>
> The process of editing and adding Dashboards is very similar. The only difference is that to edit a Dashboard, you click the **`Edit Mode`** button.

2. The new Dashboard opens in the edit mode. Enter a name for the new Dashboard.



Dashboard Name : Default
Columns : 2
Portlet Auto-Refresh Interval :

3. Add the desired portlets to the Dashboard. If necessary, change the number of columns to fit the number of portlets.



1.8. SETTING FAVORITES

Using favorites makes it easy to navigate to resources that administrators need to access routinely for configuration updates, monitoring, or alerting.

Each resource has a small ribbon icon in the upper right corner of its details page. Clicking that icon automatically adds it to the resource favorites list.

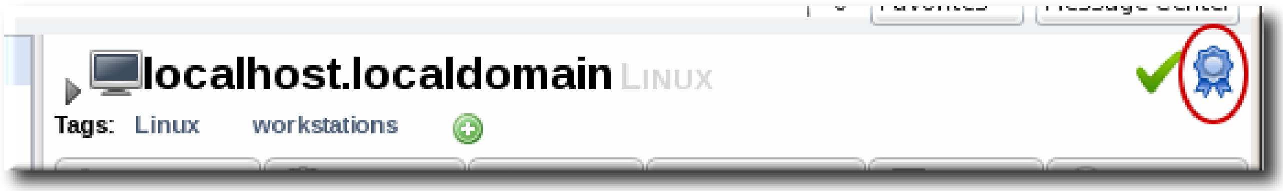


Figure 1.19. Favorites Icon

The resource and group favorites are listed in the **`Favorites`** in the shortcuts on the right of the top menu. Clicking a resource on that list automatically opens its details page without having to search for the resource. Because multiple resources may share a name or some properties, the Favorites list includes a hover with more details about the resource so you can select the right one.

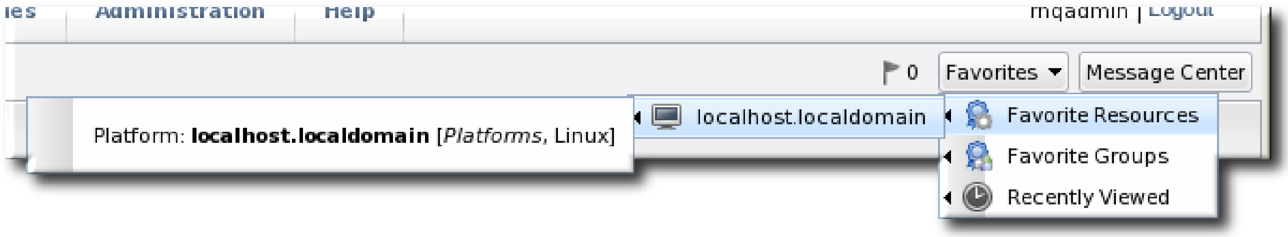
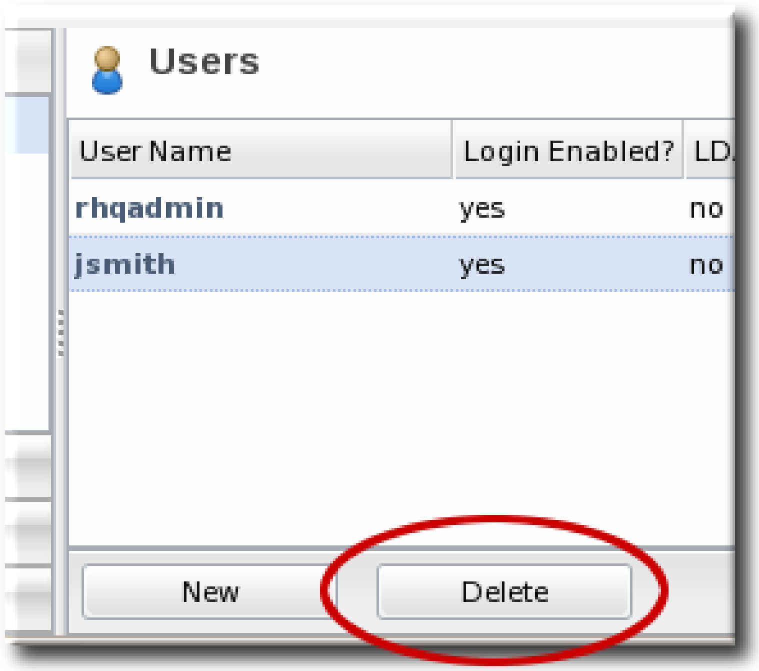


Figure 1.20. Favorites List

1.9. DELETING ENTRIES

Resource-related entries can be deleted through the inventory browser or group browser. Most JBoss ON server configuration entries cannot be deleted. Only user-supplied elements, like plug-ins, content, roles, and users, can be deleted.

If an item can be deleted, then a delete button is available in the table list or details page for that item.



User Name	Login Enabled?	LD
rhqadmin	yes	no
jsmith	yes	no

Figure 1.21. Delete Button in the Area Browser

NOTE

A user may have the right to *change* something, but that does not implicitly grant the right to *delete* something. For example, users with the configuration write permission can edit resource configuration and view configuration history and settings, but they cannot delete elements in the configuration history. Similar constraints are true for users with permission to create and edit operations and alerts – there is no right to delete elements in the resource history.

Deleting elements in the history requires the manage inventory permission.

CHAPTER 2. DYNAMIC SEARCHES FOR RESOURCES AND GROUPS

The inventory area has a dynamic search to look for resources and groups.

The dynamic search is an additional tool that can help manage your JBoss ON resources. Dynamic searches in JBoss ON can be saved to provide fast and reproducible snapshots of your JBoss ON deployment that match criteria that are relevant to your infrastructure, a kind of quick report.

A dynamic search checks both resources and groups (recursively into group members, as well) much more effectively than either a subsystem views search or a quick search. A search can begin against a specific identifying attribute of a resource (such as its name, parent, type, or JBoss ON category) and then has rules that can set how the search handles the string. Multiple search parameters can be strung together to make precise and complex searches. Dynamic searches can be saved and reused later so their results are reliably reproducible. (Section 2.2, “About the Dynamic Search Syntax” covers the details more.)

There are other aspects of dynamic searches like the autocomplete, hints, and highlight search strings that make it easier to use effectively than the limited substring and quick searches. These are covered in Section 2.1, “About Search Suggestions”.

2.1. ABOUT SEARCH SUGGESTIONS

Dynamic searches are extremely powerful, past simply finding resources. Dynamic searches can run through values in a number of different resource traits, not only the resource name. Dynamic searches can even be saved, so they're repeatable and can be used as ad hoc reporting.

Dynamic searches are easy to use because of search suggestions. A drop-down menu for every search provides three different types of suggestions:

- Saved searches, which contain previous custom search strings and a count of resources which match that search
- Query searches, which provide prompts for available resource traits
- Text searches, which provide a list of resources based on some property in the resource which matches the text prompt

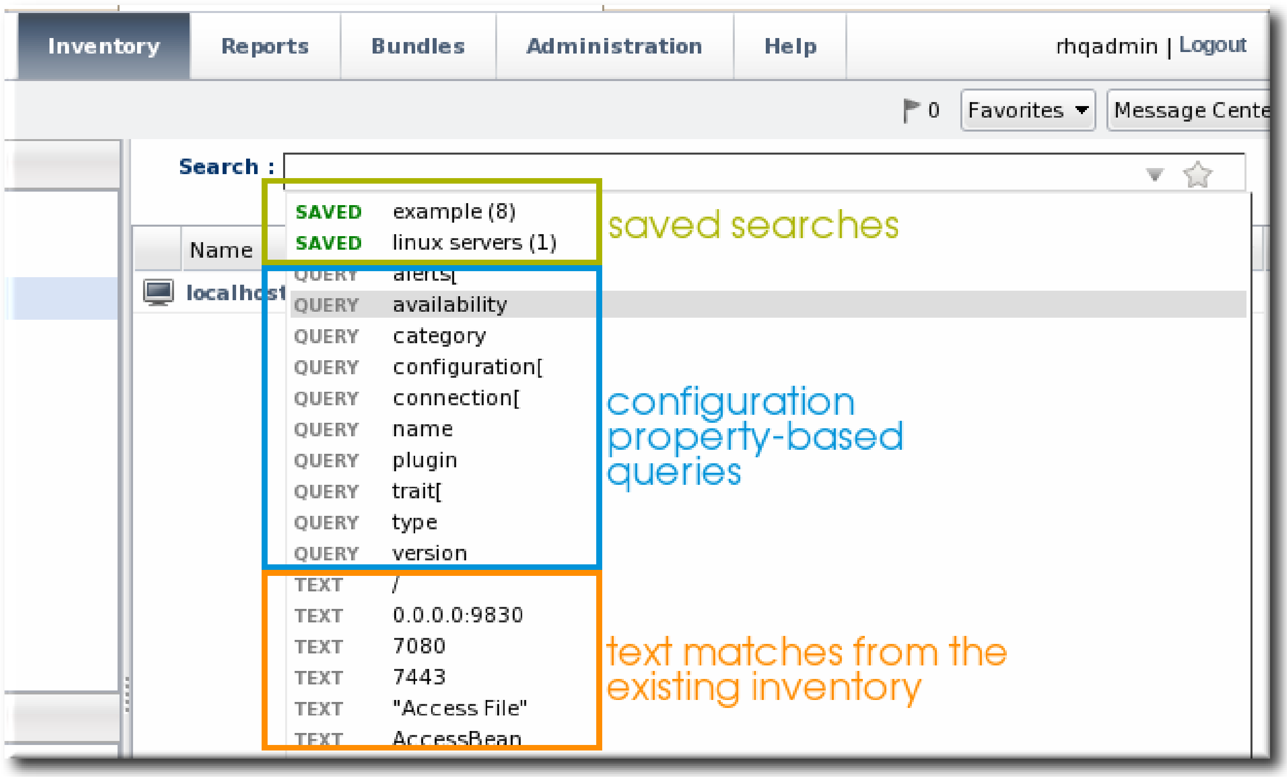


Figure 2.1. Types of Search Suggestions

When search terms are entered in the field, the matching substrings in possible matching resources are highlighted. By default, the suggestions can match any substring in the resource or in resource configuration traits. The suggestions can be limited to match the string at the beginning or end of the matching attribute using different operators (covered in Section 2.2, "About the Dynamic Search Syntax").

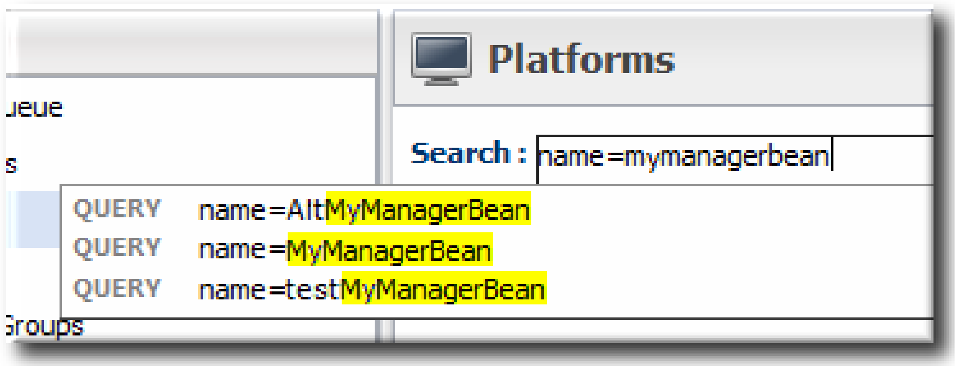

Platforms
Search : name=mymanagerbean
QUERY name=AltMyManagerBean
QUERY name=MyManagerBean
QUERY name=testMyManagerBean
ueue
s
Groups

Figure 2.2. Highlighting Search Terms

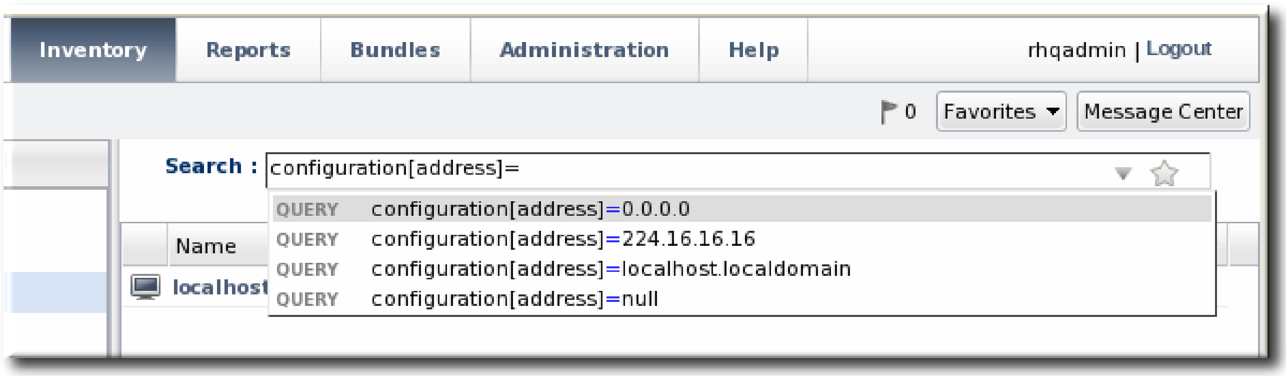
2.2. ABOUT THE DYNAMIC SEARCH SYNTAX

JBoss ON has its own search syntax for dynamic searches. The syntax is supposed to be relatively simple while covering a wide array of search-able items and allowing different phrases to be coupled together.

The basic dynamic search matched whatever text is entered in the search box in a general substring search. The search can allow a more detailed and targeted syntax, in this form:

```
[search_area].[search_property] operator value operator additional_search
```

The *search_area* identifies what type of entry — resource or group — is being searched for. This is an optional value because the search area is implied by the location of the search; i.e., searching in the **Resources** area implies a resource search, so it's not necessary to include the `resource.` part of the search.



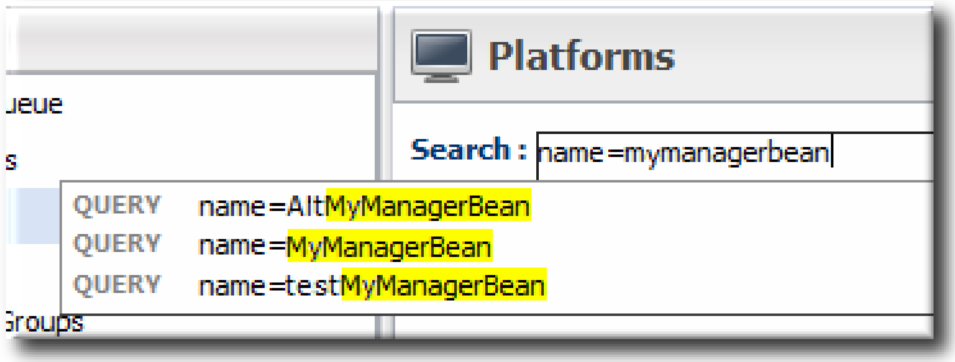
configuration[address]=

QUERY configuration[address]=0.0.0.0
QUERY configuration[address]=224.16.16.16
QUERY configuration[address]=localhost.localdomain
QUERY configuration[address]=null

Figure 2.3. Searching by Resources Traits

2.2.1. Basic String Searches

The simplest search with the dynamic search is a substring search. The given search term can match any part or all of the returned value.



name=mymanagerbean

QUERY name=AltMyManagerBean
QUERY name=MyManagerBean
QUERY name=testMyManagerBean

Figure 2.4. Matching the Search Term

The search term **agen** has search suggestions that match (case-insensitive) every resource that has that string anywhere in its name, at the beginning (**`Agent Plugin Container`**), middle (**`RHQ Agent Launcher Script`**), or end (**`rhq_agent`**). Likewise, the results include every resource with that string, regardless of where it appears in the resource entry or attribute.

When multiple words or terms are used together, the search treats them each as individual search terms in an implied AND search. (These complex searches are covered more in Section 2.2.3, “Complex AND and OR Searches”.) However, there can be instances when a multi-word search term should be treated as a single search term, and, for that, the dynamic search allows quotation marks.

IMPORTANT

Dynamic searches do *not* support wildcard characters like asterisks (`*`) or regular expressions.

The search syntax can use either single (') or double (") quotation marks to enclose a search term that includes whitespace. (Obviously, search terms without whitespaces do not require quotes.) If a search term has a series of space-separated words, then each word is treated as a separate search term in an AND search. For example:

```
postgres table myexampletable
```

Using quotation marks tells the search to treat that as a literal phrase rather than individual search terms. As with other search terms, phrases can be strung together in a space-separated list:

```
"My Compatible Group"
'test box'
plugin=jboss 123.4.5.6
trait[partitionName]='my example group' server.example.com
```

If a search term contains double or single quotation marks in the name, then the other type of boundary character must be used. For example, this group name contains a single quotation mark (apostrophe) character:

```
name="Production's Main Group"
```

Make sure to use the same opening and closing term character (you cannot mix single and double quotation marks.) Also, a single quotation mark is only treated as a search definition character when it occurs at the beginning of a search term. **`'Hello world`** requires a closing single quote; **`Production's`** does not.

A search can also look for a string where it occurs within a result. For this, the search allows *boundary characters* that set whether a string occurs at the beginning or end of a value or if it must be an exact match to the value.

The caret (^) character sets that a search term must appear at the beginning of the result string. For example:

```
resource.id=^100
```

This means that only resources with an ID which starts with 100 will be returned in the search.

Likewise, a string may occur only at the end of a value. The boundary character for an end value is a dollar sign ($). For example:

```
script$
```

This returns any resource with any value that ends in *script*.

Using both a caret and a dollar sign, together, means that the matching result must exactly match the search term, with no additional characters.

The search characters in Table 2.1, “String Operators” limit the search string by setting where in the result value the string appears.

Table 2.1. String Operators

Operator	Description
string	The string can occur anywhere in the result string.

Operator	Description
^*string*	The given string must appear at the beginning of the result value.
string$	The given string must appear at the end of the result value.
^*string*$	The result must be an exact match of the given string, with no leading or trailing characters.

NOTE

The dynamic search syntax treats null as an acceptable value for a search. Running a search which passes null will look for any resource or group with a null value for that property:

```
resource.trait[Database.startTime] = null
```

However, putting quotation marks around the term **`null`** will look for a resource or group with a value of the string null:

```
name = "null"
```

2.2.2. Property Searches

The search can be narrowed by looking for a specific value or type of attribute in the entry by using a search property. For example, looking for a resource with a CPU usage of 80% (**`trait`**) is different than looking for an entry with an ID that includes 80 (**`id`**). The available properties are listed in Table 2.2, "Resource Search Contexts" and Table 2.3, "Group Search Contexts" .

NOTE

It's possible to search using group criteria in the resource search, and the reverse, by specifying the search area and the appropriate properties. For example, it's possible to do a search in the groups area to return the list of groups that a specific resource belongs to. This is done by explicitly passing the search context and search property. For example, in the **`Groups`** page, to list any group which contains a resource managed by the Postgres plug-in:

```
resource.type.plugin = Postgres
```

IMPORTANT

The parameter suggestions for **`connection`**, **`configuration`**, and **`trait`** use the *internal property names* for the property names (**`connection[`***property_name***`]`**) rather than the names used in the JBoss ON GUI.

Table 2.2. Resource Search Contexts

Property	Description
resource.id	The resource ID number assigned by JBoss ON.
resource.name	The resource name, which is displayed in the UI.
resource.version	The version number of the resource.
resource.type.plugin	The resource type, defined by the plug-in used to manage the resource.
resource.type.name	The resource type, by name.
resource.type.category	The resource type category (platform, server, or service).
resource.availability	The resource availability, either UP or DOWN.
resource.pluginConfiguration[*property-name*]	The value of any possible configuration entry in a plug-in.
resource.resourceConfiguration[*property-name*]	The value of any possible configuration entry in a resource.
resource.trait[*property-name*]	The value of any possible measurement trait for a resource.

There are slightly fewer search properties for groups, since groups have simpler entries than resources.

Table 2.3. Group Search Contexts

Property	Description
group.name	The name of the group.
group.plug-in	For a compatible group, the plug-in which defines the resource type for this group.
group.type	For a compatible group, the resource type for this group.
group.category	The resource type category (platform, server, or service).
group.kind	The type of group, either mixed or compatible.

Property	Description
group.availability	The availability of resource in the group, either UP or DOWN.

The *operator* first refers to how the results should match the search string (*value*). This can require an exact match, every value but the one given in the search string. The *operator* then refers to how multiple search strings relate to each other (AND or OR); both explicit AND and OR statements and parenthetical statements are allowed. Complex searches are covered in Section 2.2.3, “Complex AND and OR Searches”.

Table 2.4. Search String Operators

Operator	Description
=	Case-insensitive match.
==	Case-exact match.
!=	Case-insensitive negative match (meaning, the value is *not* the string).
!==	Case-exact negative match (meaning, the value is *not* the string).

2.2.3. Complex AND and OR Searches

The dynamic search bar assumes that each individual word is a search term (unless terms are defined using quotation marks). Implicitly, multi-word searches are treated as AND searches. For example:

```
postgres server myserver
```

This is treated as a series of AND terms:

```
postgres AND server AND myserver
```

The dynamic search also allows OR searches, with terms separated by a pipe (**|**). For example:

```
postgres | jbossas
```

Both AND and OR searches can be entered, and complex searches can be written by stringing multiple search strings together. When there are both AND and OR search criteria, the AND terms are processed first. For example, this search term searches for both B and C, and then either A or B/C.

```
a | b c
```

NOTE

When there are both AND and OR terms used in a complex search, AND terms are given preference. However, terms in parenthesis are evaluated even before AND expressions, so parentheses can be used to override the natural search preference.

Search phrases can be nested to multiple levels using parentheses to group search terms. These parentheses can also be used to override the preferences for AND matches, forcing at least some OR expressions to be processed first. For example, this expression searches for the OR terms first, matching *a* OR *b* and *c* OR *d*, and then running an AND search on the results of the two OR searches:

```
(a | b) (c | d)
```

The results will contain several combinations of values: *a c*, *a d*, *b c*, and *b d*.

Multiple levels of nesting are allows. For example, this expression requires *a* AND either *b* OR *c* AND *d*:

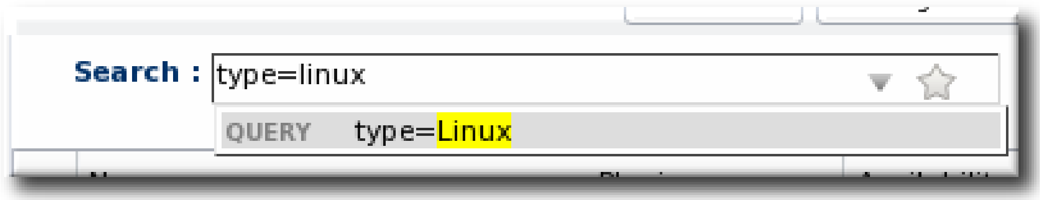
```
(a) (b | (c d))
```

The matching resources, then, can contain values matching *a c d* or *a b*.

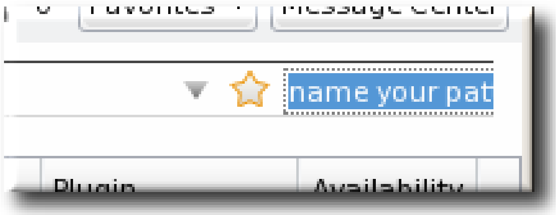
2.3. SAVING, REUSING, AND DELETING DYNAMIC SEARCHES

Dynamic searches can be saved, which makes it much easier to reuse complex or common searches. To save a search:

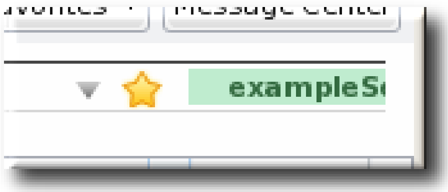
1. Run the search.



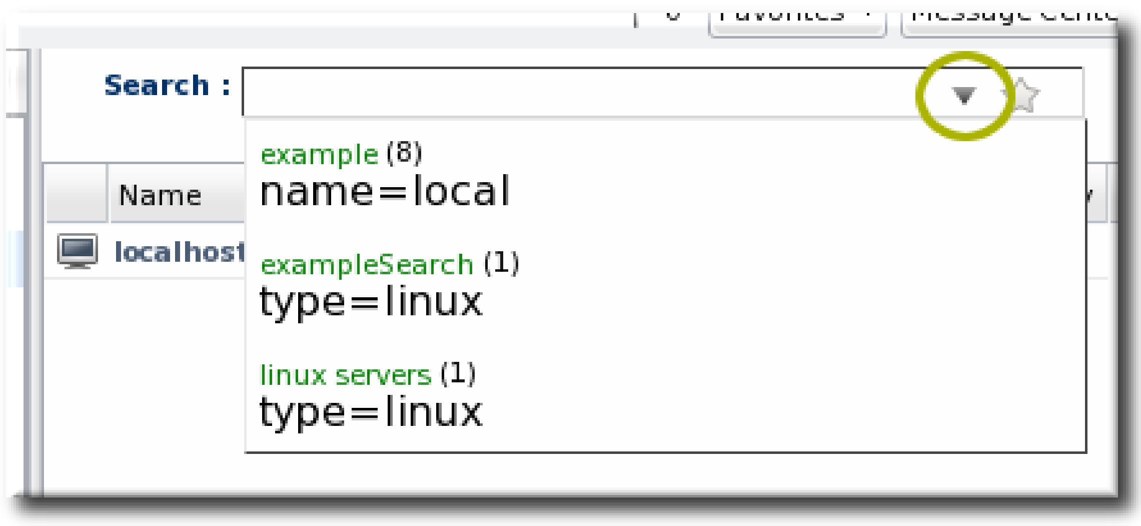
2. Click the star in the right of the search bar. When the field comes up, enter the name for the new search.



The search name is then displayed in green.



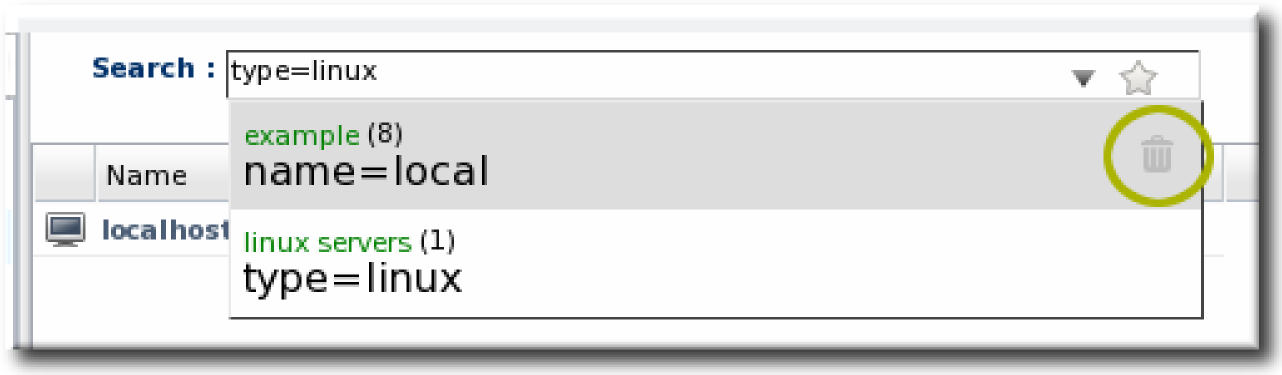
Saved searches are listed with other search results whenever the dynamic search criteria match any part of the saved search string and immediately whenever the dynamic search field is active, before search parameters are entered. The name of the search is shown in bright green. The drop-down option shows the string used in the saved search in black text beneath the name, to make it clear what the parameters for the search are.



example (8)
name=local
exampleSearch (1)
type=linux
linux servers (1)
type=linux

To edit a saved search, select the search from the list of suggestions and then click the gold star. This deletes the search, but leaves the previous search settings in the search box. This allows the search parameters to be edited and then re-saved.

To delete a search, simply click the gold star or the trashcan icon by the search name when it is highlighted in the list. It is immediately removed from the saved searches list.



type=linux

example (8)
name=local
linux servers (1)
type=linux

CHAPTER 3. VIEWING AND EXPORTING REPORTS

The purpose of JBoss ON is to deliver information about the resources in your infrastructure. There are a lot of places in the UI where information is displayed for individual resources or for defined groups.

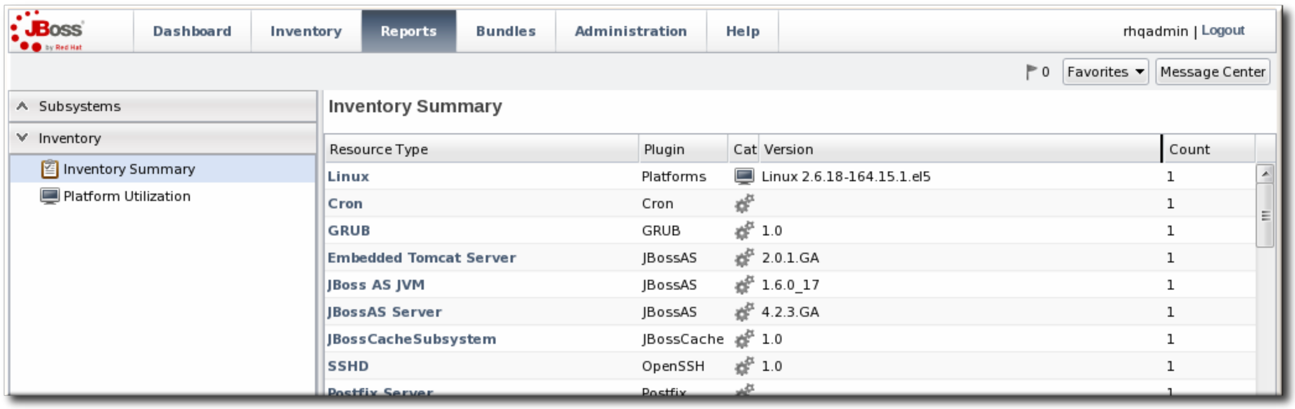
The **`Reports`** main tab has a list of predefined searches and views into different areas for *all* resources, not just a subset.

3.1. TYPES OF REPORTS

All reports show a list of information that spans everything in the JBoss ON inventory. Reports are broken into two categories, one for JBoss ON server subsystem areas and one for different inventory counts.

Subsystem reports are related to *functionality* in JBoss ON, for different aspects of monitoring, alerting, drift, and configuration. Most subsystem reports have some corollary to a resource-level chart, with much the same information displayed. Subsystem reports include additional columns to list the resource name and the resource ancestry (parent and grandparent resources) to disambiguate each resource, since names are not unique.

Inventory reports give counts. These reports usually begin with a breakdown by resource type, with additional resource lists available as "subreports."



Inventory Summary

Resource Type	Plugin	Cat	Version	Count
Linux	Platforms		Linux 2.6.18-164.15.1.el5	1
Cron	Cron			1
GRUB	GRUB		1.0	1
Embedded Tomcat Server	JBossAS		2.0.1.GA	1
JBoss AS JVM	JBossAS		1.6.0_17	1
JBossAS Server	JBossAS		4.2.3.GA	1
JBossCacheSubsystem	JBossCache		1.0	1
SSHD	OpenSSH		1.0	1
Postfix Server	Postfix			1

Figure 3.1. Inventory Summary Report

Table 3.1. Types of Reports

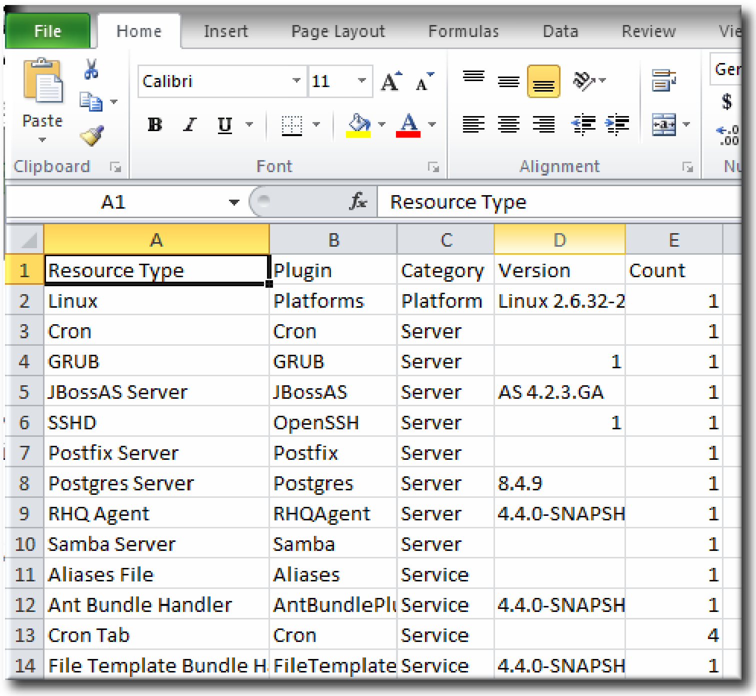
Report Name	Description	Has Filters?
Subsystem Reports		
Suspect Metrics	Lists any metrics outside the established baselines for a given resource. All suspect metrics for all resources are listed, but the baselines which mark the metric may be different for each resource, even different between resources of the same type.	No

Report Name	Description	Has Filters?
Configuration History	Lists all configuration changes, for all resources. Version numbers are incremented globally, not per resource. The configuration history shows the version number for the change, the date it was submitted and completed, its status, and the type of change (individual or through a group).	No
Recent Operations	Lists all operations for all resources, by date that the operation was submitted (not necessarily run), the operation type, and its status.	Yes
Recent Alerts	Lists every fired alert for all resources, with the name of the resource, the alert definition which was fired, and the alerting condition.	Yes
Alert Definitions	Lists all configured alert definitions, for all resources, with their priority and whether they are enabled.	No
Recent Drift	Contains a list of all snapshots, for all resources and drift definitions.	Yes
Inventory Reports		
Inventory Summary	Contains a complete list of resources currently in the inventory, broken down by resource type and version number.	No
Platform Utilization	Shows the current CPU percentage, actual used memory, and swap space.	No

Report Name	Description	Has Filters?
Drift Compliance	Shows a list of all resource types which support drift and then shows how many drift definitions are configured and whether the group is compliant. Clicking on a resource type shows the list of resources configured for drift and their individual compliance status.	No

3.2. EXPORTING REPORT DATA TO CSV

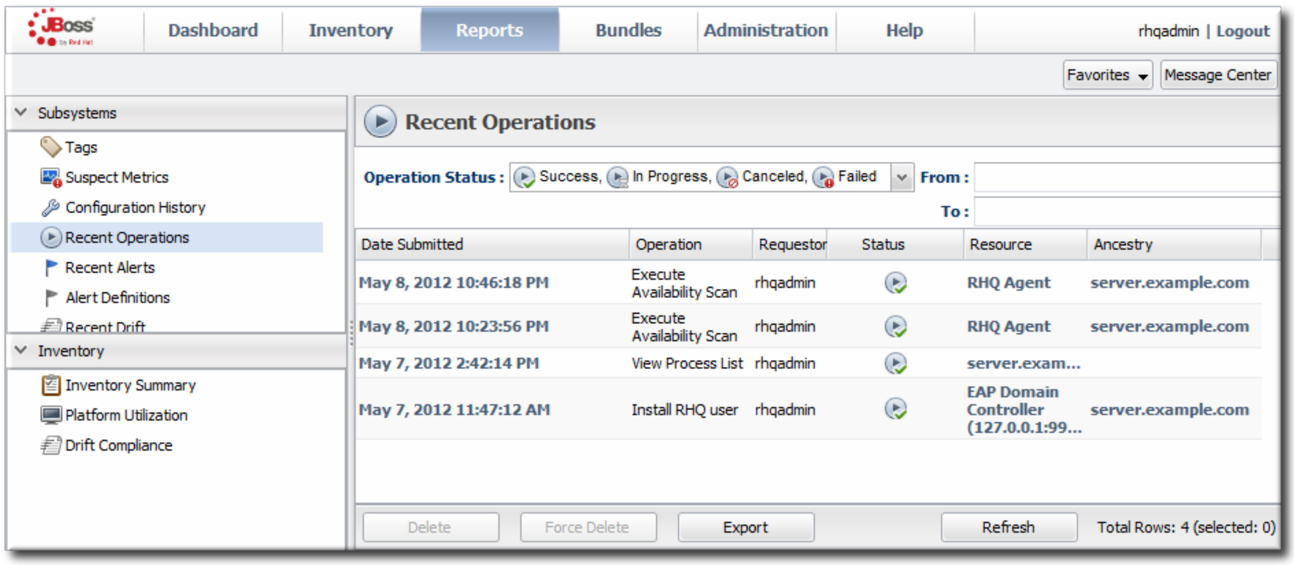
The `Reports` tab collects information that is not easily accessible in other parts of the GUI or even the CLI, without complex scripting. The information from any report can be exported to CSV simply by clicking the `Export` button.



	A	B	C	D	E
1	Resource Type	Plugin	Category	Version	Count
2	Linux	Platforms	Platform	Linux 2.6.32-2	1
3	Cron	Cron	Server		1
4	GRUB	GRUB	Server	1	1
5	JBossAS Server	JBossAS	Server	AS 4.2.3.GA	1
6	SSHD	OpenSSH	Server	1	1
7	Postfix Server	Postfix	Server		1
8	Postgres Server	Postgres	Server	8.4.9	1
9	RHQ Agent	RHQAgent	Server	4.4.0-SNAPSH	1
10	Samba Server	Samba	Server		1
11	Aliases File	Aliases	Service		1
12	Ant Bundle Handler	AntBundlePlı	Service	4.4.0-SNAPSH	1
13	Cron Tab	Cron	Service		4
14	File Template Bundle H	FileTemplate	Service	4.4.0-SNAPSH	1

Figure 3.2. Exported Inventory Summary

Only the information displayed in the report, as displayed in the report, is exported to CSV. If there is a certain sort order applied to the report or if a filter is used to limit the displayed entries, that sort order and that filter are preserved in the exported report CSV file.



Recent Operations

Operation Status : Success, In Progress, Canceled, Failed From :

To :

Date Submitted	Operation	Requestor	Status	Resource	Ancestry
May 8, 2012 10:46:18 PM	Execute Availability Scan	rhqadmin	✓	RHQ Agent	server.example.com
May 8, 2012 10:23:56 PM	Execute Availability Scan	rhqadmin	✓	RHQ Agent	server.example.com
May 7, 2012 2:42:14 PM	View Process List	rhqadmin	✓	server.exam...	
May 7, 2012 11:47:12 AM	Install RHQ user	rhqadmin	✓	EAP Domain Controller (127.0.0.1:99...	server.example.com

Total Rows: 4 (selected: 0)

Figure 3.3. Report with Date Filters

PART I. INVENTORY, RESOURCES, AND GROUPS

CHAPTER 4. INTERACTIONS WITH SYSTEM USERS FOR AGENTS AND RESOURCES

The agent runs as a specific system user, and so do servers such as JBoss and Apache which are managed by JBoss ON. The general assumption with many of the agent management tasks, including discovery, is that the agent user is the same as the resource user. If the users are different, then that can have an impact on how resources can be discovered and managed.

The common types of servers which JBoss ON manages are:

- JBoss EAP servers
- PostgreSQL databases
- Tomcat servers
- Apache servers
- Generic JVMs

For some management operations initiated by the JBoss ON agent, the agent system user is never even involved. For example, the JBoss EAP plug-in connects to the EAP instance using authentication mechanisms managed by JBoss EAP itself, so no system ACLs or user permissions are required. As long as the user can access the JBoss EAP instance, everything works.

Table 4.1. Cheat Sheet for Agent and Resource Users

Resource	User Information
PostgreSQL	No effect for monitoring and discovery. The agent user must have read/write permissions to the PostgreSQL configuration file for configuration viewing and editing.
Apache	No effect for monitoring and discovery. The agent user must have read/write permissions to the Apache configuration file for configuration viewing and editing.
Tomcat	Must use the same user or can't be discovered
JMX server or JVM	Different users are fine when using JMX remoting; cannot be discovered with different users and the attach API
JBoss AS/EAP	Different users are all right, but requires read permissions on run.jar and execute and search permission on all ancestor directories for run.jar

4.1. THE AGENT USER

There is a general assumption that the agent runs as the same user as the managed resources, and this is the cleanest option for configuration.

When the JBoss ON agent is installed from the agent installer JAR file, the system user and group who own the agent installation files is the same user who installs the JAR. So, a special system user can be created or selected, and then the agent can be installed by that user.

4.2. AGENT USERS AND DISCOVERY

An agent discovers a resource by searching for certain common properties, such as PIDs and processes or start scripts.

It does not necessarily matter whether the agent has superior privileges as the resource user.

For most resources, the agent simply requires read access to that resource's configuration. For resources like Apache and Postgres, as long as the agent can read the resource configuration, the resources can be discovered.

For some other resources, the agent user has to have very specific permissions:

- For JBoss EAP resources, the agent must have read permissions to the **`run.jar`** file, plus execute and search permissions for every directory in the path to the **`run.jar`** file.
- When a JBoss EAP 6 instance is installed from an RPM, the agent user must belong to the same system group which runs the EAP instance. This is **`jboss`**, by default.
- Tomcat servers can only be discovered if the JBoss ON agent and the Tomcat server are running as the same user. Even if the agent is running as root, the Tomcat server cannot be discovered if it is running as a different user than the agent.
- If a JVM or JMX server is running with JMX remoting, then it can be discovered if the agent is running as a different user. However, if it is running with using the attach API, it has to be running as the same user as the agent for the resource to be discovered.

4.3. USERS AND MANAGEMENT TASKS

The system user which the agent runs as impacts several common agent tasks:

- Discovery
- Deploying applications
- Executing scripts
- Running start, stop, and restart operations
- Creating child resources through the JBoss ON UI
- Viewing and editing resource configuration

The key thing to determine is what tasks need to be performed and who needs to perform that operation, based on limits on the resource or the operating system for permissions or authorization.

For some actions — discovery, deploying applications, or creating child resources — setting system ACLs that grant the agent user permission are sufficient.

For running operations or executing scripts, it may be necessary to run the task as a user other than the agent user. This can be done using **sudo**.

Whatever method, the goal is to grant the JBoss ON user all of the required system permissions necessary to carry out the operations.

4.4. USING SUDO WITH JBOSS ON OPERATIONS

The time to use **sudo** is for long-running operations, such as starting a service or a process, or for scripts which are owned by a resource user. The user which executes the script should be the same as the resource user because that user already has the proper authorization and permissions.

The user can really be the same, or the JBoss ON user can be granted **sudo** rights to the given command.

When elevating the agent user's permissions, two things must be true:

- There can be no required interaction from the user, including no password prompts.
- It should be possible for the agent to pass variables to the script.

To set up **sudo** for resource scripts:

1. Grant the JBoss ON agent user **sudo** rights to the specific script or command. For example, to run a script as the **`jbossadmin`** user:

   ```
   [root@server ~]# visudo

   jbosson-agent     hostname=(jbossadmin)  NOPASSWD: /opt/jboss-
   eap/jboss-as/bin/*myScript*.sh
   ```

   Using the **`NOPASSWD`** option runs the command without prompting for a password.

   IMPORTANT

   JBoss ON passes command-line arguments with the start script when it starts an EAP instance. This can be done either by including the full command-line script (including arguments) in the **`sudoers`** entry or by using the **`sudo -u`** *user* command in a wrapper script or a script prefix.

   The second option has a simpler **`sudoers`** entry

2. Create or edit a wrapper script to use. Instead of invoking the resource's script directly, invoke the wrapper script which uses **sudo** to run the script.

   NOTE

   For the EAP start script, it is possible to set a script prefix in the connection settings, instead of creating a separate wrapper script:

   ```
   /usr/bin/sudo -u jbosson-agent
   ```

   For example, for a start script wrapper, **`start-myScript.sh`**:

   ```
   #!/bin/sh
   ```

```
# start-myScript.sh
# Helper script to execute start-myConfig.sh as the user jbosson-
agent
#
sudo -u jbosson-agent /opt/jboss-eap/jboss-as/bin/start-myConfig.sh
```

3. Create the start script, with any arguments or settings to pass with the **`run.sh`** script. For example, for **`start-myConfig.sh`**:

```
nohup ./run.sh -c MyConfig -b jonagent-host 2>&1> jboss-MyConfig.out
&
```

CHAPTER 5. MANAGING THE RESOURCE INVENTORY

The *inventory* in JBoss ON is the repository that contains all of the servers and applications that are managed or monitored by JBoss Operations Network. The inventory tells JBoss Operations Network which resources it can manage.

Once in the inventory, resources can be organized in several different ways. Resources can be grouped automatically by their type in autogroups, resources can be added manually to user-defined groups, and they can be added manually to another resource as a child.

This section covers the process of identifying and importing resources through discovery, adding children, and managing groups.

5.1. ABOUT THE INVENTORY: RESOURCES

The JBoss ON *inventory* is the central list of every managed resource that is recognized by the JBoss ON server.

5.1.1. Managed Resources: Platforms, Servers, and Services

Each JBoss ON agent periodically scans the platform where it's installed to check for services and servers. That is the *discovery* process. When a potential resource is discovered, then it is listed in the discovery scan results, and, from there, an administrator can choose whether it should be managed by JBoss ON. If a resource should be managed, then it must be *imported* into the JBoss ON server's inventory; otherwise, it can be ignored.

There are three categories of resources in JBoss ON:

1. Platforms (operating systems)

2. Servers

3. Services

The resource hierarchy in the JBoss ON inventory mimics how programs and processes are physically structured on a platform. The highest level is the platform. The platform can have both servers and services as children. Likewise, servers can have both other servers and services as children, while services can have only other services as children within the inventory.

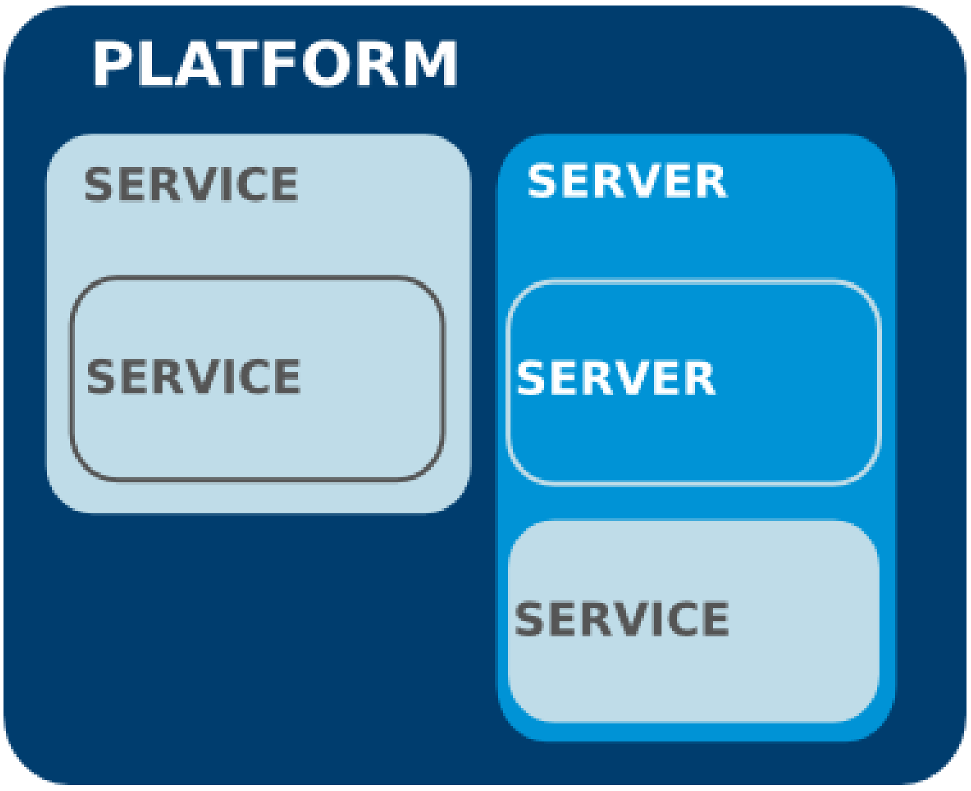


Figure 5.1. An Example Resource Hierarchy

A handful of rules govern the relationships between resources in inventory:

- A resource can only have one parent.
- A server can be a child of a platform (such as JBoss AS on Linux) or another server (such as Tomcat embedded in JBoss AS).
- A service can be a child of a platform, a server (such as the JMS queue on JBoss AS), or another service (e.g. a table inside a database).
- Platforms, servers, and services can have many children services.

JBoss ON can manage many different types of resources; each managed resource has a corresponding agent plug-in which defines things like the available monitoring metrics, operations, and supported versions for the resource.

NOTE

Additional resources can be added by writing custom plug-ins for the resource type.

5.1.2. Content-Backed Resources

For application servers, there is a close conceptual link between a child resource and content which is deployed on the server. For example, EARs and WARs are both child resources of application servers and are versioned content which can be stored in a repository, selectively deployed, and reverted.

A *content-backed resource* is treated as a resource in that it has a place in the inventory hierarchy, can have operations run against it, and can have metrics collected for it. However, it is also managed as a software package, with bits that are uploaded and stored in the JBoss ON content system and maintained within a repository.

Content-backed resources can be manually added as children or, if deployed outside JBoss ON, can be detected in a package discovery scan (which runs every 24 hours by default). These child resources can also be created and updated by deploying content from the repositories in JBoss ON.

For more information on managing content-backed resources, see Chapter 31, *Managing JBoss EAP 6 (AS 7)* and Chapter 32, *Managing JBoss EAP 5*.

IMPORTANT

Content-backed resources can have a significant impact on disk space requirements.

JBoss ON stores all versions of content. This is part of versioning control, allowing changes to content-backed resources to be reverted and managed.

Therefore, the system which hosts the backend database (Oracle or PostgreSQL) must have enough disk space to store all versions of all content for any resource. Additionally, the database itself must have adequate tablespace for the content.

When calculating the required amount of space, estimate the size of every artifact, and then the number of versions for each artifact. At a minimum, **have twice that amount of space available**; both PostgreSQL and Oracle require twice the database size to perform cleanup operations like vacuum, compression, and backup and recovery.

5.1.3. Resources in the Inventory Used by JBoss ON

Some resources are automatically added to platforms to enable certain JBoss ON-specific functionality. For example, an Ant bundle handler resource is added to platforms as a child service to allow the agent to identify and process Ant recipes. [1] Without that Ant bundle handler resource, the JBoss ON agent cannot perform provisioning on that platform. Administrators do not have to interact directly with JBoss ON-specific child resources once they are in the inventory, but these child resources must be present for JBoss ON functionality to work. Because these children are required by JBoss ON, they are imported automatically with the platform.

Other resources can be added to the inventory for the JBoss AS server used by the JBoss ON server, the JBoss ON agent, and JBoss elements associated with the agent and server such as the agent JVM, JBoss Cache, and the agent launch script and **`rhq-agent-env.sh`** script. Adding these resources to the JBoss ON inventory allows JBoss ON to monitor and manage all of the agents and servers in the deployment.

5.2. DISCOVERING RESOURCES

Before any application or platform can be managed by JBoss ON, it must be imported into the inventory. There are different ways of adding resources to the inventory, depending on how the resource was discovered.

5.2.1. Finding New Resources: Discovery

When an agent is installed and every time it starts up, it *scans* the platform, and all applications on it, for any servers, services, or other items which can be included into the inventory. The process of finding potential resources is called *discovery*.

There are different scans for each type of resource: platform, server, and service. High level scans for servers and platforms are initiated by the agent every 15 minutes. A service scan detects lower-level services that are running in servers that have already been imported into the inventory. These scans run by default every 24 hours. Both of these intervals are configurable in the JBoss ON agent configuration.

The agent always runs a scan for new resources when it starts up, and then periodically at its configured intervals. There is a delay between when the parent and immediate resources are imported and when another discovery scan is initiated to discover lower-level children; this prevents possibly long-running recursive discovery scans for resources which may potentially be ignored.

NOTE

All of the discovery scan intervals are configurable in the agent's configuration file.

JBoss ON agents send information about the platform and servers it discovers back to the JBoss ON server.

A server must be imported into the inventory before any of its child processes, servers, or services can be detected by the discovery scan.

When a platform is imported into the inventory, several of its child servers and services are imported automatically as well. This includes resources that are vital to the platform (like CPU, network adapters, and filesystems) as well as resources that are used by the JBoss ON server itself (such as the Ant bundle handler resource, which is used by the provisioning subsystem).

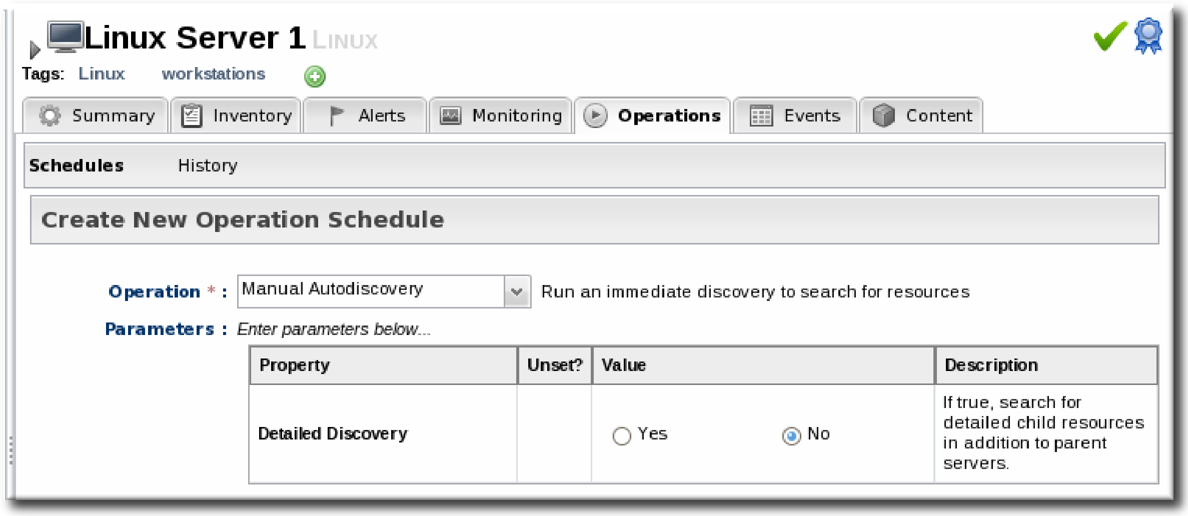
Although discovery is run automatically by the agent, discovery can also be initiated manually to capture infrastructure changes immediately.

5.2.2. Running Discovery Scans Manually

Discovery scans are run automatically by the agent to identify new resources as they are added to a platform. Server scans are run every 15 minutes and service scans every 24 hours. (Additionally, the agent runs a full discovery scan when it starts up.) New resources can be added between the discovery scans, so administrators can initiate a *manual* discovery scan apart from the scheduled discovery scan.

The simplest way to initiate a discovery scan is to run the agent's **`discovery`** command at the agent command prompt:

1. Click the **`Inventory`** tab in the top menu.
2. Open the **`Servers - Top Level Resources`** link on the left, and select the agent resource.
3. Open the **`Operations`** tab for the agent.
4. In the **`Schedules`** subtab, click the **`New`** button.
5. Select the **`Manual Discovery`** operation from the drop-down menu, and select whether to run a detailed discovery (servers and services) or a simple discovery (servers only).


Linux Server 1 LINUX

Tags: Linux workstations

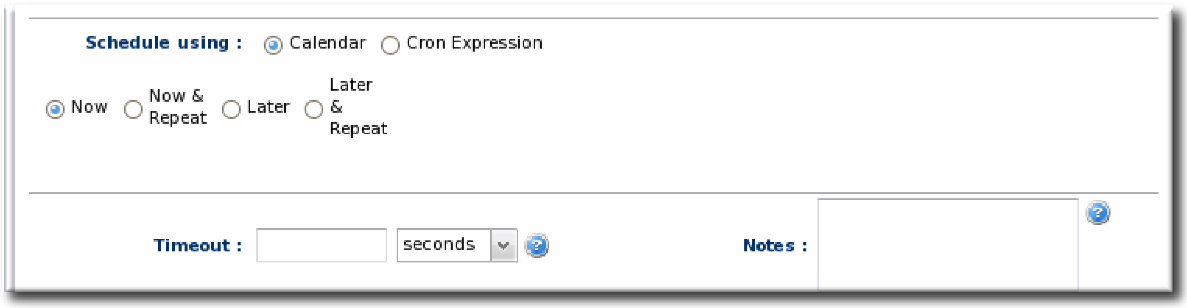
Create New Operation Schedule

Operation * : Manual Autodiscovery — Run an immediate discovery to search for resources

Parameters : *Enter parameters below...*

Property	Unset?	Value	Description
Detailed Discovery		Yes [✓ No]	If true, search for detailed child resources in addition to parent servers.

6. In the **Schedule** area, select the radio button to run the operation immediately.

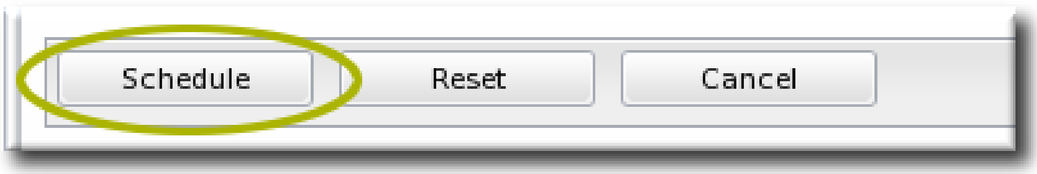

Schedule using : [✓ Calendar] Cron Expression

[✓ Now] Now & Repeat Later Later & Repeat

Timeout : seconds

Notes :

7. Click the **Schedule** button to set up the operation.



5.2.3. Importing Resources from the Discovery Queue

1. Click the **Inventory** tab in the top menu.

2. In the **Resources** menu on the left, select **Discovery Queue**.

3. Select the checkbox of the resources to be imported. Selecting a parent resource (such as a platform) gives the option to automatically import all of its children, too.

4. Click the **Import** button at the bottom of the UI.

5.2.4. Ignoring Discovered Resources

When the JBoss ON agent discovers an application or service which is to be ignored from the inventory, the server can be instructed to ignore those resources in the discovery queue. If the resource has already been imported to the inventory, see Section 5.2.5, “Ignoring Imported Resources”.

NOTE

A resource can only be ignored if its parent is *already* added to the inventory.

1. Select **`Inventory`** from the top menu.
2. Select the **`Discovery Queue`** item under the **`Resources`** menu on the left side of the screen.
3. Select the checkbox of the resource to be ignored. Selecting a parent resource automatically selects all of its children.
4. Click the **`Ignore`** button at the bottom of the page.

NOTE

It is not possible to ignore a platform. If a platform should not be in the inventory, do not run an agent on that machine.

5.2.5. Ignoring Imported Resources

Once a resource has been imported into the JBoss ON inventory, the server and the managing agent can be set to ignore those resources. There are three locations within the JBoss ON User Interface where you set a resource to ignore;

- **`Inventory Resources`** pages,
- the **`Inventory`** page of the parent resource, or
- the resource **`Groups`** inventory page.

NOTE

If the resource to be ignored has been discovered but not yet imported into inventory, see Section 5.2.4, “Ignoring Discovered Resources”.

NOTE

It is not possible to ignore a platform. If a platform should not be in the inventory, you should remove the JBoss ON *agent* on that machine.

5.2.5.1. Ignoring Resources from a `Resources` page

1. From the **`Inventory`** menu, select the relevant resource view under **`Resources`**. For example;
   - **`Inventory`** > **`Resources`** > **`All Resources`**, or
   - **`Inventory`** > **`Resources`** > **`Services`**.
2. Select the row containing the resource to ignore. Multiple resources can be selected if required.
3. Click the **`Ignore`** button at the bottom of the page.

5.2.5.2. Ignoring resources from the `Inventory` page of the parent resource

1. From the **`Inventory`** menu, select the relevant resource view under **`Resources`**. For example;
   - **`Inventory`** > **`Resources`** > **`All Resources`**, or
   - **`Inventory`** > **`Resources`** > **`Services`**.
2. Locate and select the parent resource from the resource list.
3. Within the parent resource page, select the **`Inventory`** *tab*.
4. From the parent resources **`Inventory`** *tab*, select the **`Child Resources`** sub-tab.
5. Select the row containing the resource to be ignored from the **`Child Resources`** list. Multiple rows can be selected if required.
6. Click the **`Ignore`** button at the bottom of the page.

5.2.5.3. Ignoring resources from a `Groups` page

1. From the **`Inventory`** menu, select the relevant resource group under **`Groups`**. For example;
   - **`Inventory`** > **`Groups`** > **`All Groups`**, or
   - **`Inventory`** > **`Groups`** > **`Compatible Groups`**
2. Locate the resource group that contains the resource to be ignored.
3. Within the resource group page, select the **`Inventory`** *tab*.
4. From the resource groups **`Inventory`** *tab*, select the **`Members`** sub-tab.
5. Select the row containing the resource to be ignored from the **`Members`** list. Multiple rows can be selected if required.
6. Click the **`Ignore`** button at the bottom of the page.

5.2.6. Ignoring an Entire Resource Type

The procedure in Section 5.2.4, "Ignoring Discovered Resources" is for ignoring a single, specific resource after it has been discovered on the system. Other resources of the same type on that platform or other platforms will still be discovered and included in the inventory. Ignoring that one resource has no impact on the discovery process.

However, there may be certain types of resources which will never be imported into the inventory. For example, filesystem resources may always be ignored on platforms, or a low-level EJB service may always be ignored on an EAP 6 server. In those instances, there is a maintenance overhead to discovering resources on the agent, having an administrator manually ignore them, and then maintaining them in inventory as an ignored resource.

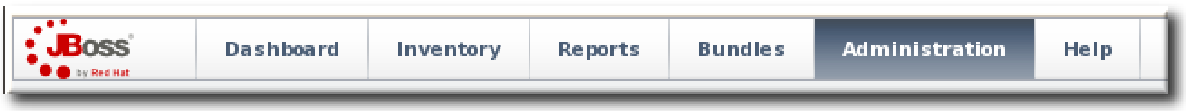
If a certain *type* of resource will never be monitored or managed by JBoss ON, then that entire resource type can be ignored. This essentially disables discovery for that resource type.

Ignoring a resource type is a global, JBoss ON-wide setting and is configured in the JBoss ON server configuration.

NOTE

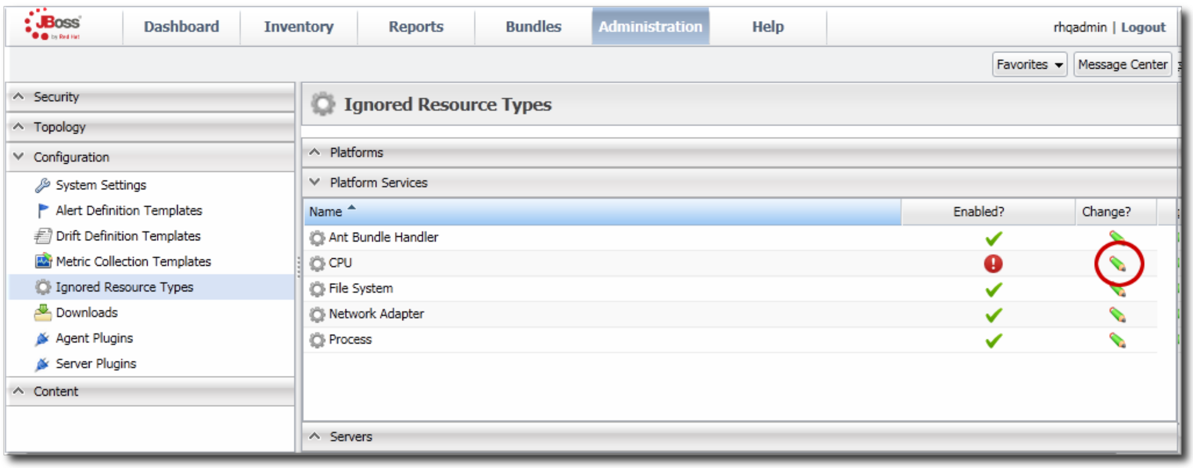
It is not possible to ignore a platform. If a platform should not be in the inventory, do not run an agent on that machine.

1. In the top menu, click the **`Administration`** tab.



2. In the **Configuration** menu table on the left, select the **Ignored Resource Types** item.

3. Every available resource type, based on the loaded agent plug-ins, is listed in the **`Ignored Resource Types`** page. To ignore a resource, click the pencil icon.



Name	Enabled?	Change?
Ant Bundle Handler	✓	✎
CPU	!	✎
File System	✓	✎
Network Adapter	✓	✎
Process	✓	✎

That toggles whatever the current enabled/disabled setting is for ignoring the resource. If a resource type is enabled, then it will be discovered by the agent. If it is disabled, it will be ignored.

4. Scroll to the bottom of the page and click the **Save** button.

5.3. RESOURCES THAT REQUIRE ADDITIONAL CONFIGURATION FOR DISCOVERY

There are some resource types which require specific configuration for the resource or for the JBoss ON agent in order for them to be discovered.

5.3.1. Configuring the Agent to Discover EAP 6 Instances

As covered in Chapter 4, *Interactions with System Users for Agents and Resources*, the system user as which the agent runs has a direct effect on how the agent can manage certain resource types. For EAP instances, the agent's system user must have the appropriate permissions to be able to manage EAP resources:

- The agent must have read permissions to the **run.jar** file, plus execute and search permissions for every directory in the path to the **run.jar** file.
- When a JBoss EAP 6 instance is installed from an RPM, the agent user must belong to the same system group which runs the EAP instance. This is **jboss**, by default.

5.3.2. Configuring Tomcat/EWS Servers for Discovery (Windows)

Tomcat servers are discovered automatically on Linux and Unix systems, but they require additional configuration before they can be discovered on Windows systems.

1. Run **regedit**.
2. Navigate to Java preferences key for the Tomcat server, **HKEY_LOCAL_MACHINE\SOFTWARE\Apache Software Foundation\Procrun2.0\Tomcat***Ver#***\Parameters\Java**.
3. Edit the **Options** attribute, and add these parameters:

```
-Dcom.sun.management.jmxremote.port=9876
-Dcom.sun.management.jmxremote.ssl=false
-Dcom.sun.management.jmxremote.authenticate=false
```

4. Restart the Tomcat service.

After a few minutes, the Tomcat instance should show up in the Discovery Queue.

5.4. IMPORTING NEW RESOURCES MANUALLY

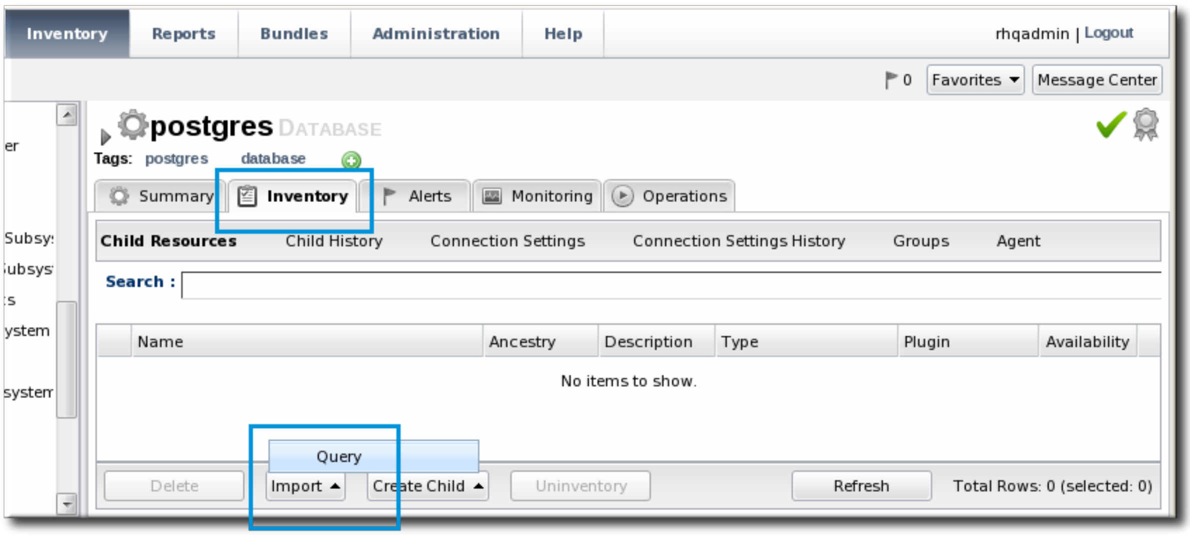
Discovery scans are run on a defined schedule. There may be an instance where you add a new server or service on a platform and want to add it immediately to the JBoss ON inventory, before the next scheduled discovery run. It is possible to add that new child resource manually by importing it into the inventory of the parent resource — without waiting for the next discovery scan.

NOTE

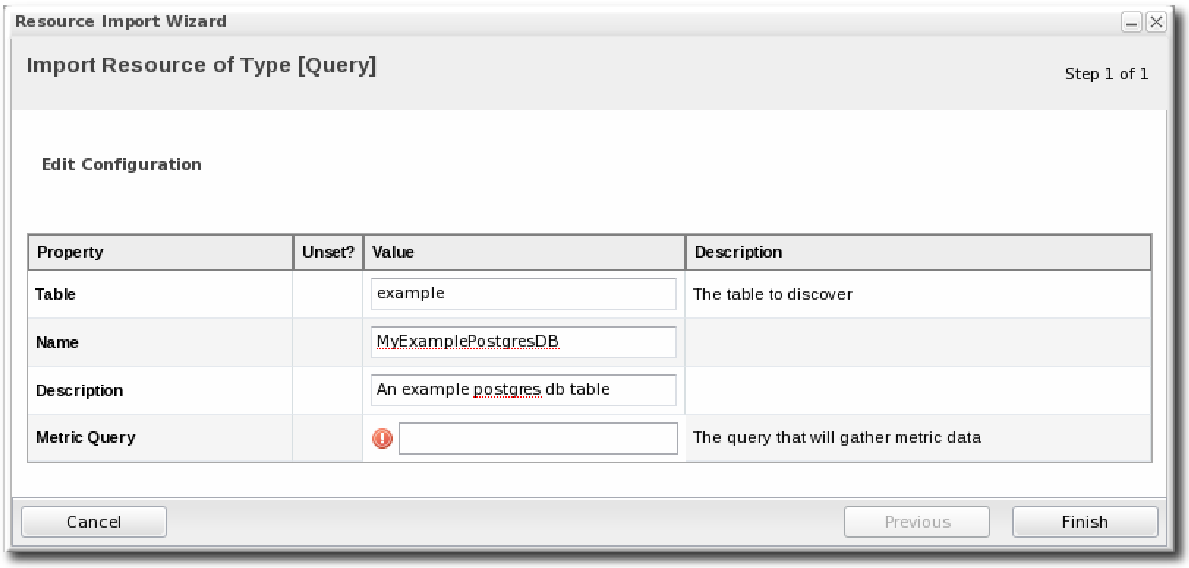
The parent resource must be in an available state in order to import a child resource.

1. Click the **Inventory** tab in the top menu.
2. Search for the parent resource of the new resource.

   Chapter 2, *Dynamic Searches for Resources and Groups*has information on searching for resources using dynamic searches.
3. Click the **Inventory** tab of the parent resource.
4. Click the **Import** button in the bottom of the **Inventory** tab, and select the type of child resource. The selection menu lists the possible types of child resources for that parent.



5. Fill in the properties to identify and connect to the new resource. Each resource type in the system has a different set of required properties.



Property	Unset?	Value	Description
Table		example	The table to discover
Name		MyExamplePostgresDB	
Description		An example postgres db table	
Metric Query			The query that will gather metric data

5.5. CREATING CHILD RESOURCES

JBoss ON can create certain types of children resources for parent resources. For example, a Postgres server can allow JBoss ON to create Postgres users. Not every allowed child resource type for a resource can be created through the JBoss ON UI; these children are usually limited to resource types that can be configured simply and remotely, such as scripts, WAR/EAR files, and server users.

NOTE

The parent resource must be available to add a child resource.

NOTE

It can take several minutes for the new child resource to be added and visible in the JBoss ON inventory because the new resource has to be created on the local system and then discovered by the agent. If the discovery scan is running when the resource is created, then it may take until the next discovery scan to be detected.

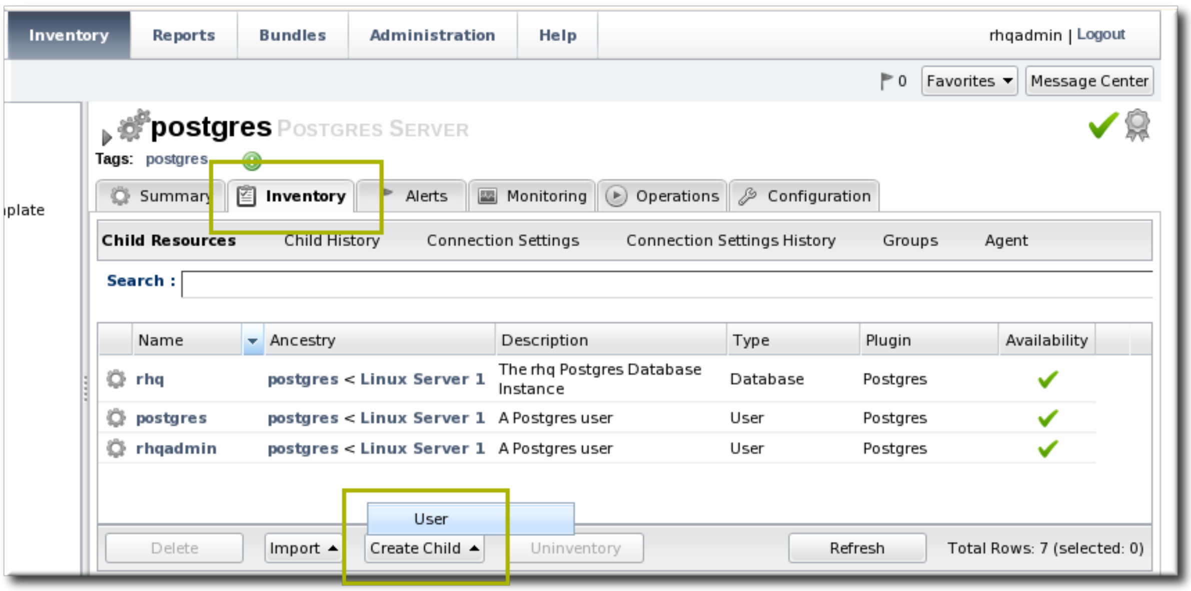
1. Click the **`Inventory`** tab in the top menu.

2. Search for the parent resource of the new resource.

   Chapter 2, *Dynamic Searches for Resources and Groups*has information on searching for resources using dynamic searches.

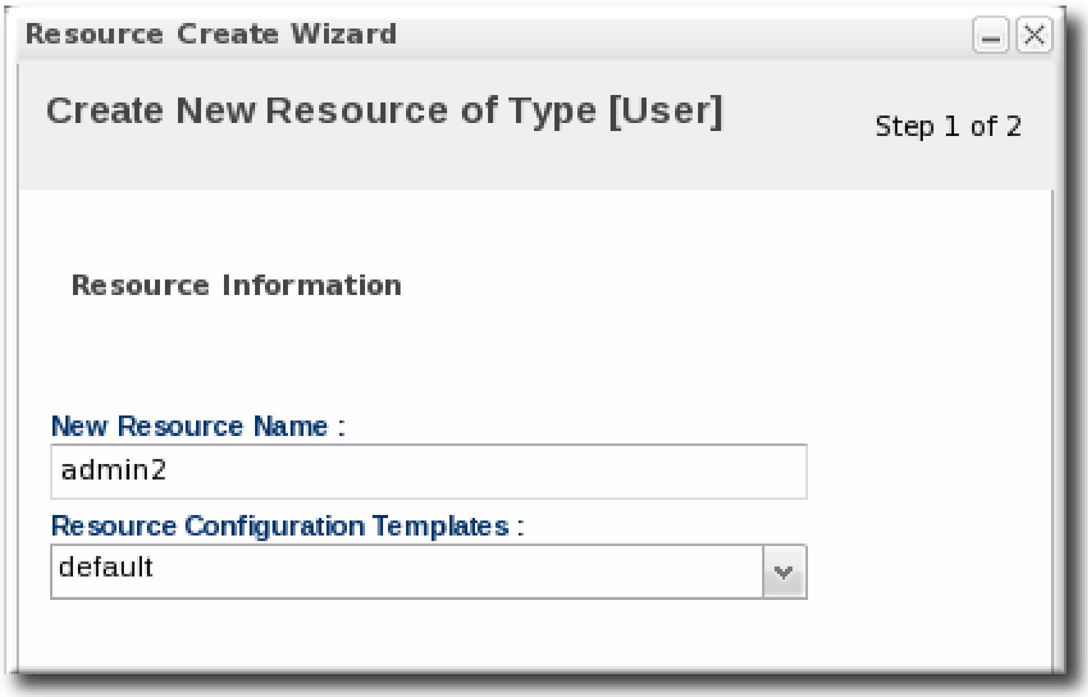
3. Click the **`Inventory`** tab of the parent resource.

4. Click the **`Create Child`** button in the bottom of the **`Inventory`** tab, and select the type of child resource. The selection menu lists the possible types of child resources for that parent.



Name	Ancestry	Description	Type	Plugin	Availability
rhq	postgres < Linux Server 1	The rhq Postgres Database Instance	Database	Postgres	✓
postgres	postgres < Linux Server 1	A Postgres user	User	Postgres	✓
rhqadmin	postgres < Linux Server 1	A Postgres user	User	Postgres	✓

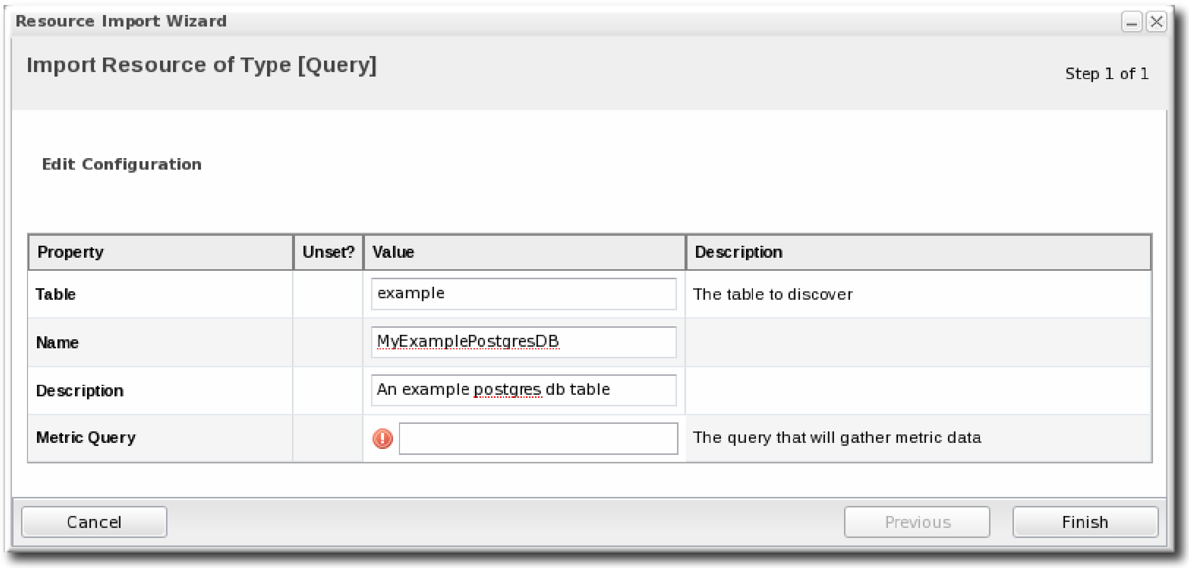
5. Give the name and description for the new resource.



New Resource Name : admin2

Resource Configuration Templates : default

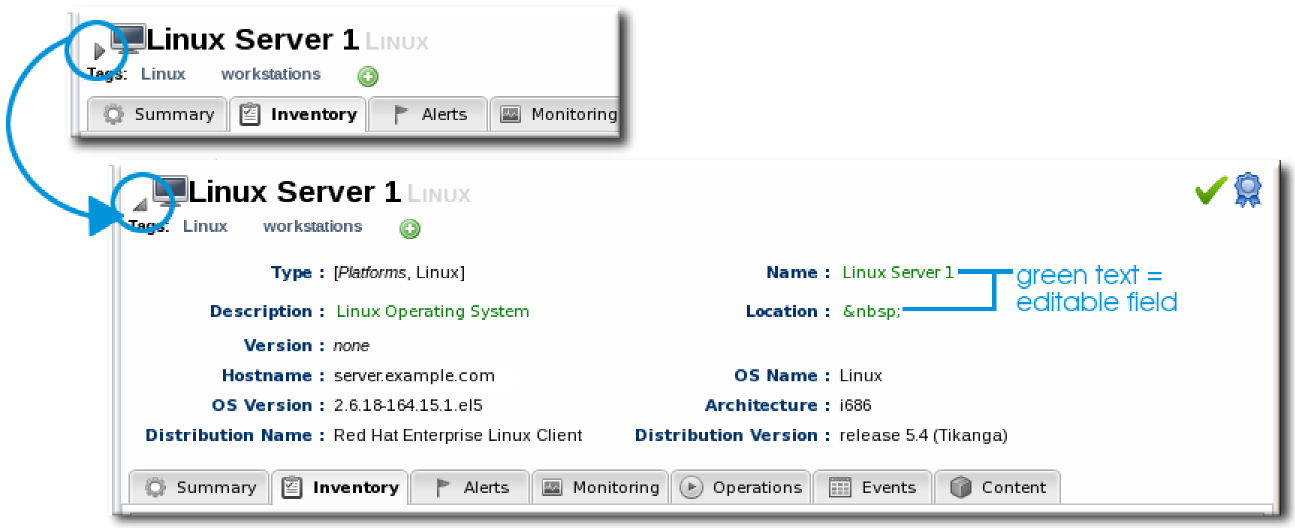
6. Fill in the properties to identify and connect to the new resource. Each resource type in the system has a different set of required properties.



Property	Unset?	Value	Description
Table		example	The table to discover
Name		MyExamplePostgresDB	
Description		An example postgres db table	
Metric Query			The query that will gather metric data

5.6. VIEWING AND EDITING RESOURCE INFORMATION

Every resource has details about the server or service that can be viewed, such as its name, description, and version. (The specific information is different for each resource type.) These details are usually hidden when viewing the resource, but they can be viewed by clicking the arrow by the resource name to expand the details area.

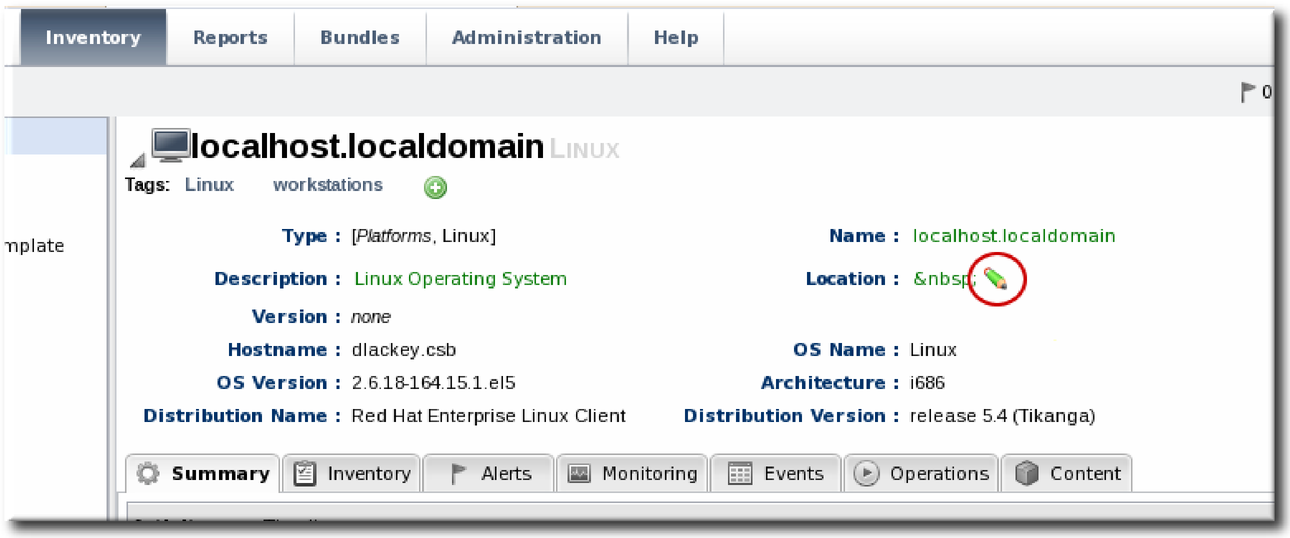


Type :	[Platforms, Linux]	Name :	Linux Server 1
Description :	Linux Operating System	Location :	
Version :	none		
Hostname :	server.example.com	OS Name :	Linux
OS Version :	2.6.18-164.15.1.el5	Architecture :	i686
Distribution Name :	Red Hat Enterprise Linux Client	Distribution Version :	release 5.4 (Tikanga)

Figure 5.2. Expanding Resource Entry Details

Any fields with green text can be edited. This allows administrators to use more specific or useful information in areas that are supplied by the agent discovery, like the resource name, or to add information, like a description or, for example, a platform's physical location.

To edit a field, hover the cursor over the name and click the pencil icon that appears.

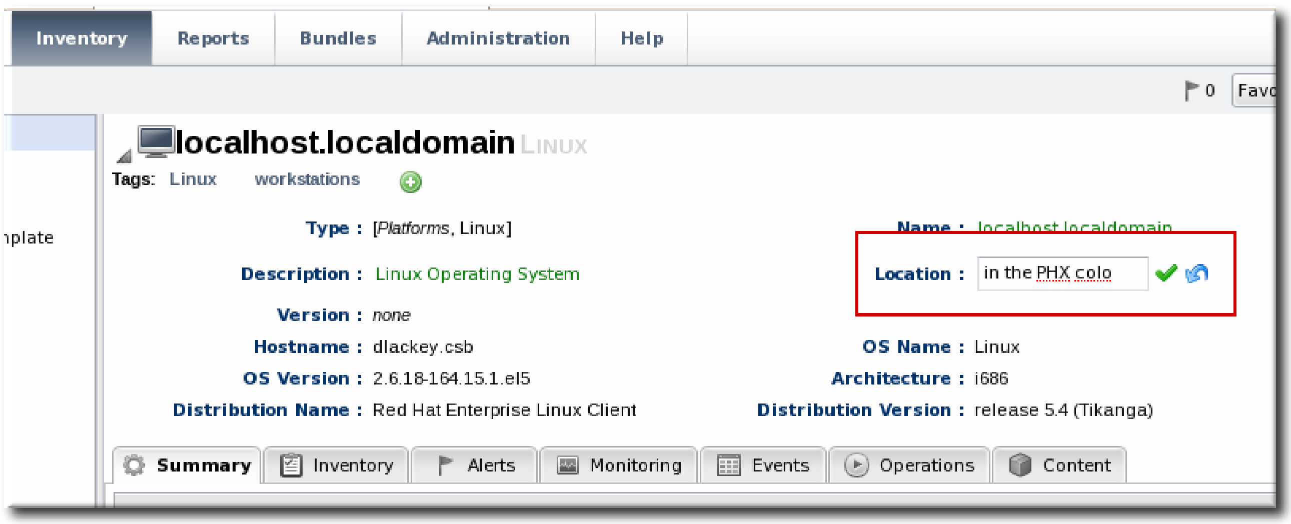


localhost.localdomain LINUX

Tags: Linux workstations

Type :	[Platforms, Linux]	Name :	localhost.localdomain
Description :	Linux Operating System	Location :	
Version :	none		
Hostname :	dlackey.csb	OS Name :	Linux
OS Version :	2.6.18-164.15.1.el5	Architecture :	i686
Distribution Name :	Red Hat Enterprise Linux Client	Distribution Version :	release 5.4 (Tikanga)

When the edits are made, click the green check mark to save the changes.



localhost.localdomain LINUX

Tags: Linux workstations

Type :	[Platforms, Linux]	Name :	localhost.localdomain
Description :	Linux Operating System	Location :	in the PHX colo
Version :	none		
Hostname :	dlackey.csb	OS Name :	Linux
OS Version :	2.6.18-164.15.1.el5	Architecture :	i686
Distribution Name :	Red Hat Enterprise Linux Client	Distribution Version :	release 5.4 (Tikanga)

5.7. MANAGING CONNECTION SETTINGS

Connection settings define how an agent can connect to a resource. These are settings defined in the agent plug-in, so they are different for each resource type.

At a minimum, connection settings provide a way for the agent to connect to a resource, such as a port number, directory path, or user credentials.

NOTE

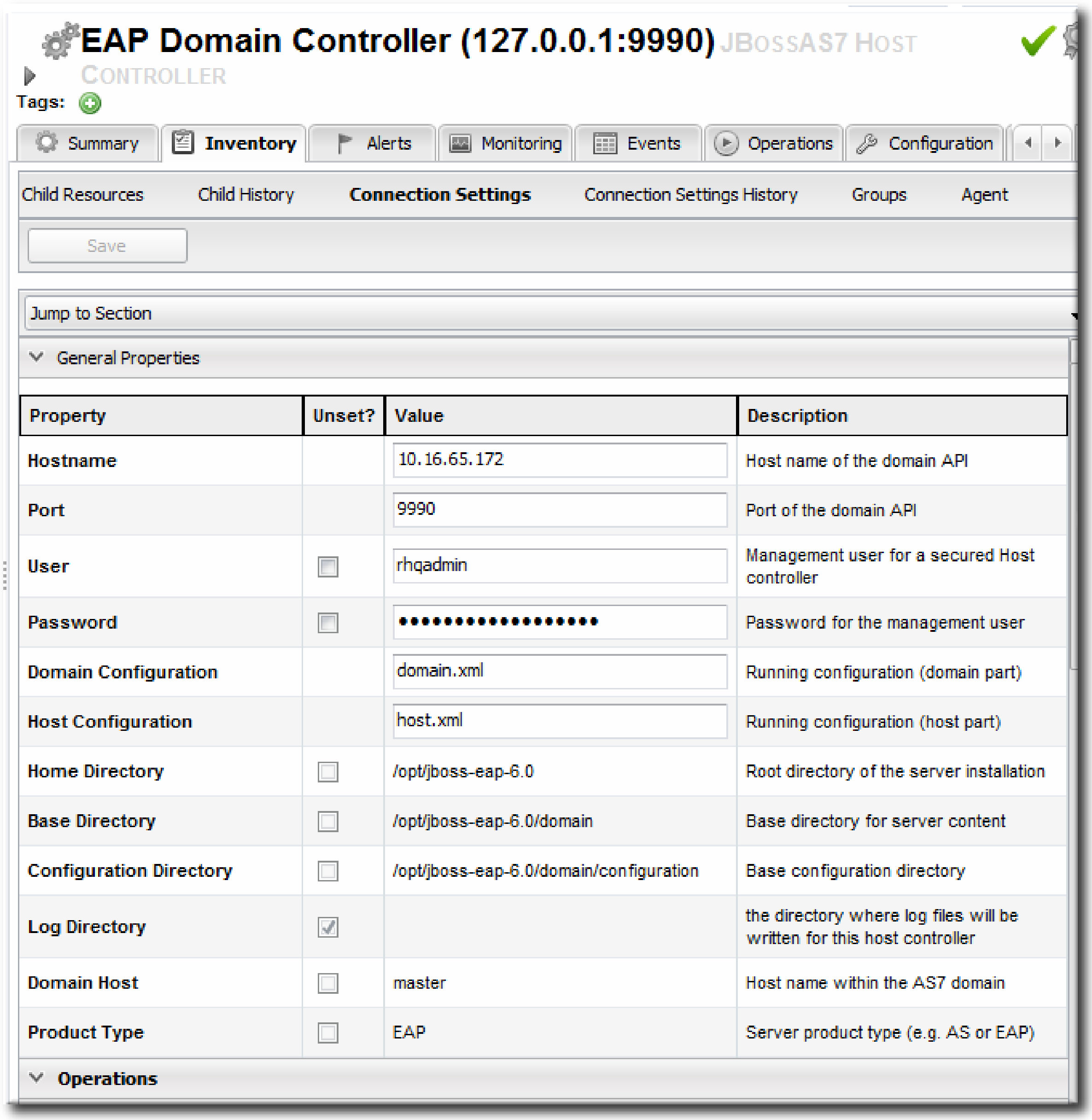
Often, if a resource shows down availability even when it is running, it is a problem with the connection settings. The agent may not have information it requires, such as a username or new port number, that it requires to connect to the resource. Since the agent cannot connect to the resource, it assumes it is down.

Connection settings can also provide configuration for other controls defined in the plug-in descriptor, like paths or options to use with scripts for operations, log file locations for event monitoring, and configuration files to allow for resource configuration editing.

To edit the connection settings:

1. Click the **Inventory** tab in the top menu.
2. Search for the resource.

   Chapter 2, *Dynamic Searches for Resources and Groups*has information on searching for resources using dynamic searches.
3. Click the name of the resource to go to its entry page.
4. Open the **Inventory** tab for the resource, and click the **Connection Settings** subtab.
5. Change the connection information for the resource.

Property	Unset?	Value	Description
Hostname		10.16.65.172	Host name of the domain API
Port		9990	Port of the domain API
User	☐	rhqadmin	Management user for a secured Host controller
Password	☐	••••••••••••••••••	Password for the management user
Domain Configuration		domain.xml	Running configuration (domain part)
Host Configuration		host.xml	Running configuration (host part)
Home Directory	☐	/opt/jboss-eap-6.0	Root directory of the server installation
Base Directory	☐	/opt/jboss-eap-6.0/domain	Base directory for server content
Configuration Directory	☐	/opt/jboss-eap-6.0/domain/configuration	Base configuration directory
Log Directory	✓		the directory where log files will be written for this host controller
Domain Host	☐	master	Host name within the AS7 domain
Product Type	☐	EAP	Server product type (e.g. AS or EAP)

   If a field is not editable immediately, select the **Unset** checkbox, and then enter new information in the field.
6. Click the **Save** button.

5.8. UNINVENTORYING AND DELETING RESOURCES

A resource can be removed from the JBoss ON inventory in one of two ways: it can be uninventoried (all resources) or it can be deleted (content-backed resources or configuration-related resources).

5.8.1. A Comparison of Uninventorying and Deleting Resources

Uninventorying a resource permanently and irrevocably removes all data about that resource from the JBoss ON inventory. It removes all historical monitoring data, configuration and operation histories, alerts, drift definitions, and any other stored data. However, the resource itself remains intact — it still exists on the machine and can be rediscovered (as a new resource) at a later time.

Any resource can be uninventoried.

Deleting a resource, on the other hand, completely removes the resource *from the machine itself*. So, deleting an EAR resource deletes the EAR from its parent EAP instance. However, the inventory information about that resource — its historic metric data, configuration history, and version history (for content resources) — remains intact, so that if a new version of that child is ever deployed, it is retains all of its original history.

Only resources not discovered through the discovery queue (such as deployed EAR files or created datasources) can be deleted.

5.8.2. Use Caution When Removing Resources

Uninventory Irrevocably Deletes the Resource History and Data

Uninventorying a resource removes all of the data that JBoss ON has for that resource: its metric data and historical monitoring data, alerts, drift and configuration history, operation history, and other data. **Once the resource is uninventoried, its data can never be recovered.**

Uninventorying or Deleting a Resource Removes All of Its Children

If a parent resource is removed from JBoss ON, then all of its children are also removed. Removing an EAP server, for example, removes all of its deployed web applications from the JBoss ON inventory. Removing a platform removes all servers, services, and resources on that platform.

Uninventoried Resources Can Still Be Discovered

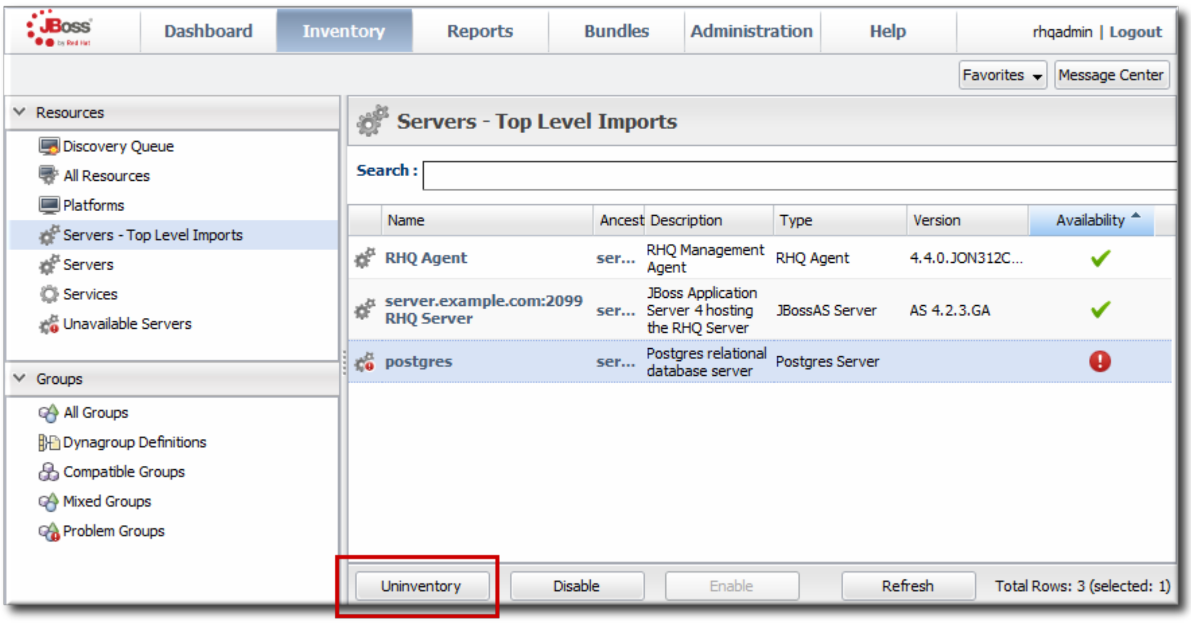
Even though a resource is uninventoried and all of its data in JBoss ON is permanently removed, the underlying resource still exists. This means that the resource can still be discovered. To prevent the resource from being discovered and re-added to the inventory, ignore the resource, as in Section 5.2.4, “Ignoring Discovered Resources”.

Anything Depending on a Deleted Resource Could Fail

Some resource types can be deleted, meaning the resource itself is removed from the machine, not just from the JBoss ON inventory. Anything that relies on that resource can experience failures because the resource is deleted. For example, if a datasource for an EAP server is deleted, that datasource is removed from the EAP server itself. Any application which attempts to connect to that datasource will then stop working, since it does not exist anymore.

5.8.3. Uninventorying through the Inventory Tab

1. Click the **`Inventory`** tab in the top menu.
2. Select the resource category in the **`Resources`** table on the left, and, if necessary, filter for the resource.
3. Select the resource to uninventory from the list, and click the **`Uninventory`** button.



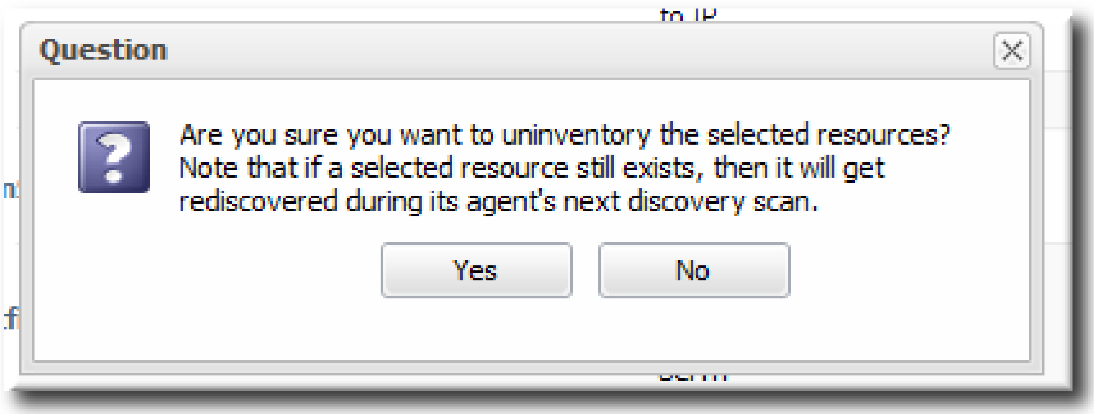
Servers - Top Level Imports

Search :

Name	Ancest	Description	Type	Version	Availability
RHQ Agent	ser...	RHQ Management Agent	RHQ Agent	4.4.0.JON312C...	✓
server.example.com:2099 RHQ Server	ser...	JBoss Application Server 4 hosting the RHQ Server	JBossAS Server	AS 4.2.3.GA	✓
postgres	ser...	Postgres relational database server	Postgres Server		●

Total Rows: 3 (selected: 1)

4. When prompted, confirm that the resource should be uninventoried.



Question

Are you sure you want to uninventory the selected resources? Note that if a selected resource still exists, then it will get rediscovered during its agent's next discovery scan.

5. To prevent the resource from being re-imported into the inventory, ignore it when it is discovered in the next discovery scan. This is covered in Section 5.2.4, “Ignoring Discovered Resources”.

5.8.4. Uninventorying through the Parent Inventory

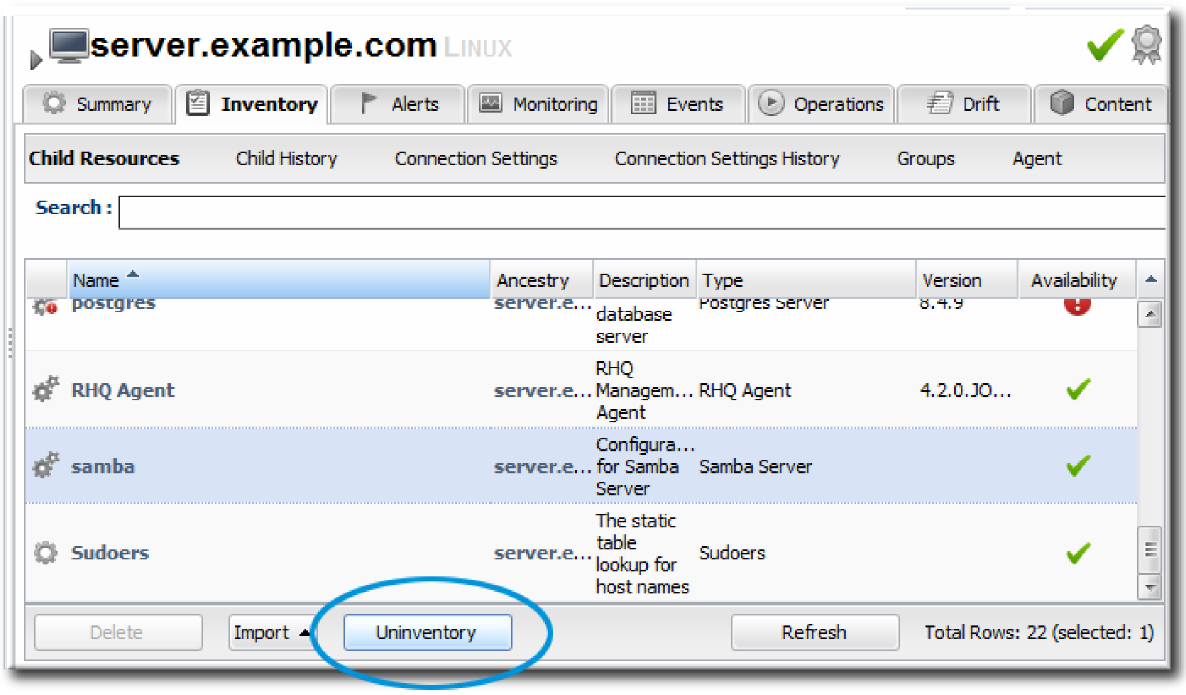
1. Click the **Inventory** tab in the top menu.

2. Search for the parent resource of the resource.

   Chapter 2, *Dynamic Searches for Resources and Groups*has information on searching for resources using dynamic searches.

3. Click the **Inventory** tab for the parent resource.

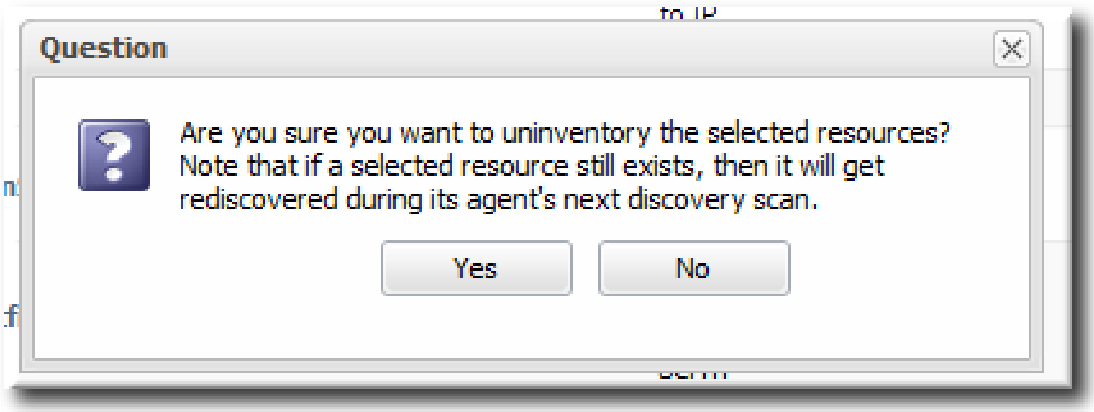
4. Click on the line of the child resource to uninventory. To select multiple entries, use the **Ctrl** key.



Name	Ancestry	Description	Type	Version	Availability
postgres	server.e...	database server	Postgres Server	8.4.9	●
RHQ Agent	server.e...	RHQ Managem... Agent	RHQ Agent	4.2.0.JO...	✓
samba	server.e...	Configura... for Samba Server	Samba Server		✓
Sudoers	server.e...	The static table lookup for host names	Sudoers		✓

5. Click the **Uninventory** button.

6. When prompted, confirm that the resource should be uninventoried.



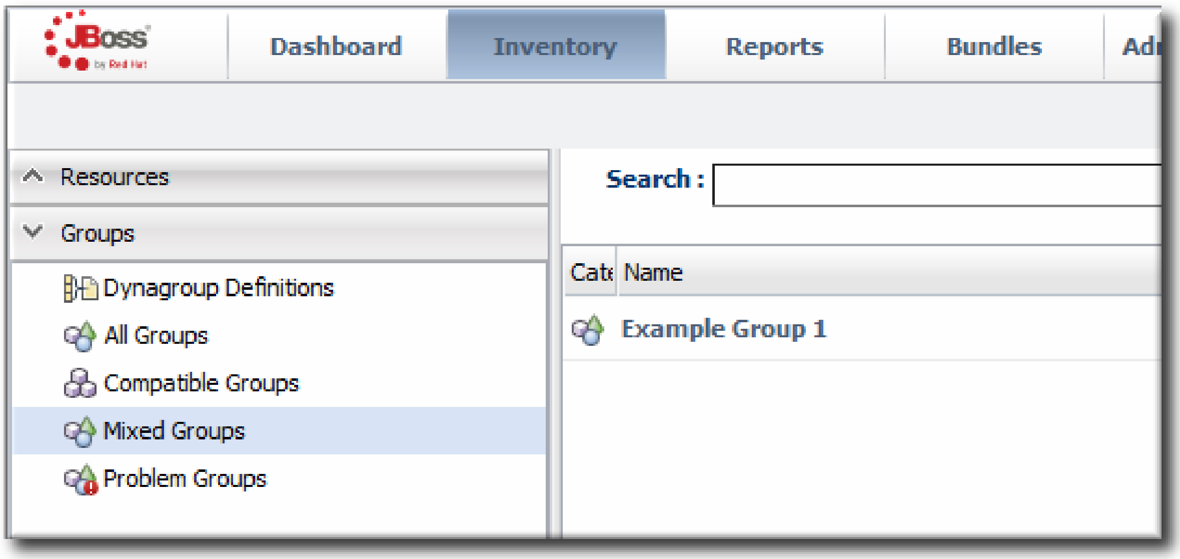
Are you sure you want to uninventory the selected resources? Note that if a selected resource still exists, then it will get rediscovered during its agent's next discovery scan.

7. To prevent the resource from being re-imported into the inventory, ignore it when it is discovered in the next discovery scan. This is covered in Section 5.2.4, "Ignoring Discovered Resources".

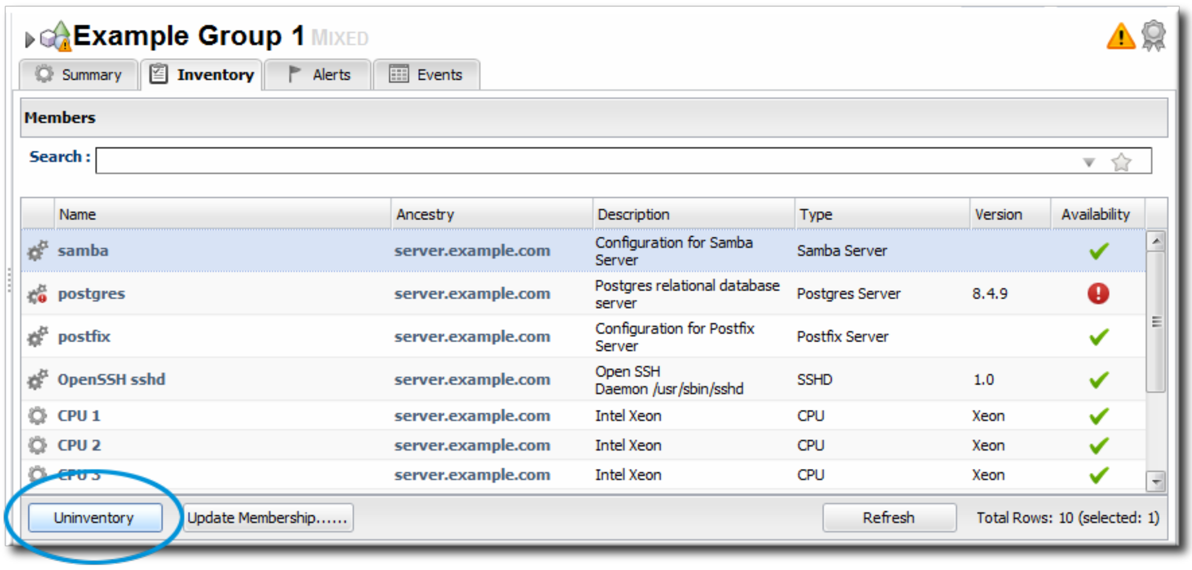
5.8.5. Uninventorying through a Group Inventory

If a resource is a member of a group, then the resource can be uninventoried through the group management pages, as part of managing the group resources.

1. In the **Inventory** tab in the top menu, select the compatible or mixed groups item in the **Groups** menu on the left.

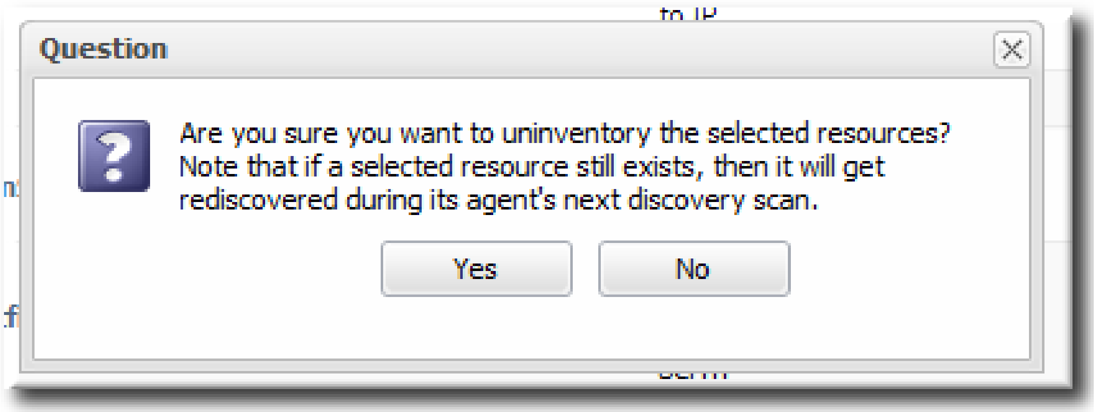


2. Click the name of the group.
3. Open the **Inventory** tab for the group, and open the **Members** submenu.
4. Click on the line of the group member to uninventory. To select multiple entries, use the **Ctrl** key.



Name	Ancestry	Description	Type	Version	Availability
samba	server.example.com	Configuration for Samba Server	Samba Server		✓
postgres	server.example.com	Postgres relational database server	Postgres Server	8.4.9	●
postfix	server.example.com	Configuration for Postfix Server	Postfix Server		✓
OpenSSH sshd	server.example.com	Open SSH Daemon /usr/sbin/sshd	SSHD	1.0	✓
CPU 1	server.example.com	Intel Xeon	CPU	Xeon	✓
CPU 2	server.example.com	Intel Xeon	CPU	Xeon	✓
CPU 3	server.example.com	Intel Xeon	CPU	Xeon	✓

5. Click the **Uninventory** button.
6. When prompted, confirm that the resource should be uninventoried.


Question

Are you sure you want to uninventory the selected resources? Note that if a selected resource still exists, then it will get rediscovered during its agent's next discovery scan.

7. To prevent the resource from being re-imported into the inventory, ignore it when it is discovered in the next discovery scan. This is covered in Section 5.2.4, "Ignoring Discovered Resources".

5.8.6. Deleting a Resource

Deleting a resource does several things:

- Deletes the resource from the underlying machine.
- Removes the resource from the inventory.
- Removes any child resources from JBoss ON.
- **Preserves the inventory information in JBoss ON for the resource, including alerts, drift definitions, metric data, and configuration and operation histories.**

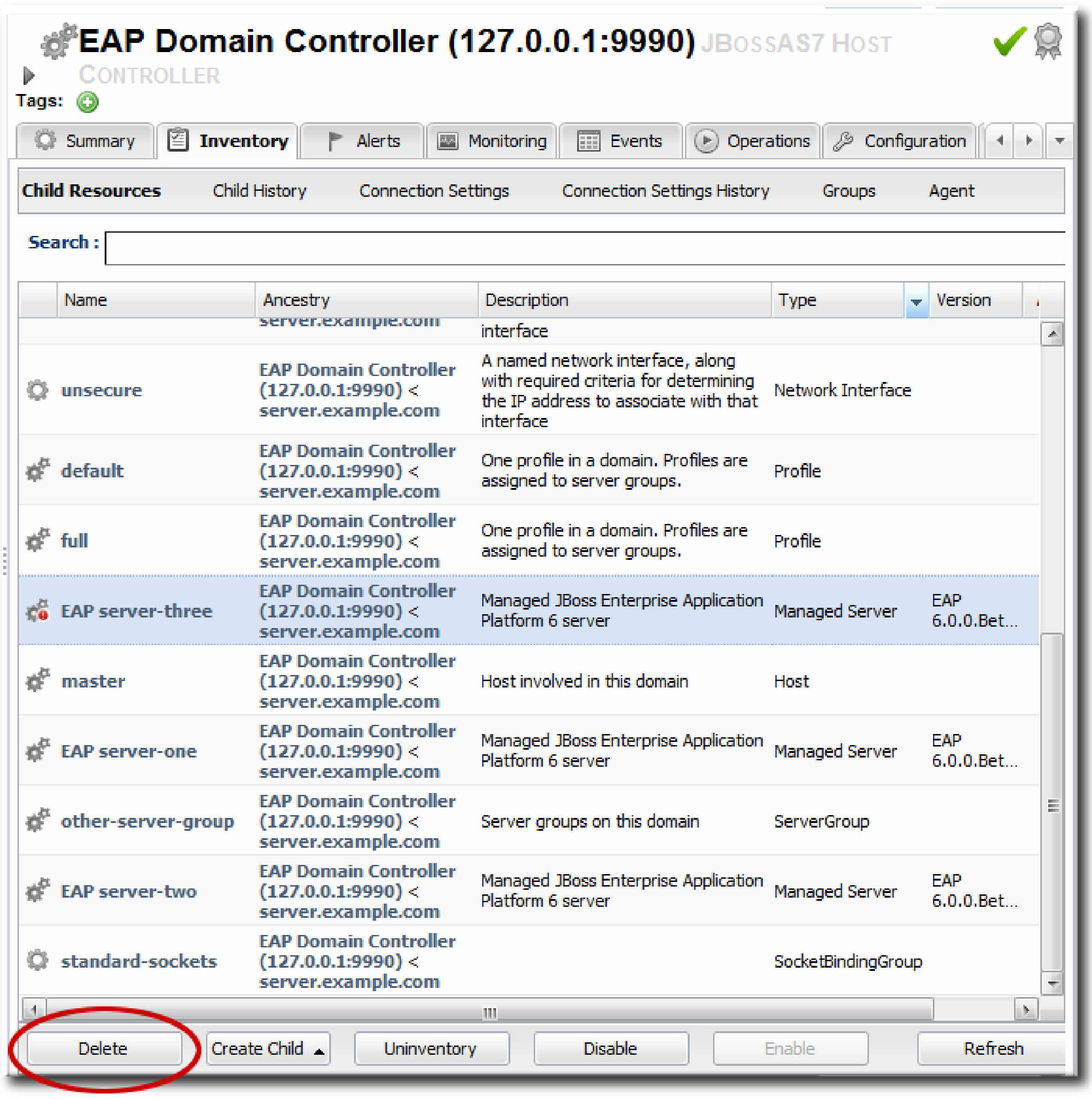
Only a resource **not** imported through the discovery queue can be deleted. This generally means that content-backed resources (EARs, WARs, and JARs) and other child resources like datasources can be deleted.

WARNING

Because the real, underlying resource is deleted (not just the inventory entry), anything relying on that resource can experience failures.

1. Click the `Inventory` tab in the top menu.
2. Search for the *parent* resource of the resource to delete.

   Chapter 2, *Dynamic Searches for Resources and Groups*has information on searching for resources using dynamic searches.
3. Click the `Inventory` tab of the parent resource.
4. Select the resource to delete from the list of children.
5. Click the `Delete` button in the bottom of the `Inventory` tab.



Name	Ancestry	Description	Type	Version
	server.example.com	interface		
unsecure	EAP Domain Controller (127.0.0.1:9990) < server.example.com	A named network interface, along with required criteria for determining the IP address to associate with that interface	Network Interface	
default	EAP Domain Controller (127.0.0.1:9990) < server.example.com	One profile in a domain. Profiles are assigned to server groups.	Profile	
full	EAP Domain Controller (127.0.0.1:9990) < server.example.com	One profile in a domain. Profiles are assigned to server groups.	Profile	
EAP server-three	EAP Domain Controller (127.0.0.1:9990) < server.example.com	Managed JBoss Enterprise Application Platform 6 server	Managed Server	EAP 6.0.0.Bet...
master	EAP Domain Controller (127.0.0.1:9990) < server.example.com	Host involved in this domain	Host	
EAP server-one	EAP Domain Controller (127.0.0.1:9990) < server.example.com	Managed JBoss Enterprise Application Platform 6 server	Managed Server	EAP 6.0.0.Bet...
other-server-group	EAP Domain Controller (127.0.0.1:9990) < server.example.com	Server groups on this domain	ServerGroup	
EAP server-two	EAP Domain Controller (127.0.0.1:9990) < server.example.com	Managed JBoss Enterprise Application Platform 6 server	Managed Server	EAP 6.0.0.Bet...
standard-sockets	EAP Domain Controller (127.0.0.1:9990) < server.example.com		SocketBindingGroup	

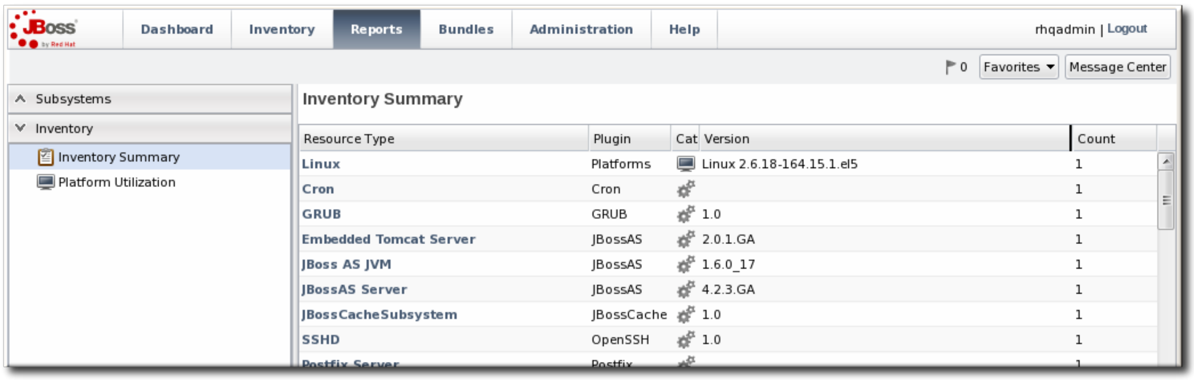
5.9. VIEWING INVENTORY SUMMARY REPORTS

One quick management tool in JBoss ON is an inventory report. The report summarizes the resources currently in the inventory, grouped by resource type and five summaries:

- Resource type
- The JBoss ON server plug-in which manages the resource
- The JBoss ON category for the resource (platform, server, or service)
- The version number or numbers for resource of the resource type in inventory
- The total number of resources of that type in the inventory

To generate the inventory report:

1. In the top menu, click the `Reports` tab.
2. In the `Inventory` menu box in the menu table on the left, select the **Inventory Summary** report.



Inventory Summary

Resource Type	Plugin	Cat	Version	Count
Linux	Platforms		Linux 2.6.18-164.15.1.el5	1
Cron	Cron			1
GRUB	GRUB		1.0	1
Embedded Tomcat Server	JBossAS		2.0.1.GA	1
JBoss AS JVM	JBossAS		1.6.0_17	1
JBossAS Server	JBossAS		4.2.3.GA	1
JBossCacheSubsystem	JBossCache		1.0	1
SSHD	OpenSSH		1.0	1
Postfix Server	Postfix			1

3. Click the name of any resource type to go to the inventory list for that resource type.

NOTE

Reports can be exported to CSV, which can be used for office systems or further data manipulation.

To export a report, simply click the **`Export`** button. The report will automatically be downloaded as **`inventorySummary.csv`**.

[1] Provisioning Ant bundles is implemented through an agent plug-in which performs the tasks on the platform and a server-side plug-in which manages the bundles in the server.

CHAPTER 6. MANAGING GROUPS

Groups are a simple, yet effective, way to organize resources. Particularly where there are large numbers of resources or where there are logical divisions between resources across departments, IT environments, or physical locations.

Groups in JBoss Operations Network provide a way to manage resources easily and more consistently. Alerts, operations, and configuration can be applied to individual resources or to entire groups of resources, while groups can be monitored from a single view.

6.1. ABOUT GROUPS

Groups are simply a means to organize resources within the JBoss ON inventory. JBoss ON has several different kinds of groups, listed in Table 6.1, “Types of Groups” , which allows an administrator to manage resources in different, flexible ways.

Table 6.1. Types of Groups

Type	Description	Static or Dynamic
Mixed groups	Contains resources of any resource type. There is no limit to how many or what types of resources can be placed into a mixed group. Mixed groups are useful for granting access permissions to users for a set of grouped resources.	Static
Compatible groups	Contains only resources of the same type. Compatible groups make it possible to perform an operation against every member of the group at the same time, removing the need to individually upgrade multiple resources of the same type, or perform other operations one at a time on resources across the entire enterprise.	Static
Recursive groups	Includes all the descendant, or child, resources of resources within the group. Recursive groups show both the explicit member availability and the child resource availability.	Static (members) and dynamic (children)

Type	Description	Static or Dynamic
Autogroups	Shows every resource as part of a resource hierarchy with the platform at the top, and child and descendant resources below the platform. Child resources of the same type are automatically grouped into an autogroup.	Dynamic

6.1.1. Dynamic and Static Groups

Groups are a way of organizing resources. The different types of groups are covered in Table 6.1, “Types of Groups”, but all of these groups fall into one of two categories. Groups are either *static* or *dynamic*, depending on how resources are assigned to the group. Static groups have resources which are explicitly assigned to the group, so the membership does not change even if the inventory changes. Dynamic groups are based on some kind of search criteria, and the group members are all of the resources returned in that search. Whenever the inventory is updated, the search results change, and the group membership is automatically updated.

Both static and dynamic groups can be valuable for managing resources and keeping a perspective on the overall IT environment.

6.1.2. About Autogroups

There are two basic types of groups in JBoss ON: static groups, where resources are added manually, and dynamic groups, where resources are added automatically based on some kind of established criteria.

Administrators can configure dynamic groups based on defined searches, which is covered in Chapter 7, *Using Dynamic Groups*. JBoss ON supports a different kind of dynamic group called an *autogroup*. Autogroups are used to construct the inventory navigation trees in the JBoss ON UI, and they are based on the underlying resource hierarchy, or parent-child relationships. Autogroups also group along resource type. For example, in Figure 6.1, “PostgreSQL Autogroup”, there are autogroups under the Postgres resource for all its children, which are further divided based on the child resource type, databases and users.

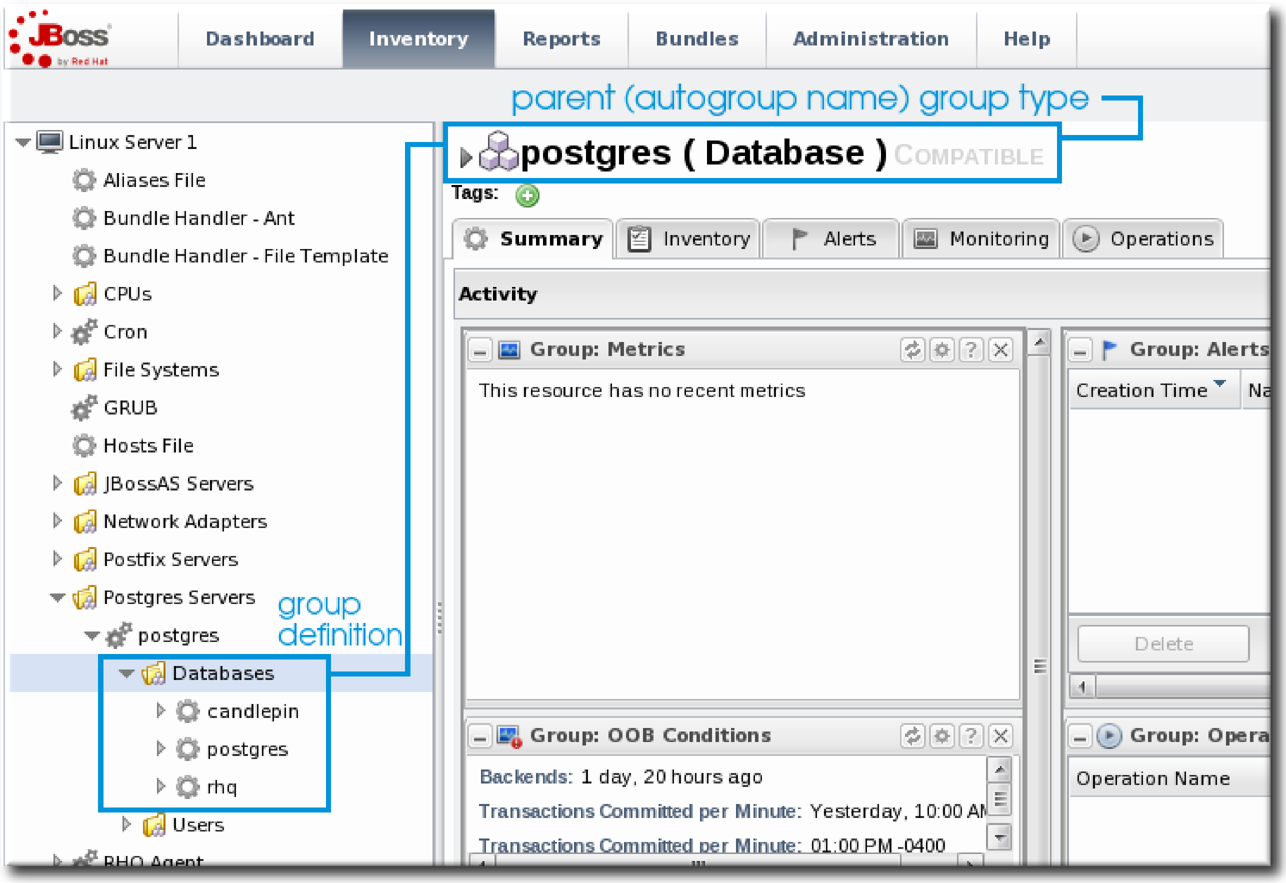


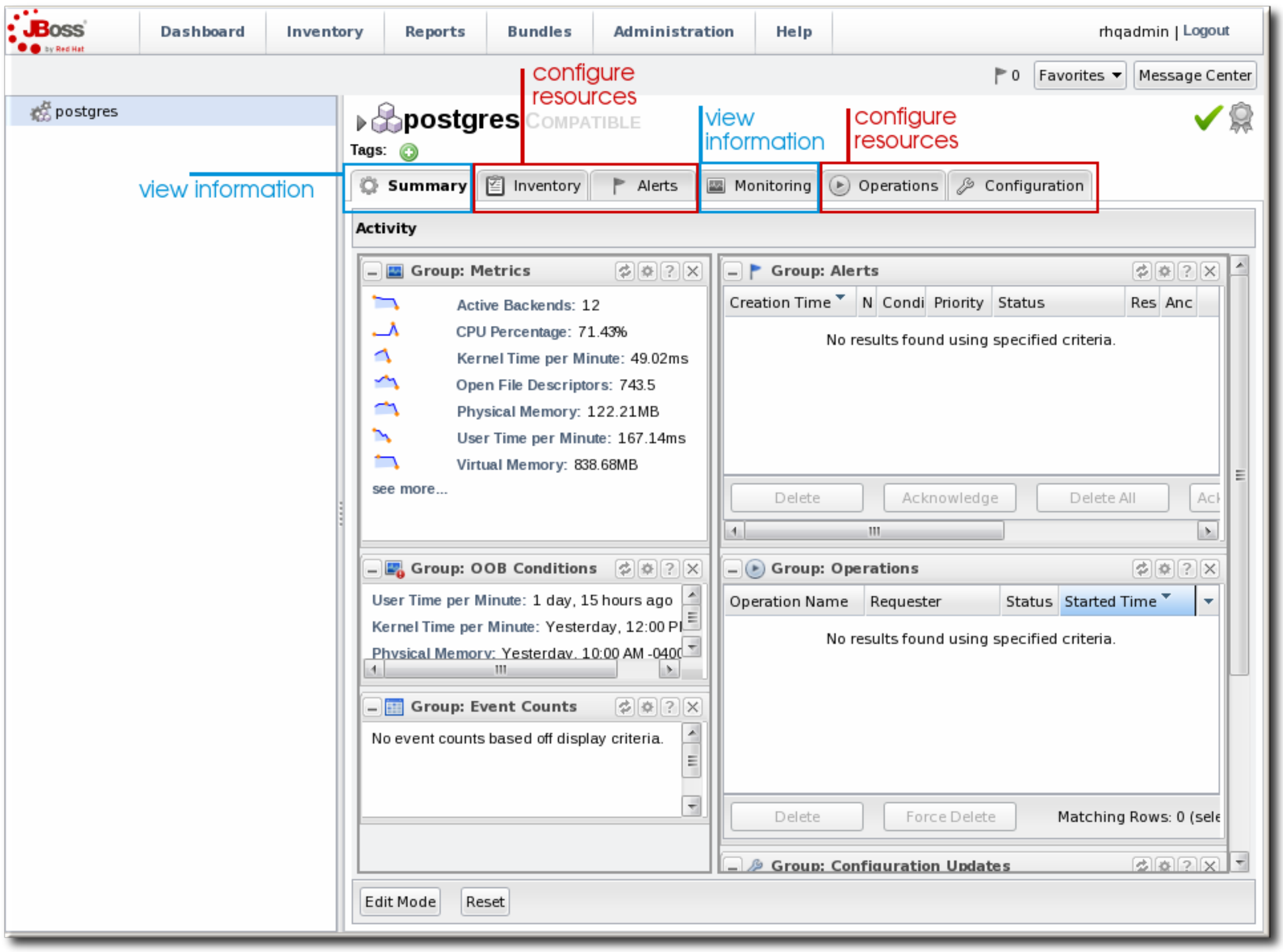
Figure 6.1. PostgreSQL Autogroup

Autogroups, unlike other groups in JBoss ON, are not configurable by JBoss ON users. Autogroups are defined internally in the JBoss ON server and are used by JBoss ON.

6.1.3. Comparing Compatible and Mixed Groups

Using groups allows multiple resources to be managed simultaneously. The type of group — compatible or mixed — specifies what kind of management can be performed on the group members.

Compatible groups, because they have members all of the same type, can be managed almost as easily as a single resource. Administrators can change resource configuration, launch operations, set alerts, and view individual and group-averaged monitoring data. Any changes can be made to a single group member, selected members, or the entire group. The list of group members, the group *inventory*, is managed through the **`Inventory`** tab.



Active Backends: 12
CPU Percentage: 71.43%
Kernel Time per Minute: 49.02ms
Open File Descriptors: 743.5
Physical Memory: 122.21MB
User Time per Minute: 167.14ms
Virtual Memory: 838.68MB
see more...

Creation Time	N	Condi	Priority	Status	Res	Anc

No results found using specified criteria.

User Time per Minute: 1 day, 15 hours ago
Kernel Time per Minute: Yesterday, 12:00 PI
Physical Memory: Yesterday, 10:00 AM -0400

Operation Name	Requester	Status	Started Time

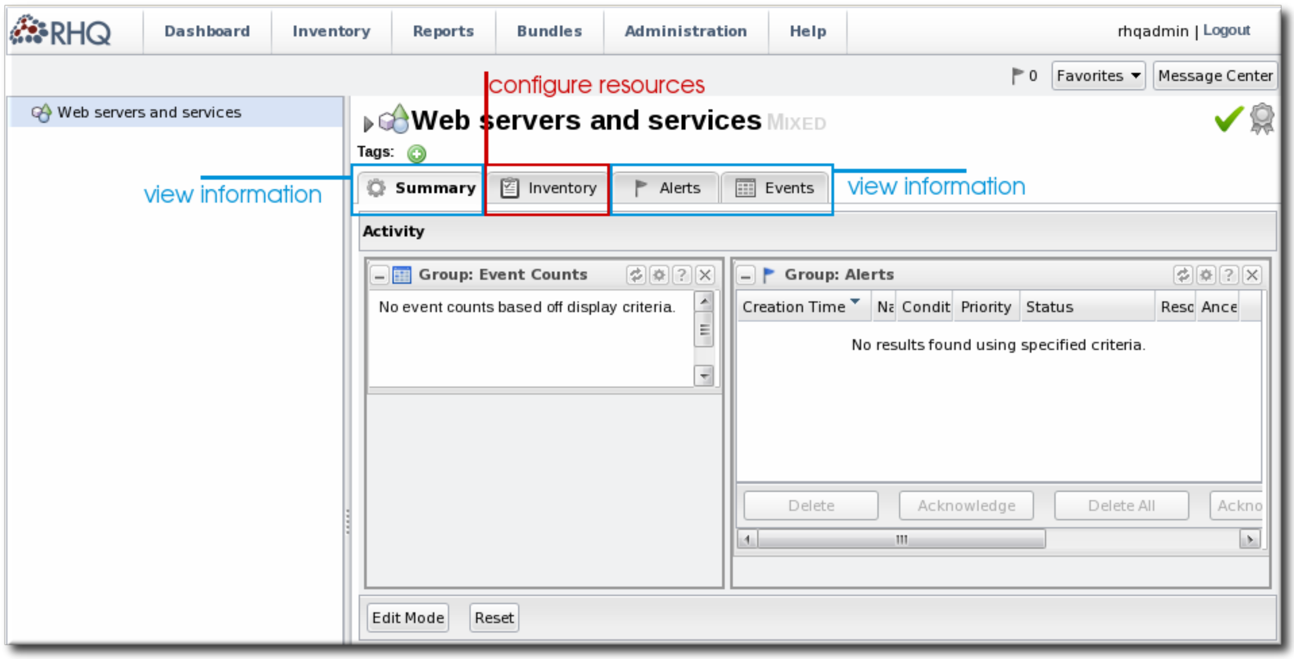
No results found using specified criteria.

Matching Rows: 0 (sele

No event counts based off display criteria.

Figure 6.2. Compatible Group Entry

Mixed groups can have members of different resource types, so group management is limited to updating the members (the group inventory) and viewing the history of alerts and events for the group members.



No event counts based off display criteria.

Creation Time	Na	Condit	Priority	Status	Resc	Ance

No results found using specified criteria.

Figure 6.3. Mixed Group Entry

6.1.4. Leveraging Recursive Groups

Compatible and mixed groups have a set, explicit membership. This static structure makes them useful for creating policies within JBoss ON because they are reliable.

Compatible and mixed groups can have a setting on them making them *recursive*. A recursive group travels down the inventory of every member and implicitly adds all of their children to the group, too. In a sense, a recursive group has two tiers of members: the explicit members to the group and then the implicit set of child members.

This idea of two levels of members allows a group to retain its own type definition — meaning specifically that it does not change a compatible group to a mixed group just because its children are added as members. The explicit membership is called the *root node*. Group operations, from adding members to setting metrics schedules to changing connection settings, are only performed on that root node.

Recursive groups can be useful in a lot of different ways.

For example, recursive compatible groups can be used to look at a subset of autoclusters in the inventory because all of the child resources are grouped by type, as are their parent resources. This provides an easy view at a subset of resources. (Mixed groups, which do not group by type, show only the root node or explicit members in the hierarchy.)

One of the more important uses is using recursive groups, particularly mixed recursive groups, for authorization control. This is covered more in Chapter 9, *Managing Roles and Access Control*, but users have to be granted access to resources explicitly, by adding the user and the resource to the same role. Using a recursive group automatically includes all of those resources' children in the role, which makes the role easier to maintain and more accurate in granting access.

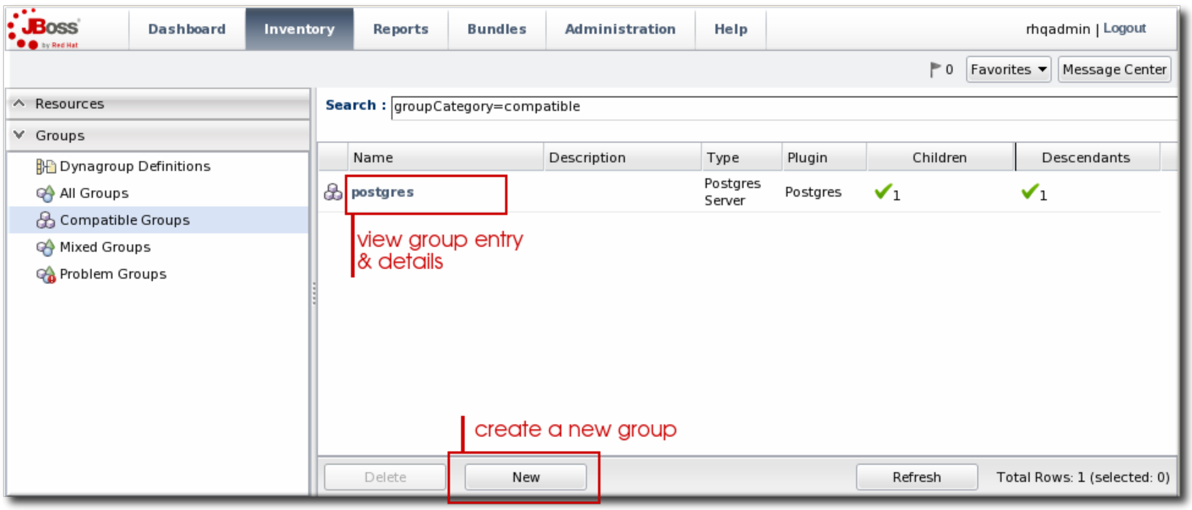
6.2. CREATING GROUPS

A user must have the global security or inventory permission to create groups.

1. Click the **`Inventory`** tab in the top menu.

2. In the **`Groups`** box in the left menu, select the type of group to create, either compatible or mixed.

   Compatible groups have resources all of the same type, while mixed groups have members of different types. The differences in the types of members means that there are different ways that compatible and mixed groups can be managed, as covered in Section 6.1.3, “Comparing Compatible and Mixed Groups”.

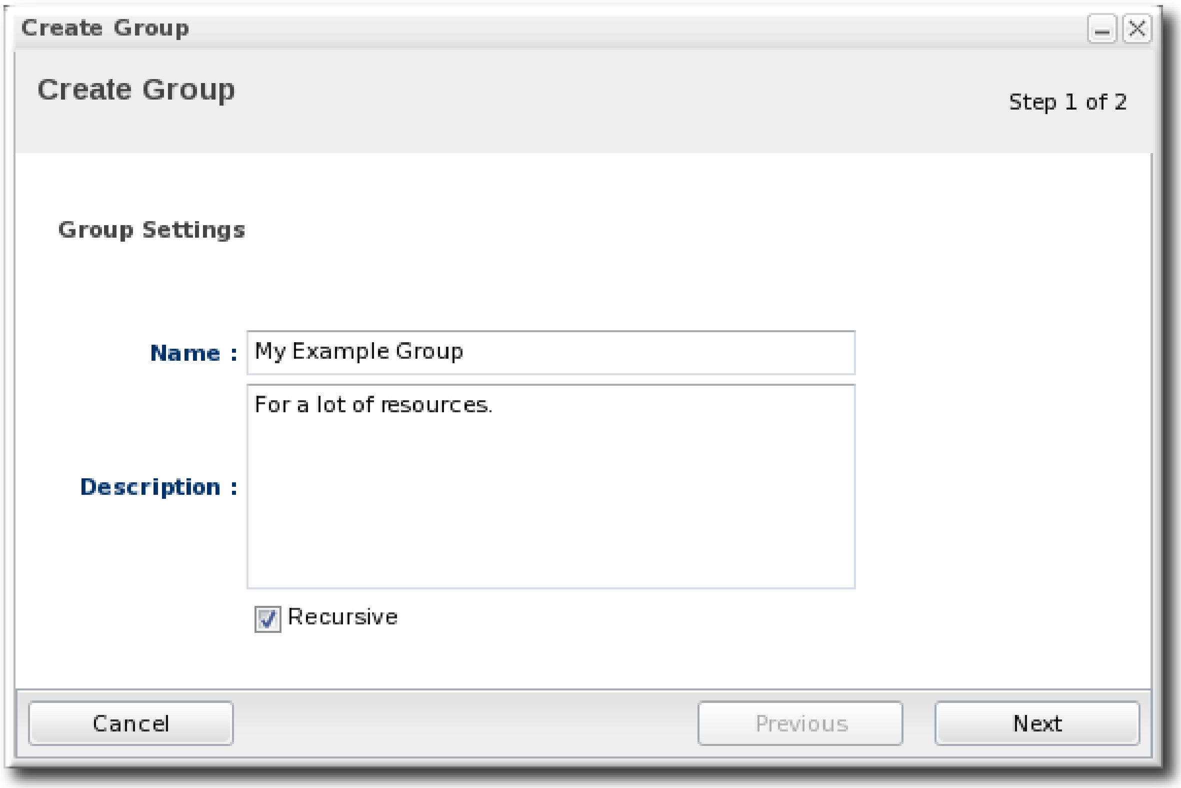


Search : groupCategory=compatible

Name	Description	Type	Plugin	Children	Descendants
postgres		Postgres Server	Postgres	✓1	✓1

Total Rows: 1 (selected: 0)

3. Enter a name and description for the group.



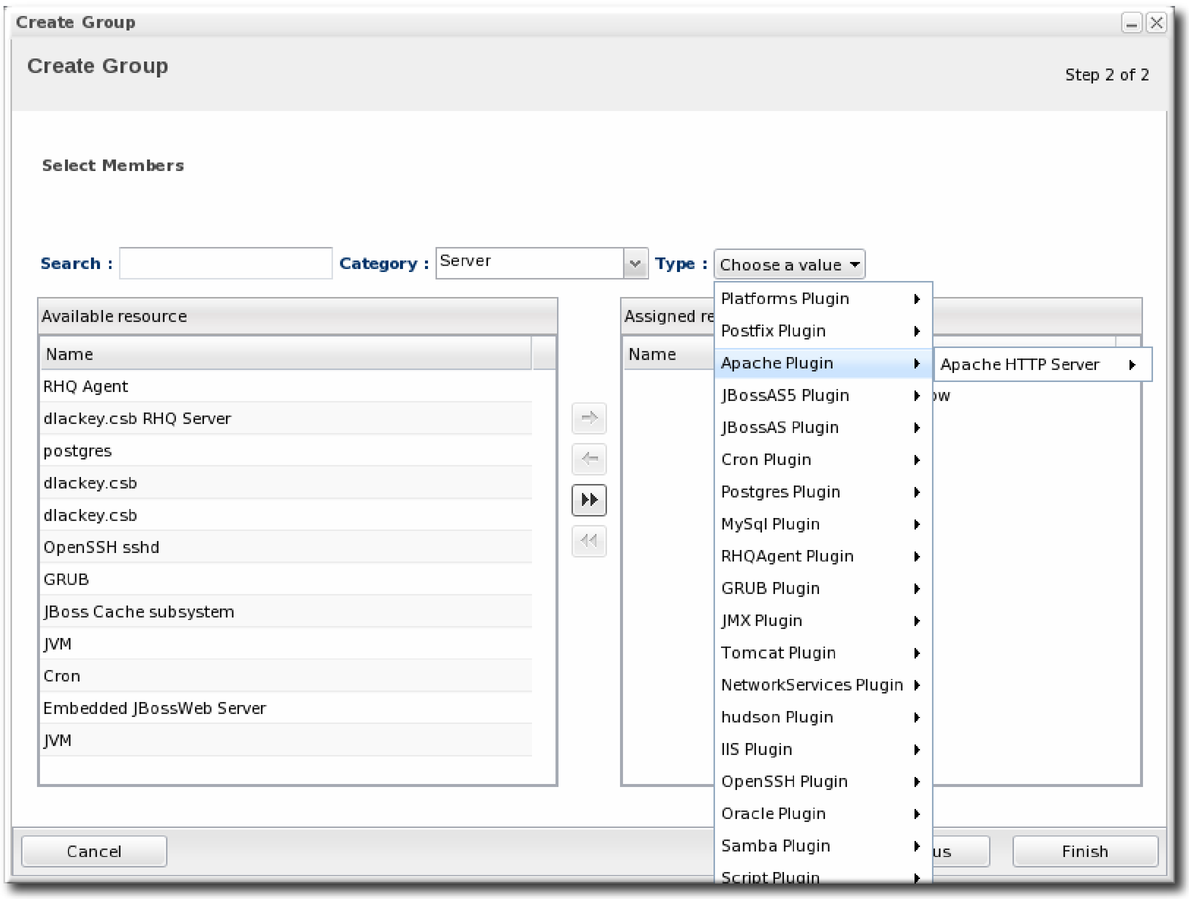
Name : My Example Group

Description : For a lot of resources.

[✓] Recursive

Marking groups recursive can make it easier to manage resources, particularly when setting role access controls. For example, administrators can grant users access to the group and automatically include any child resources of the member resources.

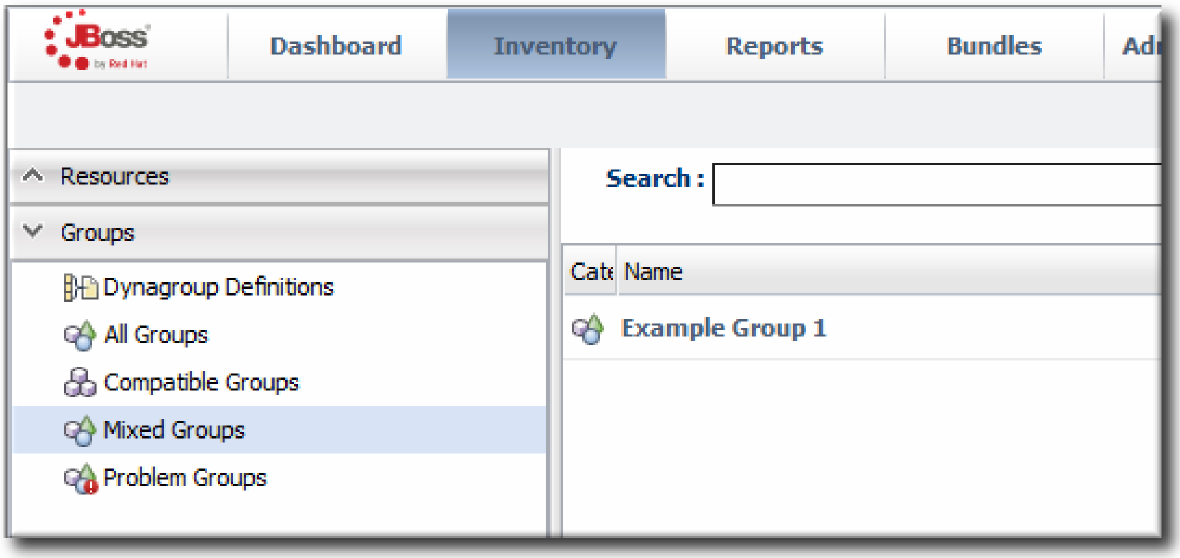
4. Select the group members. It is possible to filter the choices based on name, type, and category.



6.3. CHANGING GROUP MEMBERSHIP

Compatible and mixed groups both have static members, which means that resources are manually assigned to the group rather than being assigned dynamically based on some attribute. The group membership can be changed as the resources in the JBoss ON inventory change.

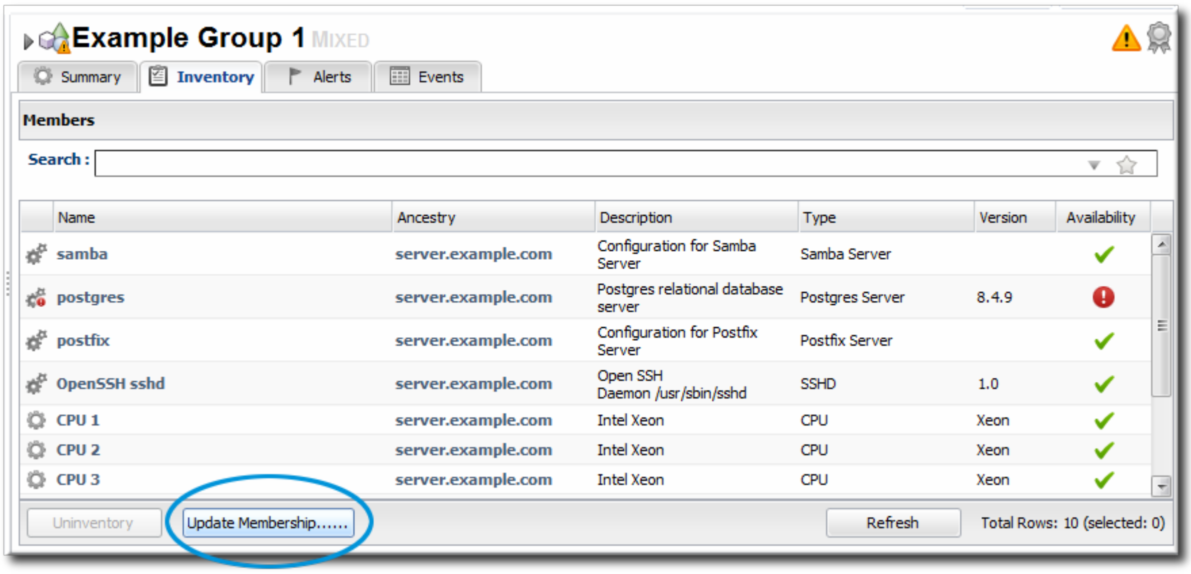
1. In the **`Inventory`** tab in the top menu, select the compatible or mixed groups item in the **`Groups`** menu on the left.



2. Click the name of the group.

3. Open the **`Inventory`** tab for the group, and open the **`Members`** submenu.

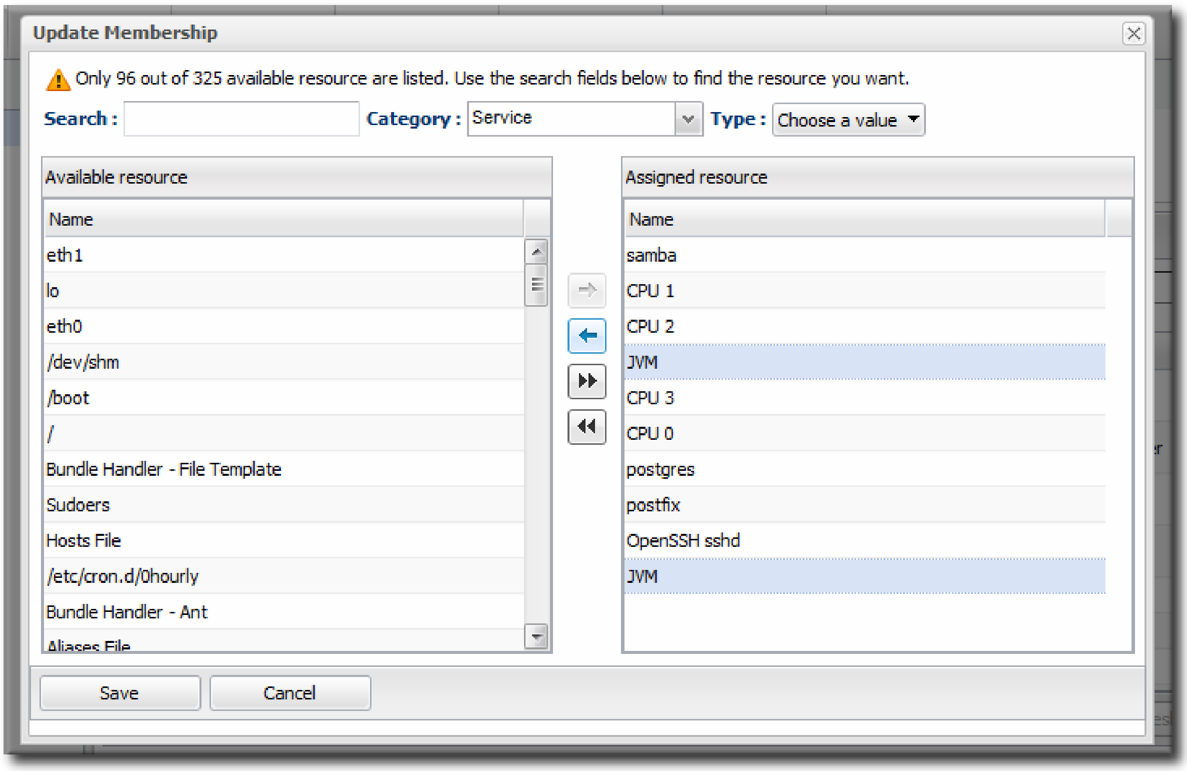
4. Click the `Update Membership` button at the bottom of the page.



Name	Ancestry	Description	Type	Version	Availability
samba	server.example.com	Configuration for Samba Server	Samba Server		✓
postgres	server.example.com	Postgres relational database server	Postgres Server	8.4.9	●
postfix	server.example.com	Configuration for Postfix Server	Postfix Server		✓
OpenSSH sshd	server.example.com	Open SSH Daemon /usr/sbin/sshd	SSHD	1.0	✓
CPU 1	server.example.com	Intel Xeon	CPU	Xeon	✓
CPU 2	server.example.com	Intel Xeon	CPU	Xeon	✓
CPU 3	server.example.com	Intel Xeon	CPU	Xeon	✓

Total Rows: 10 (selected: 0)

5. Select the resources to add to the group from the box on the left; to remove members, select them from the box on the right. Use the arrows to move the selected resources. To select multiple resources, use `Ctrl`+click.



Only 96 out of 325 available resource are listed. Use the search fields below to find the resource you want.

Available resource	Assigned resource
Name	Name
eth1	samba
lo	CPU 1
eth0	CPU 2
/dev/shm	JVM
/boot	CPU 3
/	CPU 0
Bundle Handler - File Template	postgres
Sudoers	postfix
Hosts File	OpenSSH sshd
/etc/cron.d/0hourly	JVM
Bundle Handler - Ant	
Aliases File	

6. Click the **Save** button.

6.4. EDITING COMPATIBLE GROUP CONNECTION PROPERTIES

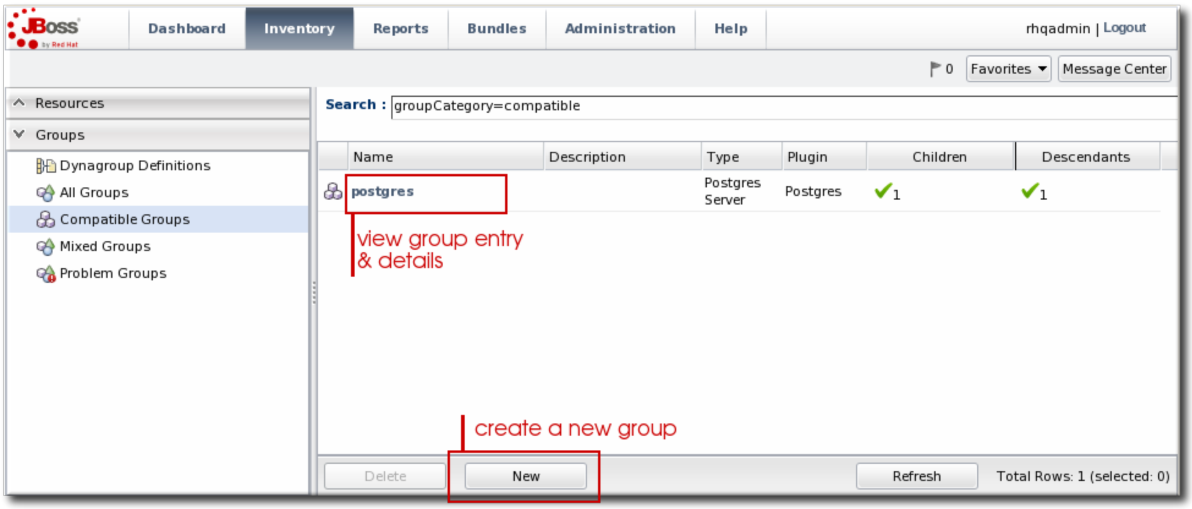
Compatible groups manage the *connection properties* of the group members as part of the inventory. Since a compatible group can only contain members of the same resource type, it is possible to see an aggregate view or average of all of their individual connection properties. The connection settings define how the agent or server connects to the resource.

The rules are for the values of compatible group connections are simple:

- If all of the resources in the group have identical values for a property, the group connection property is that exact value.
- If even one resource has a different value than the rest of the resources in the group, that property will have a special marker value of ~ *Mixed Values* ~.

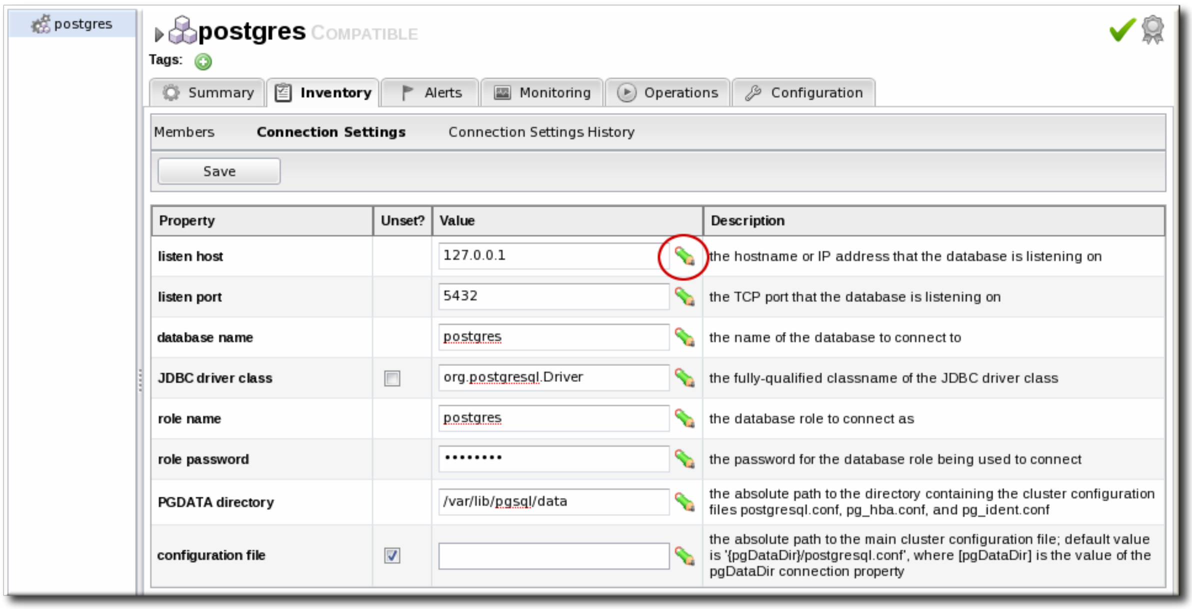
To edit the connection properties:

1. In the **`Inventory`** tab in the top menu, select the **`Compatible Groups`** item in the **`Groups`** menu on the left.



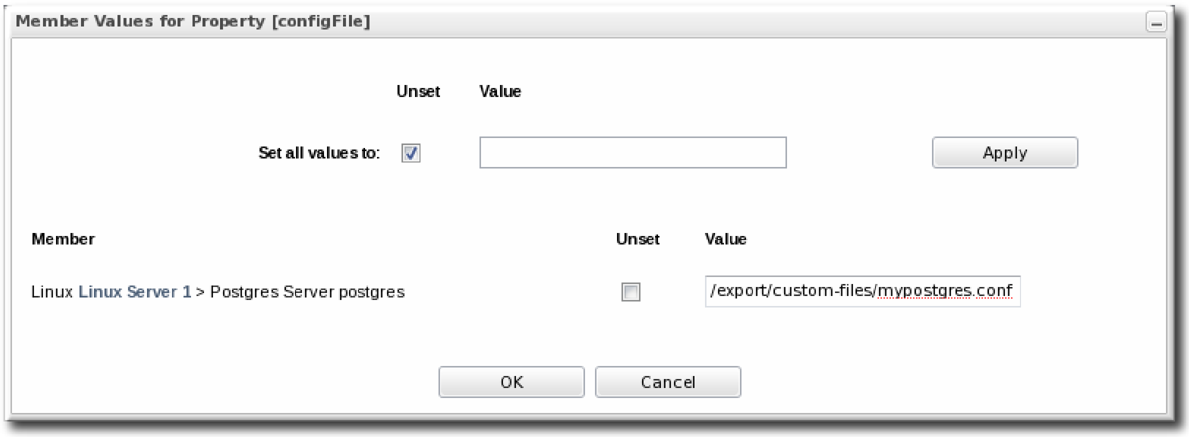
Name	Description	Type	Plugin	Children	Descendants
postgres		Postgres Server	Postgres	✓1	✓1

2. Click the name of the compatible group.
3. Open the **`Inventory`** tab for the group, and click the **`Connection Settings`** sub-item.
4. To edit a property, click the green pencil by the field.



Property	Unset?	Value	Description
listen host		127.0.0.1	the hostname or IP address that the database is listening on
listen port		5432	the TCP port that the database is listening on
database name		postgres	the name of the database to connect to
JDBC driver class	☐	org.postgresql.Driver	the fully-qualified classname of the JDBC driver class
role name		postgres	the database role to connect as
role password		••••••••	the password for the database role being used to connect
PGDATA directory		/var/lib/pgsql/data	the absolute path to the directory containing the cluster configuration files postgresql.conf, pg_hba.conf, and pg_ident.conf
configuration file	☑		the absolute path to the main cluster configuration file; default value is '{pgDataDir}/postgresql.conf', where [pgDataDir] is the value of the pgDataDir connection property

5. To change all resources to the same value, click the **`Unset`** checkbox for the field **`Set all values to...`**. To change a specific resource, click the **`Unset`** checkbox for that resource and then give the new value.


Member Values for Property [configFile]

	Unset	Value	
Set all values to:	✓		Apply

Member	Unset	Value
Linux Linux Server 1 > Postgres Server postgres	☐	/export/custom-files/mypostgres.conf

NOTE

Refreshing the inventory tab shows the *current* values of the connection properties for each resource in the group. If the update has not yet completed, but it has successfully changed some of the resource's connection properties, intermediate values are displayed. Just ignore these values. Once the updates have completed, refresh the page to view final results.

The **`Connection Settings History`** sub-item shows the changes made to the connection properties. If there is a failure, clicking the hyperlink in the **`Date Created`** column opens any relevant error messages.

CHAPTER 7. USING DYNAMIC GROUPS

A dynamic groups specifies a search term to use to search the inventory and identify matching resources to belong to the group. Since the search results change automatically as results are added and removed from the inventory, the group membership is always changing and always current. Using dynamic groups helps automate management tasks for large inventories.

NOTE

Dynamic groups are also referred to by the nickname *dynagroups*.

Enterprise resources can be grouped by cluster identifier, broadcast group, logical service layer, geographical location, security domain, or any other logical grouping.

Individual resources can potentially belong to multiple groups, in large inventories it is important to know how different group definitions will affect the enterprises resources.

7.1. ABOUT DYNAMIC GROUPS SYNTAX

Dynamic groups are configured through *group definitions*. A group definition uses *expressions* which define searches for resources, along with other information about the group like the recalculation interval.

Dynamic groups have an expression syntax very similar to the one used for dynamic searches (Section 2.2, "About the Dynamic Search Syntax").

7.1.1. General Expression Syntax

The *expression* is a search condition that is centered around a specific resource attribute, either by a specific value of an attribute or simply by the presence of an attribute.

An expression defines a way to group resources:

- By a specific resource attribute or value (a *simple* expression)
- By the resource type (a *pivoted* expression)
- By membership in another group (a *narrowing* expression)

A single group definition can have multiple expressions. The order of expressions in the group definition does not matter; for example, both of these expressions are interpreted exactly the same when calculating the group members:

```
expression 1
exprA1
exprA2
groupby exprB1
groupby exprB2

expression 2
exprA2
exprA1
groupby exprB2
groupby exprB1
```

NOTE

When multiple expressions are used in a group definition, they are treated as logical AND expressions, and a resource must match all the criteria to belong to the group.

Any empty lines between expressions in a dynagroup definition are ignored.

The possible resource properties cover resource information like the resource name, type, plug-in, version, configuration property, and inventory ID number.

Table 7.1. Dynamic Group Properties

Type	Supported Attributes
Related to the resource itself	
resource	id name version parent grandparent children
Related to the resource type	
resourceType	plug-in name category (platform, server, service)
Related to the resource configuration	
plug-inConfiguration	Any plugin configuration property
resourceConfiguration	Any resource configuration property
Related to the resource monitoring data	
traits	Any monitoring trait
availability	The current state, either UP or DOWN

If an expression has the structure **resource.***attribute*, then it applies to the resource which will be a member of the group. However, it is possible to use an attribute in an ancestor or child entry to identify a group member recursively.

For example, to add a resource as member which as an inventory ID of 10001, the expression is:

```
resource.id = 10001
```

To add all of the *children* of a resource with an ID of 10001 as group members, use the prefix **resource.parent**:

```
resource.parent.id = 10001
```

There are four possible prefixes for all of the resource attributes in Table 7.1, “Dynamic Group Properties”:

- resource
- resource.child
- resource.parent
- resource.grandParent

If a definition is so restrictive that no resources match the filters, no group is created. JBoss ON will actually suppress a group from being created by a group definition if it would result in an empty group. Because there are no empty groups, there are no extraneous groups listed in the inventory, which makes managing inventories easier and better reflects your real infrastructure.

7.1.2. Simple Expressions: Looking for a Value

A simple expression uses an attribute-value pair or triad in this format:

```
resource.attribute[string-expression] = value
```

For example:

```
resource.parent.type.category = Platform
```

Not every resource attribute has an additional *string-expression*; a *string-expression* is basically a sub-attribute. For example, **resource.trait** is the generic resource attribute, and a sub-attribute like **partitionName** identifies the actual parameter.

Simple expressions usually search for resources based on an explicit value, but resources may have attributes present with null values, and those null values would be not returned with a simple expression. The **empty** keyword searches for resources which have a specific attribute with a null value:

```
empty resource.attribute[string-expression]
```

If the **empty** keyword is used, then there is no *value* given with the expression.

Simple expressions can also use a **`not empty`** keyword, which looks for every resource with that attribute, regardless of the attribute value, as long as it is not null. As with the **`empty`** keyword, there is no reason to give a *value* with the expression, since every value matches the expression.

```
not empty resource.attribute[string-expression]
```

7.1.3. Pivot Expressions: Grouping by an Attribute

Simple expressions create a single group, because they are based on the specific results of the search. Alternatively, a *pivot expression* creates multiple groups because it identifies resources which belong to the group solely based on whether an attribute exists; it creates subgroups based on the values. A pivot expression uses the **`groupby`** keyword:

```
groupby resource.attribute
```

Pivoted expressions create groups based on unique occurrences of an attribute value. For example, the ***`parent.name`*** attribute creates a unique group based on every parent resource.

```
groupby resource.parent.name
```

For the resources in Figure 7.1, "Resources and Parents", the pivot expression creates groups for the three parents within the resource hierarchy: ResourceParentA, ResourceParentB, and ChildA2.

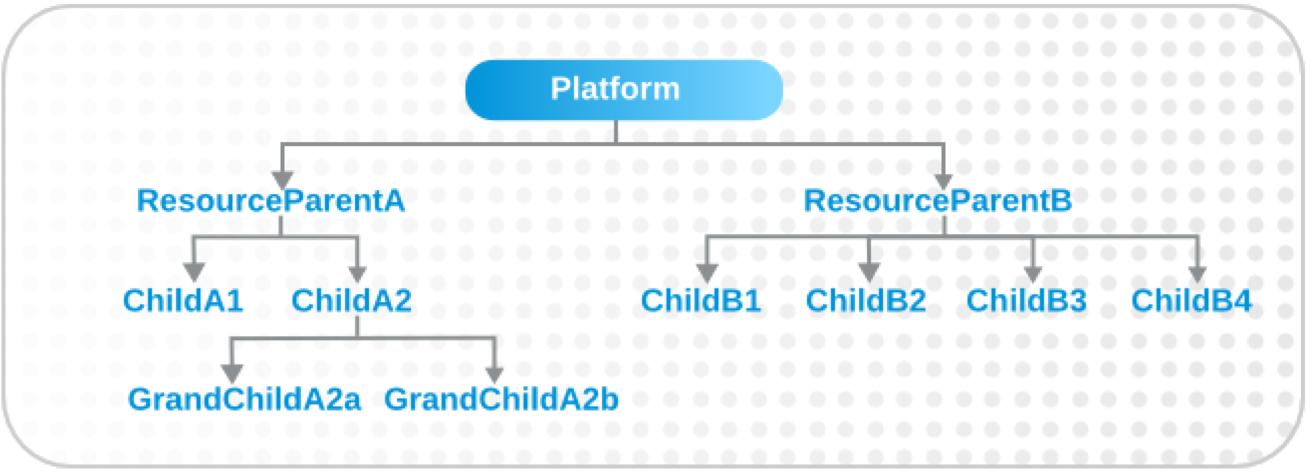


Figure 7.1. Resources and Parents

If the overall group definition includes resources with null values, then the pivot expression creates a special subgroup that contains those resources.

7.1.4. Narrowing Expressions: Members of a Group

Expressions are generally evaluated across the *entire* inventory. For example, setting a pivot expression for the JBoss AS 7 Server resource type checks the inventory for every EAP 6 or AS 7 instance.

In some occasions, particularly if trying to create fine-grained groups for access control or bundle management, it is simpler to define an expression to run against the members of an existing group, rather than trying to devise a complex expression to get a subset of the same type of resource.

This is done by specifying the *memberof* keyword. This specifies a group name (compatible group, mixed group, autogroups, recursive group, even another dynagroup), and only members of that group are evaluated for other matching expressions.

NOTE

The **memberof** keyword specifies a group name. If the group is a recursive group, then all of the recursive members are included as part of the group for evaluation.

For example, an administrator creates different groups for application development related resources for development, QE, and production teams. These are mixed groups of platforms, Postgre databases, EAP 6 servers, and web contexts. When new resources are deployed, a CLI script is run to import the resources and automatically update the resource groups. Access control rules and roles need to use both the main groups — such as *Dev Stack Resource Group*, a mixed group — *and* subgroups within that group, based on resource types. Rather than creating and updating multiple groups and roles every time resources are deployed or removed, the administrator creates a dynagroup expression which first *narrows* the scope of the expression to the given team resource group, and then pivots by resource type. The dynagroup definition only needs to be set once, and it is dynamically updated with inventory changes every time the group is recalculated.

```
memberof = "Dev Resource Group"
groupby resource.type.name
```

NOTE

The group membership, as with other expressions, is only updated with **memberof** changes when the dynagroup is recalculated.

Multiple **memberof** expressions are allowed in the group definition, with each **memberof** expression referencing a single group. If multiple **memberof** expressions are used, they are treated as AND expressions; any matching resource must be a member of *all* specified groups.

7.1.5. Compound Expressions

Multiple expressions can be used in a single dynagroup definition; these are *compound* expressions.

When multiple expressions are used in a group definition, they are treated as logical AND expressions, and a resource must match all the criteria to belong to the group. (Any empty lines between expressions in a dynagroup definition are ignored.)

For example, this basic expression searches for every resource which has the platform as its parent:

```
resource.parent.type.category = Platform
```

That could return a very long list of servers and services. That initial list can be further filtered by adding another simple expression, which filters by name:

```
resource.parent.type.category = Platform
resource.name.contains = JBossAS
```

Only resources with the platform as the parent *and* with the string *JBossAS* in their name will be added to the group.

Pivoted expressions can also be used in compound expressions. Every line must have the *groupby* keyword, not only the first line.

```
groupby resource.type.plugin
groupby resource.type.name
groupby resource.parent.name
```

A compound expression can contain both simple and pivoted expressions. This creates a compatible group for every unique server type on the platform.

```
resource.type.category = server
groupby resource.type.plugin
groupby resource.type.name
groupby resource.parent.name
```

Lastly, compound expressions can include the empty and not empty keywords. For example, simple expressions can be used to identify JBoss servers based on the resource type and name. Then, to identify which JBoss servers are unsecured, the expression can filter for JBoss servers with a principal connection property with an empty value.

```
resource.type.plugin = JBossAS
resource.type.name = JBossAS Server
empty resource.pluginConfiguration[principal]
```

7.1.6. *Un*supported Expressions

There are restrictions on how multiple expressions can be used together.

All expressions must be in the same configuration area.

All given configuration properties in an expression must be only from the resource configuration or only from the plug-in configuration. Expressions cannot be taken from both.

Each property must only be used once.

A property can only be used once in a dynagroup definition.

```
valid
resource.trait[x] = foo

not valid
resource.trait[x] = foo
resource.trait[y] = bar
```

For example, a **`resource.trait`** expression can only occur once in a definition:

```
resource.grandParent.trait[Trait.hostname].contains = stage
resource.parent.type.plugin = JBossAS5
resource.type.name = Web Application (WAR)
```

If it is used a second (or more) time, then the subsequent use fails and causes the definition not to be parsed.

```
resource.grandParent.trait[Trait.hostname].contains = stage
resource.parent.type.plugin = JBossAS5
resource.type.name = Web Application (WAR)
resource.trait[contextRoot] = jmx-console
```

This results in calculation errors:

```
There was a problem calculating the results:
java.lang.IllegalArgumentException: org.hibernate.QueryParameterException:
could not locate named parameter [arg2]
```

The *[arg2]* error is a sign that multiple expressions of the same type were used and that the second and subsequent expressions caused a calculation failure.

This is true even if the property type is used with *different* resource contexts.

```
resource.parent.trait[x] = foo
resource.grandParent.trait[y] = bar
```

In the previous example, one trait applied to the grandparent resource and one trait applied to the resource itself. This still failed because the trait property was used twice, even though it was for different resources.

7.1.7. Dynagroup Expression Examples

A single group definition can have multiple expressions, even mixing simple and pivoted expressions in a single definition. Many of these examples require multiple expressions to complete the definition.

Example 7.1. JBoss Clusters

```
resource.type.plugin = JBossAS
resource.type.name = JBossAS Server
groupby resource.trait[partitionName]
```

Example 7.2. A Group for Each Platform Type

```
resource.type.plugin = Platforms
resource.type.category = PLATFORM
groupby resource.type.name
```

Example 7.3. Autogroups

```
groupby resource.type.plugin
groupby resource.type.name
groupby resource.parent.name
```

NOTE

This could create a large number of groups in large inventories.

Example 7.4. Raw Measurement Tables

```
resource.type.plugin= Postgres
resource.type.name = Table
resource.parent.name = rhq Database
resource.name.contains = rhq_meas_data_num_
```

Example 7.5. Only Agents with Multicast Detection

```
resource.type.plugin= RHQAgent
resource.type.name = RHQ Agent
resource.resourceConfiguration[rhq.communications.multicast-
detector.enabled] =  true
```

Example 7.6. Only Windows Platforms with Event Tracking

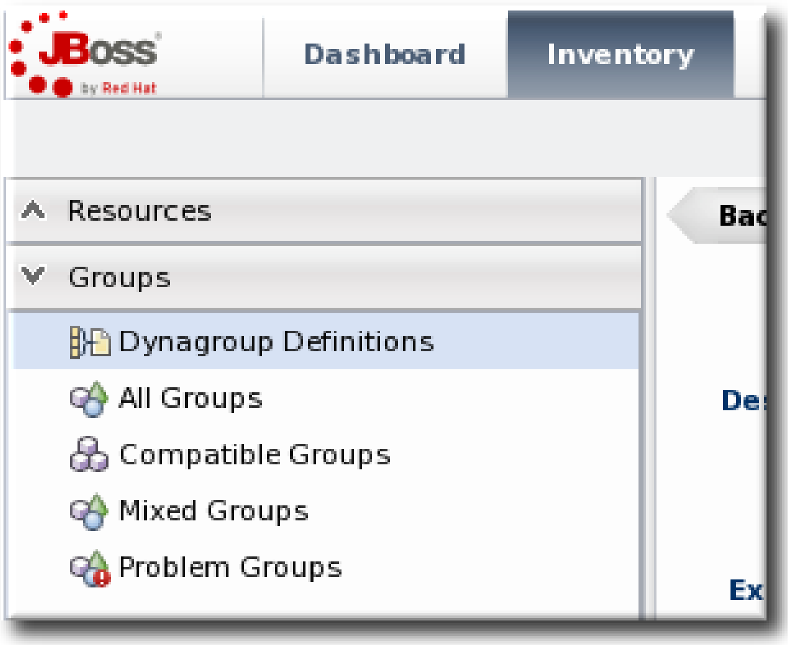
```
resource.type.plugin= Platforms
resource.type.name = Windows
resource.pluginConfiguration[eventTrackingEnabled] =  true
```

Example 7.7. JBoss AS Servers by Machine

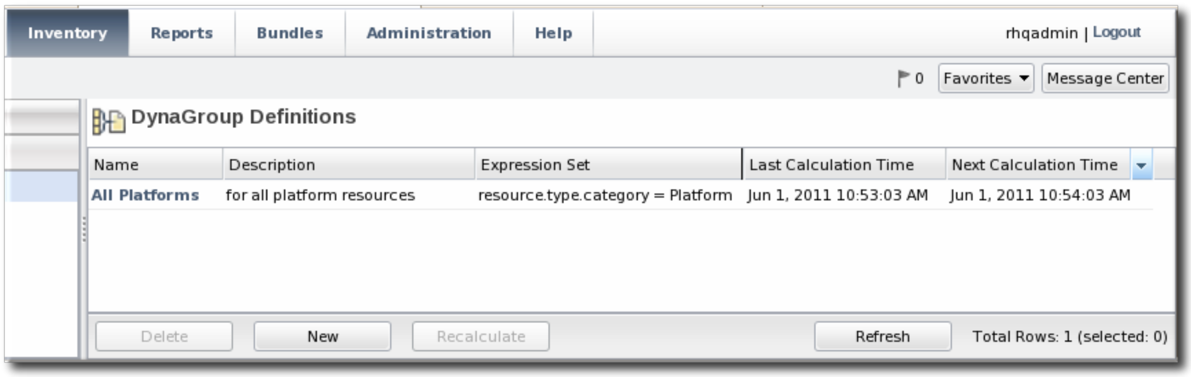
```
groupby resource.parent.trait[Trait.hostname]
resource.type.plugin = JBossAS
resource.type.name = JBossAS Server
```

7.2. CREATING DYNAMIC GROUPS

1. Click the **`Inventory`** tab in the top menu.
2. In the **`Groups`** menu box on the left, click the **`Dynagroup Definitions`** link.



3. Click the **New** button to open the dynamic group definition form.

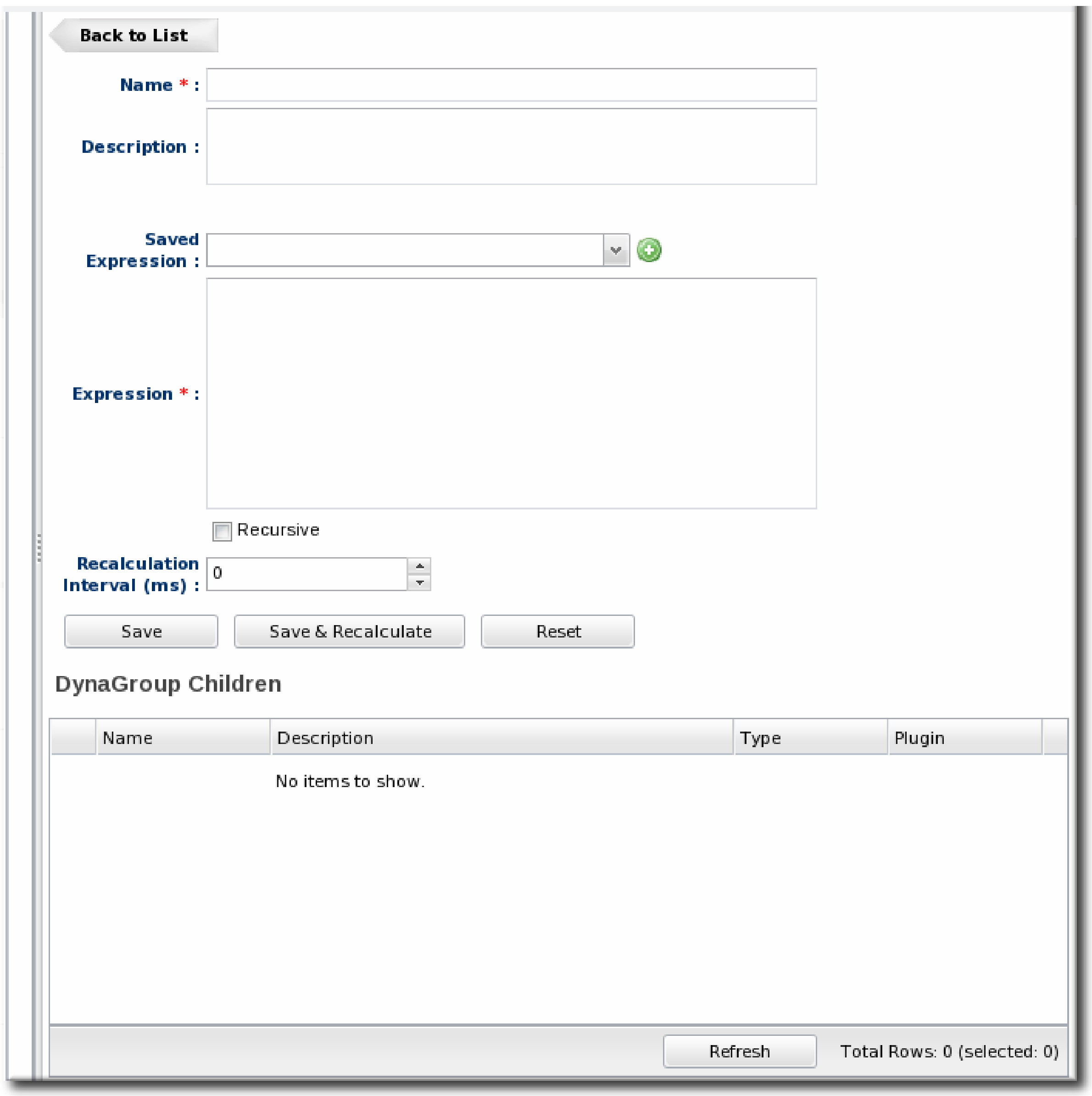


DynaGroup Definitions

Name	Description	Expression Set	Last Calculation Time	Next Calculation Time
All Platforms	for all platform resources	resource.type.category = Platform	Jun 1, 2011 10:53:03 AM	Jun 1, 2011 10:54:03 AM

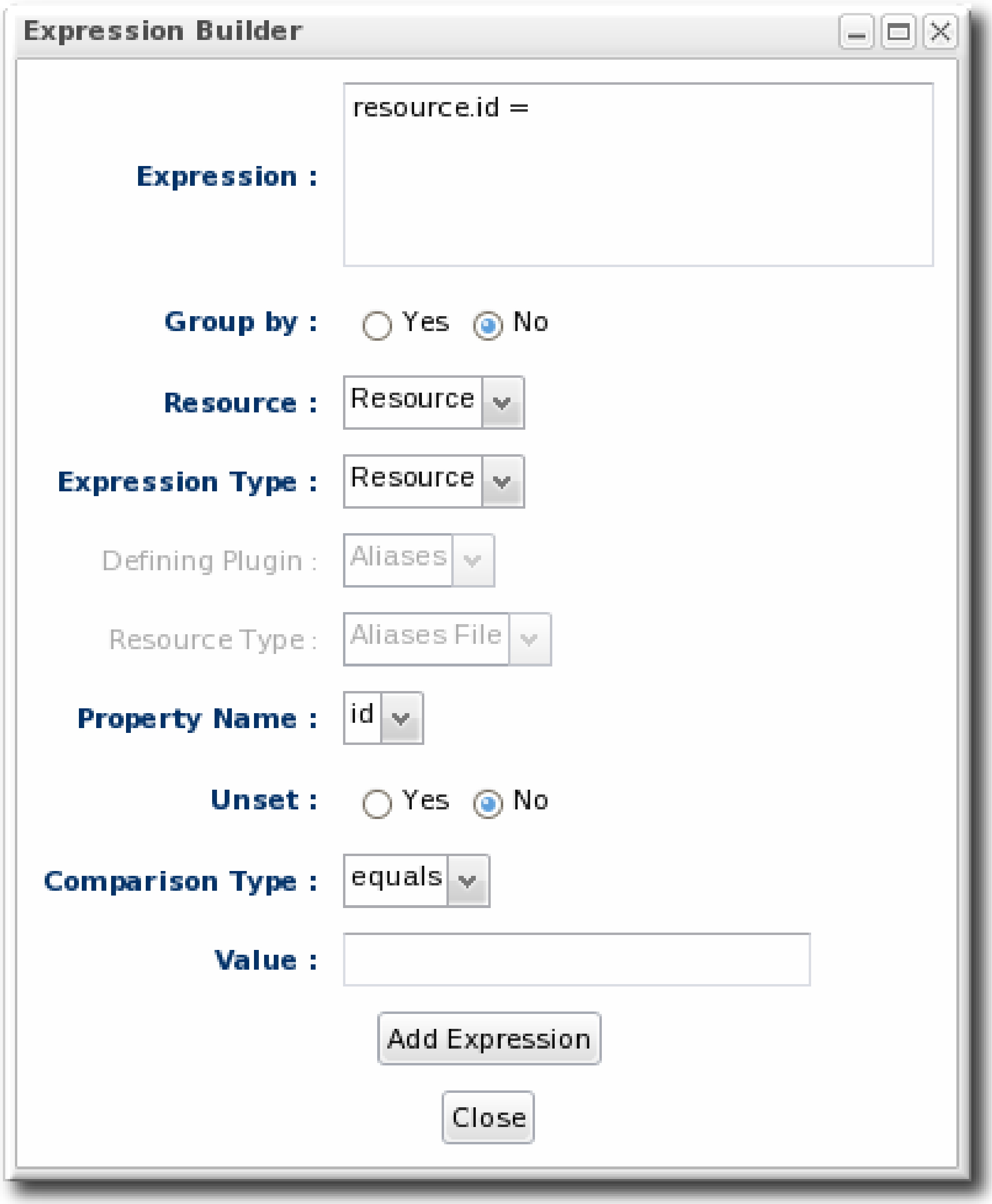
Total Rows: 1 (selected: 0)

4. Fill in the name and description for the dynamic group. The name can be important because it is prepended to any groups created by the definition, as a way of identifying the logic used to create the group.



5. Fill in the search expressions. This can be done by entering expressions directly in the **`Expression`** box or by using a saved expression.

   Saved expressions are have a wizard to help build and validate the expressions. To create a saved expression, click the green button by the drop-down menu. Several options for the expression are active or inactive depending on the other selections; this prevents invalid expressions.



Expression : resource.id =

Group by : Yes [✓ No]

Resource : Resource

Expression Type : Resource

Defining Plugin : Aliases

Resource Type : Aliases File

Property Name : id

Unset : Yes [✓ No]

Comparison Type : equals

Value :

The **Expression** box at the top shows the currently created expression.

6. After entering the expressions, set whether the dynamic group is recursive.

7. Set an optional recalculation interval. By default, dynamic groups do not recalculate their members automatically, meaning the recalculation value is set to 0. To recalculate the group membership, set the **Recalculation interval** to the time frequency, in milliseconds.

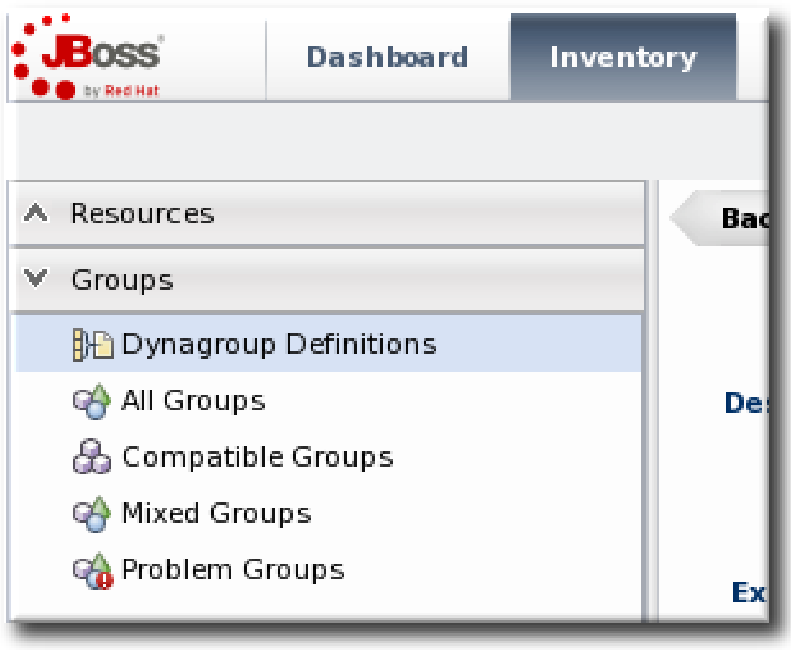
NOTE

Recalculating a group definition across large inventories could be resource-intensive for the JBoss ON server, so be careful when setting the recalculation interval. For large inventories, set a longer interval, such as an hour, to avoid affecting the JBoss ON server performance.

7.3. RECALCULATING GROUP MEMBERS

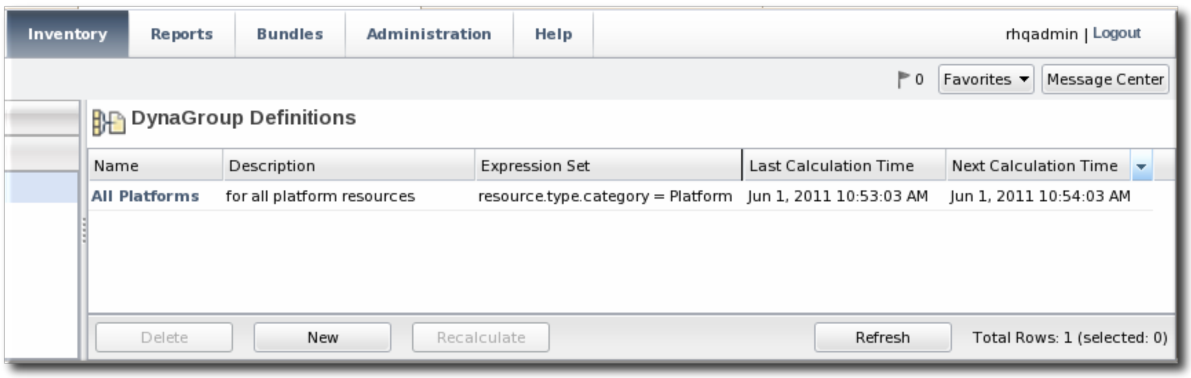
Dynamic groups can be recalculated apart from whatever interval is set in the group definition. The recalculation interval in the group definition is a relative value based on the last update time, so initiating a recalculation manually will not conflict or interfere with the normal recalculation; it simply proceeds after the specified amount of times elapses.

1. Click the **`Inventory`** tab in the top menu.



2. In the **`Groups`** menu on the left, click the **`Dynagroup Definitions`** link.

3. In the list of dynagroups, select the row of the dynagroup definition to calculate.



DynaGroup Definitions

Name	Description	Expression Set	Last Calculation Time	Next Calculation Time
All Platforms	for all platform resources	resource.type.category = Platform	Jun 1, 2011 10:53:03 AM	Jun 1, 2011 10:54:03 AM

Total Rows: 1 (selected: 0)

4. Click the **`Recalculate`** button at the bottom of the table.

CHAPTER 8. CREATING USER ACCOUNTS

Users are part of the overall security planning for JBoss ON, even though they don't have access controls set on their accounts individually.

8.1. MANAGING THE RHQADMIN ACCOUNT

When JBoss ON is installed, there is a default superuser already created, **`rhqadmin`**. This superuser has the default password **`rhqadmin`**.

NOTE

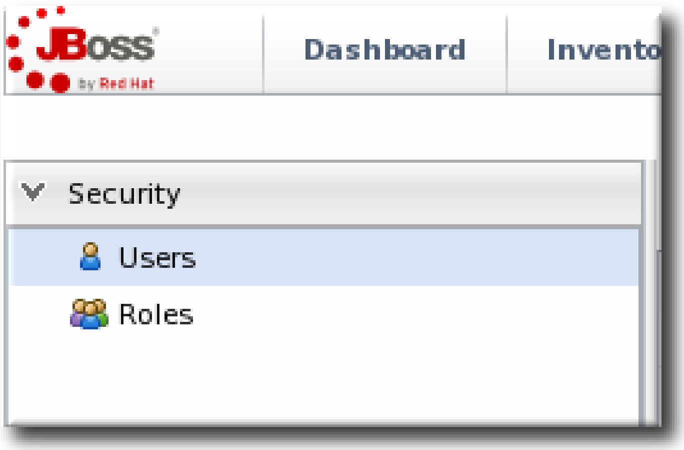
The **`rhqadmin`** account cannot be deleted, even if other superuser accounts are created. Additionally, the role assignments for **`rhqadmin`** cannot be changed; it is always a superuser account.

IMPORTANT

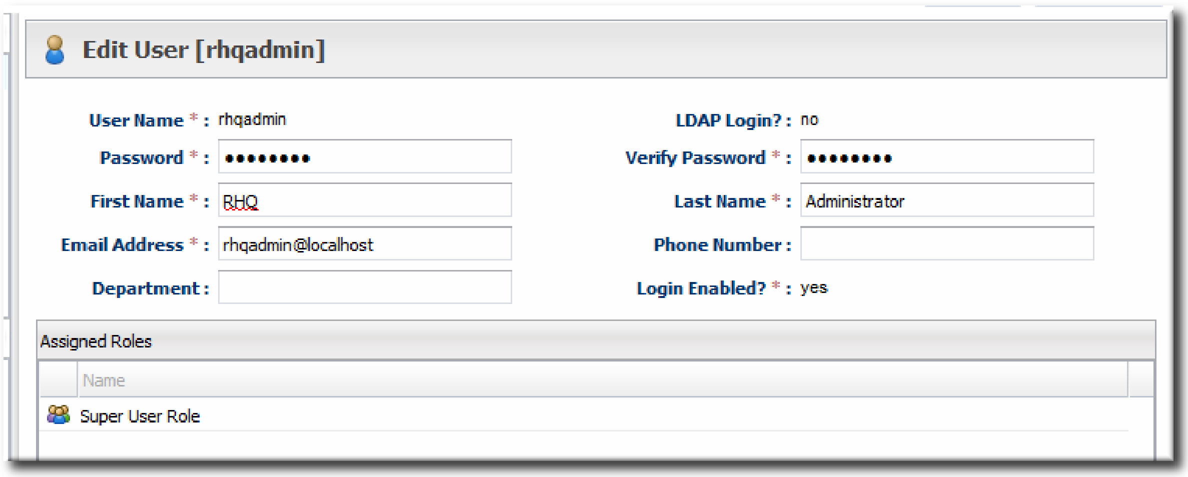
If a user is deleted, scheduled operations owned by the user are canceled.

When you first log into JBoss ON after installation, change the superuser password.

1. Click the **`Administration`** tab in the top menu.
2. In the **`Security`** table on the left, select **`Users`**.



3. Click the name of **`rhqadmin`**.
4. In the edit user form, change the password to a new, complex value.



Edit User [rhqadmin]

User Name * :	rhqadmin	LDAP Login? :	no
Password * :	••••••••	Verify Password * :	••••••••
First Name * :	RHQ	Last Name * :	Administrator
Email Address * :	rhqadmin@localhost	Phone Number :	
Department :		Login Enabled? * :	yes

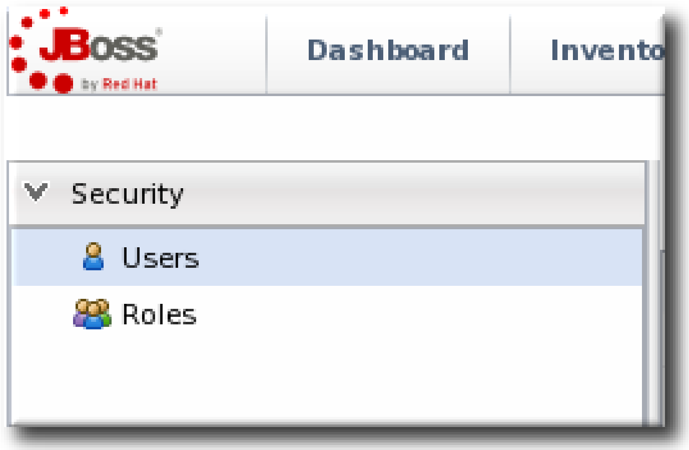
Assigned Roles

Name
Super User Role

8.2. CREATING A NEW USER

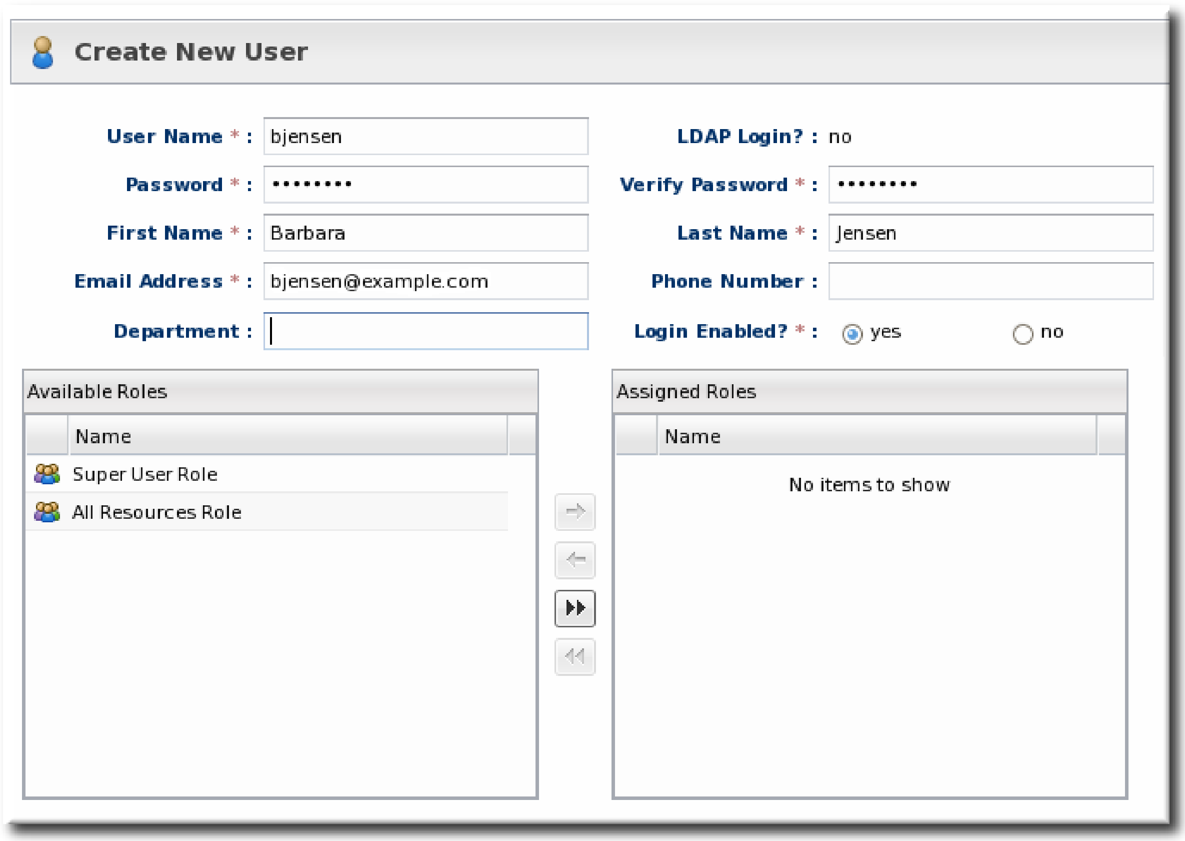
1. Click the **`Administration`** tab in the top menu.

2. In the **`Security`** table on the left, select **`Users`**.



3. Click the **`NEW`** button at the bottom of the list of current users.

4. Fill in description of the new user. The **`Enable Login`** value must be set to **`Yes`** for the new user account to be active.



Create New User

User Name * :	bjensen	LDAP Login? :	no
Password * :	•••••••••	Verify Password * :	•••••••••
First Name * :	Barbara	Last Name * :	Jensen
Email Address * :	bjensen@example.com	Phone Number :	
Department :		Login Enabled? * :	[✓ yes] no

Available Roles

Name
Super User Role
All Resources Role

Assigned Roles

Name
No items to show

5. Select the required role from the **Available Roles** area, and then click the arrow pointing to the **Assigned Roles** to assign the role.

6. Click the **Save** button to save the new user with the role assigned.

8.3. EDITING USER ENTRIES

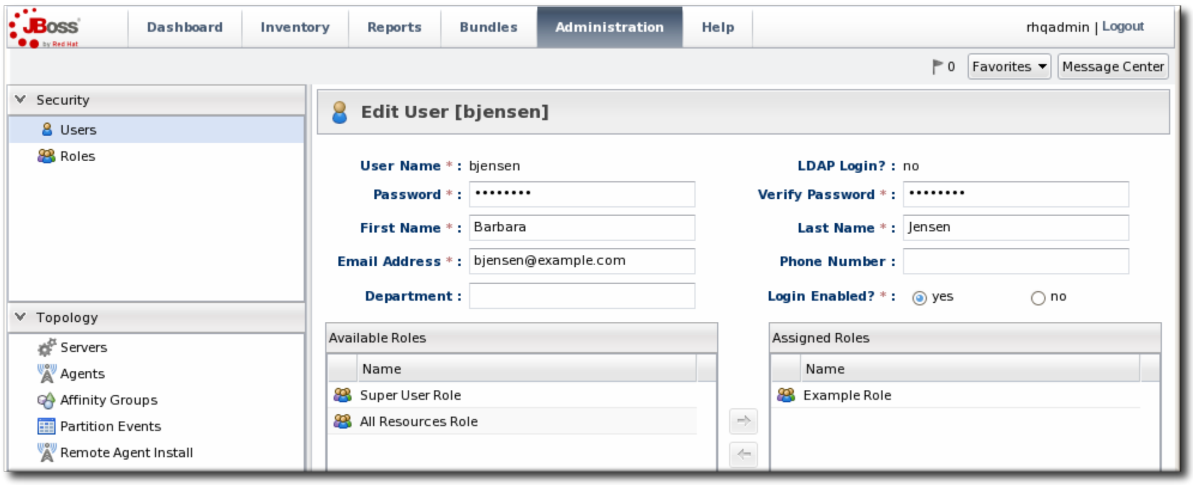
All users can edit their own account details, and users with administrative rights (who belong to a role which grants them rights over user entries) can edit other users' entries.

1. Click the **Administration** tab in the top menu.

2. From the **Security** menu, select **Users**.

3. Click the name of the user whose entry will be edited.

4. In the edit user form, change whatever details need to be changed, and save.



Edit User [bjensen]

User Name * :	bjensen	LDAP Login? :	no
Password * :	••••••••	Verify Password * :	••••••••
First Name * :	Barbara	Last Name * :	Jensen
Email Address * :	bjensen@example.com	Phone Number :	
Department :		Login Enabled? * :	[✓ yes] no

Available Roles

Name
Super User Role
All Resources Role

Assigned Roles

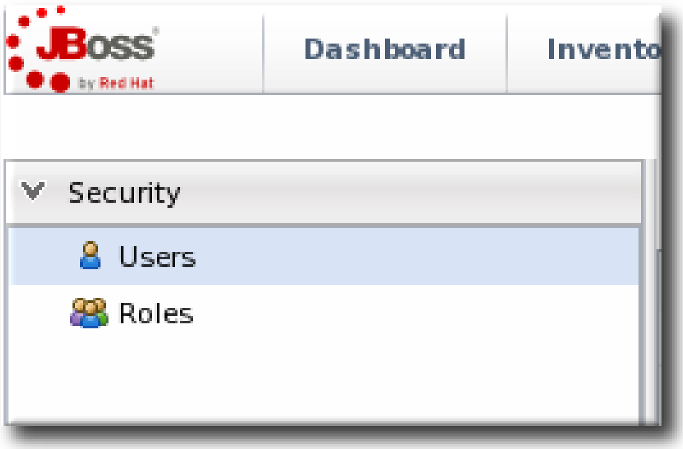
Name
Example Role

8.4. DISABLING USER ACCOUNTS

User accounts can be temporarily disabled. This can be done for a security review or when there is some kind of breach, but users don't need to be deleted. The **Enable Login** property can prevent the user from logging into the JBoss ON UI and managing resources or making configuration changes.

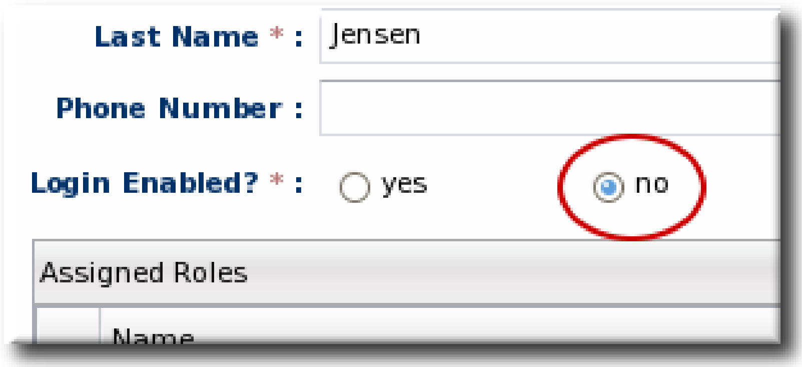
1. Click the **Administration** tab in the top menu.

2. In the **Security** table on the left, select **Users**.



3. Click the name of the user whose entry will be edited.

4. In the edit user form, change the **Enable Login** radio button to **No**.



Last Name * :	Jensen
Phone Number :	
Login Enabled? * :	yes [✓ no]

Assigned Roles

Name

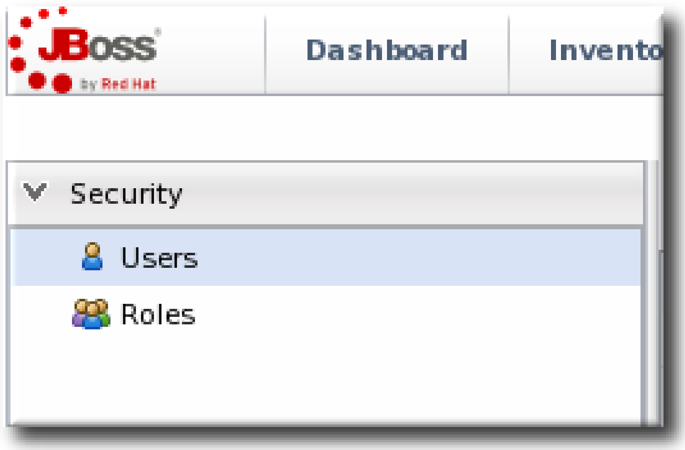
5. Click the **Save** button to save the new user with the role assigned.

The user account can be re-enabled at any time by changing the **`Enable Login`** value back to **`Yes`**.

8.5. CHANGING ROLE ASSIGNMENTS FOR USERS

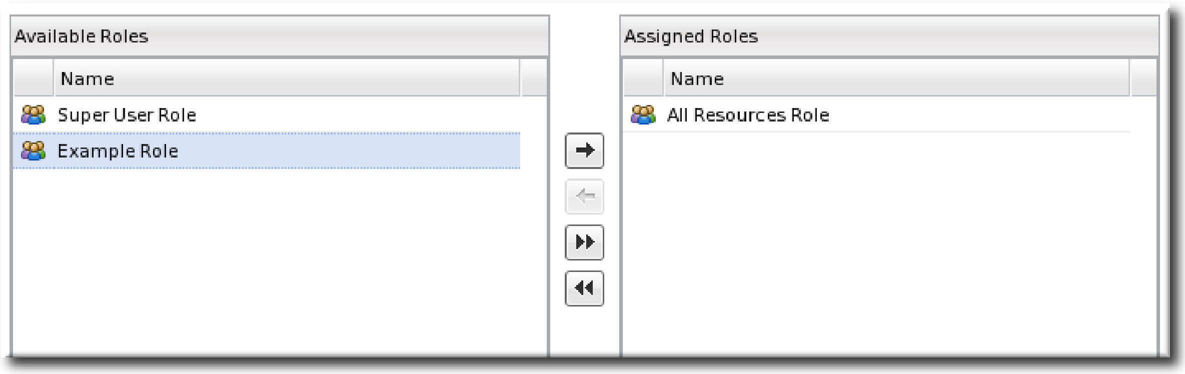
1. Click the **`Administration`** tab in the top menu.

2. From the **`Security`** menu, select **`Users`**.



3. Click the name of the user to edit.

4. To *add* a role to a user, select the required role from the **`Available Roles`** area, click the arrow pointing to the **`Assigned Roles`** area. To *remove* a role, select the assigned role on the right and click the arrow pointing to the left.



5. Click **Save** to save the role assignments.

CHAPTER 9. MANAGING ROLES AND ACCESS CONTROL

In JBoss Operations Network, security is implemented through rules that are set on *users and roles*. Restrictions are set on what users and roles can access and what operations they can perform.

9.1. SECURITY IN JBOSS ON

Security establishes precise relationships between users, resources, and the tasks users can perform. Interactions between users and resources are ordered by including or excluding those users and resources (through groups) in defined *roles*, and then granting the role the ability to perform tasks.

9.1.1. Access Control and Permissions

When a user is allowed to perform a certain operation, that is called a *permission*. All permissions must be explicitly granted to explicit resources. If a user is not given permission to a specific resource group, then the user, by default, has no access to that group — even if the user has permission to perform a task. Likewise, if a user has access to a group but has no permissions assigned, then the user cannot perform any tasks.

Any permissions set in JBoss ON are given to a role, and the members of the role inherit those permissions. Allowing or restricting permissions is *access controls*.

In JBoss ON, there are two levels where access control is granted:

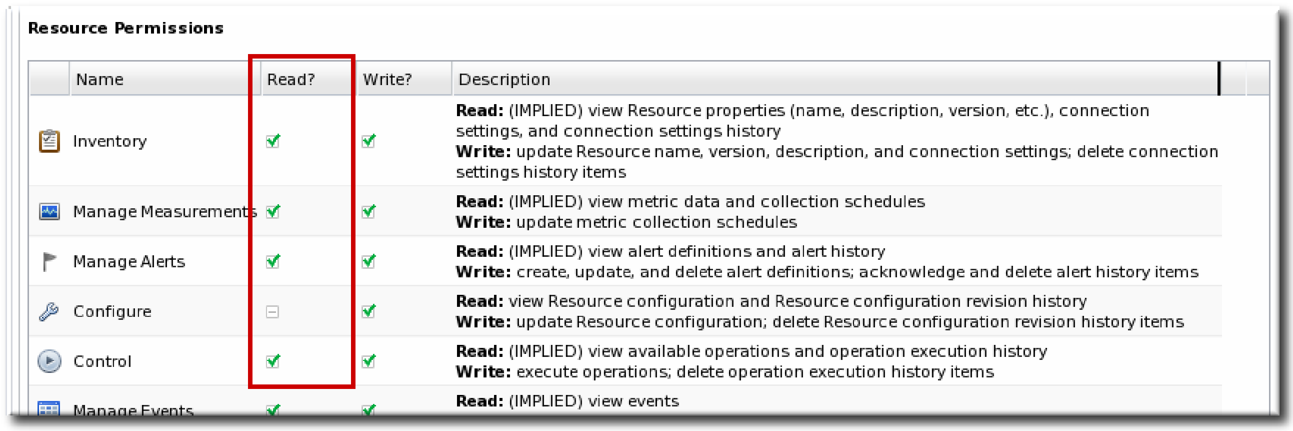
- *Global permissions* apply to JBoss ON server configuration. This covers administrative tasks, like creating users, editing roles, creating groups, importing resources into the inventory, or changing JBoss ON server properties.

- *Resource-level permissions* apply to actions that a user can perform on specific resources in the JBoss ON inventory. These cover actions like creating alerts, configuring monitoring, and changing resource configuration. Resource-level permissions are tied to the subsystem areas within JBoss ON.

All JBoss ON permissions are listed in Table 9.1, “JBoss ON Access Control Definitions” .

For resource-level rights, read and write permissions are granted independently. A user can be granted a right to view (read) resource data without automatically being granted the right to edit that configuration. For example, any user can view the operations history of a resource or view the configured alerts for a resource within the role *even if* that role has not been given edit access to those subsystem areas.

By default, read access is enabled by default for all resource-level rights, with one exception: resource configuration. Resource configuration can be considered a security risk, so even read access is denied by default. Of course, read access, like write access, can be enabled or disabled for any resource-level permission.



Resource Permissions

Name	Read?	Write?	Description
Inventory	✓	✓	**Read:** (IMPLIED) view Resource properties (name, description, version, etc.), connection settings, and connection settings history **Write:** update Resource name, version, description, and connection settings; delete connection settings history items
Manage Measurements	✓	✓	**Read:** (IMPLIED) view metric data and collection schedules **Write:** update metric collection schedules
Manage Alerts	✓	✓	**Read:** (IMPLIED) view alert definitions and alert history **Write:** create, update, and delete alert definitions; acknowledge and delete alert history items
Configure	⊟	✓	**Read:** view Resource configuration and Resource configuration revision history **Write:** update Resource configuration; delete Resource configuration revision history items
Control	✓	✓	**Read:** (IMPLIED) view available operations and operation execution history **Write:** execute operations; delete operation execution history items
Manage Events	✓	✓	**Read:** (IMPLIED) view events

Figure 9.1. Read Access Option

NOTE

Granting a role the right to *change* something does not implicitly grant the right to *delete* something. For example, users with the configuration write permission can edit resource configuration and view configuration history and settings, but they cannot delete elements in the configuration history. Similar constraints are true for users with permission to create and edit operations and alerts — there is no right to delete elements in the resource history.

Deleting elements in the history requires the manage inventory permission.

Table 9.1. JBoss ON Access Control Definitions

Access Control Type	Description
Global Permissions	
Manage Security	Equivalent to a superuser. `Security` permissions grant the user the rights to create and edit any entries in JBoss ON, including other users, roles, and resources, to change JBoss ON server settings, and to control inventory. **WARNING** The `Security` access control level is extremely powerful, so be cautious about which users are assigned it. Limit the number of superusers to as few as necessary.

Access Control Type	Description
Manage Inventory	Allows any operation to be performed on any JBoss ON resource, including importing new resources.
Manage Settings	Allows a user to add or modify any settings in the JBoss ON server configuration itself. This includes operations like deploying plug-ins or using LDAP authentication.
Manage Bundle Groups	Allows a user to add and remove members of a bundle group; implicitly, it includes the permission to view bundles. This is analogous to the Manage Inventory permission for resources. **NOTE** This permission is required for all bundle-level create, deploy, and delete permissions.
Deploy Bundles to Groups	Allows a user to deploy a bundle to any resource group to which the user has access.
View Bundles	Allows a user to view all bundles, regardless of the bundle group assignment.
Create Bundles	Allows a user to create and update bundle versions. When a bundle is created, it must be assigned to bundle group, unless the user has the View Bundles permission; in that case, a user can create a bundle and leave it unassigned.
Delete Bundles	Allows a user to delete any bundle which he has permission to view.
Manage Bundles (Deprecated)	Allows a user to upload and manage bundles (packages) used for provisioning resources. This permission has been deprecated. It is included for backward-compatibility with older bundle configuration and user roles. However, this permission offered no ability to limit access to certain bundles, groups, or resources (for deployment); without this fine-grained control, this permission could only be applied to high-level administrators to maintain security.
Manage Repositories	Allows a user to access any configured repository, including private repositories and repositories without specified owners. Users with this right can also associated content sources with repositories.

Access Control Type	Description
View Users	Allows a user to view the account details (excluding role assignments) for other users in JBoss ON.
Resource-Level Permissions	
Inventory	Allows a user to edit resource details and connection settings — meaning the information about the resource in the JBoss ON inventory. This does not grant rights to edit the resource configuration.
Manage Measurements	Allows the user to configure monitoring settings for the resource.
Manage Alerts	Allows the user to create alerts and notifications on a resource. Configuring new alert senders changes the server settings and is therefore a function of the global `Settings` permissions.
Control	Allows a user to run operations (which are also called *control actions*) on a resource.
Configure	Allows users to change the configuration settings on the resource through JBoss ON. **NOTE** The user still must have adequate permissions on the resource to allow the configuration changes to be made. This access area has two options: ● Read, which grants read-only access to the resource configuration ● Write, which grants modify access and, implicitly, read access If one of these permissions is not granted to a role, then the users in the role are denied any access to the resource configuration.
Manage Drift	Allows the user to create, modify, and delete resource and template drift definitions. It also allows the user to manage drift information, such as viewing and comparing snapshots.

Access Control Type	Description
Manage Content	Allows the user to manage content providers and repositories that are available to resources.
Create Child Resources	Allows the user to manually create a child resource for the specified resource type.
Delete Child Resources	Allows the user to delete or uninventory a child resource for the specified resource type.
Bundle-Level Permissions	
Assign Bundles to Group	Allows a user to add bundles to a group. For explicit bundle groups, this is the only permission required. To add bundles to the *unassigned* group (which essentially removes it from all group membership), this also requires the global View Bundles permission.
Unassign Bundles from Group	Allows a user to remove bundles from a group.
View Bundles in Group	Allows a user to view any bundle within a group to which the user has permissions.
Create Bundles in Group	Allows a user to create a new bundle within a bundle group to which he has permission. This also allows a user to update the version of an existing bundle within the bundle group.
Delete Bundles from Group	Allows a user to delete both bundle versions and entire bundles from the server, so long as they belong to a group to which the user has permissions.
Deploy Bundles to Group	Allows a user to deploy any bundle which he can view (regardless of create and delete permissions) to any resource within a resource group to which he has permissions.

9.1.2. Access and Roles

JBoss ON handles access to both resources and JBoss ON configuration through *roles*. A role has certain permissions assigned to it, meaning things that members of the role are allowed to do.

Only two types of JBoss ON identities can belong to a role: users and groups.

Users are assigned to a role to be granted those permissions. Users can be added to a role individually or be added as a member of an LDAP group.

Resource groups are assigned to a role to provide a list of resources that those users can perform actions on. Another way of looking at it is that users can only manage resources that they are expressly given access to. Roles define that access.

NOTE

Be sure to create resource groups to assign to any custom roles you create. If no resource groups are assigned to a role, then none of the members of the role can see any resources. Creating groups is described in Section 6.2, “Creating Groups” .

One convenient feature of roles is that users can be automatically assigned to roles by assigning an LDAP group to the role (Section 10.3.2, “Associating LDAP User Groups to Roles”). All of the LDAP users who belong to that group are automatically members of the role. (This is similar to the simplicity of using LDAP user to create JBoss ON users by enabling LDAP authentication, in Section 10.2.3, “Configuring LDAP User Authentication”.)

There are two roles already configured in JBoss ON by default:

- A superuser role provides complete access to everything in JBoss ON. This role cannot be modified or deleted. The user created when the JBoss ON server was first installed is automatically a member of this role.
- An *all resources* role exists that provides full permissions to every resource in JBoss ON (but not to JBoss ON administrative functions like creating users). This is a useful role for IT users, for example, who need to be able to change the configuration or set up alerts for resources managed by JBoss ON but who don't require access over JBoss ON server or agent settings.

9.1.3. Access and Groups

A role is essentially the intersection of four elements: users, resource groups, bundle groups, and the permissions allowed to those groups.

Access defines a relationship. The most direct relationship is with the users: a role allows a user to do something to some resource within JBoss ON. However, access in roles also define relationships *between groups*.

Group relationships function in two ways. First, it can grant different types of permission to a main group and a subgroup, such as granting view access to all resources and restricting configuration access to just some. It can also grant different types of access to two distinct groups.

Group relationships grow more complex when there are different types of groups involved, both resource groups and bundle groups. While related to a very specific type of task — content or application deployment — there are a number of different potential interactions, depending on different access to resources, to bundles, and to operation points within the lifecycle (create, deploy, delete).

For most infrastructures, users will not belong to a single role, and a single role probably cannot define all of the interactions to perform a set of tasks. Rather, roles are like a Venn diagram, which overlap each other to create the function list of access rules.

Two Roles to Define Access for a Single User to Resources and Bundles

```
Bundle Group A                  Resource Group A
     |                                |
     V                                V
  Role 1   <---  User A  --->     Role 2
```

```
      ^                              ^
      |                              |
 Permissions                    Permissions
  - view bundles in group        - deploy bundles to group
  - create bundles
```

9.2. CREATING A NEW ROLE

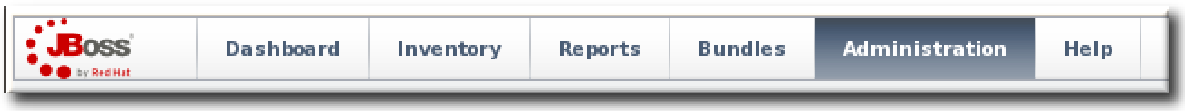
NOTE

Be sure to create resource groups to assign to any custom roles you create. If no resource groups are assigned to a role, then none of the members of the role can see any resources. Creating groups is described in Section 6.2, "Creating Groups" .

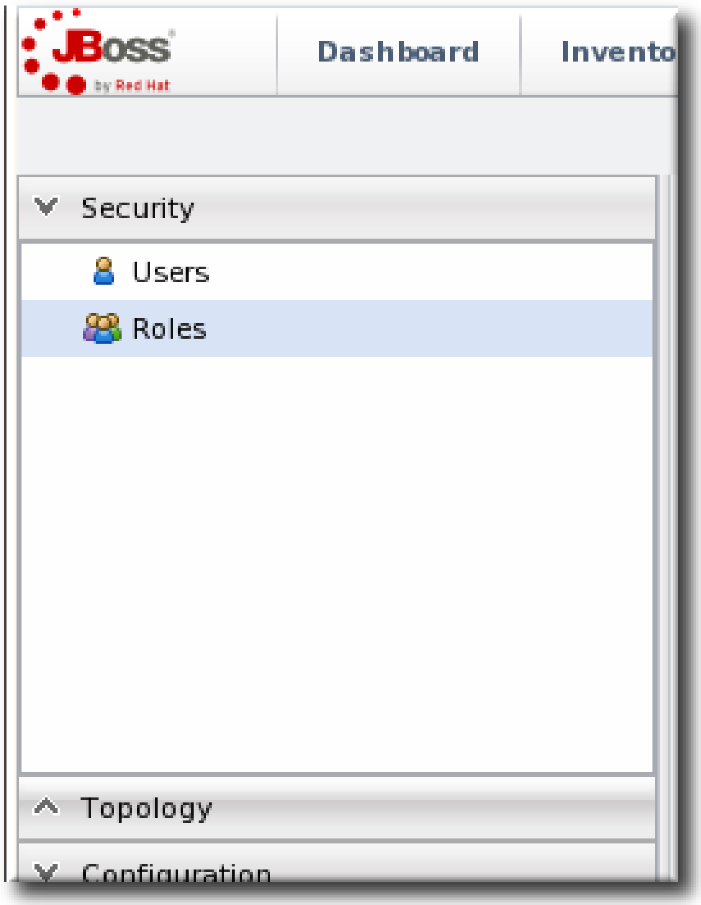
1. Create any resources groups which will be associated with the role. Creating groups is described in Section 6.2, "Creating Groups" .

   By default, JBoss ON uses only *resource groups* to associate with a role, and these are required. However, optional user groups from an LDAP directory can also be assigned to a role, so that the group members are automatically treated as role members. *LDAP groups* must be configured in the server settings, as described in Section 10.3.2, "Associating LDAP User Groups to Roles".

2. In the top menu, click the `Administration` tab.



3. In the **Security** menu table on the left, select the **Roles** item.



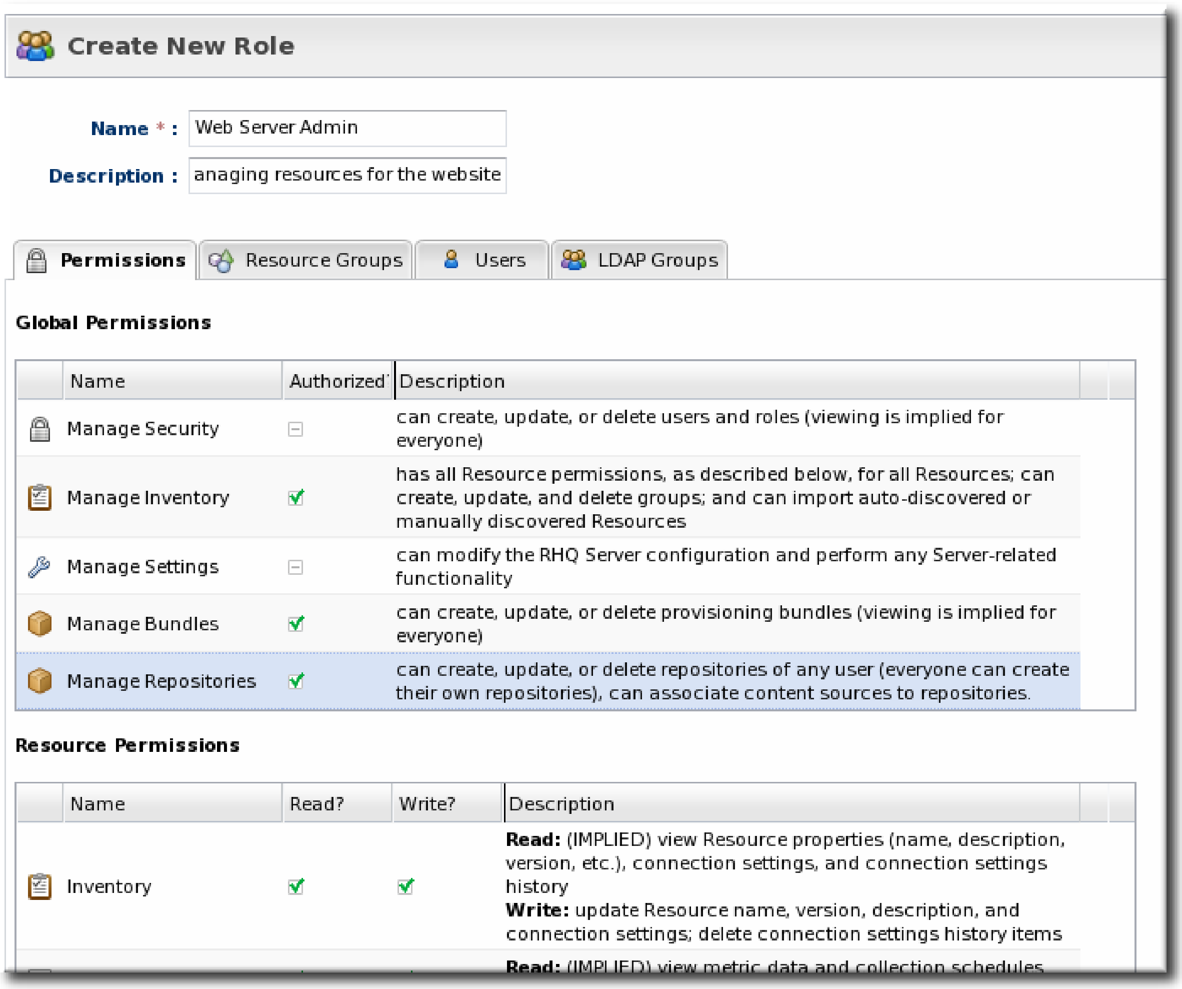
4. The list of current roles comes up in the main task window. Click the **New** button at the bottom of the list.

5. Give the role a descriptive name. This makes it easier to manage permissions across roles.

6. Set the access rights for the role in the **`Permissions`**. There are two categories of permissions:

   - *Global permissions* grant permissions to areas of the JBoss ON server and configuration.

   - *Resource permissions* grant permissions for managing resources.


Create New Role

Name * : Web Server Admin

Description : anaging resources for the website

Permissions | Resource Groups | Users | LDAP Groups

Global Permissions

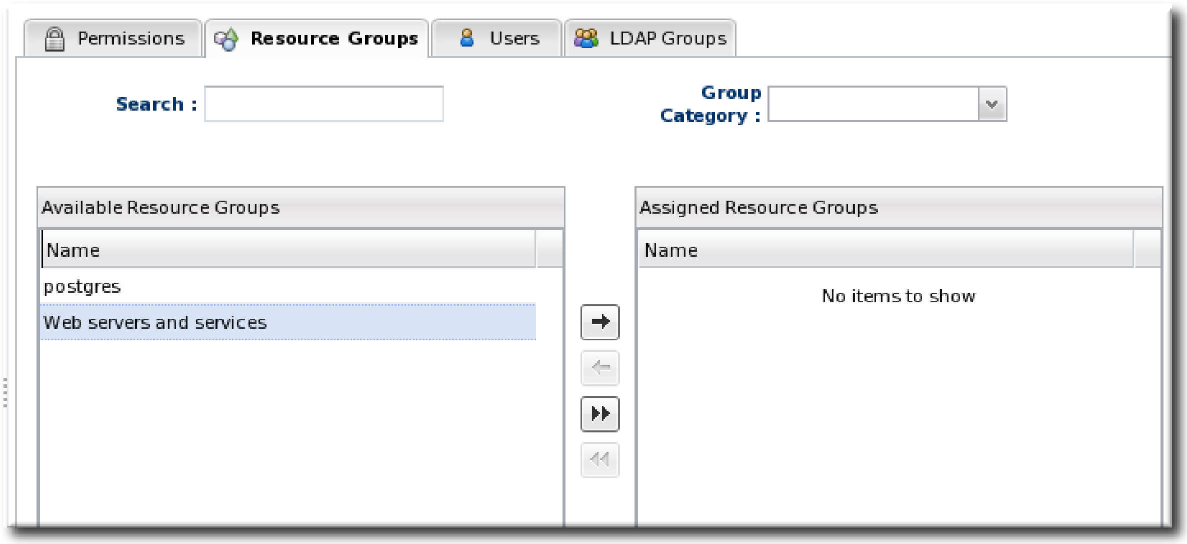
Name	Authorized	Description
Manage Security	⊟	can create, update, or delete users and roles (viewing is implied for everyone)
Manage Inventory	✓	has all Resource permissions, as described below, for all Resources; can create, update, and delete groups; and can import auto-discovered or manually discovered Resources
Manage Settings	⊟	can modify the RHQ Server configuration and perform any Server-related functionality
Manage Bundles	✓	can create, update, or delete provisioning bundles (viewing is implied for everyone)
Manage Repositories	✓	can create, update, or delete repositories of any user (everyone can create their own repositories), can associate content sources to repositories.

Resource Permissions

Name	Read?	Write?	Description
Inventory	✓	✓	**Read:** (IMPLIED) view Resource properties (name, description, version, etc.), connection settings, and connection settings history **Write:** update Resource name, version, description, and connection settings; delete connection settings history items
			Read: (IMPLIED) view metric data and collection schedules

The specific access permissions are described in Table 9.1, “JBoss ON Access Control Definitions”.

7. Select the **Resource Groups** tab to assign groups to the role.


Permissions | Resource Groups | Users | LDAP Groups

Search :

Group Category :

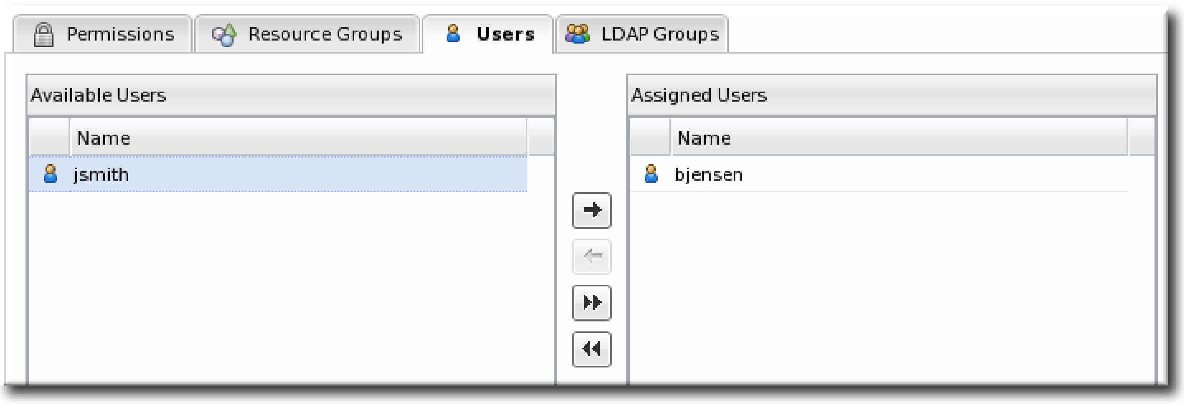
Available Resource Groups
Name
postgres
Web servers and services

Assigned Resource Groups
Name
No items to show

Move the required groups from the **Available Resource Groups** area on the left to the **Assigned Resource Groups** on the right as required.

8. At the bottom, click the **Save** button.

9. Select the **Users** tab to assign users to the role.



Move the required user from the **`Available Users`** area on the left, to the **`Assigned Users`** on the right as required.

10. Click the arrow in the upper right to close the create window.

9.3. *EXTENDED EXAMPLE:* READ-ONLY ACCESS FOR BUSINESS USERS

JBoss ON distinguishes between read permissions and write permissions. Neither read nor write access is implicit to any object or functional area in JBoss ON, which gives administrators some flexibility in where and what access is granted.

The Setup

Example Co. needs some of its management team to be able to read and access JBoss ON data to track infrastructure performance and maintenance, define incident response procedures, and plan equipment upgrades. While these business users need to view JBoss ON information, they should not be able to edit any of the configuration, which is handled by the IT and development departments.

Tim the IT Guy plans to create a special business managers role that will allow users to "see but not touch" the resource configuration.

The Plan

Tim the IT Guy first defines *what* actions the business users need to perform, and they need to be able to see everything:

- View resources in the inventory and histories for adding and deleting resources.
- View monitoring information, including measurements and events.
- View alerts.
- View content and bundles and any deployments to resources.
- View configuration drift.
- View all resource histories for configuration and operations.
- View user details to get information for auditing actions.

All of the global permissions relate to creating entries and managing configuration in JBoss ON and the inventory — which none of the business managers need to be able to do. There is one exception: the view users permission, which allows regular users to see the details of other users. That is necessary, because many actions such as running operations and changing resource configuration list what user initiated the action. Being able to view user information is required for adequate auditing of infrastructure changes.

The default selection for roles is for all resource-level permissions to grant read access to users, with the exception of configuration rights, which have no access. Tim the IT Guy decides to grant read-only access to the configuration so that managers can check the configuration history, which could be useful for policy planning. The group has read-only access to all resources and to items like reports.

The Results

Business users are given access to all of the information they need, without being able to change any configuration or inventory accidentally.

9.4. *EXTENDED EXAMPLE:* VIEW ALL RESOURCES, EDIT SOME RESOURCES

For security and for management practices, access should be limited to what is necessary for a given user. Security starts at restricting access as the default and then explicitly granting defined rights (read, modify, delete) to specific resources.

While it is possible to create a set of access control rules which define everything that is required for a user (or set of users), these rules in practice are cumbersome, complicated, and easily outdated or inaccurate. Access controls are a definition of a relationship; frequently, the different relationships that a user has are too complex to be defined in a single role.

The effect of roles is cumulative. The *total* level of access for a user is the sum of all of the roles they belong to — access to resources in all of the specified resource groups, the permissions granted to those roles.

Because the effects are cumulative, it is most effective to design access controls in a layered way, using small roles that define a very specific set of access and then adding those roles to users incrementally.

This layered approach allows administrators to define and effectively maintain complex relationships, both at a macro-level along personnel and infrastructure divisions and at a micro-level with different relationships to and between different resources.

The Setup

Example Corp. has three major groups associated with its IT infrastructure: development, QE, and production. Each group requires information from the other teams to help maintain their configuration, manage performance settings, and roll out new applications, but they should only be able to manage their own systems.

Within each group, there are two separate application management tasks: updating and deploying new versions of the application and managing system configuration for optimal performance.

The Plan

Tim the IT Guy first defines the different relationships that need to be expressed within the access controls:

- Everyone needs to be able to view performance data for all stacks within the infrastructure.
- The individual divisions need write access to their own systems.

- At least some administrators within each group require the ability to update system configuration.
- At least some administrators within each group require the ability to create and deploy bundles to manage applications within their own groups.

The first step is to define the required groups, with the minimum required resources in each group. In a simplified structure:

- A mixed group which contains all of the resources within each given stack environment. The stacks include platforms, Postgres databases, EAP servers, web contexts, and other resources used to manage the production environment.

  This results in three groups: Dev Stack, QE Stack, and Production Stack.

- An "all stacks" nested group which includes all three stack groups.

  This group is not for all resources — it only includes the stack groups, excluding JBoss ON-related resources and other managed resources not relevant to those stacks.

- Since these environments include application development, each organization also requires its own bundle group to maintain deployments.
- There has to be a mechnism to promote bundles between environments. Tim the IT Guy creates "Promote Bundles" group where bundles can be added when they are ready to be moved into a different environment.

Each of the resource and bundle groups could be broken down, if different users within the same division require different levels of access to individual resources.

Then, different users require different kinds of access. Tim the IT Guy then maps the different permissions that are required:

- View-only rights to all resources, including configuration view-only rights
- Edit rights to resources within the stacks for monitoring, alerts, drift, operations, and inventory
- Edit rights to resources within the stacks for configuration
- View bundle rights within the stacks
- Create and deploy bundle rights within the stacks

There are essentially three types of users within each functional group:

- Regular users
- Administrators which manage resource configuration
- Administrators which can create (promote) bundles between groups

The roles are going to corrspond to each resource and bundle group with permissions set for the minimal requirements for that group (local view and edit). For the administrators, the additional permissions — configuration and promoting content — are going to be added by creating additional roles with only the additional permissions.

For a regular user, he adds roles for all resources and bundles and resources within his stacks.

```
      Dev Stack
     Bundle Group
          |
      Role BG1
          |
          V
    Regular Joe
     ^         ^
     |         |
  Role RG1  Role RG2
     |         |
"All Stack"   Dev Stack
 Resource     Resource
 Group        Group
```

For the "All Stack" role, he adds read-only permission for all inventory areas *and* configuration.

```
     ^
     |
     Role RG1 <------Permissions
     |                     |
"All Stack"              View.alerts
 Resource                View.inventory
 Group                   View.measurements
                         View.etc...
                         View.configuration
```

For the stack group, Tim the IT Guy sets permissions to edit every functional area (measurements, alerts, operations, events) except for configuration, which is reserved for administrator users. The resource group also includes the permission to *deploy* bundles to the resources in the group. The ability to deploy bundles is separate from the ability to create bundles.

```
     ^
     |
     Role RG2 <------Permissions
     |                     |
 Dev Stack              Edit.alerts
 Resource               Edit.inventory
 Group                  Edit.measurements
                        Edit.etc...
                        Deploy.bundles
```

Last, he adds permissions to view and create bundles within the regular user's bundle group.

```
Dev Stack
     Bundle Group
          |
      Role BG1 <-----Permissions
          |                |
   V             View.bundles
                        Create.bundles
```

In this configuration, an administrator has two extra tasks: managing resource configuration within their stacks and promoting bundles to the next work group. The administrator is added to all of the roles of the regular user, plus additional roles for the additional tasks.

Tim the IT Guy creates one additional role to define the configuration permission. This grants configuration editing only to resource groups the administrator can see (the ones in his work groups).

```
"Regular Joe" roles
        |
        V
   Group Lead <------Role RG3
                         |
                    Permissions
                         |
                 Edit.configuration
```

Last, each administrator is added to the role for each bundle group. The bundle group roles only grant access to the bundle group to create content, not to view, deploy, or delete it. This allows administrators to promote content between work groups, but not to deploy it or affect resource configuration.

```
                Dev Stack          Permission:
             Bundle Group          Create.Bundles
                       \          /
                        \        /
                         Role BG1
                             |
                             V
        Role BG2 ---->    Group Lead    <---- Role BG3
       /      \                               /     \
      /        \                             /       \
 QE Stack      Permission:        Prod Stack         Permission:
Bundle Group   Create.Bundles   Bundle Group         Create.Bundles
```

The Result

Users within each group are granted access to view whatever performance and configuration information they need, but they can only make changes to resources within their specified group. Only administrators within each group can make configuration changes.

Application deployment is limited within each functional area (development, QE, and production). Specific administrators which are allowed to create content in other groups, but not to deploy it.

CHAPTER 10. INTEGRATING LDAP SERVICES FOR AUTHENTICATION AND AUTHORIZATION

JBoss ON can incorporate LDAP directories to help manage users, authentication, and membership in roles. This simplifies user management in JBoss ON and also leverages existing organizational configuration (user accounts, groups, passwords, and account lockout policies) so that JBoss ON mirrors other infrastructure configuration.

IMPORTANT

If LDAP is used for user account management, then the LDAP directory should be the authoritative source for creating and managing user accounts. Otherwise, there can be inconsistencies in role memberships, account settings, or other user account conflict. See Section 10.2.2, “Issues Related to Using LDAP for a User Store” .

10.1. SUPPORTED DIRECTORY SERVICES

JBoss ON supports major directory servers for user authentication and group authorization:

- Red Hat Directory Server 8.1, 8.2, and 9.0
- Microsoft Active Directory 2003 and 2008

10.2. LDAP FOR USER AUTHENTICATION

10.2.1. About LDAP Authentication and Account Creation

By default, JBoss ON stores authentication information in its internal database. JBoss ON can also use an external LDAP repository to store this user information. With LDAP authentication, the JBoss ON server sends all login requests to the LDAP directory to process.

First, the JBoss ON server searches the LDAP directory for a matching username, and then it attempts to log into (bind to) the LDAP server using the given username and password. If the bind attempt is successful, then the user is successfully authenticated to the JBoss ON server.

After the JBoss ON server is configured to use LDAP for authentication, all login attempts are authenticated against the LDAP server.

WARNING

When the JBoss ON server is reconfigured to use LDAP for authentication, the LDAP information isn't validated yet. Any errors with the LDAP authentication configuration won't show up until a user attempts to log into the UI.

NOTE

The LDAP directory can't create JBoss ON users automatically. However, using LDAP for authentication allows new users to register themselves to JBoss ON. A new user can authenticate to JBoss ON as long as they have an LDAP account. At their first login attempt, they're redirected to a registration page which records the additional JBoss ON user information.

The JBoss ON server constructs the LDAP entry name to look for based on the JBoss ON username and information about the LDAP directory, like the parent distinguished name in the directory tree and the naming attribute used for user entries; from there, it dynamically constructs a search filter every time someone logs into JBoss ON. Custom attributes can be added to the LDAP schema, such as **`JONUser=true`**, which can make it easier and more precise to locate entries.

The LDAP directory *only* verifies the login credentials. The LDAP server doesn't store any other JBoss ON user data, and it doesn't create, delete, or edit entries in JBoss ON. Likewise, JBoss ON doesn't create, delete, or edit entries in the LDAP directory. The only attributes in the LDAP database that relate to JBoss ON user accounts are the username and password. Other settings in the JBoss ON user entry are stored in the JBoss ON internal database (like the user's first name and surname, email address, and role assignments).

NOTE

The LDAP directory is used only to check the login credentials — it doesn't store any other information about the JBoss ON users, including role assignments, and it cannot create a JBoss ON user. The JBoss ON server also cannot *create* LDAP users, so the LDAP user has to be created separately.

Because the LDAP directory doesn't store other attributes related to JBoss ON, it can't store a user certificate. This means that JBoss ON cannot use an LDAP directory for certificate-based authentication.

10.2.2. Issues Related to Using LDAP for a User Store

Integrating LDAP directories introduces another area where users can be created and managed and where the membership of roles can be changed. On the one hand, this can make managing users much easier, especially by allowing existing users to register themselves seamlessly and by automatically updating role membership. However, because users can still be created in JBoss ON and added manually to JBoss ON, user and role management can become messy.

The first problem is simply determining which datastore to use to authenticate users. Even after LDAP authentication is enabled, JBoss ON still checks credentials against its own user store — and it checks its own database first. This means that a user can authenticate to JBoss ON without that request being sent to the LDAP database. All of the security features of the LDAP directory — particularly password policies and account inactivation — are lost because that is not the primary authentication mechanism.

Second, using two resources for user accounts introduces the problem of erroneously mapping JBoss ON and LDAP user accounts, creating duplicate entries, or allowing ghost entries. For example, John Smith is added as a user manually to JBoss ON and also has an LDAP user account. First, he has two duplicate, separately-managed user entries. Then, John Smith goes to a different division, and his LDAP entry is automatically deleted, but his JBoss ON user account remains because JBoss ON user accounts and LDAP user accounts aren't linked. He could still log into JBoss ON. Having duplicate user accounts can introduce other problems if there are accounts with identical *names*. For example, Jane Smith logs into JBoss ON with her JBoss ON user account (**`jsmith`**) and password, but is improperly assigned the JBoss ON role membership of LDAP user John Smith (LDAP UID `jsmith`) because her JBoss ON user ID was the same as his LDAP user ID, and her account was incorrectly mapped to his LDAP account and, therefore, his LDAP group membership.

Trying to maintain user accounts in both locations also impacts roles, at least in an administrative way. LDAP groups are added as members to the role, and then the group members are listed as user members for the role. However, the list of users assigned to the role does *not* show where those users came from. This means that the user list can be a mix of LDAP group members and JBoss ON members who were manually added to the list. Ultimately, it becomes difficult to add or remove users because it's not clear where the role users are coming from. Limiting role membership to LDAP groups simplifies maintenance; the roles are automatically updated when users are added or deleted to the groups on the LDAP side and those changes are synchronized over to the JBoss ON role dynamically.

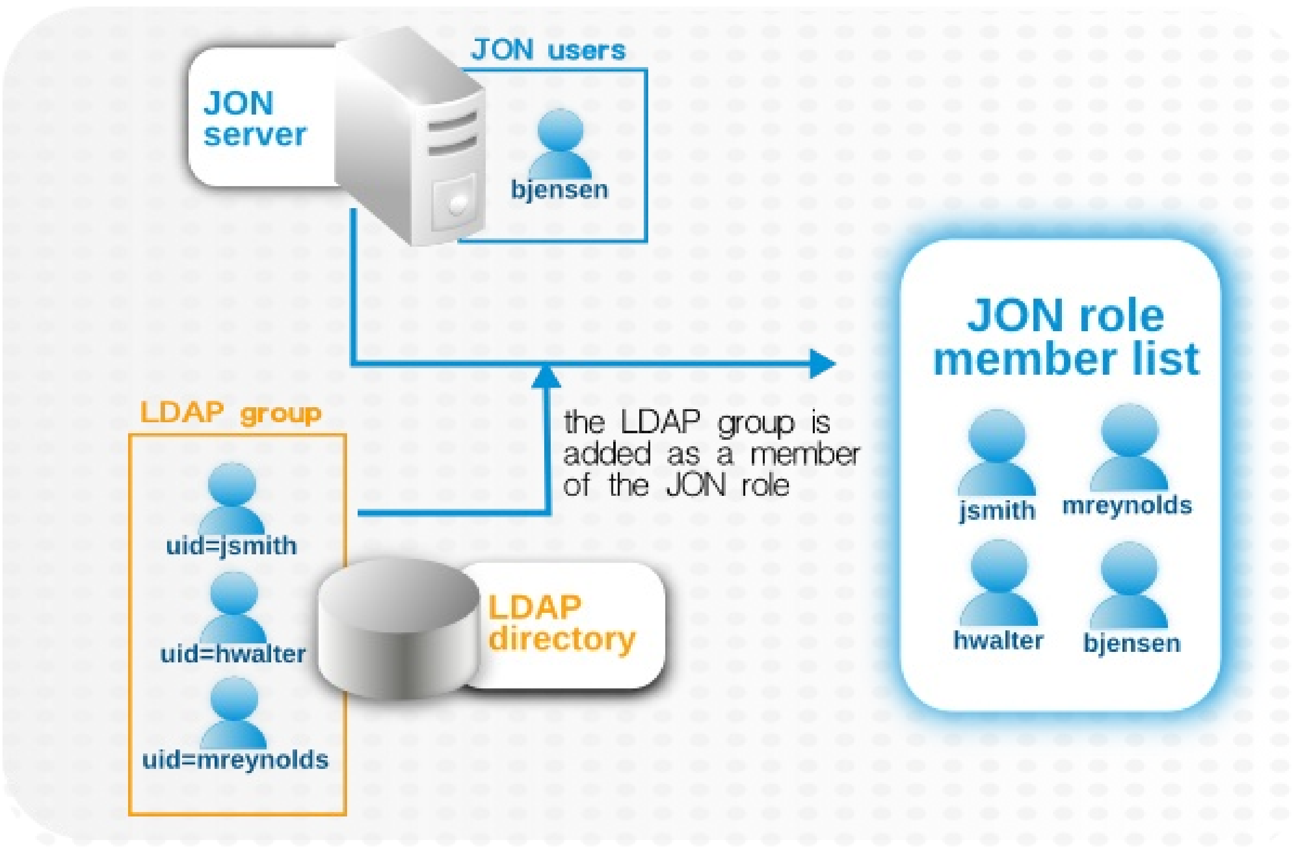


Figure 10.1. LDAP Groups, JBoss ON Roles, and Role Members

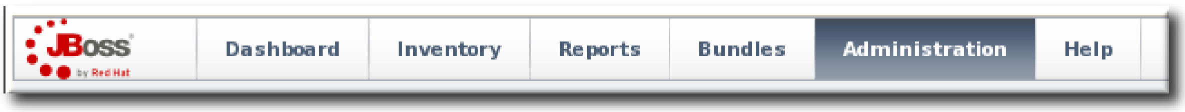
What all of this means is that there are three things to consider when using LDAP services for authentication or authorization:

- Only create regular user accounts in one place. If LDAP should be used for authentication, then only add or delete user accounts in the LDAP directory.
- Ideally, limit JBoss ON user accounts to special, administrative users and rely on the LDAP directory for regular accounts.
- Try to design roles around LDAP groups, meaning that JBoss ON user accounts in those roles should be limited to admin accounts or avoided altogether.

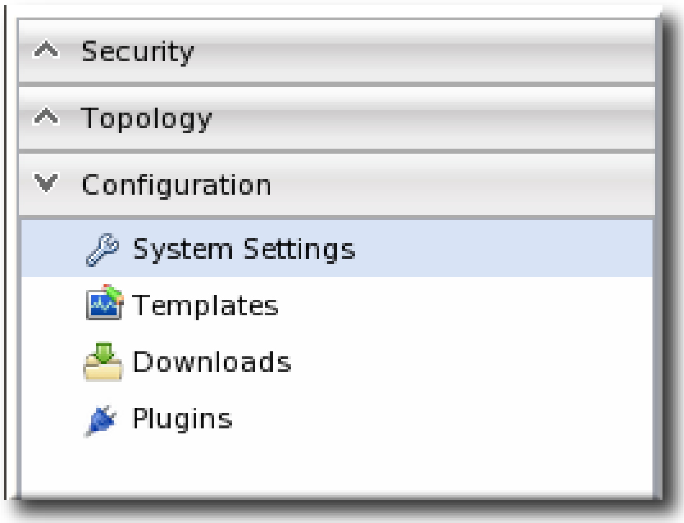
10.2.3. Configuring LDAP User Authentication

Authentication is the process of verifying the identity of an entity who attempts to access a server. JBoss ON uses simple authentication, meaning it uses simple username-password pairs to verify identity.

1. In the top menu, click the **`Administration`** tab.

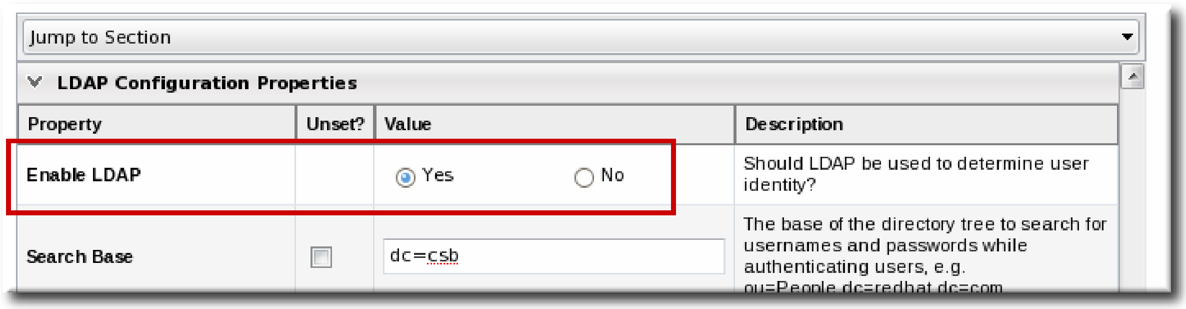


2. In the **Configuration** menu table on the left, select the **System Settings** item.



3. Jump to the **`LDAP Configuration Properties`** area.

4. Check the **`Use LDAP Authentication`** checkbox so that JBoss ON will use the LDAP user directory as its identity store.

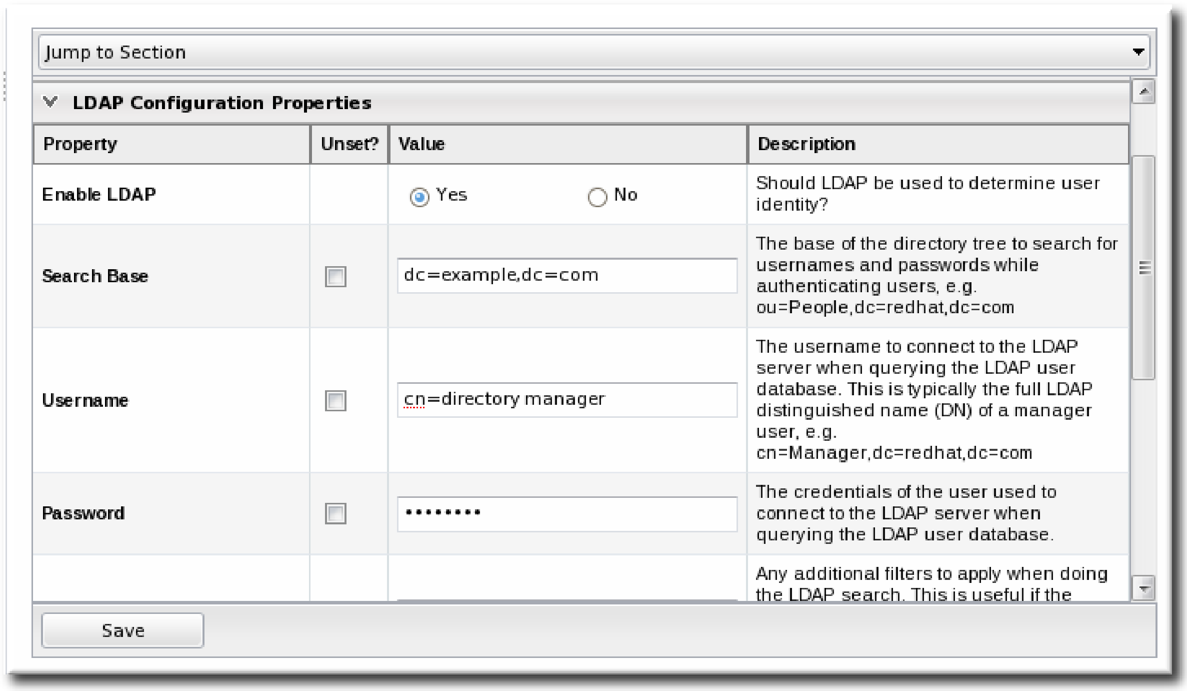


Jump to Section

LDAP Configuration Properties

Property	Unset?	Value	Description
Enable LDAP		[✓ Yes] No	Should LDAP be used to determine user identity?
Search Base	☐	dc=csb	The base of the directory tree to search for usernames and passwords while authenticating users, e.g. ou=People,dc=redhat,dc=com

5. Configure the connection settings to connect to the specific LDAP directory.



Property	Unset?	Value	Description
Enable LDAP		[✓ Yes] No	Should LDAP be used to determine user identity?
Search Base	☐	dc=example,dc=com	The base of the directory tree to search for usernames and passwords while authenticating users, e.g. ou=People,dc=redhat,dc=com
Username	☐	cn=directory manager	The username to connect to the LDAP server when querying the LDAP user database. This is typically the full LDAP distinguished name (DN) of a manager user, e.g. cn=Manager,dc=redhat,dc=com
Password	☐	•••••••••	The credentials of the user used to connect to the LDAP server when querying the LDAP user database.
			Any additional filters to apply when doing the LDAP search. This is useful if the

- Give the LDAP URL of the LDAP server. This has the format **ldap://***hostname[:port]*. For example:

  ```
  ldap://server.example.com:389
  ```

  By default, this connects to the localhost over port 389 (standard LDAP port) or 636 (secure LDAP port, if SSL is selected).

- To use a secure connection, check the **Use SSL** checkbox. When using SSL, make sure that the LDAP directory is actually running over SSL, and make sure that the connection URL points to the appropriate SSL port and protocol:

  ```
  ldaps://server.example.com:636
  ```

- Give the bind credentials to use to connect to the server. The username is the full LDAP distinguished name of the user as whom JBoss ON binds to the directory.

  NOTE

  The user *must* exist in the LDAP directory before configuring the LDAP settings in JBoss ON. Otherwise, login attempts to the JBoss ON server will fail.

  Also, make sure that the JBoss ON user has appropriate read and search access to the user and group subtrees in the LDAP directory.

6. Set the search parameters that JBoss ON uses when searching the LDAP directory for matching user entries.

   - The *search base* is the point in the directory tree where the server begins looking for entries. If this is used only for user authentication or if all JBoss ON-related entries are in the same subtree, then this can reference a specific subtree:

```
ou=user,ou=JON,dc=example,dc=com
```

If the users or groups are spread across the directory, then select the base DN:

```
dc=example,dc=com
```

- Optionally, set a search filter to use to search for a specific subset of entries. This can improve search performance and results, particularly when all JBoss ON-related entries share a common LDAP attribute, like a custom ***JonUser*** attribute. The filter can use wild cards (**`objectclass=*`**) or specific values (**`JonUser=true`**).
- Set the LDAP naming attribute; this is the element on the farthest left of the full distinguished name. For example, in **`uid=jsmith,ou=people,dc=example,dc=com`**, the far left element is **`uid=jsmith`**, and the naming attribute is ***uid***.

  The default naming attribute in Active Directory is ***cn***. In Red Hat Directory Server, the default naming attribute is ***uid***.

7. Save the LDAP settings.

NOTE

The **`Group Filter`** and **`Member Property`** fields aren't required for user authentication. They're used for configuring LDAP groups to be assigned to roles, as in Section 10.3.2, "Associating LDAP User Groups to Roles" .

10.3. ROLES AND LDAP USER GROUPS

10.3.1. About Group Authorization

Many LDAP directories already contain organizational groups with users who will need to access resources in JBoss ON. Configuring JBoss ON to connect to these directories allows JBoss ON to assign LDAP groups to roles and then pull in those member lists dynamically, so the roles are populated with pre-existing member lists. All of the LDAP users automatically inherit the permissions of that role.

In the role details page, these LDAP user groups are separated from the resource groups, so it's easy to distinguish which types of group are being added to the role.

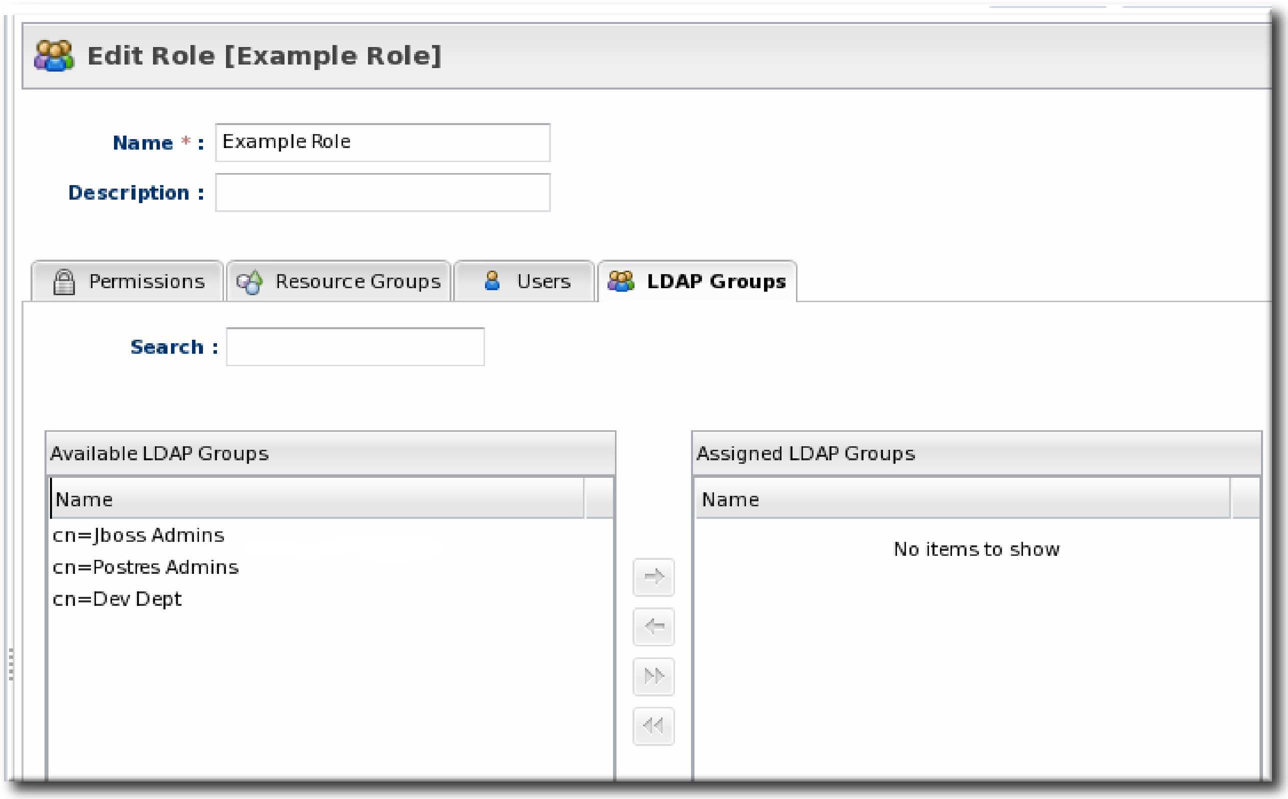


Figure 10.2. Groups Assigned to a Role

JBoss ON determines what LDAP groups a user belongs to with a simple search. Whenever a user logs into JBoss ON and an LDAP connection is configured, JBoss ON maps that JBoss ON username to a user entry in the LDAP directory server. The specific LDAP distinguished name (DN) for the user is used as part of a search to find matching member attributes in LDAP group entries. That is, the LDAP server can check the member lists in group entries to see what groups the person with that DN belongs to.

For LDAP groups to be added to roles, three things are required:

- An LDAP directory server connection has to be configured.
- There has to be an LDAP attribute given to search for *group entries*.

  For Active Directory, this is generally the **`group`** object class. For Red Hat Directory Server, this is generally **`groupOfUniqueNames`**. Other standard object classes are available, and it is also possible to use a custom, even JBoss ON-specific, object class.
- There has to be an LDAP attribute given to identify *members* in the group.

  Common member attributes are ***`member`*** and ***`uniqueMember`***.

JBoss ON constructs an LDAP search based on the group object class and member attribute in the server configuration, plus the DN of the user given when the user logs in.

```
(&(group_filter)(member_attribute=user_DN))
```

For example, this looks for the ***`member`*** attribute on an Active Directory group:

```
ldapsearch -h server.example.com -x -D
"cn=Administrator,cn=Users,dc=example,dc=com" -W -b "dc=example,dc=com" -x
'(&(objectclass=group)(member=CN=John Smith,CN=Users,DC=example,DC=com))'
```

Red Hat Directory Server uses the ***`uniqueMember`*** attribute on **`groupOfUniqueNames`** groups more commonly than ***`member`*** and ***`group`***. For example:

```
/usr/lib64/mozldap6/ldapsearch -D "cn=directory manager" -w secret -p 389
-h server.example.com -b "ou=People,dc=example,dc=com" -s sub "(&
(objectclass=groupOfUniqueNames)
(uniqueMember=uid=jsmith,ou=People,dc=example,dc=com))"
```

This search returns a list of all groups to which the user is a member. If any of these LDAP groups is assigned to a JBoss ON role, then that user is also automatically a member of that JBoss ON role.

NOTE

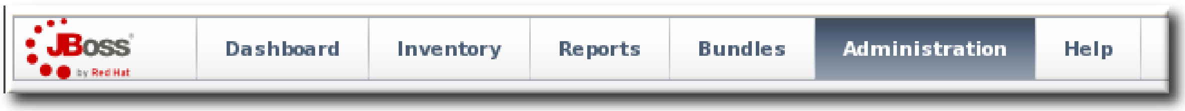
Using custom LDAP group object classes can allow you to be very specific about which groups to use for JBoss ON roles.

10.3.2. Associating LDAP User Groups to Roles

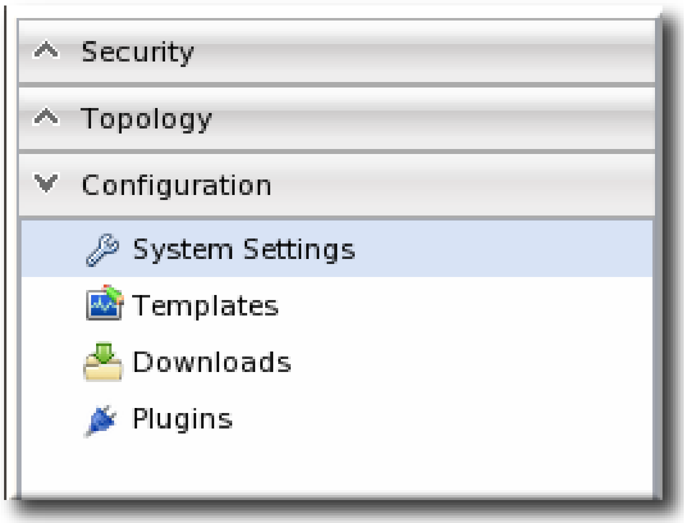
Section 10.2.3, "Configuring LDAP User Authentication" describes how an LDAP directory server can be used to authenticate users to JBoss ON. Any time a user attempts to log in, that request — with the username and password — is simply forwarded to the specified LDAP directory server to see if the credentials are correct.

Members of LDAP groups can be pulled in, automatically, as members of JBoss ON roles. The LDAP group is associated with a JBoss ON role and then the group members are *authorized* to do whatever the JBoss ON role is configured to allow. Any changes made in the LDAP group members are automatically reflected in JBoss ON, without having to edit the JBoss ON role.

1. In the top menu, click the **`Administration`** tab.



2. In the **Configuration** menu table on the left, select the **System Settings** item.



3. Jump to the **`LDAP Configuration Properties`** area.

4. Set up the LDAP connections, as described in Section 10.2.3, "Configuring LDAP User Authentication". It is not required that the LDAP directory be used as the identity store in order to configure LDAP authorization, but it is recommended.

5. Set the parameters to use for the server to use to search for LDAP groups and their members.

   The search filter that JBoss ON constructs looks like this:

   ```
   (&(group_filter)(member_attribute=user_DN))
   ```

   - The **`Group Search Filter`** field sets how to search for the group entry. This is usually done by specifying the type of group to search for through its object class:

     ```
     (objectclass=groupOfUniqueNames)
     ```

   - The **`Group Member Filter`** field gives the attribute that the specified group type uses to store member distinguished names. For example:

     ```
     uniqueMember
     ```

   The *user_DN* is dynamically supplied by JBoss ON when a user logs into the UI.

6. Save the LDAP settings.

10.4. *EXTENDED EXAMPLE:* MEMBEROF AND LDAP CONFIGURATION

The Setup

Authentication is the process of verifying someone's identity. *Authorization* is the process of determining what access permissions that identity has. Users are authorized to perform tasks based on the permissions granted to their role assignments.

All of the users and identities for Example Co. are stored in a backend Red Hat Directory Server LDAP database. To maintain a single, central user store, Tim the IT Guy wants to use existing LDAP users in JBoss ON and to determine user access to JBoss ON based on group membership, so, fundamentally, both authentication and authorization rules are determined by the LDAP configuration.

The Plan

There are two things to configure: how to identify users for authentication and how to organize users for authorization.

Groups are going to play a two-fold part in managing the LDAP configuration for JBoss ON for Example Co.:

- A single group to identify JBoss ON users in the LDAP directory
- Multiple, existing LDAP groups which are used to determine different levels of access to JBoss ON

The first thing that Tim the IT Guy determines is the way to identify users. As Section 10.2.3, "Configuring LDAP User Authentication" describes, JBoss ON identifies users to authenticate based on the results of an LDAP search, which uses a search base and optional search filter. The search filter specifies an *attribute*=*value* pair. One recommended method for identifying users is to create custom schema elements, like **JONUser**, which make it easy to search for matching users.

However, Tim the IT Guy has limited administrative access to the Red Hat Directory Server database. He has the ability to create groups and manage membership, but he cannot edit the schema. With no way to create an attribute that flags JBoss ON users, Tim the IT Guy has to use other configuration. Depending on the layout of the directory, he can use other kinds of configuration: views, manager, a class of service (CoS) virtual attribute, or group membership.

Using group membership is a good way to manage user assignments easily and dynamically while only having to manage a single entry (instead of individual group entries). In Directory Server, the ***memberOf*** attribute is automatically added to user entries to indicate a group that the user belongs to.

What Tim the IT Guy can do is set up a special group for all JBoss ON users, and then whatever users he likes. Because the Directory Server automatically adds and removes the ***memberOf*** attribute to user entries as members are added and removed to the group. Tim the IT Guy only has to use the ***memberOf*** attribute on those user accounts as the search filter for authentication.

```
dn: uid=jsmith,ou=people,dc=example,dc=com
uid: jsmith
cn: John Smith
... 8< ...
memberOf: cn=JON User Group,ou=groups,dc=example,dc=com
memberOf: cn=IT Administrators,ou=groups,dc=example,dc=com
```

The JBoss ON LDAP authentication search filter, then, would target the ***memberOf*** attribute for that specific JBoss ON group:

```
memberOf='cn=JON User Group,ou=groups,dc=example,dc=com'
```

Using groups for access control requires an entirely different set of group definitions, which do not have to be JBoss ON-specific. These groups relate to functional areas within Example Co., and Tim the IT Guy can map existing LDAP groups to JBoss ON roles. There are three relevant LDAP groups for Example Co. for managing JBoss ON:

- IT Administrators Group is mapped to a role with manage inventory permissions.
- IT Manager Group is mapped to a role with view (but no write) permissions for all of the resources and with view users permissions.

- Business Manager Group is mapped to a role with permissions to read all resource configuration, bundles, drift, measurements, operations, and alerts, but no write permissions.

The Results

Tim the IT Guy only has to create and manage one LDAP group, the JON Users Group, to set up all authentication and users for JBoss ON. He does not have to change the LDAP schema or even modify user entries directly.

For authorization, Tim the IT Guy designs JBoss ON roles around the functional groups already defined in the LDAP directory, in Example Co.'s organization, groups for IT admins, IT managers, and business managers and the level of access each requires.

As LDAP users authentication to the JBoss ON UI for the first time, they set up their own JBoss ON user details. After authenticating, they are automatically granted the appropriate level of access based on their LDAP group membership.

PART II. MANAGING RESOURCE CONFIGURATION

CHAPTER 11. EXECUTING RESOURCE OPERATIONS

11.1. OPERATIONS: AN INTRODUCTION

JBoss Operations Network provides a way to manage resources by scheduling and launching *operations*. Operations are basic management tasks. The available tasks differ for every different type of resource.

The *Resource Reference: Monitoring, Operation, and Configuration Options*contains a complete reference for all of the operations that can be scheduled for each resource type, as well as configurable parameters for the operations. Regardless of the type of operation or resource, the process for scheduling operations is similar for both resources and compatible groups in JBoss ON.

JBoss Operations Network allows administrators to manage resources by scheduling and launching *operations*. Operations manage resources by initiating or even scheduling some basic, specified tasks, such as restarting a server or running a script. Operations can be carried out on any resource in the inventory, and even on the JBoss ON agent themselves. The types of operations that are available for each resource depends on the type of resource being managed. For example, a JBoss AS server has different available operations than a cron service. The supported operations for a resource are defined by its agent plug-in, and the default operations are listed for each resource type in the *Resource Reference: Monitoring, Operation, and Configuration Options*.

11.1.1. A Summary of Operation Benefits

Operations provide a way to perform tasks in a consistent way, with a defined order both on resources and in task queuing, and in a way that can be tracked. Because operations are defined by plug-ins, they are extensible. The versatility of running specific tasks through JBoss ON provides several benefits to administrators:

- They allow additional parameters (depending on how the operation is defined in the plug-in), such as command arguments and environment variables.
- They validate any operation parameters, command-line arguments, or environment variables much as JBoss ON validates resource configuration changes.
- They can be run on group of resources as long as they are all of the same type.
- Operations can be ordered to run on group resources in a certain order.
- They can be run on a recurrently schedule or one specific time.
- Operations keep a history of both successes and failures, so that it is possible to audit the operations executed on a resource both for operations run for that specific resource and done on that resources as part of a group.

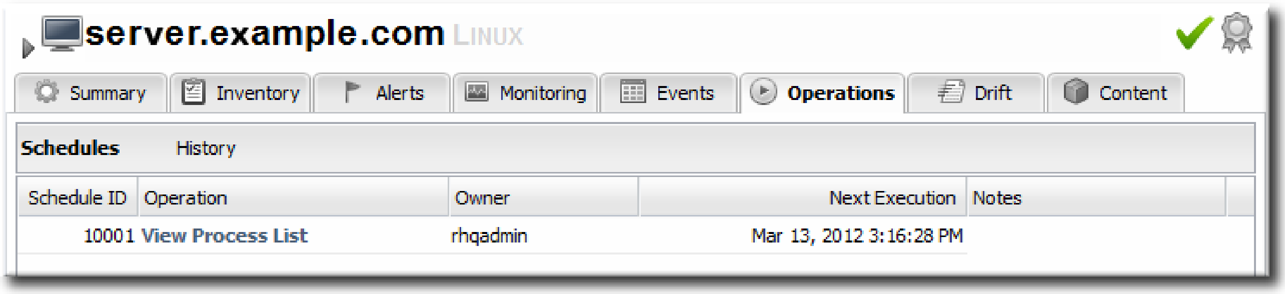
11.1.2. About Scheduling Operations

An operation schedule is the defined time when that operation can be run, including immediately.

There are two paths to schedule an operation:

- Using the calendar setting to select a time. There are three different ways to schedule an operation using the calendar: immediately, at a set point in the future, or on a recurring schedule. The recurring schedules can be indefinite or run within a specific time period.

- Using a cron expression. This is used almost exclusively for recurring jobs and can be used to set very complex execution schedules.

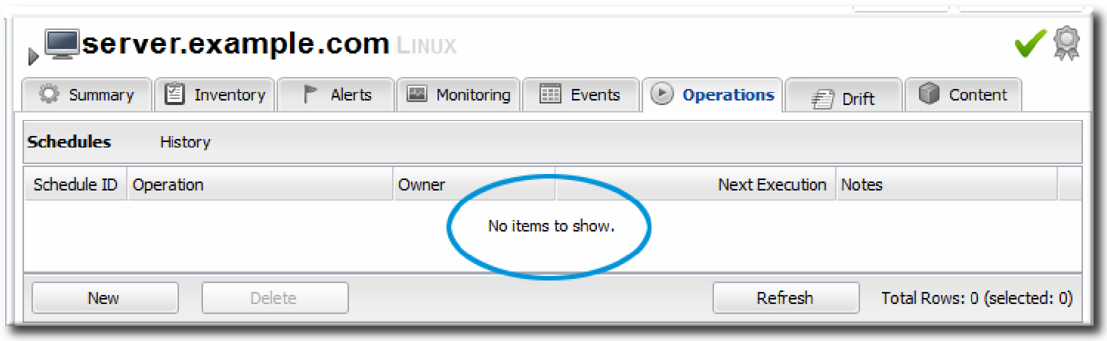


Schedule ID	Operation	Owner	Next Execution	Notes
10001	View Process List	rhqadmin	Mar 13, 2012 3:16:28 PM	

Figure 11.1. A Scheduled Operation

NOTE

The **Schedules** tab shows a list of scheduled operations, meaning operations which are configured but have not yet been run.[2]



No items to show.

Total Rows: 0 (selected: 0)

When an operation is scheduled, a new operation is added to the history record for the resource, and its state is set to in progress. A message is sent to the agent telling it to invoke a specific operation on a particular resource with the arguments that were specified when the schedule was created. The agent queues operations so that only one operation is executed on the resource at a time.

When an operation completes, its raw output is sent back to the agent's resource plug-in, which ultimately parses the output and then generates an appropriate response message. This response message is then sent to the server.

If one operation ever hangs on a resource, then it blocks any other operations from being initiated because only one operation can be run on a resource at a time. Using a timeout setting for the operation enables the agent to kill the hung operation and run other queued operations.

11.1.3. About Operation Histories

The operation history is essentially an audit trail for tasks performed on the resource through JBoss ON.

Each entry shows the name of the user who scheduled the operation, the time the operation was scheduled to run, its status, and any results. For any failures, there is an informational message which contains the error message returned.

Scheduled operations and recent operations are listed in portlets in the **Dashboard** and in JBoss ON reports. These operation lists include both resource and compatible group operations.

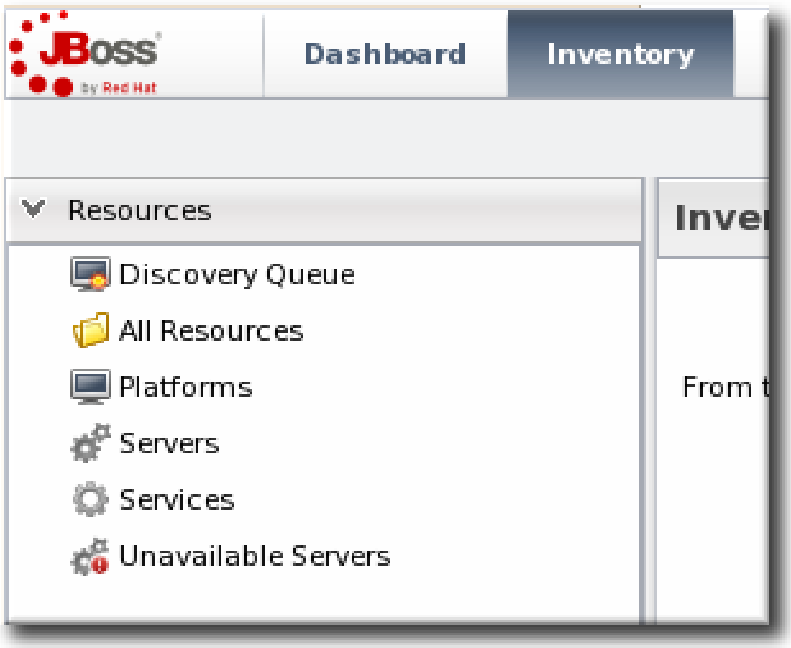
As with individual resource operations, group operations also record a history for the group. These operation histories are really a list of the same operation being run on all of the resource in the group, so the operation history shows first a summary of the scheduled group operation and then the execution details for each individual resource.

11.2. MANAGING OPERATIONS: PROCEDURES

11.2.1. Scheduling Operations

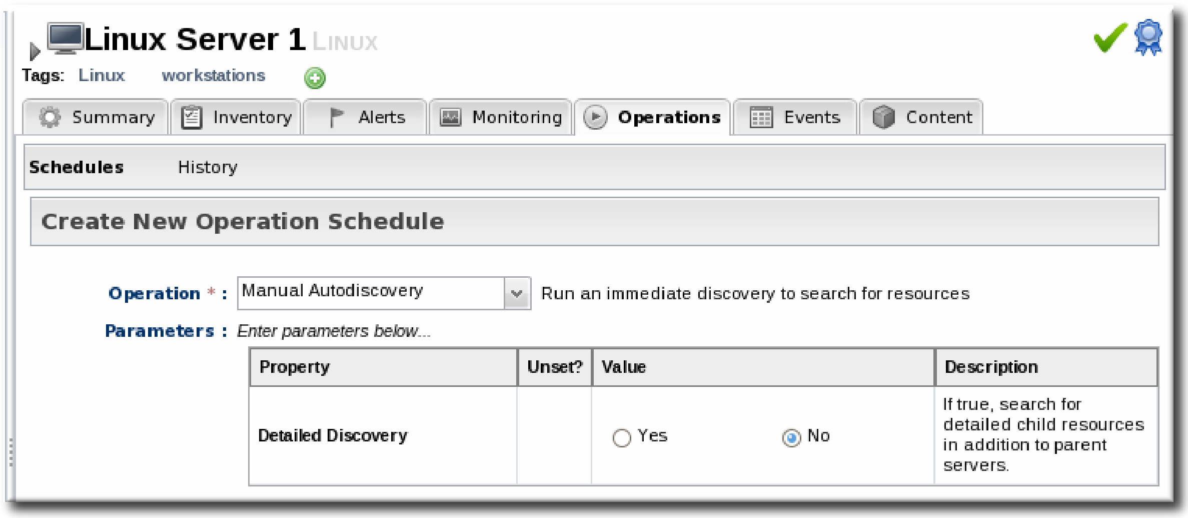
1. Click the **Inventory** tab in the top menu.

2. Select the resource type in the **Resources** menu table on the left, and then browse or search for the resource.



3. Click the **Operations** tab.

4. In the **Schedules** tab, click the **New** button.



Create New Operation Schedule

Operation * : Manual Autodiscovery — Run an immediate discovery to search for resources

Parameters : Enter parameters below...

Property	Unset?	Value	Description
Detailed Discovery		Yes [✓ No]	If true, search for detailed child resources in addition to parent servers.

The types of operations that are available vary, depending on the specific type of resource.

NOTE

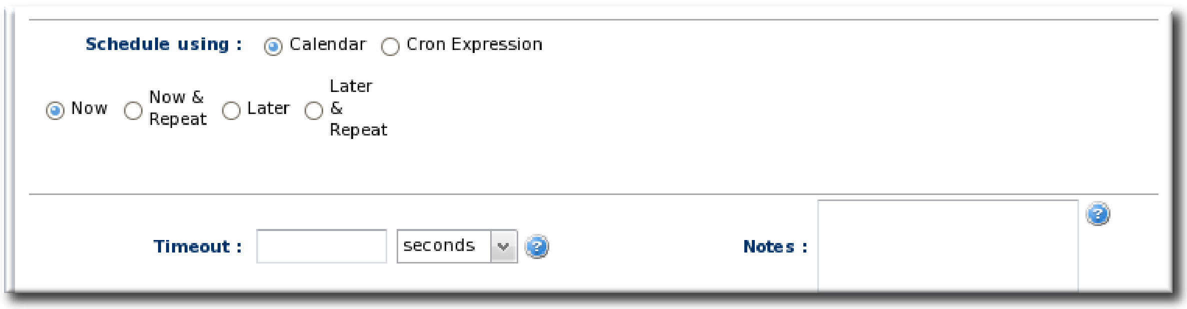
The **Schedules** tab shows a list of scheduled operations, meaning operations which are configured but have not yet been run. If there are no scheduled operations, then the tab has a description that reads *No items to show*. That does not mean that there are no operations available for the resource; it only means that no operations have been scheduled.

5. Fill in all of the required information about the operation, such as port numbers, file locations, or command arguments.

6. In the **Schedule** area, set when to run the operation.

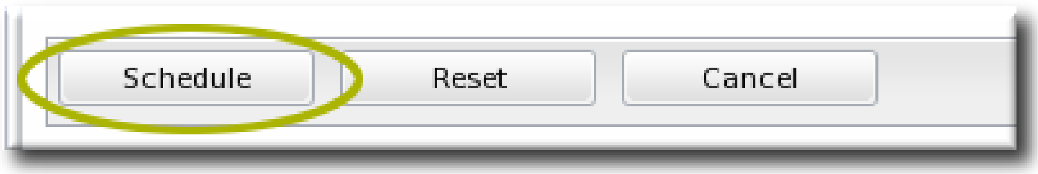
   When using the **Calendar**, the operation can run immediately, at a specified time, or on a repeatable schedule, as selected from the date widget.

   The **Cron Expression** is used for recurring jobs, based on a **cron** job. These expressions have the format *min hour day-of-month month day-of-week*, with the potential values of *0-59 0-23 1-31 1-12 1-7*; using an asterisk means that any value can be set.



7. Set other rules for the operations, like a timeout period and notes on the operation itself.

8. Click the **Schedule** button to set up the operation.



If the operation is scheduled to run immediately, the results are available in the **History** subtab as the operation is in progress and then completes. If it was scheduled on a later date or with a recurring schedule, then the operation is listed in the **Schedules** subtab.

11.2.2. Viewing the Operation History

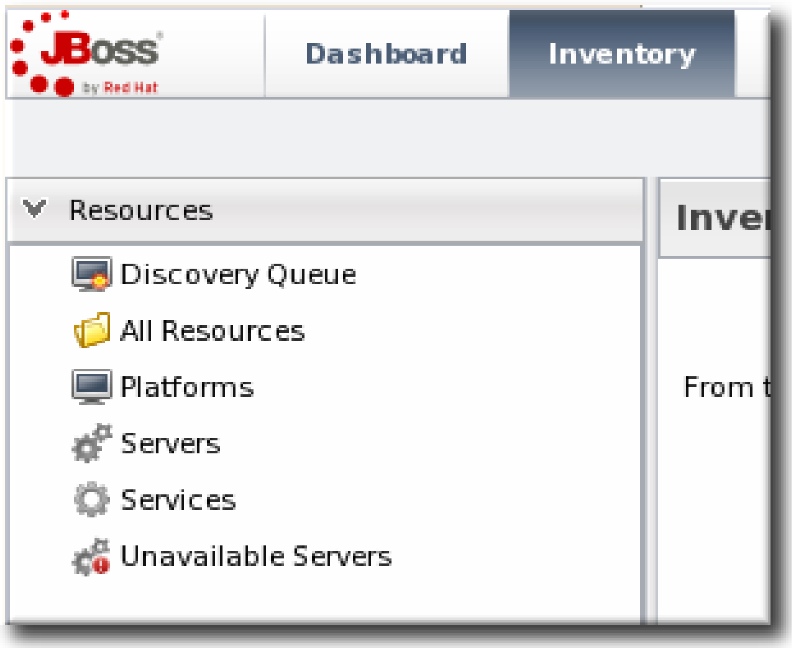
NOTE

A user may have the write to schedule and edit an operation, but that does not mean that the user has the right to delete an item from the operation history.

Deleting elements in the history requires the manage inventory permission.

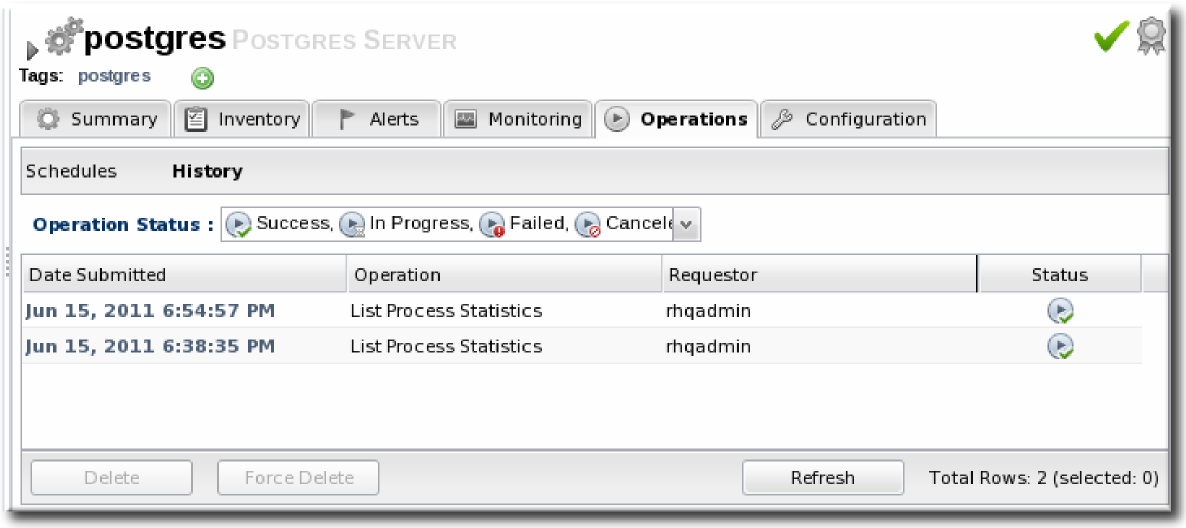
1. Click the **Inventory** tab in the top menu.

2. Select the resource type in the **Resources** menu table on the left, and then browse or search for the resource.



3. Click the **Operations** tab.

4. Click the **History** subtab.



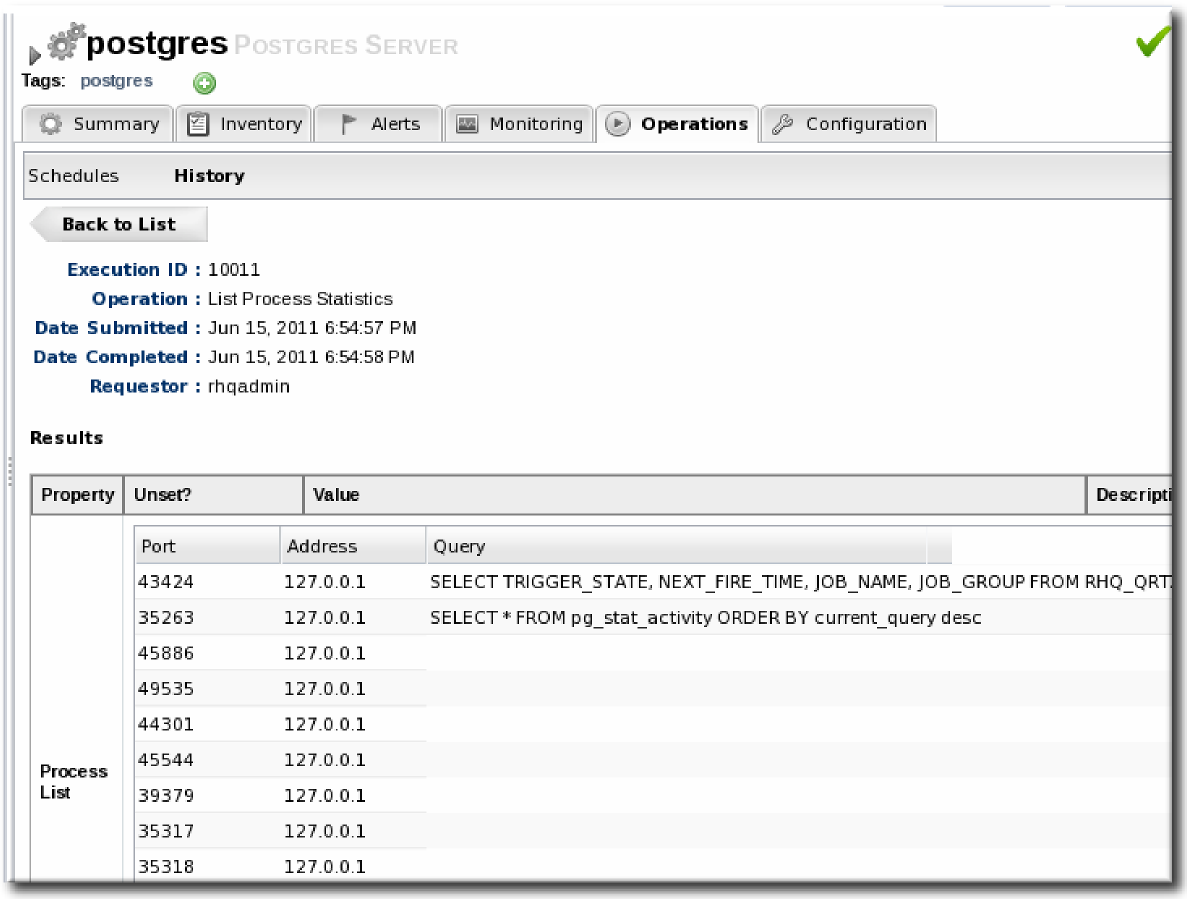
Operation Status : Success, In Progress, Failed, Cancele

Date Submitted	Operation	Requestor	Status
Jun 15, 2011 6:54:57 PM	List Process Statistics	rhqadmin	✓
Jun 15, 2011 6:38:35 PM	List Process Statistics	rhqadmin	✓

Total Rows: 2 (selected: 0)

Every completed or in progress operation is listed, with an icon indicating its current status.

5. Click the name of the operation to view further details. The history details page shows the start and end times of the operation, the stdout output of the operation or other operation messages, as well as the name of the user who scheduled the operation.



Execution ID : 10011
Operation : List Process Statistics
Date Submitted : Jun 15, 2011 6:54:57 PM
Date Completed : Jun 15, 2011 6:54:58 PM
Requestor : rhqadmin

Results

Property	Unset?	Value	Descripti

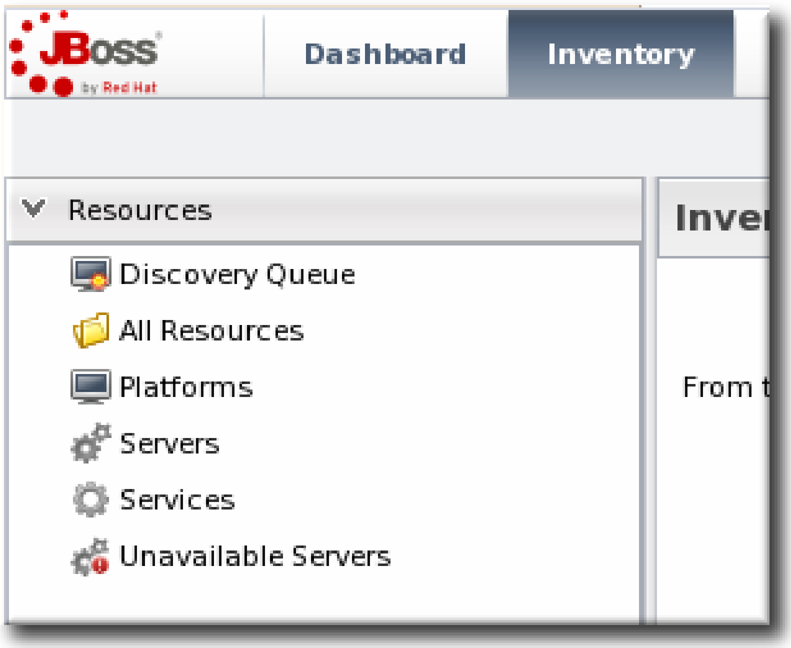
Port	Address	Query
43424	127.0.0.1	SELECT TRIGGER_STATE, NEXT_FIRE_TIME, JOB_NAME, JOB_GROUP FROM RHQ_QRTZ
35263	127.0.0.1	SELECT * FROM pg_stat_activity ORDER BY current_query desc
45886	127.0.0.1	
49535	127.0.0.1	
44301	127.0.0.1	
45544	127.0.0.1	
39379	127.0.0.1	
35317	127.0.0.1	
35318	127.0.0.1	

Process List

11.2.3. Canceling Pending Operations

1. Click the **`Inventory`** tab in the top menu.

2. Select the resource type in the **`Resources`** menu table on the left, and then browse or search for the resource.



3. Click the **`Operations`** tab.

4. In the **Schedules** tab, click the line of the operation to cancel.

5. Click **Delete**, and confirm the action.

NOTE

Once the agent has started an operation it cannot be canceled. If the user attempts to cancel an operation currently in progress, the request will be ignored.

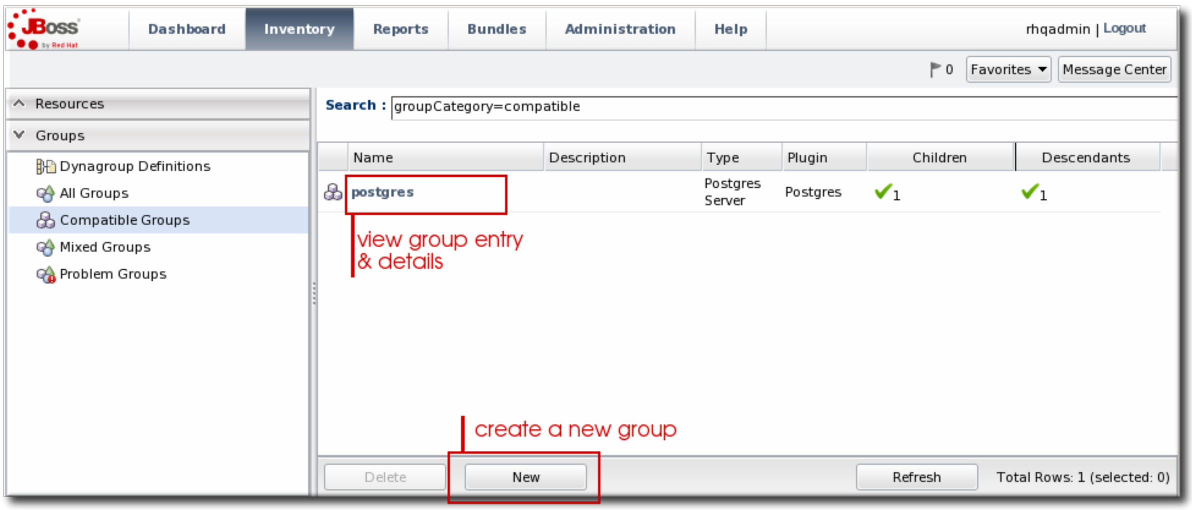
11.2.4. Ordering Group Operations

Group operations can be scheduled. This is useful when operations need to be performed in a particular order.

NOTE

This procedure assumes groups are already set up.

1. In the **Inventory** tab in the top menu, select the **Compatible Groups** item in the **Groups** menu on the left.



Search : groupCategory=compatible

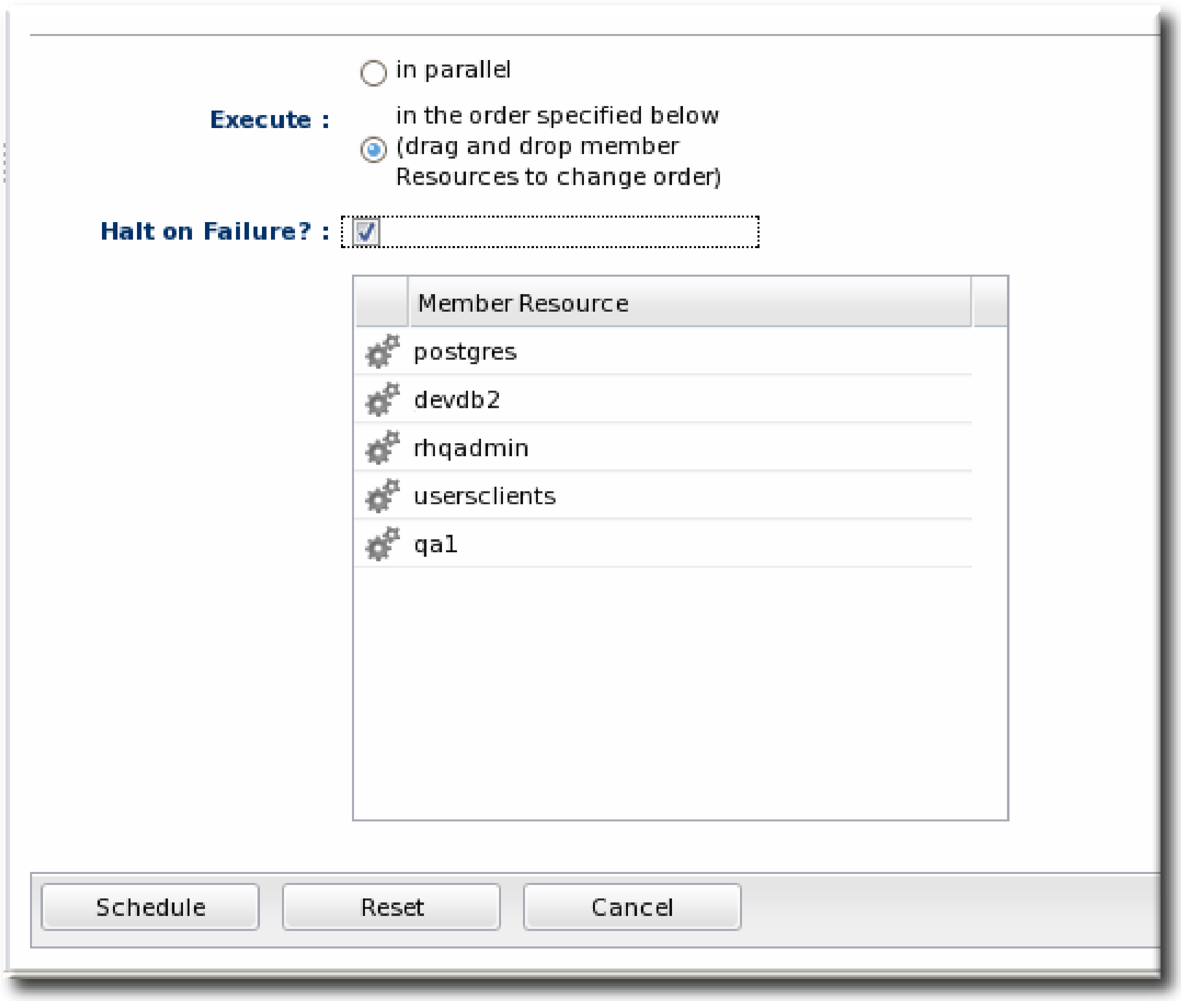
Name	Description	Type	Plugin	Children	Descendants
postgres		Postgres Server	Postgres	✓1	✓1

2. Click the name of the group to run the operation on.

3. Configure the operation, as in Section 11.2.1, “Scheduling Operations”.

4. In the **Resource Operation Order** area, set the operation to execute on all resources at the same time (in parallel) or in a specified order. If the operation must occur in resources in a certain order, then all of the group members are listed in the **Member Resources** box, and they can be rearranged by dragging and dropping them into the desired order.

   Optionally, select the **Halt on failure** checkbox to stop the group queue for the operation if it fails on any resource.



Execute : ○ in parallel [✓ ● in the order specified below (drag and drop member Resources to change order)]

Halt on Failure? : ☑

Member Resource
postgres
devdb2
rhqadmin
usersclients
qa1

11.2.5. Running Scripts as Operations for JBoss Servers

JBoss ON auto-discovers resource scripts when the resource is discovered. Scripts can be managed just like any other resource to perform operations. There are three types of scripts that JBoss ON discovers, depending on the operating system:

- `.bat` for Windows
- `.sh` for Unix and Linux
- `.pl` scripts for Unix and Linux

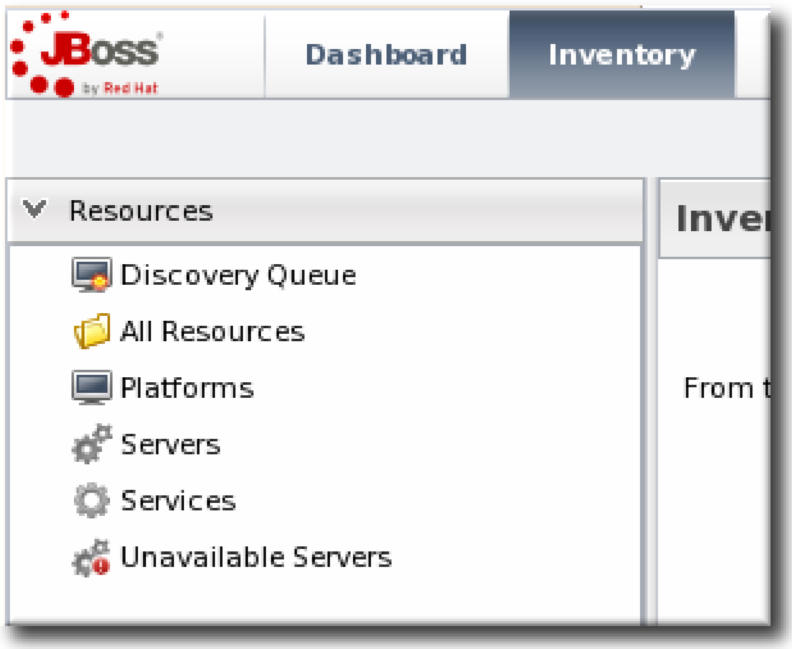
NOTE

Scripts on Linux and Unix systems need to have the x-bit set to be detected.

Connection properties and environment variables can be added to a script.

To execute a script as an operation:

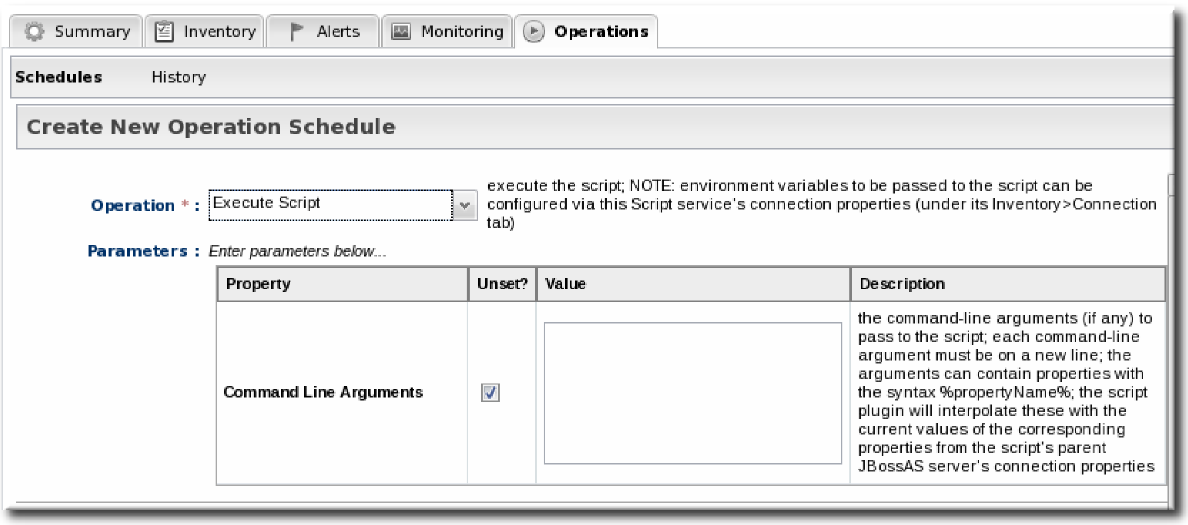
1. Click the **`Inventory`** tab in the top menu.
2. Select the resource type in the **`Resources`** menu table on the left, and then browse or search for the resource.



3. Click the **Operations** tab.

4. In the **Schedules** tab, click the **New** button.

5. Select **Execute script** as the operation type from the **Operation** drop-down menu.



Create New Operation Schedule

Operation * : Execute Script

execute the script; NOTE: environment variables to be passed to the script can be configured via this Script service's connection properties (under its Inventory>Connection tab)

Parameters : *Enter parameters below...*

Property	Unset?	Value	Description
Command Line Arguments	☑		the command-line arguments (if any) to pass to the script; each command-line argument must be on a new line; the arguments can contain properties with the syntax %propertyName%; the script plugin will interpolate these with the current values of the corresponding properties from the script's parent JBossAS server's connection properties

NOTE

The **Execute script** option is only available for JBoss AS and JBoss AS 5 resources, by default, and only if a script is available to execute.

6. Enter any command-line arguments in the **Parameters** text box.

   Each new argument has the format *name=value;* and is added on a new line. If the variable's value contains properties with the syntax **%propertyName%**, then JBoss ON interprets the value as the current values of the corresponding properties from the script's parent resource's connection properties.

7. Finish configuring the operation, as in Section 11.2.1, “Scheduling Operations”.

11.2.6. Setting an Operation Timeout Default

Only one operation can run on a resource at one time. An optional timeout setting prevents an operation from hanging indefinitely and blocking other operations from running. A global default timeout can be set in the JBoss ON server configuration to prevent operations from being blocked on a resource, even if a timeout period isn't set on a specific operation.

NOTE

This server setting is a fallback value. Operation plug-ins can define their own timeouts in the plug-in descriptor or individual operations can specify a timeout. Both of those settings override the server default.

1. Open the **`rhq-server.properties`** file.

   ```
   vim serverRoot/jon-server-3.2.GA1/bin/rhq-server.properties
   ```

2. Change or add the value of the ***`rhq.server.operation-timeout`*** parameter to the amount of time, in seconds, for the server to wait before an operation times out.

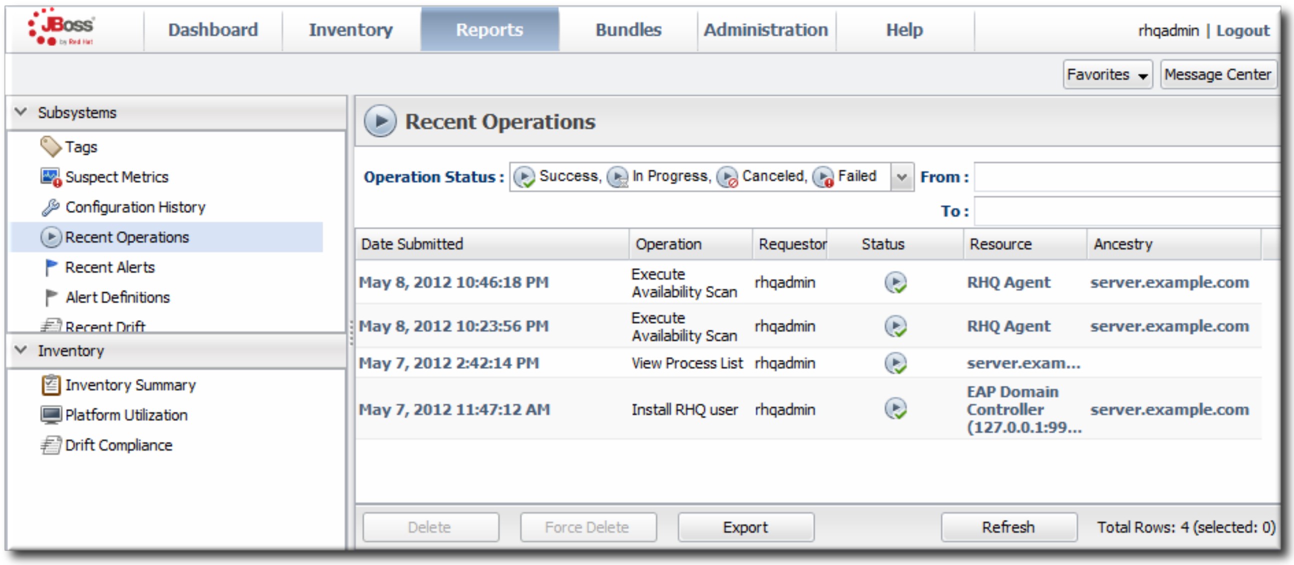
   ```
   rhq.server.operation-timeout=60
   ```

11.2.7. Operation History Report

Every resource tracks its own individual operations history, as in Section 11.2.2, “Viewing the Operation History”.

JBoss ON also keeps a master list of all operations, for all resources. This is displayed in the **`Operation History Report`**, in the **`Reports`** tab.

As with the resource-level operation history, the Operation History shows the operation name, the date it was submitted, and its status. Because all resources are listed, the **`Operation History Report`** also shows the resource name and its parent (and grandparent) to help disambiguate on which resource the operation ran.



Recent Operations

Operation Status : Success, In Progress, Canceled, Failed From :

To :

Date Submitted	Operation	Requestor	Status	Resource	Ancestry
May 8, 2012 10:46:18 PM	Execute Availability Scan	rhqadmin	✓	RHQ Agent	server.example.com
May 8, 2012 10:23:56 PM	Execute Availability Scan	rhqadmin	✓	RHQ Agent	server.example.com
May 7, 2012 2:42:14 PM	View Process List	rhqadmin	✓	server.exam...	
May 7, 2012 11:47:12 AM	Install RHQ user	rhqadmin	✓	EAP Domain Controller (127.0.0.1:99...	server.example.com

Total Rows: 4 (selected: 0)

Figure 11.2. Operation History Report

The **`Operation History Report`** can be filtered (not just sorted) by two criteria: the operation status and the date or date range that the operation was submitted.

NOTE

Reports can be exported to CSV, which can be used for office systems or further data manipulation.

Only the information displayed for the report is exported. If the **`Operation History Report`** is filtered by date or status, only the matching operations are included in the report.

To export a report, simply click the **`Export`** button. The report will automatically be downloaded as **`configurationHistory.csv`**.

[2] If there are no scheduled operations, then the tab has a description that reads *No items to show*. That does not mean that there are no operations available for the resource; it only means that no operations have been scheduled.

CHAPTER 12. SUMMARY: USING JBOSS ON TO MAKE CHANGES IN RESOURCE CONFIGURATION

One of the most basic parts of managing your applications, servers, and services is the simple ability to change their configuration.

JBoss Operations Network allows you to view the current configuration for many resource types directly in the JBoss ON UI, without having to access the platform's filesystem directly. Even more, JBoss ON allows you to edit the configuration directly for a single resource or for an entire group of compatible resources.

JBoss ON has three key ways that administrators can manage resource configuration:

- *Directly edit resource configuration.* JBoss ON can edit the configuration files of a variety of different managed resources through the JBoss ON UI.
- *Audit and revert resource configuration changes.*For the specific configuration files that JBoss ON manages for supported resources, you can view individual changes to the configuration properties and revert them to any previous version.
- *Define and monitor configuration drift.*System configuration is a much more holistic entity than specific configuration properties in specific configuration files. Multiple files for an application or even an entire platform work together to create an optimum configuration. *Drift* is the (natural and inevitable) deviation from that optimal configuration. Drift management allows you to define what the baseline, desired configuration is and then tracks all changes from that baseline.

This section has a very general overview of these three ways of managing resource configuration. More detailed descriptions and procedures are in the subsequent sections.

12.1. EASY, STRUCTURED CONFIGURATION

Basic configuration files use simple key-value pairs to define information.

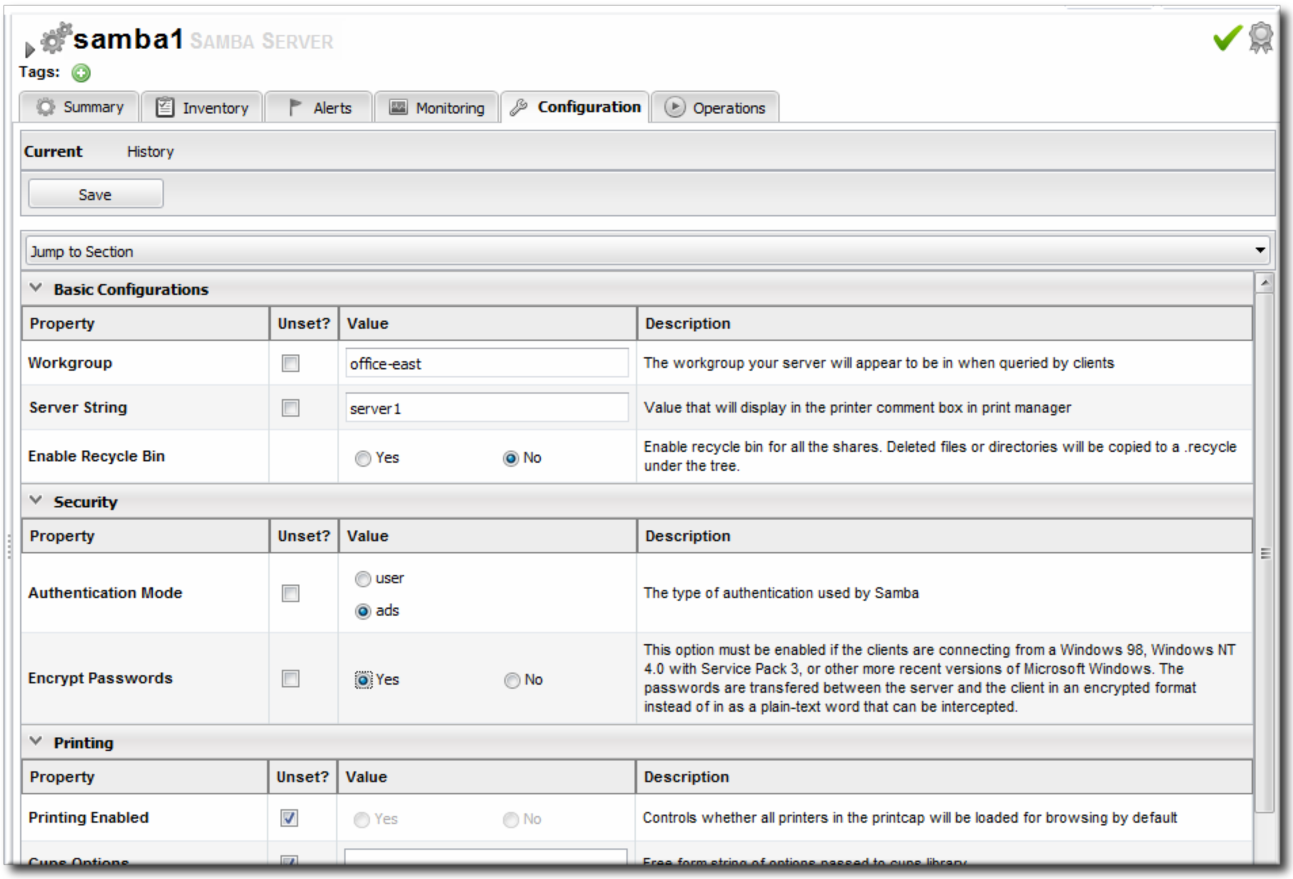
```
key1 = value1
key2 value2
```

These are *simple properties*, representing strings, numbers, or booleans — any type of information where there is one value per key.

JBoss ON also supports resource configuration using *complex properties*, which may be a list of values or a map of values (essentially a table of lists).

```
<default-configuration>
    <ci:list-property name="my-list">
        <c:simple-property name="element" type="string"/>
        <ci:values>
           <ci:simple-value value="a"/>
           <ci:simple-value value="b"/>
           <ci:simple-value value="c"/>
        </ci:values>
    </ci:list-property>
</default-configuration>
```

JBoss ON parses the configuration files — both simple properties and complex properties — and then renders a structured, easy-to-follow form in the JBoss ON GUI. Simple properties are displayed with fields or radio buttons as appropriate, while complex properties are displayed with menus or other selection options.



Basic Configurations

Property	Unset?	Value	Description
Workgroup	☐	office-east	The workgroup your server will appear to be in when queried by clients
Server String	☐	server1	Value that will display in the printer comment box in print manager
Enable Recycle Bin		Yes [✓ No]	Enable recycle bin for all the shares. Deleted files or directories will be copied to a .recycle under the tree.

Security

Property	Unset?	Value	Description
Authentication Mode	☐	user [✓ ads]	The type of authentication used by Samba
Encrypt Passwords	☐	[✓ Yes] No	This option must be enabled if the clients are connecting from a Windows 98, Windows NT 4.0 with Service Pack 3, or other more recent versions of Microsoft Windows. The passwords are transfered between the server and the client in an encrypted format instead of in as a plain-text word that can be intercepted.

Printing

Property	Unset?	Value	Description
Printing Enabled	✓	Yes No	Controls whether all printers in the printcap will be loaded for browsing by default
Cups Options	✓		Free form string of options passed to cups library

Figure 12.1. Configuration Form for a Samba Server

The structured configuration form makes it easy for you to view the current configuration quickly.

The structured form also makes it possible for JBoss ON to validate that the configuration properties have valid formats before saving changes.

NOTE

JBoss ON only validates that the given value matches the required format for that property. It does not validate that the value given is reasonable or allowed for that resource property.

Performing configuration changes in JBoss ON has major benefits for IT administrators:

- There is instant validation on the format of properties that are set through the UI.
- Audit trails for all configuration changes can be viewed in the resource history for both external and JBoss ON-initiated configuration changes.
- Configuration changes can be reverted to a previous stable state if an error occurs.
- Configuration changes can be made to groups of resource of the same type, so multiple resources (even on different machines) can be changed simultaneously.

- Alerts can be used in conjunction with configuration changes, either simply to send automatic announcements of any configuration changes or to initiate operations or scripts on related resources as configuration changes are made.

- Access control rules are in effect for configuration changes, so JBoss ON users can be prevented from viewing or initiating changes on certain resources.

12.2. IDENTIFYING WHAT CONFIGURATION PROPERTIES CAN BE CHANGED

JBoss ON supports configuration change for an extensive array of resources – including hosts and sudoers files, Samba servers, Postfix servers, databases, web app contexts, cron tabs, web servers, and scripts.

Any resource which supports configuration changes through JBoss ON has a **`Configuration`** tab on its resource page.

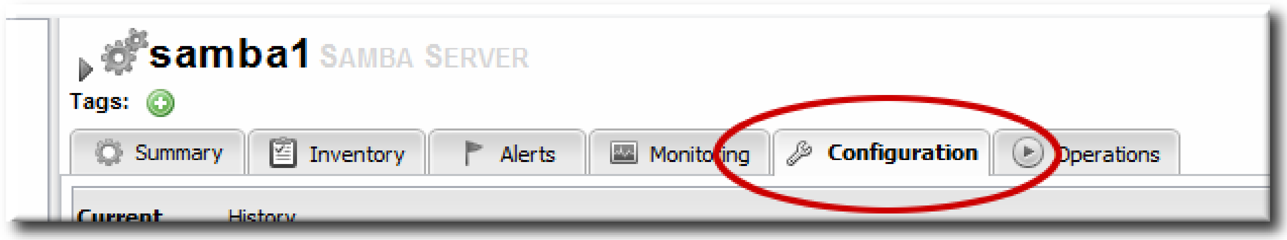


Figure 12.2. Configuration Tab

For a complete reference of the configuration properties for each resource, see the *Resource Reference: Monitoring, Operation, and Configuration Options*.

12.3. AUDITING AND REVERTING RESOURCE CONFIGURATION CHANGES

Tracking for configuration changes is a crucial part of systems administration. It's important for maintenance, for performance, and for incident recovery – particularly when it is possible to revert change or correlate changes to incidents.

Every time a change is made to the resource configuration, whether through JBoss ON GUI or on the resource itself, the change is detected by JBoss ON and logged with a revision number. When a change is made outside of JBoss ON, the change is simply noted. When the change is made through the JBoss ON UI, the timestamp and the name of the user who made the change are both recorded.

Every change is recorded in a history, and the different changesets can be viewed and compared to one another. One change can be selected and the resource configuration can be rolled back to that selected change.

Tracking the configuration history and reverting changes is covered more in Section 14.1, “Tracking and Comparing Configuration Changes” and Section 14.2, “Reverting Configuration Changes”.

12.4. TRACKING CONFIGURATION DRIFT

Much of the JBoss ON configuration management is designed around *implementing* changes for resources by editing configuration files or updating files and packages. But another aspect of managing configuration is *detecting changes*.

IT administrators must invest a significant amount of time planning the optimum configuration for systems in every type of environment, from production systems to internal resources. This ideal configuration includes file settings, software versions, and system settings. Resource configuration is going to change naturally over time, but administrators need to be able to track those changes to make sure that no unplanned or undesirable changes impact the resource. Defining a baseline configuration and tracking changes helps systems remain resilient during both maintenance and failures.

The unplanned changes that occur to a resource's configuration is called *drift*, as the configuration moves away from the designed baseline. Drift is common because of frequent software and hardware updates, particularly in a colocation facility or using virtual machines.

Production, staging, development, and recovery configurations are designed to have identical or near-identical configuration to maintain consistency. As the configurations within the different environments change, there emerges a *configuration gap*. Ultimately, this configuration gap can lead to disaster recovery failures or high availability failures because the configuration of the production system and the backup system are too different.

Drift monitoring provides a very general, freewheeling *content monitoring*. Rather than structured configuration management, drift monitoring tracks changes, any changes, in files — even binary files.

NOTE

The configuration history for a resource applies only to the supported configuration properties for that specific resource instance.

Drift management has a much more external view of configuration changes. Drift is associated with a resource — like a platform or a JBoss server — but it is not restricted to that resource or to set properties for that resource:

- Drift looks at whole files within a directory, including added and deleted files and binary files.
- Drift supports user-defined templates which can be applied to any resource which supports drift monitoring.
- Drift can keep a running history of changes where each changeset (*snapshot*) is compared against the previous set of changes. Alternatively, JBoss ON can compare each change against a defined baseline snapshot.

The drift definition that is essentially a profile that identifies a directory and files that should be monitored. Any time there is any change in that drift base directory or any of its subdirectories — file modifications, new files, or deleted files — the drift detection scan notices the change and records it.

Drift detection can be used by administrators to track scheduled changes, maintenance and updates, and server changes. There are a lot of common scenarios where administrators need to be aware that change has occurred (and even be able to identify the specific changes made), but that occur in areas outside normal JBoss ON configuration tracking:

- System password changes
- System ACL changes
- Database and server URL changes
- JBoss settings changes

- Changed JAR, WAR, and other binary files used by applications
- Script changes

NOTE

Drift is not bound or restricted to a resource managed by JBoss ON. You can create a drift definition for a platform and set it to monitor any file or directory on that platform, even if it is outside the JBoss ON inventory, as long as that directory is accessible to the system user that the JBoss ON agent runs as.

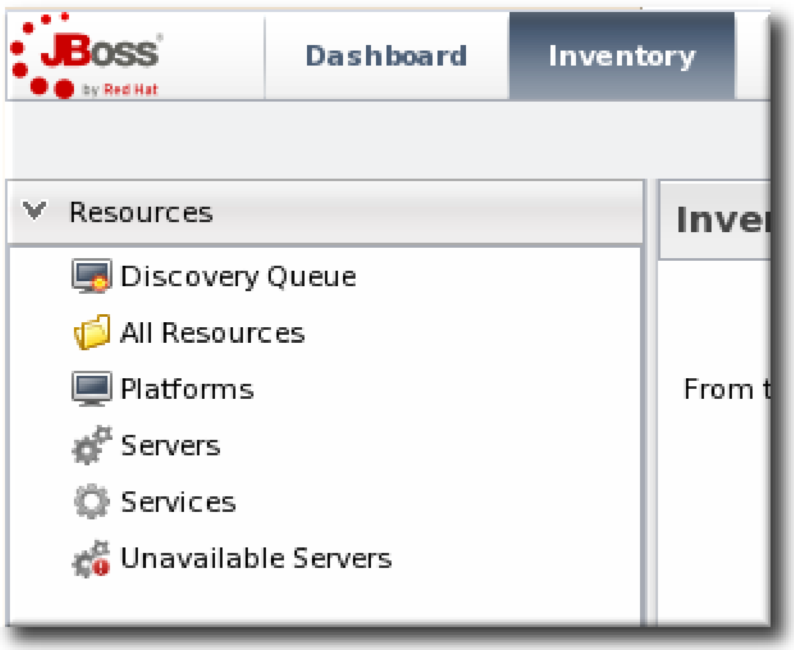
Managing configuration drift is described more in Chapter 15, *Managing Configuration Drift*.

CHAPTER 13. CHANGING THE CONFIGURATION FOR A RESOURCE

13.1. CHANGING THE CONFIGURATION ON A SINGLE RESOURCE

1. Click the **`Inventory`** tab in the top menu.

2. Select the resource type in the **`Resources`** menu table on the left, and then browse or search for the resource.



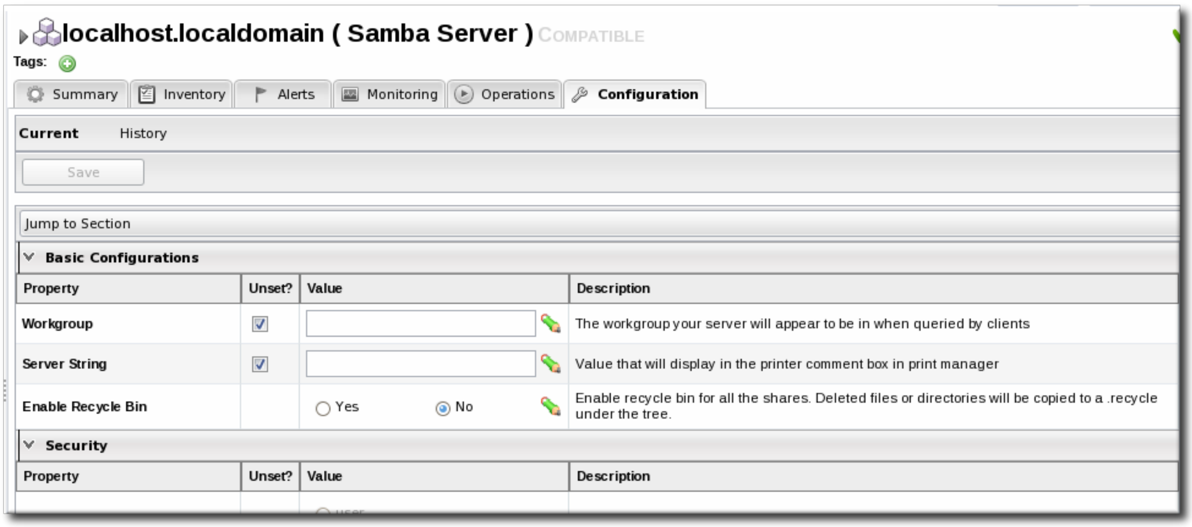
3. Open the **`Configuration`** tab for the resource.

4. Click the **`Current`** subtab.

5. To edit a field, make sure the **`Unset`** checkbox is *not* selected. The **`Unset`** checkbox means that JBoss ON won't submit any values for that resource and any values are taken from the resource itself.

   Then, make any changes to the configuration.

   The list of available configuration properties, and their descriptions, are listed for each resource type in the *Resource Reference: Monitoring, Operation, and Configuration Options*


localhost.localdomain (Samba Server) COMPATIBLE

Property	Unset?	Value	Description
Workgroup	✓		The workgroup your server will appear to be in when queried by clients
Server String	✓		Value that will display in the printer comment box in print manager
Enable Recycle Bin		Yes [✓ No]	Enable recycle bin for all the shares. Deleted files or directories will be copied to a .recycle under the tree.

6. Click the **Save** button at the top of the properties list.

13.2. CHANGING THE CONFIGURATION FOR A COMPATIBLE GROUP

Similar to other templating functions in JBoss ON, like alert templates, configuration changes can be made on a compatible or autogroup, so that all of the members of that group can be up updated simultaneously with the same settings.

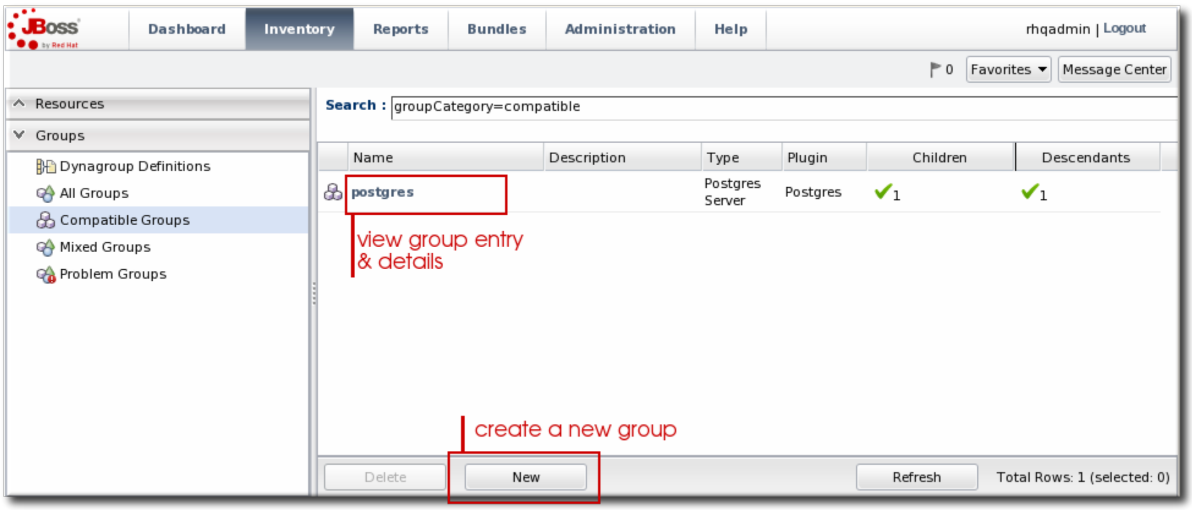
NOTE

To change the current configuration for a group, a few conditions must be true so that the current group configuration can be reliably calculated for the individual resource configurations:

- The group members must all be the same resource type.
- All group member resources must be available (**UP**).
- No other configuration update requests can be in progress for the group or any of its member resources.
- The current member configurations must be successfully retrieved from the agents.

The process for setting the configuration for a group is the same as setting it for an individual resource:

1. Click the **Inventory** tab in the top menu.
2. In the **Groups** box in the left menu, select the **Compatible Group** link.



Name	Description	Type	Plugin	Children	Descendants
postgres		Postgres Server	Postgres	✓1	✓1

3. Select the group to edit.

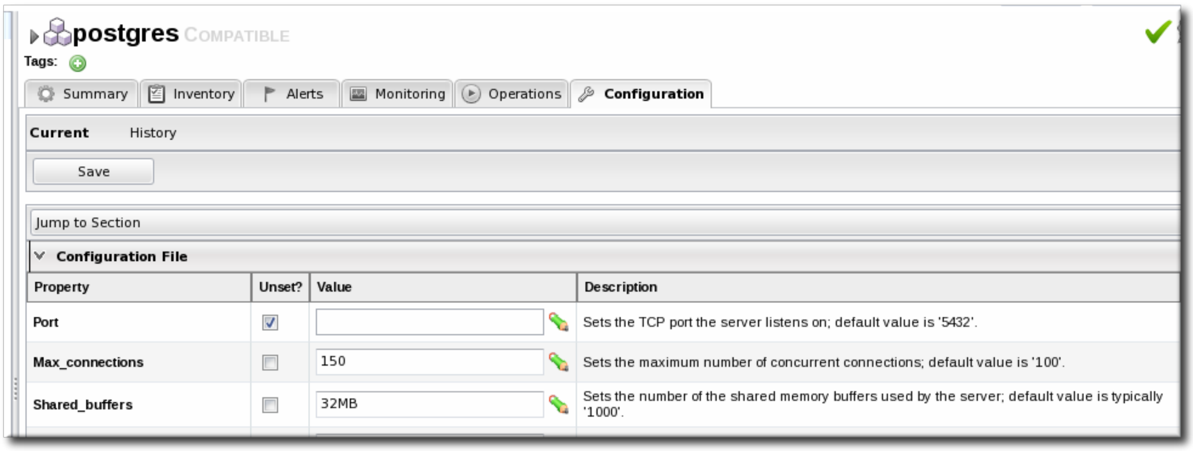
4. Open the **`Configuration`** tab.

5. Click the **`Current`** subtab.

6. To edit a field, make sure the **`Unset`** checkbox is *not* selected. The **`Unset`** checkbox means that JBoss ON won't submit any values for that resource and any values are taken from the resource itself.

   Then, make any changes to the configuration.

   The list of available configuration properties, and their descriptions, are listed for each resource type in the *Resource Reference: Monitoring, Operation, and Configuration Options*



Property	Unset?	Value	Description
Port	✓		Sets the TCP port the server listens on; default value is '5432'.
Max_connections		150	Sets the maximum number of concurrent connections; default value is '100'.
Shared_buffers		32MB	Sets the number of the shared memory buffers used by the server; default value is typically '1000'.

NOTE

It is possible to change the configuration for all members by editing the form directly, but it is also possible to change the configuration for a subset of the group members. Click the green pencil icon, and then change the configuration settings for the members individually.

7. Click the **Save** button at the top of the form.

13.3. EDITING SCRIPT ENVIRONMENT VARIABLES

Scripts are autodetected on a server, as are other applications and services on the machine. Scripts can be configured and managed like any other resource, which means that JBoss ON allows you to both define configuration settings for and set up operations to run the scripts in inventory.

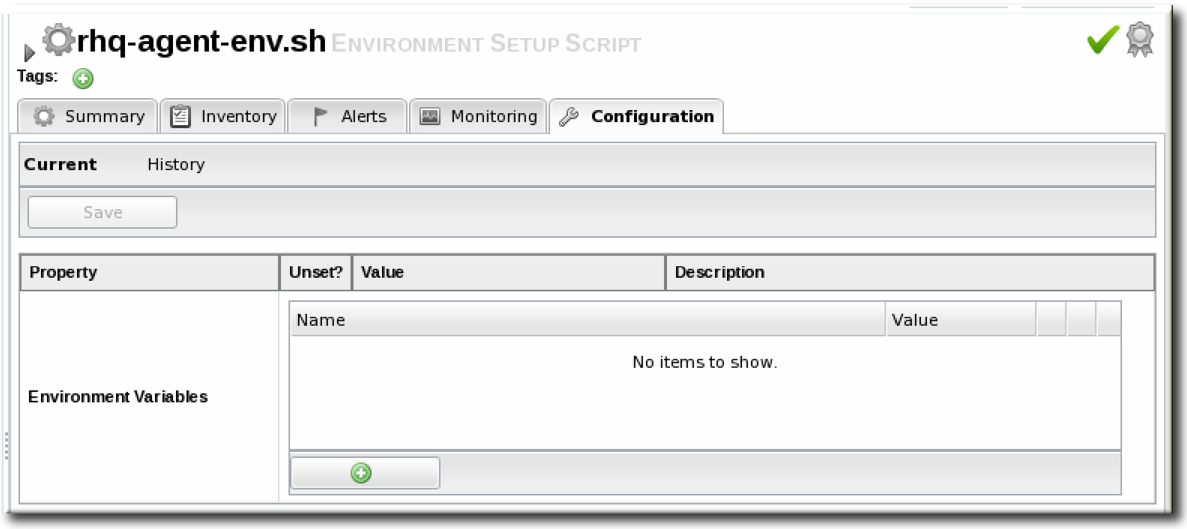
Whether a script is added or detected, there are only two configuration areas for the inventory entry: the path to the script, which places the script within the hierarchy, and any environment variables that should be set with the script.

These environment variables can be added and edited even after the script is imported:

IMPORTANT

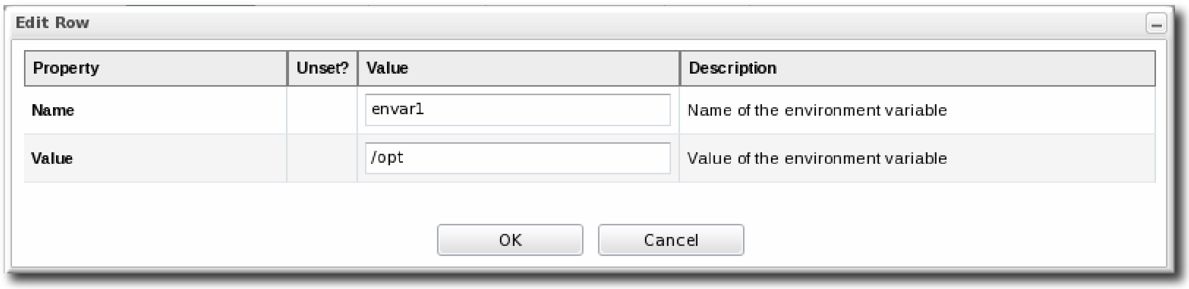
Before setting environment variables in the JBoss ON configuration, make sure that the environment *on the resource* is already configured properly.

1. Click the **`Inventory`** tab in the top menu.
2. Search for the script resource.
3. Open the **`Configuration`** tab for the script resource.
4. Click the green plus sign (+) to add an environment variable.



Property	Unset?	Value	Description
Environment Variables		Name / Value: No items to show.	

5. Enter the environment variable. Each new environment variable has the format *name=value;* and is added on a new line.



Property	Unset?	Value	Description
Name		envar1	Name of the environment variable
Value		/opt	Value of the environment variable

If the variable's value contains properties with the syntax **`%propertyName%`**, then JBoss ON interprets the value as the current values of the corresponding properties from the script's parent resource's connection properties.

6. After resetting an environment variable, restart the JBoss ON agent to propagate the changes. If the agent isn't restarted, new variables will not be propagated to the resource and will not resolve when the script is next executed, even if the configuration is correct.

NOTE

Add the line `@echo off` in Windows scripts to prevent echoing the executed commands along with the execution results.

13.4. CONFIGURING APACHE FOR CONFIGURATION MANAGEMENT (DEPRECATED)

JBoss ON manages configuration on Apache resources using an Augeas lens. A special version of Augeas is included with the JBoss ON agent which enables Apache configuration management. That Augeas lens must be enabled on the Apache resource for configuration management to work.

13.4.1. Considerations and Notes for Apache Configuration Management

Deprecated Augeas Plug-in

Apache configuration management is supported through a special Augeas agent plug-in, which connects with and manages the Augeas lens on the Linux instance. The Augeas agent plug-in is deprecated in JBoss ON 3.1.1 and may be removed in a later release.

Augeas and Apache Monitoring

The Augeas lens is not required for Apache monitoring. It is only used for Apache configuration management. An Apache resource can be monitored, with alerting, operations, and all other management tasks available without any additional configuration. The Augeas lens is used only for editing the Apache configuration files and virtual hosts through JBoss ON.

Supported Platforms for Apache Configuration

Apache configuration management is only supported for Apache instances installed on Linux.

Disabling noexec Options for Apache Directories

If the `/tmp` directory is configured a `noexec` in the `fstab` file, the agent throws exceptions because it cannot properly initialize the Augeas lens. In that case, the `Configuration` tab is unavailable for the Apache resource.

Make sure that the `/tmp` directory does not have `noexec` set as an option.

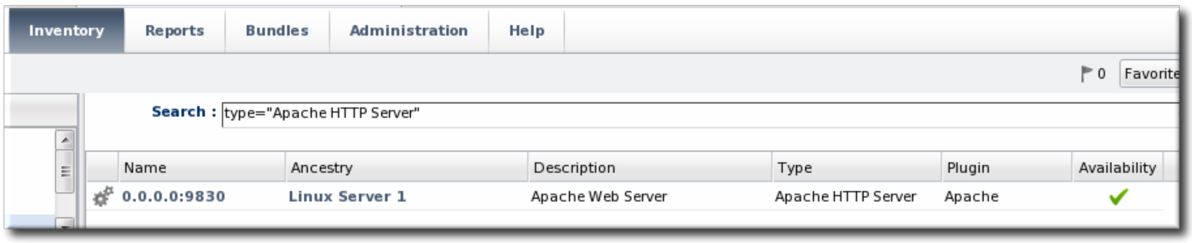
```
#
# /etc/fstab
#
tmpfs /dev/shm tmpfs defaults 0 0
devpts                  /dev/pts                devpts  gid=5,mode=620
0 0
sysfs                   /sys                    sysfs   defaults
0 0
proc                    /proc                   proc    defaults
0 0
```

13.4.2. Enabling Configuration Management

Apache configuration management is configured as one of the connection settings for the Apache resource, which sets how the agent connects to the resource.

1. Click the **`Inventory`** tab in the top menu.

2. Select the resource type in the **`Resources`** menu table on the left, and then search for the Apache resource.

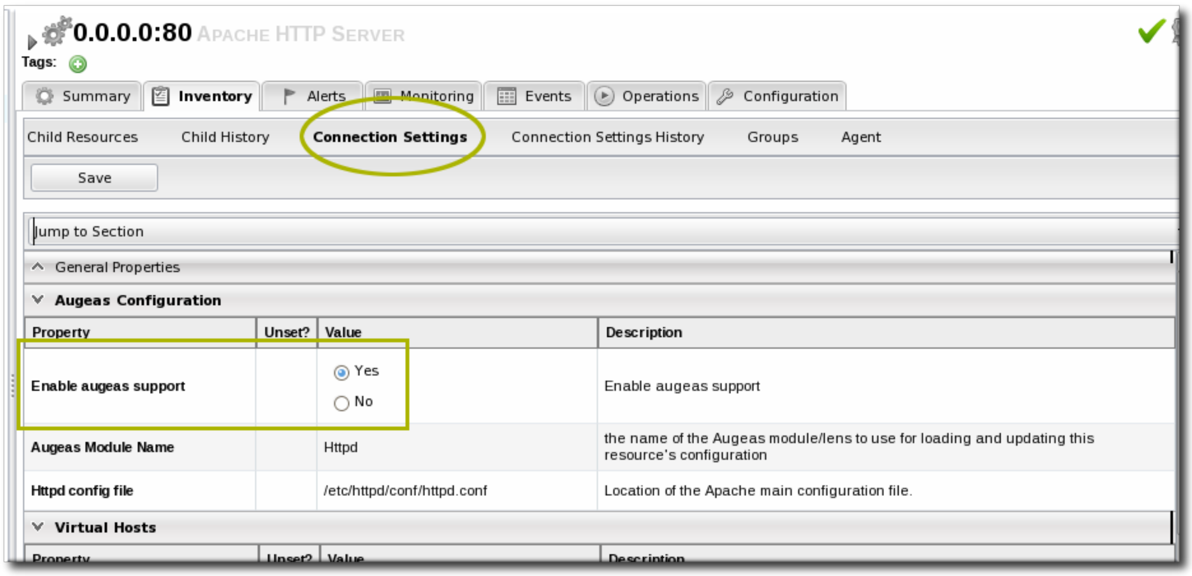


Search : type="Apache HTTP Server"

Name	Ancestry	Description	Type	Plugin	Availability
0.0.0.0:9830	Linux Server 1	Apache Web Server	Apache HTTP Server	Apache	✓

3. Click the IP address of the Apache instance.

4. Open the **`Inventory`** tab, then click the **`Connections`** subtab.



Property	Unset?	Value	Description
Enable augeas support		[✓ Yes] No	Enable augeas support
Augeas Module Name		Httpd	the name of the Augeas module/lens to use for loading and updating this resource's configuration
Httpd config file		/etc/httpd/conf/httpd.conf	Location of the Apache main configuration file.

5. Jump to the **`Augeas Configuration`** section.

6. Select the **`Yes`** radio button to enable the Augeas lens.

CHAPTER 14. TRACKING RESOURCE CONFIGURATION CHANGES

The revision numbers are global number across the JBoss ON server. For example, if Resource A is edited, then it gets revision #1. Then, when Resource B is edited, it gets revision #2, and the next edit gets #3.

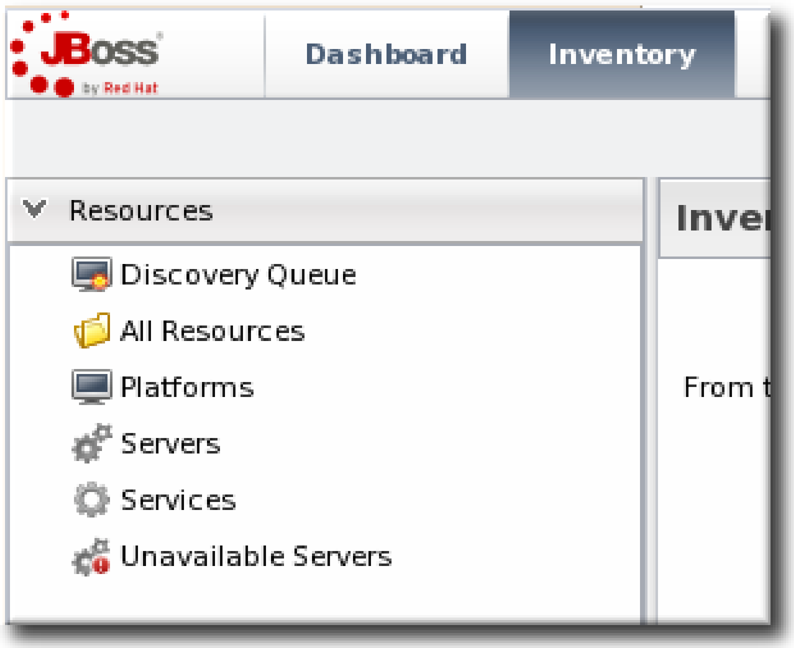
NOTE

A user may have the right to edit or revert configuration, but that does not mean that the user has the right to delete an item from the configuration history.

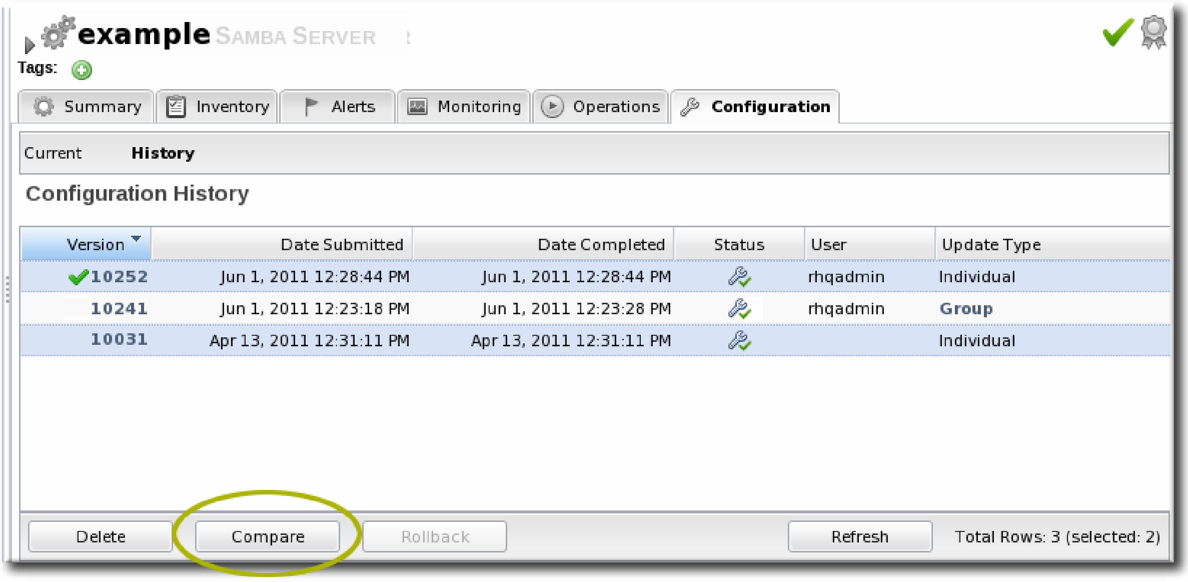
Deleting elements in the history requires the manage inventory permission.

14.1. TRACKING AND COMPARING CONFIGURATION CHANGES

1. Click the **`Inventory`** tab in the top menu.
2. Select the resource type in the **`Resources`** menu table on the left, and then browse or search for the resource.



3. Open the **`Configuration`** tab for the resource.
4. Click the **`History`** subtab.
5. Select the line of the configuration version to view or compare. Use the **`Ctrl`** key to select multiple versions. The current (most recent successful) configuration state is marked by a green check mark.



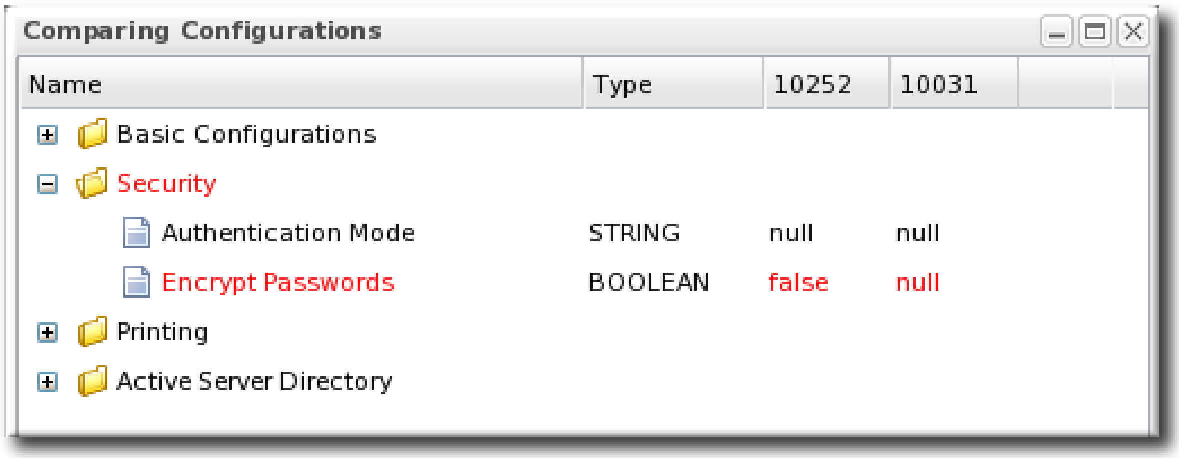
Configuration History

Version	Date Submitted	Date Completed	Status	User	Update Type
✓10252	Jun 1, 2011 12:28:44 PM	Jun 1, 2011 12:28:44 PM		rhqadmin	Individual
10241	Jun 1, 2011 12:23:18 PM	Jun 1, 2011 12:23:28 PM		rhqadmin	Group
10031	Apr 13, 2011 12:31:11 PM	Apr 13, 2011 12:31:11 PM			Individual

Total Rows: 3 (selected: 2)

6. Click the **`Compare`** button.

7. The pop-up window shows all of the changes in a directory-style layout, with each of the configuration areas as a high-level directory. Any changes are marked in red, and the values are shown for each selected version.

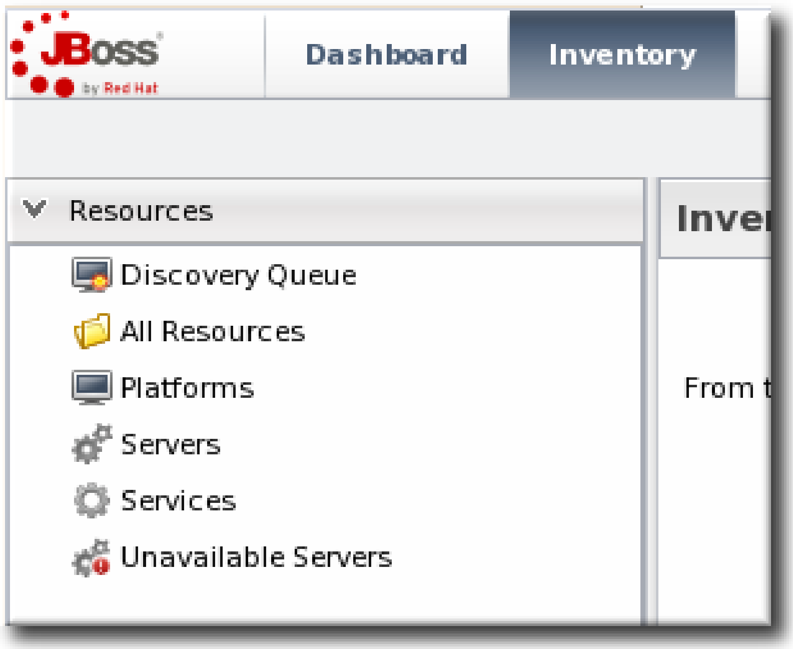

Comparing Configurations

Name	Type	10252	10031
Basic Configurations			
Security			
Authentication Mode	STRING	null	null
Encrypt Passwords	BOOLEAN	false	null
Printing			
Active Server Directory			

14.2. REVERTING CONFIGURATION CHANGES

1. Click the **`Inventory`** tab in the top menu.

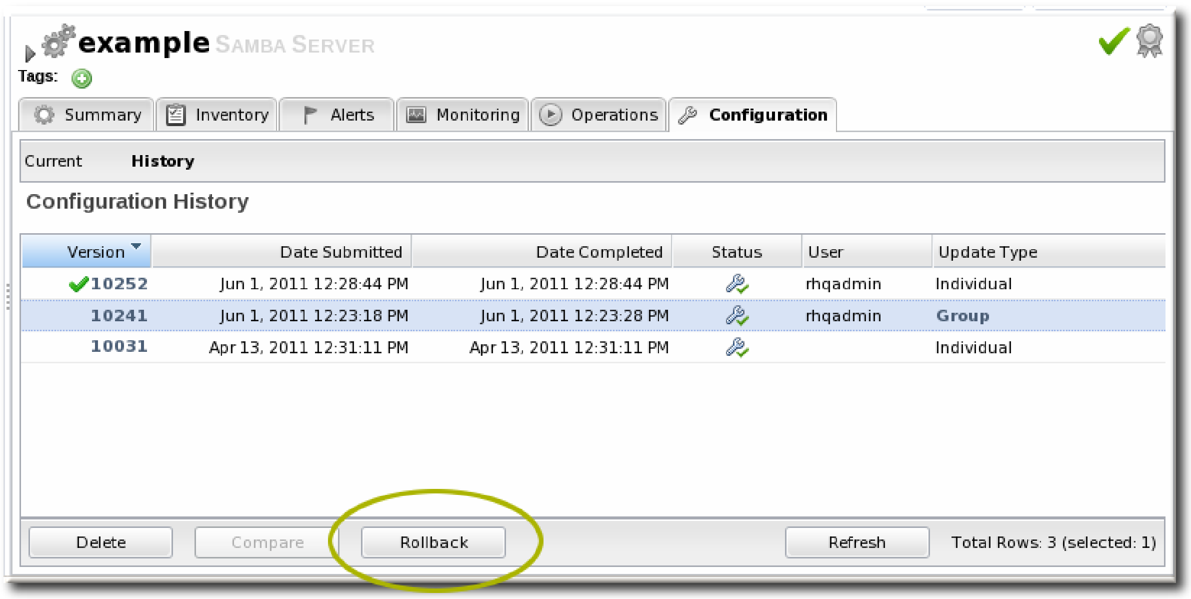
2. Select the resource type in the **`Resources`** menu table on the left, and then browse or search for the resource.



3. Open the **Configuration** tab for the resource.

4. Click the **History** subtab.

5. Select the line of the configuration version to roll back to. The current (most recent successful) configuration state is marked by a green check mark.



Version	Date Submitted	Date Completed	Status	User	Update Type
✓10252	Jun 1, 2011 12:28:44 PM	Jun 1, 2011 12:28:44 PM		rhqadmin	Individual
10241	Jun 1, 2011 12:23:18 PM	Jun 1, 2011 12:23:28 PM		rhqadmin	Group
10031	Apr 13, 2011 12:31:11 PM	Apr 13, 2011 12:31:11 PM			Individual

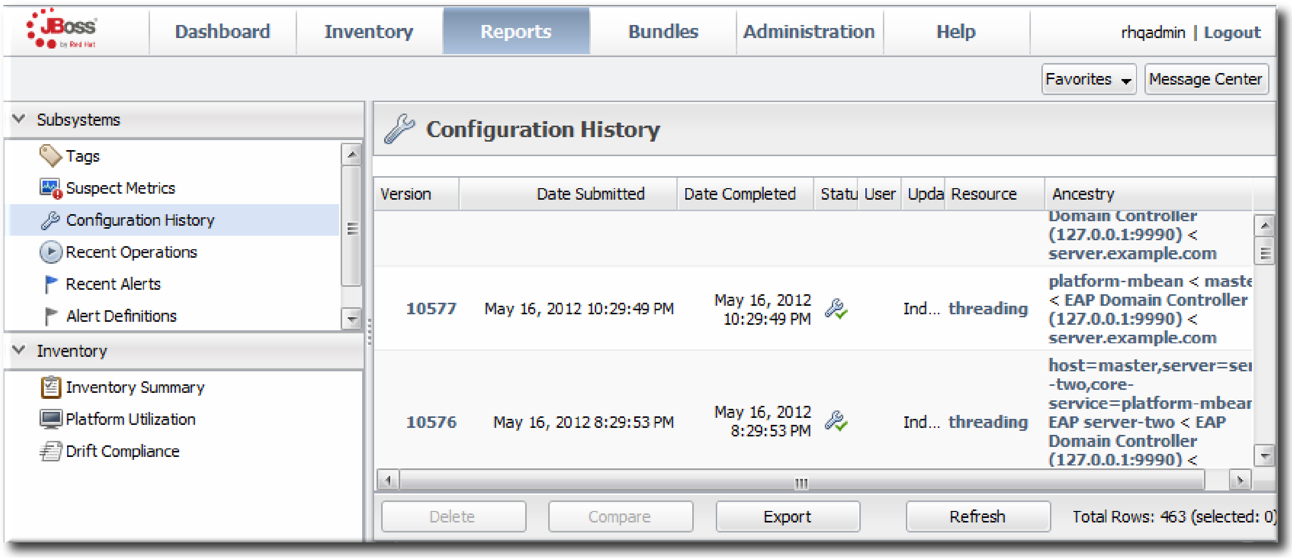
6. Click the **Rollback** button.

14.3. VIEWING THE CONFIGURATION HISTORY REPORT

Every resource (that supports configuration) tracks its own individual configuration change history, as in Section 14.1, “Tracking and Comparing Configuration Changes” .

JBoss ON also keeps a master list of all configuration changes, for all resources. This is displayed in the **Configuration History Report**, in the **Reports** tab.

As with the resource-level configuration history, the Configuration History shows the version number of the change, the time the configuration change was requested and completed, its status, and the requesting user. Because all resources are listed, the **`Configuration History Report`** also shows the resource name and its parent (and grandparent) to help disambiguate on which resource the change occurred.



Version	Date Submitted	Date Completed	Statu	User	Upda	Resource	Ancestry
							Domain Controller (127.0.0.1:9990) < server.example.com
10577	May 16, 2012 10:29:49 PM	May 16, 2012 10:29:49 PM	✓		Ind...	threading	platform-mbean < maste < EAP Domain Controller (127.0.0.1:9990) < server.example.com
10576	May 16, 2012 8:29:53 PM	May 16, 2012 8:29:53 PM	✓		Ind...	threading	host=master,server=ser -two,core-service=platform-mbear EAP server-two < EAP Domain Controller (127.0.0.1:9990) <

Total Rows: 463 (selected: 0)

Figure 14.1. Configuration History Report

The **`Configuration History Report`** supports compare operations, as with the resource-level configuration history. This is useful because you can compare not only versions of configuration for the same resource, but also for the same configuration property in different resources (of the same type). This helps administrators figure out where their infrastructures are and to pinpoint changes or diversions between the configuration for similar resources.

NOTE

Reports can be exported to CSV, which can be used for office systems or further data manipulation.

To export a report, simply click the **`Export`** button. The report will automatically be downloaded as **`configurationHistory.csv`**.

CHAPTER 15. MANAGING CONFIGURATION DRIFT

Much of the JBoss ON configuration management is designed around *implementing* changes for resources by editing configuration files or updating files and packages. But another aspect of managing configuration is *detecting changes*.

IT administrators must invest a significant amount of time planning the optimum configuration for systems in every type of environment, from production systems to internal resources. This ideal configuration includes file settings, software versions, and system settings. Resource configuration is going to change naturally over time, but administrators need to be able to track those changes to make sure that no unplanned or undesirable changes impact the resource. Defining a baseline configuration and tracking changes helps systems remain resilient during both maintenance and failures.

The unplanned changes that occur to a resource's configuration is called *drift*, as the configuration moves away from the designed baseline. Drift is common because of frequent software and hardware updates, particularly in a colocation facility or using virtual machines.

Production, staging, development, and recovery configurations are designed to have identical or near-identical configuration to maintain consistency. As the configurations within the different environments change, there emerges a *configuration gap*. Ultimately, this configuration gap can lead to disaster recovery failures or high availability failures because the configuration of the production system and the backup system are too different.

Drift monitoring provides a very general, freewheeling *content-based monitoring*. Rather than structured configuration management, drift monitoring tracks changes to content on the local filesystem. This means any changes in any files — even binary files[3].

15.1. UNDERSTANDING DRIFT

Of course, drift monitoring isn't as simple as checking for changes. One of the core questions is *what changes matter?* There are two conceptual parts to that question:

- What directories (and files within those directories) matter for drift monitoring? Even though a drift definition is defined for a resource, the actual drift detection is performed at the directory level. Drift monitoring, then, can hit anywhere on a platform — even outside resources managed by JBoss ON.
- How do you identify a change? Do you compare it to the version immediately before it or to an established baseline?

Once you identify what changes you want to monitor for drift, then you can use JBoss ON to set up monitoring and alerting effectively.

15.1.1. Drift Definitions and Detection

The first part of drift detection is identifying what you are monitoring.

JBoss ON defines a *drift definition* that sets the target location for drift monitoring. The target can be identified from some configuration element for the resource — it can be a directory or file on the filesystem, a resource configuration property, a resource plug-in parameter, or a monitoring trait. This target is the *base directory*. There are several different configuration areas within each resource that (potentially) define a filesystem location. The base directory is identified by determing what configuration type contains that information; that information area is the *value context*. A value context can be any one of four different areas:

- **From the plug-in configuration (pluginConfiguration).** This means, it can be taken from any of the connection properties for the resource. Connection properties can include log files, deployment directories, and installation directories, depending on the resource type.
- **From the resource configuration (resourceConfiguration).** This means, it can be taken from any of the configurable properties for the resource.
- **From a trait (measurementTrait).** Traits are informational measurement properties for the resource.
- **An explicit filesystem location.** If none of the resource properties have the proper location or if a different location should be used for drift, then the directory can be specified in the fileSystem property.

The actual value is the *value name*.

NOTE

Plug-in configuration (connection properties), resource configuration properties, and traits for each resource are listed in the *Resource Reference*.

For example, for a base directory of `/etc/` that only includes changes to `*.conf` files, the elements in the drift definition are:

```
Value context: fileSystem
Value name: /etc
Includes: **/*.conf
```

NOTE

Drift detection is performed at the filesystem level. This means that drift detection is not bound or restricted to a resource managed by JBoss ON. You can create a drift definition for a platform and set it to monitor any file or directory on that platform, even if it is outside the JBoss ON inventory, as long as that directory is accessible to the system user that the JBoss ON agent runs as.

By default, every subdirectory and file underneath the base directory is monitored for drift. The *includes/excludes* options define subdirectories or files that are explicitly included or explicitly excluded from drift monitoring. If *includes* is used, then only the specified directories or files are monitored and everything else is implicitly excluded, and vice versa for *excludes*. Included/excluded directories and files are identified by a path and a pattern. The path is the starting point beneath the base directory, and the pattern matches the file to monitor.

Table 15.1. Combinations to Include Specific Files

Files to Monitor for Drift	'Includes' Path	'Includes' Pattern
/etc and all its subdirectories	Blank	Blank
For *.conf files in /etc and all subdirectories	.	**/*.conf[a]

Files to Monitor for Drift	'Includes' Path	'Includes' Pattern
For *.conf files only in the /etc directory, with no subdirectories (/etc/*.conf)	.	*.conf
For *.conf files only in a subdirectory one level below /etc (/etc/*/*.conf)	Not possible	Not possible
For any file in a specific subdirectory (yum.repos.d/) below /etc	yum.repos.d (subdirectory name)	Blank

[a] This must have a double asterisk for the directory part. It will not work with a single asterisk.

The drift definition also sets an interval, or frequency, for how often the agent checks for drift. This is a very important setting for performance, both for JBoss ON and for data management. Setting a frequency that is too high risks missing changes or lumping changes together into large (and therefore difficult to manage) snapshots. However, setting the interval too low can impact JBoss ON agent and server performance.

The key thing about the drift definition is that it sets what to look at and how often to look.

NOTE

All drift detection runs are performed outside the agent plug-in and independent of the resource state. A drift detection scan can be run even if the resource is not running.

15.1.2. Snapshots, Deltas, and Baseline Images

The second part of drift detection is identifying how you want to define a *change*. Change is comparative. It takes the current version of a file and compares it to some previous version. The question for drift management is what previous version to compare to.

When a drift definition is first created, the agent collects all of the files in that base directory and subdirectories and sends information about them to the server. This collection is the initial file set.

From that point onward, the agent only sends change information about the files. Each set of changes is a *snapshot*. For text files, the change information includes the content of the file and diffs (both constructed on the JBoss ON server based on patches sent by the agent). For binary files, drift only records that the file change and displays a SHA and timestamp. A snapshot is always based on real files from a real resource.

NOTE

The agent does not send the actual files to the JBoss ON server. The agent sends information about file changes back to the server. These updates only contain the deltas between versions; they're not full files. This minimizes the network I/O.

The actual diffs are generated on the server from the content that the server stores.

The way that a snapshot is created is by comparing the current files against the agent's version. There are two ways that this comparison can be made:

- It can compare against the next-most recent version of the files.
- It can compare against a defined, stable baseline.

The first option is a *rolling snapshot*. This is the simplest setting, because it just keeps a running tally of changes.

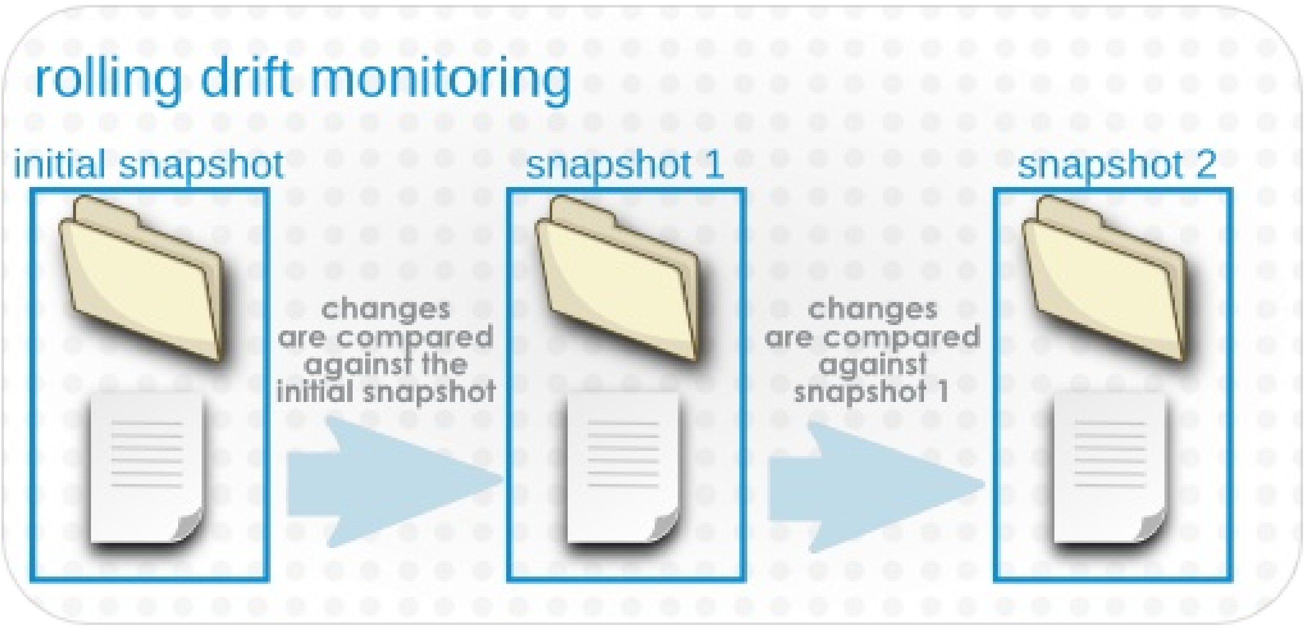

rolling drift monitoring
initial snapshot
snapshot 1
snapshot 2
changes are compared against the initial snapshot
changes are compared against snapshot 1

Figure 15.1. Rolling Snapshots

The second option is a *pinned snapshot*, and this is the method that gives administrators the most insight and control over drift. A pinned snapshot means that some image of the base directory has the optimal, approved configuration and this snapshot is selected as the baseline. It is in essence pinned in place, and every subsequent change is compared against that pinned snapshot, rather than being compared against each other.

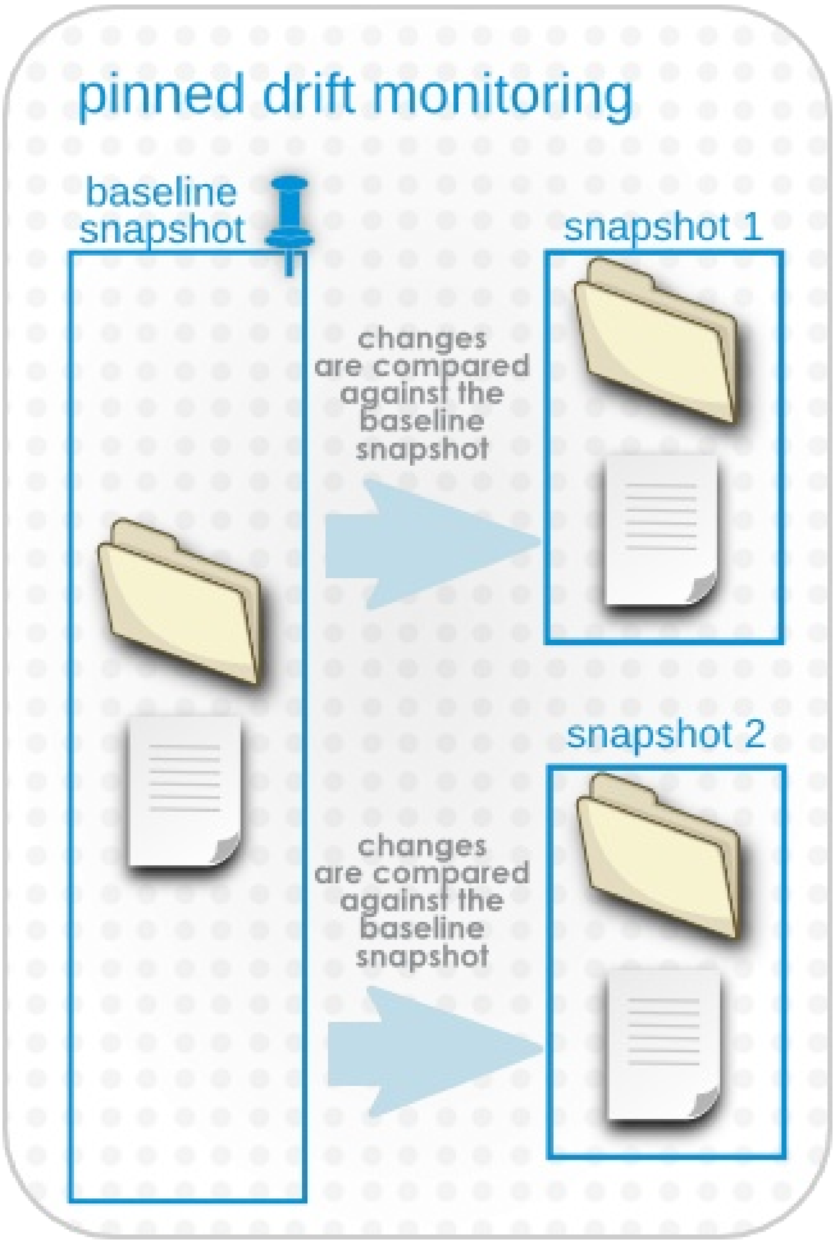

pinned drift monitoring
baseline snapshot
snapshot 1
changes are compared against the baseline snapshot
snapshot 2
changes are compared against the baseline snapshot

Figure 15.2. Pinned Snapshots

Snapshots exist at the resource-level, because they are based on the real files that exist on a system. When a snapshot is pinned to a resource-level definition, then any changes made on that system are compared to that snapshot. When the current file version matches the pinned snapshot, the resource is *compliant*.

A snapshot for a resource can also be pinned to a drift template — then, it is applied to every definition attached to that template. This is really powerful for administrators. For example, you can use a staging or development server to create the best system configuration for EAP performance, and then apply that EAP baseline snapshot to every EAP server in the production environment by pinning it to a drift template. You can see immediately all the EAP configuration relative to your defined ideal.

15.1.3. Destination Directories with Special File Types

Drift looks at both files and directories on the local system to generate snapshots and identify changes. The majority of these files and directories are going to be real files, but Unix does have some special file types, and the drift operation may encounter those files as part of processing the destination directory. There are some behavior implications, particularly with symbolic links and named pipes.

With symbolic links, drift detection follows any links back to the original file or directory, and includes those files in the snapshot. For example, if a symlink is set up for some library:

```
ln -ls /home/dev/libs /usr/share/jbossas/server/libs
```

If drift is configured on the **libs/** directory in the JBoss Enterprise Application Platform home directory, it will follow the symlink back to **/home/dev/libs**, and include all of those files in the drift snapshot.

IMPORTANT

Be careful when configuring drift against a directory which contains symlinks. All of the linked files will be included as part of the drift target.

If the linked directory has a large number of files, then drift detection runs may take longer than expected. Additionally, changes in that symlinked directory may have an unexpected impact on drift detection by recording many changes as drift when they weren't intended to be.

If you do not want to include the symlinked directory in the drift definition, use the **excludes** parameter in the drift definition to exclude the symlink.

The other special file type that is common on Unix systems is a *pipe*. As with a symlink, drift detection runs can detect a fifo file within the target directory. However, unlike symlinks, drift cannot process the fifo file, which causes drift detection to hang.

NOTE

Use the **excludes** parameter in the drift definition to exclude any named pipes in the target directory.

Table 15.2. Drift Definitions and Unix File Types

File Type	Supported by Drift?
File	Yes
Directory	Yes
Symbolic link	Yes
Pipe	No
Socket	No
Device	No

15.1.4. Drift and Resource Types

Whether drift is supported is defined in the resource type (which is discussed in Section 15.3.1, “About Resources and Drift Definition Templates”). If a drift template is defined in the resource type's **rhq-**

`plugin.xml` descriptor, then that resource type supports drift. The template is a starting point (not an enforced configuration, like alert or metric collection templates).

Three JBoss ON standard resource types support drift:

- All platforms
- JBoss EAP 6 (AS 7), and all resources which use the JBoss AS 5 plug-in
- JBoss AS/EAP 5, and all resources which use the JBoss AS 5 plug-in
- JBoss AS/EAP 4 (deprecated)

Because drift support is defined in the plug-in descriptor, custom plug-ins can be created that add drift support for those resource types. For examples of writing agent plug-ins with drift support, see "Writing Custom JBoss ON Plug-ins ."

NOTE

Drift is not supported on embedded web applications, such as an embedded WAR under an EAR application.

Drift detection is performed at the *directory level*. It is not tied to a specific resource. This means that drift detection can be run even when a resource is not running. It also means that drift detection can occur for an application, service, or file that is not managed by JBoss ON or for a resource type that does not otherwise support drift.

To monitor an entity outside the three supported resources, just configure drift detection on the platform resource and define the base directory as whatever directory path is used by the application or service you want to monitor.

15.1.5. Space Considerations for Drift Monitoring

Configuring drift monitoring can have a significant impact on disk space requirements.

JBoss ON stores multiple snapshots. This is part of versioning control, allowing changes to monitored directories to be reverted and managed.

Therefore, the system which hosts the backend database (Oracle or PostgreSQL) must have enough disk space to store all versions of all bundles. Additionally, the database itself must have adequate tablespace for the content.

Size considerations can also affect how drift monitoring is configured, in several ways:

- The size of the directory being monitored. In some cases, it may be better to monitor multiple smaller subdirectories rather than one large, high-level directory.
- The frequency of drift detection runs, balancing the need to capture changes versus the number of backup copies.
- How long drift snapshots are stored. By default, unused snapshots (meaning, unpinned snapshots) are stored for 31 days and then deleted. Changing how long snapshots are stored can help manage the database size.

When calculating the required amount of space, estimate the size the targeted directories, and then the frequency that snapshots are taken to get an idea of how many snapshots will be stored. At a minimum, **have twice that amount of space available**; both PostgreSQL and Oracle require twice the database size to perform cleanup operations like vacuum, compression, and backup and recovery.

15.1.6. Back to Drift Monitoring

The goal of setting up drift detection is to provide clarity into how systems and application servers are being modified. JBoss ON provides two ways to manage drift.

Drift Monitoring

Drift monitoring is the ability to track changes to target locations. The JBoss ON GUI allows you to view snapshots all together, compare changes for individual files between snapshots, view the current configuration, and view change details. It also provides inventory and drift reports and indicates, at a glance, whether a resource is compliant with an associated pinned snapshot.

Drift Alerting

A specific alert condition exists that will trigger an alert whenever there is drift. For rolling snapshots, this will send an alert once (and only once) for each drift snapshot. For pinned snapshots, the drift alert is fired for every detection run for as long as the resource is out of compliance, even if there are no subsequent changes.

NOTE

There is no direct way to remedy drift through the JBoss ON GUI. However, it is possible to launch a JBoss ON CLI script in response to a drift alert. For example, you can create a patch of your ideal EAP configuration. If an EAP server drifts from that configuration, then you can use a JBoss ON CLI script to deploy that EAP patch bundle to the drifting EAP server.

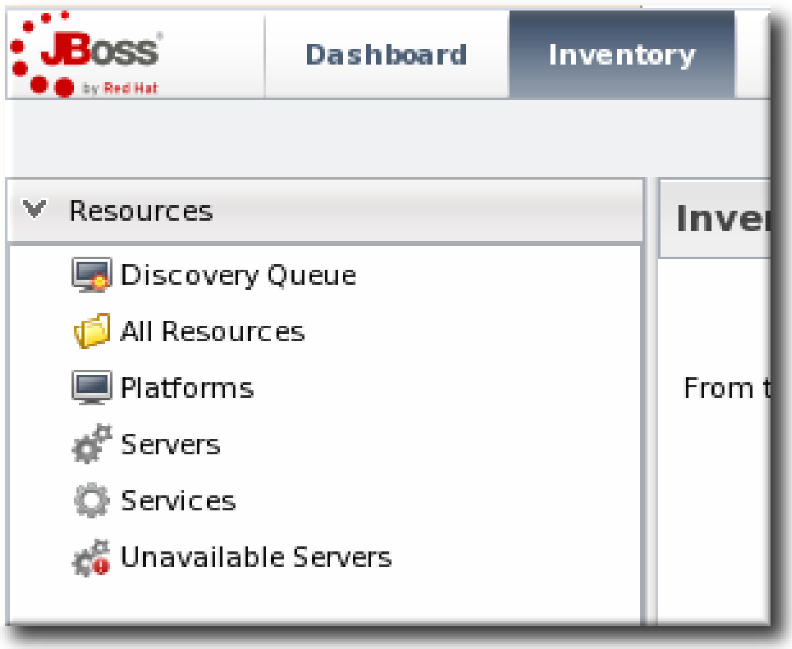
15.2. ADDING A DRIFT DEFINITION FOR A RESOURCE

IMPORTANT

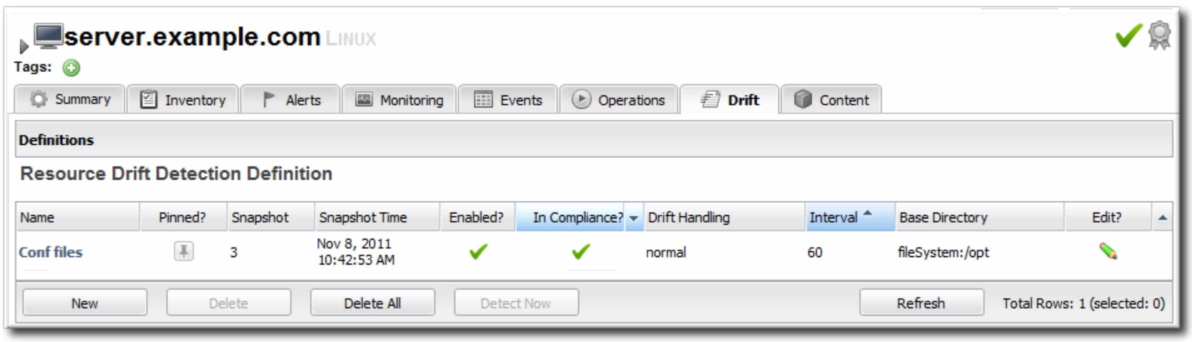
The directories where drift detection is being run *cannot* be changed after the definition is created. Be careful to get the base directory and the included and excluded files properly configured before saving.

1. Click the **`Inventory`** tab in the top menu.

2. Select the resource type in the **`Resources`** menu table on the left, and then browse or search for the resource.



3. Open the **`Drift`** tab for the resource.



Definitions

Resource Drift Detection Definition

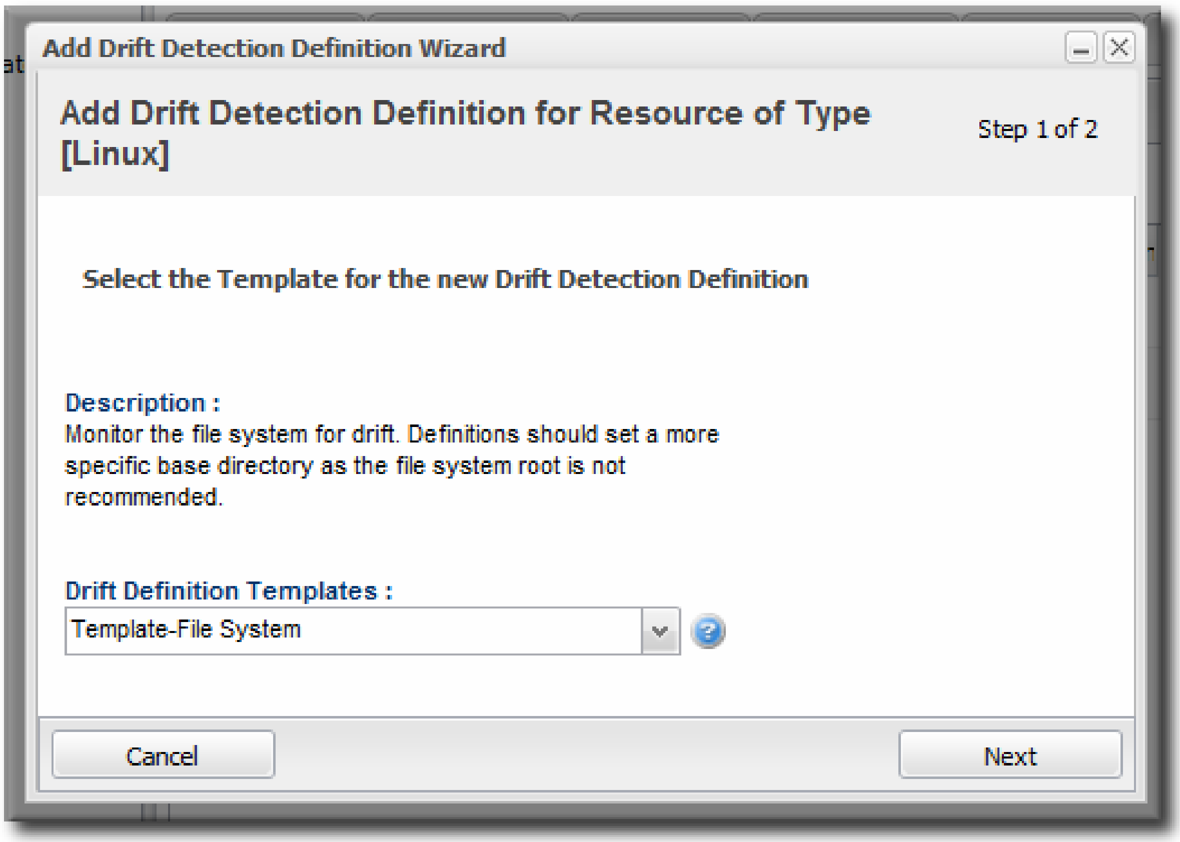
Name	Pinned?	Snapshot	Snapshot Time	Enabled?	In Compliance?	Drift Handling	Interval	Base Directory	Edit?
Conf files		3	Nov 8, 2011 10:42:53 AM	✓	✓	normal	60	fileSystem:/opt	

Total Rows: 1 (selected: 0)

4. Click the **New** at the bottom to add a new definition.

5. Select the template to use to as the basis for the new definition.

   Plug-in defined templates are defined in the platform and JBoss server resources, as well as any other resource which supports drift monitoring. Additional, user-defined templates can be also be created and applied.



Add Drift Detection Definition for Resource of Type [Linux]

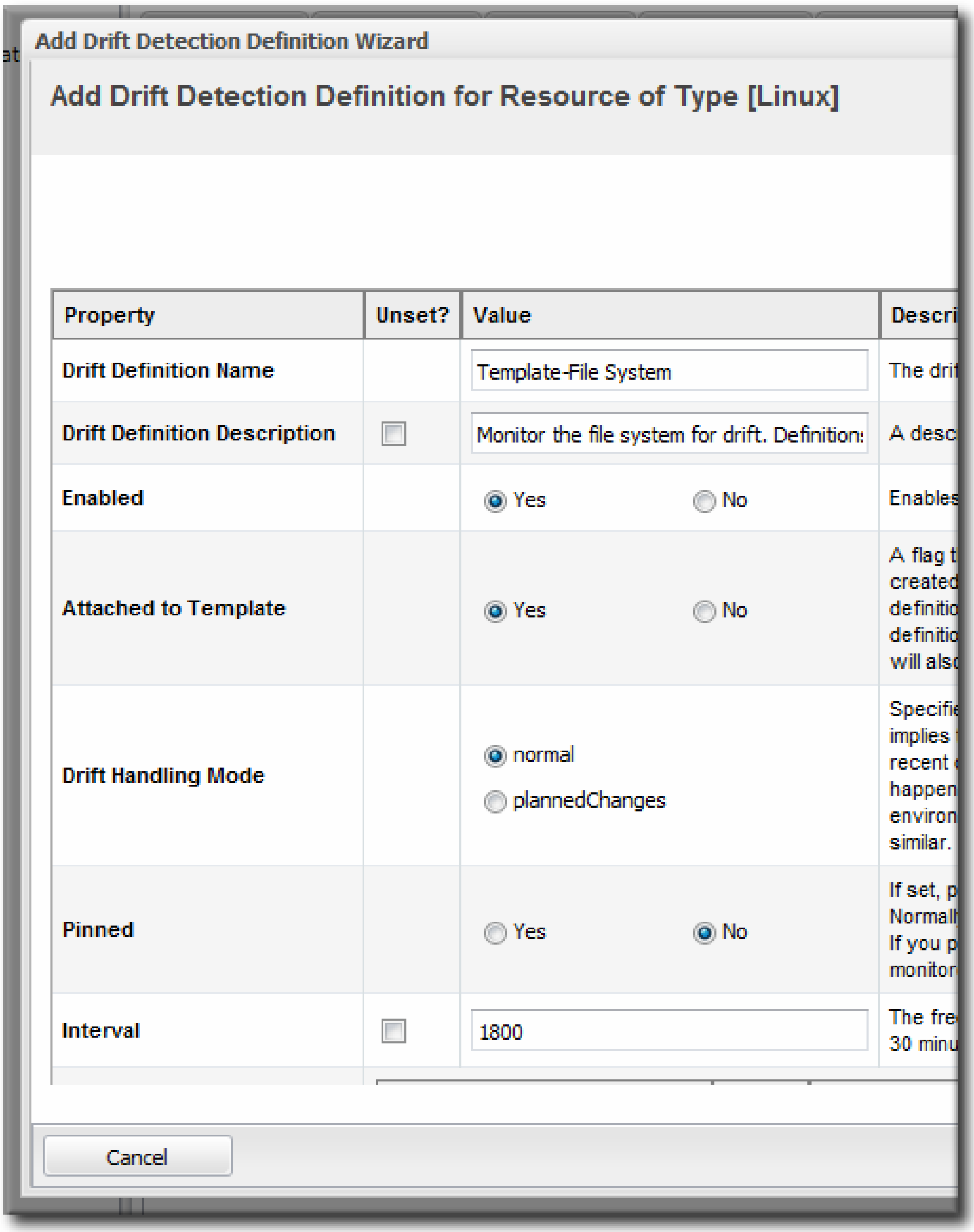
Step 1 of 2

Select the Template for the new Drift Detection Definition

Description :
Monitor the file system for drift. Definitions should set a more specific base directory as the file system root is not recommended.

Drift Definition Templates :
Template-File System

6. Give a unique name to the definition. The name and the base directory are combined to identify the definition within JBoss ON.



Add Drift Detection Definition for Resource of Type [Linux]

Property	Unset?	Value	Descri
Drift Definition Name		Template-File System	The dri
Drift Definition Description	☐	Monitor the file system for drift. Definition:	A desc
Enabled		[✓ Yes] No	Enables
Attached to Template		[✓ Yes] No	A flag t created definitio definitio will also
Drift Handling Mode		[✓ normal] plannedChanges	Specifie implies recent happen environ similar.
Pinned		Yes [✓ No]	If set, p Normall If you p monitor
Interval	☐	1800	The fre 30 minu

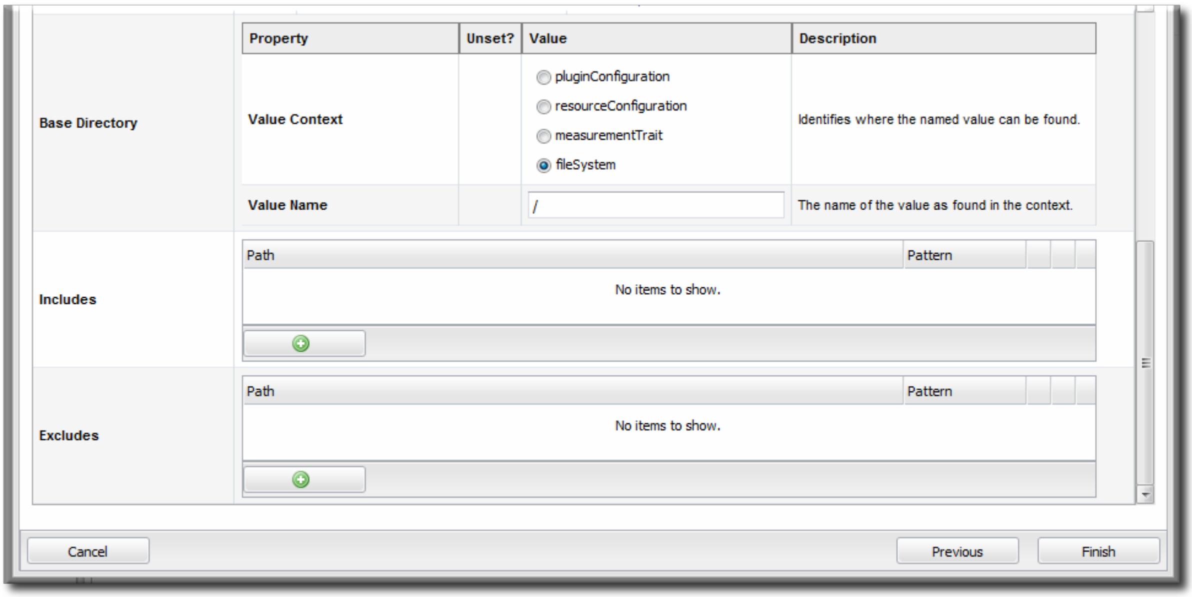
7. Define the settings for the definition, like the interval and whether it is associated with the template. The properties are listed in Table 15.3, “Drift Definition Properties”.

8. Set the base directory. This is the top-most directory where drift detection is run for the definition, and the scan recourses down.

   The template itself defines an initial directory, but it may be useful to set a more specific directory to use.

9. Click the button with the green plus (+) sign to add a subdirectory to include or exclude. The directory can be the base directory by specifying a period (.) as the directory. The pattern identifies which files within the directory to recognize by the service, either to explicitly include or explicitly exclude.

The filters support Ant-like FilePatterns, using a path and pattern. The patterns support asterisks (*) as wildcards for any number of characters and question marks (?) for single character wild cards. For example, `**/*.conf` can be used to include only `.conf` files in any subdirectory.



	Property	Unset?	Value	Description
Base Directory	Value Context		( ) pluginConfiguration ( ) resourceConfiguration ( ) measurementTrait (●) fileSystem	Identifies where the named value can be found.
	Value Name		/	The name of the value as found in the context.

Includes

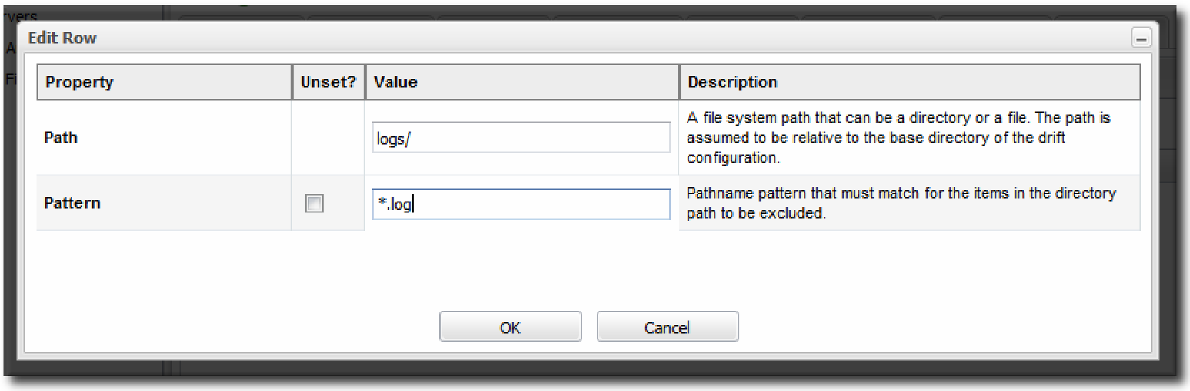
Path	Pattern
No items to show.	

Excludes

Path	Pattern
No items to show.	

Cancel · Previous · Finish

There can be multiple include/exclude filters. Each directory and pattern can be added separately.



Edit Row

Property	Unset?	Value	Description
Path		logs/	A file system path that can be a directory or a file. The path is assumed to be relative to the base directory of the drift configuration.
Pattern	☐	*.log	Pathname pattern that must match for the items in the directory path to be excluded.

OK · Cancel

Table 15.3. Drift Definition Properties

Property	Description
Name	A name for the drift detection definition. The name and the base directory, together, uniquely identify the definition.

Property	Description
Base Directory: Value Context	The type of configuration property which is used to identify the base directory. This identifies what type of element in the resource supplies the value. There are four options: • File system, which is simply an absolute directory path on the resource. This directory must exist for drift to work. • Resource configuration, which is a configuration property defined for the resource. • Trait, which is one of the monitored traits for that resource. • Plug-in configuration property, which is a property defined in the resource plug-in.
Base Directory: Value Name	The actual value for the drift detection definition to use for the base directory context. For example, if this is a file system context, then the value name is the directory path.
Includes	Explicitly includes directories, files, or files and directories matching a pattern, relative to the base directory, in the drift detection. The filters support Ant-like FilePatterns, using a path and pattern. The patterns support asterisks (*) as wildcards for any number of characters and question marks (?) for single character wild cards. If a pattern is used, then a path must be specified, even if the path is the base directory. For example, to include only `.conf` files in the base directory, the pattern is `*.conf` and the path is a period (**.**) to indicate the local directory.
Excludes	Explicitly excludes directories, files, or files and directories matching a pattern, relative to the base directory, from the drift detection. The filters support Ant-like FilePatterns, using a path and pattern. The patterns support asterisks (*) as wildcards for any number of characters and question marks (?) for single character wild cards. If a pattern is used, then a path must be specified, even if the path is the base directory. For example, to include only `.conf` files in the base directory, the pattern is `*.conf` and the path is a period (**.**) to indicate the local directory.

Property	Description
Enabled	Enables or disables the definition. Disabling a definition means that no detection scans are run.
Interval	Sets the frequency, in seconds, where the definition is eligible for a detection run. This is not a hard setting. Because load or other scheduled operations for the agent, the detection run is not guaranteed to run at the specified interval.
Pinned	Sets whether drift is determined in a rolling way or if it is associated (pinned) with a baseline snapshot. If this is set when the definition is created, then the initial snapshot is used as the baseline. Definitions attached to a pinned template cannot be unpinned. Definitions which are attached to an unpinned template or which are not attached to a template can be pinned or unpinned freely.
Drift Handling Mode	Sets whether drift changes are treated as events which trigger an alert (the default) or as expected, so that no alerts are triggered.
Attached to Template	Sets whether the resource-level definition is subordinate to a template. If it is attached to a template, then any changes to the template are reflected in the resource definition, including if the template is deleted. By default, definitions are attached to the template from which they are created.
Description	A simple text description of the definition.

15.3. CREATING A DRIFT DEFINITION TEMPLATE

Every time a new drift definition is created, it is based on an existing template for the resource type. At least one template is defined for resource types (by default, platforms and JBoss application servers) in their resource plug-in. Additional templates can be created by users.

15.3.1. About Resources and Drift Definition Templates

Resources of the same type frequently need to have the same, or similar, configuration settings. Particularly for an area like configuration drift, consistency is crucial for accurate and timely IT maintenance. JBoss ON allows this consistency using *drift definition templates*. Much like alert and monitoring templates, drift definition templates are defined for a resource *type* (regardless of whether any resources of that type actually exist) and can then be applied to specific resources in the inventory.

Drift templates are a little different than other template types in JBoss ON. First, a drift definition template is exactly that — it is an outline of default settings and values to use when creating a resource-level drift definition. It is not automatically applied to resources.

Additionally, there are two types of drift templates: plug-in defined templates and user-defined templates.

At least one drift definition template is actually defined as part of the plug-in for a resource type. Defining a template in the plug-in descriptor is what indicates that a resource type supports drift monitoring. These are *plug-in defined* templates; these are the default templates.

Having a plug-in defined template is the way that the JBoss ON agent recognizes that a particular resource type supports drift monitoring. So, the plug-in defined template has a dual purpose. It lets JBoss ON know what resource types support drift, and it gives basic input to help administrators start making their own drift definitions.

Example 15.1. A JBoss Server Drift Definition Template

```
<drift-definition name="Template-Base Files"
                  description="Monitor base application server files
for drift. It defines monitoring for some standard sub-directories of
the HOME directory.  Note, it is not recommeded to monitor all files for
an application server. There are many files, and many temp files.">
    <basedir>
         <value-context>pluginConfiguration</value-context>
         <value-name>homeDir</value-name>
    </basedir>
    <includes>
 <include path="bin" />
        <include path="lib" />
        <include path="client" />
    </includes>
</drift-definition>
```

New drift definition templates can be added by administrators, in addition to the plug-in defined template; these are *user-defined* templates. These templates can reflect the unique infrastructure and application environment.

A resource-level drift definition is always based on a drift template, which provides some default values to the definition during creation. That template can be plug-in defined or user-defined. Resource-level drift definitions do not have to be attached to a template, so they do not have to be changed every time the template changes, but they are always based on an existing template.

There are some things to remember about drift definition templates:

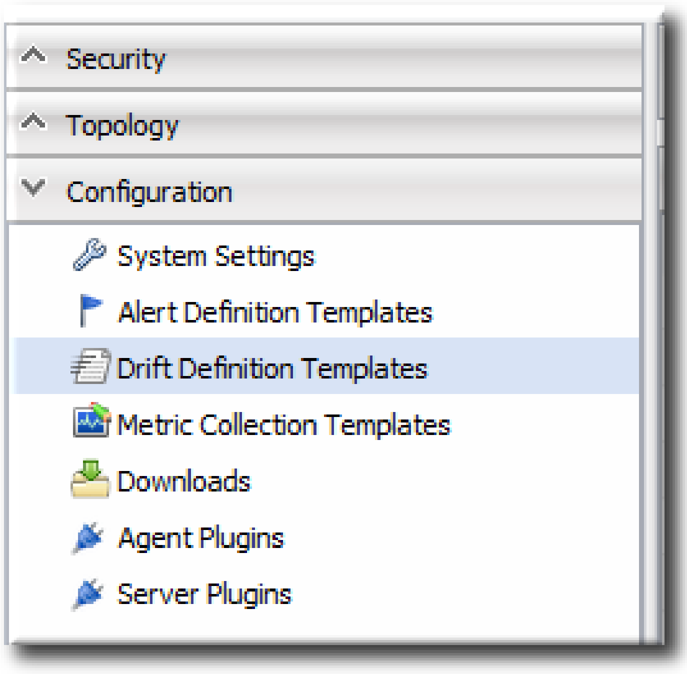
- Drift templates **are not** automatically applied to a resource, unlike other template types in JBoss ON. Drift templates are used as the basis for creating resource-level definitions.
- Default drift templates are defined for resources as part of their plug-in descriptor. Custom, user-defined templates can be added along with those defaults.
- Every drift definition is based on a template initially, even if that definition is not attached to that template post-creation.
- Snapshots (the file sets associated with drift definitions) always originate on a resource with a drift definition first. For any content to be associate with a template, the resource-level snapshot has to be promoted up to the template. Drift templates do not generate snapshots or files and then push that down to the resource.

15.3.2. Creating a Drift Definition Template

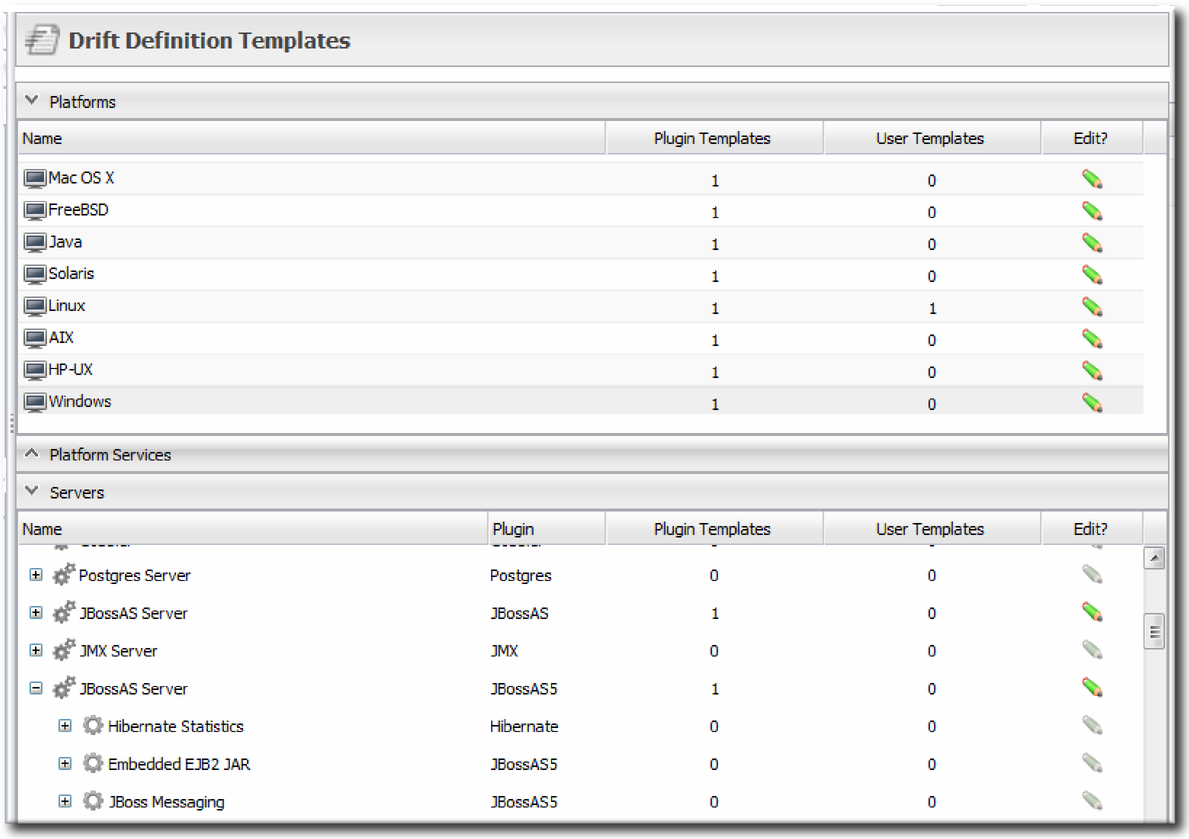
A drift template creation form is almost identical to a resource-level drift definition, with two exceptions: it cannot be pinned to a snapshot at the time it is created and it cannot be associated with another template. Obviously, a template is not dependent on another template (even though it *is* created from another template.) Being unable to pin a template to a snapshot is also logical; when a template is created, it is not associated with any resources. So, it is not possible to generate snapshots, which means that there is nothing to pin the template to.

1. Click the **`Administration`** tab in the top menu.

2. Select the **`Drift Definition Templates`** menu table on the left.



3. Click the pencil icon for the resource type to add the template to. Not all resources support drift, so they cannot be selected.



Drift Definition Templates

Platforms

Name	Plugin Templates	User Templates	Edit?
Mac OS X	1	0	
FreeBSD	1	0	
Java	1	0	
Solaris	1	0	
Linux	1	1	
AIX	1	0	
HP-UX	1	0	
Windows	1	0	

Platform Services

Servers

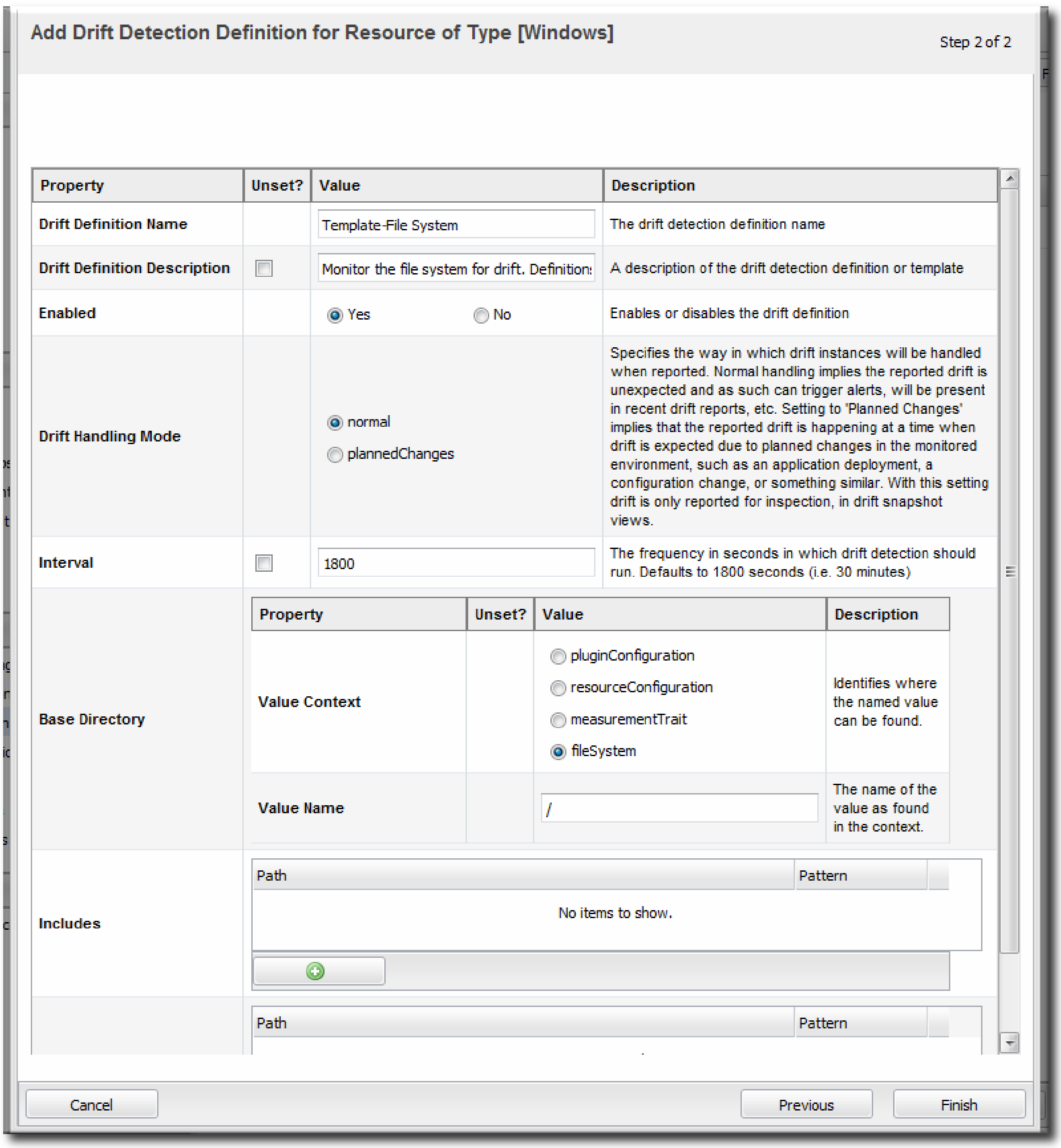
Name	Plugin	Plugin Templates	User Templates	Edit?
Postgres Server	Postgres	0	0	
JBossAS Server	JBossAS	1	0	
JMX Server	JMX	0	0	
JBossAS Server	JBossAS5	1	0	
Hibernate Statistics	Hibernate	0	0	
Embedded EJB2 JAR	JBossAS5	0	0	
JBoss Messaging	JBossAS5	0	0	

4. Click the **New** at the bottom to add a new template.

5. Select the template to use to as the basis for the new template.

   Plug-in defined templates are defined in the platform and JBoss server resources, as well as any other resource which supports drift monitoring. Additional, user-defined templates can be also be created and applied.

6. Give a unique name to the template. The name and the base directory are combined to identify the definition within JBoss ON.



Add Drift Detection Definition for Resource of Type [Windows]

Step 2 of 2

Property	Unset?	Value	Description
Drift Definition Name		Template-File System	The drift detection definition name
Drift Definition Description	☐	Monitor the file system for drift. Definition:	A description of the drift detection definition or template
Enabled		[✓ Yes] No	Enables or disables the drift definition
Drift Handling Mode		[✓ normal] plannedChanges	Specifies the way in which drift instances will be handled when reported. Normal handling implies the reported drift is unexpected and as such can trigger alerts, will be present in recent drift reports, etc. Setting to 'Planned Changes' implies that the reported drift is happening at a time when drift is expected due to planned changes in the monitored environment, such as an application deployment, a configuration change, or something similar. With this setting drift is only reported for inspection, in drift snapshot views.
Interval	☐	1800	The frequency in seconds in which drift detection should run. Defaults to 1800 seconds (i.e. 30 minutes)

Base Directory

Property	Unset?	Value	Description
Value Context		pluginConfiguration resourceConfiguration measurementTrait [✓ fileSystem]	Identifies where the named value can be found.
Value Name		/	The name of the value as found in the context.

Includes

Path	Pattern
No items to show.	

Path	Pattern

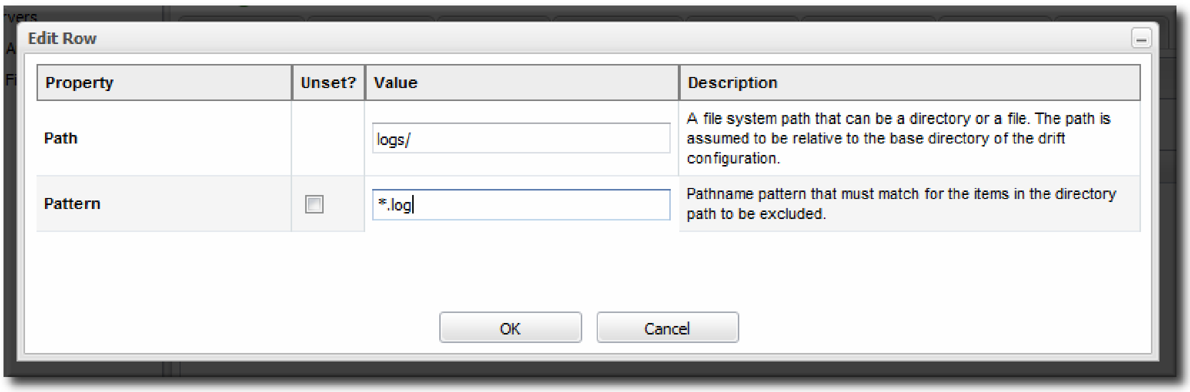
Cancel Previous Finish

7. Define the settings for the definition, like the interval and whether it is enabled by default. The properties are listed in Table 15.3, "Drift Definition Properties".

8. Set the base directory. This is the top-most directory where drift detection is run for the definition, and the scan recourses down.

9. Click the button with the green plus (+) sign to add a subdirectory to include or exclude. The directory can be the base directory by specifying a period (.) as the directory. The pattern identifies which files within the directory to recognize by the service, either to explicitly include or explicitly exclude.

   The filters support Ant-like FilePatterns, using a path and pattern. The patterns support asterisks (*) as wildcards for any number of characters and question marks (?) for single character wild cards. For example, `**/*.conf` can be used to include only `.conf` files in any subdirectory.

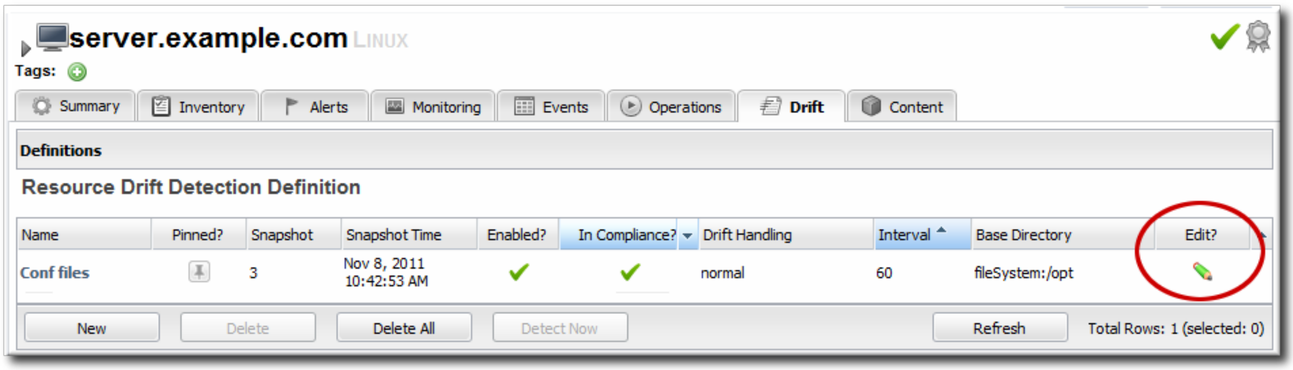


Property	Unset?	Value	Description
Path		logs/	A file system path that can be a directory or a file. The path is assumed to be relative to the base directory of the drift configuration.
Pattern	☐	*.log	Pathname pattern that must match for the items in the directory path to be excluded.

15.4. EDITING DRIFT DEFINITIONS

Most entries in JBoss ON are edited by clicking their name or double-clicking their row in a list. However, for drift definitions, clicking the name or double-clicking the row opens up the list of snapshots for that definition — not the definition entry itself.

To edit a drift detection definition, click the pencil icon.



Name	Pinned?	Snapshot	Snapshot Time	Enabled?	In Compliance?	Drift Handling	Interval	Base Directory	Edit?
Conf files		3	Nov 8, 2011 10:42:53 AM	✓	✓	normal	60	fileSystem:/opt	✎

Total Rows: 1 (selected: 0)

15.5. VIEWING SNAPSHOTS AND CHANGES

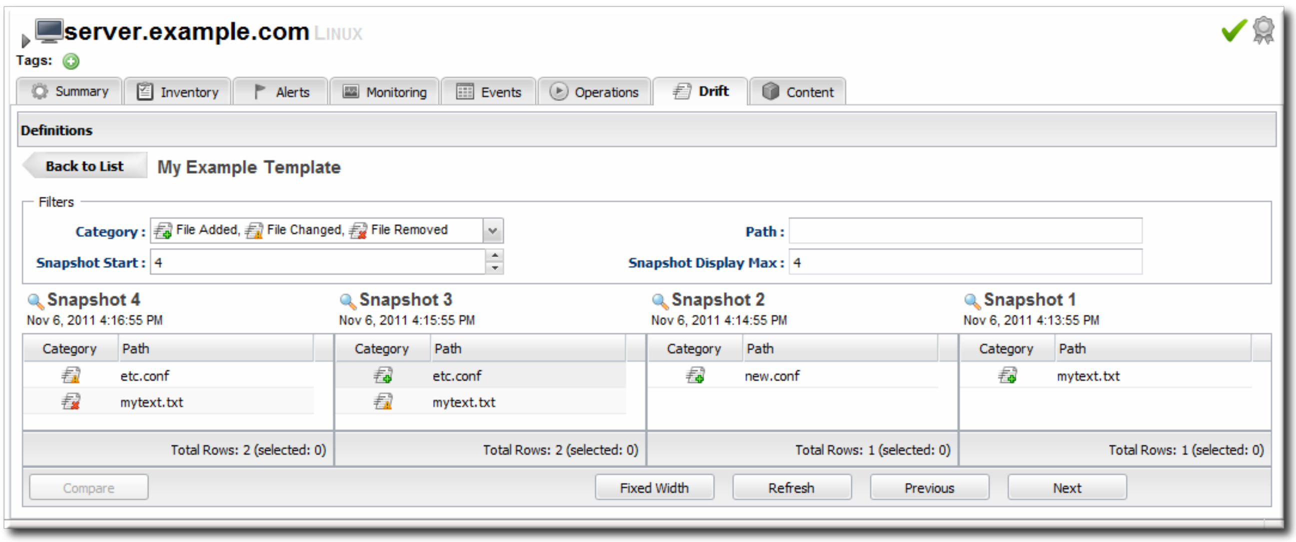
NOTE

The initial snapshot is *snapshot 0*. The snapshots in the carousel begin at version 1 — meaning it begins at the first change, not the initial file set.

If a snapshot is pinned, so that it is set as a baseline, then it is not displayed in the carousel because it is snapshot 0. However, it can be viewed by clicking the pinned icon in the definition list.

15.5.1. Viewing the Snapshot Carousel

Snapshots for a drift definition are displayed in a horizontal stream of windows, starting with the most recent change. This is colloquially called a *carousel*, because it is a rotating view of snapshots.



Category : File Added, File Changed, File Removed

Path :

Snapshot Start : 4

Snapshot Display Max : 4

Snapshot 4
Nov 6, 2011 4:16:55 PM

Category	Path
⚠	etc.conf
✖	mytext.txt

Total Rows: 2 (selected: 0)

Snapshot 3
Nov 6, 2011 4:15:55 PM

Category	Path
✚	etc.conf
⚠	mytext.txt

Total Rows: 2 (selected: 0)

Snapshot 2
Nov 6, 2011 4:14:55 PM

Category	Path
✚	new.conf

Total Rows: 1 (selected: 0)

Snapshot 1
Nov 6, 2011 4:13:55 PM

Category	Path
✚	mytext.txt

Total Rows: 1 (selected: 0)

Figure 15.3. Viewing Snapshots

To open the carousel:

1. Click the **`Inventory`** tab in the top menu.

2. Search for the resource.

3. Click the **`Drift`** tab for the resource.

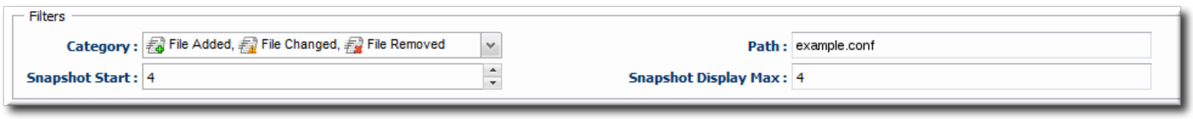
4. Click the name of the drift definition.

5. The snapshot carousel shows, by default, the four most recent snapshots.

6. Optionally, filter the snapshots to view. There are two elements that can be used to search for snapshots:

    - The change type within the snapshot, whether a file was added, deleted, or modified.

    - The path of a change within the snapshot. This path filter is a substring filter based on the paths and files in the drift entries.



Category : File Added, File Changed, File Removed

Path : example.conf

Snapshot Start : 4

Snapshot Display Max : 4

There can be slight differences in the way that changes are recorded in snapshots if the definition is pinned. The most obvious is that if a new file is added, it will show up as a new file in every subsequent snapshot because it is always compared against the pinned snapshot, where the file does not exist. Likewise, if a file is deleted, it is listed in every snapshot as deleted.

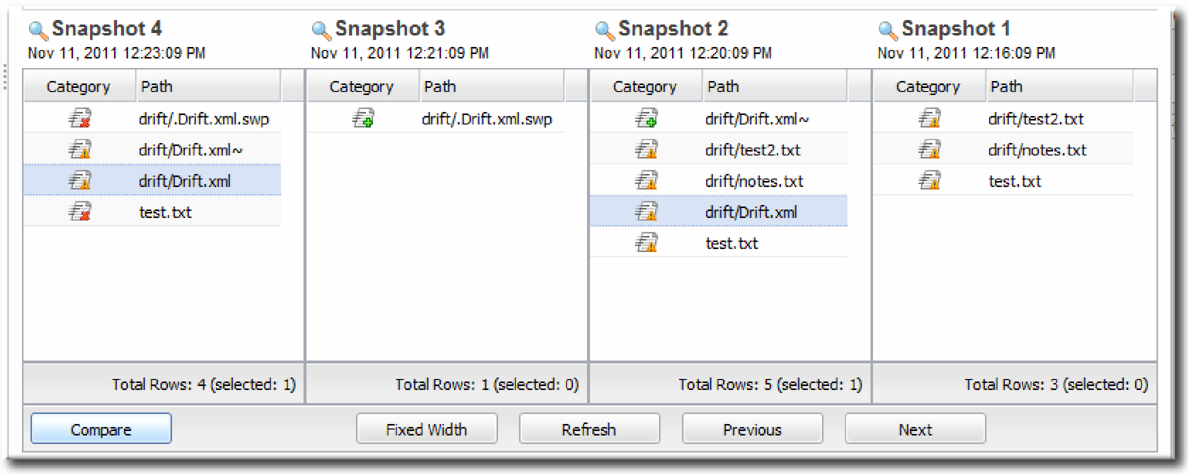
15.5.2. Comparing Drift Changes

Changes are diffed at the file level, not the full snapshot level. Administrators can view the specific changes made between versions on the selected files.

NOTE

Only changes for text files can be compared. Drift detection will identify binary files that have changed and show a timestamp and SHA, but it does not display the binary file contents or diff changes between versions of a binary file.

1. Click the `Inventory` tab in the top menu.
2. Search for the resource.
3. Click the `Drift` tab for the resource.
4. Click the name of the drift definition.
5. Click the names of the files to compare.



Snapshot 4
Nov 11, 2011 12:23:09 PM

Category	Path
	drift/.Drift.xml.swp
	drift/Drift.xml~
	drift/Drift.xml
	test.txt

Total Rows: 4 (selected: 1)

Snapshot 3
Nov 11, 2011 12:21:09 PM

Category	Path
	drift/.Drift.xml.swp

Total Rows: 1 (selected: 0)

Snapshot 2
Nov 11, 2011 12:20:09 PM

Category	Path
	drift/Drift.xml~
	drift/test2.txt
	drift/notes.txt
	drift/Drift.xml
	test.txt

Total Rows: 5 (selected: 1)

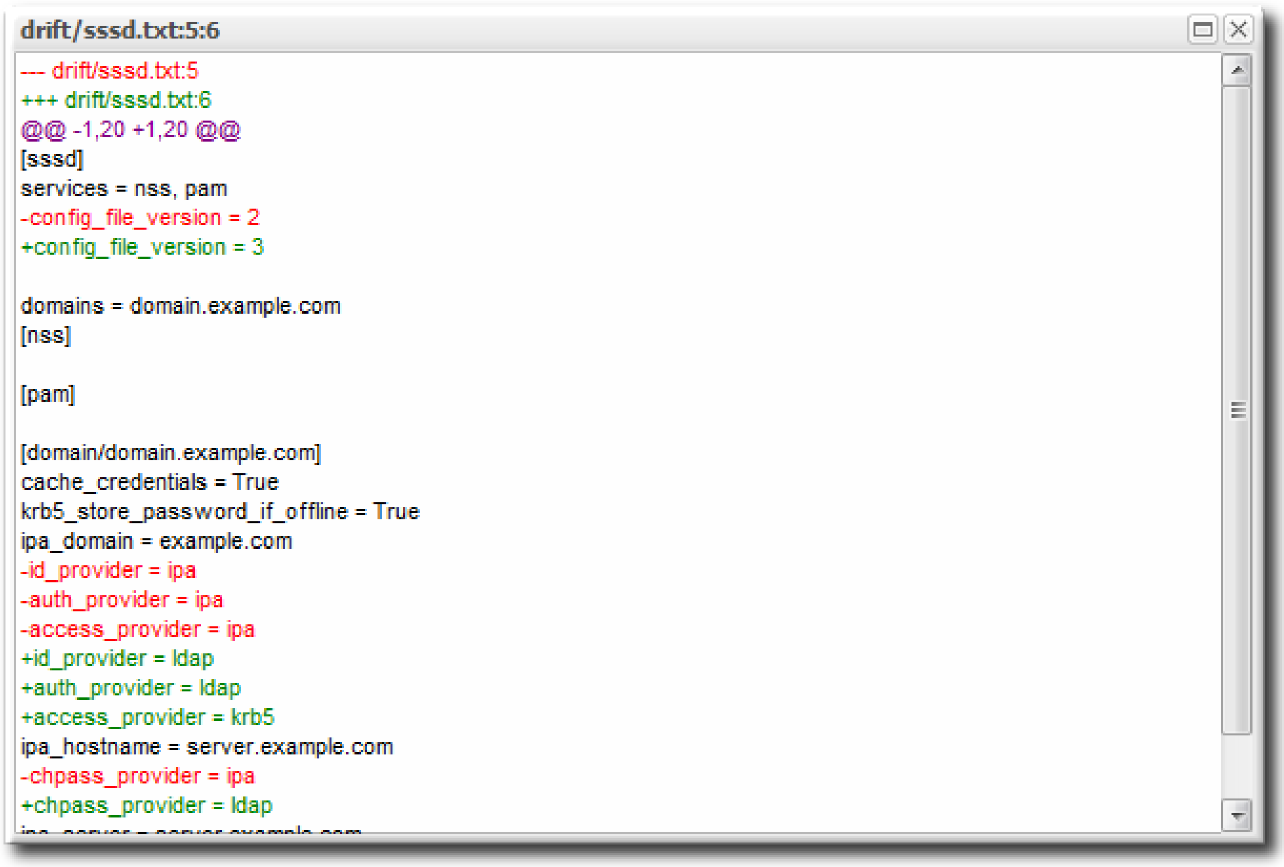
Snapshot 1
Nov 11, 2011 12:16:09 PM

Category	Path
	drift/test2.txt
	drift/notes.txt
	test.txt

Total Rows: 3 (selected: 0)

6. Click `Compare`.

The diff uses standard text formatting for displaying file diffs.



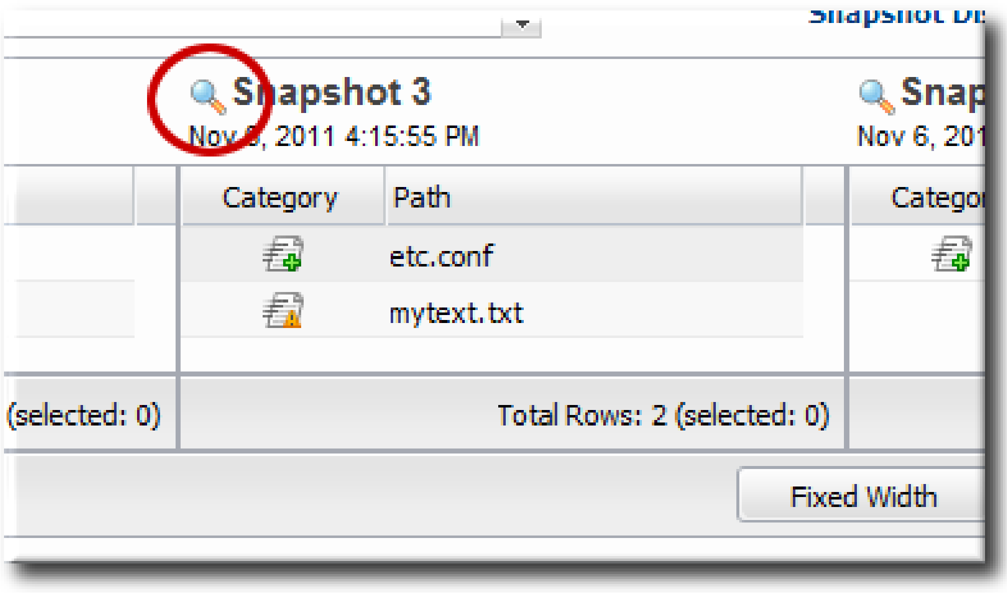
drift/sssd.txt:5:6

```
--- drift/sssd.txt:5
+++ drift/sssd.txt:6
@@ -1,20 +1,20 @@
[sssd]
services = nss, pam
-config_file_version = 2
+config_file_version = 3

domains = domain.example.com
[nss]

[pam]

[domain/domain.example.com]
cache_credentials = True
krb5_store_password_if_offline = True
ipa_domain = example.com
-id_provider = ipa
-auth_provider = ipa
-access_provider = ipa
+id_provider = ldap
+auth_provider = ldap
+access_provider = krb5
ipa_hostname = server.example.com
-chpass_provider = ipa
+chpass_provider = ldap
ipa_server = server.example.com
```

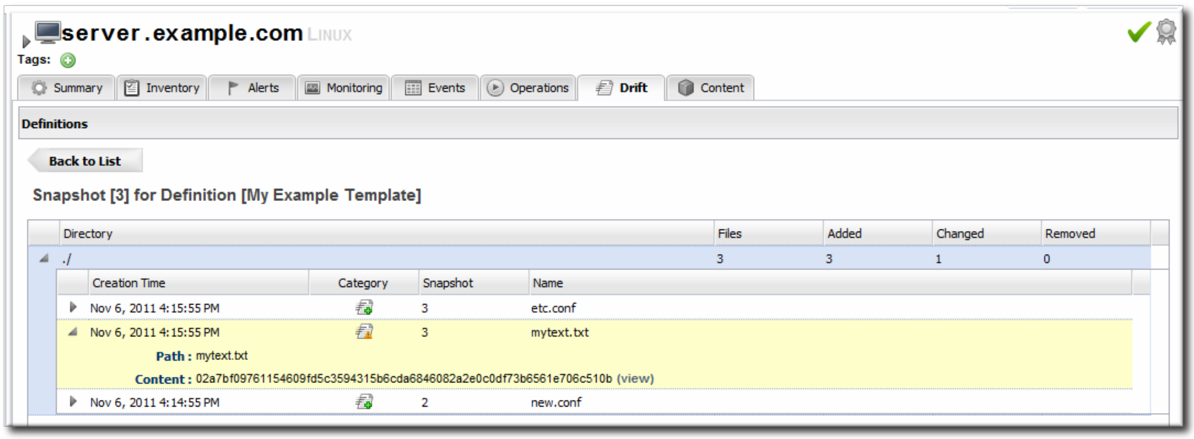
Figure 15.4. Change Set Diffs

15.5.3. Viewing Snapshot Details

1. Click the **`Inventory`** tab in the top menu.
2. Search for the resource.
3. Click the **`Drift`** tab for the resource.
4. Click the name of the drift definition.
5. In the snapshot carousel, click the magnifying glass by the name of the snapshot to view.



6. Expand the directory to show the list of changes for that snapshot.



Snapshot [3] for Definition [My Example Template]

Directory	Files	Added	Changed	Removed
./	3	3	1	0

Creation Time	Category	Snapshot	Name
Nov 6, 2011 4:15:55 PM		3	etc.conf
Nov 6, 2011 4:15:55 PM		3	mytext.txt

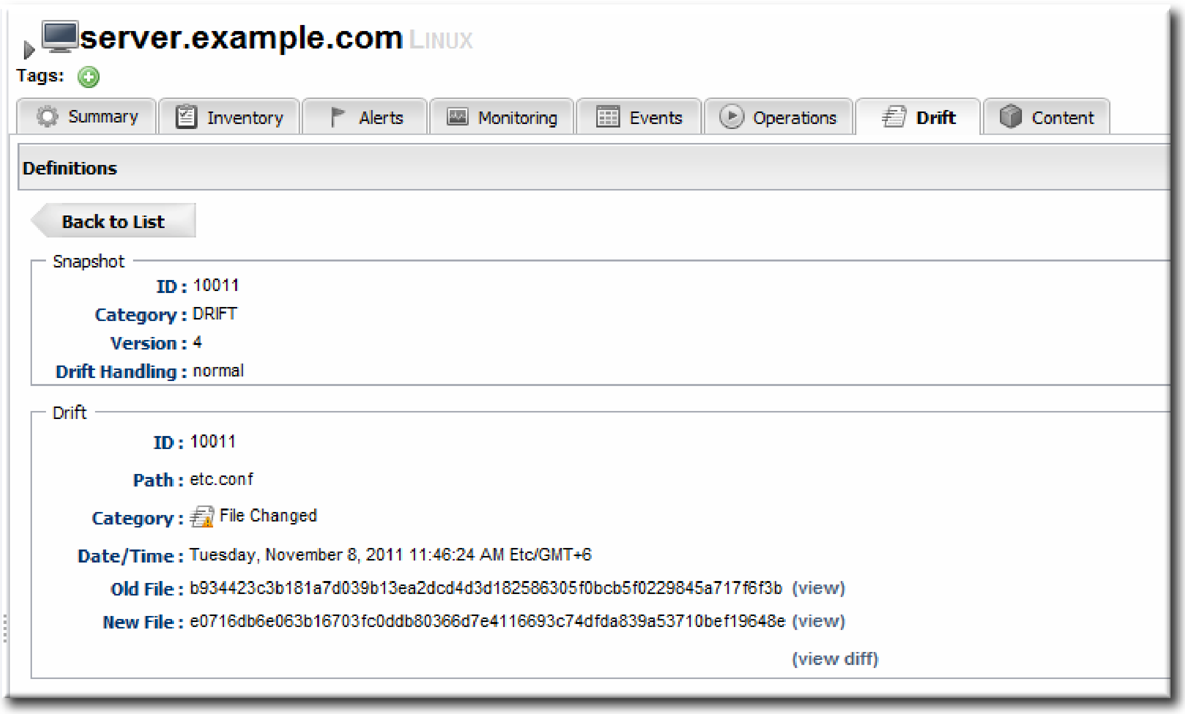
Path : mytext.txt

Content : 02a7bf09761154609fd5c3594315b6cda6846082a2e0c0df73b6561e706c510b (view)

Creation Time	Category	Snapshot	Name
Nov 6, 2011 4:14:55 PM		2	new.conf

7. To see the details of a specific change, click the `(view)` link.

8. The details for that file shows links to display the immediate previous version of the file, the changed version of the file, and a diff between the two.



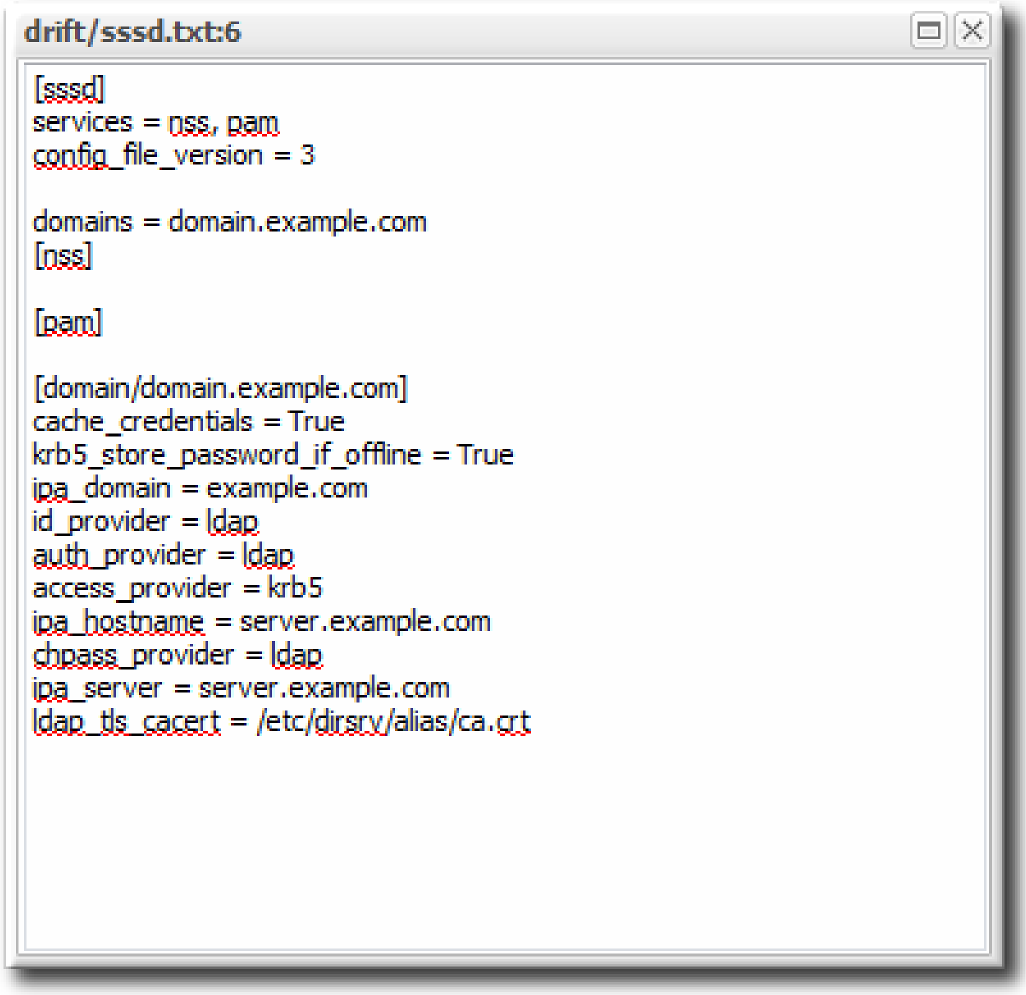
Snapshot

ID : 10011
Category : DRIFT
Version : 4
Drift Handling : normal

Drift

ID : 10011
Path : etc.conf
Category : File Changed
Date/Time : Tuesday, November 8, 2011 11:46:24 AM Etc/GMT+6
Old File : b934423c3b181a7d039b13ea2dcd4d3d182586305f0bcb5f0229845a717f6f3b (view)
New File : e0716db6e063b16703fc0ddb80366d7e4116693c74dfda839a53710bef19648e (view)
(view diff)

When clicking the **view** link, the page title has the version number along with the file name. For example, when viewing version 6 of **myfile.txt**, the title is *myfile.txt:6*.


drift/sssd.txt:6

```
[sssd]
services = nss, pam
config_file_version = 3

domains = domain.example.com
[nss]

[pam]

[domain/domain.example.com]
cache_credentials = True
krb5_store_password_if_offline = True
ipa_domain = example.com
id_provider = ldap
auth_provider = ldap
access_provider = krb5
ipa_hostname = server.example.com
chpass_provider = ldap
ipa_server = server.example.com
ldap_tls_cacert = /etc/dirsrv/alias/ca.crt
```

15.5.4. Seeing Drift Events in the Timeline

Whenever drift is detected, it shows up as an event in the events timeline for the resource.

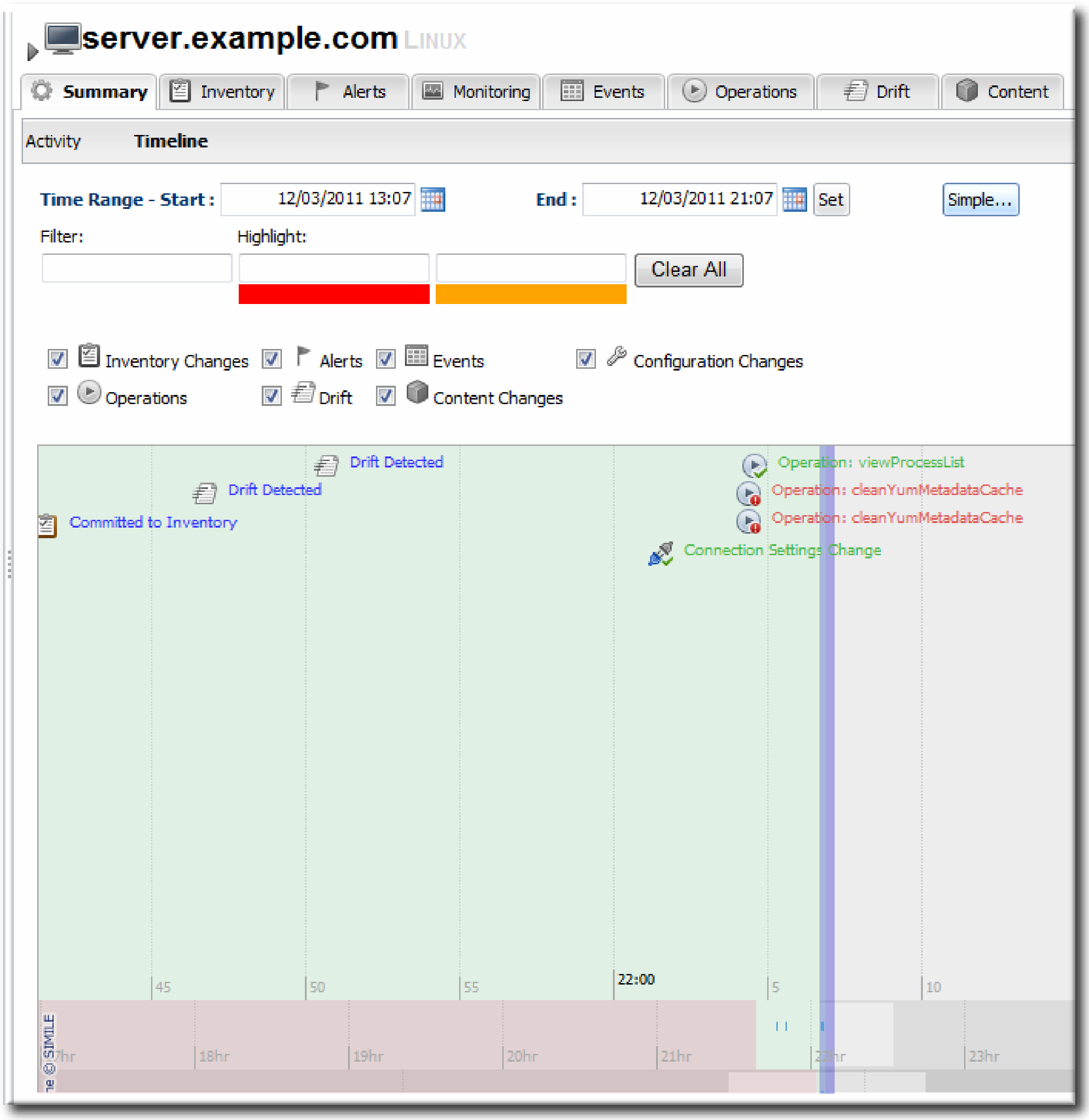
1. Click the **`Inventory`** tab in the top menu.

2. Search for the resource.

3. In the **`Summary`** tab, click the **`Timeline`** subtab.

4. The detection runs where drift was detected show up in the timeline as **`Drift Detected`**. To see only drift events in the timeline, clear all but the **`Drift`** checkbox.

   The time interval can be reset to adjust the span of the timeline.



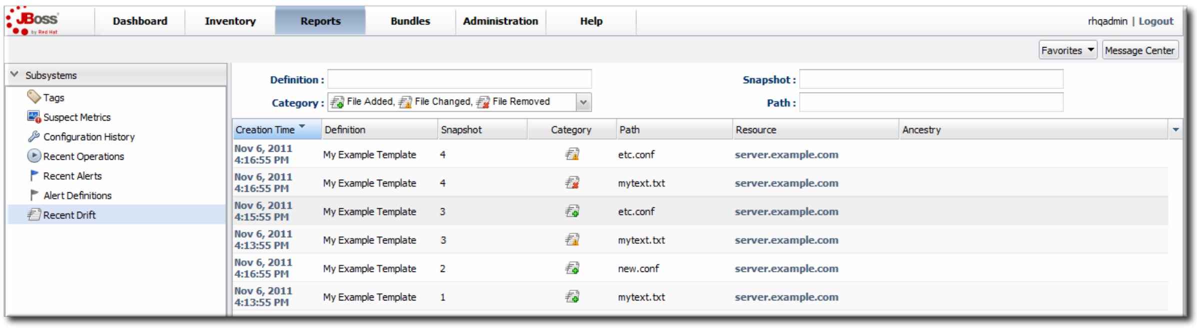
15.5.5. Checking Drift Snapshot Reports

The snapshot carousel (Section 15.5.1, “Viewing the Snapshot Carousel”) shows all of the snapshots for a single drift definition on a single resource. To view a list of all snapshots, for all definitions across all resources, check the Recent Drift Report.

1. Click the **`Reports`** tab in the top navigation menu.

2. Select the **`Recent Drift`** report from the **`Subsystems`** report list.

3. Every drift instance is listed, sorted by the snapshot creation time.



Definition :
Category : File Added, File Changed, File Removed
Snapshot :
Path :

Creation Time	Definition	Snapshot	Category	Path	Resource	Ancestry
Nov 6, 2011 4:16:55 PM	My Example Template	4	⚠	etc.conf	server.example.com	
Nov 6, 2011 4:16:55 PM	My Example Template	4	✗	mytext.txt	server.example.com	
Nov 6, 2011 4:15:55 PM	My Example Template	3	✚	etc.conf	server.example.com	
Nov 6, 2011 4:13:55 PM	My Example Template	3	⚠	mytext.txt	server.example.com	
Nov 6, 2011 4:16:55 PM	My Example Template	2	✚	new.conf	server.example.com	
Nov 6, 2011 4:13:55 PM	My Example Template	1	✚	mytext.txt	server.example.com	

4. Optionally, filter the list of drift changes. There are four filter options:

   - The definition name
   - The snapshot number (which crosses drift definitions)
   - The change type within the snapshot, whether a file was added, deleted, or modified.
   - The path of a change within the snapshot. This path can be a directory, a specific file name, or a search expression.

NOTE

Reports can be exported to CSV, which can be used for office systems or further data manipulation.

Only the information displayed for the report is exported. If the **`Recent Drift Report`** is filtered by date, definition, snapshot or version, or category, only the matching operations are included in the report.

To export a report, simply click the **`Export`** button. The report will automatically be downloaded as **`recentDrift.csv`**.

15.6. PINNING SNAPSHOTS AND MANAGING COMPLIANCE

As discussed in Section 15.1.2, “Snapshots, Deltas, and Baseline Images” , a specific snapshot, with its complete current fileset, can be associated or *pinned* to a drift definition. Pinning a snapshot creates an entirely new style of drift definition. Rather than simply tracking changes, a pinned snapshot allows an administrator to establish a clear, blessed configuration for a system or application. It sets a standard with which the system configuration should comply.

15.6.1. More About Pinning Snapshots

A snapshot is a picture of the actual, current files that are on a specific resource. A snapshot is a real-world view. In normal drift conditions, each snapshot is compared to the one immediately before it to show changes. However, it is possible to select a specific snapshot as a fixed baseline to compare changes against. This is a *pinned snapshot*.

A drift definition sets the rules for running drift detection, but it does not add or define or overwrite any files on a resource. A drift definition does not define content or contain a file set. Content has to be added to a definition (or a definition template). A file set (a snapshot) has to be manually added to the drift definition, after the snapshot exists. This is pinning. Pinning takes a real, existing set of files from a snapshot and links it to a drift definition on a resource or a drift definition template.

Pinning is one method that administrators can use to standardize resource configuration. An administrator can use a single resource as a test box to get a resource's configuration tuned to its ideal settings. Then, that file set can be pinned to a template and re-applied to other resources of the same type. Because the pinned snapshot is based on a real resource, administrators can be confident that the configuration is realistic and functional.

Pinning a snapshot alters some fundamental behaviors with drift management in JBoss ON:

- It removes any snapshots that were created before that snapshot. For example, if an administrator decides to pin Snapshot 7, Snapshot 0 (the initial image) through Snapshot 6 are all deleted, and Snapshot 7 becomes the new Snapshot 0.
- It creates a baseline image that every change is compared against rather than keeping a moving tally of changes.
- It changes the behavior of drift alerts (Section 15.8, “Defining Drift Alerts”) so that alerts are sent continually until the system configuration is back in compliance with the pinned snapshot.
- The definition it is pinned to cannot be deleted until the snapshot is unpinned.
- If a snapshot is pinned to a template, then all of the resource-level definitions attached to that template automatically use the pinned snapshot as their baseline.
- Any new file added after a snapshot is pinned (or any file deleted) is going to be reported as a new file in every subsequent snapshot. This is because the new snapshot is always compared against the baseline snapshot, so the file is always new to the baseline.

  There is some logic to prevent drift from reporting the same change incessantly. If `file1.txt` is added, the agent creates snapshot 1. When the agent does its next detection run, it recognizes that `file1.txt` is not in the baseline, but as long as the SHA for `file1.txt` has not changed, the agent does not report it as new drift and does not take a new snapshot. If `file1.txt` is modified, however, the agent notices the new SHA and sends a new snapshot — with the modified `file1.txt` still listed as a new file, because it is compared against the baseline, not the previous version.

15.6.2. When to Pin to a Resource and When to Pin to a Template

When a snapshot perfectly matches the configuration that an administrator desires, it can be associated with a drift definition. That snapshot can be pinned to a resource-level definition or a definition template, and there are slightly different reasons to do one or the other.

- Pinning a snapshot to a resource-level definition establishes a baseline for that resource alone. This makes sense while you are still developing an ideal baseline image or for unique environments that may not transition over to other resources.

  Pinning to a resource definition allows a lot of flexibility. It is easy to pin and unpin and select a new snapshot as the baseline, to let administrators develop an ideal configuration with a minimal impact on drift events, alerting, and monitoring because the changes are contained.
- Pinning a snapshot to a template means that baseline can be applied to every resource that uses that template; it allows that one single snapshot to be used across multiple resources. This is makes sense for any kind of repeatable configuration areas and for production or critical systems which must have consistent configuration.

Pinning to a template is very powerful for maintaining consistency across an entire infrastructure once an ideal configuration has been developed.

Pinning always takes a snapshot that was created on a specific resource and then promotes it to be the baseline for that definition. So the question is — why does a resource-level snapshot need to be pinned to a template? Why can't a template create and use its own snapshot?

The key is to remember that a drift definition template is associated with a resource *type*. The template is not defined as part of a specific resource.

For a resource-level drift definition, the very first drift detection run creates an initial snapshot based on real and existing files. That initial snapshot can be automatically applied as the baseline, pinned snapshot or any snapshot after the initial can be used as the baseline.

However, a drift definition template (Section 15.3.1, “About Resources and Drift Definition Templates”) is not associated with a resource. Therefore, templates do not have a real set of files to work with and it never has its own snapshots to use. The only way that a drift template can be associated with a snapshot is if a resource-level snapshot is pinned to the template.

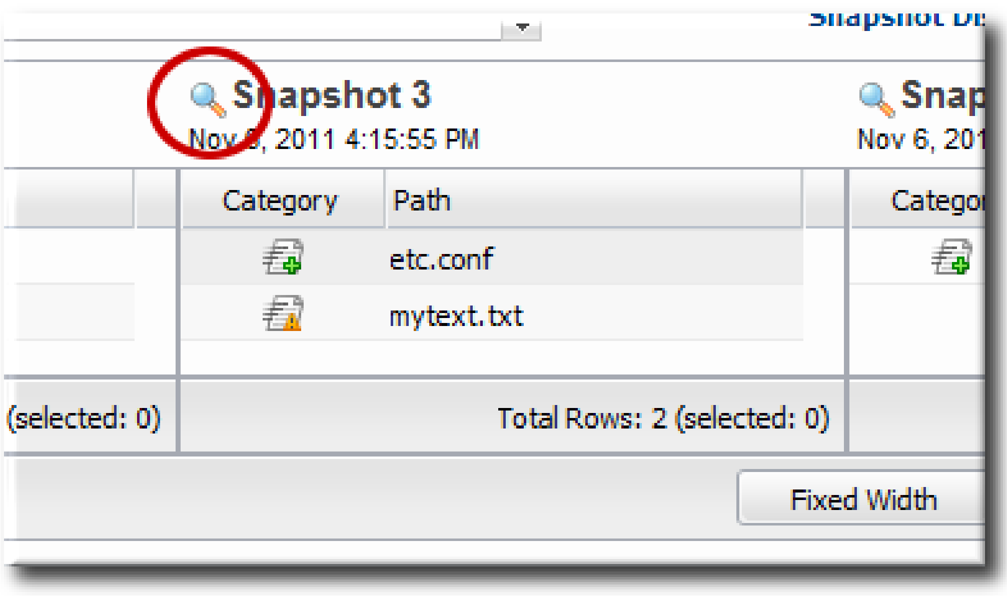
In a sense, pinning a snapshot has a backward workflow from defining a drift definition. A definition starts with a template, then moves to a resource-level definition, which generates a snapshot of that resource. Pinning always begins with a snapshot on a resource, and then moves up to a definition or a definition template.

NOTE

A drift definition sets a very clear and limited set of criteria to use for drift detection. When a snapshot is associated with a drift definition template, the template must use the same settings as the original resource-level drift definition which generated the snapshot. If a matching template does not exist, then a new template can be created, using those criteria.

15.6.3. Pinning to a Resource-Level Definition

1. Click the `Inventory` tab in the top menu.
2. Search for the resource.
3. Click the `Drift` tab.
4. Click the name of the drift definition.
5. In the snapshot carousel, click the magnifying glass by the name of the snapshot to pin.

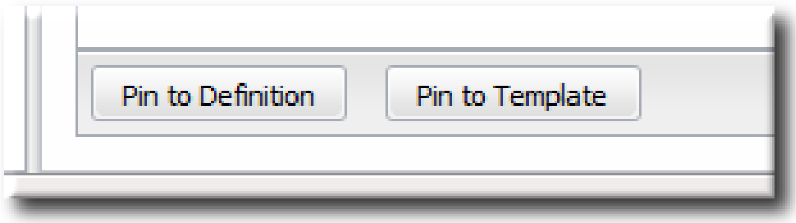


NOTE

The initial snapshot is not displayed in the carousel. To pin the initial snapshot, click the thumbtack icon in the **`Pinned`** column of the drift definition list. That opens the initial snapshot.

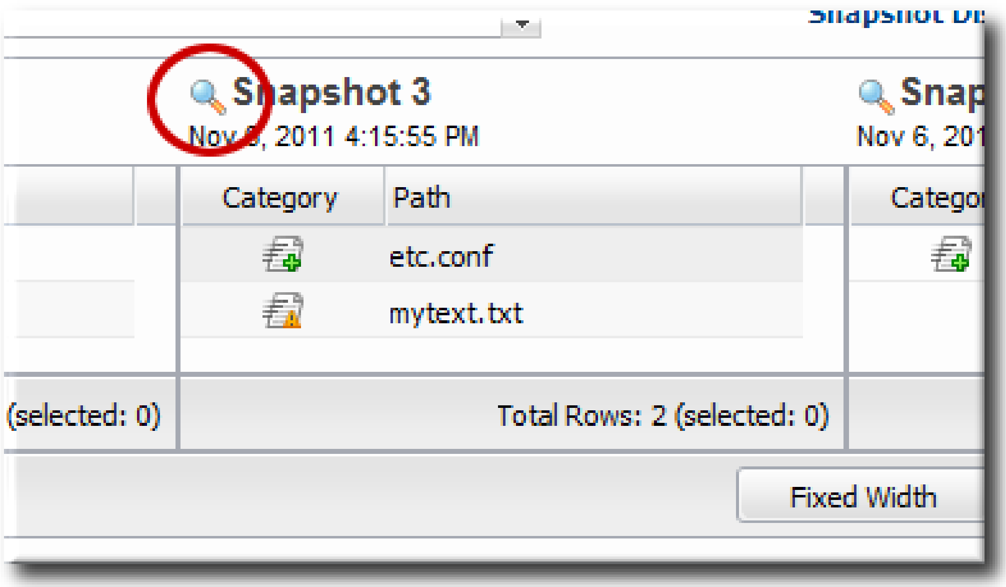
If a snapshot has already been pinned, then clicking the thumbtack icon opens the pinned snapshot.

6. At the bottom of the change list, click the **`Pin to Definition`** button.



15.6.4. Pinning to a Template

1. Click the **`Inventory`** tab in the top menu.
2. Search for the resource.
3. Click the **`Drift`** tab.
4. Click the name of the drift definition.
5. In the snapshot carousel, click the magnifying glass by the name of the snapshot to pin.

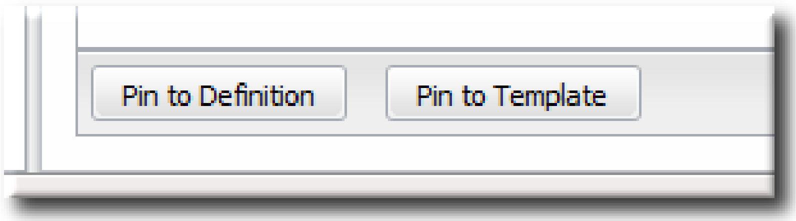


NOTE

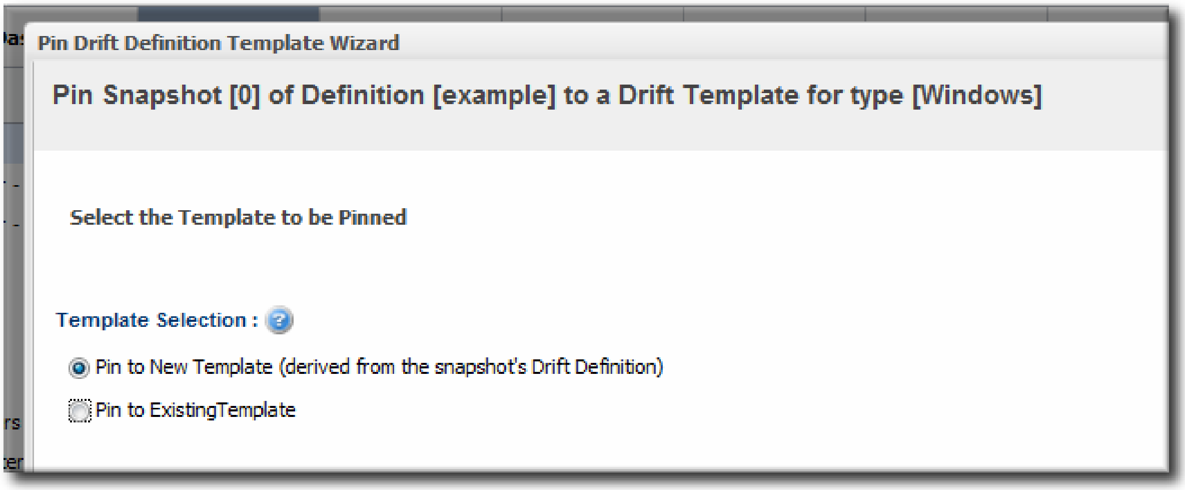
The initial snapshot is not displayed in the carousel. To pin the initial snapshot, click the thumbtack icon in the **`Pinned`** column of the drift definition list. That opens the initial snapshot.

If a snapshot has already been pinned, then clicking the thumbtack icon opens the pinned snapshot.

6. At the bottom of the change list, click the **`Pin to Template`** button.



7. If the resource-level template is based on or attached to an existing template, then you can associate the snapshot with that existing template. If the base directory for the resource-level snapshot does not match any existing drift template, then you must create a new template.



Pin Drift Definition Template Wizard

Pin Snapshot [0] of Definition [example] to a Drift Template for type [Windows]

Select the Template to be Pinned

Template Selection :

[✓ Pin to New Template (derived from the snapshot's Drift Definition)]

Pin to ExistingTemplate

8. Create the drift template, as in Section 15.3, “Creating a Drift Definition Template” .

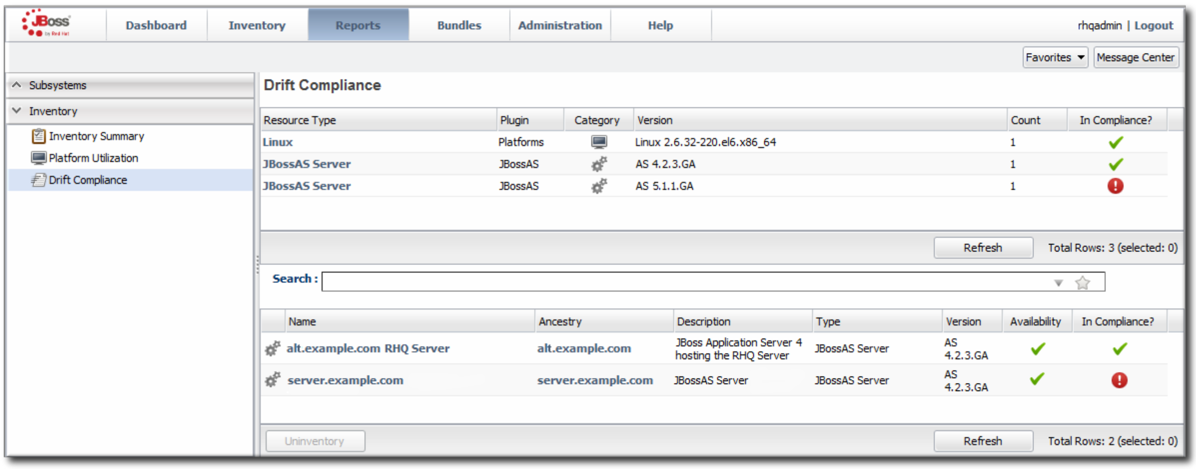
15.6.5. Checking Drift Compliance Reports

The compliance report is a variant of an inventory report. It lists all resources which currently have a drift definition configured and then shows whether they are compliant. Compliance is cumulative; if a resource has multiple drift definitions and is noncompliant on a single one, it will show as non-compliant in the report.

1. Click the **`Reports`** tab in the top navigation menu.

2. Select the **`Drift Compliance`** report from the **`Inventory`** report list.

3. Every resource with a drift definition is listed **by type** and with an icon to indicate whether it is compliant (✓) or non-compliant (❗).



Drift Compliance

Resource Type	Plugin	Category	Version	Count	In Compliance?
Linux	Platforms		Linux 2.6.32-220.el6.x86_64	1	✓
JBossAS Server	JBossAS		AS 4.2.3.GA	1	✓
JBossAS Server	JBossAS		AS 5.1.1.GA	1	❗

Total Rows: 3 (selected: 0)

Name	Ancestry	Description	Type	Version	Availability	In Compliance?
alt.example.com RHQ Server	alt.example.com	JBoss Application Server 4 hosting the RHQ Server	JBossAS Server	AS 4.2.3.GA	✓	✓
server.example.com	server.example.com	JBossAS Server	JBossAS Server	AS 4.2.3.GA	✓	❗

Total Rows: 2 (selected: 0)

4. To get information about the specific resources, click the resource type name; this opens a second inventory report under the main report. All of the resources of that type are listed with their compliance state.

NOTE

Reports can be exported to CSV, which can be used for office systems or further data manipulation.

To export a report, simply click the **`Export`** button. The report will automatically be downloaded as **`driftCompliance.csv`**.

15.6.6. Unpinning a Snapshot

A snapshot can be unpinned — or disassociated — from either a resource-level definition or a drift template. Unpinning a snapshot moves the definition back to a rolling drift detection mode, and any resources that were out of compliance are no longer marked as non-compliant.

1. Click the **`Inventory`** tab in the top menu.

2. Search for the resource.

3. Click the **`Drift`** tab.

4. Click the pin icon for the drift definition.

15.7. *EXTENDED EXAMPLE:* DEFINING REQUIRED EAP CONFIGURATION

The Setup

Tim the IT Guy at Example Corp. has one EAP server running in his production environment. Because of the production load, the EAP server was routinely running out of memory, which was degrading its performance and causing downtime for Example Corp.'s website.

To resolve his immediate memory problem, all Tim has to do is change the heap size setting for his EAP instance. However, Tim needs another strategy for managing the configuration long-term. If he adds another production EAP instance or deploys a new one to replace his current one, it is going to hit the same memory-related performance problems without the new heap size setting.

What to Do

There are three things that Tim wants to accomplish to maintain his EAP performance:

- *Find a way to consistently apply configuration to EAP instances.*

  He defines a template for JBoss EAP instances (Section 15.3, "Creating a Drift Definition Template"). To maintain consistency, the template sets the **`Attach to template`** value to true, and each resource-level drift definition will preserve that settings. This ensures that any changes to the template are automatically applied to the JBoss resource drift definitions.

- *Use his current production settings as a basis for future EAP instances.*

  He pins his latest snapshot, with the higher heap settings, to the template definition (Section 15.6.4, "Pinning to a Template"). Every EAP instance is going to be compared against that baseline, so any with the wrong heap setting will immediately be marked out of compliance.

- *Be made aware of specific differences between his current EAP settings and his preferred settings.*

  He creates an alert definition (Section 15.8, "Defining Drift Alerts") which specifically targets the **`bin/run.conf`** file. This way, he knows precisely whether the heap settings and other JVM settings are wrong for his new instance. He can even use alerts to gather more information about how his EAP instance configuration is different, like using a CLI script to compare the current EAP configuration against the pinned snapshot and then send him the diff.

Expected Results

Tim brings a new server online, with a new EAP instance for the production environment. He applies the drift template to the new resource and, within a few minutes, receives a notification that his **`run.conf`** file is not compliant with his preferred configuration. He changes the heap settings on the new EAP instance without having to wait for performance degradation to remember the change.

15.8. DEFINING DRIFT ALERTS

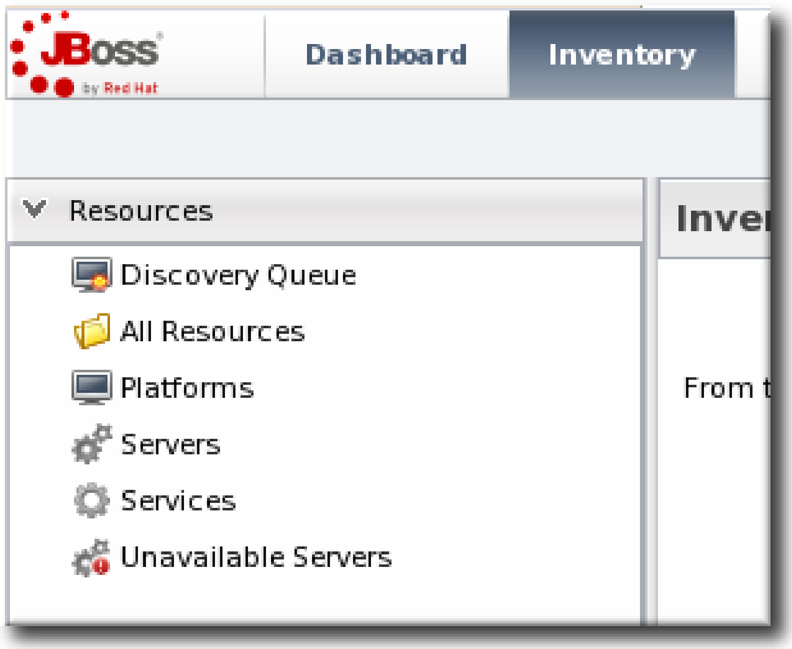
Drift changes have their own alert condition.

NOTE

Recovery alerts are not supported for drift.

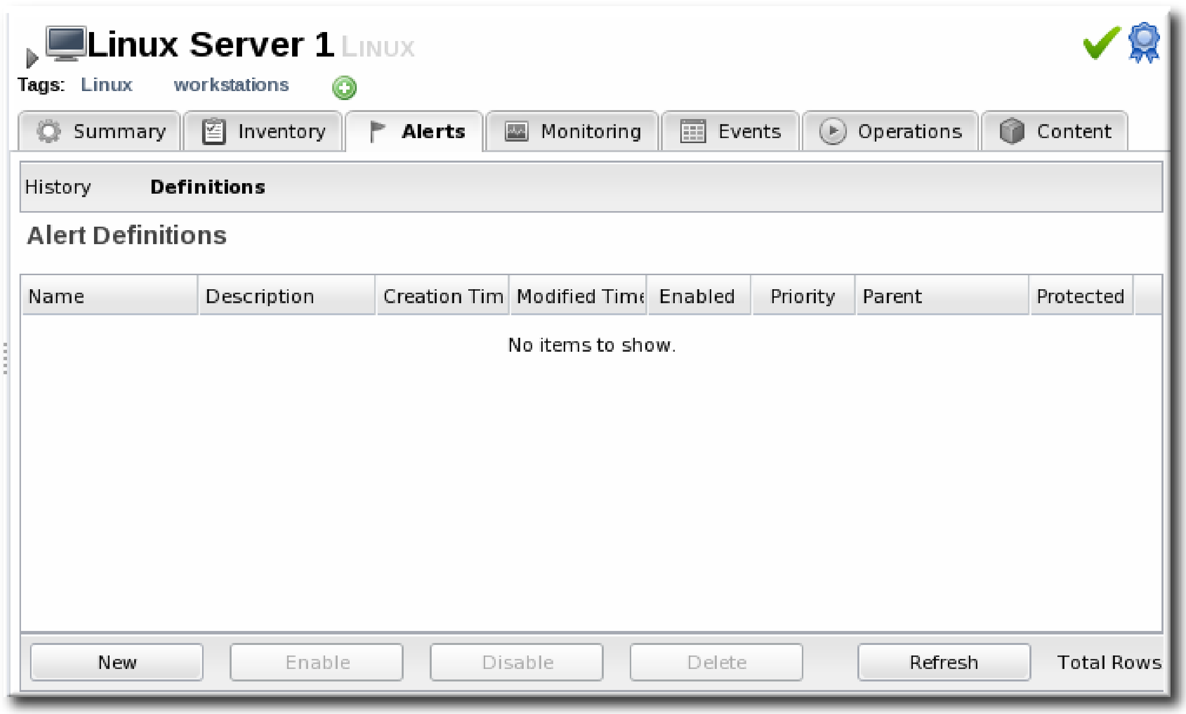
1. Click the **Inventory** tab in the top menu.

2. Select the resource type in the **Resources** menu table on the left, and then browse or search for the resource.



3. Click the resource name in the list.

4. Click the **Alerts** tab for the resource.



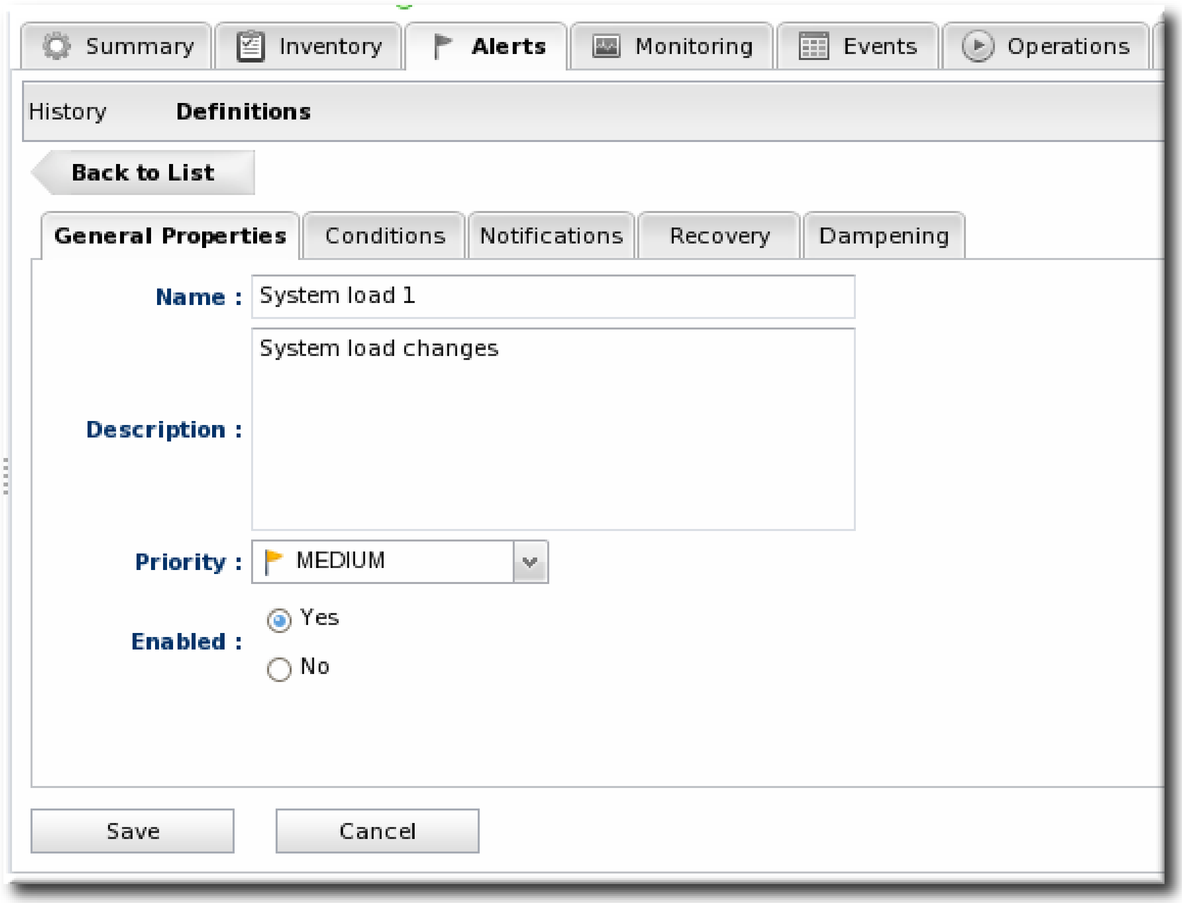
Alert Definitions

Name	Description	Creation Tim	Modified Time	Enabled	Priority	Parent	Protected

No items to show.

5. In the **Definitions** subtab, click the **New** button to create the new alert.

6. In the **General Properties** tab, give the basic information about the alert.



Name : System load 1

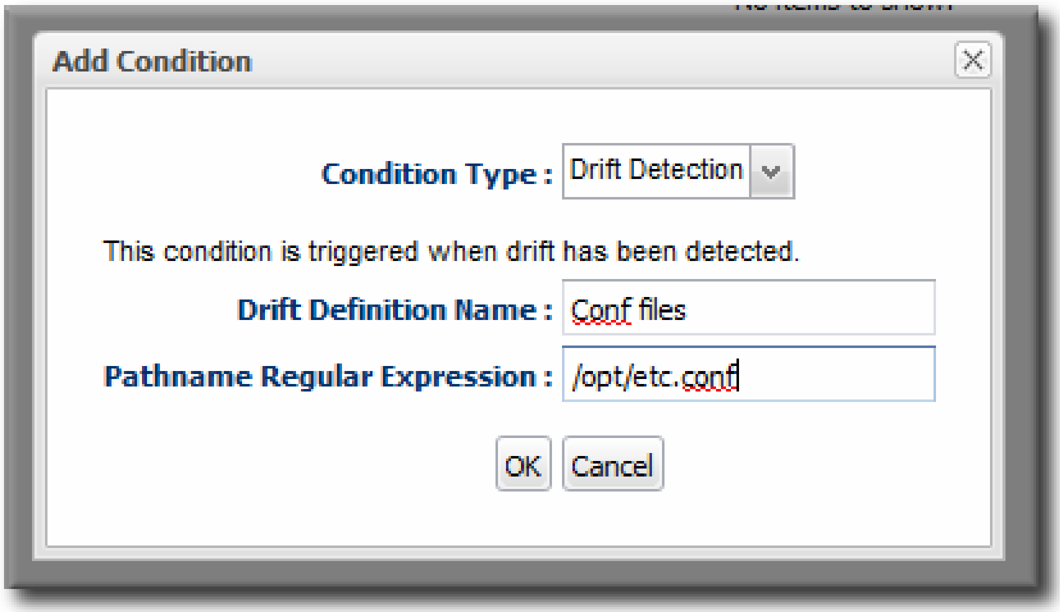
Description : System load changes

Priority : MEDIUM

Enabled : [✓ Yes] No

It may be useful to set a *Priority* if the drift definition contains critical configuration files.

7. In the **Conditions** tab, select the **`Drift Detection`** option from the conditions list. To use the alert for all drift changes, leave the fields blank. Otherwise, enter the specific drift definition name and (optionally) the directories or files that must be modified for the alert to be triggered.



Condition Type : Drift Detection

This condition is triggered when drift has been detected.

Drift Definition Name : Conf files

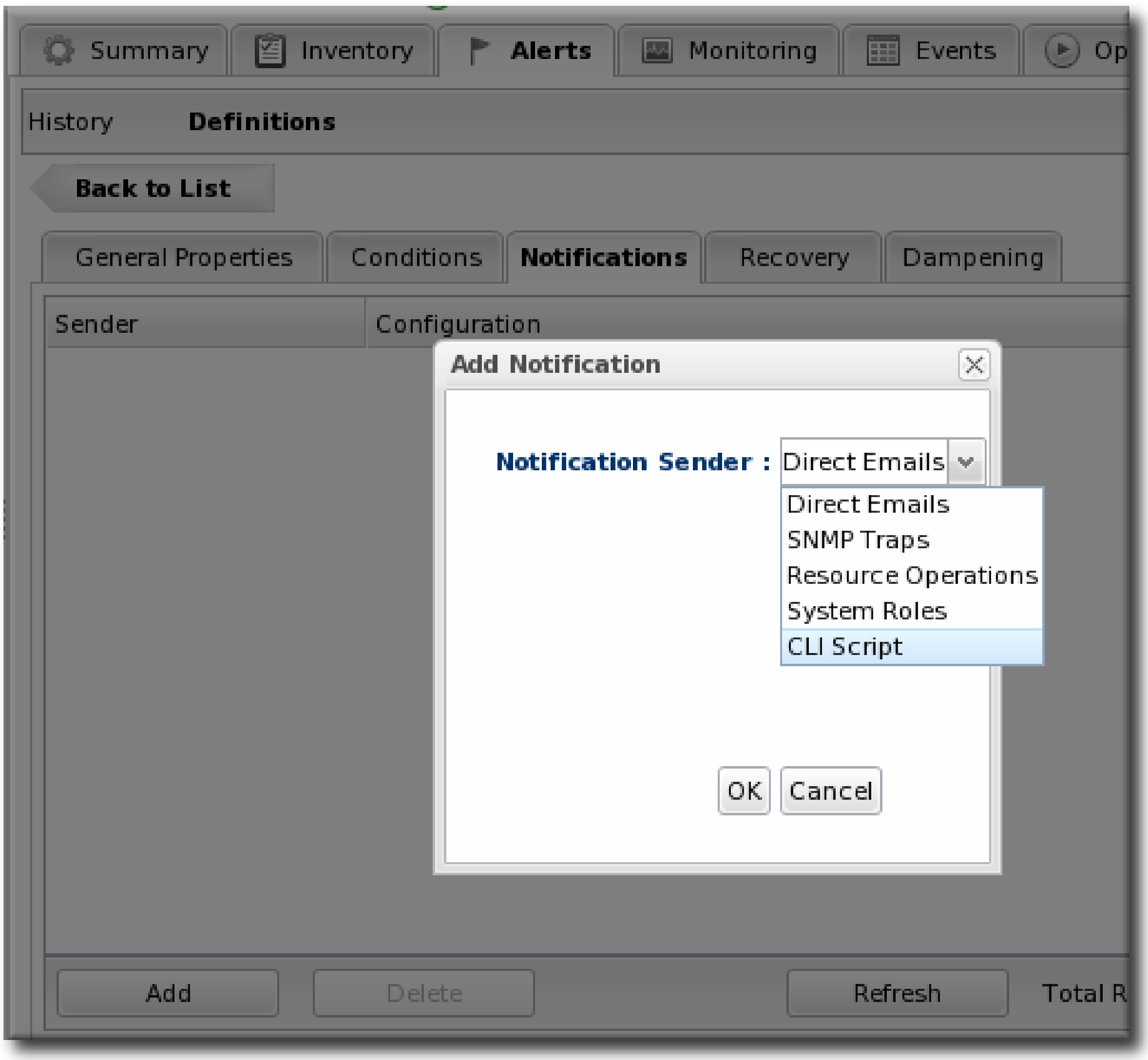
Pathname Regular Expression : /opt/etc.conf

NOTE

There can be more than one condition set to trigger an alert, meaning that you can use the same alert for multiple drift definitions or files.

8. In the **`Notifications`** tab, click **Add** to set a notification for the alert.

   Select the method to use to send the alert notification in the *Sender* option, and fill in the required information.



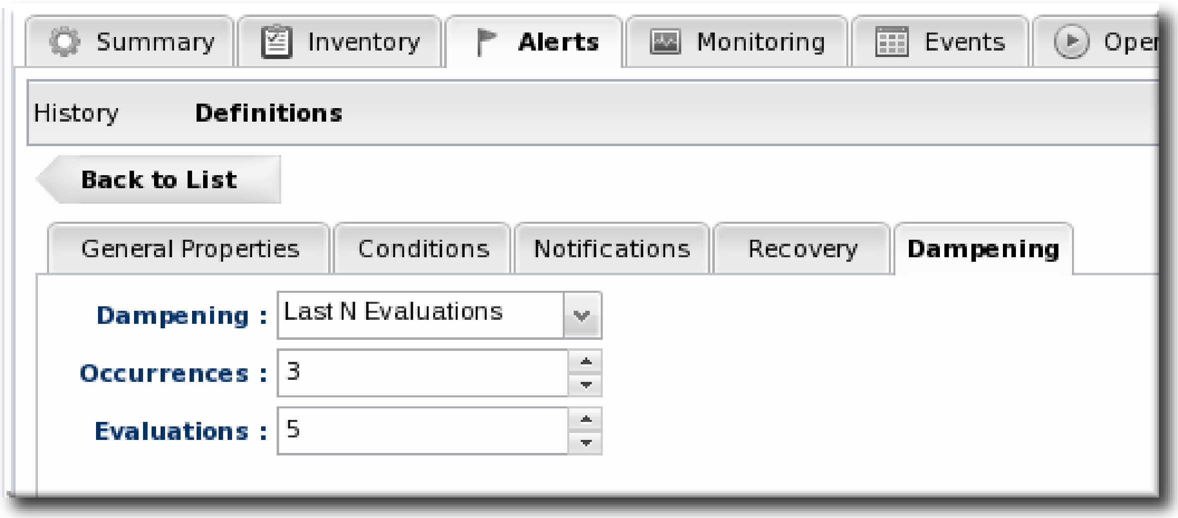
   The *Sender* option first sets the specific type of alert method (such as email or SNMP) and then opens the appropriate form to fill in the details for that specific method.

9. Optionally, in the **`Dampening`** tab, give the dampening (or frequency) rule on how often to send notifications for drift.

NOTE

For pinned snapshots, it can be useful to use dampening rules to keep from getting a flood of alerts before a drift problem is remedied.

Dampening only makes sense for a definition with a pinned snapshot. A pinned definition will fire alerts with every alert scan (every 10 minutes) for as long as it is out of compliance, even if there are no further changes. A rolling definition only fires an alert once, when drift is detected.



Dampening : Last N Evaluations
Occurrences : 3
Evaluations : 5

Any of the dampening rules can be used. The ultimate goal is to limit the number of times that the same alert is set for a resource that is out of compliance with a pinned definition. For example, *Time period* sets a limit on the number of times in a given time period that an alert is issued if the alert condition occurs. Setting the occurrence to 1 and the time period to 4 hours means that when drift is detected once, the server sends an alert and then waits another 4 hours before sending the next alert.

10. Click **OK** to save the alert definition.

15.9. *EXTENDED EXAMPLE:* REVERTING A JBOSS SERVER TO ITS ORIGINAL CONFIGURATION USING BUNDLES AND SERVER SCRIPTS

The Setup

In Section 15.7, "*Extended Example:* Defining Required EAP Configuration", Tim the IT Guy at Example Corp. set up drift templates and alerts to help manage the configuration on his production EAP servers. However, his resolution was done manually. When the drift alert notified him that his EAP server was out of compliance, he edited the `run.conf` directly to adjust the heap size.

Manual updates are fine for small infrastructures or infrequent changes. A better management tool, though, is to automate any remediation required for drift.

What to Do

The goal is to have JBoss ON respond intelligently to drift without requiring any action from Tim the IT Guy. There are two features that allow automated responses:

- Using *bundles* to provision updated files or applications. A bundle is a ZIP file that contains an Ant recipe and any required content (such as configuration files or JARs) for an application. JBoss ON can provision this content on a platform or a JBoss server in a specified directory.

- Launching JBoss ON CLI scripts in response to an alert. One of the possible alert notifications is a server-side alert sender. A JBoss ON CLI script is loaded as content and stored in the JBoss ON server; when the alert fires, it initiates the specified, stored CLI script.

There are a few steps to remediation using bundles and CLI scripts:

1. Create a bundle file based on the pinned snapshot configuration. The content of the bundle depends on the needs of the deployment. It can be specific configuration files, like **`bin/run.conf`**, or it can be a full EAP server.

   NOTE

   If the bundle contains the full EAP server, then it can be used to create the initial EAP server.

2. Deploy the bundle with the full EAP server to create the new EAP instance. (Or, if the bundle only has configuration files, create the EAP instances.)

3. Set up the drift definitions, based on the previously configured template (Section 15.7, "*Extended Example:* Defining Required EAP Configuration"), for the new EAP instance.

4. Create a JBoss ON CLI script (in JavaScript) that will automatically deploy the specified bundle to the appropriate destination. An example is in Example 15.2, "fix-eap.js Script"; in that script, replace the ***`destinationId`*** and ***`bundleVersionId`*** with the real ID numbers for the destination entry and bundle version entry in JBoss ON.

5. Create an alert definition that triggers on the drift detection condition and uses the CLI script notification type, pointing to the JavaScript file that you created.

Expected Results

Any time drift is detected on the EAP server, it triggers an alert, same as in Section 15.7, "*Extended Example:* Defining Required EAP Configuration". This time, the alert launches the CLI script in response and automatically deploys the bundle — which already has the approved EAP configuration — to the resource. This means that the EAP server is never more than a few minutes out of compliance, roughly the length of one alert scan. All without requiring intervention from Tim the IT Guy.

Example 15.2. fix-eap.js Script

```
/**
 * If obj is a JS array or a java.util.Collection, each element is
passed to
 * the callback function. If obj is a java.util.Map, each map entry is
passed
 * to the callback function as a key/value pair. If obj is none of the
 * aforementioned types, it is treated as a generic object and each of
its
 * properties is passed to the callback function as a name/value pair.
 */
function foreach(obj, fn) {
  if (obj instanceof Array) {
    for (i in obj) {
      fn(obj[i]);
    }
  }
  else if (obj instanceof java.util.Collection) {
    var iterator = obj.iterator();
    while (iterator.hasNext()) {
      fn(iterator.next());
    }
  }
  else if (obj instanceof java.util.Map) {
    var iterator = obj.entrySet().iterator()
    while (iterator.hasNext()) {
      var entry = iterator.next();
      fn(entry.key, entry.value);
    }
  }
  else {   // assume we have a generic object
    for (i in obj) {
      fn(i, obj[i]);
    }
  }
}

/**
 * Iterates over obj similar to foreach. fn should be a predicate that
evaluates
 * to true or false. The first match that is found is returned.
 */
function find(obj, fn) {
  if (obj instanceof Array) {
    for (i in obj) {
      if (fn(obj[i])) {
        return obj[i]
      }
    }
  }
  else if (obj instanceof java.util.Collection) {
    var iterator = obj.iterator();
    while (iterator.hasNext()) {
      var next = iterator.next();
      if (fn(next)) {
        return next;
      }
    }
  }
  else if (obj instanceof java.util.Map) {
    var iterator = obj.entrySet().iterator();
    while (iterator.hasNext()) {
      var entry = iterator.next();
      if (fn(entry.key, entry.value)) {
        return {key: entry.key, value: entry.value};
      }
    }
  }
  else {
    for (i in obj) {
      if (fn(i, obj[i])) {
        return {key: i, value: obj[i]};
      }
    }
  }
  return null;
}

/**
 * Iterates over obj similar to foreach. fn should be a predicate that
evaluates
 * to true or false. All of the matches are returned in a
java.util.List.
 */
function findAll(obj, fn) {
  var matches = java.util.ArrayList();
  if ((obj instanceof Array) || (obj instanceof java.util.Collection)) {
    foreach(obj, function(element) {
      if (fn(element)) {
        matches.add(element);
      }
    });
  }
  else {
    foreach(obj, function(key, value) {
      if (fn(theKey, theValue)) {
        matches.add({key: theKey, value: theValue});
      }
    });
  }
  return matches;
}

/**
 * A convenience function to convert javascript hashes into RHQ's
configuration
 * objects.
 * <p>
 * The conversion of individual keys in the hash follows these rules:
 * <ol>
 * <li> if a value of a key is a javascript array, it is interpreted as
PropertyList
 * <li> if a value is a hash, it is interpreted as a PropertyMap
 * <li> otherwise it is interpreted as a PropertySimple
 * <li> a null or undefined value is ignored
 * </ol>
 * <p>
 * Note that the conversion isn't perfect, because the hash does not
contain enough
 * information to restore the names of the list members.
 * <p>
 * Example: <br/>
 * <pre><code>
 * {
 *   simple : "value",
 *   list : [ "value1", "value2"],
 *   listOfMaps : [ { k1 : "value", k2 : "value" }, { k1 : "value2", k2
: "value2" } ]
 * }
 * </code></pre>
 * gets converted to a configuration object:
 * Configuration:
 * <ul>
 * <li> PropertySimple(name = "simple", value = "value")
 * <li> PropertyList(name = "list")
 *      <ol>
 *      <li>PropertySimple(name = "list", value = "value1")
 *      <li>PropertySimple(name = "list", value = "value2")
 *      </ol>
 * <li> PropertyList(name = "listOfMaps")
 *      <ol>
 *      <li> PropertyMap(name = "listOfMaps")
 *           <ul>
 *           <li>PropertySimple(name = "k1", value = "value")
 *           <li>PropertySimple(name = "k2", value = "value")
 *           </ul>
 *      <li> PropertyMap(name = "listOfMaps")
 *           <ul>
 *           <li>PropertySimple(name = "k1", value = "value2")
 *           <li>PropertySimple(name = "k2", value = "value2")
 *           </ul>
 *      </ol>
 * </ul>
 * Notice that the members of the list have the same name as the list
itself
 * which generally is not the case.
 */
function asConfiguration(hash) {

 config = new Configuration;

 for(key in hash) {
  value = hash[key];

  if (value == null) {
   continue;
  }

  (function(parent, key, value) {
   function isArray(obj) {
    return typeof(obj) == 'object' && (obj instanceof Array);
   }

   function isHash(obj) {
    return typeof(obj) == 'object' && !(obj instanceof Array);
   }

   function isPrimitive(obj) {
    return typeof(obj) != 'object';
   }

   //this is an anonymous function, so the only way it can call itself
   //is by getting its reference via argument.callee. Let's just assign
   //a shorter name for it.
   var me = arguments.callee;
```

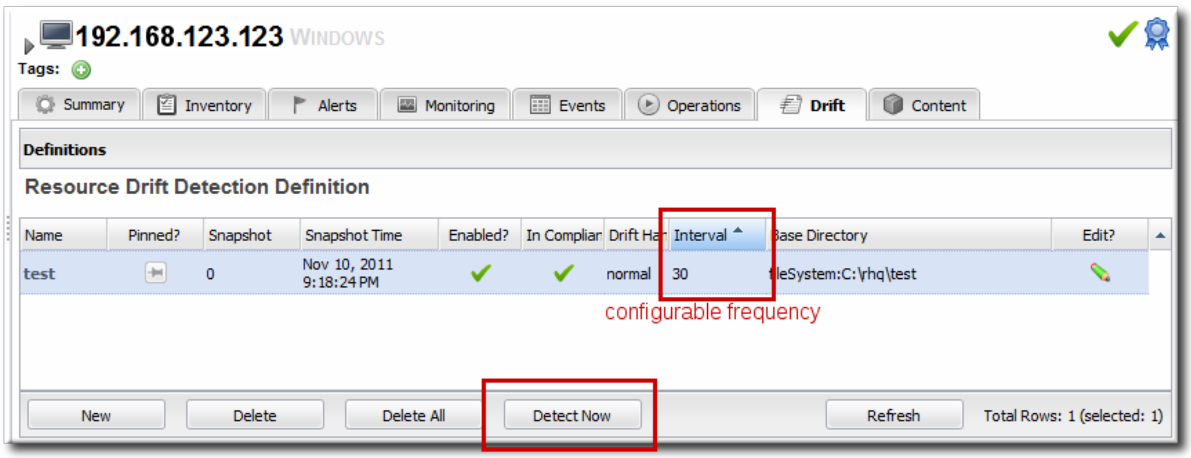
```
      var prop = null;

      if (isPrimitive(value)) {
       prop = new PropertySimple(key, new java.lang.String(value));
      } else if (isArray(value)) {
       prop = new PropertyList(key);
       for(var i = 0; i < value.length; ++i) {
        var v = value[i];
        if (v != null) {
         me(prop, key, v);
        }
       }
      } else if (isHash(value)) {
       prop = new PropertyMap(key);
       for(var i in value) {
        var v = value[i];
        if (value != null) {
         me(prop, i, v);
        }
       }
      }

      if (parent instanceof PropertyList) {
       parent.add(prop);
      } else {
       parent.put(prop);
      }
     })(config, key, value);
    }

    return config;
   }

   /**
    * Opposite of <code>asConfiguration</code>. Converts an RHQ's
   configuration object
    * into a javascript hash.
    *
    * @param configuration
    */
   function asHash(configuration) {
    ret = {}

    iterator = configuration.getMap().values().iterator();
    while(iterator.hasNext()) {
     prop = iterator.next();

     (function(parent, prop) {
      function isArray(obj) {
       return typeof(obj) == 'object' && (obj instanceof Array);
      }

      function isHash(obj) {
       return typeof(obj) == 'object' && !(obj instanceof Array);
      }
      var me = arguments.callee;

      var representation = null;

      if (prop instanceof PropertySimple) {
       representation = prop.stringValue;
      } else if (prop instanceof PropertyList) {
       representation = [];

       for(var i = 0; i < prop.list.size(); ++i) {
        var child = prop.list.get(i);
        me(representation, child);
       }
      } else if (prop instanceof PropertyMap) {
       representation = {};

       var childIterator = prop.getMap().values().iterator();
       while(childIterator.hasNext()) {
        var child = childIterator.next();

        me(representation, child);
       }
      }

      if (isArray(parent)) {
       parent.push(representation);
      } else if (isHash(parent)) {
       parent[prop.name] = representation;
      }
     })(ret, prop);
   }
   (function(parent) {

   })(configuration);

   return ret;
  }

  /**
   * A simple function to create a new bundle version from a zip file
  containing
   * the bundle.
   *
   * @param pathToBundleZipFile the path to the bundle on the local file
  system
   *
   * @return an instance of BundleVersion class describing what's been
  created on
   * the RHQ server.
   */
  function createBundleVersion(pathToBundleZipFile) {
   var bytes = getFileBytes(pathToBundleZipFile)
   return BundleManager.createBundleVersionViaByteArray(bytes)
  }

  /**
```

```
 * This is a helper function that one can use to find out what base
directories
 * given resource type defines.
 * <p>
 * These base directories then can be used when specifying bundle
destinations.
 *
 * @param resourceTypeId
 * @returns a java.util.Set of ResourceTypeBundleConfiguration objects
 */
function getAllBaseDirectories(resourceTypeId) {
 var crit = new ResourceTypeCriteria;
 crit.addFilterId(resourceTypeId);
 crit.fetchBundleConfiguration(true);

 var types = ResourceTypeManager.findResourceTypesByCriteria(crit);

 if (types.size() == 0) {
  throw "Could not find a resource type with id " + resourceTypeId;
 } else if (types.size() > 1) {
  throw "More than one resource type found with id " + resourceTypeId +
"! How did that happen!";
 }

 var type = types.get(0);

 return
type.getResourceTypeBundleConfiguration().getBundleDestinationBaseDirect
ories();
}

/**
 * Creates a new destination for given bundle. Once a destination
exists,
 * actual bundle versions can be deployed to it.
 * <p>
 * Note that this only differs from the
<code>BundleManager.createBundleDestination</code>
 * method in the fact that one can provide bundle and resource group
names instead of their
 * ids.
 *
 * @param destinationName the name of the destination to be created
 * @param description the description for the destination
 * @param bundleName the name of the bundle to create the destination
for
 * @param groupName name of a group of resources that the destination
will handle
 * @param baseDirName the name of the basedir definition that represents
where inside the
 *                    deployment of the individual resources the bundle
will get deployed
 * @param deployDir the specific sub directory of the base dir where the
bundles will get deployed
 *
 * @return BundleDestination object
 */
function createBundleDestination(destinationName, description,
bundleName, groupName, baseDirName, deployDir) {
 var groupCrit = new ResourceGroupCriteria;
 groupCrit.addFilterName(groupName);
 var groups =
ResourceGroupManager.findResourceGroupsByCriteria(groupCrit);

 if (groups.empty) {
  throw "No group called '" + groupName + "' found.";
 }

 var group = groups.get(0);

 var bundleCrit = new BundleCriteria;
 bundleCrit.addFilterName(bundleName);
 var bundles = BundleManager.findBundlesByCriteria(bundleCrit);

 if (bundles.empty) {
  throw "No bundle called '" + bundleName + "' found.";
 }

 var bundle = bundles.get(0);

 return BundleManager.createBundleDestination(bundle.id,
destinationName, description, baseDirName, deployDir, group.id);
}

/**
 * Tries to deploy given bundle version to provided destination using
given configuration.
 * <p>
 * This method blocks while waiting for the deployment to complete or
fail.
 *
 * @param destination the bundle destination (or id thereof)
 * @param bundleVersion the bundle version to deploy (or id thereof)
 * @param deploymentConfiguration the deployment configuration. This can
be an ordinary
 * javascript object (hash) or an instance of RHQ's Configuration. If it
is the former,
 * it is converted to a Configuration instance using the
<code>asConfiguration</code>
 * function from <code>util.js</code>. Please consult the documentation
of that method
 * to understand the limitations of that approach.
 * @param description the deployment description
 * @param isCleanDeployment if true, perform a wipe of the deploy
directory prior to the deployment; if false,
 * perform as an upgrade to the existing deployment, if any
 *
 * @return the BundleDeployment instance describing the deployment
 */
function deployBundle(destination, bundleVersion,
deploymentConfiguration, description, isCleanDeployment) {
 var destinationId = destination;
 if (typeof(destination) == 'object') {
  destinationId = destination.id;
 }

 var bundleVersionId = bundleVersion;
 if (typeof(bundleVersion) == 'object') {
  bundleVersionId = bundleVersion.id;
 }

 var deploymentConfig = deploymentConfiguration;
 if (!(deploymentConfiguration instanceof Configuration)) {
  deploymentConfig = asConfiguration(deploymentConfiguration);
 }

 var deployment = BundleManager.createBundleDeployment(bundleVersionId,
destinationId, description, deploymentConfig);

 deployment = BundleManager.scheduleBundleDeployment(deployment.id,
isCleanDeployment);

 var crit = new BundleDeploymentCriteria;
 crit.addFilterId(deployment.id);

 while (deployment.status == BundleDeploymentStatus.PENDING ||
deployment.status == BundleDeploymentStatus.IN_PROGRESS) {
  java.lang.Thread.currentThread().sleep(1000);
  var dps = BundleManager.findBundleDeploymentsByCriteria(crit);
  if (dps.empty) {
   throw "The deployment disappeared while we were waiting for it to
complete.";
  }

  deployment = dps.get(0);
 }

 return deployment;
}


var destinationId = 10002;
var bundleVersionId = 10002;
var deploymentConfig = null;
var description = "redeploy due to drift";
// NOTE: It's essential that isCleanDeployment=true, otherwise files
that have drifted will not be replaced with their
//        original versions from the bundle.
var isCleanDeployment = true;
deployBundle(10002, 10002, deploymentConfig, description, true);
```

15.10. RUNNING DRIFT DETECTION MANUALLY

The drift detection scan runs periodically, according to the interval set in the definition. (The default is 1800 seconds, or 30 minutes.) There can be times when you know that files in the directory have changed and you need a snapshot to be created immediately, but you do not want to change the interval permanently. Simply run a detection scan manually.

1. Click the **`Inventory`** tab in the top menu.
2. Search for the resource.
3. Click the **`Drift`** tab.
4. Select the drift definition to run the scan for.
5. Click the **`Detect Now`** button.



Name	Pinned?	Snapshot	Snapshot Time	Enabled?	In Compliar	Drift Ha	Interval	Base Directory	Edit?
test		0	Nov 10, 2011 9:18:24 PM	✓	✓	normal	30	fileSystem:C:\rhq\test	

15.11. SETTING PLANNED CHANGES OR DISABLING DRIFT DEFINITIONS

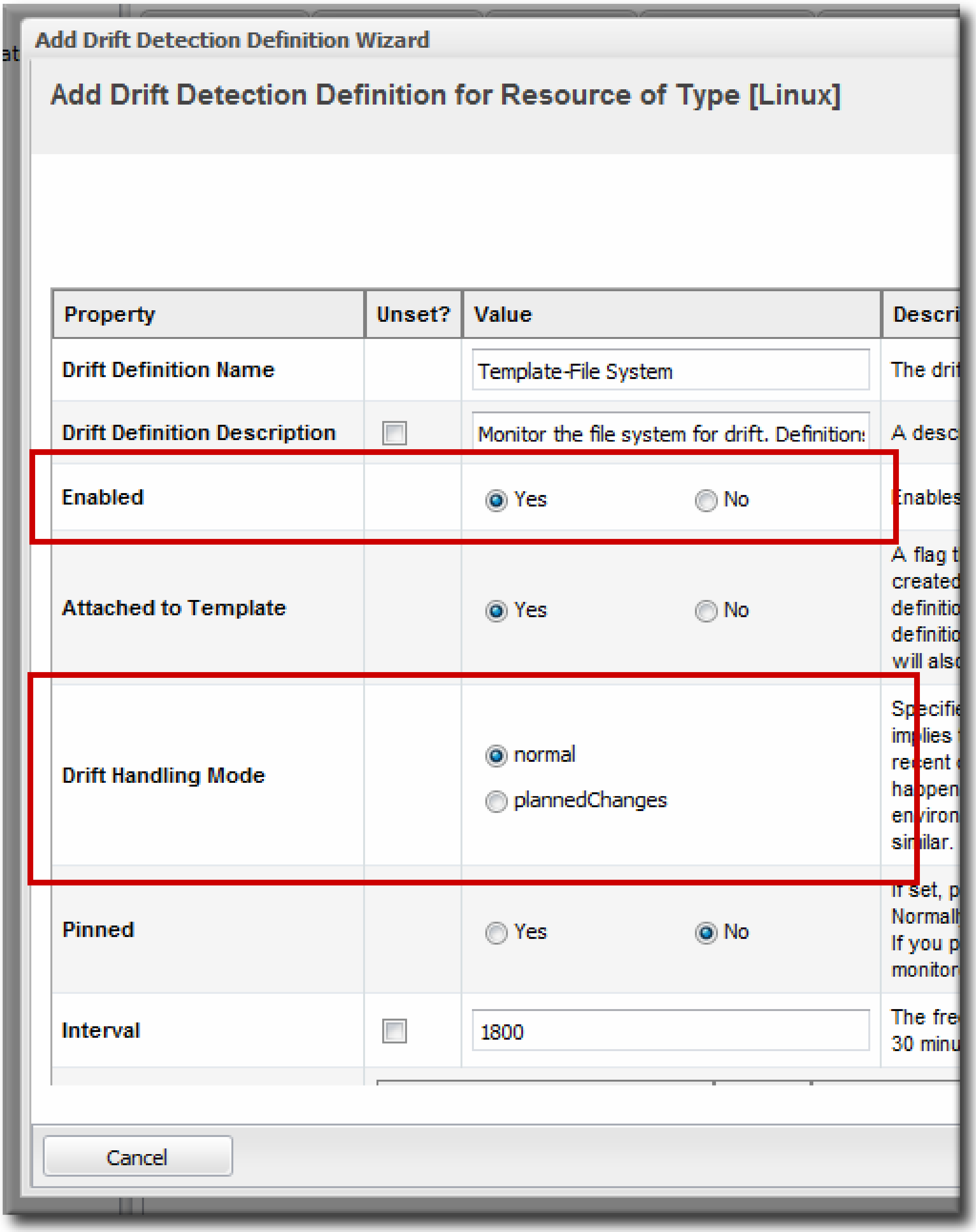
The assumption behind drift monitoring is that there is an identified and specific configuration for a platform or application and that that configuration should be preserved. Changes, therefore, are undesirable and need to be monitored.

However, there can be times when changes are expected, such as scheduled maintenance and upgrade periods. In that situation, it's beneficial to suspend drift monitoring to keep from creating unnecessary static with drift alerts.

There are two ways to suspend drift monitoring:

- Set the drift handling mode to *planned changes*. This keeps running drift detection scans and records changes. Since the changes are expected, though, it doesn't trigger a drift detection event, so it does not issue a drift alert.
- Actually *disable* the drift definition. This suspends the drift detection runs for the definition, not just drift events.

The drift handling mode and the enable option for the drift definition can be edited in the definition entry, as in Section 15.4, “Editing Drift Definitions”.


Add Drift Detection Definition Wizard

Add Drift Detection Definition for Resource of Type [Linux]

Property	Unset?	Value	Descri
Drift Definition Name		Template-File System	The drif
Drift Definition Description	☐	Monitor the file system for drift. Definition:	A desc
Enabled		[✓ Yes] No	Enables
Attached to Template		[✓ Yes] No	A flag t created definitio definitio will also
Drift Handling Mode		[✓ normal] plannedChanges	Specifie implies recent happen environ similar.
Pinned		Yes [✓ No]	If set, p Normall If you p monitor
Interval	☐	1800	The fre 30 minu

Figure 15.5. Drift Handling Mode and Enable Options

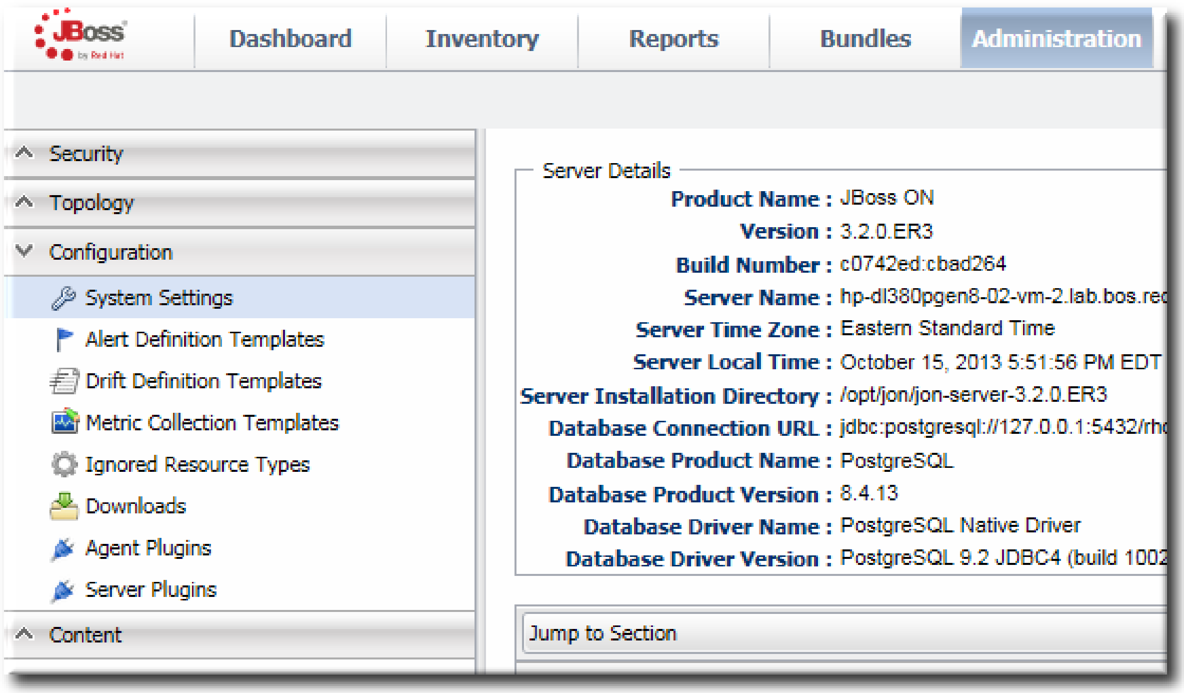
15.12. CHANGING HOW LONG DRIFT SNAPSHOTS ARE STORED

Drift snapshots are stored within the JBoss ON database for a limited period of time (31 days). This allows enough time to remediate any unauthorized changes, but maintains some resource limits on how much data is stored.

Any *unused* snapshots are removed once the time limit is reached. Unused snapshots are snapshots which are not pinned or which are associated with a disabled or deleted drift definition (orphaned).

Baseline snaphots (snapshot 0) and pinned snapshots are always saved.

1. In the **`System Configuration`** menu, select the **Settings** item.

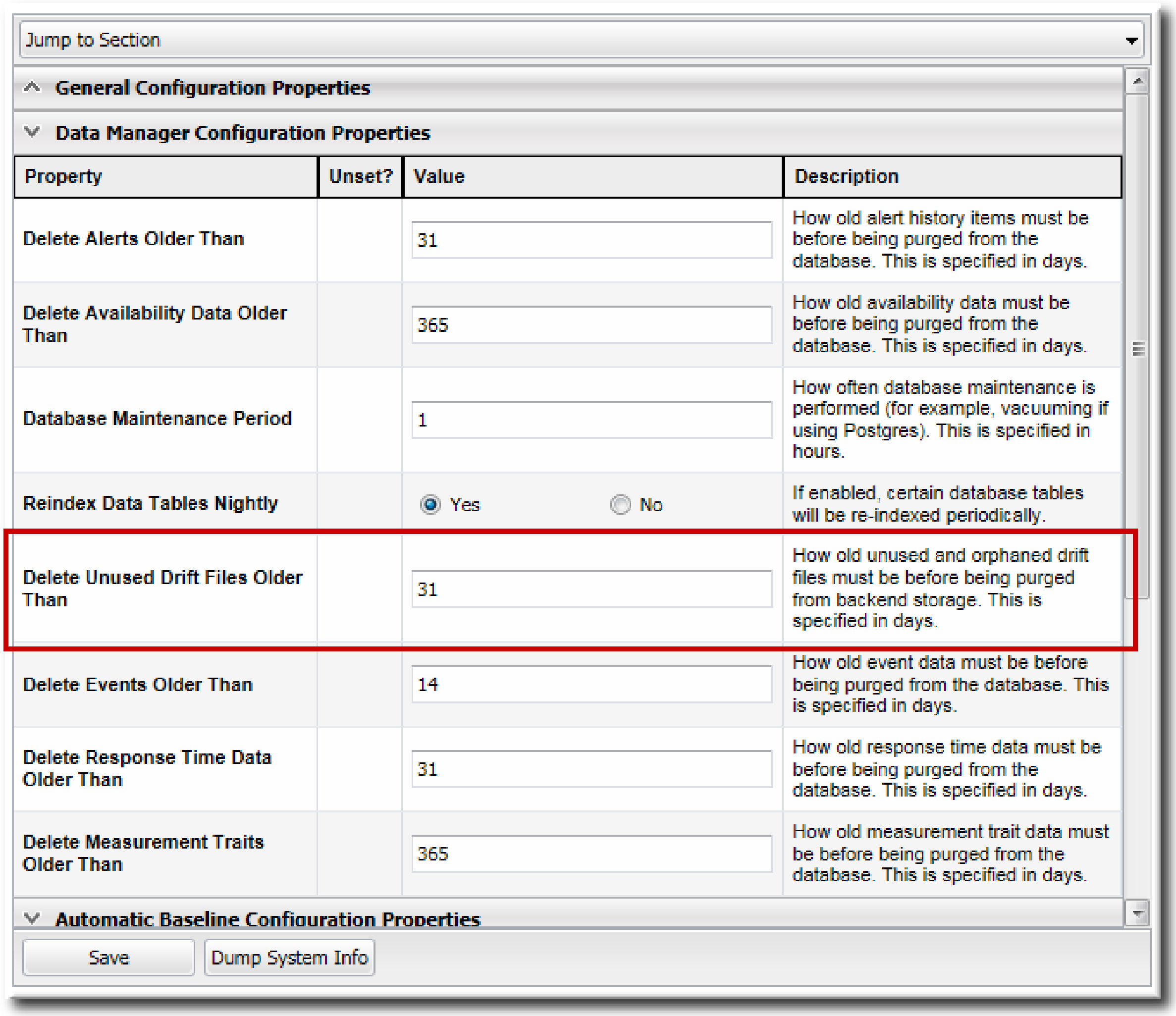


Server Details

Product Name :	JBoss ON
Version :	3.2.0.ER3
Build Number :	c0742ed:cbad264
Server Name :	hp-dl380pgen8-02-vm-2.lab.bos.red
Server Time Zone :	Eastern Standard Time
Server Local Time :	October 15, 2013 5:51:56 PM EDT
Server Installation Directory :	/opt/jon/jon-server-3.2.0.ER3
Database Connection URL :	jdbc:postgresql://127.0.0.1:5432/rhq
Database Product Name :	PostgreSQL
Database Product Version :	8.4.13
Database Driver Name :	PostgreSQL Native Driver
Database Driver Version :	PostgreSQL 9.2 JDBC4 (build 1002

2. Scroll to the **`Data Manager Configuration Properties`** section.

3. Change the storage times for the drift snapshots. Unused snapshots are not pinned or a baseline, while orphaned snapshots are related to disabled definitions.


Jump to Section

General Configuration Properties

Data Manager Configuration Properties

Property	Unset?	Value	Description
Delete Alerts Older Than		31	How old alert history items must be before being purged from the database. This is specified in days.
Delete Availability Data Older Than		365	How old availability data must be before being purged from the database. This is specified in days.
Database Maintenance Period		1	How often database maintenance is performed (for example, vacuuming if using Postgres). This is specified in hours.
Reindex Data Tables Nightly		[✓ Yes] No	If enabled, certain database tables will be re-indexed periodically.
Delete Unused Drift Files Older Than		31	How old unused and orphaned drift files must be before being purged from backend storage. This is specified in days.
Delete Events Older Than		14	How old event data must be before being purged from the database. This is specified in days.
Delete Response Time Data Older Than		31	How old response time data must be before being purged from the database. This is specified in days.
Delete Measurement Traits Older Than		365	How old measurement trait data must be before being purged from the database. This is specified in days.

Automatic Baseline Configuration Properties

15.13. UNDERSTANDING DRIFT AND JBOSS ON AGENTS AND SERVERS

15.13.1. Drift Inventory

Both the JBoss ON agent and the JBoss ON server maintain their own inventories of the resources, directories, and files monitored for drift. When the agent starts up, it compares its inventory with the server inventory.

The drift information is stored, with the other agent data, in the ***`agentRoot`*****`/rhq-agent/data/`** directory. The information in this directory is deleted if the agent is started with new configuration (**`--cleanconfig`**) or it can be intentionally purged (**`--purgedata`**). If the drift information is lost, then the agent requests the last snapshot from the JBoss ON server.

The agent always sends the latest changeset to the server as a snapshot. If the server is offline for some period and misses updates, then the agent sends the most current snapshot, which effectively rolls all changes into one snapshot, even if the changes accumulated over several drift detection runs.

15.13.2. The Drift Server Plug-in

The server processes drift changes through a server-side plug-in. This plug-in must be enabled for the server to recognize and process drift data sent from the agent.

As with other server-side plug-ins, the drift plug-in can be disabled. However, this effectively and entirely disables drift monitoring on that server, and no drift information is processed or stored. That is slightly different than the behavior of other server subsystems. For example, an individual alert sender can be disabled, but alert detections are still run and alert information is still processed, stored, and displayed by the JBoss ON server.

WARNING

If the drift server-side plug-in is disabled, then the server ignores any incoming drift reports. Even if the drift server-side plug-in is re-enabled, any information sent while the plug-in was disabled is lost.

[3] JBoss ON detects that changes have been made to a binary file. It does not display binary files or compare or diff changes between versions for binary files, only text files.

PART III. MONITORING

CHAPTER 16. INTRODUCTION: MONITORING AND RESPONDING TO RESOURCE ACTIVITY

One of the core functions of JBoss Operations Network is that it lets administrators stay aware of the state of their JBoss servers, platforms, and overall IT environment.

The current state of individual servers and applications provides critical information to IT staff about traffic and usage, equipment failures, and server performance. JBoss Operations Network can supply a clearer picture of these critical data by automatically *monitoring* resources in its inventory.

The most powerful aspect of management is the ability to know, accurately, where your resources are and to respond to that ever-changing situation reliably.

16.1. MONITORING AND TYPES OF DATA

Monitoring gives insight into how a specific machine, application, or service is performing. JBoss ON collects different types of information from different native and external sources for its managed resources.

JBoss Operations Network is not a real-time monitor or an archive of data points or a profiler. What JBoss ON does is, in essence, filter and process raw data so that long-term trends, operating parameters, and performance histories — the purpose of monitoring — are clear and accessible from the data. JBoss ON uses *schedules* to define what information to gather and how frequently (anywhere from 30 seconds to hours). This prioritizes the performance information for a resource and makes important information more visible and coherent.

Although the precise information gathered is different depending on the resource type, there are a few broad categories of monitoring data. Each category obtains information from a different place and is useful to determine a different aspect of resource behavior.

Availability or "up and down" monitoring

This is both basic and critical. Availability is *status* information about the resource, whether it is running or stopped.

Numeric metrics

Metrics are the core performance data for a resource. Almost every software product exposes some sort of information about itself, some measurable facet that can be checked. This is usually This numeric information is collected by JBoss ON, on defined schedules.

Metric information is processed by the server. There are three states of the monitoring data used:

- *Raw data*, which are the readings collected on schedule by the agent and sent to the server
- *Aggregated data*, which is compressed data processed by the server into 1-hour, 6-hour, and 24-hour averages and used to calculate baselines and normal operating ranges for resources. These aggregated data are the information displayed in the monitoring graphs and returned in the CLI as metrics.
- *Live values*, which are ad hoc requests for the current value of a metric.

  Metric values are rolling live-streams of the resource state; they are essentially snapshots that the agent takes of the readings on predefined schedules. Those data are then aggregated into means and averages to use to track resource performance.

  Live values are immediate, aggregated, current readings of a metric value.

Metric information is especially important because it is collected and stored long-term. This allows for historical views on resource performance, as well as recent views.

Logfile messages (events)

While JBoss ON is not a log viewer, it can monitor specified logs and check for important log messages based on severity or strings within the log messages. This is *event* monitoring, and it allows JBoss ON to identify incidents for a resource and to send an alert notification and, if necessary, take corrective action based on dynamic information outside normal metrics.

Response time metrics

Certain types of resources (URLs for web servers or session beans) depend on responsiveness as a component of overall performance. Response time or call-time data tracks how quickly the URL or session bean responds to client requests and helps determine that the overall application is performant.

Descriptive strings (traits)

Most resources have some relatively static information that describe the resource itself, such as an instance name, build date, or version number. This information is a *trait*. As with other attributes for a resource, this can be monitored. Traits are useful to identify changes to the underlying application, like a version update.

16.2. ALERTS AND RESPONSES TO CHANGING CONDITIONS

A critical part of monitoring is being aware of when undesirable events occur. Alerting works with other functions in JBoss ON management (monitoring data and configuration drift detection) to define *conditions* for triggering an alert.

When an alert condition is met, alerting in JBoss ON serves two important functions:

- **Alerts communicate** that there has been a problem, based on parameters defined by an administrator.
- **Alerts respond** to incidents automatically. Administrators can automatically initiate an operation, run a JBoss ON CLI script to change JBoss ON or resource configuration, redeploy content, or run a shell script, all in response to an alert condition.

  Automatic, administrator-defined responses to alerts make it significantly easier for administrators to address infrastructure problems quickly, and can mitigate the effect of outages.

Alerts are based on metrics information, call-time data, availability, and events, all normal monitoring elements. Alerting can also be based on critical changes to a resource, defined in drift definitions that track configuration drift. Tracking configuration for resources along with monitoring data lets administrators remedy unplanned or undesirable system changes easily and consistently.

16.3. POTENTIAL IMPACT ON SERVER PERFORMANCE

Theoretically, there is no limit to the number of metrics that can collected or the number of alerts that can be fired.

In reality, there are natural constraints within the IT environment that limit both monitoring and alert settings:

- Database performance, which is the primary factor in most environments
- Network bandwidth

There are no hard limits on JBoss ON's alerting and monitoring configuration since it depends on the number of resources, number of metrics, collection frequency, and the number of alerts.

As a rule of thumb, there are these performance thresholds:

- Up to 30,000 metrics can be collected per minute
- Up to 100,000 alerts can be fired per day (roughly 70 per minute)

Plan how to implement metrics collection and alerting. Prioritize resources and then the information required from those resources when enabling metrics schedules and setting collection frequencies. Then, based on those priorities, plan what alerts are required.

Clear monitoring and alerting strategies can help maintain performance while still gathering critical information.

NOTE

Additional storage nodes can be added to the storage cluster to extend the amount of metrics collected.

The storage nodes can be permanent, as part of the JBoss ON design, or can be created dynamically, using an alert condition on the existing nodes to trigger a new node deployment.

Creating additional nodes is covered in Section 24.3, “Deploying and Managing Storage Nodes”.

16.4. DIFFERENCES WITH MONITORING BASED ON DIFFERENT RESOURCE TYPES

Available metrics, events, traits, and other monitoring settings are defined for each resource type in its plug-in descriptor.

Obviously, software of completely different types have different possible monitoring configuration.

However, monitoring settings can be different between releases of the same software. Either different metrics are available or the same metric may have different configuration names. For example, JBoss EAP 4 and 5 have the same metrics, related to monitoring the EAP server JVM, threads, and transactions. Because of the different management structure in JBoss EAP 6, there are different metrics, related to management requests between the servers in the EAP 6 domain.

The Resource Reference: Monitoring, Operation, and Configuration Options has a complete references of available metrics for the official JBoss ON agent plug-ins. Check this guide to see what differences there are between release versions.

CHAPTER 17. MONITORING REPORTS AND DATA

Monitoring information in JBoss ON is easy to find. Resources and groups both have dashboards which contain snapshot views of the most recent metrics values and a series of graphs and tables which break down the different metrics values over a given time window.

Monitoring information is available in several areas:

- Dashboards with metrics portlets for individual resources, compatible groups, and the main dashboard
- Timelines, which aggregate all collected data, events, configuration, operations, and other changes for a resource
- Resource-level charts and tables for metrics
- A suspect metrics report for outlier or out-of-bounds metrics

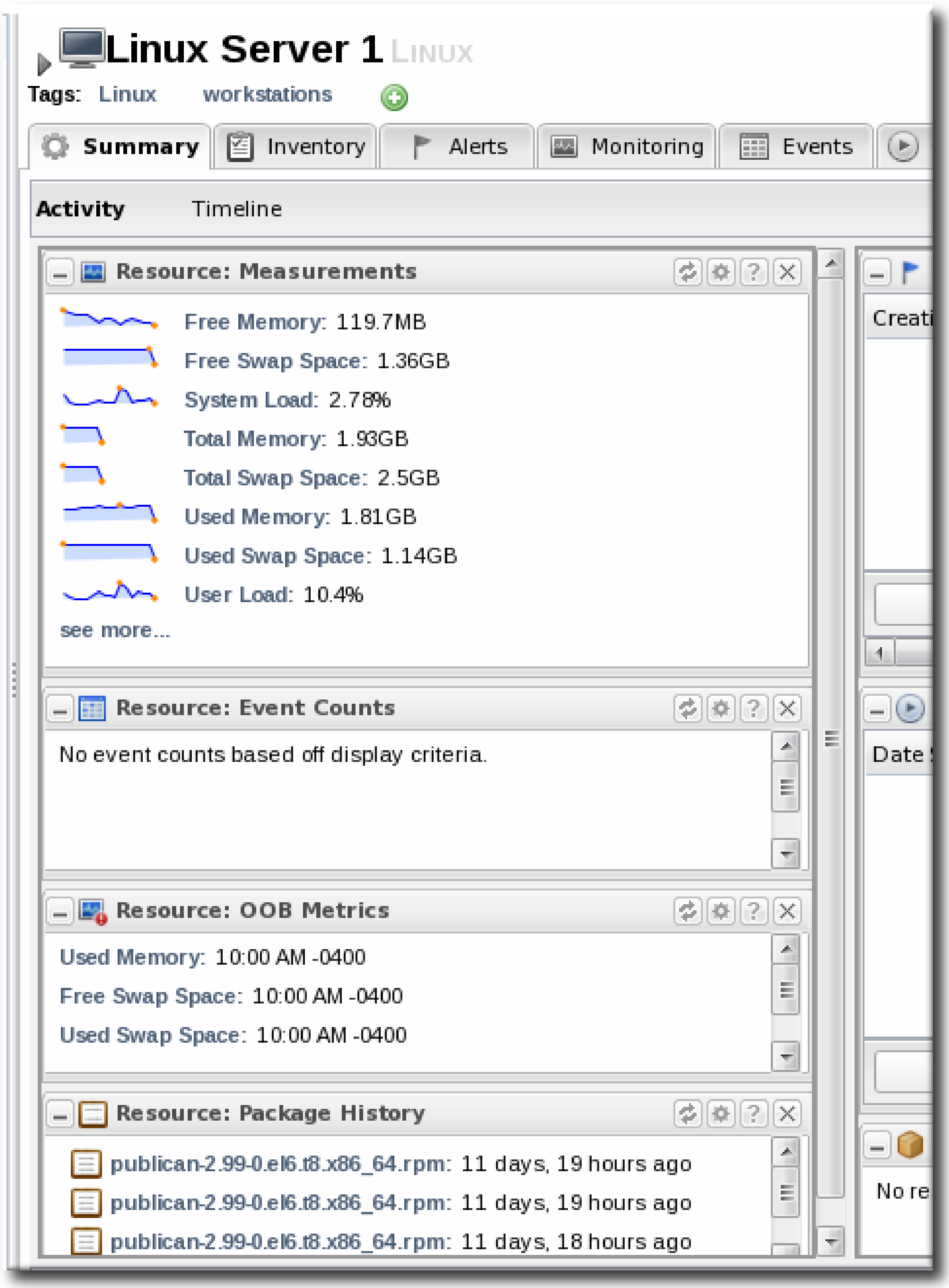
17.1. DASHBOARDS AND PORTLETS

The fastest place to view monitoring and alerting information is through one of the JBoss ON dashboards. The dashboards collect almost all monitoring, event, alert, and operations data into a single location.

Each data set is collected in a separate box or *portlets* displayed in the dashboard. These portlets can be edited, added, and removed from the dashboards; this is covered in the *Admin: Initial Setup for the Resource Inventory, Groups, and Users*.

17.1.1. Resource-Level Dashboards

The **`Summary > Activity`** tab for an individual resource (or compatible group) shows a snapshot of all recent actions on the resource, such as new packages and content, inventory changes, events, operations, and alerts. There is also a portlet that displays the most recent detected value of the primary metrics for the resource.



Free Memory: 119.7MB
Free Swap Space: 1.36GB
System Load: 2.78%
Total Memory: 1.93GB
Total Swap Space: 2.5GB
Used Memory: 1.81GB
Used Swap Space: 1.14GB
User Load: 10.4%
see more...

No event counts based off display criteria.

Used Memory: 10:00 AM -0400
Free Swap Space: 10:00 AM -0400
Used Swap Space: 10:00 AM -0400

publican-2.99-0.el6.t8.x86_64.rpm: 11 days, 19 hours ago
publican-2.99-0.el6.t8.x86_64.rpm: 11 days, 19 hours ago
publican-2.99-0.el6.t8.x86_64.rpm: 11 days, 18 hours ago

Figure 17.1. Resource Summary Tab

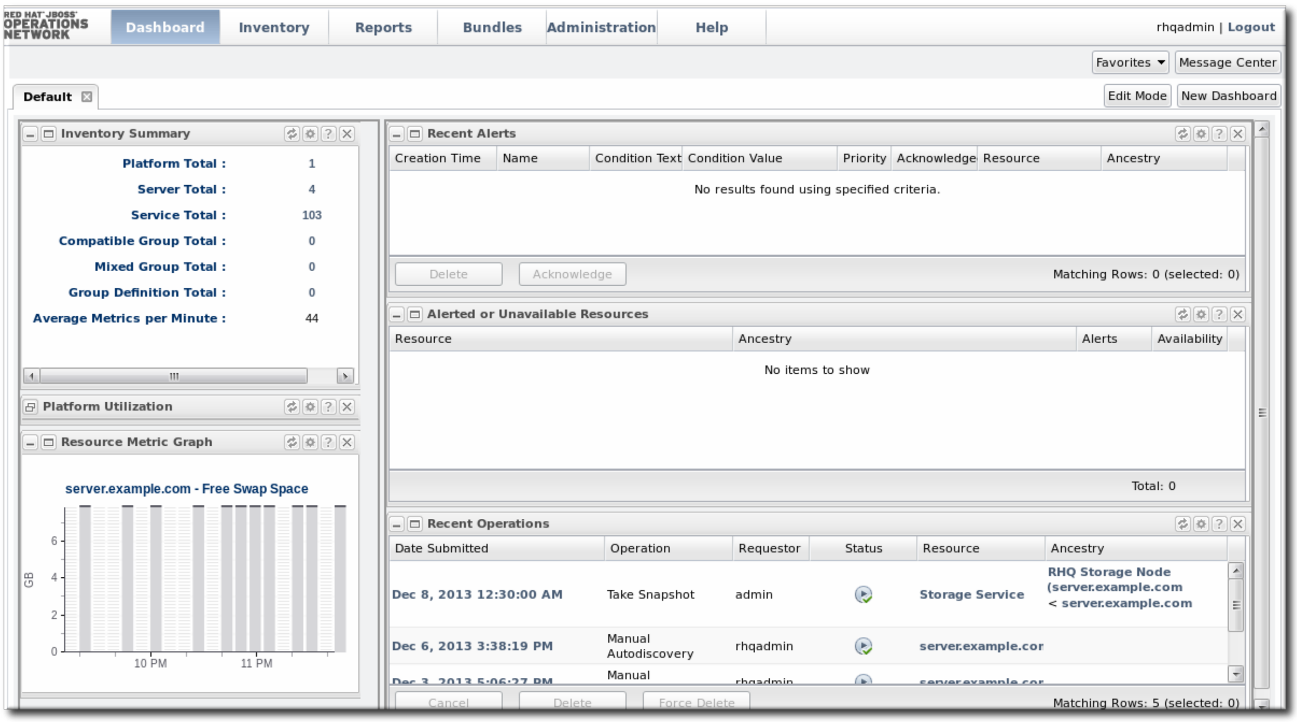
Click on any metric name in the `Resource: Measurements` portlet opens the metric graph. Clicking the `see more...` link opens the metrics charts in the `Monitoring` tab.

17.1.2. Main Dashboard

The **Dashboard** main page has a global view of all resources in the inventory. By default, this page shows only alerting data and unavailable resources. However, the **Dashboard** can be customized to show different portlets of monitoring data. Additionally, the main page can have multiple dashboards, so a dashboard can be created to look at different metrics for the same resource, the same metrics for different resources, or a combination of relevant metrics for a group of related resources — whatever you design.

The main dashboard has several types of portlets specifically for monitoring data:

- Platform Utilization, which shows free memory, CPU usage, and other metrics related to platform performance.
- Alerted or Unavailable Resources, which shows a list of the most recent five resources which have issued an alert or been reported as down
- A graph for a specific metric for a compatible group
- A graph for a specific metric for a resource



Inventory Summary

Platform Total :	1
Server Total :	4
Service Total :	103
Compatible Group Total :	0
Mixed Group Total :	0
Group Definition Total :	0
Average Metrics per Minute :	44

Recent Alerts

Creation Time	Name	Condition Text	Condition Value	Priority	Acknowledge	Resource	Ancestry

No results found using specified criteria.

Matching Rows: 0 (selected: 0)

Alerted or Unavailable Resources

Resource	Ancestry	Alerts	Availability

No items to show

Total: 0

Recent Operations

Date Submitted	Operation	Requestor	Status	Resource	Ancestry
Dec 8, 2013 12:30:00 AM	Take Snapshot	admin	✓	Storage Service	RHQ Storage Node (server.example.com < server.example.com
Dec 6, 2013 3:38:19 PM	Manual Autodiscovery	rhqadmin	✓	server.example.cor	
Dec 3, 2013 5:06:27 PM	Manual	rhqadmin		server.example.cor	

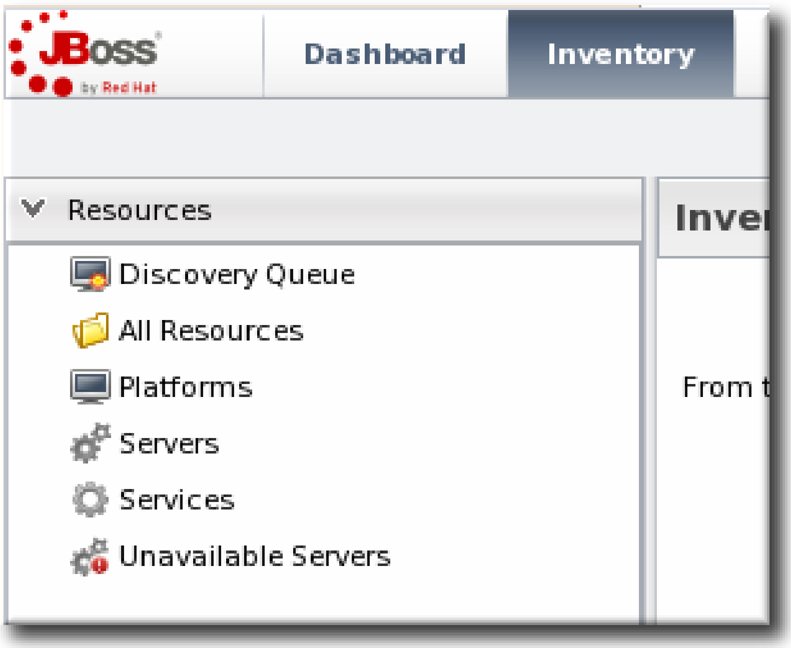
Matching Rows: 5 (selected: 0)

Figure 17.2. Dashboard Portlets with MOnitoring Data

17.1.3. Adding Monitoring Metrics to the Main Dashboard

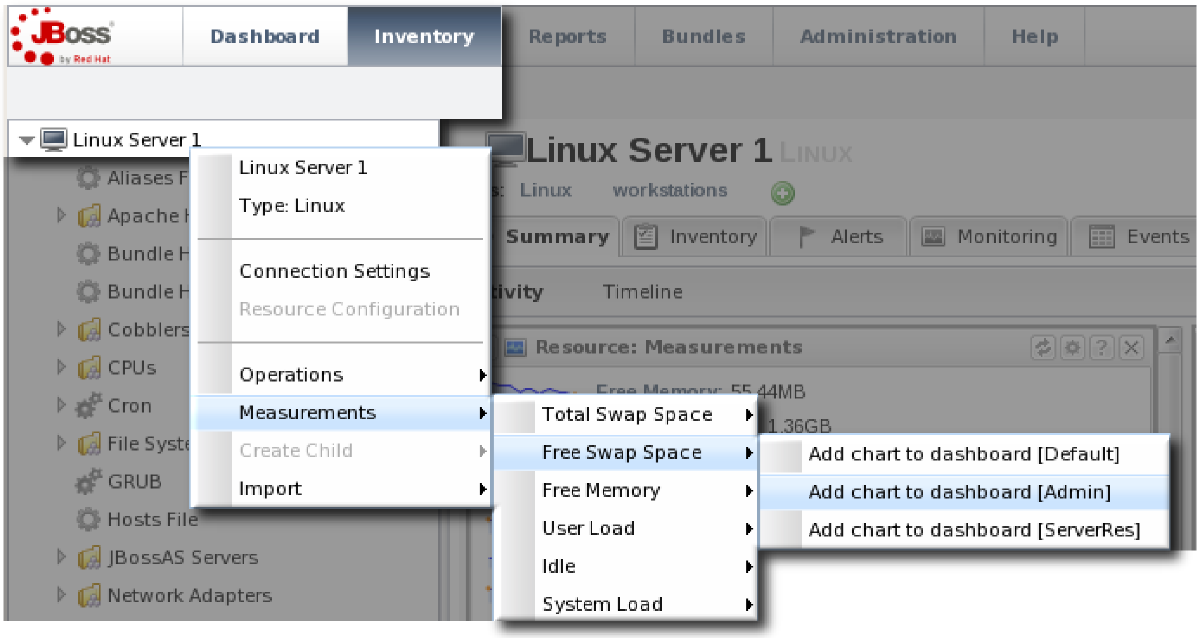
Charts for a specific metric for a resource can be added to the Dashboard. This makes it easier to see the current state of important readings for common or critical resources immediately, without having to configure alerts or check resource entries.

1. Click the `Inventory` tab in the top menu.
2. Select the resource type in the `Resources` menu table on the left, and then browse or search for the resource.

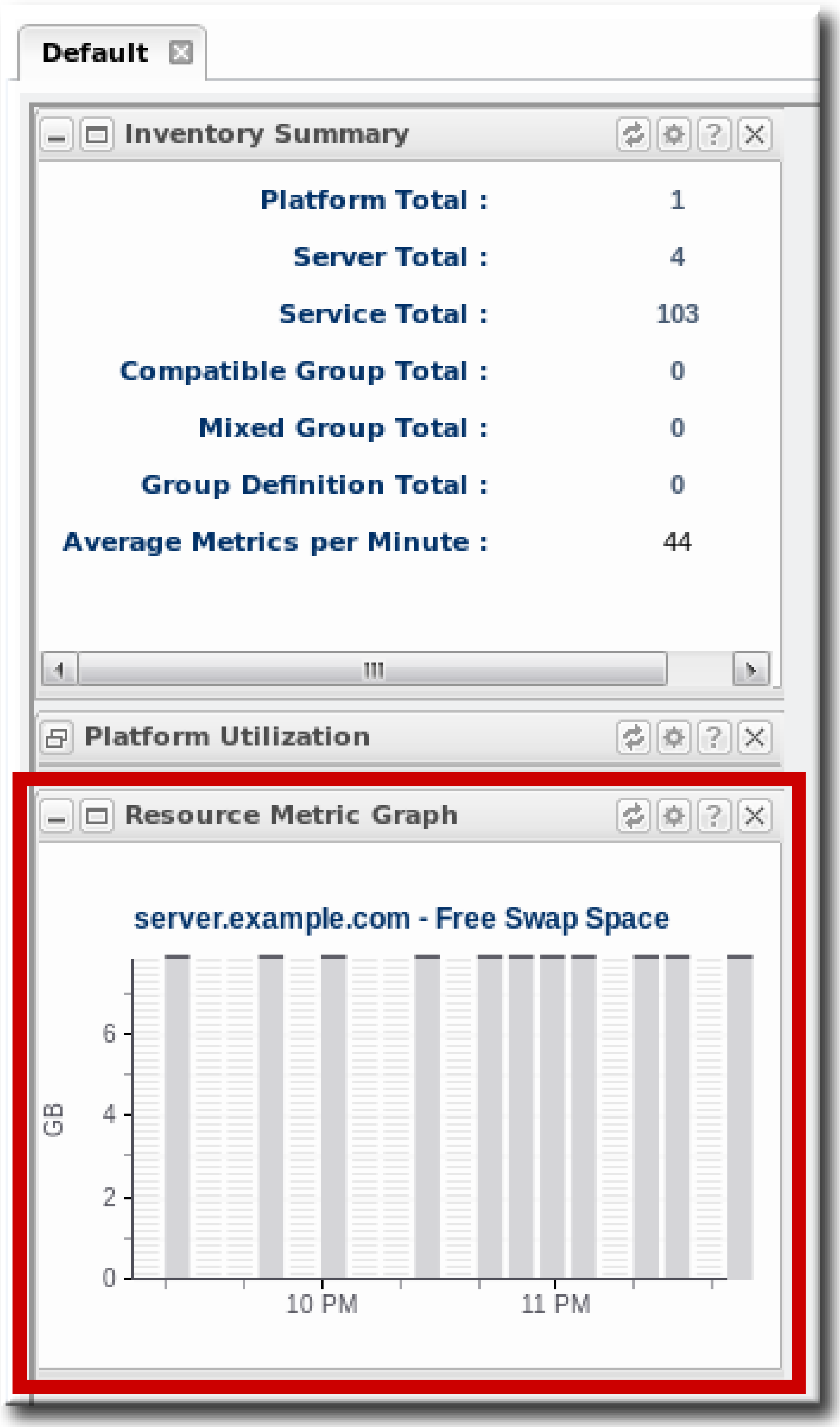


3. In the resource hierarchy on the left, right-click the resource name.

4. Scroll down to the **Measurements** menu item, select the metric from the list, and then select the dashboard to add the chart to.



A chart for that specific metric on that specific resource is automatically added to the Dashboard that was selected.


Default

Inventory Summary

Platform Total :	1
Server Total :	4
Service Total :	103
Compatible Group Total :	0
Mixed Group Total :	0
Group Definition Total :	0
Average Metrics per Minute :	44

Platform Utilization

Resource Metric Graph

17.2. SUMMARY TIMELINES

The **Timeline** subtab in the **Summary** tab shows a line chart of all of the activity for the resource (with the exception of metrics collection, which is all under the **Monitoring** tab and charts). The **Timeline** aggregates all configuration changes, inventory changes, drift, events, content and bundle changes, operations, and alerts. Clicking any given point opens up the details for that specific action.

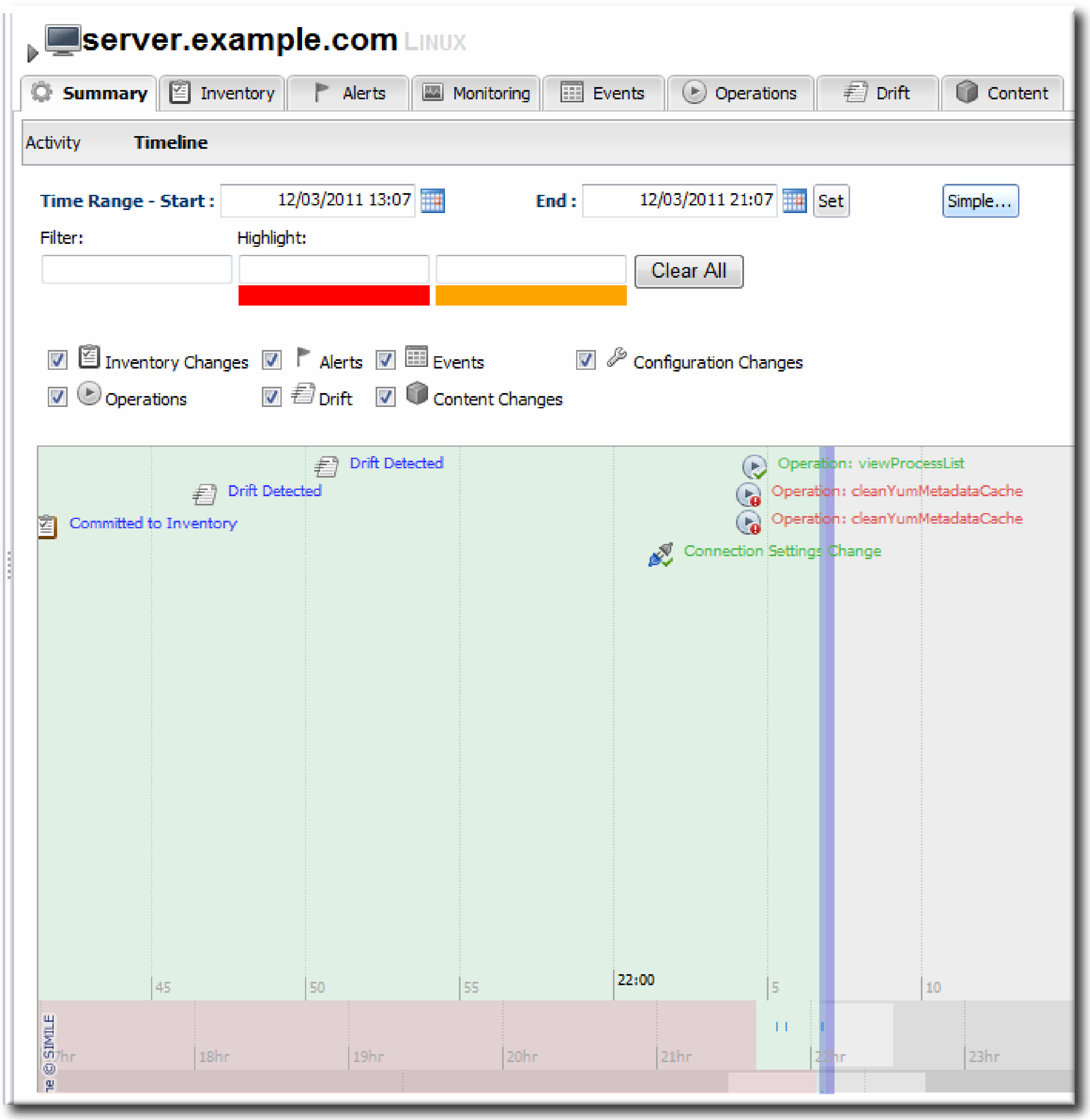


Figure 17.3. Summary Timeline

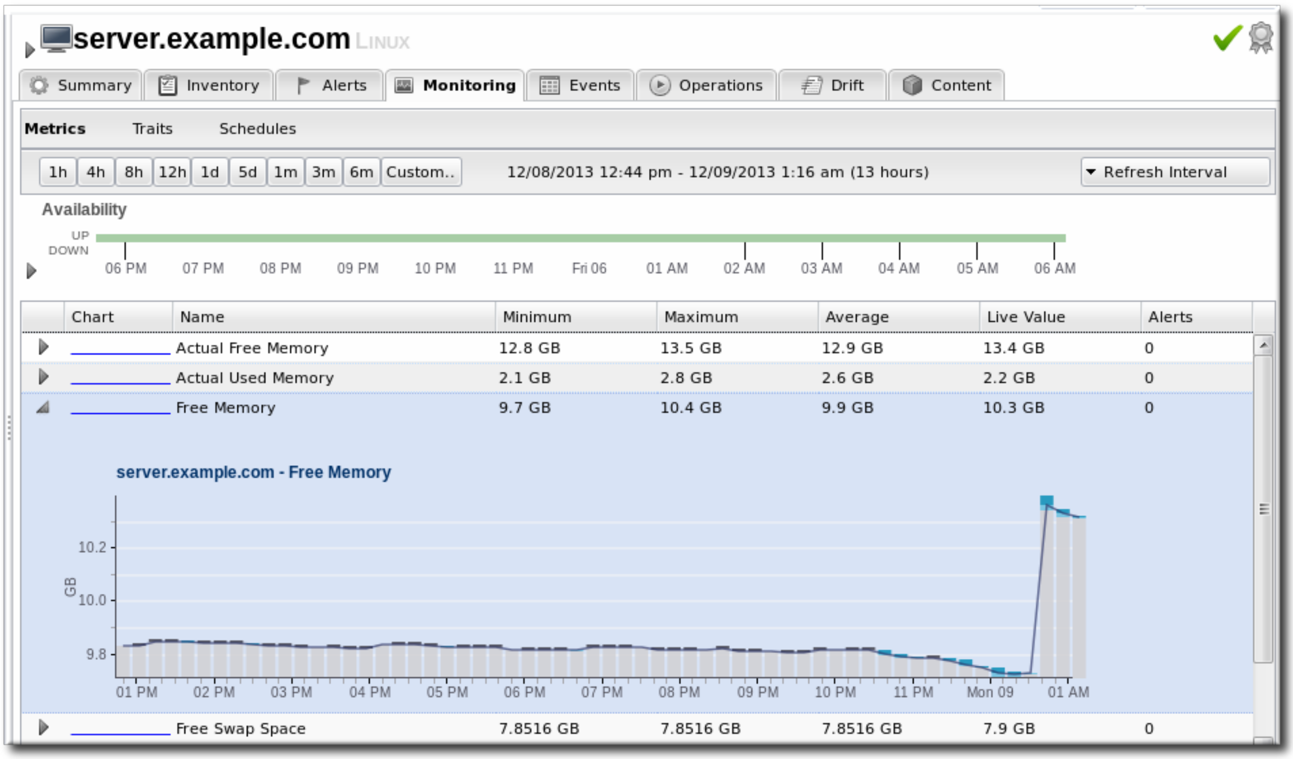
Because all information is on a single timeline, it becomes must easier to correlate incidents and events and to get a better understanding of the overall activity on that resource.

17.3. RESOURCE-LEVEL METRICS CHARTS

The `Monitoring` tab for a resource (or for a compatible group) has a series of different subtabs, each marking a different type of monitoring data. Each data group has its own monitoring charts.

- Section 18.2, “Viewing a Resource's Availability Charts”
- Section 19.2, “Viewing Metrics and Baseline Charts” for graphs and tables of the same metrics information
- Section 20.4, “Viewing Events”
- Section 22.2, “Viewing Traits”

The **Metrics** subtab initially displays a table, laid out with the maximum, minimum, and average values for the baseline period for the metric, the most recent reading, and a trendline. Expanding any given metric shows a graph with the same information, displayed visually.



12/08/2013 12:44 pm - 12/09/2013 1:16 am (13 hours)

Chart	Name	Minimum	Maximum	Average	Live Value	Alerts
	Actual Free Memory	12.8 GB	13.5 GB	12.9 GB	13.4 GB	0
	Actual Used Memory	2.1 GB	2.8 GB	2.6 GB	2.2 GB	0
	Free Memory	9.7 GB	10.4 GB	9.9 GB	10.3 GB	0
	Free Swap Space	7.8516 GB	7.8516 GB	7.8516 GB	7.9 GB	0

Figure 17.4. Metrics Chart

Hovering over any given data point shows the value of that aggregated point (which corresponds to the time period for the size of the graph: the average value for one-sixtieth, 1/60, of the total graphed period; for example, for a 60-hour graph, a data point is one hour).

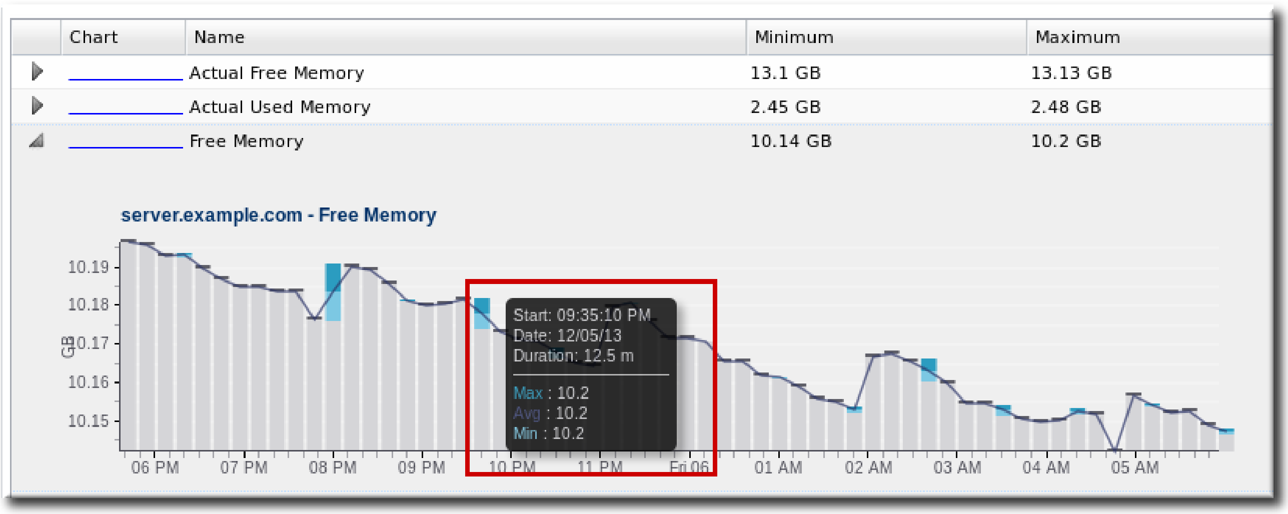


Chart	Name	Minimum	Maximum
	Actual Free Memory	13.1 GB	13.13 GB
	Actual Used Memory	2.45 GB	2.48 GB
	Free Memory	10.14 GB	10.2 GB

```
Start: 09:35:10 PM
Date: 12/05/13
Duration: 12.5 m

Max : 10.2
Avg : 10.2
Min : 10.2
```

Figure 17.5. Hovering over a Data Point

There are buttons at the top of the **Metrics** tab to change the date range for the displayed data. Additionally, it is possible to drag over a portion of the graph and generate a new graph, based on the free selected range.

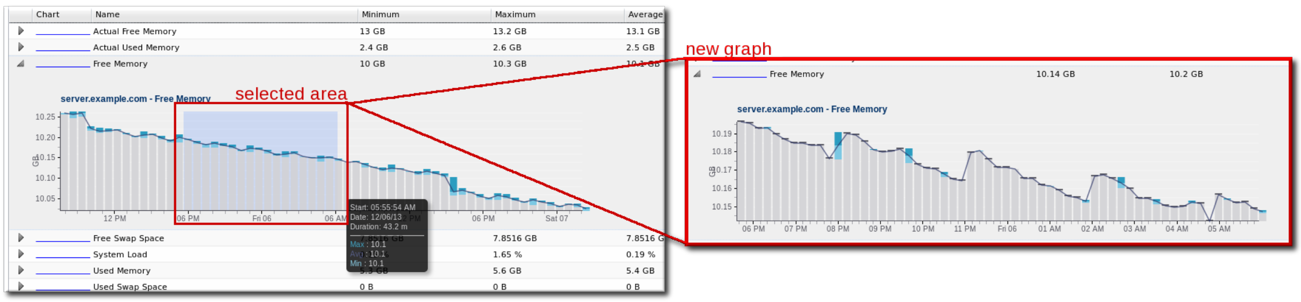
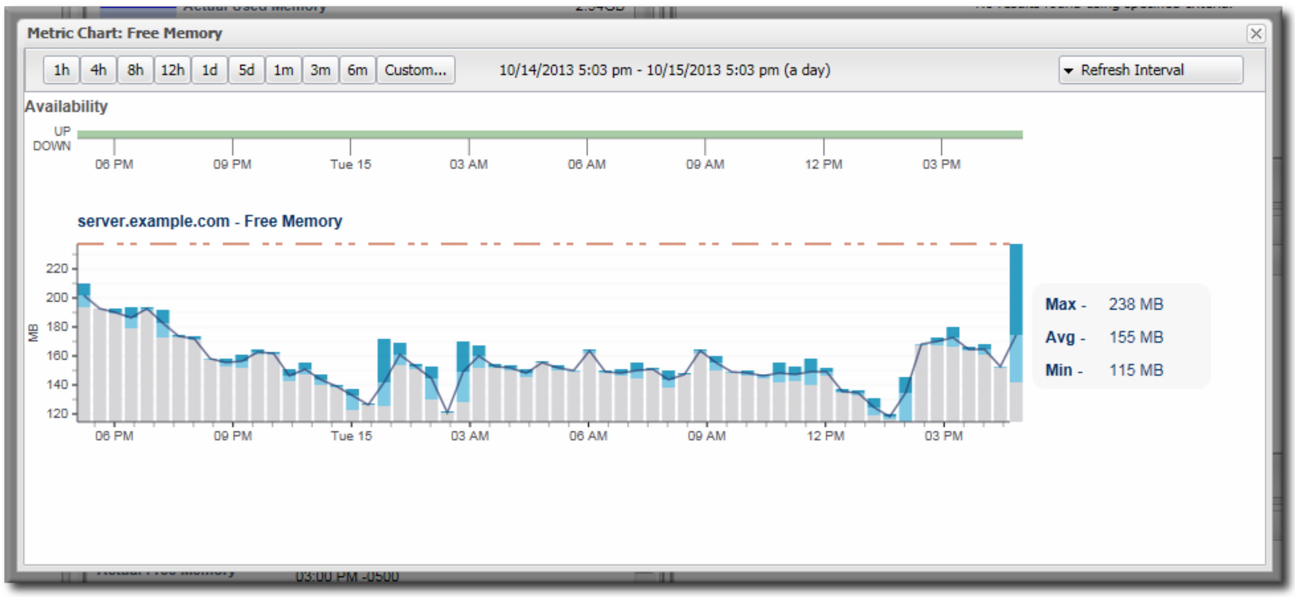


	Chart	Name	Minimum	Maximum	Average
▶		Actual Free Memory	13 GB	13.2 GB	13.1 GB
▶		Actual Used Memory	2.4 GB	2.6 GB	2.5 GB
◢		Free Memory	10 GB	10.3 GB	10.1 GB
▶		Free Swap Space	7.8516 GB	7.8516 GB	7.8516 GB
▶		System Load	[illegible]	1.65 %	0.19 %
▶		Used Memory	5.3 GB	5.6 GB	5.4 GB
▶		Used Swap Space	0 B	0 B	0 B

Figure 17.6. Selecting a Subset of the Graph

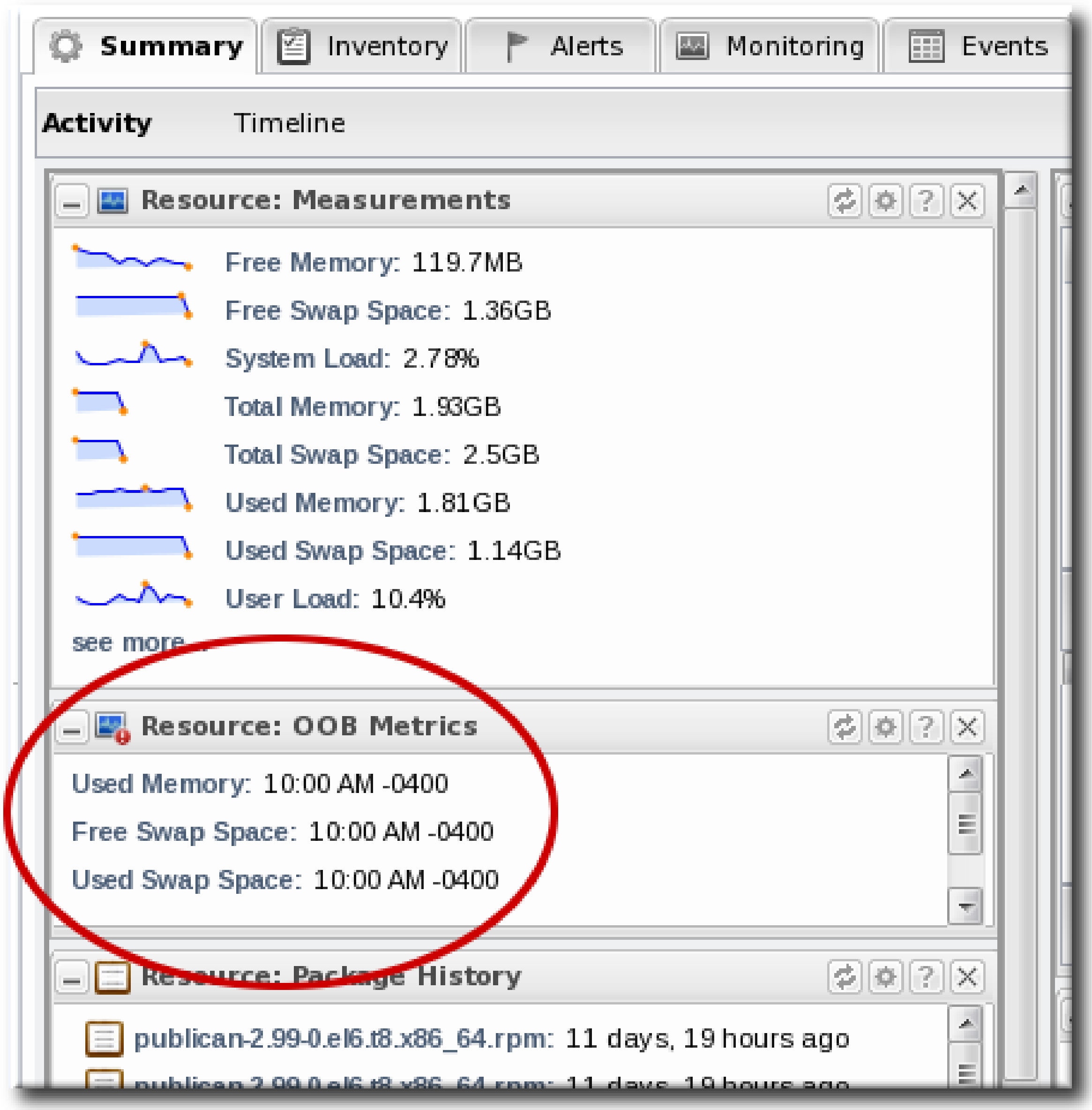
17.4. SUSPECT METRICS REPORT

As described in Section 19.1.4, “Baselines and Out-of-Bounds Metrics”, once metrics have been collected a few times, JBoss ON begins calculating a normal operating range for that specific resource and that specific metric. This creates a range based on the lowest and highest values.



If a metric data point comes in that is outside that normal range, higher or lower, that is a *suspect metric*. It could be a fault of the metric collection or it could indicate a resource problem.

Each individual resource has a portlet on its **Summary** tab which lists suspect, or out-of-bounds, metrics.



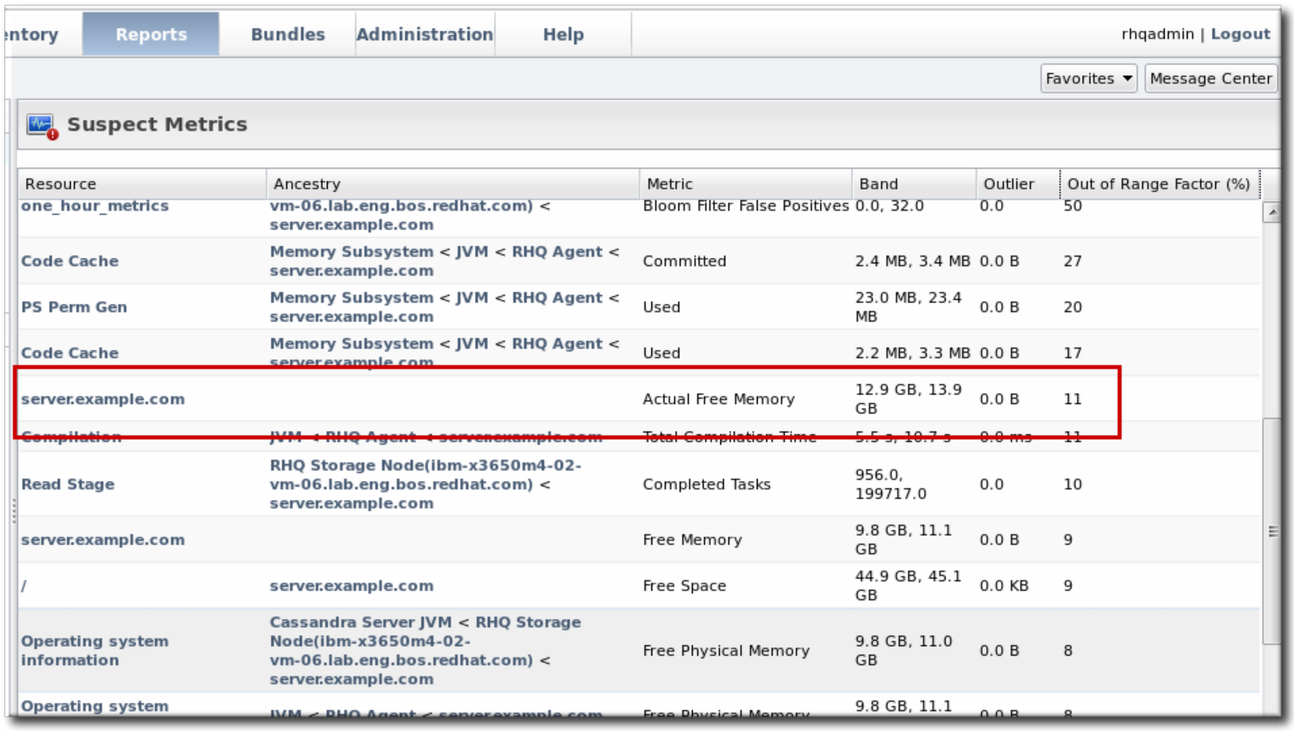
Free Memory: 119.7MB
Free Swap Space: 1.36GB
System Load: 2.78%
Total Memory: 1.93GB
Total Swap Space: 2.5GB
Used Memory: 1.81GB
Used Swap Space: 1.14GB
User Load: 10.4%

Used Memory: 10:00 AM -0400
Free Swap Space: 10:00 AM -0400
Used Swap Space: 10:00 AM -0400

publican-2.99-0.el6.t8.x86_64.rpm: 11 days, 19 hours ago

Figure 17.7. Out of Bounds Portlet

All resources, across the inventory, which have a suspect metric are listed in the `Suspect Metrics` report with the metric, its normal range, its suspect reading, and the factor or percentage of how far outside the metric is from normal readings.



Suspect Metrics

Resource	Ancestry	Metric	Band	Outlier	Out of Range Factor (%)
one_hour_metrics	vm-06.lab.eng.bos.redhat.com) < server.example.com	Bloom Filter False Positives	0.0, 32.0	0.0	50
Code Cache	Memory Subsystem < JVM < RHQ Agent < server.example.com	Committed	2.4 MB, 3.4 MB	0.0 B	27
PS Perm Gen	Memory Subsystem < JVM < RHQ Agent < server.example.com	Used	23.0 MB, 23.4 MB	0.0 B	20
Code Cache	Memory Subsystem < JVM < RHQ Agent < server.example.com	Used	2.2 MB, 3.3 MB	0.0 B	17
server.example.com		Actual Free Memory	12.9 GB, 13.9 GB	0.0 B	11
Compilation	JVM < RHQ Agent < server.example.com	Total Compilation Time	5.5 s, 10.7 s	0.0 ms	11
Read Stage	RHQ Storage Node(ibm-x3650m4-02-vm-06.lab.eng.bos.redhat.com) < server.example.com	Completed Tasks	956.0, 199717.0	0.0	10
server.example.com		Free Memory	9.8 GB, 11.1 GB	0.0 B	9
/	server.example.com	Free Space	44.9 GB, 45.1 GB	0.0 KB	9
Operating system information	Cassandra Server JVM < RHQ Storage Node(ibm-x3650m4-02-vm-06.lab.eng.bos.redhat.com) < server.example.com	Free Physical Memory	9.8 GB, 11.0 GB	0.0 B	8
Operating system	JVM < RHQ Agent < server.example.com	Free Physical Memory	9.8 GB, 11.1	0.0 B	8

Figure 17.8. Suspect Metrics Reports

NOTE

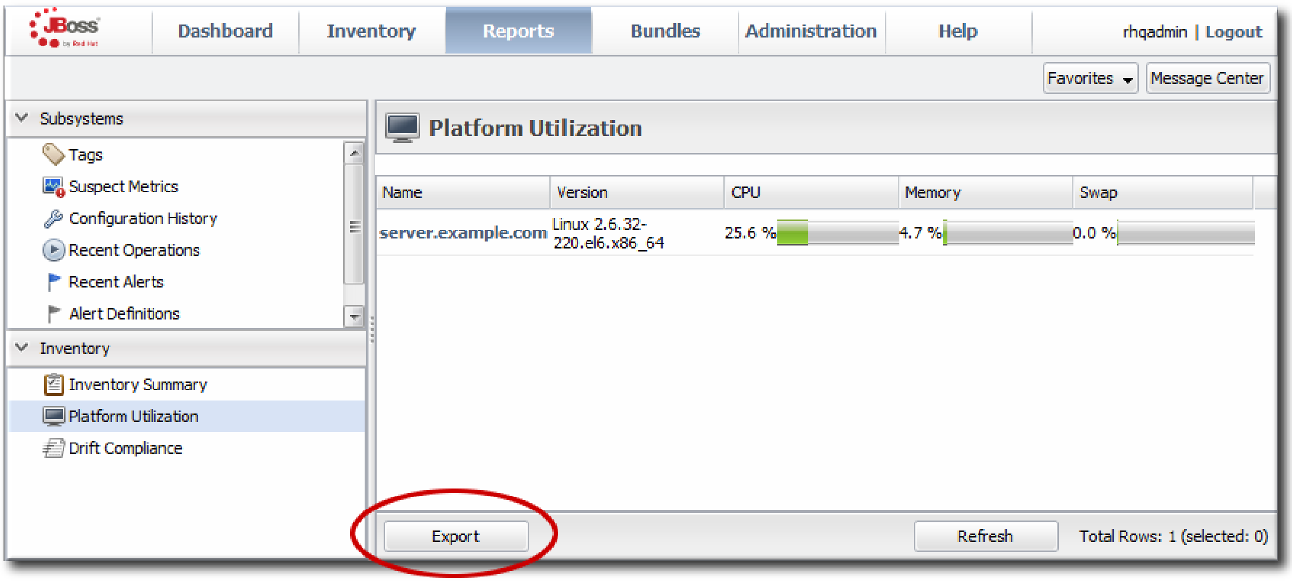
Reports can be exported to CSV, which can be used for office systems or further data manipulation.

To export a report, simply click the **`Export`** button. The report will automatically be downloaded as **`suspectMetrics.csv`**.

17.5. PLATFORM UTILIZATION REPORT

For general infrastructure monitoring, the primary resource is the platform. The **`Platform Utilization`** report shows a very quick snapshot on the health of every platform in the inventory by showing its current system performance, in three metrics:

- Current CPU percentage
- The actual memory usage, based on the available physical memory, buffer, and cache
- Swap



Platform Utilization

Name	Version	CPU	Memory	Swap
server.example.com	Linux 2.6.32-220.el6.x86_64	25.6 %	4.7 %	0.0 %

Figure 17.9. Platform Utilization Report

NOTE

This report can also be added to the main **Dashboard** or a resource-level **Summary** dashboard as a portlet.

There are a couple of caveats. Only *available* platforms are listed. Other platforms in the inventory that are not in an available state are not listed. Also, the utilization is based on the most recent live data, not averages or historical values. It provides an immediate look at the platform resources.

NOTE

Reports can be exported to CSV, which can be used for office systems or further data manipulation.

To export a report, simply click the **Export** button. The report will automatically be downloaded as **platformUtilization.csv**.

CHAPTER 18. AVAILABILITY

One of the most basic elements for monitoring is knowing whether your server or application is running. *Availability* monitoring tells administrators that a certain process is running and minimally responsive.

18.1. CORE "UP AND DOWN" MONITORING

The first question with monitoring is *is the resource running?* A resource's availability is the first thing to check for overall performance, for determining service levels, and for maintaining infrastructure.

Availability (sometimes called *up or down monitoring*) determines whether a resource is *up* or whether it is in some other state.

Up means that the resource is running and that it responds to the agent within a prescribed time.

How availability is determined depends on the resource; it could be checking a process ID or a JVM or something else. Availability for a resource type is defined in its plug-in descriptor. Therefore, the plug-in container is the intermediary between the resource and the agent. The agent checks the plug-in container for resource availability; the container obtains it from the resource component.

Usually, an availability check takes a fraction of a second; for certain types of resources or in certain environments, it could take longer. There is a timeout period for availability scans, set to five (5) seconds by default. If a resource is running and responds to the availability scan within that five-second window, the resource is up.

Because availability — or "up and down" — monitoring is so critical to IT administrators, availability states in JBoss ON are highly visible. Availability is displayed on resource details pages, in every list of resources, in groups, and in monitoring reports. The idea is that it should only take a glance to be able to determine whether your resource is up.

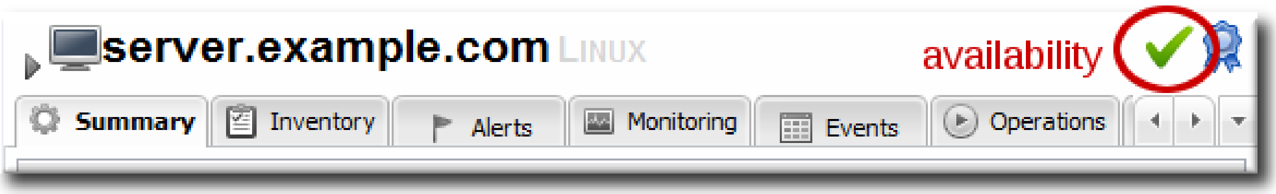


Figure 18.1. Resource Availability

Even though availability is not a true monitoring metric, the `Monitoring > Metrics` page even shows the percentage of time, within the display time period, that the resource has been in an up state. This is because availability (and concomitant uptime) impacts every other metric collected by the agent.

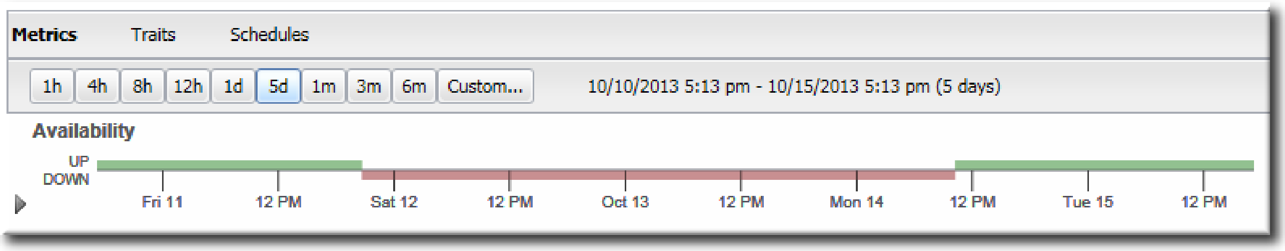


Figure 18.2. Availability Uptime Percentage

NOTE

Often, if a resource shows down availability even when it is running, it is a problem with the connection settings. The agent may not have information it requires, such as a username or new port number, that it requires to connect to the resource. Since the agent cannot connect to the resource, it assumes it is down.

18.1.1. Long Scan Times and Async Availability Collection

Availability scans are performed by a resource plug-in itself, for its defined resource types, and then reported to the plug-in container.

Availability checks are typically very fast, fractions of a second, but there can be situations where an availability check takes longer. The plug-in container limits how long an availability check can run to five seconds, to prevent a rogue plug-in from delaying availability reporting for all other resources managed by the agent.

There can be instances where a certain plug-in or resource type consistently has scans longer than the five-second timeout period.

For custom plug-ins, plug-in writers can configure *asynchronous availability checking*. Basically, with async availability checks, the resource component creates its own, independent thread to run availability checks. Within that thread, the availability checks can take as long as they need to complete. The availability checks can also be run fairly frequently, every minute by default, to make sure that the availability state is current, even if the full check takes longer to complete.

The component caches and then reports the most recent availability result to the plug-in container. That stored last availability can be delivered very quickly, in the fractions of a second that the plug-in container expects.

Async availability checks are implemented through the **`AvailabilityCollectorRunnable`** class in the JBoss ON plug-in API. Details for this class are available in the plug-in API and Writing Custom Plug-ins.

NOTE

It is also possible to address long availability check times by extending the scan timeout period in the agent configuration itself. For example, add a new timeout period to the **`ADDITIONAL_JAVA_OPTIONS`** parameters in the **`rhq-agent-env.sh`** file:

```
RHQ_AGENT_ADDITIONAL_JAVA_OPTS="$RHQ_AGENT_ADDITIONAL_JAVA_OPTS
-Drhq.agent.plugins.availability-scan.timeout=15000"
```

However, that timeout period applies to the *entire* plug-in container, not just one specific, slow-running plug-in. If there are several plug-ins that are running sluggish availability checks, then the availability report may take too long to complete, causing the agent to delay or even miss sending availability reports to the JBoss ON server.

Generally, it is preferable to configure async availability on a custom plug-in, rather than trying to reset the scan interval for all plug-ins.

18.1.2. Synchronous Availability

Availability scans are run on defined schedules, anywhere from every minute to every 20 minutes by default. This means that most availability data is asynchronous — it is displayed in the availability timeline, in reports, and in most of the UI based on the most recent (but not necessarily current) value.

There is one way to get synchronous, near-real time availability information: by viewing the resource's `Monitoring` tab. **As long as the Monitoring tab is open, the availability reading is checked every 15 seconds, rather than their configured collection schedule.** This is as close as possible to real-time availability information.

NOTE

This altered schedule for collecting availability could impact dampening rules for any alerts for a resource.

For example, if availability is scheduled to be checked every 10 minutes with dampening set to fire an alert after three (3) occurrances of a certain state, then an alert could fire after less than a minute if a certain state is read — even though the intent of the alert is to fire only after *half an hour* of the condition persisting.

18.1.3. Availability States

There is a gray area between *up* and *not up*. While a resource may not be up, it may be not up for different reasons. For instance, an agent could have been restarted, so no resource states are known. Or a resource may have been taken offline for maintenance, so no availability reports are being sent.

The different resource states are listed in Table 18.1, “Availability States”.

Table 18.1. Availability States

State	Description	Icon
Available (UP)	The resource is running and responding to availability status checks.	✓
Down	The resource is not responding to availability checks.	!
Unknown	The agent does not have a record of the resource's state. This could be because the resource has been newly added to the inventory and has not had its first availability check or because the agent is down.	?

State	Description	Icon
Disabled	The resource has been administratively marked as unavailable. The resource (in reality) could be running or stopped. Disabling a resource means that the server ignores the availability reports from the agent to prevent unnecessary alerts based on a (known) down or cycling state.	⊘
Mixed (*For groups only*)[a]	The resources in a group have different availability states.	⚠

[a] A similar warning sign can be displayed next to the resource availability at the top of the resource details page. That warning indicates that an error message or suspect metric has been returned for that resource, not that the resource's availability is in a warning state.

18.1.4. Parent-Child States and Backfilling

Availability is assessed from the top of the resource tree downward. For example, if an application server is down, it is safe to assume that all of its dependent webapp children are also down.

This is called *backfilling*. The parent's state is propagated to its children without running additional availability scans for each child. Backfilling can set children to down, unknown, or disabled states.

In some cases, backfilling even includes up states. Some dependent child resources (low priority services that only run if the parent is running) may not even have their own availability assessed independently by default. When a child's availability checking is disabled, the child presumptively uses its parent's state. If the parent is up, those children are assumed to be up.

There is one slight variation on backfilling — if a platform is marked as down. A platform being down is the same as the agent being down. It means that the agent has not reported to the server. There could be a number of reasons for that, apart from any servers or services actually being offline. In this case, the platform (functionally, the agent) is set to down, but its children are set to unknown.

18.1.5. Collection Intervals and Agent Scan Periods

As alluded to, an availability reading is not the same as a metric collection. There are some superficial similarities, mainly in that they both are collected on schedules and that they both relate to resource performance.

Internally, availability and metrics are treated differently. Availability is called through different functions and reported separately, and, more important, availability reports are prioritized higher than other reports sent by the agent, including monitoring reports.

While availability reports are sent as first priority messages, resources themselves have different priorities for availability scans. Higher priority (more critical) resources are, by default, checked for availability more frequently:

- An agent heartbeat ping (analogous to the platform's availability) is sent to the server every minute.
- Server availability is checked every minute.
- Service availability is checked every 10 minutes.

The agent itself runs an availability scan at 30-second intervals. *Not every resource is checked with every scan.* When the agent scan runs, only those resources scheduled to be checked are checked. So, there are functionally two availability schedules working together in tandem, the agent scan interval and the resource collection schedule. For example, if a server is configured with a 60-second interval for availability checks and the agent scan period is 30 seconds, the server is eligible to be checked every two scans. That means that the server is checked *roughly* every 60 seconds, but that is a best effort estimate; if the agent is under a heavy load or if there are a large number of resources, the agent may run its scans longer than every 30 seconds, so the *actual* interval between checks for a specific resource would be longer.

The agent only sends an availability report to the server if there is an availability state change for one of its managed resources.

If an agent goes down suddenly, it shows a down state within five minutes, the (default) agent quiet period. If the agent shuts down gracefully, the JBoss ON server recognizes the state change within about a minute. Once the server recognizes the agent is down, it begins backfilling the states of all of the resources in that agent's inventory (Section 18.1.4, “Parent-Child States and Backfilling”).

Down servers typically record a down state between one and two minutes after going down. This is not exactly real-time, but it is close enough for most infrastructure to be able to establish a reliable baseline of performance and even calculate service levels and uptime. A short window of 90 seconds can catch most resource cycling.

The default agent scan interval is 30 seconds, but, depending on a resource schedule, it could be over 10 minutes before some services are detected as down. If an administrator suspects that there has been a state change, it is possible to force an immediate availability scan for all resources for the agent through the interactive agent prompt:

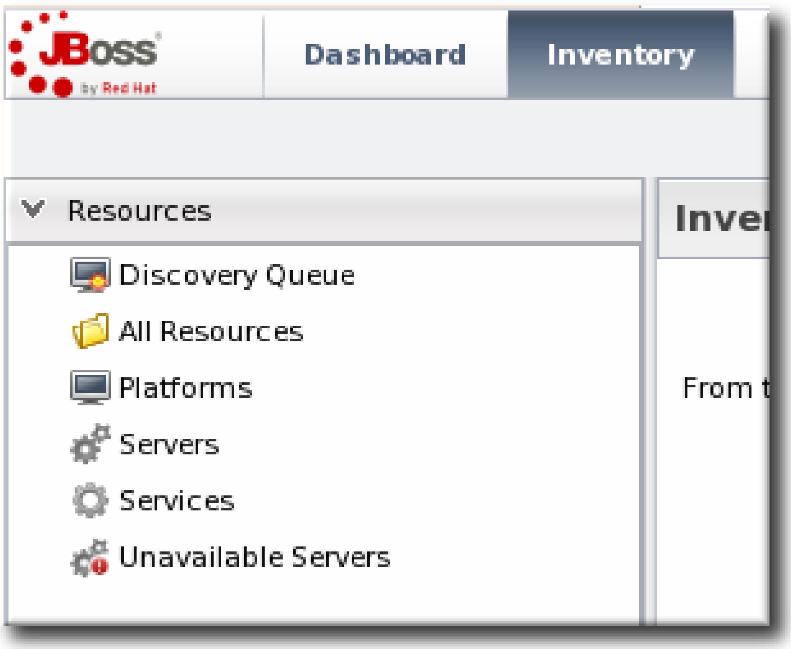
```
> avail -- force
```

Using simply the **`avail`** command runs the check for the next scheduled resources, not all resources.

Additionally, resource plug-ins can be written so that any operation which could cause a state change (such as start, stop, and restart operations) automatically requests an availability check for the resource when the operation ends.

18.2. VIEWING A RESOURCE'S AVAILABILITY CHARTS

1. Click the **`Inventory`** tab in the top menu.
2. Select the resource category, such as servers or services, in the **`Resources`** menu table on the left. Then browse or search for the resource.

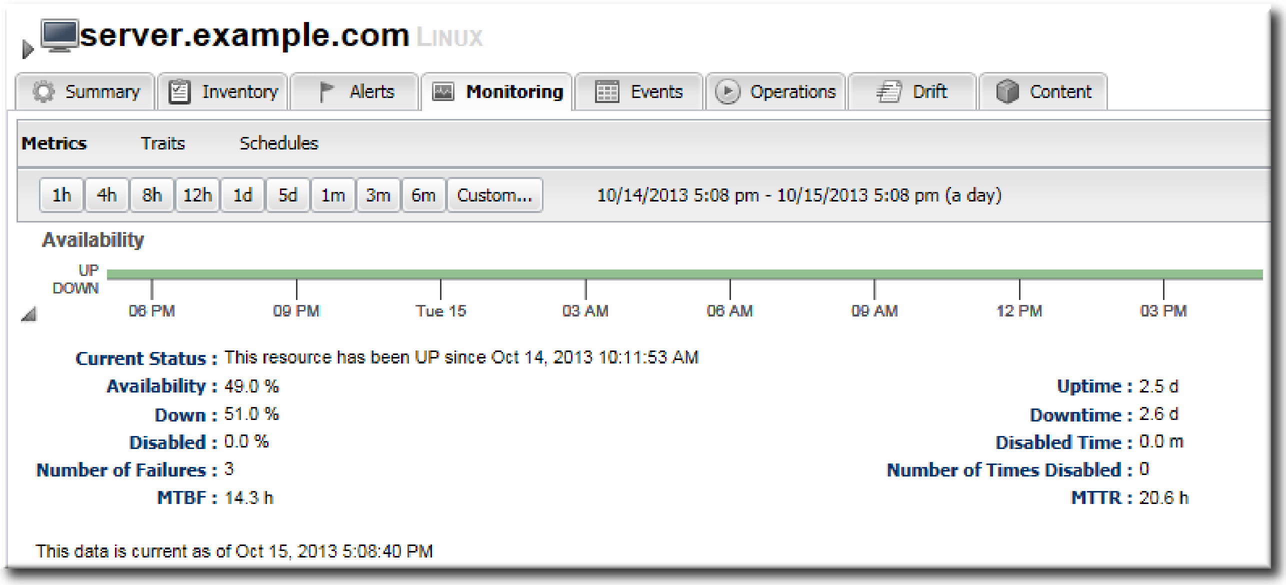


3. Click the name of the resource in the list.

4. Open the resource's **`Monitoring`** tab.

5. Click the **`Availability`** subtab.

The **`Availability`** chart for a resource shows when, and for how long, a resource changes states. This includes timestamps of whenever the availability changes and total counts of how much time the resource spends in the up and down states.



10/14/2013 5:08 pm - 10/15/2013 5:08 pm (a day)

Current Status : This resource has been UP since Oct 14, 2013 10:11:53 AM

Availability :	49.0 %	**Uptime :**	2.5 d
Down :	51.0 %	**Downtime :**	2.6 d
Disabled :	0.0 %	**Disabled Time :**	0.0 m
Number of Failures :	3	**Number of Times Disabled :**	0
MTBF :	14.3 h	**MTTR :**	20.6 h

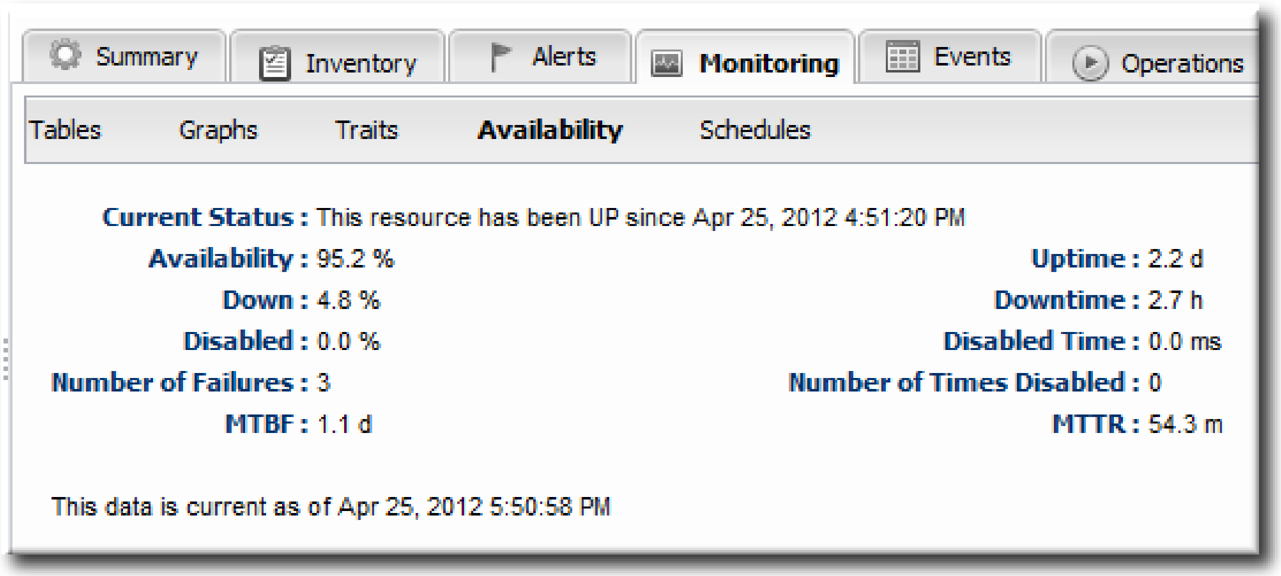
This data is current as of Oct 15, 2013 5:08:40 PM

Figure 18.3. Availability Charts

18.3. *DETAILED DISCUSSION:* AVAILABILITY DURATION AND PERFORMANCE

Availability as a monitoring mechanism has two important facets: the immediate effect of when it changes and then the historic perspective on how changes in availability reflect resource performance.

An historic perspective introduces the idea of *availability duration*. How long was a resource in a particular state? How often does it change?



Current Status : This resource has been UP since Apr 25, 2012 4:51:20 PM

Availability :	95.2 %	Uptime :	2.2 d
Down :	4.8 %	Downtime :	2.7 h
Disabled :	0.0 %	Disabled Time :	0.0 ms
Number of Failures :	3	Number of Times Disabled :	0
MTBF :	1.1 d	MTTR :	54.3 m

This data is current as of Apr 25, 2012 5:50:58 PM

Figure 18.4. Availability Counts

The idea of availability duration is important to get an accurate picture of how a resource is performing. There are several ways that JBoss ON breaks out that information:

- Total time in up, down, and disabled states
- Percentage of time time in up, down, and disabled states
- The number of times the resource has been in a down or disabled state
- The mean time between failures (MTBF) and mean time to recovery (MTTR)

NOTE

Unknown states are not included in calculating the resource's overall availability history.

The last element is particularly important in assessing the resource's performance in light of its availability. The *mean time between failures* is the time between when a resource comes up and when it next goes down – it is the mean[4] of all of its up periods. This gives an idea of how stable a system is. The *mean time to recovery* gives an idea of how long the resource stays down, which indicates its resilience or fault tolerance. A low MTBF and high MTTR indicate some potential maintenance problems or application instability on a resource.

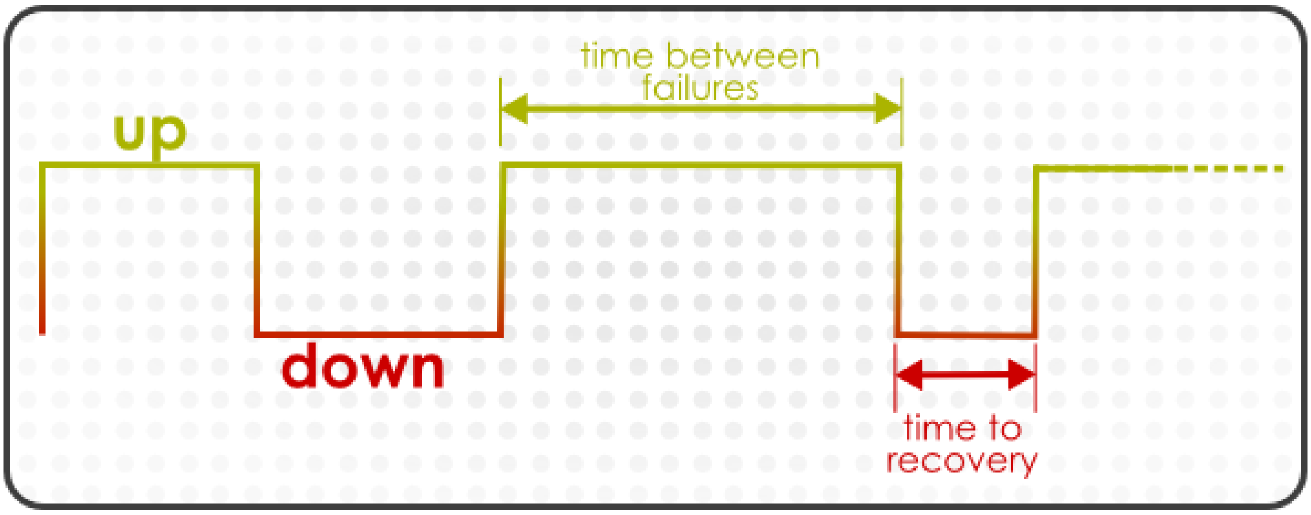


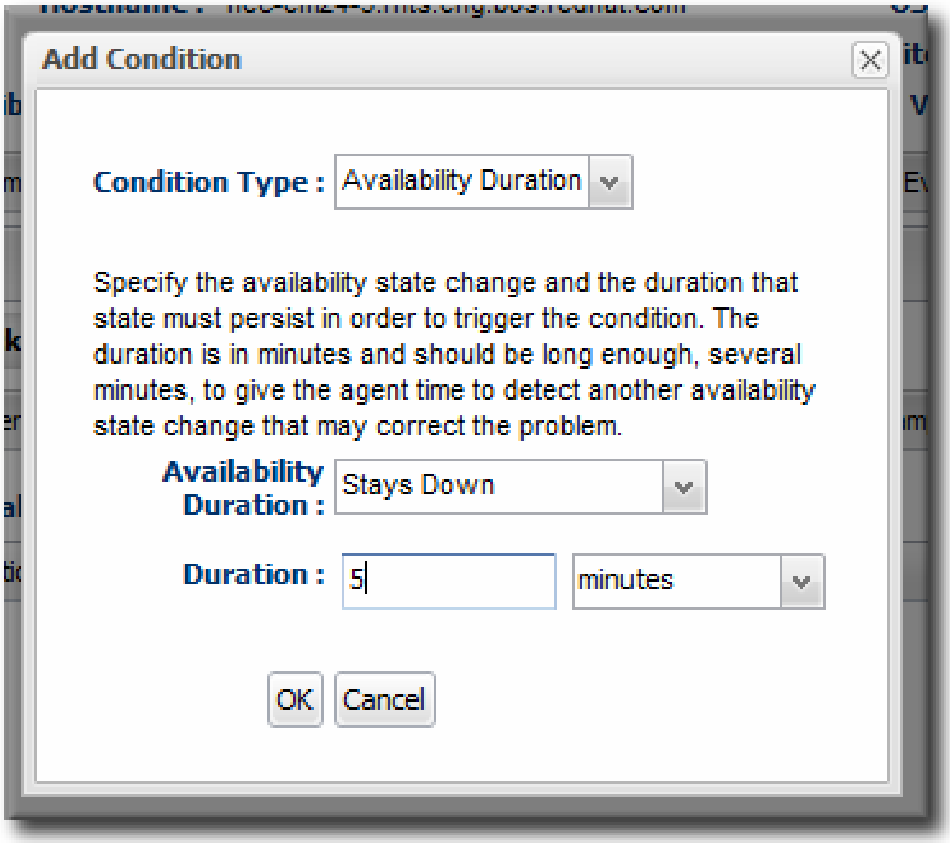
Figure 18.5. Up and Down Monitoring

From a monitoring perspective, the historic perspective is critical, particularly when planning equipment replacements and upgrades.

From an alerting perspective — from an immediate response perspective — only availability changes matter.

The first and most obvious alert condition issues an alert based solely on a state change.

However, resources can cycle or can have a few seconds or minutes where they are inaccessible but that doesn't affect the overall performance of the resource or of whatever function it performs. A resource hits a certain state and has to stay there for a certain amount of time before the state becomes important.


Add Condition

Condition Type : Availability Duration

Specify the availability state change and the duration that state must persist in order to trigger the condition. The duration is in minutes and should be long enough, several minutes, to give the agent time to detect another availability state change that may correct the problem.

Availability Duration : Stays Down

Duration : 5 minutes

Figure 18.6. Availability Duration Alert

NOTE

An availability alert does not lend itself to dampening, because the state changes and then stays, such as an availability alert that fires when the resource changes to a down state. If a resource is cycling, it may go down and up several times, each time triggering a new alert, but it may all be related to the same performance issue on the resource.

Instead of dampening, a disable setting on the alert will fire the alert once, then disable that alert definition until it is acknowledged by an administrator, as described in Section 25.2.5, "*Detailed Discussion:* Automatically Disabling and Recovering Alerts" . (In this case, do not set a corresponding recover setting; otherwise, if the resource is cycling, every UP reading would reset the alert and then the next DOWN report would fire another notification — essentially undoing the dampening effect of disabling the alert until acknowledgment.)

18.4. *DETAILED DISCUSSION*: "NOT UP" ALERT CONDITIONS

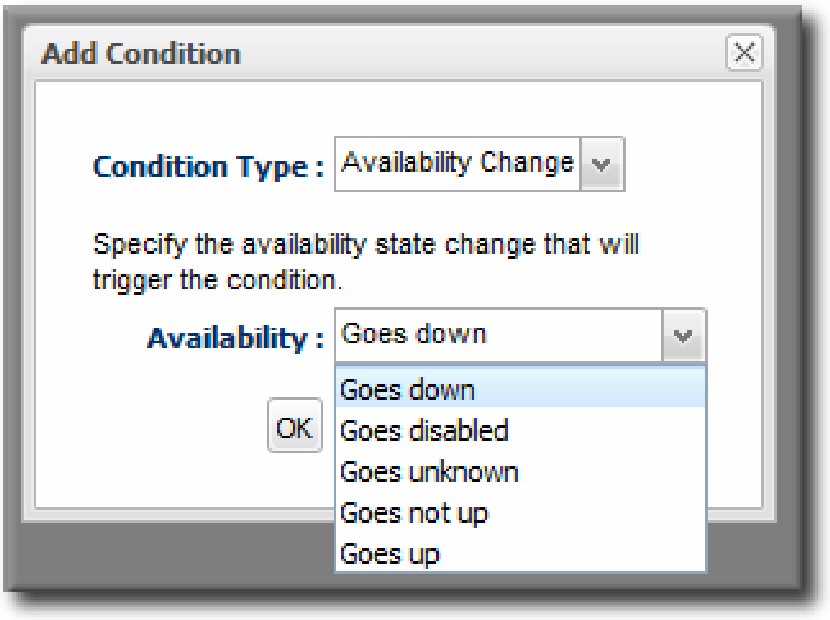
There are four possible availability states for a resource:

- Up
- Down
- Unknown
- Disabled

These are summarized in Section 18.1.3, "Availability States".

Since one of the core monitoring factors for a resource is knowing its availability, alerts can be defined on any availability state change.

Generally, the condition can be set to send an alert on any explicit state. For example, a *goes down* condition alerts only when the availability state changes to DOWN. Any other state change is ignored.


Add Condition

Condition Type : Availability Change

Specify the availability state change that will trigger the condition.

Availability : Goes down

Figure 18.7. Availability Change Conditions

For critical platforms or resources, however, *any* change in availability other than UP may need to trigger an alert. Even known state changes like DISABLED.

The *goes not up* condition triggers an alert if there is a change to any availability state other than UP, so it is a logical OR combination of DOWN, UNKNOWN, and DISABLED conditions.

NOTE

Availability change conditions are well suited to using *recovery alerts*. When a resource goes down (or not up) an alert can fire that informs the administrators and then enables (or recovers) a companion alert that will inform them when the resource is available again.

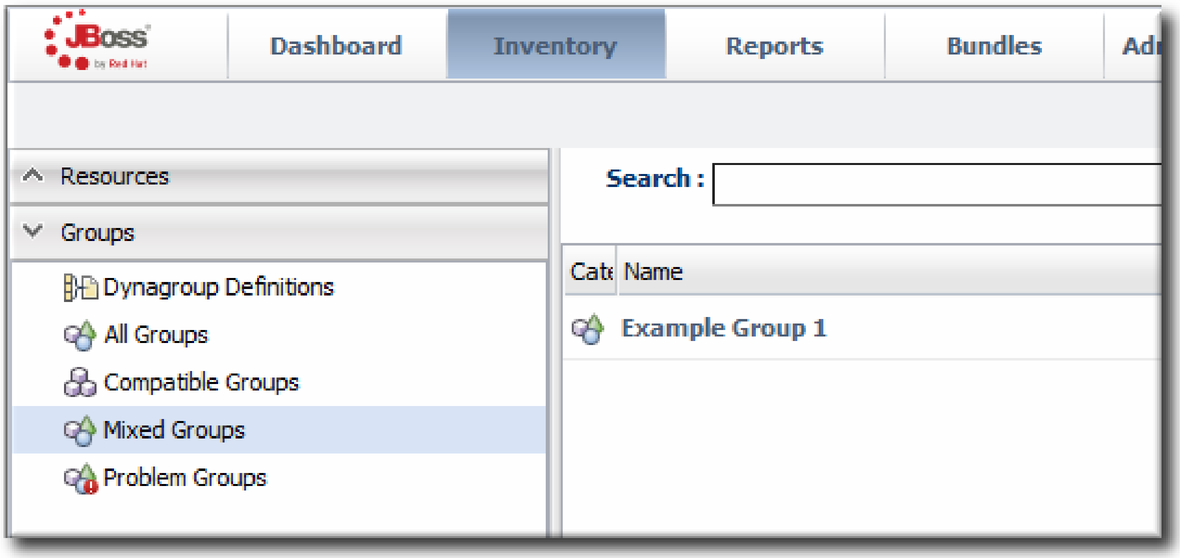
See Section 25.2.5, "*Detailed Discussion:* Automatically Disabling and Recovering Alerts" for more information.

18.5. VIEWING GROUP AVAILABILITY

To view group availability:

1. Click the **`Inventory`** tab in the top menu.

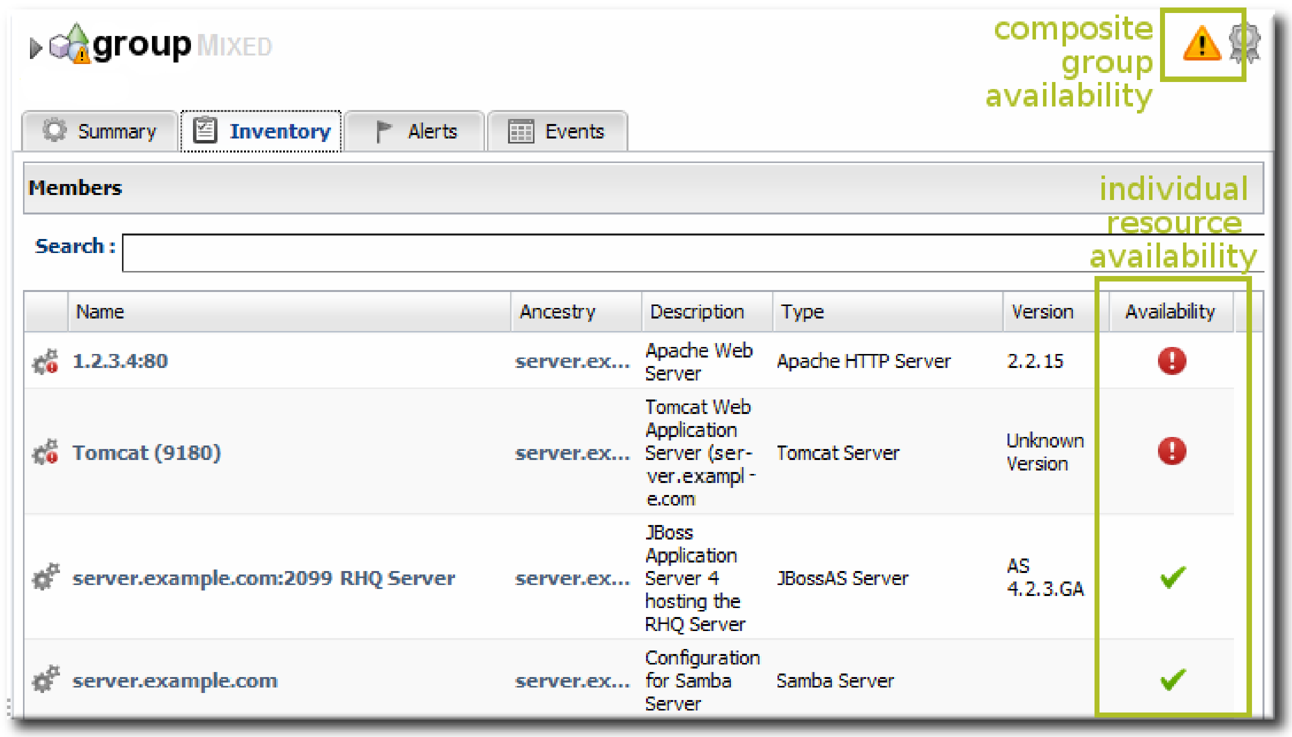
2. Select the compatible or mixed groups item in the **`Groups`** menu on the left.



3. Click the name of the group.

4. Click the **`Inventory`** tab for the group.

Group availability is a composite of the states of its member resources. If all resources are in one state or another, the group as a whole is in that state. If the resources are in different states, then the group state is determined based on the mix of resource states.


group MIXED
composite group availability
Summary
Inventory
Alerts
Events
Members
Search :
individual resource availability

	Name	Ancestry	Description	Type	Version	Availability
	1.2.3.4:80	server.ex...	Apache Web Server	Apache HTTP Server	2.2.15	❗
	Tomcat (9180)	server.ex...	Tomcat Web Application Server (ser-ver.exampl - e.com	Tomcat Server	Unknown Version	❗
	server.example.com:2099 RHQ Server	server.ex...	JBoss Application Server 4 hosting the RHQ Server	JBossAS Server	AS 4.2.3.GA	✓
	server.example.com	server.ex...	Configuration for Samba Server	Samba Server		✓

Figure 18.8. Group Availability

NOTE

Availability states are evaluated "top down." If a resource is down, disabled, or unknown, then all of its children are immediately assumed to be in that state, as well.

Table 18.2. Group Availability States

If the Resource States Are	... the Group State Is ...
Empty Group (Unknown)	Empty
All Red (Down)	Red (Down)
Some Down or Unknown	Yellow (Mixed)
Some Orange (Disabled)	Orange (Disabled)
All Green (Up)	Green (Up)

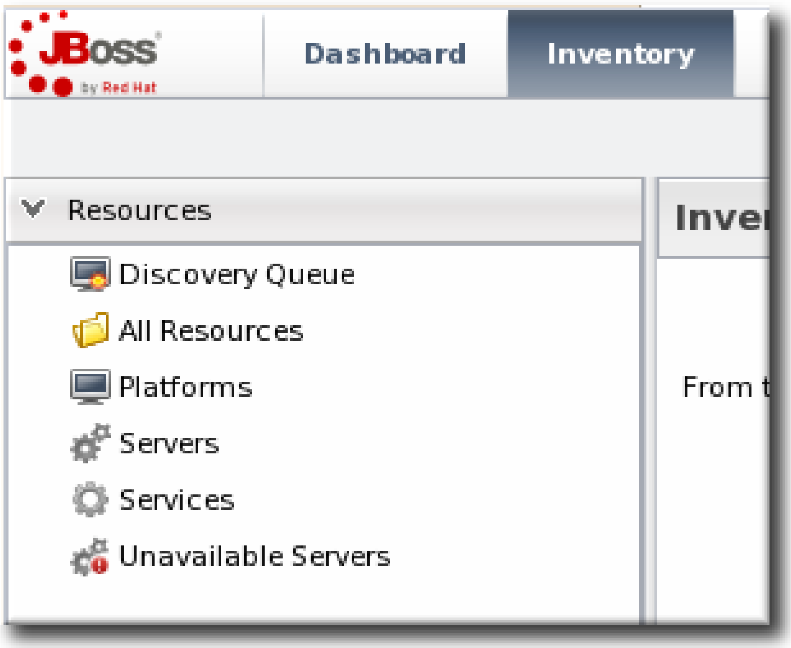
18.6. DISABLING RESOURCES FOR MAINTENANCE

Disabling a resource essentially removes it from the JBoss ON server's view. There can be a lot of reasons why a resource will be taken offline — a machine could be moved to a new colocation facility, the platform may be upgraded, or there could be hardware changes. When an IT administrator knows that a resource will be unavailable, there is no reason to have an availability check which could trigger white noise of unnecessary reports. The resource can be *disabled*, which signals to the JBoss ON server that the resource availability is down (or cycling) and should be ignored.

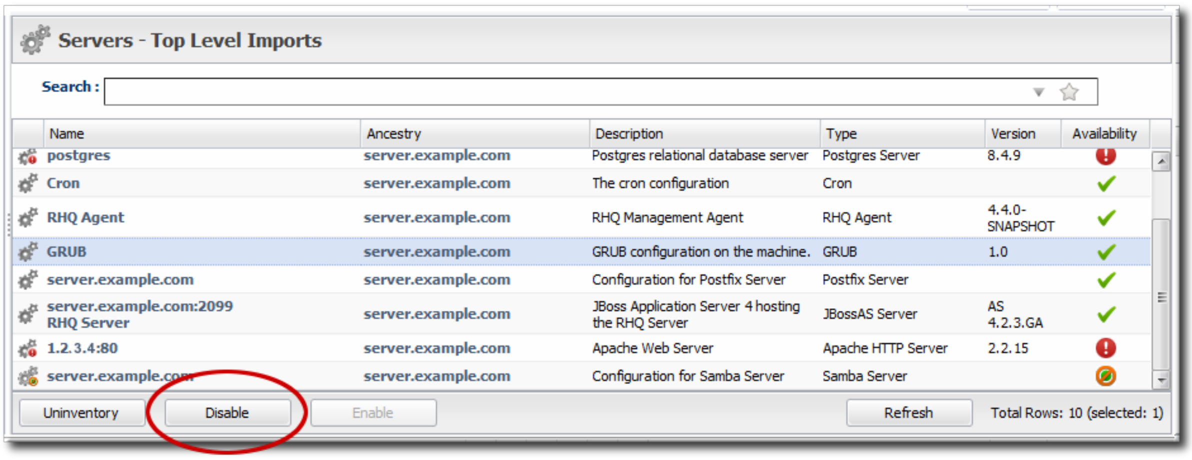
There are two things to remember when disabling a resource:

- If the agent is still up, then the resource availability is still reported. It is just ignored by the JBoss ON server, and is not included in any availability calculations.
- Disabling a parent resource automatically disables all of its children, too.

1. Click the **`Inventory`** tab in the top menu.
2. Select the resource category, such as servers or services, in the **`Resources`** menu table on the left. Then browse or search for the resource.



3. Select the resource in the list.
4. Click the **`Disable`** button at the bottom of the page.


Servers - Top Level Imports

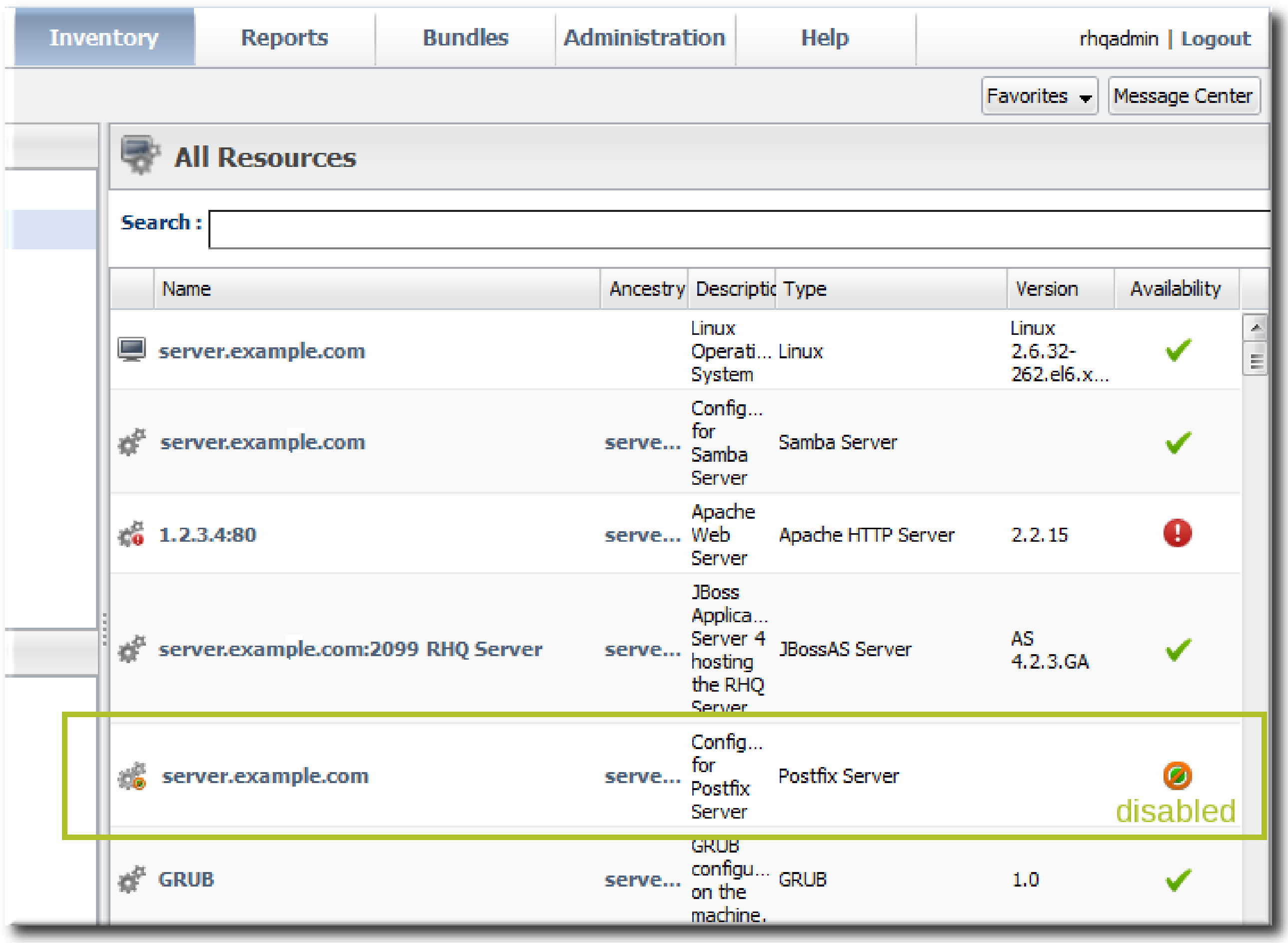
Search :

Name	Ancestry	Description	Type	Version	Availability
postgres	server.example.com	Postgres relational database server	Postgres Server	8.4.9	❗
Cron	server.example.com	The cron configuration	Cron		✓
RHQ Agent	server.example.com	RHQ Management Agent	RHQ Agent	4.4.0-SNAPSHOT	✓
GRUB	server.example.com	GRUB configuration on the machine.	GRUB	1.0	✓
server.example.com	server.example.com	Configuration for Postfix Server	Postfix Server		✓
server.example.com:2099 RHQ Server	server.example.com	JBoss Application Server 4 hosting the RHQ Server	JBossAS Server	AS 4.2.3.GA	✓
1.2.3.4:80	server.example.com	Apache Web Server	Apache HTTP Server	2.2.15	❗
server.example.com	server.example.com	Configuration for Samba Server	Samba Server		●

Total Rows: 10 (selected: 1)

5. When prompted, confirm that the resource should be disabled.

The disabled resource has an orange icon marking its state.



Name	Ancestry	Descriptic	Type	Version	Availability
server.example.com		Linux Operati... System	Linux	Linux 2.6.32-262.el6.x...	✓
server.example.com	serve...	Config... for Samba Server	Samba Server		✓
1.2.3.4:80	serve...	Apache Web Server	Apache HTTP Server	2.2.15	❗
server.example.com:2099 RHQ Server	serve...	JBoss Applica... Server 4 hosting the RHQ Server	JBossAS Server	AS 4.2.3.GA	✓
server.example.com	serve...	Config... for Postfix Server	Postfix Server		⊘
GRUB	serve...	GRUB configu... on the machine.	GRUB	1.0	✓

Figure 18.9. Disabled Resource

NOTE

When the resource is re-enabled, it has an unknown state until the next scheduled availability scan.

18.7. ALLOWING PLUG-INS TO DISABLE AND ENABLE RESOURCES AUTOMATICALLY

Some child or dependent resources may consistently use a disabled state to indicate that the resource is inactive. For example, a managed server in a JBoss EAP 6 domain or a web context under mod_cluster may be offline because it is inactive, and this should be treated differently than being explicitly down. In this case, the parent resource can start or stop the dependent child automatically; when not started, the child is off, but not down.

The resource plug-in itself can automatically disable and enable dependent resources by using the `AvailabilityContext.disable()` and `AvailabilityContext.enable()` methods as part of its availability definition in its component JAR files.

IMPORTANT

Be careful when allowing a resource plug-in to enable or disable a resource automatically. This potentially allows the plug-in to override whatever state the administrator has set.

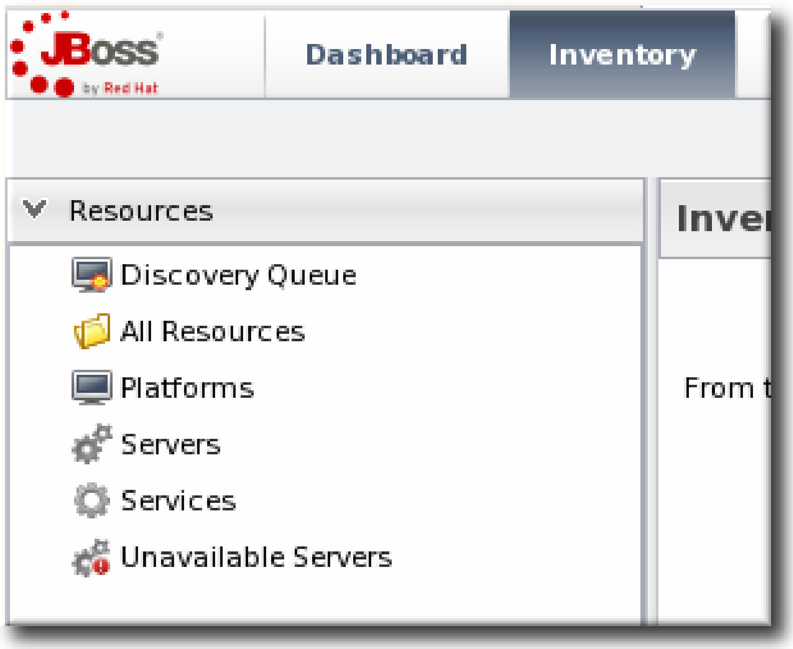
For more information on writing resource plug-ins, see the *Development: Writing Custom Plug-ins*.

18.8. CHANGING THE AVAILABILITY CHECK INTERVAL

While the availability check is not strictly a metric, it does have a collection schedule that can be edited with the other metric collection schedules.

1. Click the **`Inventory`** tab in the top menu.

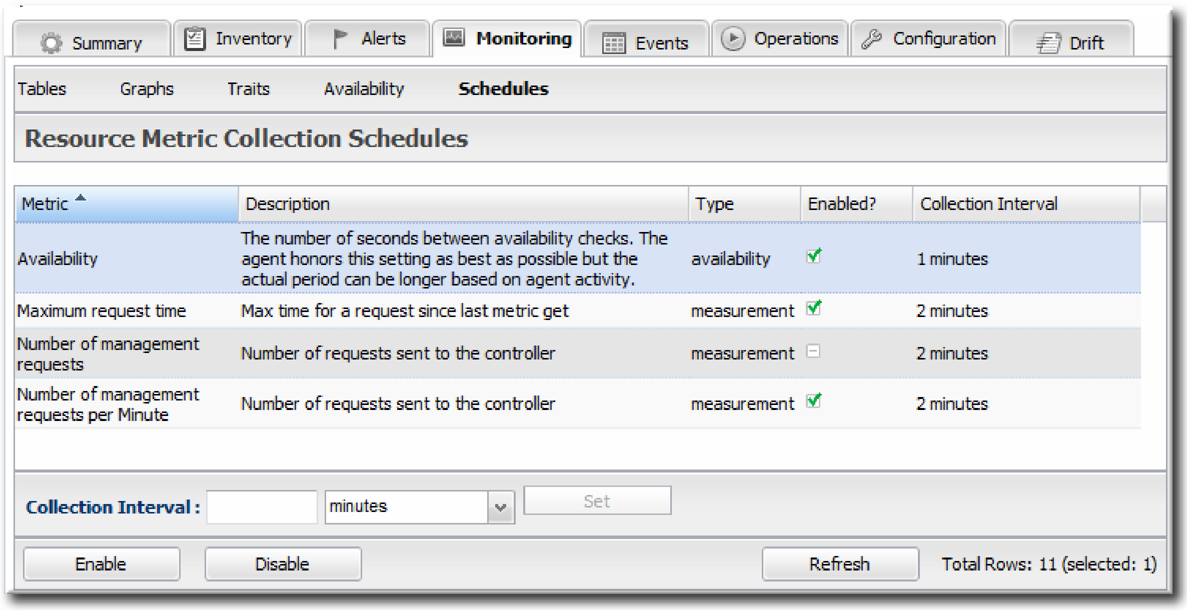
2. Select the resource category, such as servers or services, in the **`Resources`** menu table on the left. Then browse or search for the resource.



3. Click the **`Monitoring`** tab on the resource entry.

4. Click the **`Schedules`** subtab.

5. Select the availability metric, and enter the desired collection period in the **`Collection Interval`** field, with the appropriate time unit (seconds, minutes, or hours).



Resource Metric Collection Schedules

Metric	Description	Type	Enabled?	Collection Interval
Availability	The number of seconds between availability checks. The agent honors this setting as best as possible but the actual period can be longer based on agent activity.	availability	✓	1 minutes
Maximum request time	Max time for a request since last metric get	measurement	✓	2 minutes
Number of management requests	Number of requests sent to the controller	measurement	☐	2 minutes
Number of management requests per Minute	Number of requests sent to the controller	measurement	✓	2 minutes

Collection Interval : minutes Set

Enable Disable Refresh Total Rows: 11 (selected: 1)

NOTE

Availability schedules can be set on compatible groups or resource type templates. Setting it at the group or resource type level changes multiple resources simultaneously.

6. Click **Set**.

18.9. CHANGING THE AGENT'S AVAILABILITY SCAN PERIOD

Since availability is processed on the server, large environments with hundreds of agents and tens of thousands of resources can stress the server and hurt performance. In that case, the default scan interval may be too short, and setting a longer scan interval may improve JBoss ON server performance.

NOTE

When changing core agent or server settings, especially ones that impact JBoss ON performance, contact Red Hat Support Services for assistance.

1. Open the agent configuration file.

```
vim agentRoot/rhq-agent/conf/agent-configuration.xml
```

2. Uncomment the lines in the XML file, and set the new scan time (in seconds).

```
<entry key="rhq.agent.plugins.availability-scan.period-secs"
value="60"/>
```

3. Restart the agent in the foreground of a terminal. Use the **--cleanconfig** option to force the agent to read the new configuration from the configuration file.

```
agentRoot/rhq-agent/bin/rhq-agent.sh --cleanconfig
```

[4] This is mean in the statistical sense. It is the middle data point of all collected uptime lengths.

CHAPTER 19. METRICS AND MEASUREMENTS

Every operating system, application, and server has some mechanism for gaging its performance. A database has page hits and misses, servers have open connection counts, platforms have memory and CPU usage. These performance measurements can be monitored by JBoss Operations Network as *metrics*.

19.1. DIRECT INFORMATION ABOUT RESOURCES

Metrics are a way of measuring a resource's performance or a way of measuring its load. The key word is measurement. A metric is some data point which software exposes, which is relevant to the operations or purpose of that software, that provides insight into the quantifiable behavior of that software.

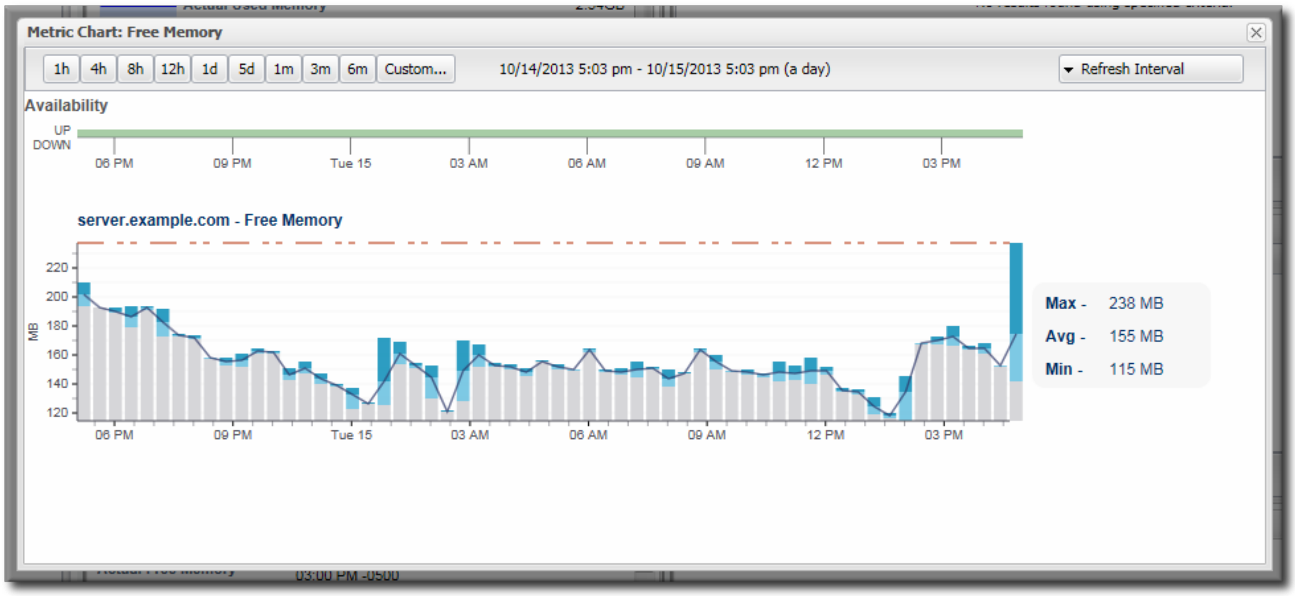


Figure 19.1. Metric Graph

Every type of resource has its own set of metrics, relevant to the resource type. Metrics are defined in the plug-in descriptor for that resource type. The plug-in descriptor lists the types of measurements which are possible and allowed for that resource; that's not necessarily the same thing as the metrics which are actually collected for a resource. Metrics themselves must be enabled (per resource or per metric template) and are then collected on schedule.

19.1.1. Raw Metrics, Displayed Metrics, and Storing Data

The most recent (and unprocessed) reading of metric information is *raw data*. This raw data is stored in the backend server, but it is not the information that is displayed in the web UI.

The information displayed in the web UI is *aggregated data*. In other words, the metrics displayed in JBoss ON and used for monitoring charts are calculated values, not raw data points. Once every hour, a job is run that compresses these metric values into *one hour aggregates*. These aggregates contain the minimum, maximum, and average value of the measured data for the aggregate period. Aggregates are also made for 6-hour and 24-hour windows.

These aggregates are then used to calculate the data displayed in the UI, according to the range of the graph and the size of the display space. The web UI has a limited display space, segmented into 60 x-axis segments. The JBoss ON server averages the raw data to create the data points for whatever the display time period is. For example, if the display range is 60 hours, each x-axis segment is 1-hour wide, and that data point is an average of all readings collected in that 1-hour segment. This aggregation is dynamic, depending on the monitoring window given in the chart views.

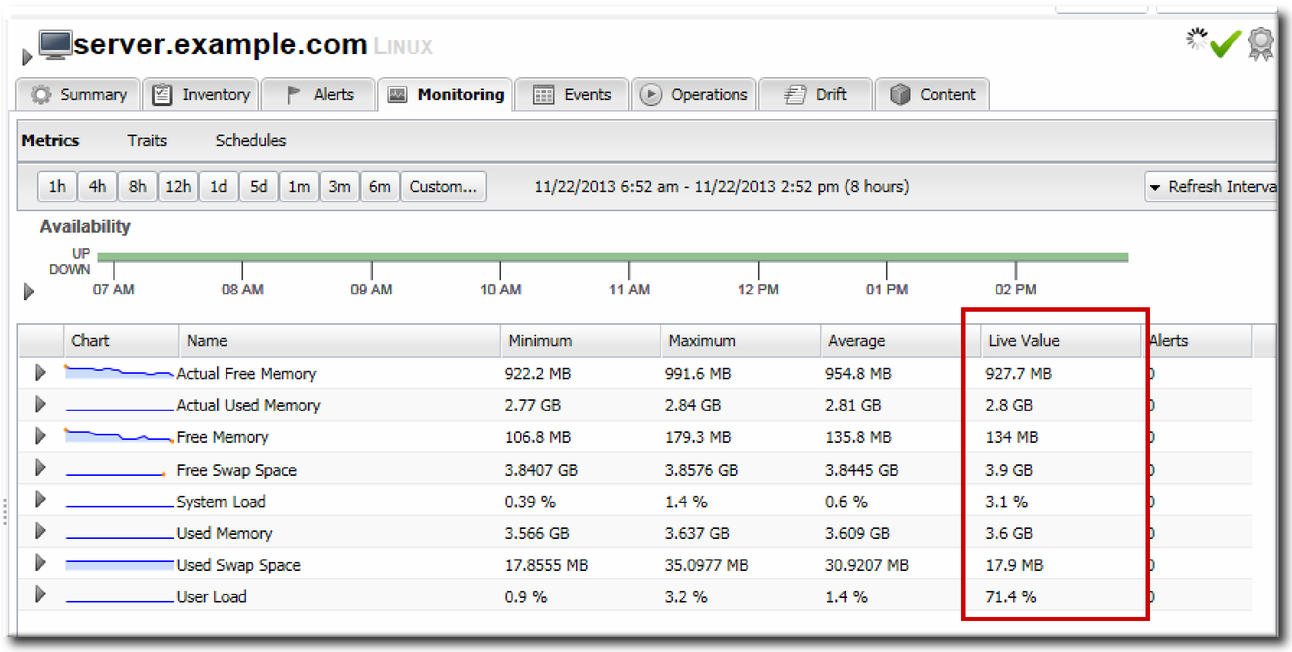
As Section 19.1.4, “Baselines and Out-of-Bounds Metrics” describes, the baseline calculations themselves are aggregates of the raw data, with 1-hour, 6-hour, and 24-hour windows to set minimum, maximum, and average baselines. Unlike the UI aggregates, these aggregated data are calculated and then stored as monitoring data in the server database.

Raw data are only stored for one week, by default, while aggregated values are stored for up to a year. The data storage times are configurable.

19.1.2. Current Values

As Section 19.1.1, “Raw Metrics, Displayed Metrics, and Storing Data” describes, most of the information displayed in JBoss ON is aggregated data. It is the cumulative result of multiple data points gathered over a monitoring period, and then processed and displayed within the given chart.

While JBoss ON is not a real-time monitor, it is continually gathering data. The last collected value is displayed on the `Monitoring` tab of a resource shows the last, raw value for the given metric.



11/22/2013 6:52 am - 11/22/2013 2:52 pm (8 hours)

Chart	Name	Minimum	Maximum	Average	Live Value	Alerts
	Actual Free Memory	922.2 MB	991.6 MB	954.8 MB	927.7 MB	
	Actual Used Memory	2.77 GB	2.84 GB	2.81 GB	2.8 GB	
	Free Memory	106.8 MB	179.3 MB	135.8 MB	134 MB	
	Free Swap Space	3.8407 GB	3.8576 GB	3.8445 GB	3.9 GB	
	System Load	0.39 %	1.4 %	0.6 %	3.1 %	
	Used Memory	3.566 GB	3.637 GB	3.609 GB	3.6 GB	
	Used Swap Space	17.8555 MB	35.0977 MB	30.9207 MB	17.9 MB	
	User Load	0.9 %	3.2 %	1.4 %	71.4 %	

Figure 19.2. Live Values Column

As long as the Monitoring tab is open, the metrics are collected and the live values (and other averaged data) are updated along with the web UI refresh setting, rather than their configured collection schedule. (Availability is checked even more frequently than the refresh schedule, every 15 seconds.) This means that, when viewing the metrics for a resource, the most recent information is always gathered and displayed, and that information is updated as quickly as every minute.

NOTE

This altered schedule for collecting metrics could impact dampening rules for any alerts for a resource.

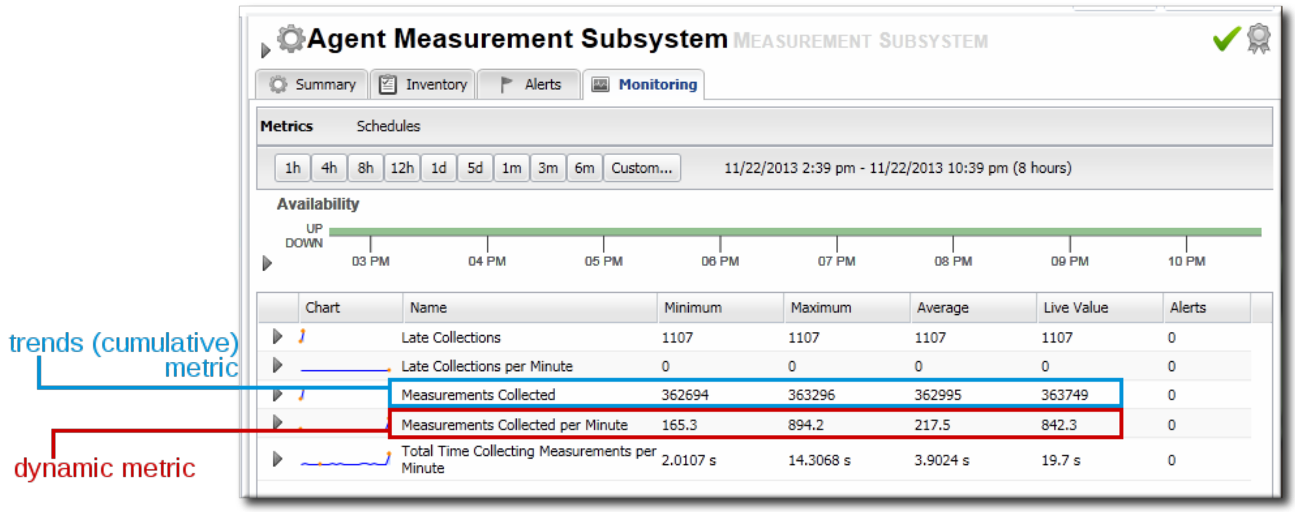
For example, if a metric is scheduled to be collected every 10 minutes with dampening set to fire an alert after three (3) occurrances of a certain condition, and the refresh interval of the UI is one minute, then an alert could fire after three minutes if a certain condition is read — even though the intent of the alert is to fire only after *half an hour* of the condition persisting.

19.1.3. Counting Metrics: Dynamic Values and Trend Values

It may seem obvious, but understanding metrics data includes understanding how the data are counted. There are two types of counted values:

- *Dynamic values* show a momentary and changeable value, a current state. This includes things like the current number of connections to an application server or the CPU usage on a platform.
- *Trend values* are cumulative counts, totals since the resource was started or over its lifetime. These values only progress in a single direction (ususally, but not always, higher)

For example, there are two similar metrics for the agent's measurement subsystem: metrics collected and metrics collected *per minute*. The latter is a dynamic metric, meaning that its value goes up and down depending on whatever number of metrics has actually been collected in the last minute. Metrics collected (the first metric) is a cumulative number; it is the total number of metrics collected by the agent, since it started. So, these two metrics have very different values, despite counting the same data.



11/22/2013 2:39 pm - 11/22/2013 10:39 pm (8 hours)

Chart	Name	Minimum	Maximum	Average	Live Value	Alerts
	Late Collections	1107	1107	1107	1107	0
	Late Collections per Minute	0	0	0	0	0
	Measurements Collected	362694	363296	362995	363749	0
	Measurements Collected per Minute	165.3	894.2	217.5	842.3	0
	Total Time Collecting Measurements per Minute	2.0107 s	14.3068 s	3.9024 s	19.7 s	0

Figure 19.3. Dynamic and Trend Values for Metrics

As Figure 19.3, “Dynamic and Trend Values for Metrics” shows, it is possible to calculate an average for trend data, but that value is meaningless. Likewise, the "minimum" for a trend value is the starting value of the selected time period, while the "maximum" is the last value for the selected time period. Other automatic calculations — such as out-of-bound values and baselines — are also meaningless with trend data, but are valuable with dynamic data.

19.1.4. Baselines and Out-of-Bounds Metrics

After metrics have been collected for a reliable amount of time, JBoss ON automatically calculates a *baseline* for the metric. A baseline is the normal operating range for that metric on that resource. The baseline is caluclated, by default, every three days using the aggregated data. The baseline uses a rolling window of seven days' of data.

Baseline metrics compare changes in actual data against a baseline value. Baselines allow effective trending analysis, SLAs management, and overall application health assessments as a form of fault management.

Baselines allow JBoss ON to identify metric values collected that fall outside (out-of-bounds) of the high and low baselines. Out-of-bounds metrics are reported as *problem metrics*.

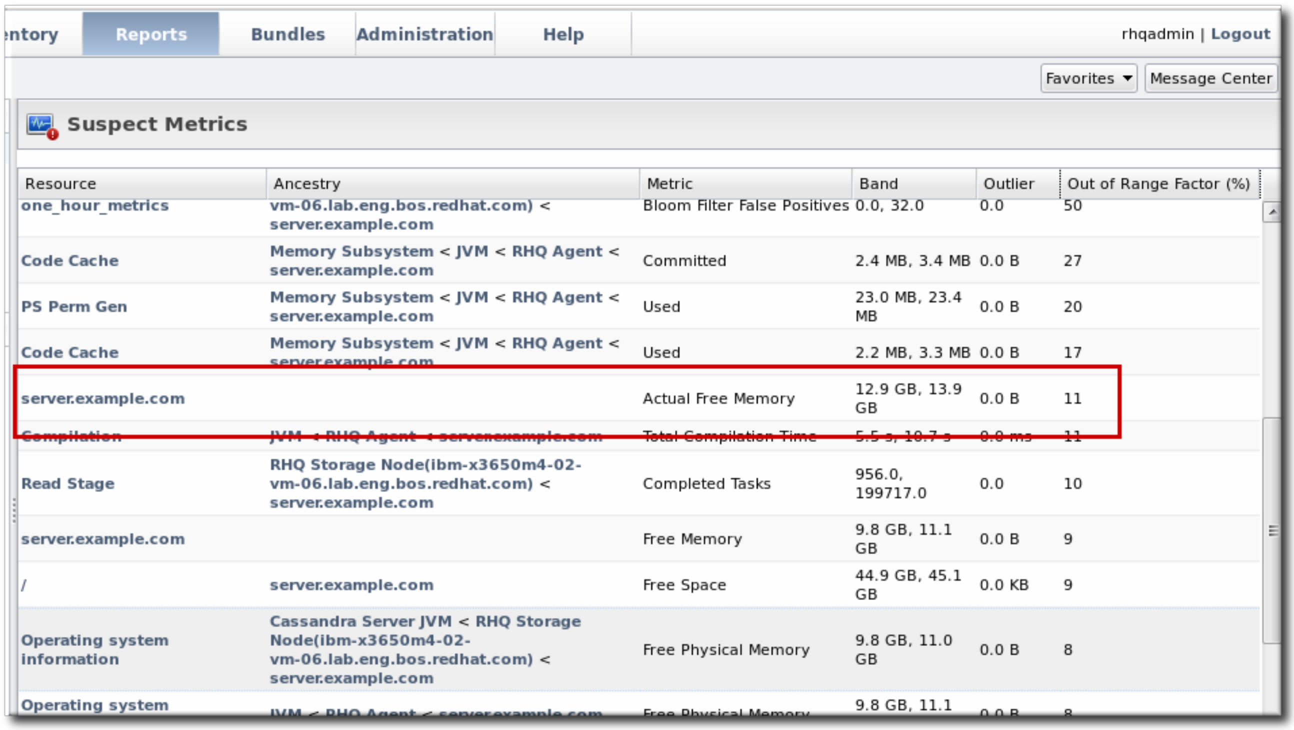
NOTE

When an alert is triggered in response to a metric value, the alerting event is tracked as a problem metric.

If there are no baselines present, because they have not yet been computed or because the metric is a trends metric (meaning it is a cumulative value), no out-of-bounds factors will be calculated.

A baseline has a *bandwidth* that is the difference between its minimum and maximum values. The *difference* is the absolute amount that the problem metric is outside the baseline. To be able to compare out-of-bound values, an *out-of-bounds-factor* is computed by dividing the difference by the bandwidth. This creates a ratio to show comparatively how far out of the normal operation range the problem metric is.

In the `Suspect Metrics` report, the baselines are reported as *minimum,maximum*, and then the out-of-bounds metric is listed as the *outlier*. The difference between the baselines and the outliers is shown as a percentage: the difference of the outlier to its nearest baseline, divided by the baseline bandwidth.



Resource	Ancestry	Metric	Band	Outlier	Out of Range Factor (%)
one_hour_metrics	vm-06.lab.eng.bos.redhat.com) < server.example.com	Bloom Filter False Positives	0.0, 32.0	0.0	50
Code Cache	Memory Subsystem < JVM < RHQ Agent < server.example.com	Committed	2.4 MB, 3.4 MB	0.0 B	27
PS Perm Gen	Memory Subsystem < JVM < RHQ Agent < server.example.com	Used	23.0 MB, 23.4 MB	0.0 B	20
Code Cache	Memory Subsystem < JVM < RHQ Agent < server.example.com	Used	2.2 MB, 3.3 MB	0.0 B	17
server.example.com		Actual Free Memory	12.9 GB, 13.9 GB	0.0 B	11
Compilation	JVM < RHQ Agent < server.example.com	Total Compilation Time	5.5 s, 10.7 s	0.0 ms	11
Read Stage	RHQ Storage Node(ibm-x3650m4-02-vm-06.lab.eng.bos.redhat.com) < server.example.com	Completed Tasks	956.0, 199717.0	0.0	10
server.example.com		Free Memory	9.8 GB, 11.1 GB	0.0 B	9
/	server.example.com	Free Space	44.9 GB, 45.1 GB	0.0 KB	9
Operating system information	Cassandra Server JVM < RHQ Storage Node(ibm-x3650m4-02-vm-06.lab.eng.bos.redhat.com) < server.example.com	Free Physical Memory	9.8 GB, 11.0 GB	0.0 B	8
Operating system	JVM < RHQ Agent < server.example.com	Free Physical Memory	9.8 GB, 11.1	0.0 B	8

Figure 19.4. Suspect Metrics Reports

NOTE

Calculating baselines can sometimes output non-intuitive results, as a band of (1,2) and an outlier value of 3 seems to be less than a band of (100, 200 MB) and an outlier value of 250 MB. The former is actually 100% outside the expected band, while the latter is only 50% outside.

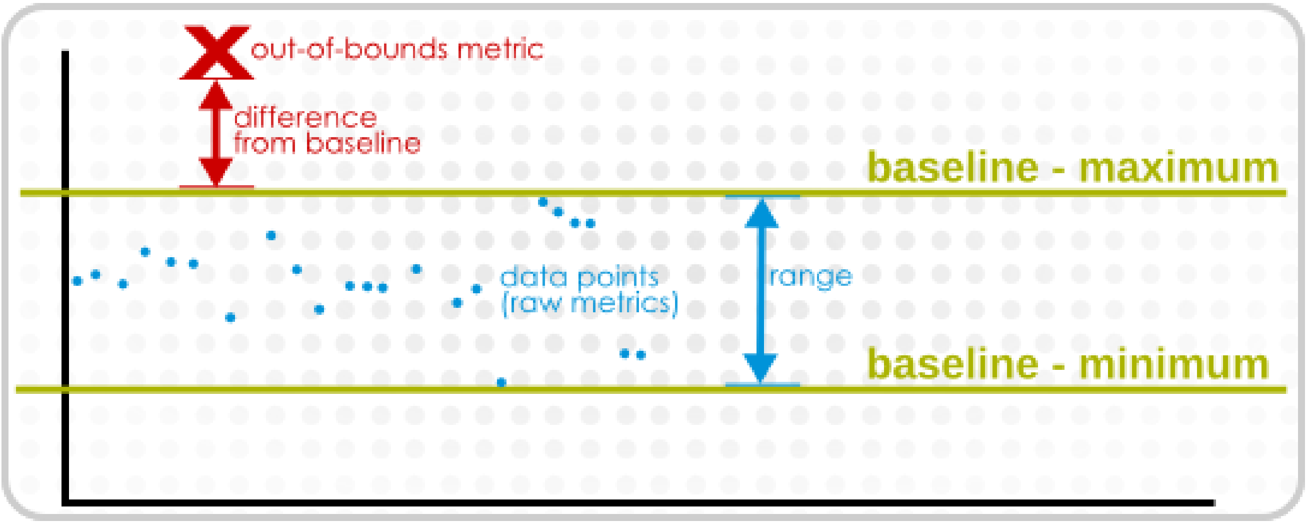


Figure 19.5. Out-of-Bound Factors

Out-of-bounds-factors are recalculated each hour during a calculation job. The job assesses the aggregate and determines if there is a more severe outlier than before. The chart always displays the most severe outlier.

When the baselines for a metric change, all recorded out-of-bounds values become invalid and are removed because the out-of-bounds measurement was computed against an old baseline.

19.1.5. Collection Schedules

The metric collection schedule is defined individually for each metric in the resource type's plug-in descriptor.

There is no rule on how frequently metrics are collected. Default intervals range between 10 minutes and 40 minutes for most metrics. While some metrics are commonly important (like free memory or CPU usage on platforms), the importance of many metrics depends on the general IT and production environments and the resource itself. Set reasonable intervals to collect important metrics with a frequency that adequately reflects the resource's real life performance.

The shortest configurable interval is 30 seconds, although an interval that short should be used sparingly because the volume of metrics reported could impact database performance.

19.1.6. Metric Schedules and Resource Type Templates

Unlike other types of monitoring data which are unique to an resource (availability, events, traits), metrics can be universal for all resources of that type.

Metric collection schedules define whether an allowed metric for a resource is actually enabled and what its collection interval is. A schedule is set at the resource-level, but administrator-defined default settings can be applied to all resources of a type by using *metrics collection templates*.

Templates are a server configuration setting. They define what metrics are active and what the collection schedules are for all resources of a specific type. When templates are used, they supplant whatever default metrics settings are given in the plug-in descriptor. (A metric template only defines whether a metric is enabled and what its interval is — the plug-in descriptor alone defines what metrics are available for a resource type.)

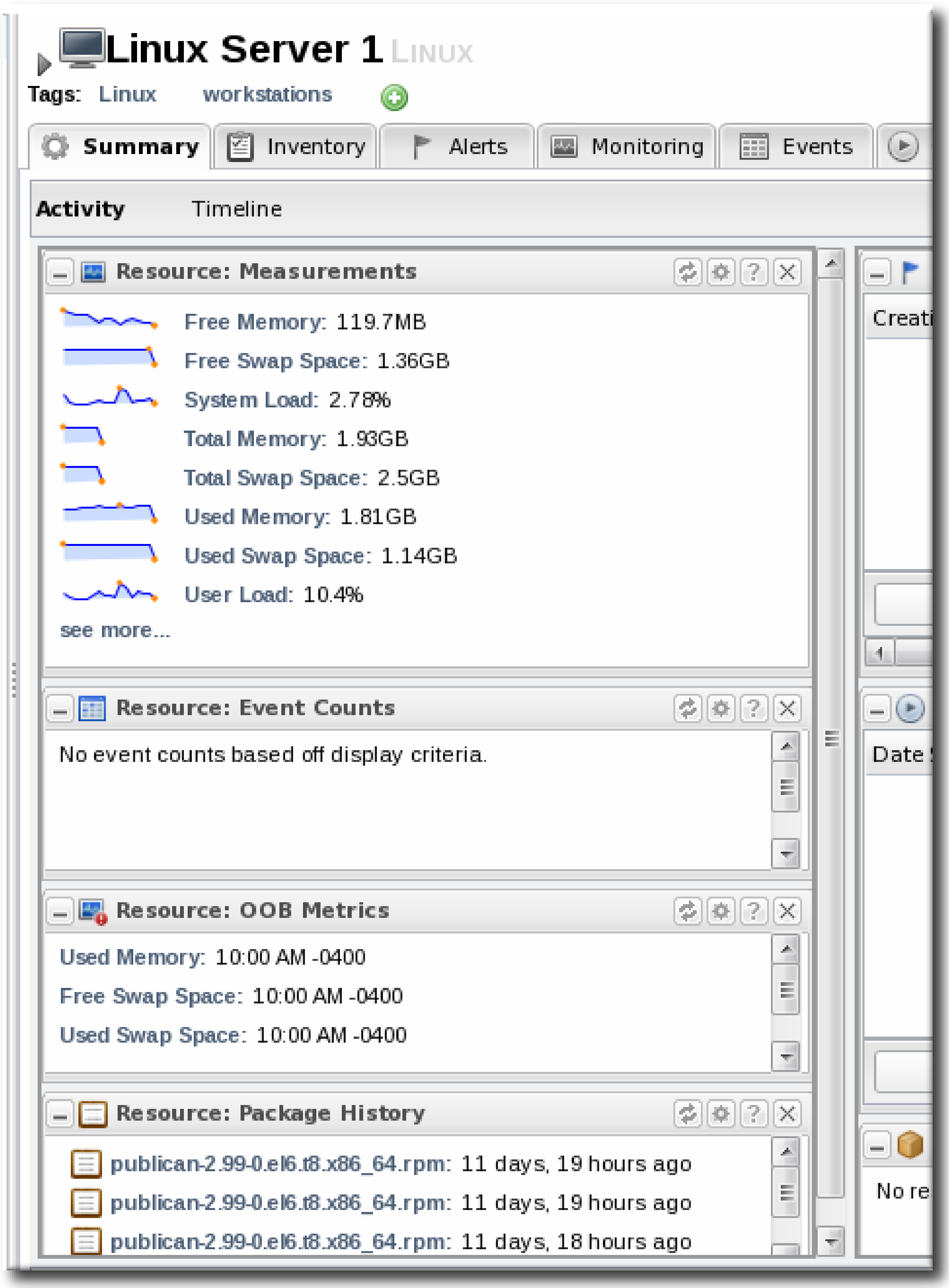
These settings can be overridden at the resource-level, as necessary. Still, metrics collection templates provide a simple way to apply metrics settings consistently across resources and machines.

19.2. VIEWING METRICS AND BASELINE CHARTS

The core of monitoring is the metric information that is collected for a resource. Each resource has different metrics (and these are listed in the *Resource Reference: Monitoring, Operation, and Configuration Options*). Three monitoring charts show the same information, but in different perspectives and different levels of detail:

- The resource-level Summary
- Graphs
- Tables

The `Summary` tab for resources, much like the Dashboard for the entire JBoss ON inventory, has portlets that show different resource information. Most resources have three portlets for measurements, events, and out-of-bound metrics. The `Measurements` portlet has small thumbnail charts that show the trend for the metric, along with the current reading.



Resource: Measurements

Free Memory: 119.7MB
Free Swap Space: 1.36GB
System Load: 2.78%
Total Memory: 1.93GB
Total Swap Space: 2.5GB
Used Memory: 1.81GB
Used Swap Space: 1.14GB
User Load: 10.4%
see more...

Resource: Event Counts

No event counts based off display criteria.

Resource: OOB Metrics

Used Memory: 10:00 AM -0400
Free Swap Space: 10:00 AM -0400
Used Swap Space: 10:00 AM -0400

Resource: Package History

publican-2.99-0.el6.t8.x86_64.rpm: 11 days, 19 hours ago
publican-2.99-0.el6.t8.x86_64.rpm: 11 days, 19 hours ago
publican-2.99-0.el6.t8.x86_64.rpm: 11 days, 18 hours ago

Clicking any of the metrics will open the baseline chart for that metric. As is described in Section 19.1.4, "Baselines and Out-of-Bounds Metrics", baselines calculate an average reading for a given period of time, with the high and low measurements in that period creating upper and lower bounds. Baselines, by default, are calculated every three days using the data from the previous seven days for the calculation. Baseline measurements are essential for establishing operating norms so that administrators can effectively set alerts for resources.

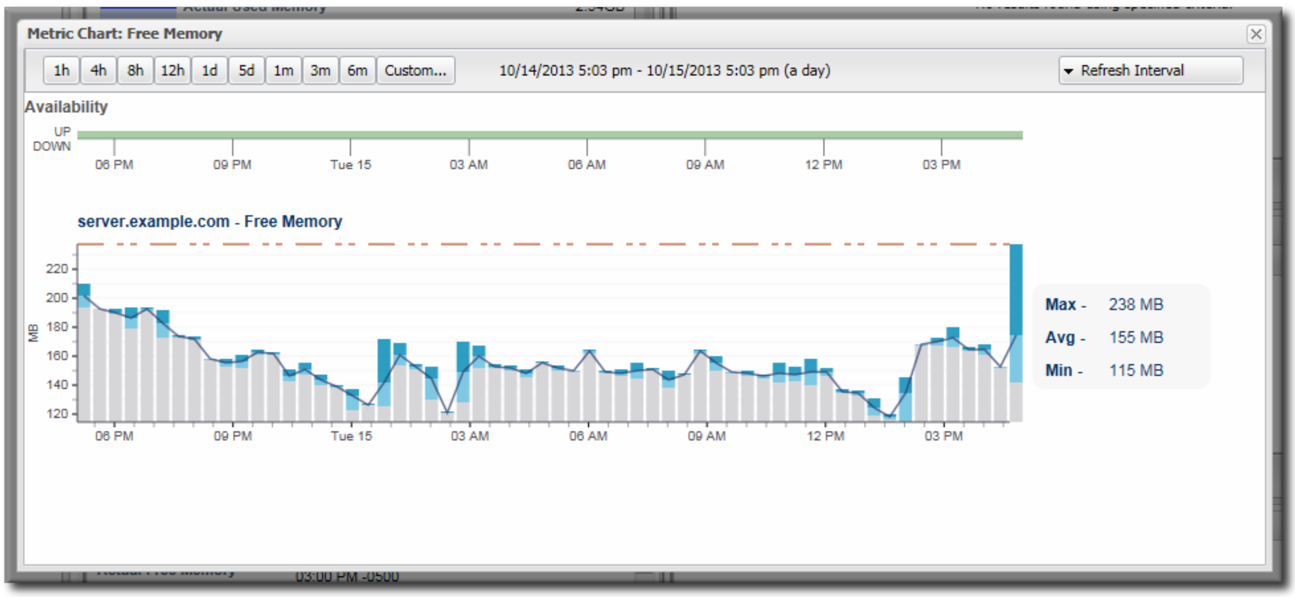


Figure 19.6. Individual Metric Graph

The `Metrics` area in the `Monitoring` tab shows all of the metrics in a table, with columns for the high, low, and current readings. There is also a column which shows the number of active alerts for each metric.

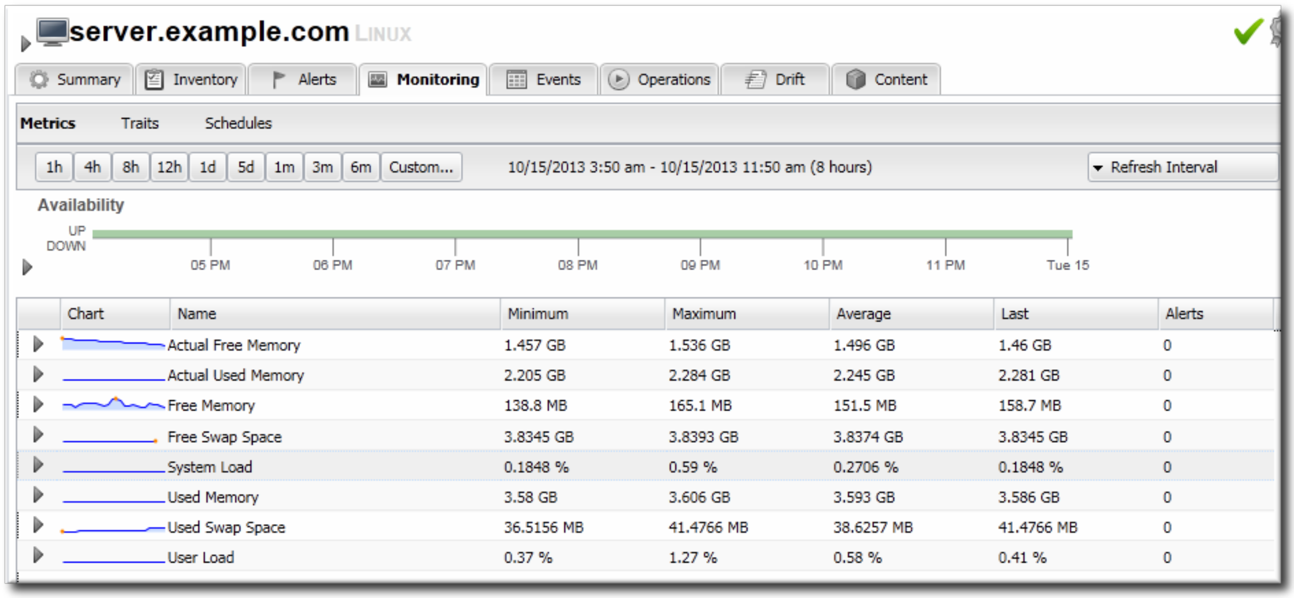
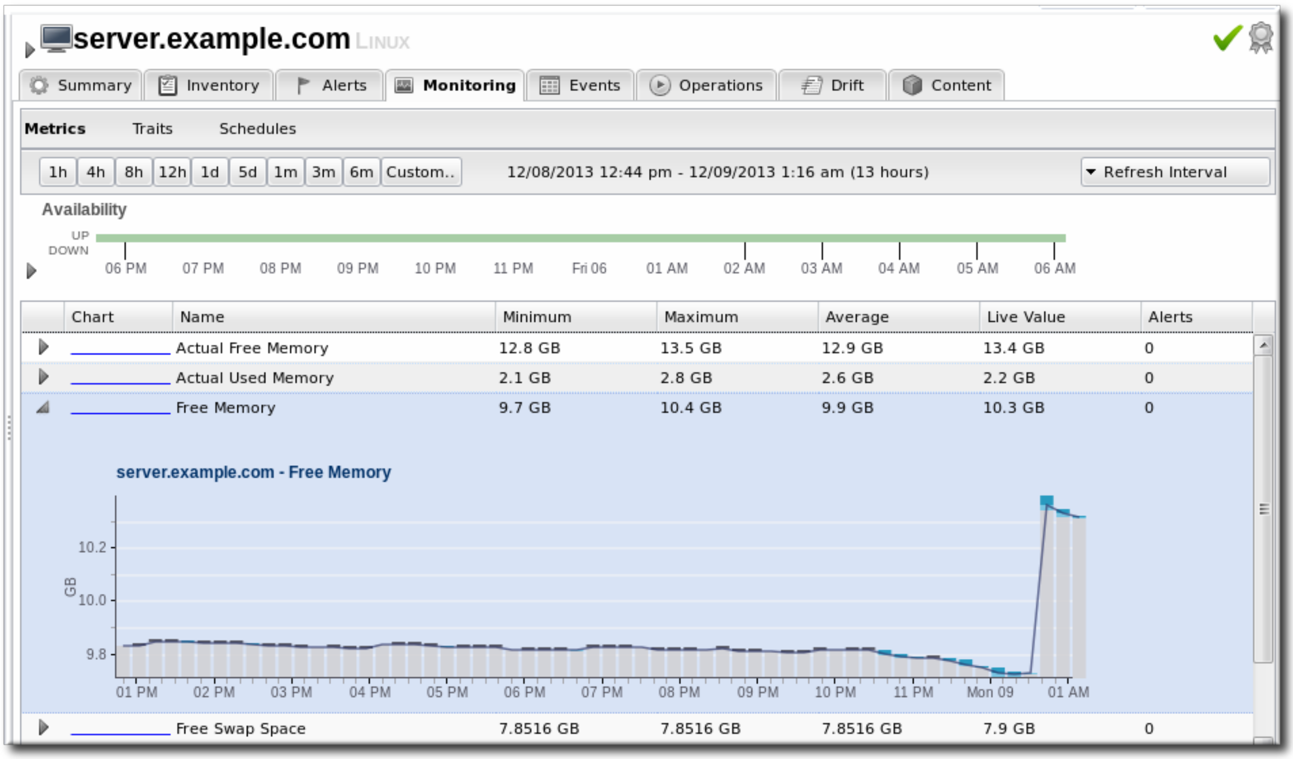


Chart	Name	Minimum	Maximum	Average	Last	Alerts
	Actual Free Memory	1.457 GB	1.536 GB	1.496 GB	1.46 GB	0
	Actual Used Memory	2.205 GB	2.284 GB	2.245 GB	2.281 GB	0
	Free Memory	138.8 MB	165.1 MB	151.5 MB	158.7 MB	0
	Free Swap Space	3.8345 GB	3.8393 GB	3.8374 GB	3.8345 GB	0
	System Load	0.1848 %	0.59 %	0.2706 %	0.1848 %	0
	Used Memory	3.58 GB	3.606 GB	3.593 GB	3.586 GB	0
	Used Swap Space	36.5156 MB	41.4766 MB	38.6257 MB	41.4766 MB	0
	User Load	0.37 %	1.27 %	0.58 %	0.41 %	0

Figure 19.7. Metrics Table

Expanding a metric opens its individual metric graph, giving the trend for the past eight hours. This provides more granular detail than the summary or baselines charts, showing the readings for each collection period and the precise readings.



12/08/2013 12:44 pm - 12/09/2013 1:16 am (13 hours)

Chart	Name	Minimum	Maximum	Average	Live Value	Alerts
	Actual Free Memory	12.8 GB	13.5 GB	12.9 GB	13.4 GB	0
	Actual Used Memory	2.1 GB	2.8 GB	2.6 GB	2.2 GB	0
	Free Memory	9.7 GB	10.4 GB	9.9 GB	10.3 GB	0
	Free Swap Space	7.8516 GB	7.8516 GB	7.8516 GB	7.9 GB	0

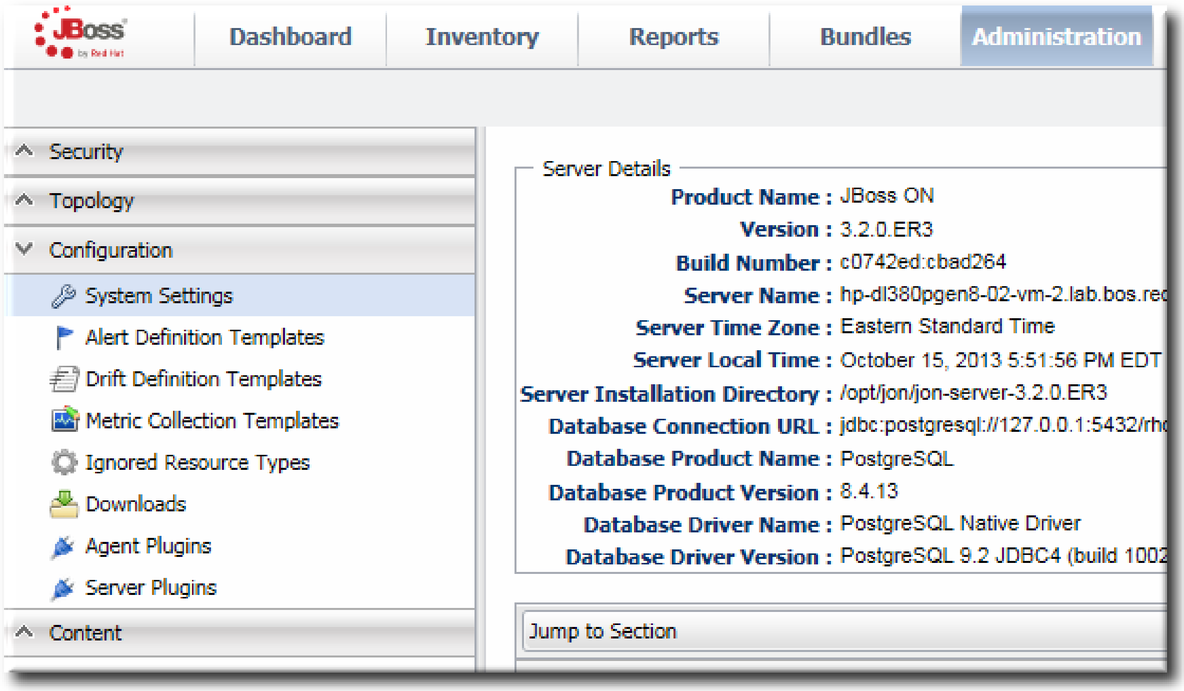
Figure 19.8. Metric Graph within the Table

19.3. DEFINING METRICS COLLECTION

19.3.1. Setting Baseline Calculation Properties

The monitoring baselines have two configuration properties that define *how* the automatic metric baselines are calculated. These properties don't set the value; they set the window of time used for the baseline averages.

1. In the `System Configuration` menu, select the **Settings** item.

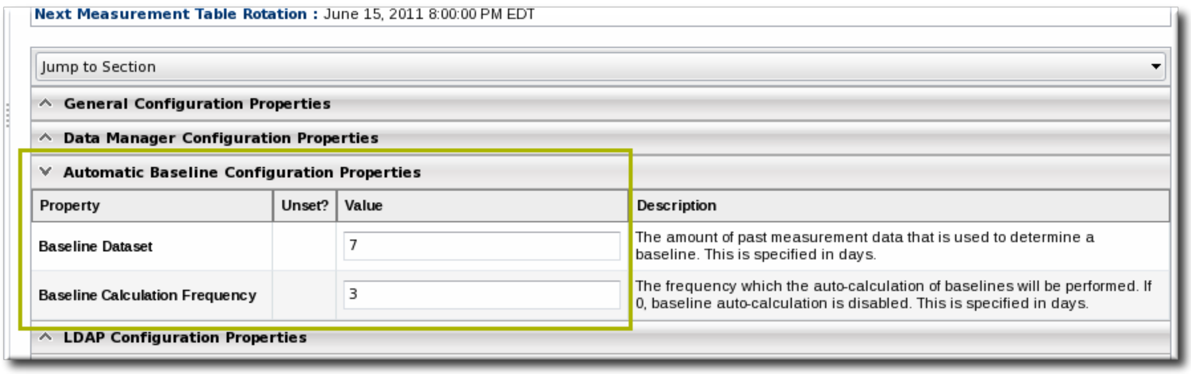


Server Details

Product Name :	JBoss ON
Version :	3.2.0.ER3
Build Number :	c0742ed:cbad264
Server Name :	hp-dl380pgen8-02-vm-2.lab.bos.red
Server Time Zone :	Eastern Standard Time
Server Local Time :	October 15, 2013 5:51:56 PM EDT
Server Installation Directory :	/opt/jon/jon-server-3.2.0.ER3
Database Connection URL :	jdbc:postgresql://127.0.0.1:5432/rhq
Database Product Name :	PostgreSQL
Database Product Version :	8.4.13
Database Driver Name :	PostgreSQL Native Driver
Database Driver Version :	PostgreSQL 9.2 JDBC4 (build 1002

2. Scroll to the **Automatic Baseline Configuration Properties** section.

3. Change the settings to define the window used for calculation.



Next Measurement Table Rotation : June 15, 2011 8:00:00 PM EDT

Jump to Section

General Configuration Properties

Data Manager Configuration Properties

Automatic Baseline Configuration Properties

Property	Unset?	Value	Description
Baseline Dataset		7	The amount of past measurement data that is used to determine a baseline. This is specified in days.
Baseline Calculation Frequency		3	The frequency which the auto-calculation of baselines will be performed. If 0, baseline auto-calculation is disabled. This is specified in days.

LDAP Configuration Properties

- *Baseline Frequency* sets the interval, in days, for how often baselines are recalculated. The default is three days.

- *Baseline Dataset* sets the time interval, in days, used to calculate the baseline. The default is seven days.

19.3.2. Setting Collection Intervals for a Specific Resource

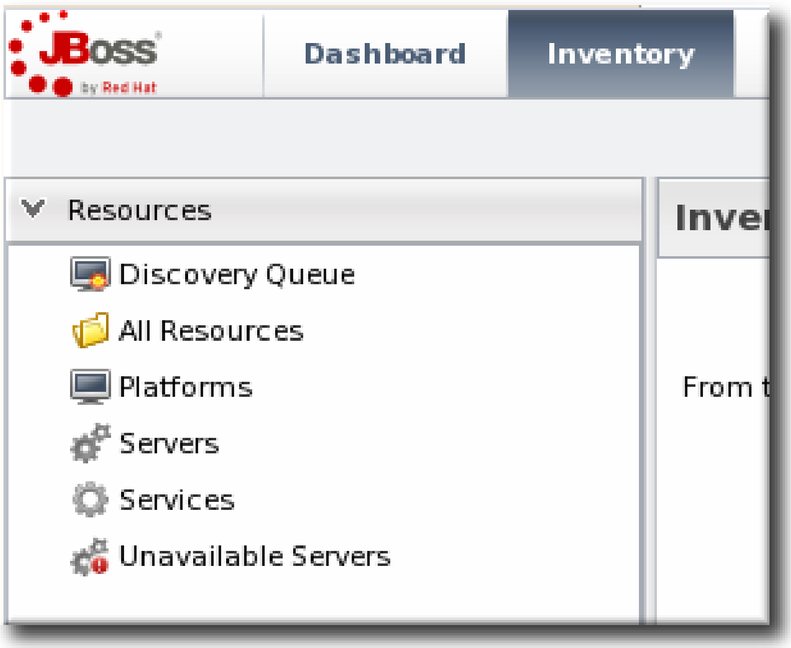
Metrics are collected at the intervals specified by the collection schedule. Because not all metrics are mission critical or even likely to change, JBoss ON has different collection schedules for different metrics, with critical metrics collected more frequently.

For most environments, setting a daily collection schedule (once every 24 hours) is sufficient.

To change the collection interval for a specific metric:

1. Click the **Inventory** tab in the top menu.

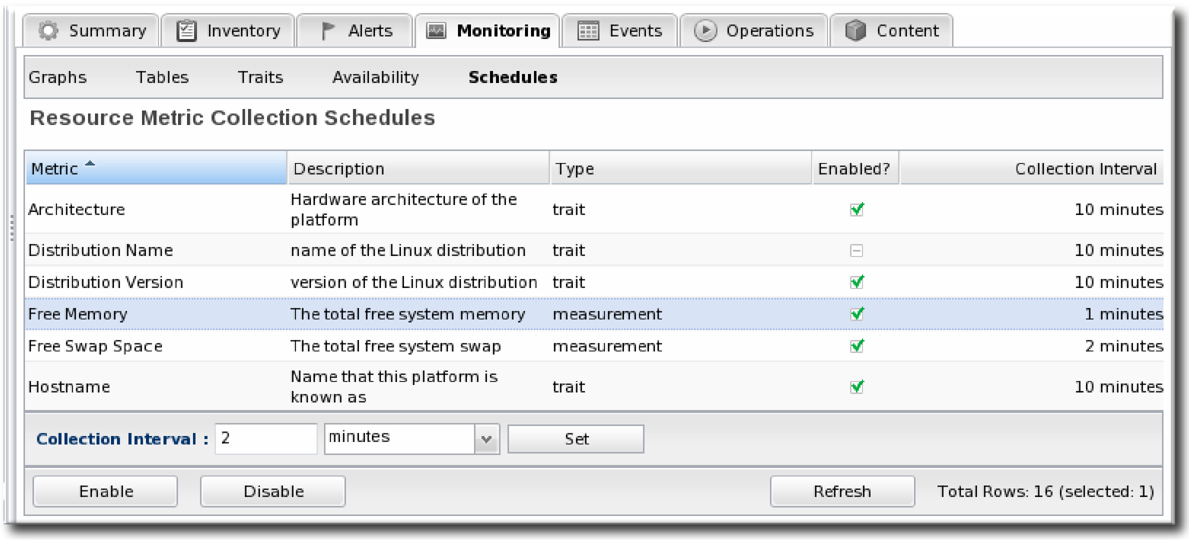
2. Select the resource category, such as servers or services, in the **Resources** menu table on the left. Then browse or search for the resource.



3. Click the **Monitoring** tab on the resource entry.

4. Click the **Schedules** subtab.

5. Select the metric for which to change the monitoring frequency. Multiple metrics can be selected, if they will all be changed to the same frequency.



Resource Metric Collection Schedules

Metric	Description	Type	Enabled?	Collection Interval
Architecture	Hardware architecture of the platform	trait	✓	10 minutes
Distribution Name	name of the Linux distribution	trait	☐	10 minutes
Distribution Version	version of the Linux distribution	trait	✓	10 minutes
Free Memory	The total free system memory	measurement	✓	1 minutes
Free Swap Space	The total free system swap	measurement	✓	2 minutes
Hostname	Name that this platform is known as	trait	✓	10 minutes

Collection Interval : 2 minutes Set

Enable Disable Refresh Total Rows: 16 (selected: 1)

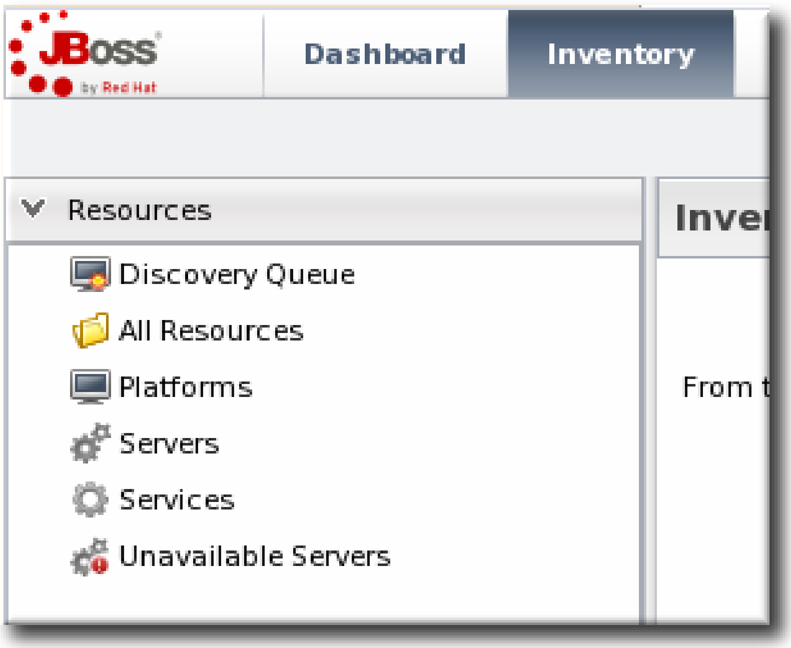
6. Enter the desired collection period in the **Collection Interval** field, with the appropriate time unit (seconds, minutes, or hours).

7. Click **Set**.

19.3.3. Enabling and Disabling Metrics for a Specific Resource

1. Click the **Inventory** tab in the top menu.

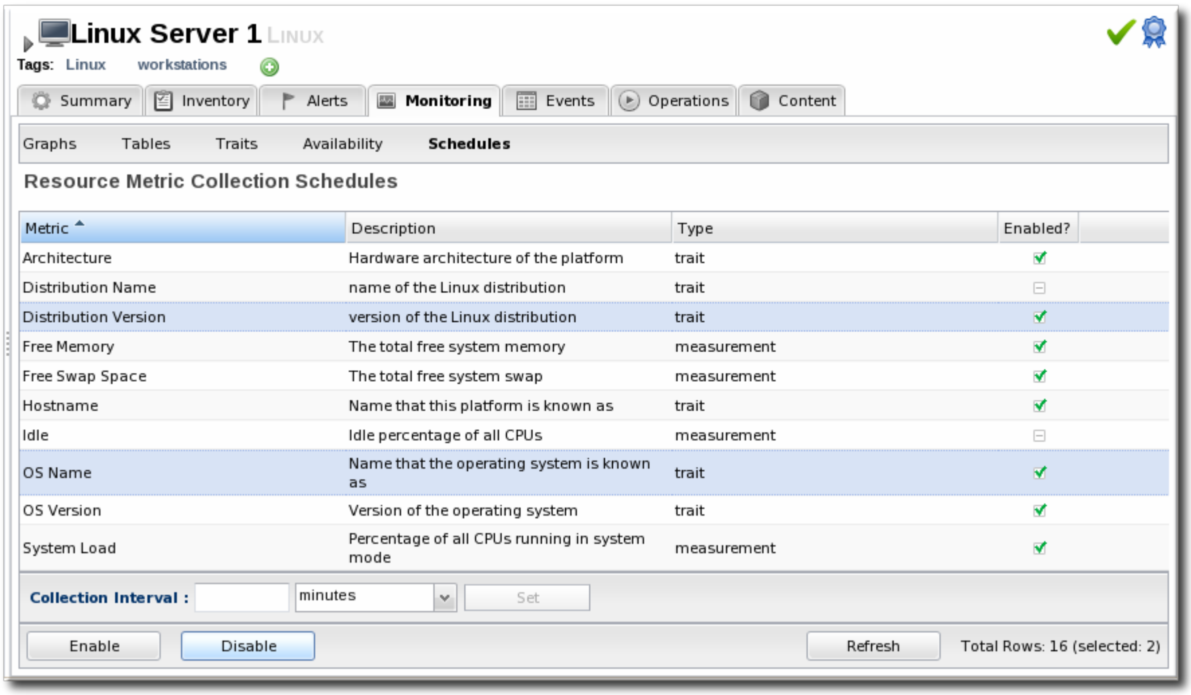
2. Select the resource type in the **Resources** menu table on the left, and then browse or search for the resource.



3. Click the **Monitoring** tab on the resource entry.

4. Click the **Schedules** sub tab.

5. Select the metrics to enable or disable.



Resource Metric Collection Schedules

Metric	Description	Type	Enabled?
Architecture	Hardware architecture of the platform	trait	✓
Distribution Name	name of the Linux distribution	trait	⊟
Distribution Version	version of the Linux distribution	trait	✓
Free Memory	The total free system memory	measurement	✓
Free Swap Space	The total free system swap	measurement	✓
Hostname	Name that this platform is known as	trait	✓
Idle	Idle percentage of all CPUs	measurement	⊟
OS Name	Name that the operating system is known as	trait	✓
OS Version	Version of the operating system	trait	✓
System Load	Percentage of all CPUs running in system mode	measurement	✓

Collection Interval : minutes Set

Enable Disable Refresh Total Rows: 16 (selected: 2)

6. Click the **Enable** or **Disable** button.

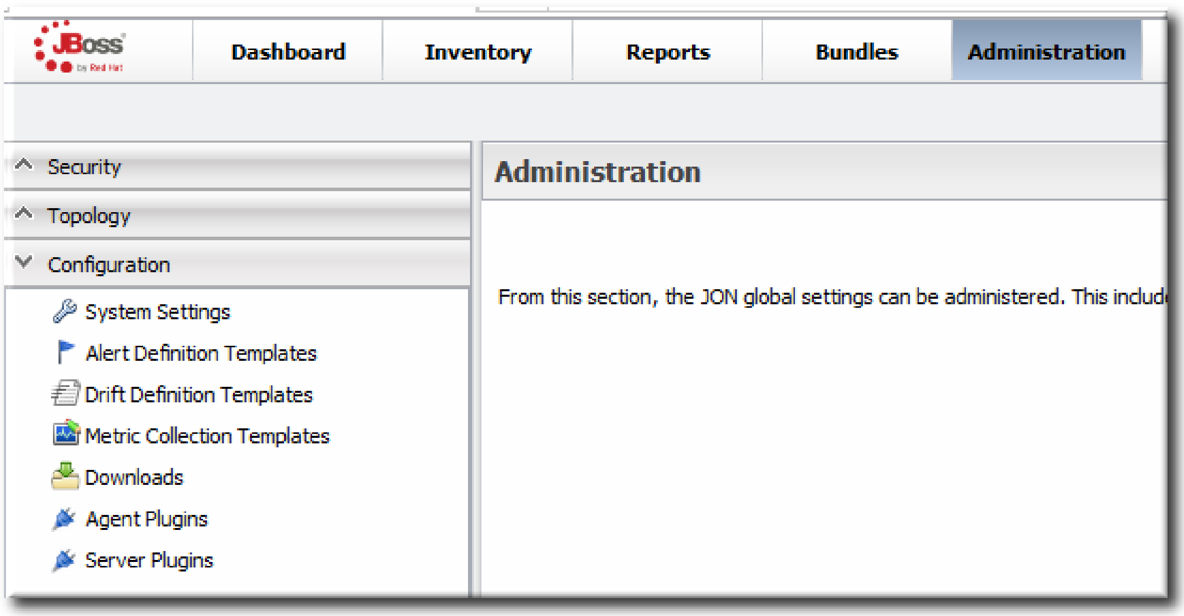
19.3.4. Changing Metrics Templates

The metrics which are collected for a resource type are defined in the *monitoring template* for the resource type. Each resource type has some metrics disabled by default, and these must be manually enabled. Likewise, metrics which are enabled by default can be disabled.

NOTE

Metric templates only apply to new resources of that resource type unless the checkbox is selected to apply them to existing resources as well as new resources.

1. In the top navigation, open the **Administration** menu, and then the **System Configuration** menu.

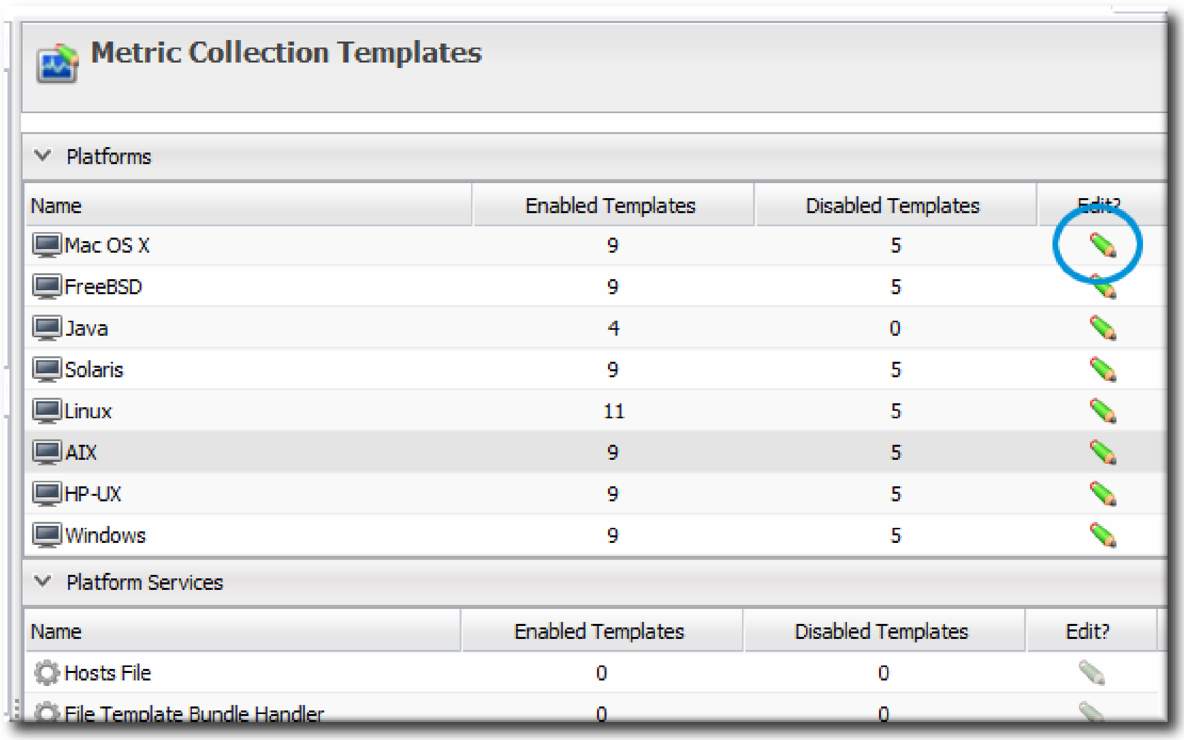


Administration

From this section, the JON global settings can be administered. This includ

2. Select the `Metric Collection Templates` menu item. This opens a long list of resource types, both for platforms and server types.

3. Locate the type of resource for which to create the template definition.



Metric Collection Templates

Platforms

Name	Enabled Templates	Disabled Templates	Edit?
Mac OS X	9	5	✎
FreeBSD	9	5	✎
Java	4	0	✎
Solaris	9	5	✎
Linux	11	5	✎
AIX	9	5	✎
HP-UX	9	5	✎
Windows	9	5	✎

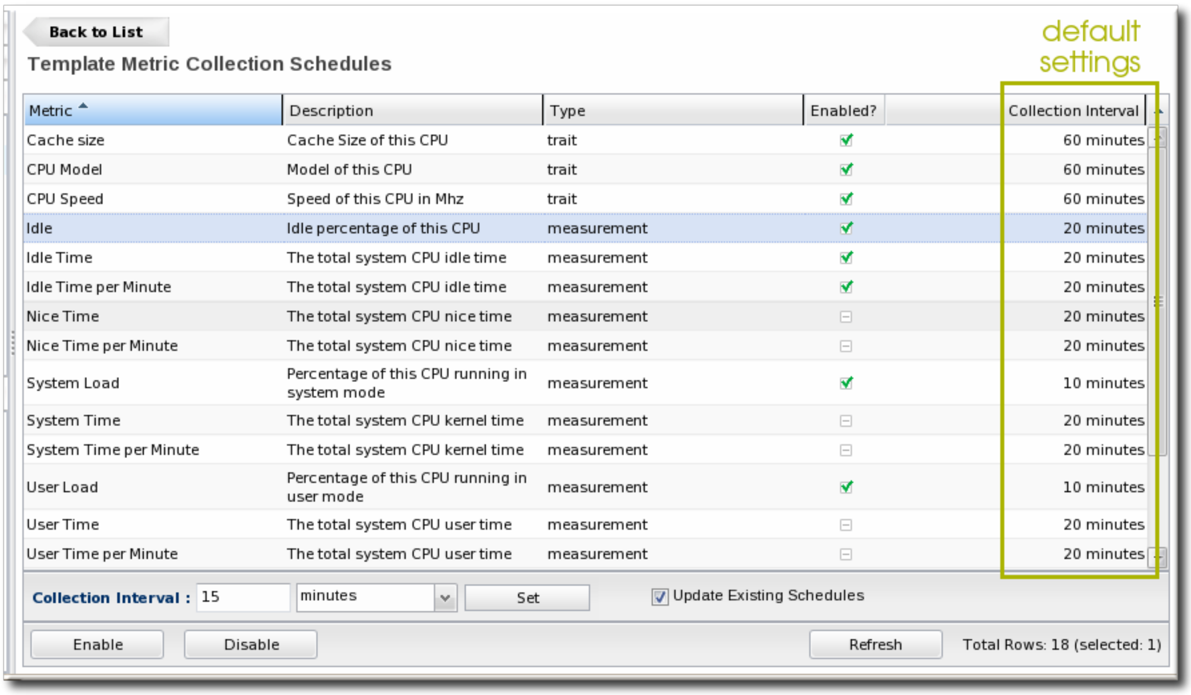
Platform Services

Name	Enabled Templates	Disabled Templates	Edit?
Hosts File	0	0	✎
File Template Bundle Handler	0	0	✎

4. Click the pencil icon to edit the metric collection schedule templates.

5. Select the required metrics to enable or disable, and click the **`Enable`** or **`Disable`** button.

6. To edit the frequency that a metric is collected, select the **`Update schedules for existing resources of marked type`** checkbox, and then enter the desired time frame into the **`Collection Interval for Selected:`** field.



Template Metric Collection Schedules

Metric	Description	Type	Enabled?	Collection Interval
Cache size	Cache Size of this CPU	trait	✓	60 minutes
CPU Model	Model of this CPU	trait	✓	60 minutes
CPU Speed	Speed of this CPU in Mhz	trait	✓	60 minutes
Idle	Idle percentage of this CPU	measurement	✓	20 minutes
Idle Time	The total system CPU idle time	measurement	✓	20 minutes
Idle Time per Minute	The total system CPU idle time	measurement	✓	20 minutes
Nice Time	The total system CPU nice time	measurement	☐	20 minutes
Nice Time per Minute	The total system CPU nice time	measurement	☐	20 minutes
System Load	Percentage of this CPU running in system mode	measurement	✓	10 minutes
System Time	The total system CPU kernel time	measurement	☐	20 minutes
System Time per Minute	The total system CPU kernel time	measurement	☐	20 minutes
User Load	Percentage of this CPU running in user mode	measurement	✓	10 minutes
User Time	The total system CPU user time	measurement	☐	20 minutes
User Time per Minute	The total system CPU user time	measurement	☐	20 minutes

Collection Interval : 15 minutes Set ✓ Update Existing Schedules

Enable Disable Refresh Total Rows: 18 (selected: 1)

7. Click the **Set** button.

19.3.5. Adding a PostgreSQL Query as a Metric

A SQL query can be added to a PostgreSQL database as a child resource. That entry becomes a custom metric for that PostgreSQL database.

A query metric **must** have two columns that allow the JBoss ON agent to collect data for the query:

- metricColumn
- count(id)

The query has to return a single row with those two columns. The first column signals that it is a collected metric, and the second gives the count for the metric.

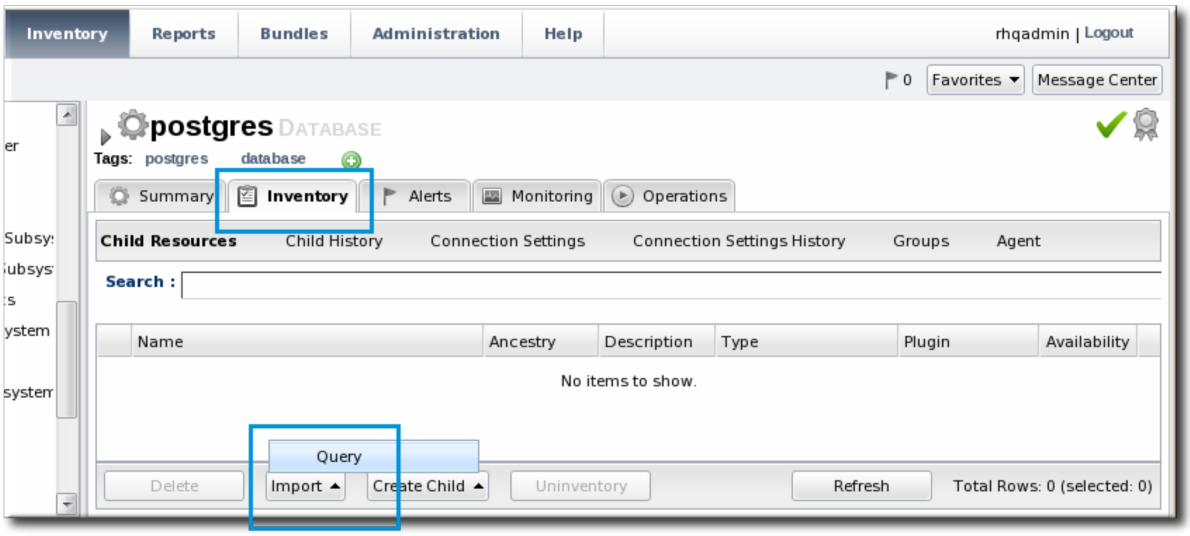
For example, to track logged-in users:

```
SELECT 'metricColumn', count(id) FROM my_application_user WHERE
is_logged_in = true
```

The **SELECT** statement defines the metric for the JBoss ON agent. The rest of the query collects the data from the database. Simple as that.

To add a metric based on a query:

1. Click the **Inventory** tab in the top menu.
2. Search for the PostgreSQL resource.
3. Click the **Inventory** tab for the PostgreSQL database.
4. Click the **Import** button in the bottom of the **Inventory** tab, and select **Query**.

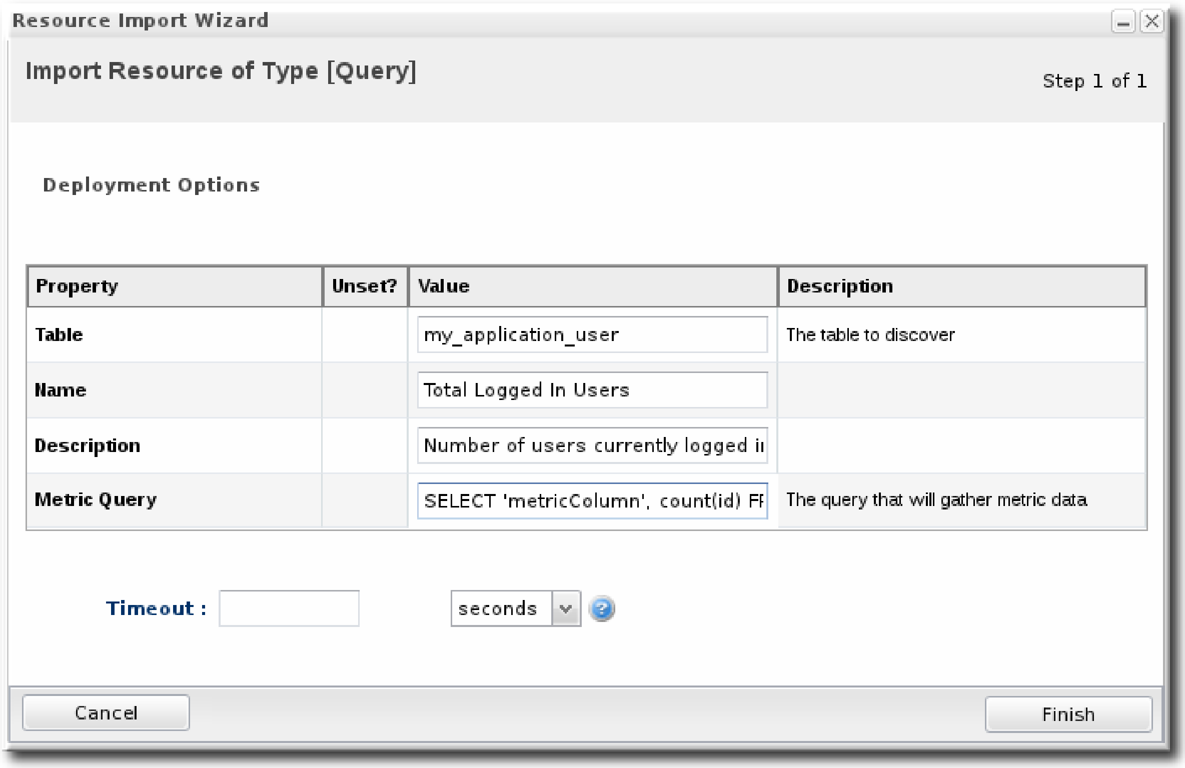


5. Fill in the properties for the query metric. Three fields are particularly important:

   - The **`Table`** gives which table within the database contains the data; this is whatever is in the **`FROM`** statement in the query.
   - The **`Metric Query`** contains the full query to run. The **`SELECT`** statement must be **`'metricColumn',count(id)`** to format the query properly for the JBoss ON agent to interpret it as a metric.

     ```
     SELECT 'metricColumn', count(id) FROM my_application_user WHERE
     is_logged_in = true
     ```

   - The **Name** field is not important in configuring the metric, but it is important identifying the metric later.


Resource Import Wizard

Import Resource of Type [Query]

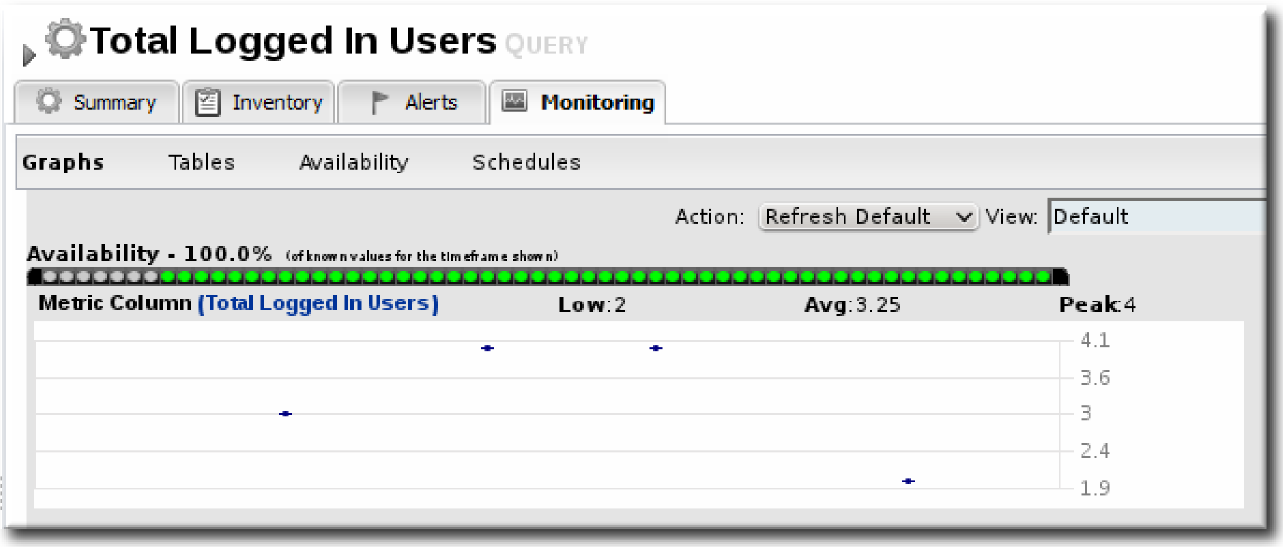
Step 1 of 1

Deployment Options

Property	Unset?	Value	Description
Table		my_application_user	The table to discover
Name		Total Logged In Users	
Description		Number of users currently logged i	
Metric Query		SELECT 'metricColumn', count(id) FF	The query that will gather metric data

Timeout : seconds

Once the query is created, then the agent begins collecting the counts for the data.



Availability - 100.0% (of known values for the timeframe shown)

Metric Column (Total Logged In Users) Low:2 Avg:3.25 Peak:4

Figure 19.9. Query: Total Logged-in User Count

CHAPTER 20. EVENTS

Metric data are collected according to a schedule. However, some actions occur on a resource sporadically, such as sudden system shutdowns. These are *events*. Since event data can be generated randomly, events are sent to agents immediately when they are detected.

20.1. EVENTS, LOGS, AND RESOURCES

Operating and some server types keep their own log files and register a steady stream of information about incidents for that resource, from simple debug information to critical errors. Metrics are collected on a (more or less) set schedule. Events are entirely random, because they are based on real actions as they occur. Events, then, give a different perspective on resource performance.

Events monitoring tracks those streaming log file messages. In a sense, JBoss ON event monitoring is a filtered log viewer. When events are enabled, all of the log messages flow through JBoss ON's event viewer. However, the types of messages that JBoss ON records and displays can be limited in the event configuration so that only certain types of messages or messages matching certain strings are included in the events view.

A handful of resource types record and display events:

- Linux (syslog)
- Windows (Windows event logs)
- Apache server (log files)
- JBoss EAP server (log files)

NOTE

Events must be enabled and configured for a resource before they become active.

Default events are taken from standard log files. Custom resource types can identify event sources in logs or in asynchronous messaging systems such as a JMX notification or a JMS messaging system.

20.2. EVENT DATE FORMATTING

JBoss ON expects log messages to follow the log4j log pattern, over all.

```
date severity [class] message
```

The *date* format is configurable when event monitoring is enabled. For custom logs, a custom pattern can be added. If no date format is given, then the three standard formats used by log4j are tried.

```
YYYY-mm-dd HH:mm:ss,SSS
HH:mm:ss,SSS
dd MM yyyy HH:mm:ss,SSS
```

The severity must be next. It can be plain, surrounded by brackets, or surrounded by parentheses.

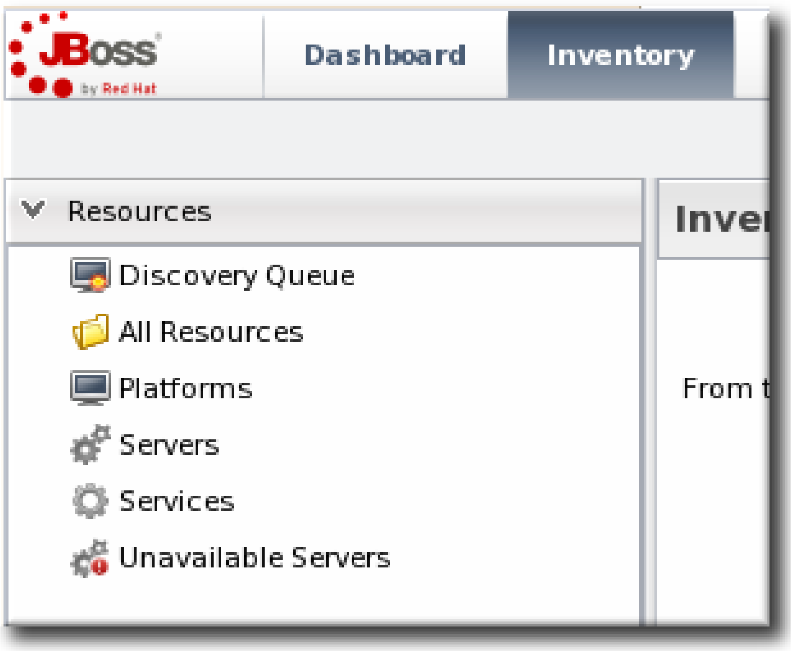
```
date SEVERITY [org.foo.bar] my message
date [SEVERITY] [org.foo.bar] my message
date ( SEVERITY ) [org.foo.bar] my message
```

After the severity, there are no real constraints on the format of the log entry. If classes or other identifiers are passed, they are properly displayed.

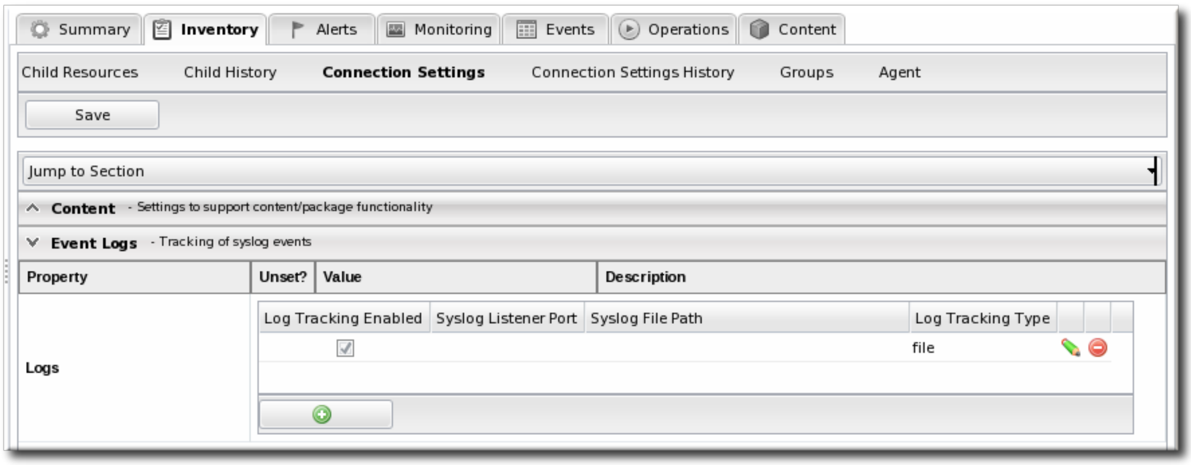
20.3. DEFINING A NEW EVENT

Events are only recognized by the monitoring service if events logging is properly enabled for the specific service being logged. This requires creating a log event for the log or system service, specifying a log path on the resource, and setting a date format which matches the format for the log.

1. Click the **`Inventory`** tab in the top menu.
2. Select the resource type in the **`Resources`** menu table on the left, and then browse or search for the resource.



3. Click the **`Inventory`** tab on the resource entry.
4. Select the **`Connection Settings`** subtab.
5. Click green plus icon under the **`Events Log`** section to add a log instance to monitor.



Property	Unset?	Value	Description
Logs			

Log Tracking Enabled	Syslog Listener Port	Syslog File Path	Log Tracking Type
✓			file

6. Enable event log collection.

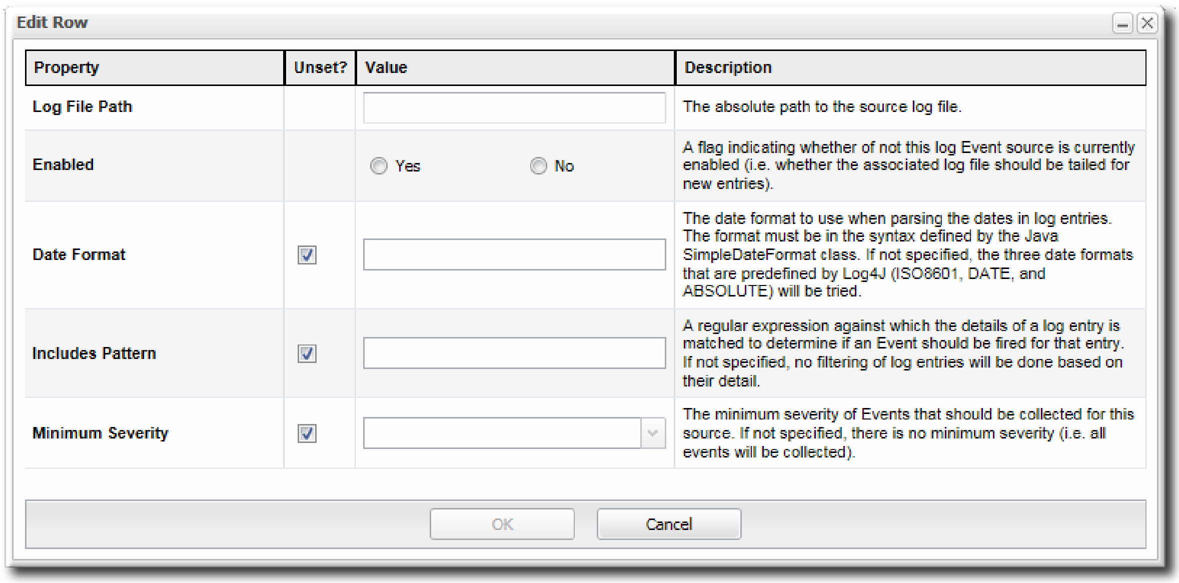
7. Set the parameters for the event log collection.

   Depending on the resource being configured, there are slightly different options for the event log configuration. All of the resources all allow different filters to identify which log messages to include:

   - A minimum severity of any error message.
   - A regular expression or pattern to use on log message strings.

   Additionally, the application servers and Linux allow different log locations to be specified. (The Windows resource uses its System Event Log.) Along with accommodating custom log locations, it also allows for other logging services to be used. For Linux, this allows both platform and program logs to be monitored; for application services, this allows logs within a messaging service to be checked.

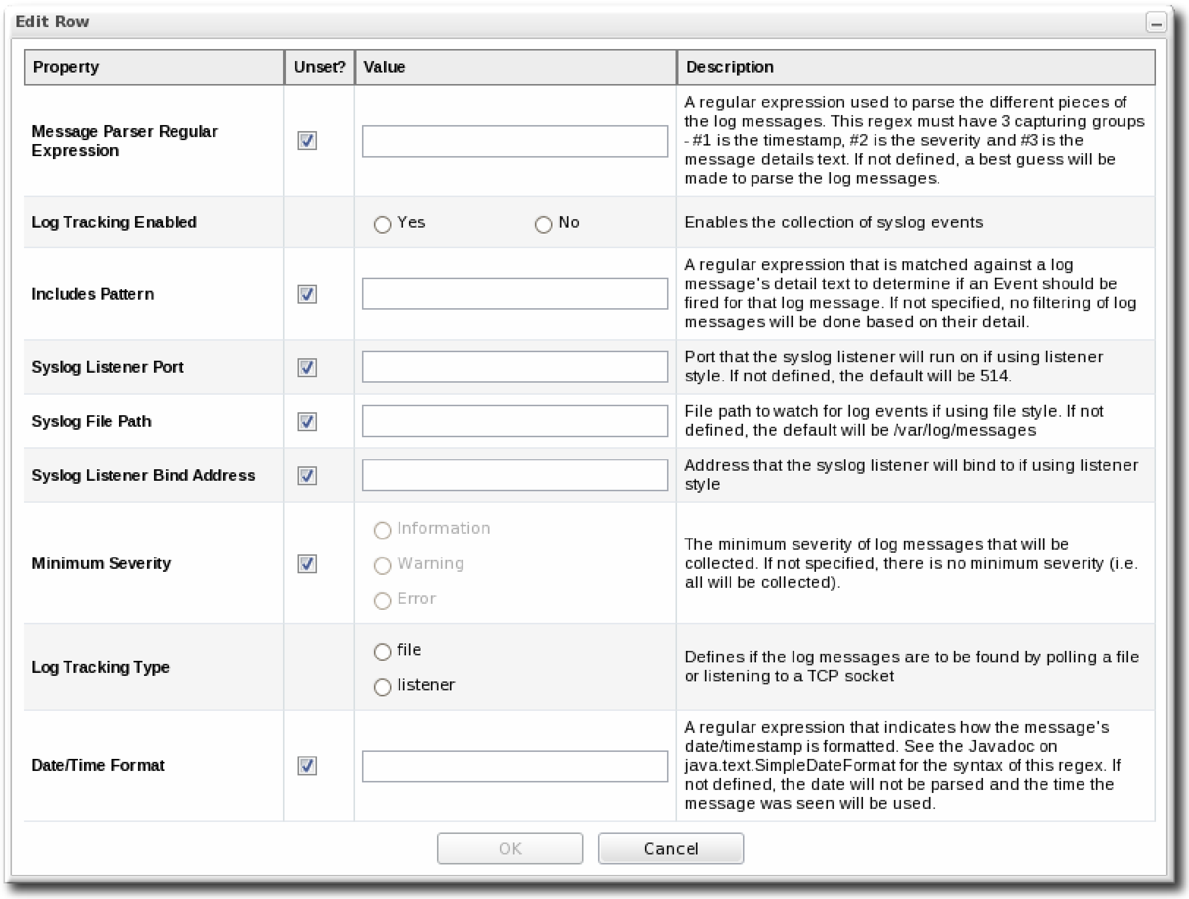
   As discussed in Section 20.2, "Event Date Formatting" , there are different potential date formats that can be used in logging; if anything other than log4j is used, the pattern can be specified so that the agent can read the log entries.



Property	Unset?	Value	Description
Log File Path			The absolute path to the source log file.
Enabled		Yes No	A flag indicating whether of not this log Event source is currently enabled (i.e. whether the associated log file should be tailed for new entries).
Date Format	✓		The date format to use when parsing the dates in log entries. The format must be in the syntax defined by the Java SimpleDateFormat class. If not specified, the three date formats that are predefined by Log4J (ISO8601, DATE, and ABSOLUTE) will be tried.
Includes Pattern	✓		A regular expression against which the details of a log entry is matched to determine if an Event should be fired for that entry. If not specified, no filtering of log entries will be done based on their detail.
Minimum Severity	✓		The minimum severity of Events that should be collected for this source. If not specified, there is no minimum severity (i.e. all events will be collected).

Figure 20.1. EAP Event Log Configuration

Unlike either the application servers or Windows, Linux systems can log events in a system file *or* in a listener. If an rsyslog server or local syslog listener is configured, then it is possible to select a listener (rather than a local file) and to add the listener port and bind address for a remote server.

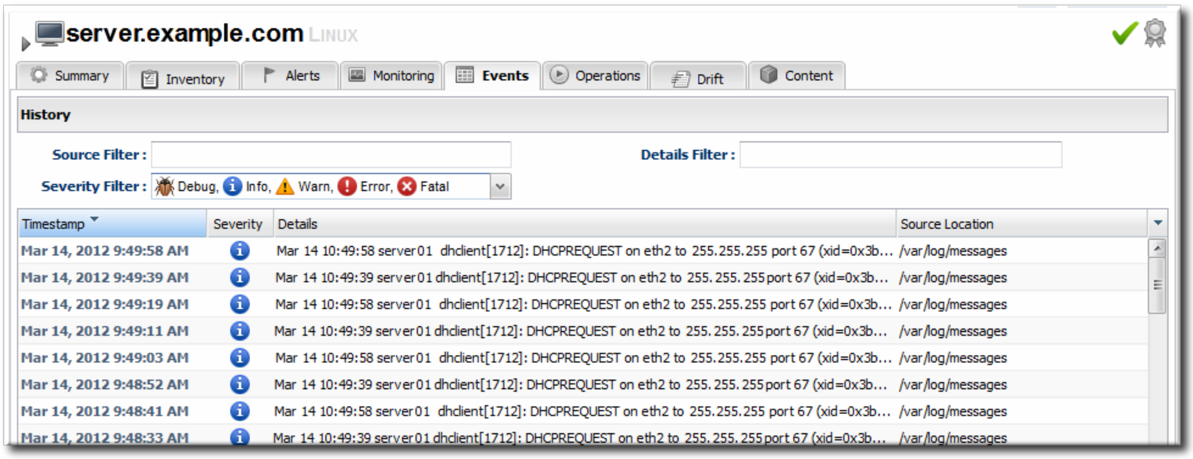


Property	Unset?	Value	Description
Message Parser Regular Expression	✓		A regular expression used to parse the different pieces of the log messages. This regex must have 3 capturing groups - #1 is the timestamp, #2 is the severity and #3 is the message details text. If not defined, a best guess will be made to parse the log messages.
Log Tracking Enabled		○ Yes ○ No	Enables the collection of syslog events
Includes Pattern	✓		A regular expression that is matched against a log message's detail text to determine if an Event should be fired for that log message. If not specified, no filtering of log messages will be done based on their detail.
Syslog Listener Port	✓		Port that the syslog listener will run on if using listener style. If not defined, the default will be 514.
Syslog File Path	✓		File path to watch for log events if using file style. If not defined, the default will be /var/log/messages
Syslog Listener Bind Address	✓		Address that the syslog listener will bind to if using listener style
Minimum Severity	✓	○ Information ○ Warning ○ Error	The minimum severity of log messages that will be collected. If not specified, there is no minimum severity (i.e. all will be collected).
Log Tracking Type		○ file ○ listener	Defines if the log messages are to be found by polling a file or listening to a TCP socket
Date/Time Format	✓		A regular expression that indicates how the message's date/timestamp is formatted. See the Javadoc on java.text.SimpleDateFormat for the syntax of this regex. If not defined, the date will not be parsed and the time the message was seen will be used.

Figure 20.2. Linux Event Log Configuration

20.4. VIEWING EVENTS

1. Click the **`Inventory`** tab in the top menu.
2. Select the resource type in the **`Resources`** menu table on the left, and then browse or search for the resource.
3. Click the **`Events`** tab on the resource entry. Events can be filtered by severity (debug, info, warn, error, and fatal).
4. Click the specific event for further details.



History

Source Filter :

Details Filter :

Severity Filter : Debug, Info, Warn, Error, Fatal

Timestamp	Severity	Details	Source Location
Mar 14, 2012 9:49:58 AM	ℹ	Mar 14 10:49:58 server01 dhclient[1712]: DHCPREQUEST on eth2 to 255.255.255 port 67 (xid=0x3b...	/var/log/messages
Mar 14, 2012 9:49:39 AM	ℹ	Mar 14 10:49:39 server01 dhclient[1712]: DHCPREQUEST on eth2 to 255. 255. 255 port 67 (xid=0x3b...	/var/log/messages
Mar 14, 2012 9:49:19 AM	ℹ	Mar 14 10:49:58 server01 dhclient[1712]: DHCPREQUEST on eth2 to 255.255.255 port 67 (xid=0x3b...	/var/log/messages
Mar 14, 2012 9:49:11 AM	ℹ	Mar 14 10:49:39 server01 dhclient[1712]: DHCPREQUEST on eth2 to 255. 255. 255 port 67 (xid=0x3b...	/var/log/messages
Mar 14, 2012 9:49:03 AM	ℹ	Mar 14 10:49:58 server01 dhclient[1712]: DHCPREQUEST on eth2 to 255.255.255 port 67 (xid=0x3b...	/var/log/messages
Mar 14, 2012 9:48:52 AM	ℹ	Mar 14 10:49:39 server01 dhclient[1712]: DHCPREQUEST on eth2 to 255. 255. 255 port 67 (xid=0x3b...	/var/log/messages
Mar 14, 2012 9:48:41 AM	ℹ	Mar 14 10:49:58 server01 dhclient[1712]: DHCPREQUEST on eth2 to 255.255.255 port 67 (xid=0x3b...	/var/log/messages
Mar 14, 2012 9:48:33 AM	ℹ	Mar 14 10:49:39 server01 dhclient[1712]: DHCPREQUEST on eth2 to 255. 255. 255 port 67 (xid=0x3b...	/var/log/messages

20.5. *DETAILED DISCUSSION:* EVENT CORRELATION

There are a lot of moving pieces in IT infrastructure, and one change can cause a cascade of results. One factor in being able to respond to events and manage infrastructure is the ability to correlate an event with its cause.

While there is not an exact way to pinpoint what change caused what event or alert, there is a way to help visualize how unrelated JBoss ON elements — alerts, configuration changes, event logs, drift detection, or inventory changes — might be related. Each resource has a **`Timeline`** area on its **`Summary`** tab. This collects all of the major operations that have occurred on that resource, either through JBoss ON or as detected by JBoss ON.

The thing to look for is clusters. For example, in this case, there was a configuration change on the JBoss server, and shortly after there was a critical alert and a simultaneous error in the JBoss server's logs. That is a reasonable indication that the configuration change caused a performance problem.

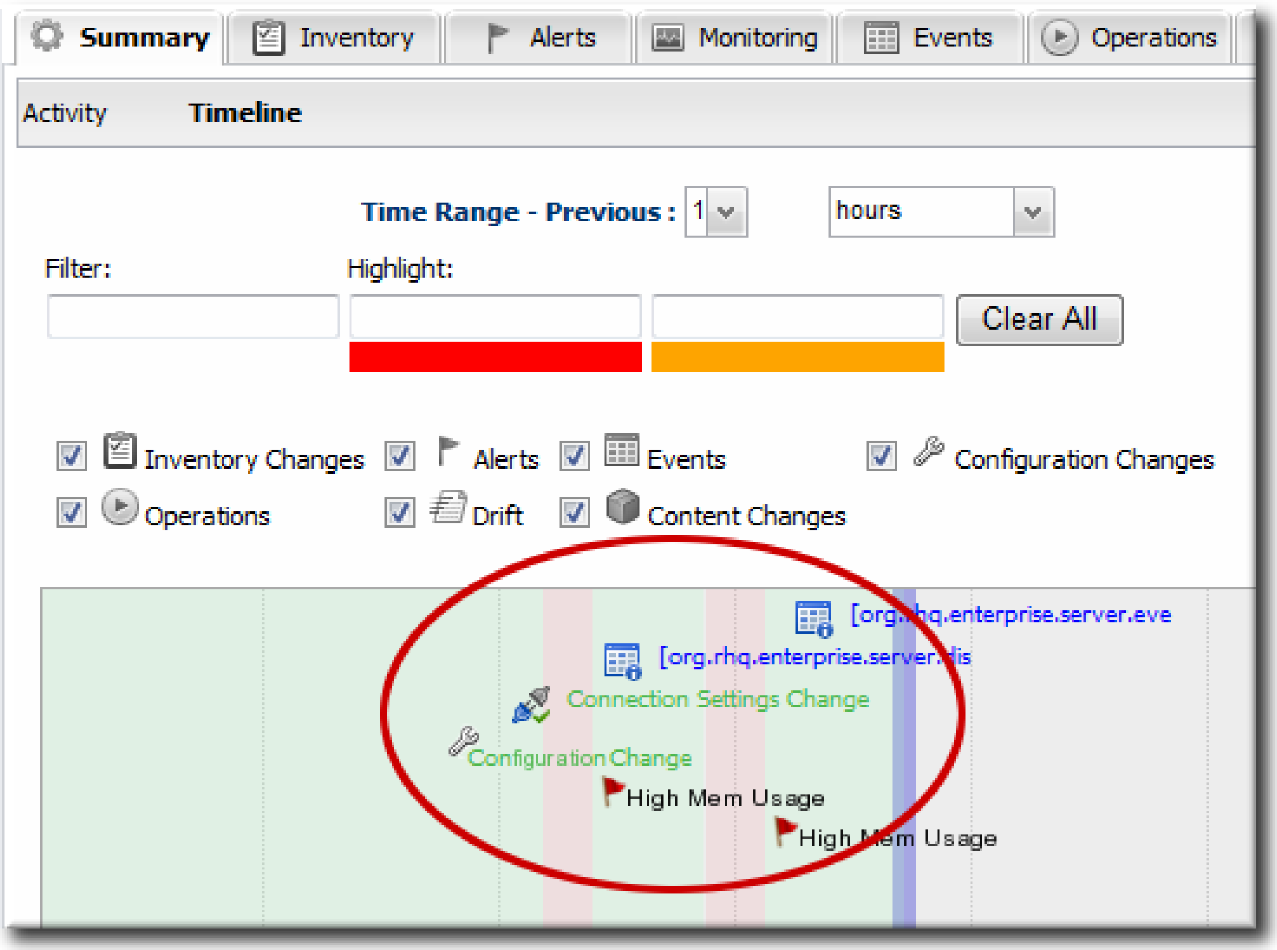


Figure 20.3. Resource Timeline Cluster

CHAPTER 21. URL RESPONSE TIME MONITORING

Part of web application performance is determined by how responsive it is to requests. JBoss Operations Network supplies an extra monitoring setting called *response time filters* which measures the amount of time it takes for a URL to respond to a request.

21.1. CALL-TIME (OR RESPONSE TIME) MONITORING FOR URLS

Performance for web applications is more complex than simple availability. How quickly an application can respond to requests is as important as whether the application is running.

The time that it takes an application to respond to some kind of request is *call-time* or *response time* data.

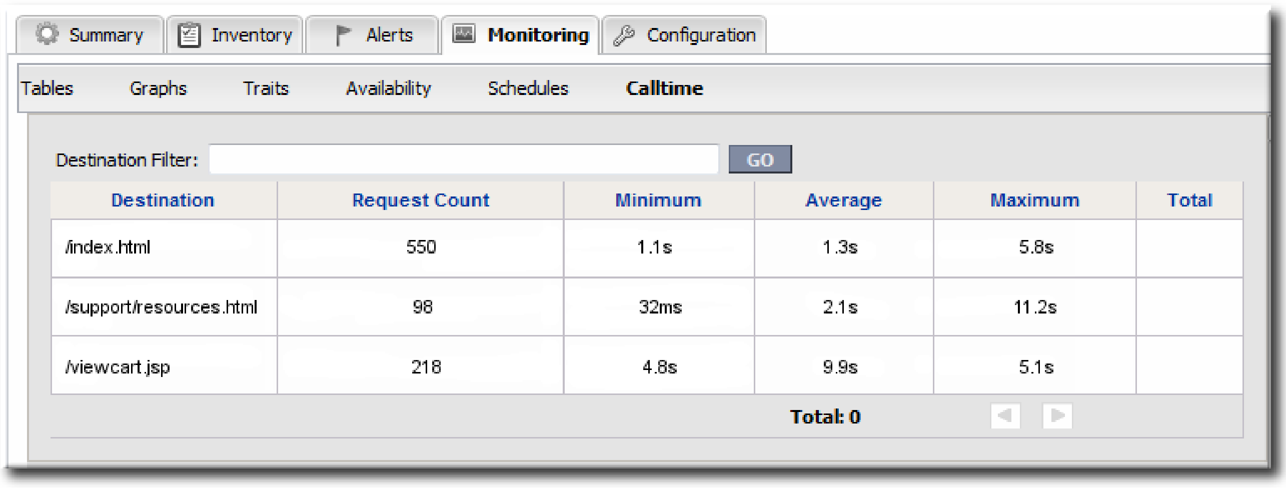
Two types of resources support call-time data by default:

- Session beans, for EJB method calls.
- Web servers (standalone or embedded in an application server), for URL responses. Web servers require an additional response time filter with configuration on what URL resources to measure for response times.

Response time monitoring is more based upon performance than it is on capturing a single data point. Response or call-time data are displayed as aggregates, showing maximum, minimum, and average response times per URL or per method.

21.2. VIEWING CALL TIME METRICS

Both session bean resources and web server resources have an additional `Monitoring` subtab called `Calltime`. All of the call-time data or response time data ranges (minimum, maximum, and averages) are displayed for each URL resource or method. As new URLs or methods are accessed, they are dynamically added to the results table.



Destination	Request Count	Minimum	Average	Maximum	Total
/index.html	550	1.1s	1.3s	5.8s	
/support/resources.html	98	32ms	2.1s	11.2s	
/viewcart.jsp	218	4.8s	9.9s	5.1s	

Total: 0

Figure 21.1. URL Metrics for a Web Server

21.3. *EXTENDED EXAMPLE:* WEBSITE PERFORMANCE

The Setup

A significant amount of Example Co.'s business, services, and support is tied to its website. Customers have to be able to access the site to purchase products, schedule training or consulting, and to receive most support and help. If the site is slow or if some resources are inaccessible, customers immediately have a negative experience.

The goal is not to monitor whether the web server is running, but whether the web application are responsive and performing as Example Co.'s customers expect.

What to Do

Tim the IT Guy identifies three different ways that he can capture web application performance information:

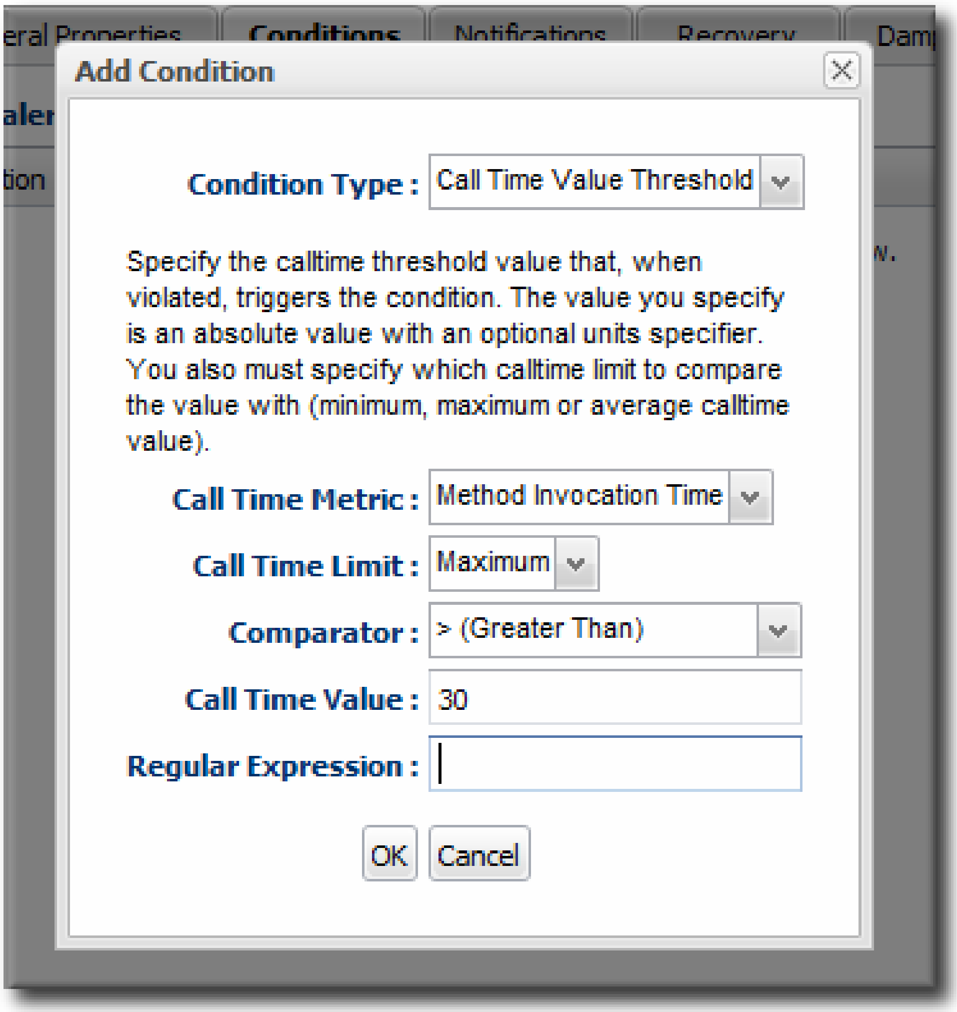
- Response times for individual URLs
- Throughput information like total number of requests and responses
- Counts for critical HTTP response codes

Both monitoring and alerting can be configured based solely off response time and throughput metrics. However, bad website performance is indicative of an underlying problem with the web server or its associated database. Therefore, Tim not only wants to be informed when website performance it poor; he wants to correlate some performance metrics with underlying server and database performance and launch operations that can mitigate poor responsiveness.

Tim maps a few common scenarios which cause poor website or web server performance and plans simple, immediate operations that JBoss ON can perform until an IT staffer can analyze the problem. Tim attempts to narrow down potential causes to a performance. Alerts can be issued for a single condition or for a combination of conditions. In Tim's case, he creates a three different alerts based on different combinations of underlying causes for performance problems (Section 25.1.2, “Basic Procedure for Setting Alerts for a Resource”).

- If there are poor response times and a high number of HTTP error 500 responses, then the alert can be configured with an operation to restart the web server (Section 25.3.2, “*Detailed Discussion:* Initiating an Operation”).
- If there are poor response times and a high number of HTTP error 404 response (meaning that resources may not be delivered properly), then the alert is configured to restart the database.
- If there are poor response times and a high number of total requests per minute, then it may mean that there is simply too much load on the server. The alert can be configured to create another web server instance to help with load balancing; using a JBoss ON CLI script allows the JBoss ON server to create new resources as necessary and deploy bundles of the appropriate web apps (Section 25.3.3, “*Detailed Discussion:* Initiating Resource Scripts”).

The most critical factor is the response time, which is a factor in every alert. Each alert has one condition based on the call time data, specifically of the call-time data moves past a certain threshold.



Condition Type : Call Time Value Threshold

Specify the calltime threshold value that, when violated, triggers the condition. The value you specify is an absolute value with an optional units specifier. You also must specify which calltime limit to compare the value with (minimum, maximum or average calltime value).

Call Time Metric : Method Invocation Time

Call Time Limit : Maximum

Comparator : > (Greater Than)

Call Time Value : 30

Regular Expression :

Tim picks a reasonable threshold, about 15 seconds, for performance. If performance degrades so that the HTTP Response Time metric returns a value higher than 20 seconds to load pages, JBoss ON issues an alert.

Alternatively, he could alert on simple call-time changes. Call-time changes will trigger an alert for any change from the established baseline, meaning a new minimum, maximum, or average value. A change of any kind can alert in either a decrease in performance *or* an increase in performance. A threshold alert only alerts on a specific change.

Tim then adds the other condition, with an AND operator, to each alert he configures.

Also, most web app-related metrics are not enabled by default. Tim enables the Total Number of Requests per Minute, Total Number of Responses per Minute, Number of 404 Responses per Minute, and Number of 500 Responses per Minute metrics for each web server (Section 19.3.4, “Changing Metrics Templates”).

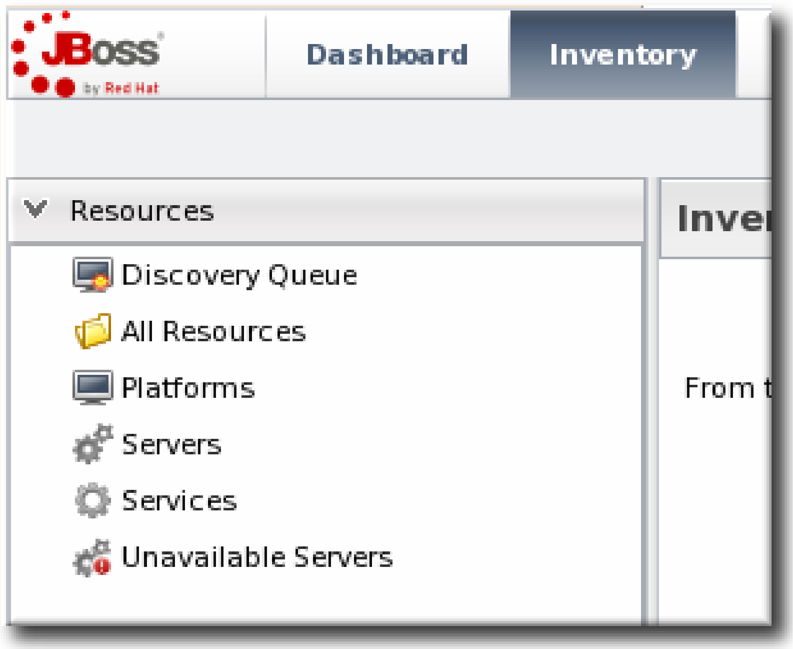
For every alert, Tim also configures an email notification along with the other responses, so that a member of the IT staff can evaluate any website performance problems and take additional actions if necessary.

21.4. CONFIGURING EJB CALL-TIME METRICS

EJB method call-time measurements are not collected by default.

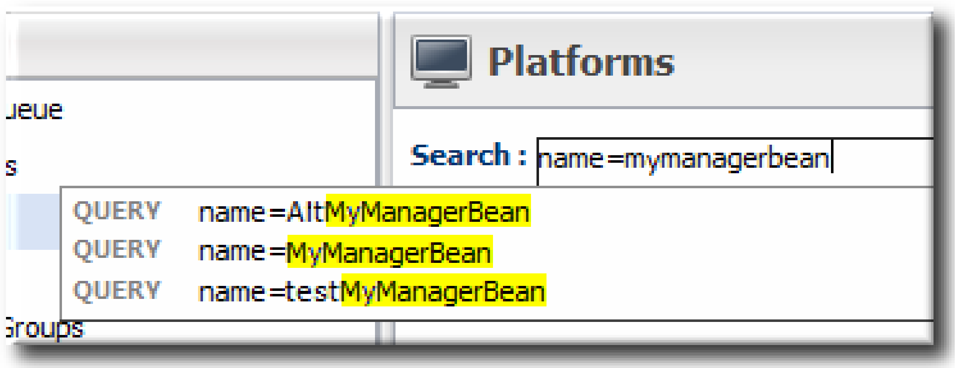
1. Click the **`Inventory`** tab in the top menu.

2. Select the **Services** menu table on the left, and then navigate to the EJB resource.



NOTE

It is probably easier to search for the session bean by name, if you know it.



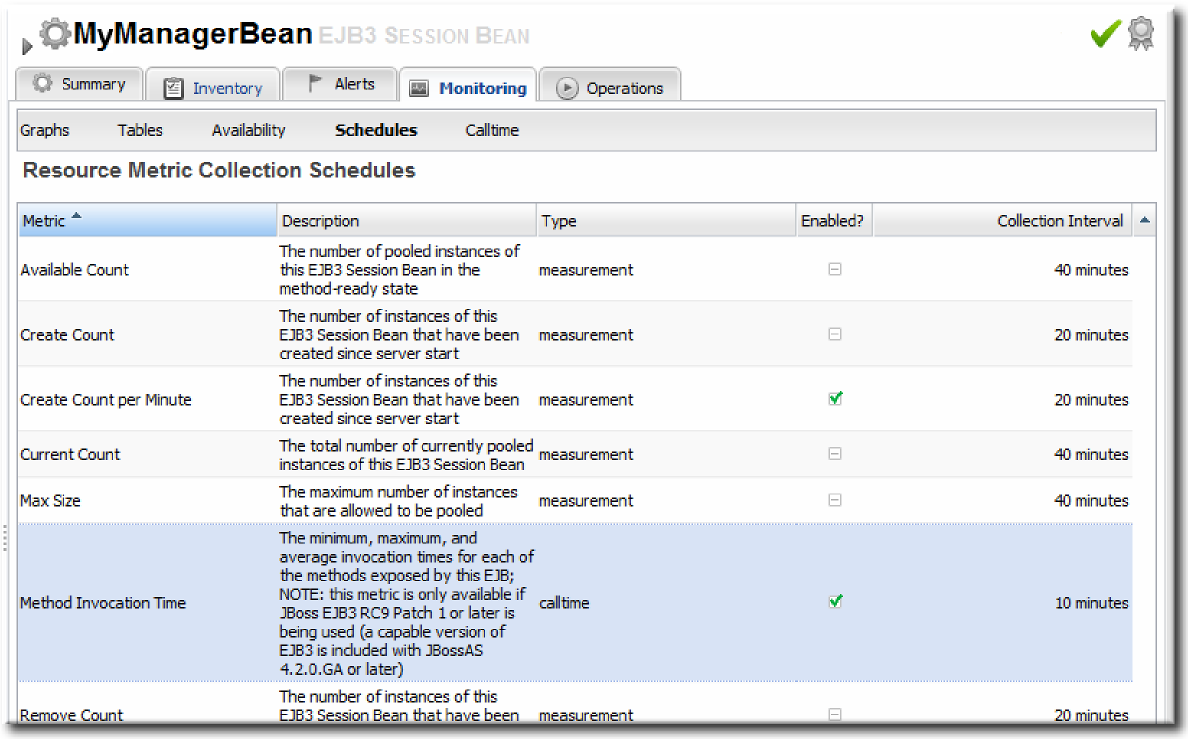
name=mymanagerbean

QUERY name=AltMyManagerBean
QUERY name=MyManagerBean
QUERY name=testMyManagerBean

3. Click the **Monitoring** tab on the EJB resource entry.

4. Click the **Schedules** subtab.

5. Select the **Method Invocation Time** metric. This metric is the *calltime* type.



Resource Metric Collection Schedules

Metric	Description	Type	Enabled?	Collection Interval
Available Count	The number of pooled instances of this EJB3 Session Bean in the method-ready state	measurement	☐	40 minutes
Create Count	The number of instances of this EJB3 Session Bean that have been created since server start	measurement	☐	20 minutes
Create Count per Minute	The number of instances of this EJB3 Session Bean that have been created since server start	measurement	✓	20 minutes
Current Count	The total number of currently pooled instances of this EJB3 Session Bean	measurement	☐	40 minutes
Max Size	The maximum number of instances that are allowed to be pooled	measurement	☐	40 minutes
Method Invocation Time	The minimum, maximum, and average invocation times for each of the methods exposed by this EJB; NOTE: this metric is only available if JBoss EJB3 RC9 Patch 1 or later is being used (a capable version of EJB3 is included with JBossAS 4.2.0.GA or later)	calltime	✓	10 minutes
Remove Count	The number of instances of this EJB3 Session Bean that have been	measurement	☐	20 minutes

6. Click the **Enable** at the bottom of the list.

21.5. CONFIGURING RESPONSE TIME METRICS FOR JBOSS EAP 6/AS 7

JBoss ON can monitor application performance by testing how responsive deployed applications are to client requests. Specifically, JBoss ON can determine how quickly an application responds to requests. This is called a *response time measurement*.

To collect response time metrics for a JBoss EAP 6/AS 7 server, first install the servlet JAR for the response time filter and configure the web server to use it. Then, enable the metrics collection for the web resource.

21.5.1. Installing the Response Time Filters

1. Make sure that you have created a management user to access the JBoss EAP 6 instance.

   For more information, see the JBoss AS 7.1 documentation.

2. In the connection properties for the EAP 6 instance, configure the agent to connect using the secure HTTPS management interface.

   NOTE

   To collect response time metrics, the EAP 6 agent plug-in must be configured to connect to the secure HTTPS management interface.

3. Download the response time packages for JBoss from the JBoss ON UI. The response time filters are packaged as AS 7 modules. There are two modules to obtain:

```
rhq-rtfilter-module.zip
rhq-rtfilter-subsystem-module.zip
```

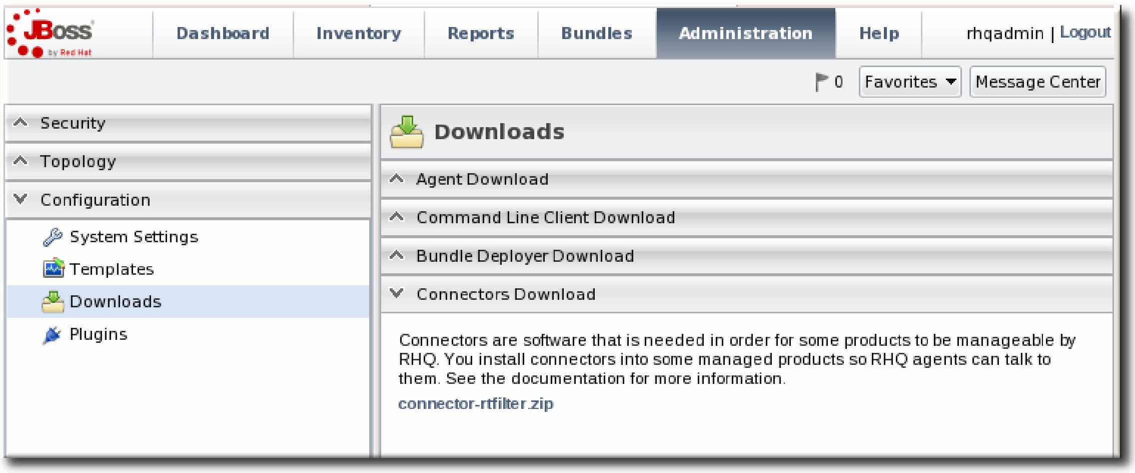
NOTE

This can also be done from the command line using **wget**:

```
[root@server ~]# wget
http://server.example.com:7080/downloads/connectors/rhq-
rtfilter-module.zip
[root@server ~]# wget
http://server.example.com:7080/downloads/connectors/rhq-
rtfilter-subsystem-module.zip
```

1. Click the **Administration** tab in the top menu.

2. In the **Configuration** menu box on the left, select the **Downloads** item.



Connectors are software that is needed in order for some products to be manageable by RHQ. You install connectors into some managed products so RHQ agents can talk to them. See the documentation for more information.

connector-rtfilter.zip

3. Click the **rhq-rtfilter-module.zip** and **rhq-rtfilter-subsystem-module.zip** links, and save the files to an accessible directory, like the **/tmp** directory.

4. Open the **modules/** directory for the JBoss EAP 6 instance. For example:

```
[root@server ~]# cd /opt/jboss-eap-6.0/modules/
```

5. Unzip the **rhq-rtfilter-module.zip** archive to install the response time filter JAR and the associated **module.xml** file.

```
[root@server modules]# unzip /tmp/rhq-rtfilter-module.zip
```

6. Open the configuration file for the server, **domain.xml** or **standalone.xml**.

7. Deploy the response time module globally by adding the module to the list of global modules in the **<subsystem>** element.

```
<profile...
 <subsystem xmlns="urn:jboss:domain:ee:1.1">
  <global-modules>
      <module name="org.rhq.helpers.rhq-rtfilter" slot="main"/>
  </global-modules>
 </subsystem>
</profile>
```

8. Save the file.

9. Unzip the **rhq-rtfilter-subsystem-module.zip** archive to install the subsystem response time filter JAR and the associated **module.xml** file.

   ```
   [root@server modules]# unzip /tmp/rhq-rtfilter-subsystem-module.zip
   ```

   This installs the filters as a subsystem for the application server or individual web apps.

10. After the filters have been installed, the JBoss EAP 6 server needs to be configured to use them.

    The response time filter can be deployed globally, for all web applications hosted by the EAP/AS instance, or it can be configured for a specific web application.

    To deploy the filter as a global subsystem:

    1. Open the configuration file for the server, **domain.xml** or **standalone.xml**.

    2. Add the an **<extensions>** element for the response time filter.

       ```
       <extension module="org.rhq.helpers.rhq-rtfilter-subsystem"/>
       ```

    3. Add a **<subsystem>** element beneath the **<profile** element.

       All that is required for response time filtering to work is the default **<subsystem>** element, without any optional parameters. However, the parameters can be uncommented and set as necessary; the different ones are described in Table 21.1, “Parameters Available for User-Defined <filter> Settings”.

       The **<subsystem>** element should be added even if none of the optional parameters are set.

       ```
       <subsystem xmlns="urn:rhq:rtfilter:1.0">
           <!-- Optional parameters.

               <init-param>
                   <param-name>chopQueryString</param-name>
                   <param-value>true</param-value>
               </init-param>
               <init-param>
                   <param-name>logDirectory</param-name>
                 <param-value>/tmp</param-value>
               </init-param>
               <init-param>
                   <param-name>logFilePrefix</param-name>
                   <param-value>localhost_7080_</param-value>
               </init-param>
               <init-param>
                   <param-name>dontLogRegEx</param-name>
                   <param-value></param-value>
               </init-param>
               <init-param>
                 <param-name>matchOnUriOnly</param-name>
       ```

```
                <param-value>true</param-value>
            </init-param>
            <init-param>
                <param-name>timeBetweenFlushesInSec</param-name>
                <param-value>73</param-value>
            </init-param>
            <init-param>
                <param-name>flushAfterLines</param-name>
                <param-value>13</param-value>
            </init-param>
            <init-param>
                <param-name>maxLogFileSize</param-name>
                <param-value>5242880</param-value>
            </init-param>
-->
</subsystem>
```

To configure the response time filters for an individual web application:

1. Open the web application's **web.xml** file.

```
[root@server ~]# vim WARHomeDir/WEB-INF/web.xml
```

2. Add the filter and, depending on the configuration, filter mapping elements to the file. This activates the response time filtering.

   All that is required for response time filtering to work is the default **<filter>** element, without any optional parameters. However, the parameters can be uncommented and set as necessary; the different ones are described in Table 21.1, “Parameters Available for User-Defined <filter> Settings”.

```
    <filter>
        <filter-name>RhqRtFilter</filter-name>
        <filter-
class>org.rhq.helpers.rtfilter.filter.RtFilter</filter-class>

<!-- Optional parameters.

            <init-param>
                <param-name>chopQueryString</param-name>
                <param-value>true</param-value>
            </init-param>
            <init-param>
                <param-name>logDirectory</param-name>
               <param-value>/tmp</param-value>
            </init-param>
            <init-param>
                <param-name>logFilePrefix</param-name>
                <param-value>localhost_7080_</param-value>
            </init-param>
            <init-param>
                <param-name>dontLogRegEx</param-name>
                <param-value></param-value>
            </init-param>
            <init-param>
```

```
                <param-name>matchOnUriOnly</param-name>
                <param-value>true</param-value>
             </init-param>
             <init-param>
                 <param-name>timeBetweenFlushesInSec</param-name>
                 <param-value>73</param-value>
             </init-param>
             <init-param>
                 <param-name>flushAfterLines</param-name>
                 <param-value>13</param-value>
             </init-param>
             <init-param>
                 <param-name>maxLogFileSize</param-name>
                 <param-value>5242880</param-value>
             </init-param>
-->

   </filter>

   <!-- Use this only when also enabling the RhqRtFilter in the
filter
   <filter-mapping>
       <filter-name>RhqRtFilter</filter-name>
       <url-pattern>/*</url-pattern>
   </filter-mapping>
   -->
```

11. Restart the JBoss EAP/AS server to load the new `web.xml` settings.

Table 21.1. Parameters Available for User-Defined <filter> Settings

Parameter	Description
chopQueryString	Only the URI part of a query will be logged if this parameter is set to true. Otherwise the whole query line will be logged. Default is true.
logDirectory	The directory where the log files will be written to. Default setting is `{jboss.server.log.dir}/rt/` (usually `server/xxx/log/rt`). If this property is not defined, the fallback is `{java.io.tmpdir}/rt/` (`/tmp/` on UNIX®, and `~/Application Data/Local Settings/Temp` – check the **TEMP** environment variable) is used. If you specify this init parameter, no directory `rt/` will be created, but the directory you have provided will be taken literally.
logFilePrefix	A prefix that is put in front of the log file names. Default is the empty string.
dontLogRegEx	A regular expression that is applied to query strings. See java.util.regex.Pattern. If the parameter is not given or an empty string, no pattern is applied.

Parameter	Description
matchOnUriOnly	Should the dontLogRegEx be applied to the URI part of the query (true) or to the whole query string (false). Default is true.
timeBetweenFlushesInSec	Log lines are buffered by default. When the given number of seconds have passed and a new request is received, the buffered lines will be flushed to disk even if the number of lines to flush after (see next point) is not yet reached.. Default value is 60 seconds (1 Minute).
flushAfterLines	Log lines are buffered by default. When the given number of lines have been buffered, they are flushed to disk. Default value is 10 lines.
maxLogFileSize	The maximum allowed size, in bytes, of the log files; if a log file exceeds this limit, the filter will truncate it; the default value is 5242880 (5 MB).

vHostMappingFile	This properties file must exist on the Tomcat process classpath. For example, in the ../conf/vhost-mappings.properties. The file contains mappings from the 'incoming' vhost (server name) to the vhost that should be used as the prefix in the response time log file name. If no mapping is present (no file or no entry response times are set), then the incoming vhost (server name) is used. For example:
	
	`pickeldi.users.acme.com=pickeldi`
	`pickeldi=`
	`%HOST%=`
	
	The first mapping states that if the incoming vhost is 'host1.users.acme.com', then the log file name should get a vhost of 'host1' as prefix, separated by a _ from the context root portion. The second mapping states that if the 'incoming' vhost is 'host1', then no prefix, and no _, should be used. The third mapping uses a special left-hand-side token, '%HOST%'. This mapping states that if the 'incoming' vhost is a representation of localhost then no prefix, and no _ , should be used.
	
	%HOST% will match the host name, or canonical host name or IP address, as returned by the implementation of InetAddress.getLocalHost().
	
	The second and third mappings are examples of empty right hand side, but could just as well have provided a vhost.
	
	This is a one time replacement. There is no recursion in the form that the result of the first line would then be applied to the second one.

21.5.2. Enabling the Call-Time Metric

Response time metrics are configured on a live application deployment. A deployment resource is a child of either a standalone EAP 6 server or of a server group.

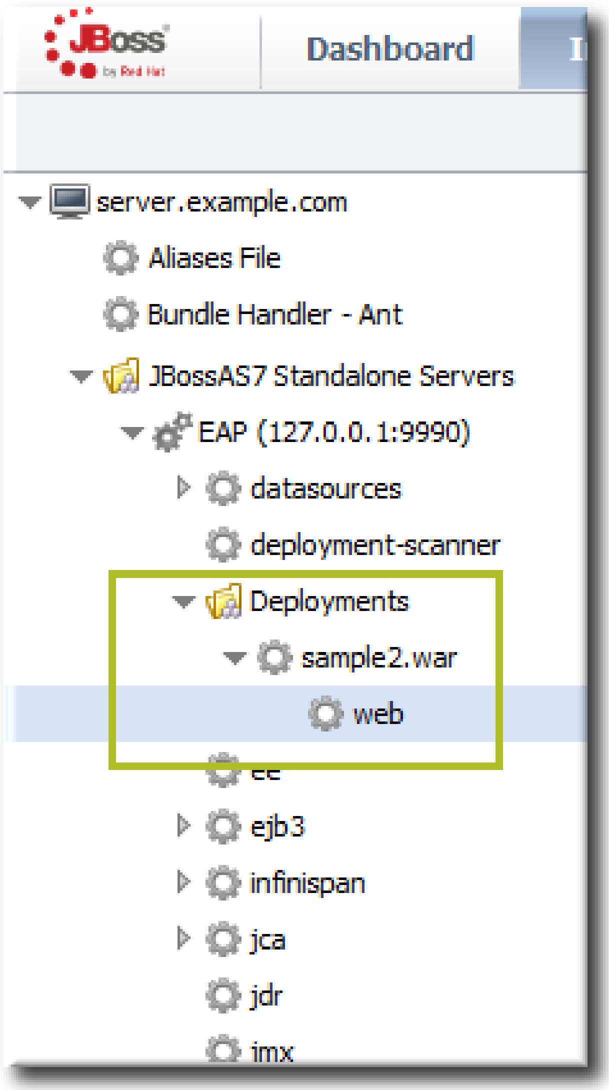
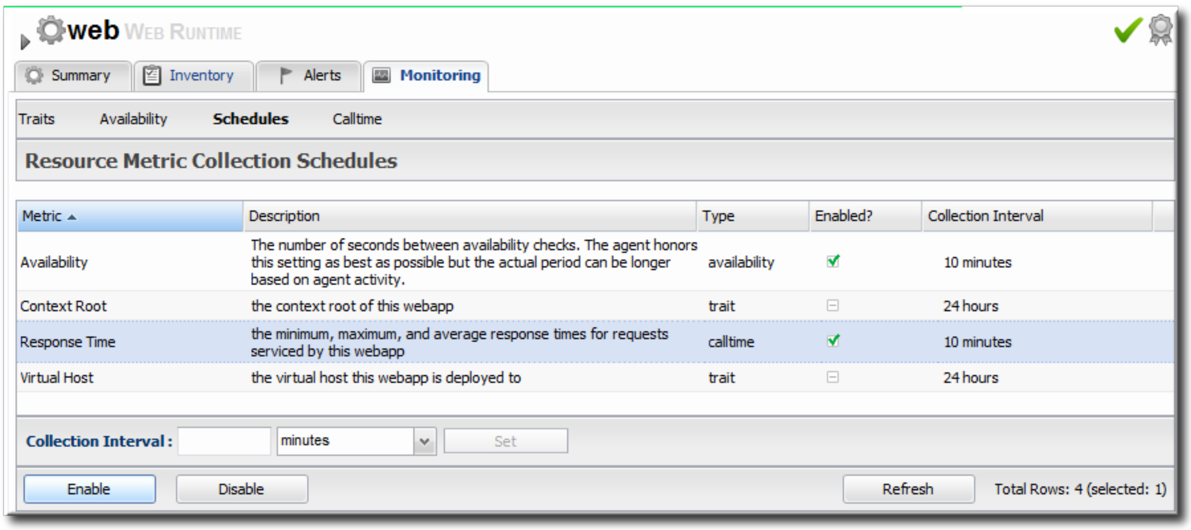


Figure 21.2. Web Runtime Resource

1. Click the **`Inventory`** tab in the top menu.
2. Click the **`Servers - Top Level Imports`** item, and select the JBoss EAP 6 resource.
3. Navigate to the deployment resource, and expand the application to the web subsystem.
4. Click the **`Monitoring`** tab on the web resource entry.
5. Click the **`Schedules`** subtab.
6. Select the **`Response Time`** metric. This metric is the *calltime* type.



Resource Metric Collection Schedules

Metric ▲	Description	Type	Enabled?	Collection Interval
Availability	The number of seconds between availability checks. The agent honors this setting as best as possible but the actual period can be longer based on agent activity.	availability	✓	10 minutes
Context Root	the context root of this webapp	trait		24 hours
Response Time	the minimum, maximum, and average response times for requests serviced by this webapp	calltime	✓	10 minutes
Virtual Host	the virtual host this webapp is deployed to	trait		24 hours

Collection Interval: minutes Set

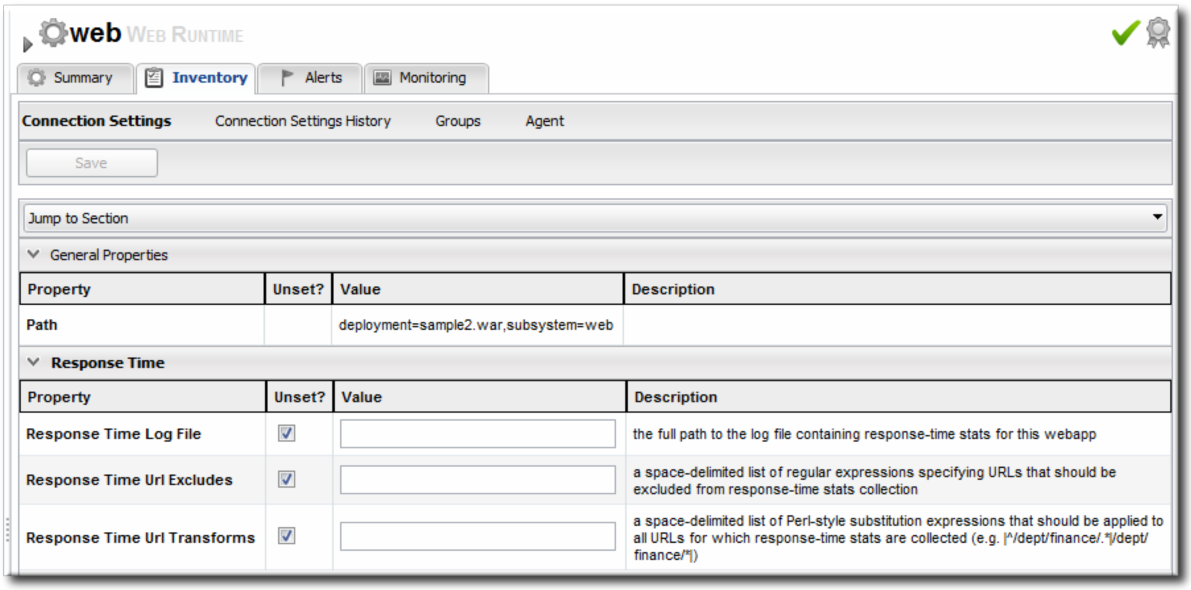
Enable Disable Refresh Total Rows: 4 (selected: 1)

7. Click the **`Enable`** at the bottom of the list.

8. Click the **`Inventory`** tab on the web entry.

9. Select the **`Connection Settings`** subtab.

10. Unset the check boxes for the response time configuration and fill in the appropriate values for the web application.



General Properties

Property	Unset?	Value	Description
Path		deployment=sample2.war,subsystem=web	

Response Time

Property	Unset?	Value	Description
Response Time Log File	✓		the full path to the log file containing response-time stats for this webapp
Response Time Url Excludes	✓		a space-delimited list of regular expressions specifying URLs that should be excluded from response-time stats collection
Response Time Url Transforms	✓		a space-delimited list of Perl-style substitution expressions that should be applied to all URLs for which response-time stats are collected (e.g. |^/dept/finance/.*|/dept/finance/*|)

- The response times log which is used by that specific web application. **The log file is a required setting for call-time data collection to work..**

- Any files, elements, or pages to exclude from response time measurements. The response times log records times for all resources the web server serves, including support files like CSS files and icons or background images.

- The same page can be accessed with different parameters passed along in the URL. The **`Response Time Url Transforms`** field provides a regular expression that can be used to strip or substitute the passed parameters.

21.6. SETTING UP RESPONSE TIME MONITORING FOR APACHE, EWS/TOMCAT, AND JBOSS EAP 5

Response time monitoring is not available from application or web servers by default; a special module must be installed which allows JBoss ON to collect response time metrics.

After the module is installed, then the HTTP metrics can be enabled for the resource.

21.6.1. Parameters for User-Defined <filter>s

Administrators can set rules for how response time metrics are collected for application and web servers. This are response time *filters*.

For response time monitoring to work, JBoss, Apache, and EWS/Tomcat all must have response time filters installed. Additional configuration options can be added to those default filters to customize the monitoring for the resource.

Table 21.2. Parameters for User-Defined <filter>s

Parameter	Description
chopQueryString	Only the URI part of a query will be logged if this parameter is set to true. Otherwise the whole query line will be logged. Default is true.
logDirectory	The directory where the log files will be written to. Default setting is {`jboss.server.log.dir}/rt/` (usually `server/xxx/log/rt`). If this property is not defined, the fallback is {`java.io.tmpdir}/rt/` (`/tmp/` on UNIX®, and `~/Application Data/Local Settings/Temp` – check the `TEMP` environment variable) is used. If you specify this init parameter, no directory `rt/` will be created, but the directory you have provided will be taken literally.
logFilePrefix	A prefix that is put in front of the log file names. Default is the empty string.
dontLogRegEx	A regular expression that is applied to query strings. See java.util.regex.Pattern. If the parameter is not given or an empty string, no pattern is applied.
matchOnUriOnly	Should the dontLogRegEx be applied to the URI part of the query (true) or to the whole query string (false). Default is true.
timeBetweenFlushesInSec	Log lines are buffered by default. When the given number of seconds have passed and a new request is received, the buffered lines will be flushed to disk even if the number of lines to flush after (see next point) is not yet reached.. Default value is 60 seconds (1 Minute).

Parameter	Description
flushAfterLines	Log lines are buffered by default. When the given number of lines have been buffered, they are flushed to disk. Default value is 10 lines.
maxLogFileSize	The maximum allowed size, in bytes, of the log files; if a log file exceeds this limit, the filter will truncate it; the default value is 5242880 (5 MB).

vHostMappingFile	This properties file must exist in the Tomcat process classpath. For example, in the `conf/vhost-mappings.properties` file. The file contains mappings from the 'incoming' vhost (server name) to the vhost that should be used as the prefix in the response time log file name. If no mapping is present (no file or no entry response times are set), then the incoming vhost (server name) is used. For example:
	
	 `pickeldi.users.acme.com=pickeldi`
	`pickeldi=`
	`%HOST%=`
	
	 The first mapping states that if the incoming vhost is 'host1.users.acme.com', then the log file name should get a vhost of 'host1' as prefix, separated by a _ from the context root portion. The second mapping states that if the 'incoming' vhost is 'host1', then no prefix, and no _, should be used. The third mapping uses a special left-hand-side token, '%HOST%'. This mapping states that if the 'incoming' vhost is a representation of localhost then no prefix, and no _ , should be used.
	
	 %HOST% will match the host name, or canonical host name or IP address, as returned by the implementation of InetAddress.getLocalHost().
	
	 The second and third mappings are examples of empty right hand side, but could just as well have provided a vhost.
	
	 This is a one time replacement. There is no recursion in the form that the result of the first line would then be applied to the second one.

21.6.2. Installing Response Time Filters for JBoss EAP/AS 5

To collect response time metrics for a JBoss EAP/AS 5 server, first install the servlet JAR for the response time filter and configure the web server to use it.

1. Download the Response Time packages for JBoss from the JBoss ON UI.

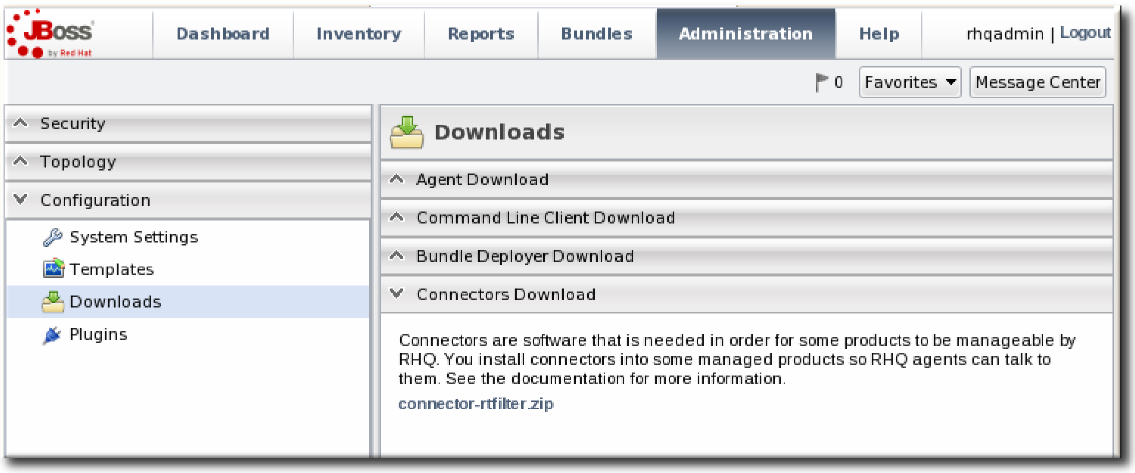
NOTE

This can also be done from the command line using **wget**:

```
[root@server ~]# wget
http://server.example.com:7080/downloads/connectors/conne
ctor-rtfilter.zip
```

1. Click the **Administration** tab in the top menu.

2. In the **Configuration** menu box on the left, select the **Downloads** item.



Connectors are software that is needed in order for some products to be manageable by RHQ. You install connectors into some managed products so RHQ agents can talk to them. See the documentation for more information.

3. Click the **connector-rtfilter.zip** link, and save the file.

2. Unzip the connectors.

```
[root@server ~]# unzip connector-rtfilter.zip
```

3. Copy the **rhq-rtfilter-*version*.jar** file into the **lib/** directory for the profile.

```
[root@server ~]# cp connector-rtfilter/rhq-rtfilter-version.jar
JbossHomeDir/server/profileName/lib/
```

JBoss EAP/AS already includes the **commons-logging.jar** file, which is also required for response time filtering.

4. Then, configure the **web.xml** for the EAP/AS instance.

The response time filter can be deployed globally, for all web applications hosted by the EAP/AS instance or it can be configured for a specific web application.

To configure it globally, edit the global **web.xml** file:

```
[root@server ~]# vim
JbossHomeDir/server/configName/deployers/jbossweb.deployer/web.xml
```

To configure it for a single web app, edit that one web app's **web.xml** file:

```
[root@server ~]# vim WARLocation/WEB-INF/web.xml
```

■

5. Add the filter and, depending on the configuration, filter mapping elements to the file. This activates the response time filtering.

   All that is required for response time filtering to work is the default **`<filter>`** element, without any optional parameters. However, the parameters can be uncommented and set as necessary; the different ones are described in Table 21.2, “Parameters for User-Defined <filter>s”.

```
   <filter>
       <filter-name>RhqRtFilter</filter-name>
       <filter-
class>org.rhq.helpers.rtfilter.filter.RtFilter</filter-class>

       <!-- Optional parameters.

       <init-param>
           <param-name>chopQueryString</param-name>
           <param-value>true</param-value>
       </init-param>
       <init-param>
           <param-name>logDirectory</param-name>
          <param-value>/tmp</param-value>
       </init-param>
       <init-param>
           <param-name>logFilePrefix</param-name>
           <param-value>localhost_7080_</param-value>
       </init-param>
       <init-param>
           <param-name>dontLogRegEx</param-name>
           <param-value></param-value>
       </init-param>
       <init-param>
          <param-name>matchOnUriOnly</param-name>
          <param-value>true</param-value>
       </init-param>
       <init-param>
           <param-name>timeBetweenFlushesInSec</param-name>
           <param-value>73</param-value>
       </init-param>
       <init-param>
           <param-name>flushAfterLines</param-name>
           <param-value>13</param-value>
       </init-param>
       <init-param>
           <param-name>maxLogFileSize</param-name>
           <param-value>5242880</param-value>
       </init-param>
       -->
   </filter>

   <!-- Use this only when also enabling the RhqRtFilter in the
filter
   <filter-mapping>
       <filter-name>RhqRtFilter</filter-name>
```

```
        <url-pattern>/*</url-pattern>
    </filter-mapping>
    -->
```

6. Restart the JBoss EAP/AS server to load the new `web.xml` settings.

7. Enable the HTTP metrics, as described in Section 21.6.5, “Configuring HTTP Response Time Metrics”, so that JBoss ON checks for the response time metrics on the application server.

21.6.3. Configuring Apache Servers for Response Time Metrics

1. To use the Response Time module, the Apache server needs to have been compiled with shared object support. For Red Hat Enterprise Linux systems and EWS servers, this is enabled by default.

   To verify that the Apache server was compiled with shared object support, use the **`apachectl -l`** command to list the compiled modules and look for the **`mod_so.c`** module:

   ```
   [root@server ~]# apachectl -l
   Compiled in modules:
     core.c
     prefork.c
     http_core.c
     mod_so.c
   ```

   Use the **`- -enable-module=so`** option:

   ```
   [jsmith@server ~]$ ./configure - -enable-module=so
   [jsmith@server ~]$ make install
   ```

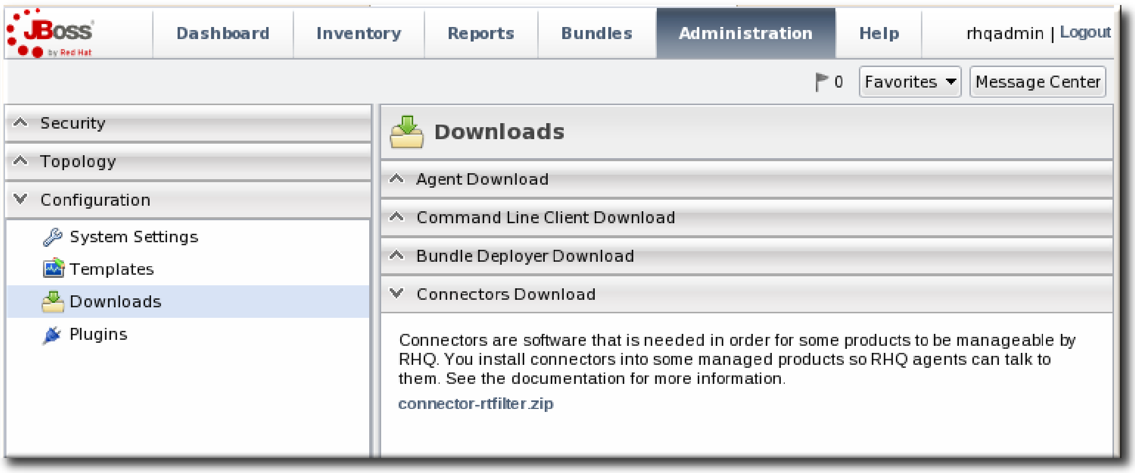
2. Download the Apache binaries from the JBoss ON UI.

   1. Log into the JBoss ON UI.

      ```
      https://server.example.com:7080
      ```

   2. Click the **`Administration`** tab in the top menu.

   3. In the **`Configuration`** menu box on the left, select the **Downloads** item.



Connectors are software that is needed in order for some products to be manageable by RHQ. You install connectors into some managed products so RHQ agents can talk to them. See the documentation for more information.
connector-rtfilter.zip

4. Click the **connector-apache.zip** link, and save the file.

3. Extract the Apache connectors.

```
[root@server ~]# unzip connector-apache.zip
```

4. Compile the Response Time module.

NOTE

apxs must be installed, and **make** must be installed and in the user PATH.

```
[root@server ~]# cd apacheMOduleRoot/apache-rt/sources
[root@server sources]# chmod +x build_apache_module.sh
[root@server sources]# ./build_apache_module.sh 2.x
apache_install_directory/bin/apxs
```

5. Then, install the Response Time module on the Apache server.

```
[root@server sources]# cp apache2.x/.libs/mod_rt.so
apache_install_directory/modules
```

6. Open the **httpd.conf** file. For example:

```
[root@server ~]# vim apache_install_directory/conf/httpd.conf
```

7. Enable the module in the Apache's **httpd.conf** file by appending this line to the end of the file:

```
LoadModule  rt_module  modules/mod_rt.so
LogFormat  "%S"  rt_log
```

When setting the log format, the variable **%S** has a capital S.

8. To configure response time logging for the main Apache server, add the following line at the top level of the file:

```
CustomLog  logs/myhost.com80_rt.log  rt_log
```

To configure response time logging for a virtual host, add the following line somewhere within the **<VirtualHost>** block:

```
CustomLog  logs/myhost.com8080_rt.log  rt_log
```

Make sure the response time log file name is different for the main server and each virtual host. Consider using the host and port from the **ServerName** directive be used to form the file name, such as *host_port_***rt.log**.

9. Restart the Apache server:

```
[root@server ~]# apachectl -k restart
```

10. To confirm that the Response Time module was installed successfully, check that the response time log files configured via the CustomLog directive now exist.

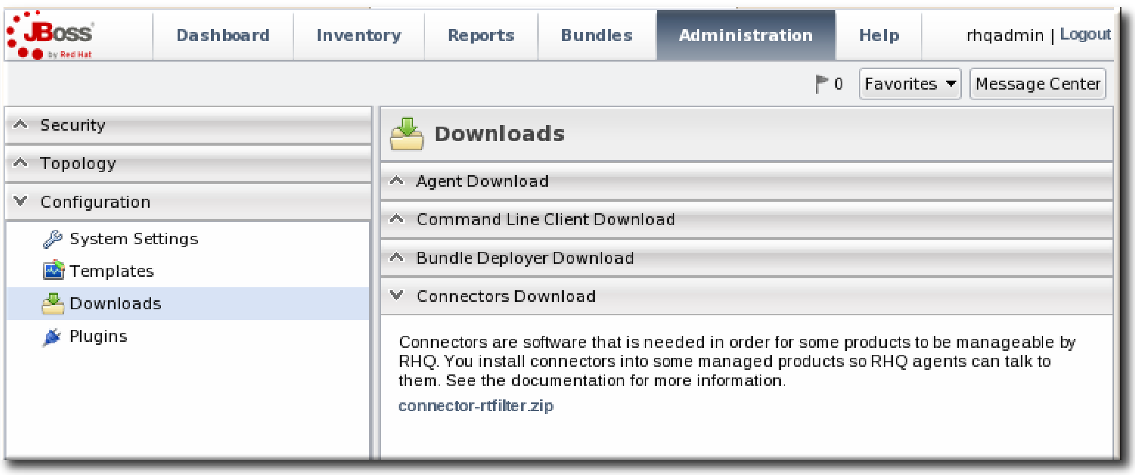
11. Enable the HTTP metrics, as described in Section 21.6.5, “Configuring HTTP Response Time Metrics”, so that JBoss ON checks for the response time metrics on the application server.

21.6.4. Installing Response Time Filters for Tomcat

1. Download the Response Time packages for Tomcat from the JBoss ON UI.

    1. Click the **Administration** tab in the top menu.

    2. In the **Configuration** menu box on the left, select the **Downloads** item.



Connectors are software that is needed in order for some products to be manageable by RHQ. You install connectors into some managed products so RHQ agents can talk to them. See the documentation for more information.

connector-rtfilter.zip

    3. Click the **connector-rtfilter.zip** link, and save the file.

2. Unzip the Response Time connectors.

```
unzip connector-rtfilter.zip
```

The package contains two JAR files, **commons-logging-*version*.jar** and **rhq-rtfilter-*version*.jar**. Tomcat 5 servers use only the **commons-logging-*version*.jar** file, while Tomcat 6 servers require both files.

3. Copy the appropriate JAR files into the Tomcat configuration directory. The directory location depends on the Tomcat or JBoss instance (for embedded Tomcat) being modified.

For example, on a standalone Tomcat 5.5:

```
cp commons-logging-version.jar /var/lib/tomcat5/server/lib/
```

On Tomcat 6:

```
cp rhq-rtfilter-version.jar /var/lib/tomcat6/lib/
cp commons-logging-version.jar /var/lib/tomcat6/lib/
```

For example, on an embedded Tomcat instance:

```
cp rhq-rtfilter-version.jar
JBoss_install_dir/server/default/deploy/jboss-web.deployer/
```

```
cp commons-logging-version.jar
JBoss_install_dir/server/default/deploy/jboss-web.deployer/
```

4. Open the **web.xml** file to add the filter definition. The exact location of the file depends on the server instance and whether it is a standalone or embedded server; several common locations are listed in Table 21.3, “web.xml Configuration File Locations” .

5. Add either a **<filter>** or a **<filter-mapping>** entry to configuration the Response Time filter in the Tomcat server. Either a **<filter>** or a **<filter-mapping>** entry can be used, but *not* both.

   The most basic filter definition references simply the Response Time filter name and class in the **<filter>** element. This loads the response time filter with all of the default settings.

```
<filter>
        <filter-name>RhqRtFilter </filter-name>
        <filter-class>org.rhq.helpers.rtfilter.filter.RtFilter
</filter-class>
</filter>
```

   The filter definition can be expanded with user-defined configuration values by adding **<init-param** elements. This loads the response time filter with all of the default settings.

```
<filter>
        <filter-name>RhqRtFilter </filter-name>
        <filter-class>org.rhq.helpers.rtfilter.filter.RtFilter
</filter-class>
        <init-param>
                <description>Name of vhost mapping file. This
properties file must be in the Tomcat process
classpath.</description>
                <param-name>vHostMappingFile</param-name>
                <param-value>vhost-mappings.properties</param-
value>
        </init-param>
...
</filter>
```

   The available parameters are listed in Table 21.2, “Parameters for User-Defined <filter>s”.

   Alternatively, set a **<filter-map>** entry which gives the name of the response time filter and pattern to use to match the URL which will be monitored.

```
<filter-mapping>
        <filter-name>RhqRtFilter </filter-name>
        <url-pattern>/* </url-pattern>
</filter-mapping>
```

NOTE

Put the Response Time filter in front of any other configured filter so that the response time metrics will include all of the other response times, total, in the measurement.

6. Restart the Tomcat instance to load the new configuration.

7. Enable the HTTP metrics, as described in Section 21.6.5, “Configuring HTTP Response Time Metrics”, so that JBoss ON checks for the response time metrics on the application server.

Table 21.3. web.xml Configuration File Locations

Tomcat Version	Embedded Server Type	File Location
Tomcat 6	Standalone Server	/var/lib/tomcat6/webapps/*project*/WEB-INF/web.xml
Tomcat 5	Standalone Server	/var/lib/tomcat5/webapps/*project*/WEB-INF/web.xml
Tomcat 6	EAP 5	*JBOSS_HOME*/server/*config*/deployers/jbossweb.deployer/web.xml
Tomcat 6	JBoss 4.2, JBoss EAP4	*JBOSS_HOME*/server/*config*/deploy/jboss-web.deployer/conf/web.xml
Tomcat 5.5	JBoss 4.0.2	*JBOSS_HOME*/server/*config*/deploy/jbossweb-tomcat55.sar/conf/web.xml
Tomcat 5.0	JBoss 3.2.6	*JBOSS_HOME*/server/*config*/deploy/jbossweb-tomcat50.sar/conf/web.xml
Tomcat 4.1	JBoss 3.2.3	*JBOSS_HOME*/server/*config*/deploy/jbossweb-tomcat41.sar/web.xml

21.6.5. Configuring HTTP Response Time Metrics

Configuring response time metrics is in some respects analogous to configuring events. The JBoss ON agent polls certain log files kept by the web server to identify the performance times for different resources served by the web server.

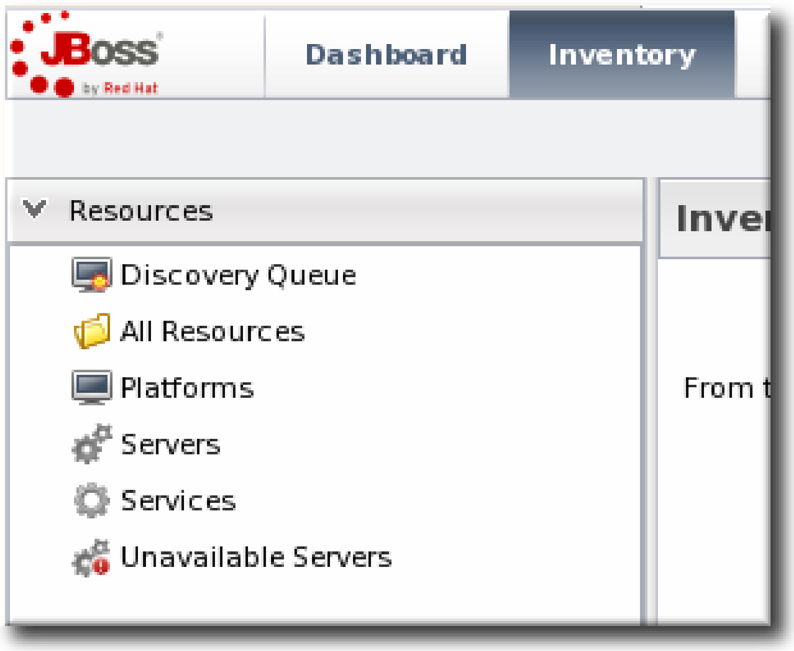
1. Install the response time filter for the web server.

   For Apache, simply install the filters; for Tomcat, there is additional configuration required to set up the filter entry in the `web.xml` file.

   If necessary, set up the filter entry in the `web.xml` file. See Section 21.6, “Setting up Response Time Monitoring for Apache, EWS/Tomcat, and JBoss EAP 5”.

2. Click the `Inventory` tab in the top menu.

3. Select the `Servers` menu table on the left, and then navigate to the web application, and select the web application context for which to run the response time monitoring.



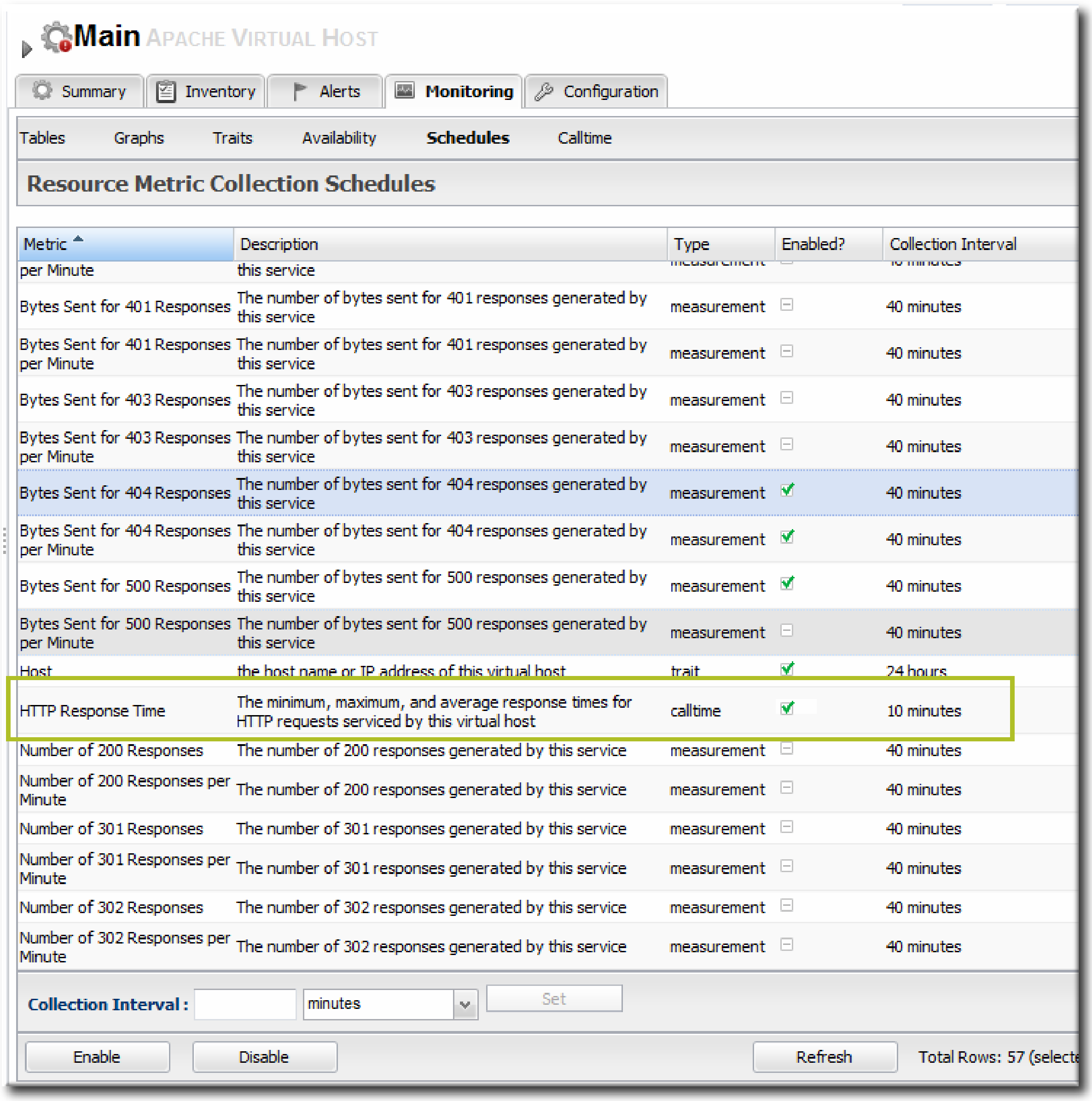
4. Click the **`Connection Settings`** tab on the web application context resource, and scroll to the **`Response Time`** configuration section.

5. Configure the response time properties for the web server. The agent has to know what log file the web server uses to record response time data.

   Optionally, the server can perform certain transformations on the collected data.

   - The response times log records times for all resources the web server serves, including support files like CSS files and icons or background images. These resources can be excluded from the response time calculations in the **`Response Time Url Excludes`** field.

   - The same page can be accessed with different parameters passed along in the URL. The **`Response Time Url Transforms`** field provides a regular expression that can be used to strip or substitute the passed parameters.



Resource Metric Collection Schedules

Metric	Description	Type	Enabled?	Collection Interval
per Minute	this service			
Bytes Sent for 401 Responses	The number of bytes sent for 401 responses generated by this service	measurement	☐	40 minutes
Bytes Sent for 401 Responses per Minute	The number of bytes sent for 401 responses generated by this service	measurement	☐	40 minutes
Bytes Sent for 403 Responses	The number of bytes sent for 403 responses generated by this service	measurement	☐	40 minutes
Bytes Sent for 403 Responses per Minute	The number of bytes sent for 403 responses generated by this service	measurement	☐	40 minutes
Bytes Sent for 404 Responses	The number of bytes sent for 404 responses generated by this service	measurement	✓	40 minutes
Bytes Sent for 404 Responses per Minute	The number of bytes sent for 404 responses generated by this service	measurement	✓	40 minutes
Bytes Sent for 500 Responses	The number of bytes sent for 500 responses generated by this service	measurement	✓	40 minutes
Bytes Sent for 500 Responses per Minute	The number of bytes sent for 500 responses generated by this service	measurement	☐	40 minutes
Host	the host name or IP address of this virtual host	trait	✓	24 hours
HTTP Response Time	The minimum, maximum, and average response times for HTTP requests serviced by this virtual host	calltime	✓	10 minutes
Number of 200 Responses	The number of 200 responses generated by this service	measurement	☐	40 minutes
Number of 200 Responses per Minute	The number of 200 responses generated by this service	measurement	☐	40 minutes
Number of 301 Responses	The number of 301 responses generated by this service	measurement	☐	40 minutes
Number of 301 Responses per Minute	The number of 301 responses generated by this service	measurement	☐	40 minutes
Number of 302 Responses	The number of 302 responses generated by this service	measurement	☐	40 minutes
Number of 302 Responses per Minute	The number of 302 responses generated by this service	measurement	☐	40 minutes

Collection Interval : minutes Set

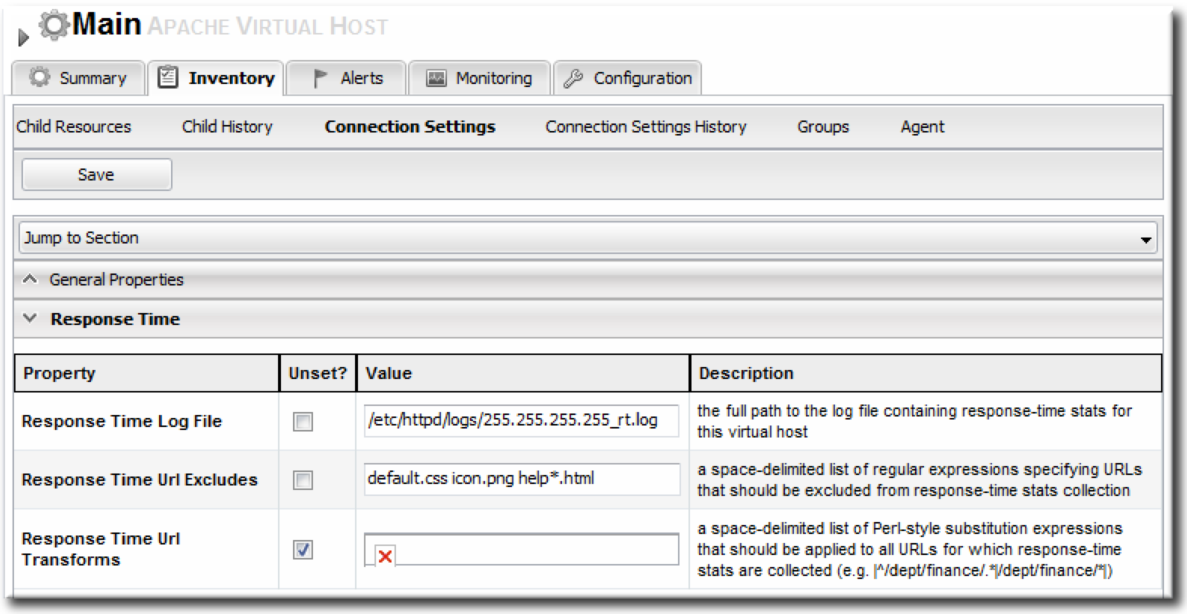
Enable Disable Refresh Total Rows: 57 (selecte

6. Click the **Save** button.

7. Click the **`Monitoring`** tab on the web server resource entry.

8. Click the **`Schedules`** subtab.

9. Select the **`HTTP Response Time`** metric. This metric is the *calltime* type.



Property	Unset?	Value	Description
Response Time Log File	☐	/etc/httpd/logs/255.255.255.255_rt.log	the full path to the log file containing response-time stats for this virtual host
Response Time Url Excludes	☐	default.css icon.png help*.html	a space-delimited list of regular expressions specifying URLs that should be excluded from response-time stats collection
Response Time Url Transforms	☑		a space-delimited list of Perl-style substitution expressions that should be applied to all URLs for which response-time stats are collected (e.g. |^/dept/finance/.*|/dept/finance/*|)

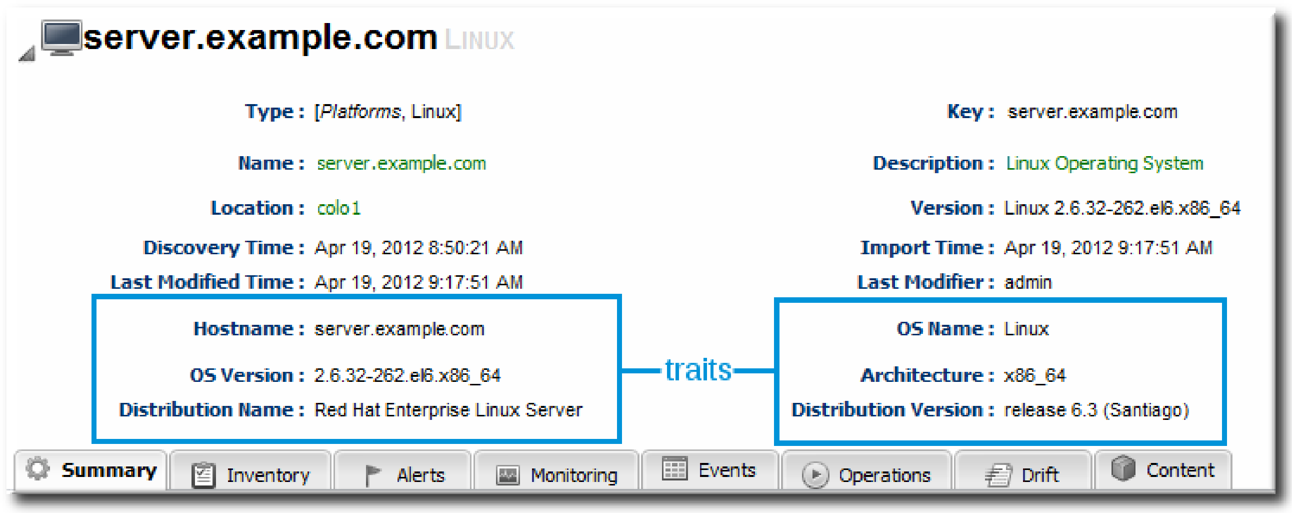
10. Click the **`Enable`** at the bottom of the list.

CHAPTER 22. RESOURCE TRAITS

One category of information that is collected about a resource is its *traits*. Traits are descriptive information, usually information that does not change very frequently.

For example, traits for a platform include its operating system name and version, its distribution, its architecture, and its hostname. Most resources have similar identifying information, such as a version number or vendor information.

Traits are in-between information. They are detectable and are detected by the JBoss ON agent's monitoring processes. But they are also generalized descriptive information. Trait information is even shown on the resource's details page.



server.example.com LINUX

Type :	[*Platforms*, Linux]	Key :	server.example.com
Name :	server.example.com	Description :	Linux Operating System
Location :	colo1	Version :	Linux 2.6.32-262.el6.x86_64
Discovery Time :	Apr 19, 2012 8:50:21 AM	Import Time :	Apr 19, 2012 9:17:51 AM
Last Modified Time :	Apr 19, 2012 9:17:51 AM	Last Modifier :	admin
Hostname :	server.example.com	OS Name :	Linux
OS Version :	2.6.32-262.el6.x86_64	Architecture :	x86_64
Distribution Name :	Red Hat Enterprise Linux Server	Distribution Version :	release 6.3 (Santiago)

Figure 22.1. Resource Details

The traits that are collected are defined in the resource plug-in itself, so this information is viewable but not configurable through the UI. The list of traits for each resource type is covered in the *Resource Reference: Monitoring, Operation, and Configuration Options*.

22.1. COLLECTION INTERVAL

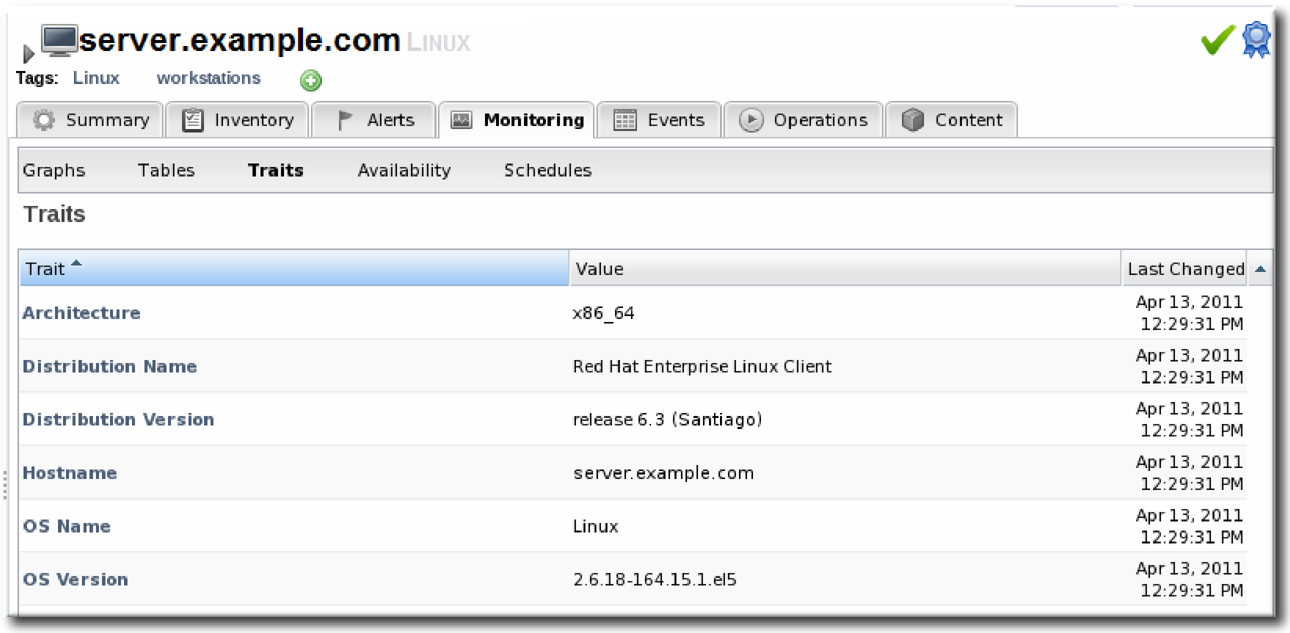
By default for most resource types, traits are checked every 24 hours. Because traits change infrequently, they do not need to be collected very often.

The trait collection interval is configured the same as metrics collection intervals. To change the collection interval for a trait, see Section 19.3.2, “Setting Collection Intervals for a Specific Resource” .

22.2. VIEWING TRAITS

The **`Traits`** subtab in the **`Monitoring`** tab for a resource displays three pieces of information:

- The trait name. The traits which are monitored for a resource are defined with other monitoring settings in the resource type's plug-in descriptor.
- The trait value.
- The time of the last collection where a change in trait information was detected.



Trait	Value	Last Changed
Architecture	x86_64	Apr 13, 2011 12:29:31 PM
Distribution Name	Red Hat Enterprise Linux Client	Apr 13, 2011 12:29:31 PM
Distribution Version	release 6.3 (Santiago)	Apr 13, 2011 12:29:31 PM
Hostname	server.example.com	Apr 13, 2011 12:29:31 PM
OS Name	Linux	Apr 13, 2011 12:29:31 PM
OS Version	2.6.18-164.15.1.el5	Apr 13, 2011 12:29:31 PM

Figure 22.2. Trait Charts

22.3. *EXTENDED EXAMPLE*: ALERTING AND TRAITS

The Setup

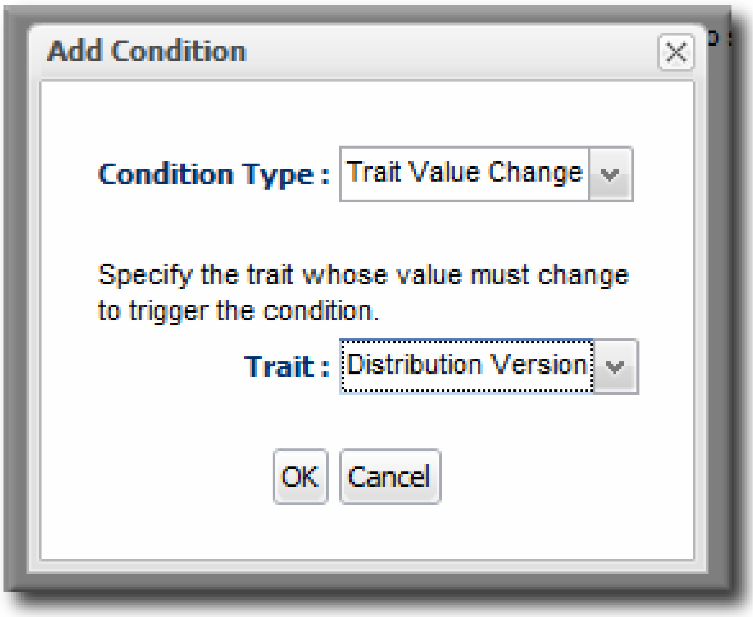
Trait information tends to be static. While traits can, and do, change, they do so infrequently. Also, traits convey descriptive information about a resource, not state data or dynamic measurements, so traits are not critical for IT administrators to track closely.

However, trait information can be valuable in letting administrators know when some configuration or package on the resource has changed because one trait that most resources collect is version information. If a version number changes, then some update has occurred on the underlying resource.

What to Do

For example, Tim the IT Guy has automatic updates configured for his Red Hat Enterprise Linux development and QA servers. Because his production environment has controlled application and system updates, there are no automatic updates for those servers.

Tim wants to be informed of version changes, but they are not necessarily a big deal. JBoss ON can issue an alert when a trait changes, with the `Trait Value Change` condition. (Cf. Section 25.1.2, “Basic Procedure for Setting Alerts for a Resource”.)



Condition Type : Trait Value Change

Specify the trait whose value must change to trigger the condition.

Trait : Distribution Version

Figure 22.3. Trait Alert Condition

The alert for the development and QA systems is simple:

- He sets two conditions, using an OR operator. The alert triggers when the distribution version changes *or* when the operating system version changes. This catches both minor and major updates to the operating system or kernel.
- It is set to low priority so it is informative but not critical.
- Tim decides that the alert notification is sent to his JBoss ON user, so he sees notifications when he logs in. He could also configure an email notification for high-priority resources.

For Tim's production resources, he sets the alert priority to high and uses an email notification to multiple IT administrators so that they are quickly aware of any change to the production systems.

CHAPTER 23. RESOURCES WHICH REQUIRE SPECIAL CONFIGURATION FOR MONITORING

Web servers (Tomcat and Apache resources) require additional configuration for JBoss ON to be able to manage and monitor those resources.

23.1. CONFIGURING TOMCAT/JWS SERVERS FOR MONITORING

For instructions on setting up Tomcat or Red Hat JBoss Web Server (JWS) for monitoring with JBoss Operations Network, see the JBoss Web Server Installation Guide chapter on Monitoring Red Hat JBoss Web Server with JBoss ON

NOTE

For more detail on configuring Tomcat, see the Tomcat documentation.

23.2. CONFIGURING THE APACHE SNMP MODULE

To discover an Apache server's virtual hosts and collect metrics for them, the SNMP module must be configured on that Apache server.

Apache 2.2 is supported on Red Hat Enterprise Linux and Windows platforms.

IMPORTANT

To use the Response Time module, the Apache server needs to have been compiled with shared object support. For Red Hat Enterprise Linux systems and EWS servers, this is enabled by default.

To verify that the Apache server was compiled with shared object support, use the **apachectl -l** command to list the compiled modules and look for the **mod_so.c** module:

```
[root@server ~]# apachectl -l
Compiled in modules:
  core.c
  prefork.c
  http_core.c
  mod_so.c
```

Use the **--enable-module=so** option:

```
$ ./configure --enable-module=so
$ make install
```

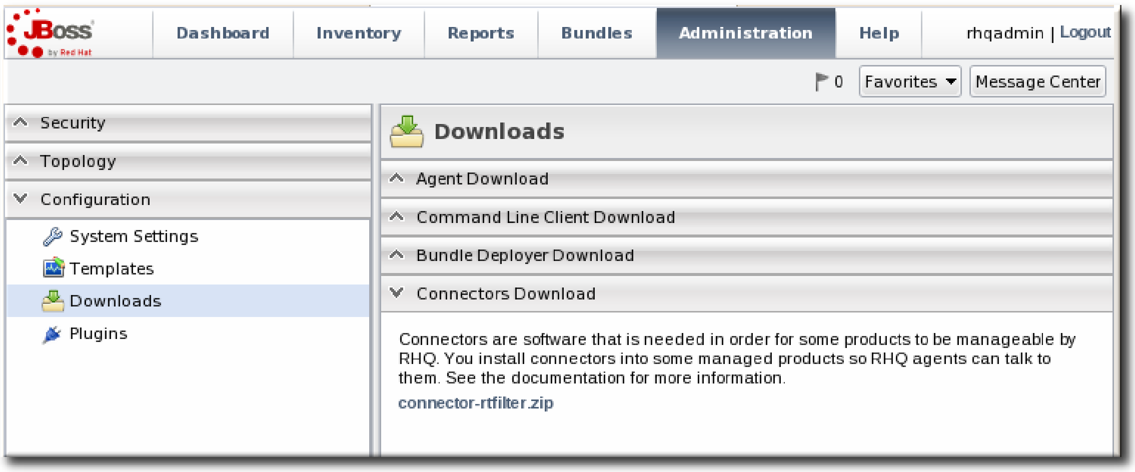
1. Download the Apache binaries from the JBoss ON UI.

    1. Log into the JBoss ON UI.

    ```
    https://server.example.com:7080
    ```

2. Click the **Administration** tab in the top menu.

3. In the **Configuration** menu box on the left, select the **Downloads** item.



Connectors are software that is needed in order for some products to be manageable by RHQ. You install connectors into some managed products so RHQ agents can talk to them. See the documentation for more information.
connector-rtfilter.zip

4. Scroll to **Connector Downloads**, and click the **connector-apache.zip** link to download the Apache connectors.

2. Unzip the Apache connectors in a directory that is accessible to the JBoss ON agent.

```
unzip connector-apache.zip
```

3. Each Apache version and platform has its own package that contains the Apache-SNMP connectors. Extract the Apache connectors in a directory that is accessible to the JBoss ON agent. Binaries are available for Red Hat Enterprise Linux 32-bit and 64-bit and Windows 32-bit.

   For example, on Red Hat Enterprise Linux 32-bit:

```
[jsmith@server ~]$ cd apacheModuleRoot/apache-snmp/binaries/
[jsmith@server binaries]$ tar xjvf snmp_module-x86-linux-
apache#.tar.bz2
```

   # is the Apache server version number.

NOTE

Apache connectors can be compiled for other platforms, like Solaris, from the source files in ***apacheRoot*/apache-snmp/binaries/sources**. For example:

```
[jsmith@server ~]$ cd
JON_AGENT_INSTALL_DIR/product_connectors/apache-
snmp/sources
[jsmith@server sources]$ ./build_apache_snmp.sh
APACHE_VERSION APACHE_2.x_INSTALL_DIR/bin/apxs
```

To compile the Apache-SNMP connector, **apxs**, **perl**, **make**, and **automake** must all be installed and in user **PATH**.

4. Install the module. For example:

```
[root@server ~]# cd apacheModuleRoot/apache-
snmp/binaries/snmp_module_#

[root@server snmp_module]# cp module/*
apache_install_directory/modules

[root@server snmp_module]# cp conf/* apache_install_directory/conf

[root@server snmp_module]# mkdir apache_install_directory/var
```

On Windows:

```
> xcopy /e JON_AGENT_INSTALL_DIR\product_connectors\apache-
snmp\binaries\x86
```

5. Open the **`httpd.conf`** file for editing. For example:

```
[root@server ~]# vim apache_install_directory/conf/httpd.conf
```

6. Enable the module by adding these lines to the **`httpd.conf`** file.

```
LoadModule snmpcommon_module modules/libsnmpcommon.so
LoadModule snmpagt_module modules/libsnmpmonagt.so

SNMPConf   conf
SNMPVar    var
```

For Windows:

```
LoadModule snmpcommon_module modules/snmpcommon.so
LoadModule snmpagt_module modules/snmpmonagt.so

SNMPConf   conf
SNMPVar    var
```

7. Make sure the main Apache configuration section, as well as each **`<VirtualHost>`** configuration block, contains a **`ServerName`** directive with a port. The SNMP module uses this directive to uniquely identify the main server and each virtual host, so each **`ServerName`** directive must contain a unique value. For example:

```
ServerName main.example.com:80
...

<VirtualHost vhost1.example.com:80>
ServerName vhost1.example.com:80
...
</VirtualHost>
```

8. If there is more than one Apache instance on the same machine, it is possible to use different SNMP files for each instance.

   1. Each Apache instance has its own **`httpd.conf`** file. Set the **`SNMPConf`** directory in each file to its own SNMP configuration directory. For example, for instance1:

```
vim instance1-httpd.conf

SNMPConf /opt/apache-instance1/conf
```

Then, for instance2:

```
vim instance2-httpd.conf

SNMPConf /opt/apache-instance2/conf
```

Each **`snmpd.conf`** file should be in the specified directory.

2. Edit the **`agentaddress`** property in *apache_install_directory*/**`conf/snmpd.conf`** so that each instance has a different value agent address and port, so there is no conflict between instances.

   See the snmpd.conf documentation for a description of this property's syntax.

9. Restart the Apache server. For example:

```
apachectl -k restart
```

10. Verify that the SNMP module was properly installed. If the module is loaded, then there will be lines referencing the SNMP module in the errors log:

```
grep SNMP apache_installation_dir/logs/error_log

[Wed Mar 19 09:54:34 2008] [notice] Apache/2.0.63 (Unix)
CovalentSNMP/2.3.0 configured -- resuming normal operations
[Wed Mar 19 09:54:35 2008] [notice] SNMP: CovalentSNMP/2.3.0 started
(user '1000' - SNMP address '1610' - pid '26738')
```

23.3. METRICS COLLECTION CONSIDERATIONS WITH APACHE AND SNMP

Three metrics show values of zero when monitoring an Apache instance with the SNMP module:

- Bytes received for GET requests per minute
- Bytes received for POST requests per minute
- Total number of bytes received per minute

This is because of how SNMP interprets information from the request body. First, SNMP provides various length values for the request body and a GET request does not have a body, so GET responses are not calculated and, therefore, have a value of zero. Second, Apache does not calculate a request body size if there is request chunking.

CHAPTER 24. STORING MONITORING DATA

JBoss ON monitoring information reveals both current measurements and historical trends and averages. JBoss ON stores data in a kind of cascade, where raw data are aggregated and compressed on a schedule. This preserves the trends of data without inflating the size of the monitoring data. Data are handled like this:

- Raw metrics are collected every few minutes and are aggregated in a rolling average in one-hour windows to produce minimum, average, and maximum values.
- One-hour values are combined and averaged in six-hour periods.
- Six-hour periods are combined and aggregated into 24-hour (1 day) windows.

24.1. CHANGING STORAGE LENGTHS FOR MONITORING DATA

24.1.1. Default Storage Lengths

The raw measurements, one-hour periods, and six-hour periods are preserved in the JBoss ON database for a **predefined** lengths of time.

Table 24.1. Storage Times for Data

Data	Length of Time	Configurable or Hardcoded	SQL or NoSQL Database
Raw measurements	7 days	Hardcoded	NoSQL
1-hour aggregate	14 days	Hardcoded	NoSQL
6-hour aggregate	31 days	Hardcoded	NoSQL
24-hour aggregate	365 days	Hardcoded	NoSQL
Traits	365 days	Configurable	SQL
Availability data	365 days	Configurable	SQL
Events data	14 days	Configurable	SQL
Response-time metrics	31 days	Configurable	SQL

NOTE

The storage size of raw metrics can grow very quickly. Be careful when adjusting the storage time of raw metrics to account for the size of your database and the number and frequency of all metrics collected across your inventory.

If you need to store raw data for long periods, consider exporting the raw data from the database and archiving it separately.

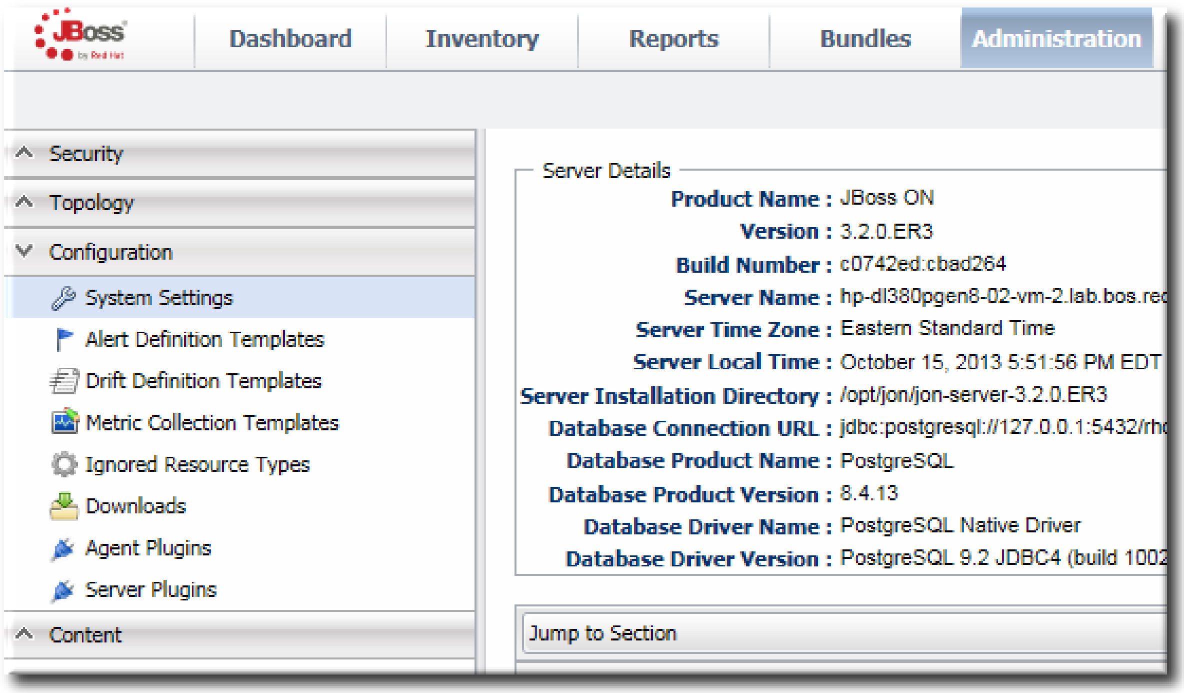
NOTE

All collected metrics are stored in a NoSQL database, in a specified storage node. Additionl nodes can be added on other systems, in a cluster, as the number of resources or collected metrics increases.

All other monitoring data (as well as all other data stored and used by JBoss ON) are stored in the backend SQL database.

24.1.2. Changing the Storage Times for Different Monitoring Data

1. In the `System Configuration` menu, select the **Settings** item.

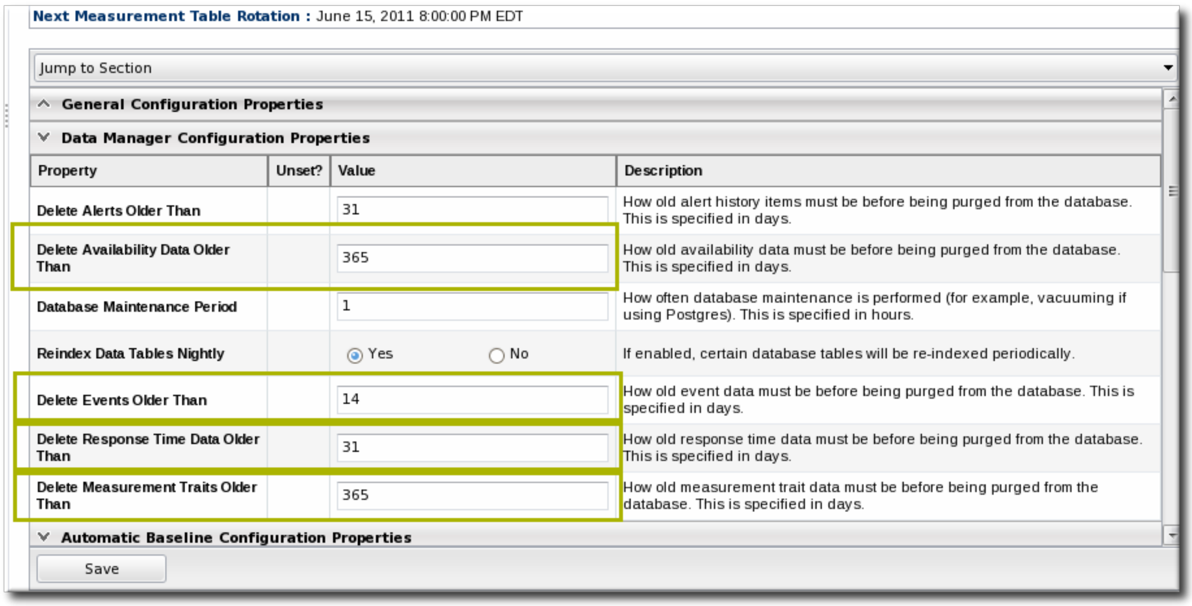


Server Details

Product Name :	JBoss ON
Version :	3.2.0.ER3
Build Number :	c0742ed:cbad264
Server Name :	hp-dl380pgen8-02-vm-2.lab.bos.red
Server Time Zone :	Eastern Standard Time
Server Local Time :	October 15, 2013 5:51:56 PM EDT
Server Installation Directory :	/opt/jon/jon-server-3.2.0.ER3
Database Connection URL :	jdbc:postgresql://127.0.0.1:5432/rhq
Database Product Name :	PostgreSQL
Database Product Version :	8.4.13
Database Driver Name :	PostgreSQL Native Driver
Database Driver Version :	PostgreSQL 9.2 JDBC4 (build 1002

2. Scroll to the `Data Manager Configuration Properties` section.

3. Change the storage times for the different types of monitoring data.



Next Measurement Table Rotation : June 15, 2011 8:00:00 PM EDT

Jump to Section
General Configuration Properties
Data Manager Configuration Properties

Property	Unset?	Value	Description
Delete Alerts Older Than		31	How old alert history items must be before being purged from the database. This is specified in days.
Delete Availability Data Older Than		365	How old availability data must be before being purged from the database. This is specified in days.
Database Maintenance Period		1	How often database maintenance is performed (for example, vacuuming if using Postgres). This is specified in hours.
Reindex Data Tables Nightly		[✓ Yes] No	If enabled, certain database tables will be re-indexed periodically.
Delete Events Older Than		14	How old event data must be before being purged from the database. This is specified in days.
Delete Response Time Data Older Than		31	How old response time data must be before being purged from the database. This is specified in days.
Delete Measurement Traits Older Than		365	How old measurement trait data must be before being purged from the database. This is specified in days.

There are four settings that relate directly to storing monitoring data:

- Response time data for web servers and EJB resources. This is kept for one month (31 days) by default.
- Events information, meaning all of the log files generated by the agent for the resource. The default storage time for event logs is two weeks.
- Traits for resources. The default time is one year (365 days).
- Availability information. The default time is one year (365 days).

24.2. EXPORTING RAW DATA

Raw monitoring data is, by default, purged from the database every week. To save the raw data, export it using the CLI.

The **`MeasurementDataManager`** class has a method to find the metric values for a specific resource within a certain time range:

```
findDataForResource(resourceId,
[metricId],startTime,endTime,numberOfRecords)
```

For example, for a resource with the ID 10003 and a metric ID of 10473:

```
exporter.file = '/export/metrics/metrics.csv'
exporter.format = 'csv'
var start = new Date() - 8* 3600 * 1000;
var end = new Date()
var data = MeasurementDataManager.findDataForResource(10003,
[10473],start,end,60)
exporter.write(data.get(0))
```

24.3. DEPLOYING AND MANAGING STORAGE NODES

Raw metric data and aggregated metric data are stored in a dedicated, scalable distributed database. Much like the JBoss ON server, there can be multiple storage nodes in a cluster (installed independently of the server). Nodes can be added and dropped from the cloud easily, allowing the metrics storage to be dynamically expanded according to the needs of the environment.

24.3.1. About High-Speed Metrics Storage

As touched on in Section 16.3, “Potential Impact on Server Performance”, collecting metrics and generating alerts can be resource-intensive. It results in near-constant write operations on the backend database. That creates a natural threshold for the number of metrics that can be collected (30,000 per minute) before encountering performance degradation.

NOTE

The natural threshold of 30,000 metrics per minute is based on internal performance testing. This threshold varies depending on system resources and overall load.

JBoss ON uses two databases to store its information. One is a central relational database

(PostgreSQL or Oracle) which stores all configuration about the JBoss ON servers and agents, all resource inventory data, resource configuration, and other data. The other database is a distributed database (a cluster of storage nodes) which stores all numeric monitoring data — in other words, all collected metrics.

The metrics storage node can be installed on its own dedicated machine, which can significantly improve write performance to the database (and, therefore, improve monitoring performance):

- Dedicated CPU
- More available physical memory
- Faster disks
- More disk space

These are the same performance considerations as installing the relational database on a separate machine from the JBoss ON server. By using two databases, it is also possible to move write-intensive and resource-intensive metrics storage away from resource-intensive configuration data, such as drift snapshots and bundle configuration.

Additionally, the distributed database can be expanded, with multiple nodes in a cluster. This ability to add additional nodes according to load is a crucial management tool for administrators. Rather than encountering that hardware-driven limit of 30,000 metrics collected per minute, additional nodes can be added to improve performance.

The storage node cluster is created and managed by JBoss ON (which also minimizes the metrics storage node management overhead). A storage node is installed on a system, using the **`rhqctl install --storage`** command. The storage node always requires a companion agent.

At least one storage node must be created and managed by JBoss ON (which minimizes the metrics storage node management overhead since no external tools are required).

There are several paths of communication to handle metrics data, all working in parallel.

1. The agent sends the storage node configuration to the JBoss ON server. The JBoss ON server then sends that updated storage cluster information to every agent associated with a storage node.

   Each companion agent then updates its storage cluster configuration, in the **`rhq-storage-auth.conf`**, with the hostname or IP address of the new node. (Likewise, when a node is removed, the server sends the information to each of the companion agents, and the agent removes the hostname or IP address from the list in the local storage node's **`rhq-storage-auth.conf`** file.)

2. The server receives monitoring data from all agents (not just those associated with a storage node), and sends that information to an available storage node to be stored.

3. The storage nodes replicate their monitoring data among each other for high availability and integrity.

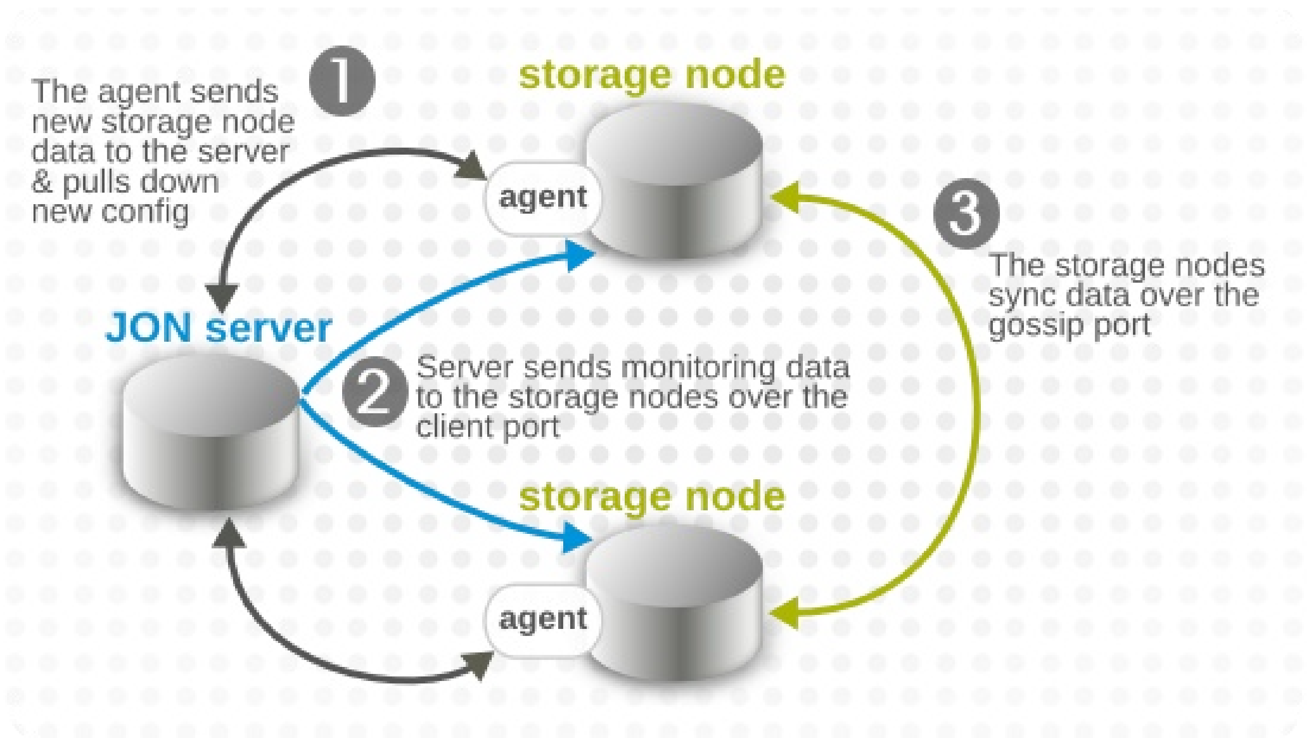

1
The agent sends new storage node data to the server & pulls down new config
storage node
agent
3
The storage nodes sync data over the gossip port
JON server
2
Server sends monitoring data to the storage nodes over the client port
storage node
agent

Figure 24.1. Server, Agent, and Metrics Storage Node Communication

Node cluster communication requires three elements:

- The hostname or IP address of every storage node, stored in the `rhq-storage-auth.conf`
- A common port number for the JBoss ON server to use to communicate with the storage node (the *client port*)
- A common port number for the other storage nodes in the cluster to use to sync data between each other (the *gossip port*)

The metrics storage provides data availability and integrity by backing up the data in two ways:

- Replicating data between the storage nodes (over the gossip port)
- Taking local snapshots and backing up the data locally

24.3.2. Deploying and Undeploying Storage Nodes

There can be multiple metrics storage databases in the JBoss ON environment. Much like the JBoss ON servers can operate in a cloud, multiple storage nodes communicate with each other to function in a cluster.

Nodes can be added and removed from the cluster by administrators or dynamically using JBoss ON scripts in response to changes in the environment.

With the default cluster configuration, a node is deployed as soon as it is installed and configured in the JBoss ON server. These are actually two separate steps.

1. The bits are installed on a local system and the storage node is registered with the JBoss ON server.
2. The new node information is deployed to the cluster.

Whether to deploy nodes automatically or manually can be determined by the provisioning and monitoring requirements of the environment.

24.3.2.1. Storage Node Requirements

There are two critical requirements for deploying nodes:

- The hostnames and IP addresses of all storage nodes and the hostname and bind addresses of the JBoss ON server and agents must all be fully-resolvable in DNS. If the IP addresses and hostnames of the storage nodes, servers, or agents are not properly formatted in DNS, then all communication between the different JBoss ON components will fail.

- The firewall must allow communication over the two ports used by the storage nodes. By default, the ports are 9142 and 7100 for the server/client and gossip ports, respectively.

24.3.2.2. Installing Additional Nodes

When creating a new storage node, two components are always installed: the storage node and a companion agent. The only required information to set up the node is the IP address of the JBoss ON server. The agent registers with the server and then sends the storage node hostname or IP address. The server then distributes that information among the other agents and it gets propagated into the cluster.

If the cluster is using custom client and gossip ports, then edit the **`rhq-storage.properties`** file with the correct ports before running the installation script.

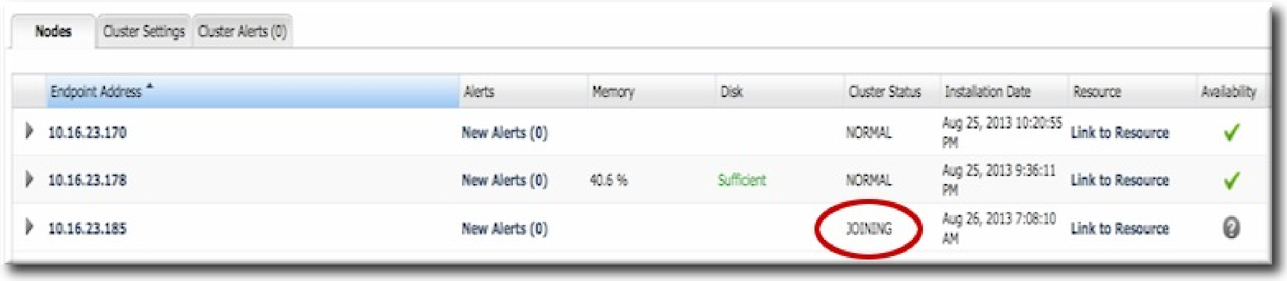
Run the installation script with the **`--agent-preference`** option to supply the server bind address. For example:

```
[root@server ~]# serverRoot/jon-server-3.2.GA1/bin/rhqctl install --
storage --agent-preference="rhq.agent.server.bind-address=0.0.0.0"
```

NOTE

When installing on Linux, the **`rhqctl`** command must be run as root. On Windows, the command prompt must be opened with the option **`Run as Administrator`**.

The deployment operation can take several minutes to complete, as the new node information is propagated among the existing nodes. Until the deploy operation is complete, the node shows a status of *Joining*.



Endpoint Address	Alerts	Memory	Disk	Cluster Status	Installation Date	Resource	Availability
10.16.23.170	New Alerts (0)			NORMAL	Aug 25, 2013 10:20:55 PM	Link to Resource	✓
10.16.23.178	New Alerts (0)	40.6 %	Sufficient	NORMAL	Aug 25, 2013 9:36:11 PM	Link to Resource	✓
10.16.23.185	New Alerts (0)			JOINING	Aug 26, 2013 7:08:10 AM	Link to Resource	?

Figure 24.2. Joining the Cluster

WARNING

Deploying a node lists that node's host in the cluster configuration and *any* allowed host can gain access to the data in the storage cluster.

Restrict access to the `rhq-storage-auth.conf` file so that the allowed hosts list cannot be altered to allow an attacker to gain access to the cluster and the stored data.

24.3.2.3. Deploying Nodes Manually

By default, when a new storage node is installed, it is automatically deployed to the cluster. However, there is a cluster setting that disables automatic deployments. This can be useful if machines are being provisioned and taken offline repeatedly (such as in virtual environments) or if nodes should only be online during certain periods.

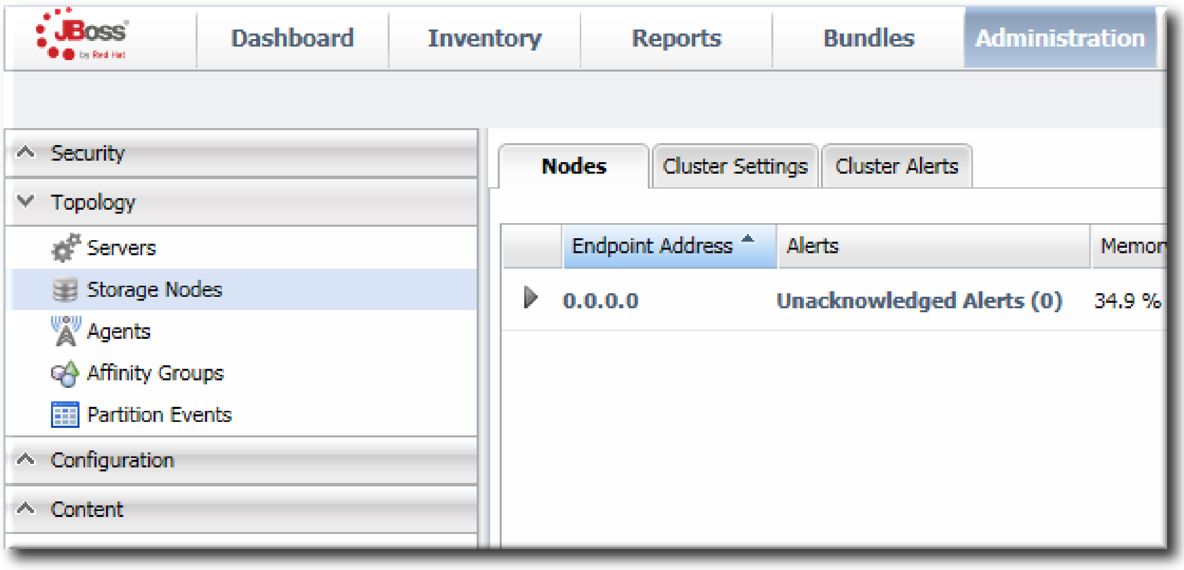
WARNING

Deploying a node lists that node's host in the cluster configuration and *any* allowed host can gain access to the data in the storage cluster.

Restrict access to the `rhq-storage-auth.conf` file so that the allowed hosts list cannot be altered to allow an attacker to gain access to the cluster and the stored data.

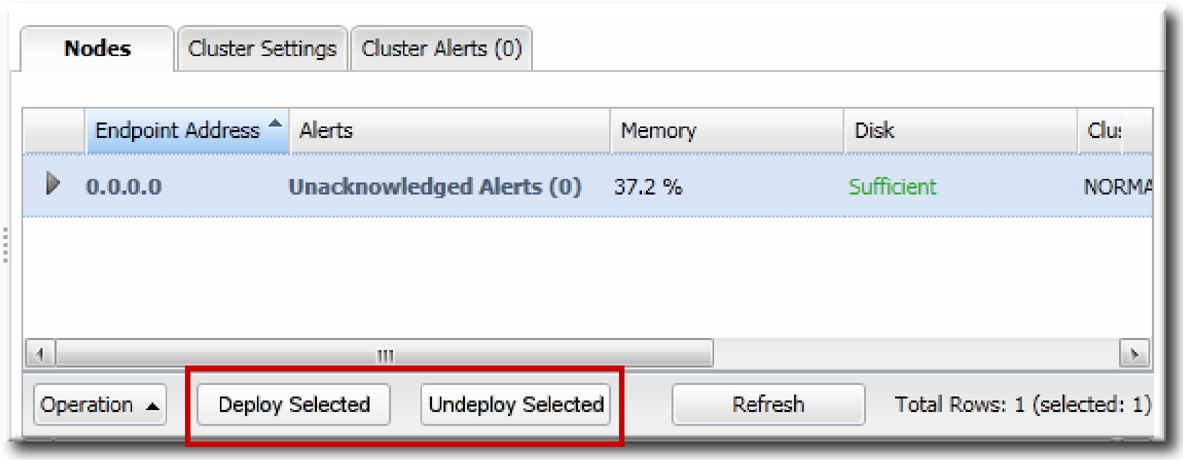
To deploy a node manually:

1. Install a node using the `rhqctl install` command.
2. Click the `Administration` tab in the top navigation bar.
3. In the `Topology` area on the left, select the `Storage Nodes` item.



Endpoint Address	Alerts	Memor
0.0.0.0	Unacknowledged Alerts (0)	34.9 %

4. In the **Nodes** tab, select the row of the node to deploy.



Endpoint Address	Alerts	Memory	Disk	Clu
0.0.0.0	Unacknowledged Alerts (0)	37.2 %	Sufficient	NORMA

Total Rows: 1 (selected: 1)

5. Click the **Deploy** button.

A node can also be deployed by running a **Deploy** operation on the storage node resource.

NOTE

Storage nodes can be deployed using the JBoss ON CLI and scripts, as well. This can be useful when provisioning new systems in the infrastructure.

For example:

```
// deploy a storage node
nodes =
StorageNodeManager.findStorageNodesByCriteria(StorageNodeCriter
ia());
node = nodes.get(0);
StorageNodeManager.deployStorageNode(node);
```

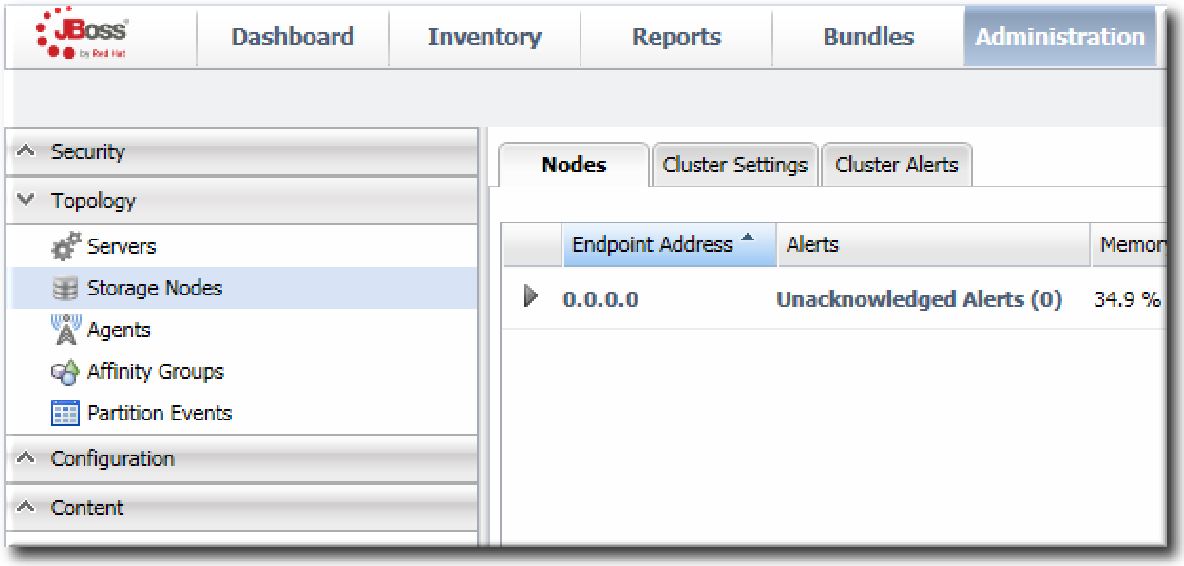
This can be key in dynamically adding storage nodes as demands on the infrastructure increase. Nodes can be installed in advance and then hot-deployed and removed as necessary.

24.3.2.4. Removing Nodes

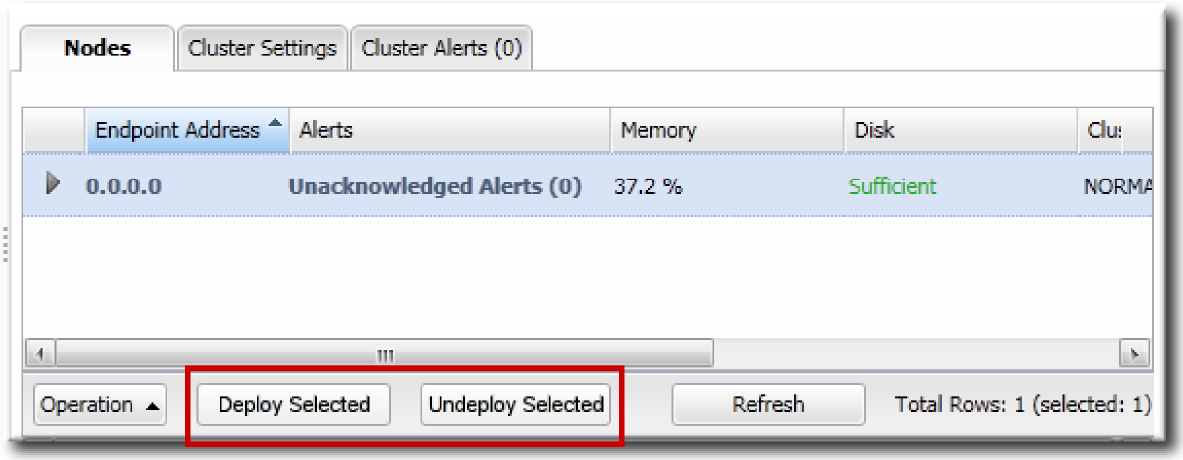
Undeploying a storage node removes it from the cluster and then removes the resource from the inventory and uninstalls the storage node bits from the machine.

1. Click the **`Administration`** tab in the top navigation bar.

2. In the **`Topology`** area on the left, select the **`Storage Nodes`** item.



3. In the **`Nodes`** tab, select the row of the node to remove. To select multiple rows, hold the **`Ctrl`** key and click the desired rows.



4. Click the **`Undeploy Selected`** button, and confirm the operation.

A node can also be removed by running an **`Undeploy`** operation on the storage node resource.

NOTE

Storage nodes can be removed using the JBoss ON CLI and scripts, as well. This can be useful when provisioning and removing systems from the infrastructure.

For example:

```
// undeploy a storage node
nodes =
StorageNodeManager.findStorageNodesByCriteria(StorageNodeCriter
ia());
node = nodes.get(0);
StorageNodeManager.undeployStorageNode(node);
```

CHAPTER 25. DEFINING ALERTS

25.1. PLANNING ALERTS

An *alert* is a configuration setting that lets an administrator know that something has happened to a resource. Conditions and notifications are configured together in an *alert definition* for a resource.

There are three major components to an alert definition:

- The information that identifies that specific alert definition — the name, priority, and whether it is active (Section 25.1.2, “Basic Procedure for Setting Alerts for a Resource”)
- The conditions that trigger the alert, which depends on the area of the resource being monitored (Section 25.2, “Alert Conditions”)
- The method and settings to use to send the alert (Section 25.3, “Alert Responses”)

25.1.1. An Alerting Strategy in Four Questions

Alerting can be very basic, an immediate notification of something going on with your IT infrastructure. Alerting can also define complex scenarios and detailed responses, allowing administrators to respond automatically and proactively to problems in resources.

Both approaches are equally valid and equally useful, depending on your environment, management processes, and plans. Putting a little effort into planning your alerts — simple and complex — can make your overall management much easier.

25.1.1.1. What's the Condition?

This is probably the single most important thing to determine. What event do you want to know about?

An alert definition can have one condition or multiple conditions, and those conditions can trigger an alert if only one is true or only if all are true. One of the easiest things to do is picture the scenario that needs to be reported and work backward to determine what condition or set of conditions best represents that scenario.

Alerts can be very general ("there was a change") or very specific. General conditions are more informational. Very specific conditions can allow very specific responses — meaning that detailed and useful responses can be automated until an administrator can review and intervene.

There is no limit on the number of alert definitions that can be configured, apart from reasonable performance constraints. It may make sense to have one alert that sends a notification for a simple metric value change, while a series of other alerts look at, say, the change for that metric in relation to a different metric, and then runs a certain JBoss ON CLI script depending on what other factors are involved.

25.1.1.2. What's the Frequency?

A condition can occur and recur, across multiple monitoring scans. How many times do you need to be informed of the same condition?

There are a lot of factors to consider: what kind of response will be taken, what kind of audit trail you need to keep, how common a condition is likely to be, and how long a condition may exist.

Some factors in the alert definition such as dampening rules and disable/recover alert settings throttle how many notifications are sent.

One thing to remember is that an alert notification is every response that the JBoss ON server takes when an alert is triggered. It can be sending an email, posting a message in the UI, running a CLI script, or initiating an operation.

For common, relatively low-priority alerts, probably a single notification is sufficient, so dampening and disable rules can be set to prevent spamming administrators with emails or flooding the recent alerts portlet.

For infrequent issues, critical failures, or high priority resources, the frequency of the alert notification depends on what actions you need to take. Informational alerts may sent for as along as the condition persists, while an alert with an aggressive response like running a CLI script may only be run once and then disabled to prevent from repeatedly running the same script.

25.1.1.3. What's the Response to Take?

Responding to an alert has two phases: the immediate response and then long-term mitigation.

When an alert for a certain condition or event on your resource is fired, what is the very first thing that should happen? Should there be an email to an administrator? Should the JBoss ON server take some action to manage the resource? JBoss ON supports SNMP traps, so should an SNMP notification be sent to an external auditing/monitoring application?

Like conditions, there is no limit to the number of notifications that can be sent when an alert occurs. Multiple responses may make sense in some cases, where JBoss ON should email an administrator and restart a resource can simultaneously.

Operations and CLI script allow administrators to automate responses, and particularly for business-critical systems, this allows JBoss ON to take an immediate action before an administrator may even be aware there is a problem.

Once an immediate action has been taken (either by JBoss ON or by an administrator), then the administrator can acknowledge the alert, essentially dismissing it. That final step can be an important procedure, a way of stating that an administrator has reviewed the alert, identified what caused it, and found a solution.

Having a way of reviewing and signing off on events — not just as a JBoss ON task, but in real life — establishes a reliable procedure for handling IT problems and improving overall maintenance strategies.

25.1.1.4. How Many Resources Does This Affect?

Alert definitions can be set for an individual resource, for a group of compatible resources, or for an entire resource type.

Being able to apply the same definition to multiple resources is invaluable because of how easy it is to manage complex definitions.

NOTE

Alert templates for a resource type automatically apply to all existing and new resources of that type.

Be cognizant, though, of how different resources of the same type need to be alerted. For example, production servers require a very high level of reporting, monitoring metrics, and, possibly, automated response. QA or development servers may only require general alert information with little scripting or advanced responses. The number, frequency, and type of metrics collected on a production system may vary from those collected on QE or development systems, and that has an impact on what conditions can be used for alert definitions.

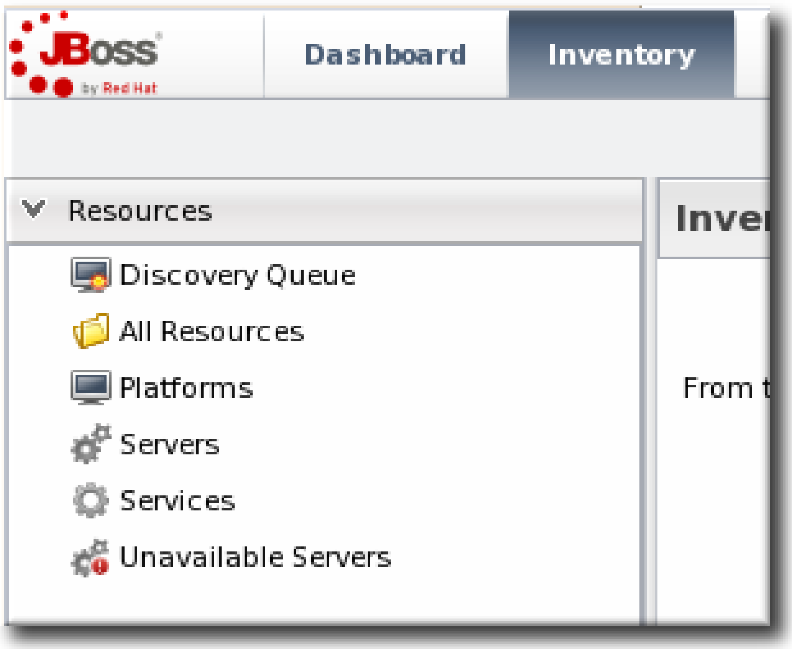
Grouping is an asset. QE, development, staging, and production resources can be separated into different groups, with the appropriate levels of monitoring and alerting set on each.

25.1.2. Basic Procedure for Setting Alerts for a Resource

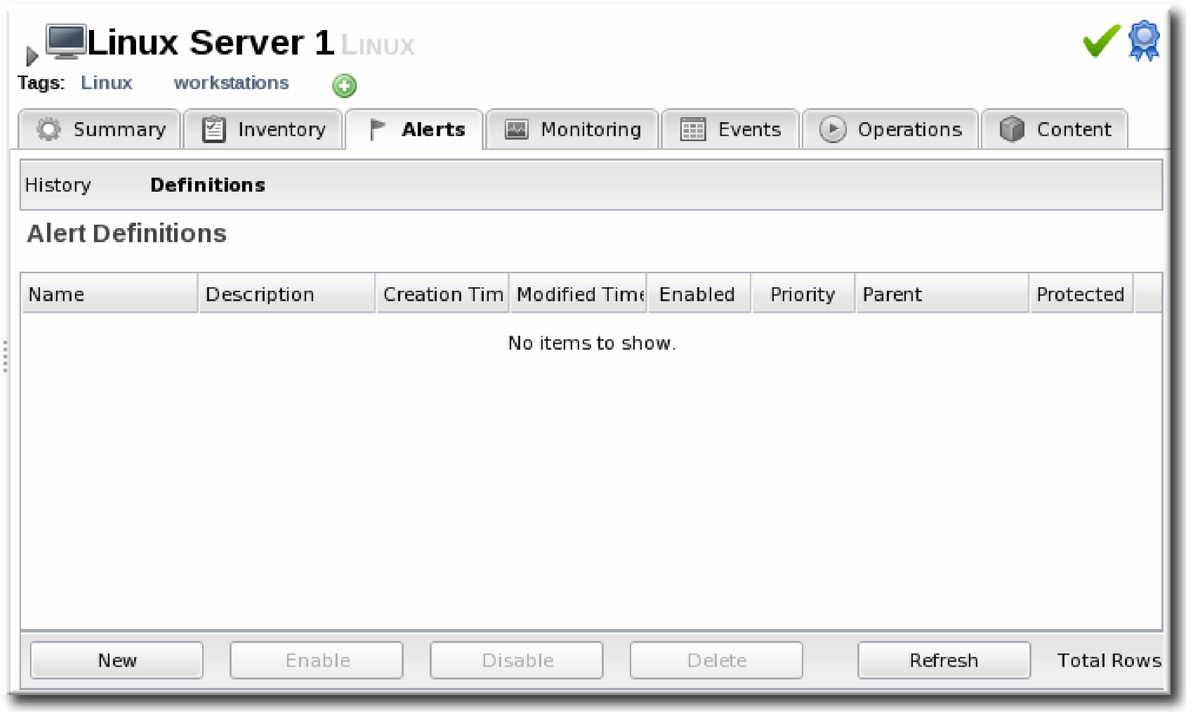
NOTE

It is not possible to edit an alert condition or an alert notification after they are set. To change the conditions or notifications for an alert definition, delete the condition or notification and create a new one.

1. Click the **`Inventory`** tab in the top menu.
2. Select the resource type in the **`Resources`** menu table on the left, and then browse or search for the resource.



3. Click the resource name in the list.
4. Click the **`Alerts`** tab for the resource.



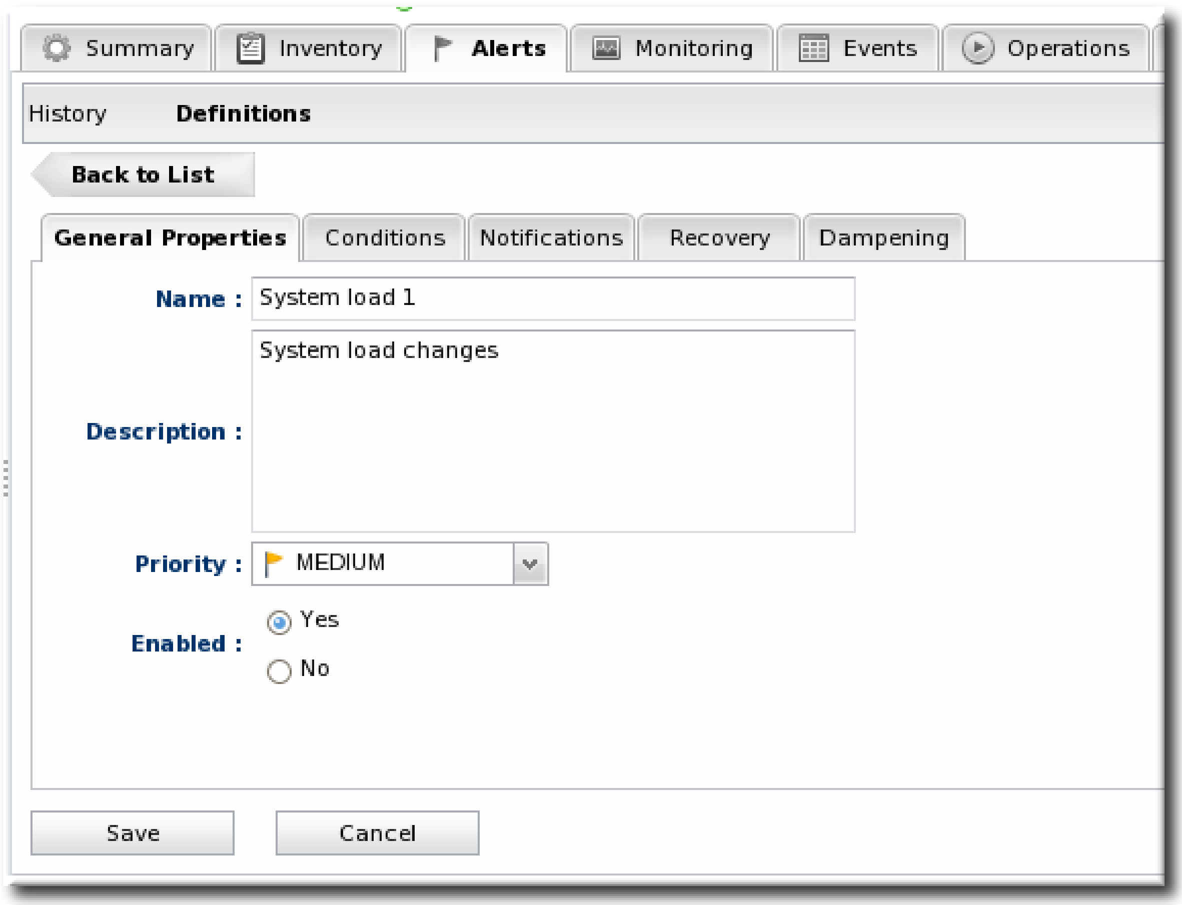
Alert Definitions

Name	Description	Creation Tim	Modified Time	Enabled	Priority	Parent	Protected

No items to show.

5. In the **`Definitions`** subtab, click the **`New`** button to create the new alert.

6. In the **`General Properties`** tab, give the basic information about the alert.



Name : System load 1

Description : System load changes

Priority : MEDIUM

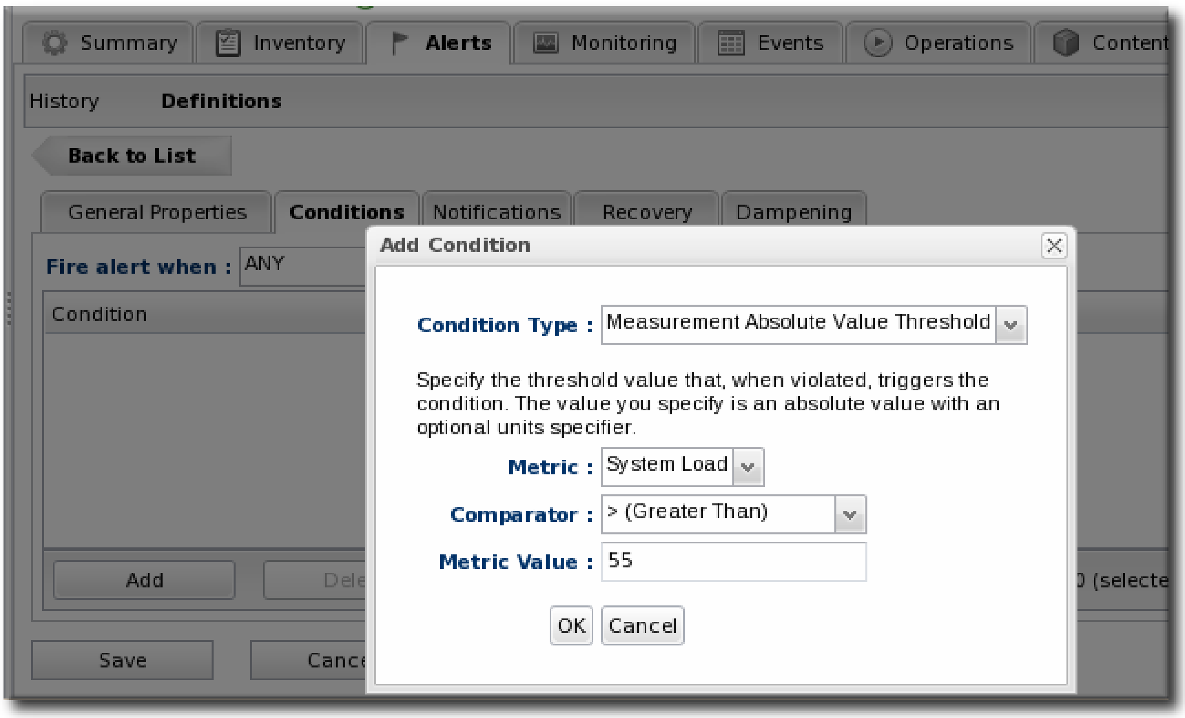
Enabled : [✓ Yes] No

- *Name*. Gives the name of the specific alert definition. This must be unique for the resource.

- *Description*. Contains an optional description of the alert; this can be very useful if you want to trigger different kinds of alert responses at different conditions for the same resource.

- *Priority.* Sets the priority or severity that is given to an alert triggered by this definition.
- *Enabled.* Sets whether the alert definition is active. Alert definitions can be disabled to prevent unnecessary or spurious alerts if there is, for instance, a network outage or routine maintenance window for the resource.

7. In the **`Conditions`** tab, set the metric or issue that triggers the alert. Click the **Add** button to bring up the conditions form.



Condition Type : Measurement Absolute Value Threshold

Specify the threshold value that, when violated, triggers the condition. The value you specify is an absolute value with an optional units specifier.

Metric : System Load

Comparator : > (Greater Than)

Metric Value : 55

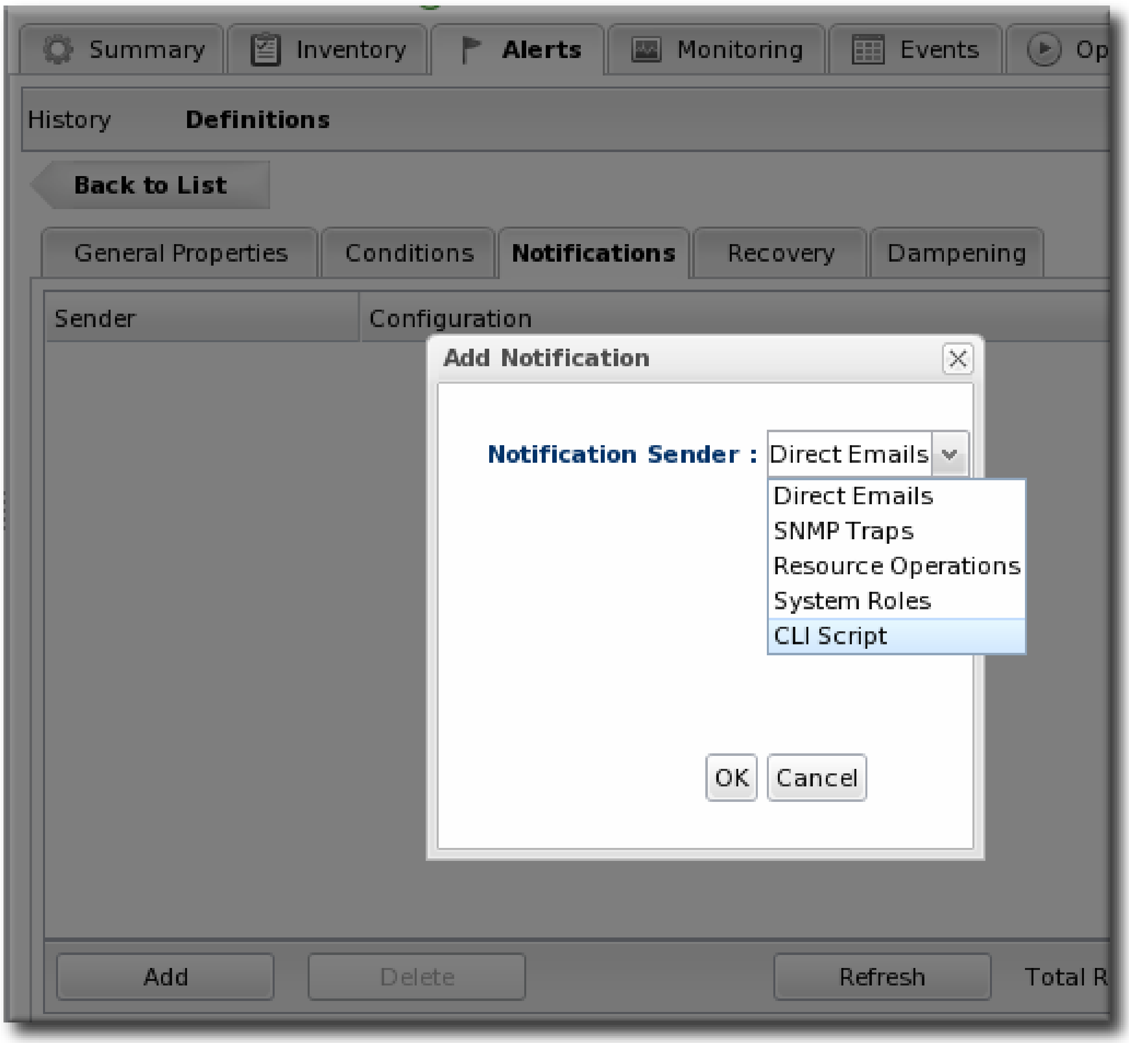
NOTE

There can be more than one condition set to trigger an alert. For example, you may only want to receive a notification for a server if its CPU goes above 80% *and* its available memory drops below 25MB. The **ALL** setting for the conditions restricts the alert notification to only when both criteria are met. Alternatively, you may want to know when either one occurs so that you can immediately change the load balancing configuration for the network. In that case, the **ANY** setting fires off a notification as soon as even one condition threshold is met.

   1. Click the **`Add a new condition`** button.
   2. From the initial drop-down menu, select the type of condition. The categories of conditions are described in Table 25.1, “Types of Alert Conditions”, and the exact conditions available to be set for every resource are listed in the *Resource Monitoring Reference*.
   3. Set the values for the condition.

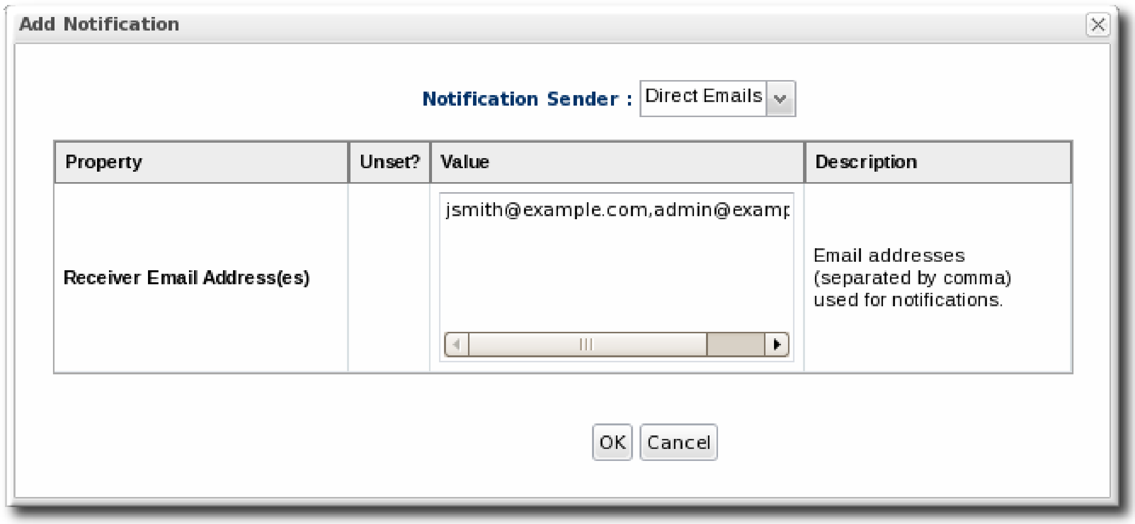
8. In the **`Notifications`** tab, click **Add** to set a notification for the alert.

   1. Select the method to use to send the alert notification in the *Sender* option.



The *Sender* option first sets the specific type of alert method (such as email or SNMP) and then opens the appropriate form to fill in the details for that specific method.

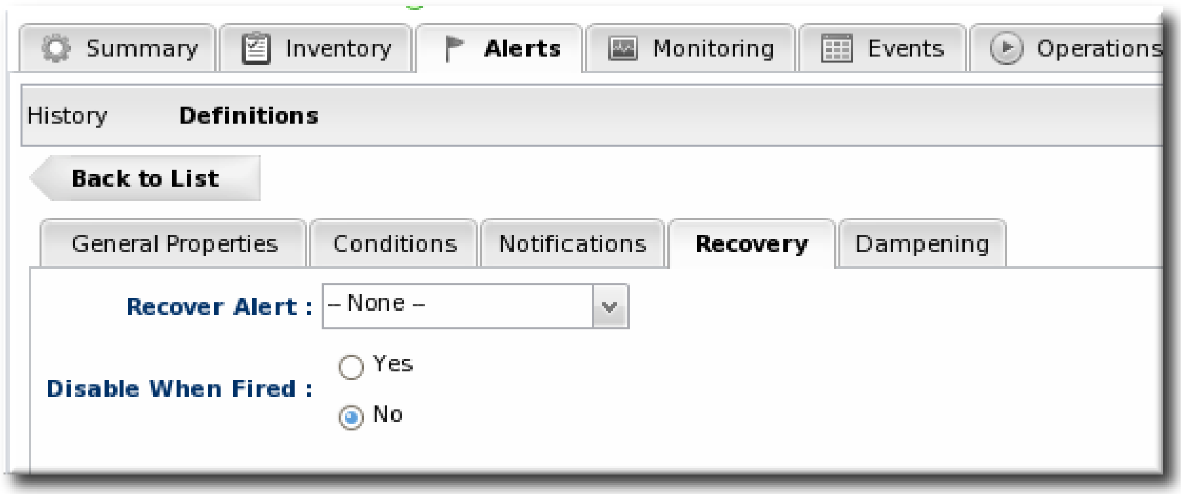
2. Fill in the required information for the alert sender method. The method may require contact information, SNMP settings, operations, or scripts, depending on what is selected.


Add Notification

Notification Sender : Direct Emails

Property	Unset?	Value	Description
Receiver Email Address(es)		jsmith@example.com,admin@examp	Email addresses (separated by comma) used for notifications.

9. In the **Recovery** tab, set whether to disable an alert until the resource state is recovered. Optionally, select another alert to enable (or recover) when this alert fires.

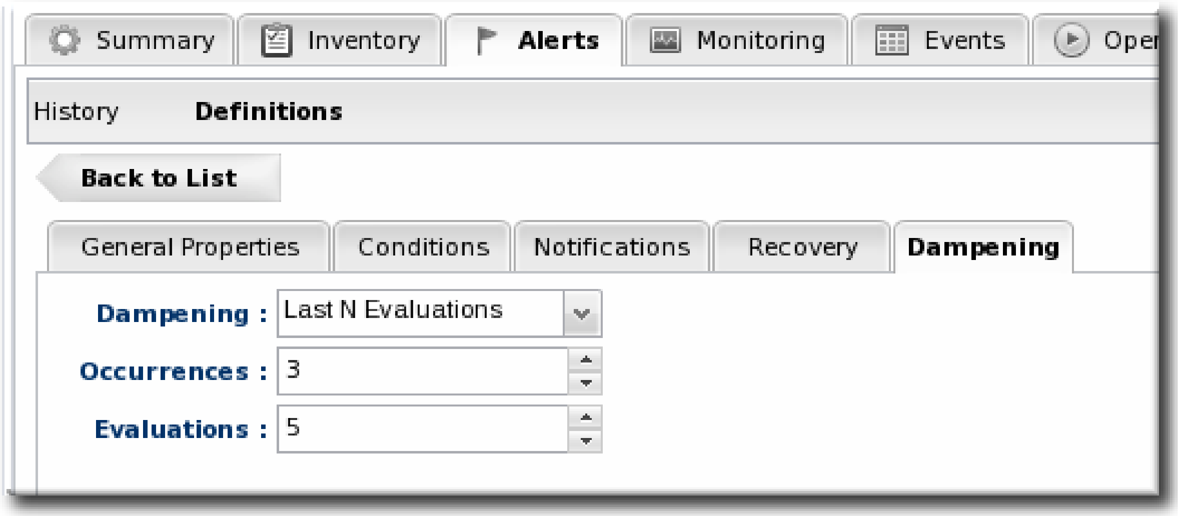


Recover Alert : – None –

Disable When Fired : Yes [✓ No]

A recover alert takes a disabled alert and re-enables it. This is used for two alerts which show changing states, like a pair of alerts to show when availability goes down and then back up.

10. In the **Dampening** tab, give the dampening (or frequency) rule on how often to send notifications for the same alert event.



Dampening : Last N Evaluations

Occurrences : 3

Evaluations : 5

The frequency for sending alerts depends on the expected behavior of the resource. There has to be a balance between sending too many alerts and sending too few. There are several frequency settings:

- *Consecutive*. Sends an alert if the condition occurs a certain number of times in a row for metric calculations. For example, if this is set to three, then the condition must be detected in three consecutive metric collection periods for the alert to be fired. If this is set to one, then it sends an alert every time the condition occurs.

- *Last N evaluations*. This sets a number of times that the condition has to occur in a given number of monitoring evaluations cycles before an alert is sent.

- *Time period*. The other two similar dampening rules set a recurrence based on the JBoss ON monitoring cycles. This sets the alerting rule based on a specific time period.

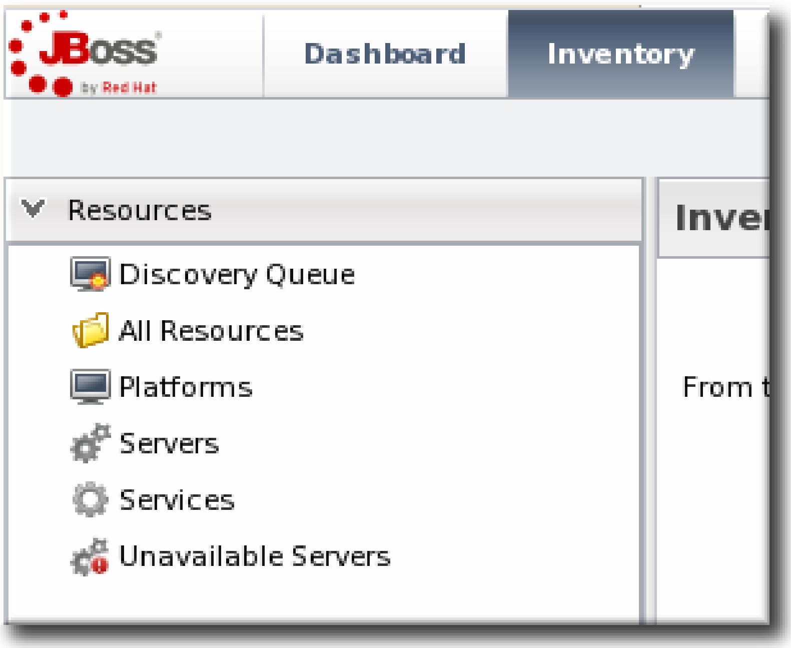
11. Click **OK** to save the alert definition.

25.1.3. Enabling and Disabling Alert Definitions

When an alert definition is disabled, no alert notifications are triggered for that set of conditions. Disabling definitions is very useful when resources are being taken offline for a know reason (such as upgrades or maintenance) and any alerts triggered during that time would be wrong. Alert definitions can be re-enabled later just as easily.

1. Click the **`Inventory`** tab in the top menu.

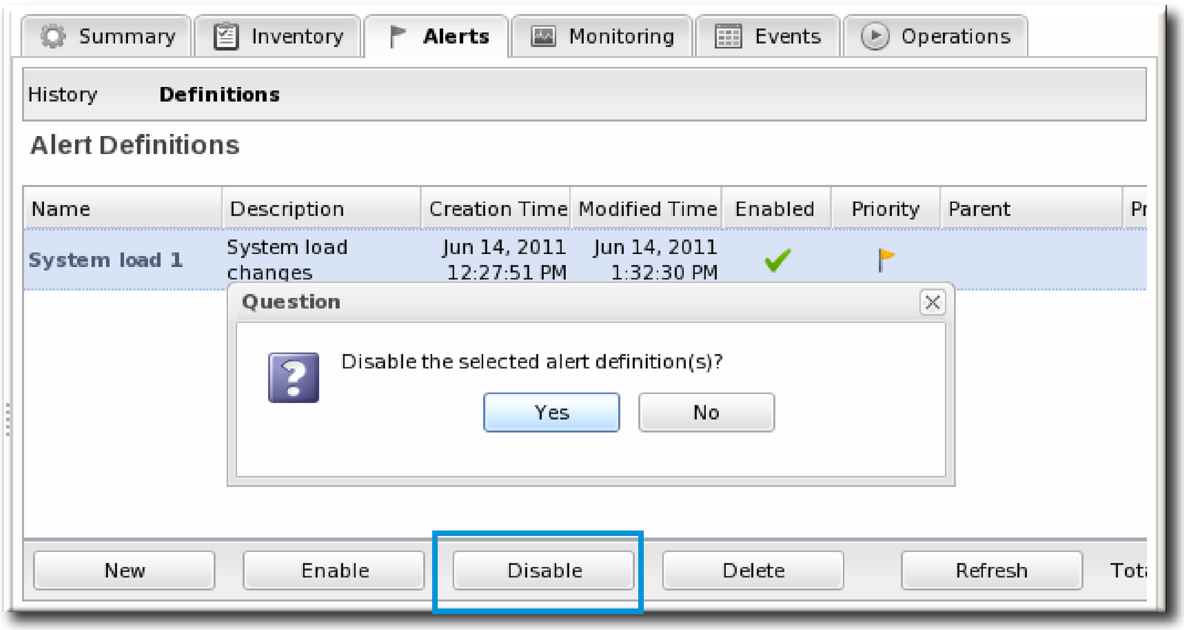
2. Select the resource type in the **`Resources`** menu table on the left, and then browse or search for the resource.



3. Click the **`Alerts`** tab.

4. In the **`Definitions`** subtab, select any of the definitions to enable or disable.

5. Click the **`Enable`** or **`Disable`** button.



Alert Definitions

Name	Description	Creation Time	Modified Time	Enabled	Priority	Parent	Pr
System load 1	System load changes	Jun 14, 2011 12:27:51 PM	Jun 14, 2011 1:32:30 PM	✓			

Disable the selected alert definition(s)?

6. Confirm the action.

25.1.4. Group Alerting and Alert Templates

Most alerts can be defined consistently for multiple resources of the same type. JBoss ON has two ways to accomplish this:

- Alert templates
- Alerts on compatible groups

An alert template is a configuration setting for the JBoss ON server. An alert is configured for a specific resource type (even if no resource of that type exists in the inventory yet). Whenever a resource is added, any alert templates in the JBoss ON configuration are automatically applied to that resource. Alert templates can be configured to allow local changes (for example, Resource A may have different baselines or expected behavior, so the alert conditions can be altered). Templates can also be strictly enforced, so that every resource of that type has exactly the same settings.

NOTE

Alert templates for a resource type automatically apply to **all** existing and new resources of that type. This can be deselected when the template is edited, so that changes only apply to new resources.

Alerts can be configured on compatible groups. As with alert templates, the compatible group's alert definitions trickle down to the rest of the group members. When a resource is added to a group, the alerts are automatically added to the resource. When the resource is removed from the group, the alert is automatically deleted. Group alerting works for both manual groups and dynamic groups. As with alert templates, group alerts can allow local changes or enforce the group alert settings.

Templates make configuration really easy to apply consistently and often, and JBoss ON allows templates to be set for alerts based on their general resource type.

Group alerts, like alert templates, apply equally to every member of a compatible group. Group alerts offer more control over which resources have the alert definition, however, since resources can be manually added to the group or selected based on a search filter. When a resource joins or leaves a group, its alert definitions are automatically updated.

25.1.4.1. Creating Alert Definition Templates

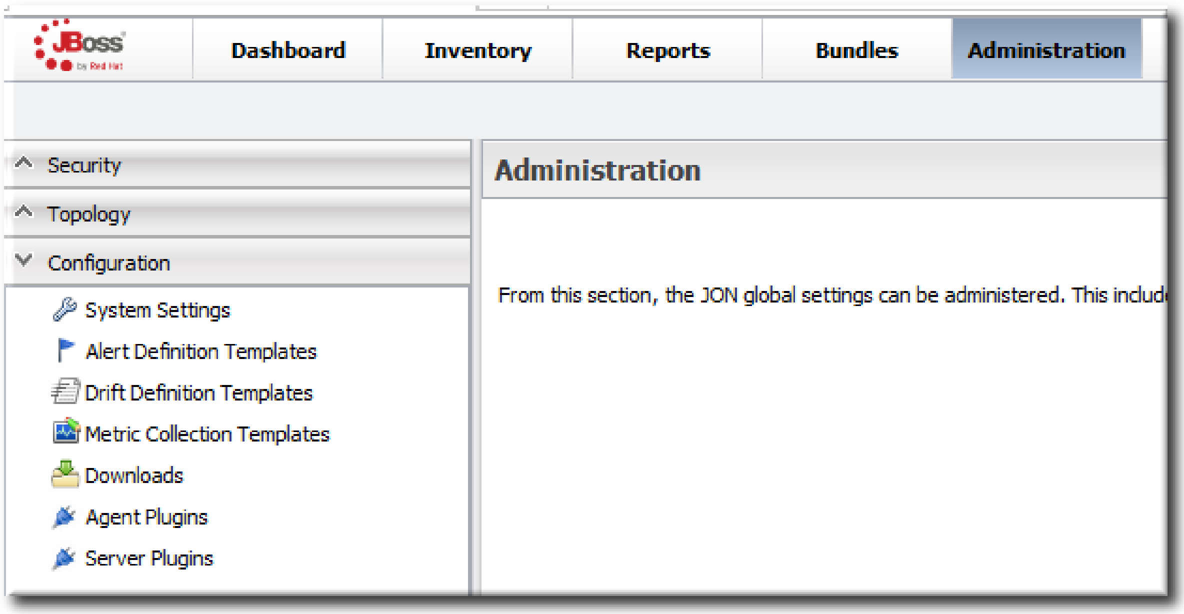
Alert templates are fully defined alert definitions — from conditions to notification methods — that are created for any of the managed resource types in JBoss ON. Servers or applications of the same type will probably have the same set of alert conditions that apply, such as free memory or CPU usage. An alert definition template creates an alert based on the *general type of resource* So, there can be alert templates for Windows, Linux, and Solaris servers, Tomcat and Apache servers, and services like sshd and cron. Every time a resource of that type is added, then the alert definition is automatically added to the resource with the predefined settings. Any alert assigned to a resource through a template can be edited locally for that resource, so these alert definitions are still flexible and customizable.

NOTE

Alert templates for a resource type automatically apply to **all** existing and new resources of that type. This can be deselected when the template is edited, so that changes only apply to new resources.

To create an alert definition template:

1. In the top navigation, open the **Administration** menu, and then the **System Configuration** menu.

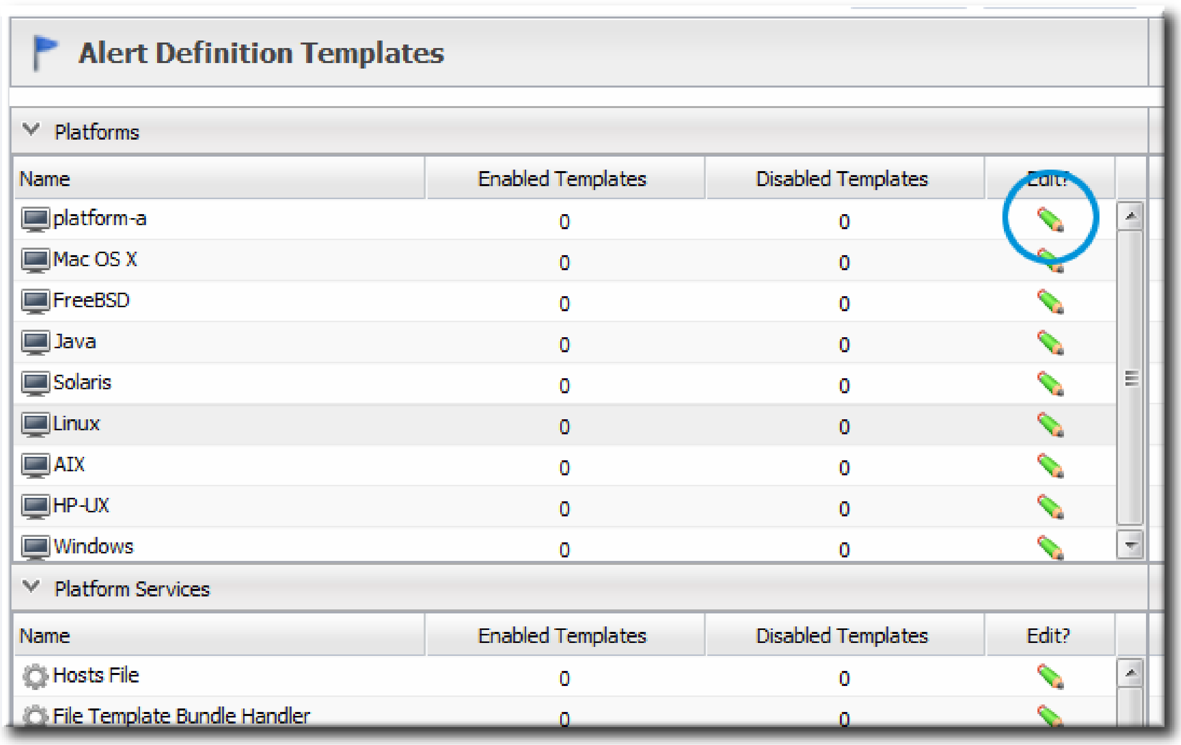


Administration

From this section, the JON global settings can be administered. This includ

2. Select the **Alert Templates** menu item. This opens a long list of resource types, both for platforms and server types.

3. Locate the type of resource for which to create the template definition.



Alert Definition Templates

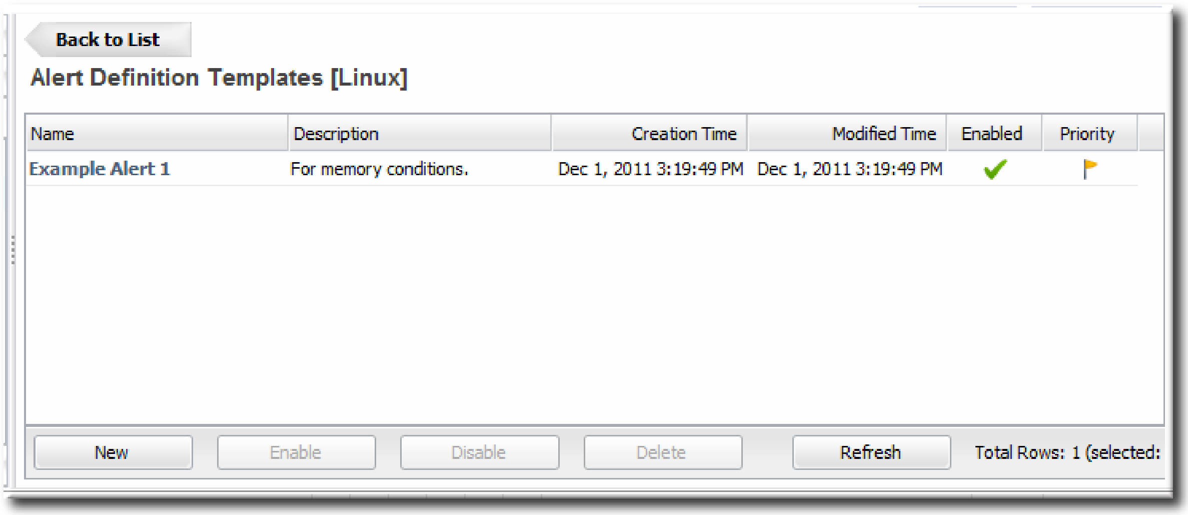
Platforms

Name	Enabled Templates	Disabled Templates	Edit?
platform-a	0	0	✎
Mac OS X	0	0	✎
FreeBSD	0	0	✎
Java	0	0	✎
Solaris	0	0	✎
Linux	0	0	✎
AIX	0	0	✎
HP-UX	0	0	✎
Windows	0	0	✎

Platform Services

Name	Enabled Templates	Disabled Templates	Edit?
Hosts File	0	0	✎
File Template Bundle Handler	0	0	✎

4. Click the **New** button to create a global alert definition. Set up the alert exactly the same way as setting an alert for a single resource (as in Section 25.1.2, “Basic Procedure for Setting Alerts for a Resource”).



Back to List

Alert Definition Templates [Linux]

Name	Description	Creation Time	Modified Time	Enabled	Priority
Example Alert 1	For memory conditions.	Dec 1, 2011 3:19:49 PM	Dec 1, 2011 3:19:49 PM	✓	⚑

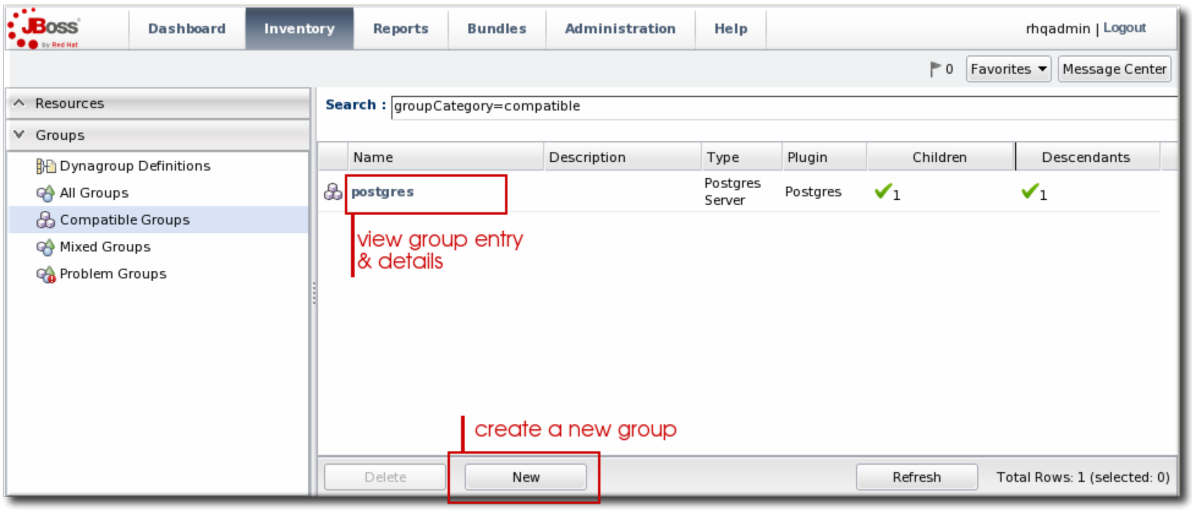
5. Save the template.

The template definition is then applied to all current and new resources of that type.

25.1.4.2. Configuring Group Alerts

Group alerts can only be set on compatible groups.

1. In the **`Inventory`** tab in the top menu, select the **`Compatible Groups`** item in the **`Groups`** menu on the left.



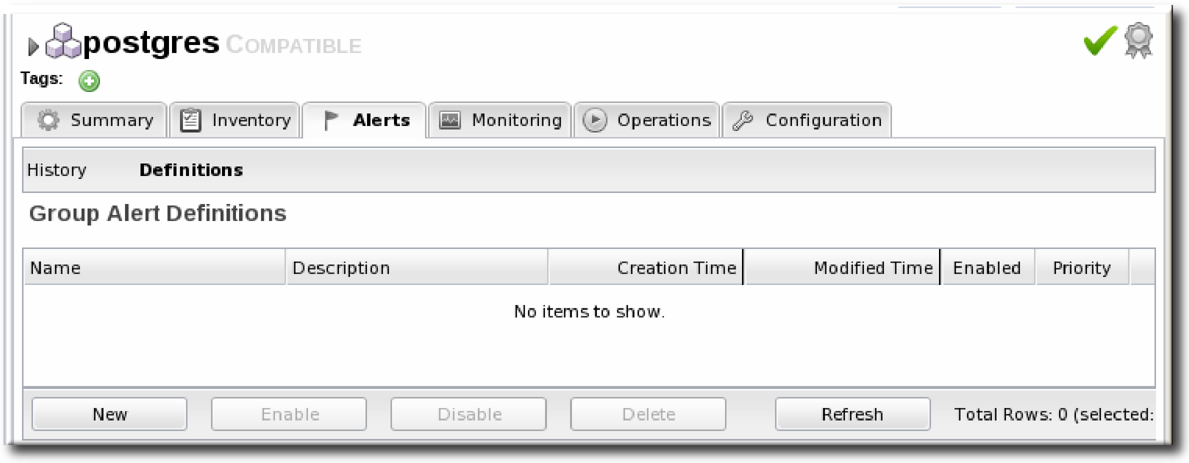
Search : groupCategory=compatible

Name	Description	Type	Plugin	Children	Descendants
postgres		Postgres Server	Postgres	✓1	✓1

2. In the main window, select the group to add the alert to.

3. Click the **`Alerts`** tab for the group.

4. In the **`Definitions`** subtab, click the **New** button.



Group Alert Definitions

Name	Description	Creation Time	Modified Time	Enabled	Priority

No items to show.

Total Rows: 0 (selected:

5. Configure the basic alert definition and notifications, as in Section 25.1.2, “Basic Procedure for Setting Alerts for a Resource”.

25.2. ALERT CONDITIONS

Monitoring (Chapter 19, *Metrics and Measurements*) is closely associated to alerting, in a larger work flow of keeping administrators aware of what is happening in their network. Alerting is based on *conditions*, the signal that an alert should be issued.

Conditions can be pretty straightforward — if A happens, do B — or they can be complex, with multiple conditions or combinations of readings required before an alert is initiated.

25.2.1. Reasons for Firing an Alert

The *condition* is any situation, event, or level on a resource that crosses a certain threshold. Basically, a condition sets parameters on what is "normal" behavior or performance for a resource. Once it crosses that boundary, JBoss ON issues an alert. This can be a metric value that has changed to an undesirable level, an event, or a recurring metric reading.

Alerts through alert definitions against are defined for individual resources or for compatible groups of resources. An alert definition specifies the conditions that trigger the alert and the type and settings of any notification that should be triggered.

When an alert is registered, the alert identifies the alert definition which was triggered (which identifies the alert condition) and the metric or event value which precipitated the alert.

An alert conditions answers four questions: *what*, *when*, *who*, and *where*. The *what* is the threshold or *condition* that triggers the alert (such as the free memory drops below a certain point). The *when* sets the frequency or timing for sending an alert using a defined *dampening* rule. And the *who* and *where* controls how administrators are *notified* of the alert.

A single condition can be enough to issue an alert, or an alert definition can require that an alert is issued only if multiple conditions are met simultaneously. This provides very granular control over when an alert is issued, which makes alerting information more valuable and relevant.

A condition can be based on any detectable monitoring or system metric, listed in Table 25.1, “Types of Alert Conditions”. These alert conditions correspond directly to the monitoring metrics available for that type of resource. All of the possible metrics for each resource type are listed in the *Resource Monitoring Reference*.

Table 25.1. Types of Alert Conditions

Condition Type	Description
Metric	A specific monitoring area that is checked and the thresholds for that area which trigger a response. Metrics are usually numeric responses of some sort (e.g., percent CPU usage, number of requests, or a cache hit ratio).
Trait	A change in a value for a specific setting. Traits are usually string values.
Availability	A sudden change in whether the resource is available or unavailable.
Operation	A specific action or task that is performed on the resource.
Events	A certain type of error message, matching a given string, is recorded. Events are filtered from system or application log files, and the types of events recognized in JBoss ON depend on the event configuration for the resource.
Drift	A resource has changed from a predefined configuration.

25.2.2. *Detailed Discussion*: Ranges, AND, and OR Operators with Conditions

Alerting is based on monitoring information. It is an extension that allows an administrator to receive a notification or define an action to take if a certain event or metrics value occurs.

The monitoring point that triggers an alert is the alert condition. At its most simplistic, an alert condition is a single event or reading. If X occurs, then that triggers an alert.

In real life, X may not be enough to warrant an alert or to adequately describe the state of a resource. Different conditions may require the same response or a situation may only be critical if multiple conditions are true. Alerting is very flexible because it allows multiple conditions to be defined with established relationships between those conditions.

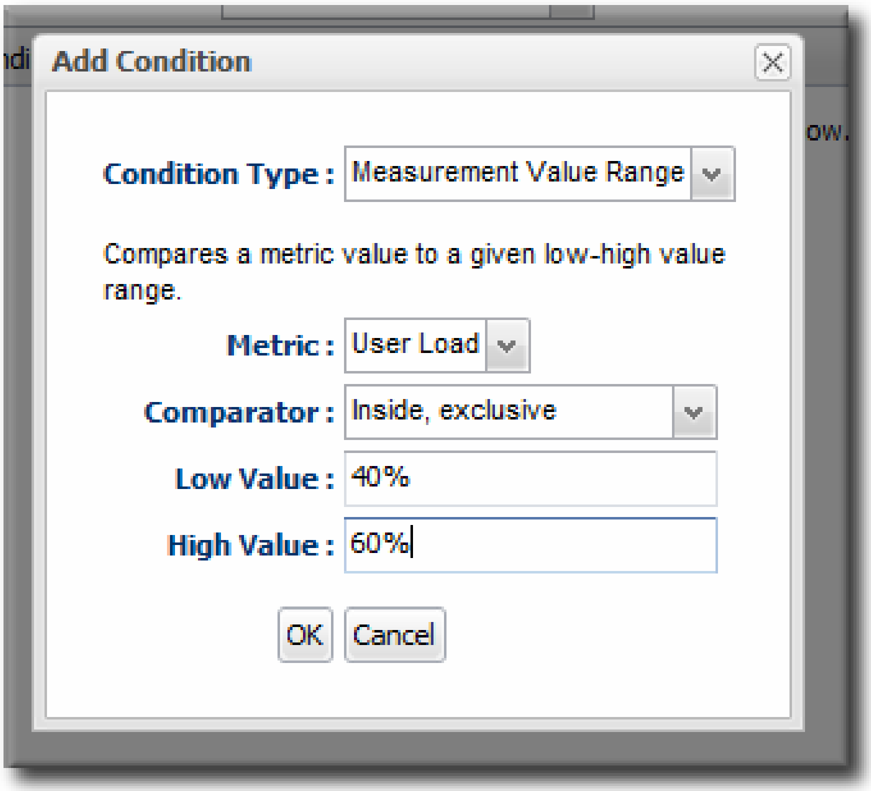
The next level of complexity is to send an alert if either X *or* Y is true. In the alert definition, this is the *ANY* option, which is a logical OR. The alert definition checks for any of those conditions, but those conditions are still unrelated to each other.

The last level of complexity is when the conditions have to relate to each other for an alert to be issued. This is the *ALL* option, which is a logical AND. Both X and Y must occur for the alert to be issued. In this case, when one condition occurs, the server puts a lock on that definition and begins waiting for the second condition to occur. When the second condition occurs, then the alert is issued.

An AND operator is very effective on different metrics, but because the conditions do not have to occur simultaneously, using a simple AND operator does not make sense for the *same* metric.

For example, Tim the IT Guy only wants an alert to be issued when the user load is between 40% to 60%, indicating slightly increased loads on his platform. Attempting to use an AND operator returns strange values when the load spikes over 70% (which trips the above 40% condition) and then falls back to 15% (which triggers the below 60% condition).

In this case, Tim uses a *range* condition. A range requires two values from the same metric that are within the given boundaries. A range can be inside values (40-60%) or it can be an outside range (below 40% and above 60%).


Add Condition

Condition Type : Measurement Value Range

Compares a metric value to a given low-high value range.

Metric : User Load

Comparator : Inside, exclusive

Low Value : 40%

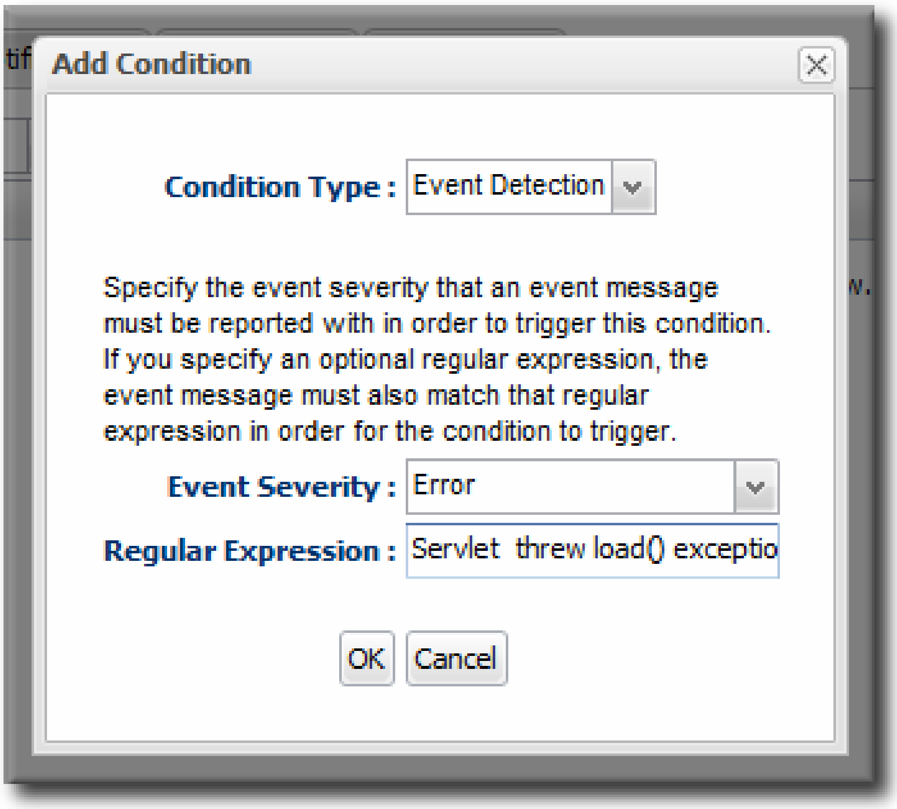
High Value : 60%

Figure 25.1. Alert Condition Range

25.2.3. *Detailed Discussion*: Conditions Based on Log File Messages

Events (Chapter 20, *Events*) are filtered log messages. Certain resource in JBoss ON maintain their own error logs, like platforms and JBoss EAP servers. JBoss ON can scan these error logs to detect events of certain severity or events matching certain patterns. This allows JBoss ON administrators to provide an easy way to identify and view important error messages.

Because JBoss ON can detect log events, JBoss ON can alert on log events. An event-based condition requires the severity of the log file message and, optionally, a pattern to use to match specific messages.



Condition Type : Event Detection

Specify the event severity that an event message must be reported with in order to trigger this condition. If you specify an optional regular expression, the event message must also match that regular expression in order for the condition to trigger.

Event Severity : Error

Regular Expression : Servlet threw load() exceptio

Figure 25.2. Log File Conditions

Setting only a severity alerts on any event with that severity. That is useful for severe or fatal errors, which are relatively infrequent and need immediate attention.

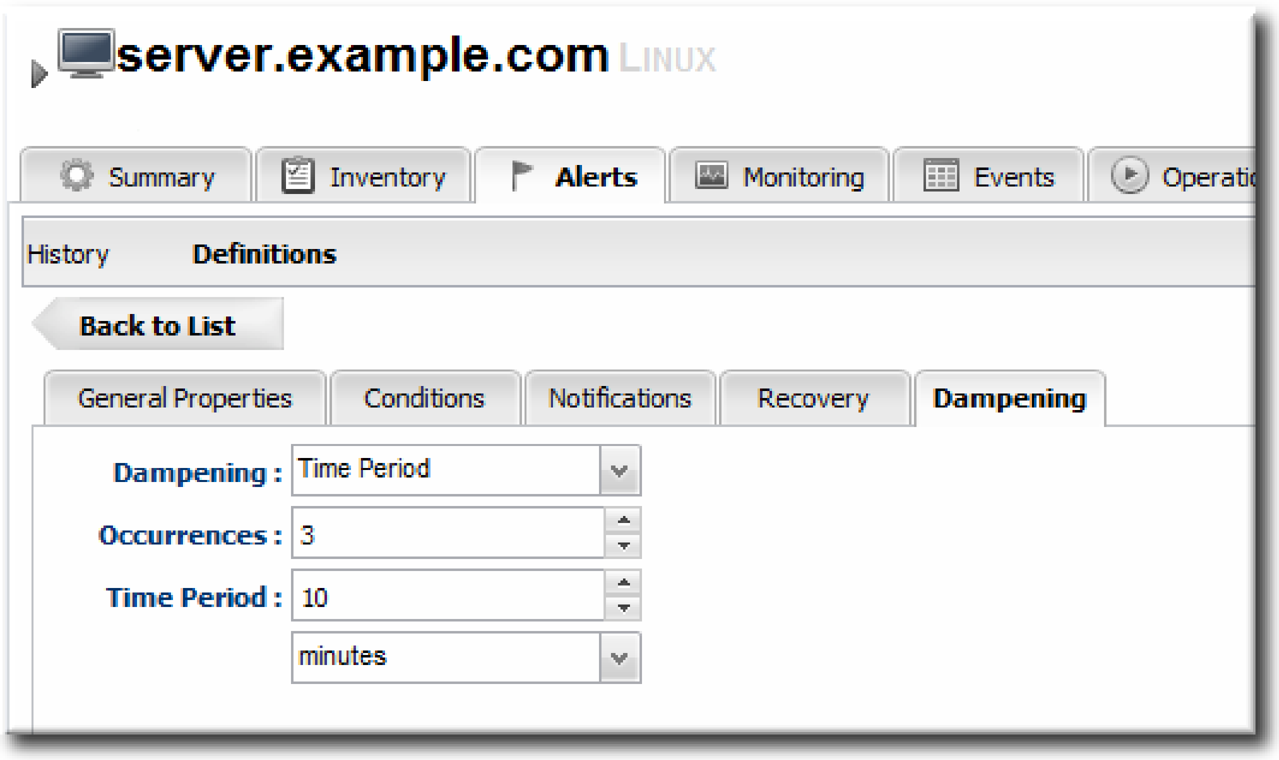
NOTE

For general error message, use a pattern to filter for a specific error type. Then, use a resource operation or CLI script to take a specific action to address that specific error, like restarting a resource or starting a new web app.

25.2.4. *Detailed Discussion:* Dampening

Dampening does not define an alert condition, but it tells the JBoss ON server how to handle recurring conditions.

An alert can be issued every single time a condition is met, or an alert can be issued and then disabled until an administrator acknowledges it. *Dampening* a condition is useful to prevent multiple alerts and notifications from being sent for a single ongoing set of circumstances.



Dampening : Time Period
Occurrences : 3
Time Period : 10
minutes

Figure 25.3. Dampening Filter

For example, Tim the IT Guy sets an alert to let him know if his platform CPU usage spikes above 80%. Every time the monitoring scan runs, the metric is again determined to be true, and is treated as a new and discrete instance. If the metric schedule is set for 10 minutes and it takes him an hour to respond to the alert, he could have six or seven alert notifications before he has a chance to respond. If the only alert response is an email, this is an annoyance, but not a problem.

If, however, Tim the IT Guy has an alert response to create a new JBoss server if his EAP connection count goes above a certain point, the same response could be taken six or seven times, when it actually should be done once and then the condition will naturally resolve itself.

Dampening is another set of instructions on how to evaluate the condition before triggering another alert. It tells JBoss ON how to interpret those monitoring data.

- JBoss ON could send an alert every time the condition is encountered. In that case, there would be multiple alerts issued if the CPU percentage bounced around, while only one alert would be sent if it hit it briefly or hit it and stayed there.

- JBoss ON could send an alert only if the condition was encountered a certain number of times consecutively or X number of times out of Y number of polls. In this case, only a recurring or sustained problem would trigger an alert. A momentary spike or trough wouldn't be enough to fire a notification.

  A condition may need to occur several times over a short period of time for it to be a problem, but once is not a problem. For example, a server may bounce between 78% and 80% CPU over several minutes, it could hit 80% once for only a few seconds, or it could hit 80% and stay there. The condition may only be relevant if the CPU hits 80% and stays, and the other readings can be ignored.

- A notification is sent only if the problem occurs within a set time period. This can be useful to track the frequency of recurring problems or to track how long a condition persisted.

NOTE

Dampening is only relevant for metrics which are variable and compared to some kind of baseline. Monitoring metrics with thresholds and value changes are dynamic, so each reading really is a new condition, even if it matches the previous reading. Dampening, then, controls those multiple similar readings.

Conditions which relate directly to a status change do not compare themselves against a baseline — they only compare against a previous state. For example, if the system configuration changes from the drift template or if the availability changes, that is a one-time change. After that, the resource has a new status and future changes would be compared against the new status — so, in a sense, it is a different condition.

Dampening does not apply conceptually for drift and availability changes.

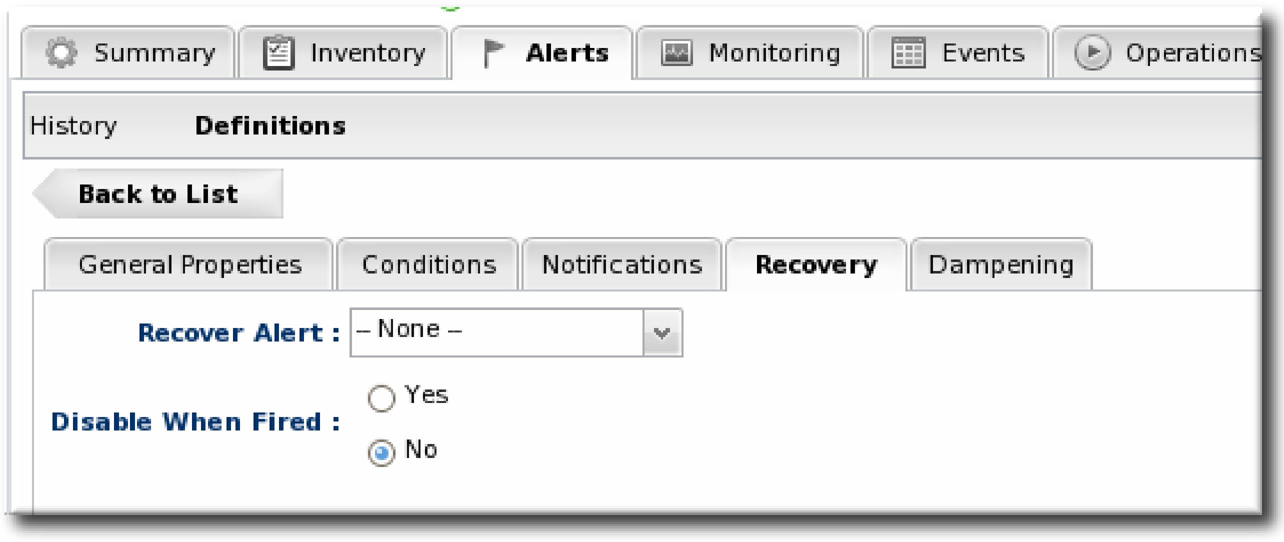
25.2.5. *Detailed Discussion:* Automatically Disabling and Recovering Alerts

There are a couple of different ways to limit how often an alert notification is sent for the same observed condition. One method is a dampening rule. An alternative is to disable an alert the first time it is fired, and then only re-enabling the alert when an administrator does it manually *or* if the condition resets itself. This second option — disabling and then resetting itself — recovers the alert.

A *recover alert* is actually a pair of alerts which work in tandem to disable and enable relevant alerts as conditions change.

A couple of workflows are common with recover alerts:

- A pair of alerts work as mutual toggle switches. When one alert is active, the other is disabled. When Alert A is fired, it can be set to recover a specified Alert B — so Alert B essentially takes its place.
- Alerts work as a kind of cascade. If Alert A is fired, that enables Alert B, which then enables Alert C. In some situations, any one given condition may not be a problem, but it becomes a problem if they occur sequentially in a short amount of time.



Recover Alert : – None –

Disable When Fired : Yes [✓ No]

Figure 25.4. Disable and Recover Alerts

For example, Alert A triggers an alert when a resource availability goes down. When Alert A fires, it disables itself and recovers (or enables) Alert B. Alert B fires an alert when the resource's availability goes up. When Alert B fires, it likewise disables itself and recovers Alert A.

Recover alerts inform an administrator first of when an issue occurs and then second when it is resolved. In the availability examples, the first alert lets the administrator know that a resource is offline, while the second alert lets the administrator know that the resource is back online.

The Setup: Toggle Recover Alerts for Availability

Tim the IT Guy has several servers that he uses for email routing and other business operations, and then he has a couple of machines that he holds in reserve as backups.

He has **`mail-server-a.example.com`** has his primary mail server, and he only wants to bring **`mail-server-b.example.com`** online if **`mail-server-a`** goes offline, and then he wants it to go back in reserve when **`mail-server-a`** comes back.

The Plan

Tim creates a set of alert definitions to help handle the transition between his mail servers.

- The first alert definition fires when the **`mail-server-a`** platform changes availability state to **goes down**.

  The notification does a couple of things:

  - Deploy a bundle with the latest mail server configuration to another platform, **`mail-server-b`**.
  - Execute a command-line script on **`mail-server-b`** to start the mail service.
  - Email Tim the IT Guy to let him know that **`mail-server-a`** is unavailable.

  For recovery, the alert does two things:

  - Disable the current alert. It only needs to fire once, to get the backup server online.
  - Recover (or enable) Alert B, so that JBoss ON waits for **`mail-server-a`** to come back up.

- The second alert definition, Alert B, is only in effect *while mail-server-a* is offline. This alert fires as soon as **`mail-server-b`** changes availability state to **goes up**.

  - This alert definition basically waits around as long as **`mail-server-a`** is down. When **`mail-server-a`** is back online, Alert B's notification is to execute a command-line script on **`mail-server-b`** to stop the mail service.
  - Alert B also sends a notification email to Tim the IT Guy to let him know that **`mail-server-a`** is available again.

  For recovery, the alert does two things:

  - Disable the current alert. Like with Alert A, Alert B only needs to fire once, to shut off the backup as soon as the primary server is back.
  - Recover (or enable) Alert A, so the JBoss ON waits again for **`mail-server-a`** to go down.

25.3. ALERT RESPONSES

Alert conditions define *when* an alert is supposed to be triggered. An alert notification defines *how* that alert notification is communicated. An alert notification can be a way of informing administrators and users of the condition, but it can also be a way of having JBoss ON itself address an issue.

A single alert can have multiple responses, to make managing potential problems easier.

IMPORTANT

Alert notifications cannot be edited after they are added to an alert definition. To change an alert notification, delete the original notification and create a new one with the desired settings.

25.3.1. Notifying Administrators and Responding to Alerts

Every alert is recorded and viewable in the JBoss ON GUI. Alerts have an optional configuration, though, of sending an external notification whenever the alert is issued.

Once an incident occurs, there has to be a way to let a systems administrator know what is going on, so they can respond to an issue. This is done by configuring a *notification*. Alert notifications fall into two categories: a way of communicating the alert and an action to take in response to an alert.

There are three methods, by default, of communicating that an alert has been fired:

- Email, to one or multiple addresses
- SNMP traps
- Messages to JBoss ON users

There are also a few default methods of having JBoss ON respond to an alert:

- Running a resource operation (on the alerting resource or any other resource in inventory)
- Running a resource script (specific type of resource operation)
- JBoss ON CLI scripts

NOTE

It is also possible to write custom alert methods, which are implemented as server-side plug-ins. Creating custom plug-ins is described in the *JBoss Operations Network Plug-ins Writing Guide*.

Alert notifications can be *clustered*. That just means that the same alert can be broadcast through several different methods at the same time. For example, if a public website goes down, then a company may want notifications to be sent to their head web administrator and to have the web server restarted at the same time.

NOTE

A single alert can initiate multiple notifications. Alert notifications are run in the order they are listed in the alert definition.

25.3.2. *Detailed Discussion:* Initiating an Operation

A parallel response to an alert is to launch an *operation*. Resource operations (which, like metrics, are defined in the resource type agent plug-in) are launched, like a notification, in response to a triggered alert. Alert operations can be run on the resource that issued the alert or on any other resource in the inventory, which allows immediate and automatic responses to alert conditions. For instance, a JBoss server may begin performing badly because its JVM is out of memory. The JVM is the resource which issues its alert, but the response by the agent is to restart the JBoss server.

When a certain alert condition occurs, the JBoss ON agent can respond by initiating an operation on a resource. This is part of the alert definition configuration, but it's worth calling out because it is such a useful tool for managing responses to alerts. Whenever an alert is fired, the agent can perform some kind of action, like restarting a server. This can be done either on the resource which issued the alert or on another resource.

Remote operations can be exceedingly useful (and versatile). For example, a JBoss server may begin performing badly because its JVM is out of memory. The JVM is the resource which issues its alert, but the response by the agent is to restart the JBoss server.

Regular operations are either initiated immediately or run on defined schedules for a specific configured resource. Alert operations are even more flexible than regular operations for two reasons:

- Alert operations are fired responsively to address any alert or event.
- Alert operations can be initiated on *any* resource in the JBoss ON inventory, not only the resource which sent the alert. That means that an operation can be run for a different application on the same host server or even on an entirely different server.

The operations performed in response to an alert are the same as the operations which can be scheduled to run on a resource. The operations available for an alert depend on the target resource on which the operation will run — not the resource where the alert is set.

Alert operations senders can be used to run scripts on remote resources. For example, if a resource goes down, a diagnostic script can be run on its parent platform or another resource can be brought online and properly configured to take its place.

25.3.2.1. Using Tokens with Alert Operations

Alert operations can use tokens to either send information or supply information about the event. For example, tokens can be used to supply resource information in a command-line script.

Alert operations can accept tokens to fill in certain values automatically. These tokens have the following form:

```
<%space.param_name%>
```

The *space* gives the JBoss ON configuration area where the value is derived; this will commonly be either **`alert`** or **`resource`**. The *param_name* gives the entry value that is being supplied. For example, to point to the URL of the specific fired alert, the token would be **`<%alert.url%>`**, while to pull in the resource name, the token would be **`<%resource.name%>`**.

JBoss ON has pre-defined token values that relate to the fired alert, the resource which issued the alert, the resource which is the target of the operation, and the operation that was initiated. These are listed in Table 25.2, “Available Alert Operation Tokens” . All of these potential token values are Java properties the belong to the operation's parent JBoss ON server.

The alert operations plug-in resolves the token value itself when the alert operation is processed to find the value. The realized value is sent to the script service, which ultimately plugs the value into the command-line argument or script which referenced the token.

Table 25.2. Available Alert Operation Tokens

Information about ...	Token	Description
Fired Alert	alert.willBeDisabled	Will the alert definition be disabled after firing?
Fired Alert	alert.id	The id of this particular alert
Fired Alert	alert.url	Url to the alert details page
Fired Alert	alert.name	Name from the defining alert definition
Fired Alert	alert.priority	Priority of this alert
Fired Alert	alert.description	Description of this alert
Fired Alert	alert.firedAt	Time the alert fired
Fired Alert	alert.conditions	A text representation of the conditions that led to this alert
Alerting Resource	resource.id	ID of the resource
Alerting Resource	resource.platformType	Type of the platform the resource is on
Alerting Resource	resource.platformName	Name of the platform the resource is on
Alerting Resource	resource.typeName	Resource type name
Alerting Resource	resource.name	Name of the resource
Alerting Resource	resource.platformId	ID of the platform the resource is on
Alerting Resource	resource.parentName	Name of the parent resource
Alerting Resource	resource.parentId	ID of the parent resource
Alerting Resource	resource.typeId	Resource type id
Target Resource	targetResource.parentId	ID of the target's parent resource
Target Resource	targetResource.platformName	Name of the platform the target resource is on

Information about ...	Token	Description
Target Resource	targetResource.platformId	ID of the platform the target resource is on
Target Resource	targetResource.parentName	Name of the target's parent resource
Target Resource	targetResource.typeId	Resource type of the target resource id
Target Resource	targetResource.platformType	Type of the platform the target resource is on
Target Resource	targetResource.name	Name of the target resource
Target Resource	targetResource.id	ID of the target resource
Target Resource	targetResource.typeName	Resource type name of the target resource
Operation	operation.id	ID of the operation fired
Operation	operation.name	Name of the operation fired

25.3.2.2. Setting Alert Operations

NOTE

A single alert can initiate multiple operations. All alert operations, as with all alert notifications, are run in the order they are listed in the alert definition.

The **Notifications** tab has a **Resource Operations** method.

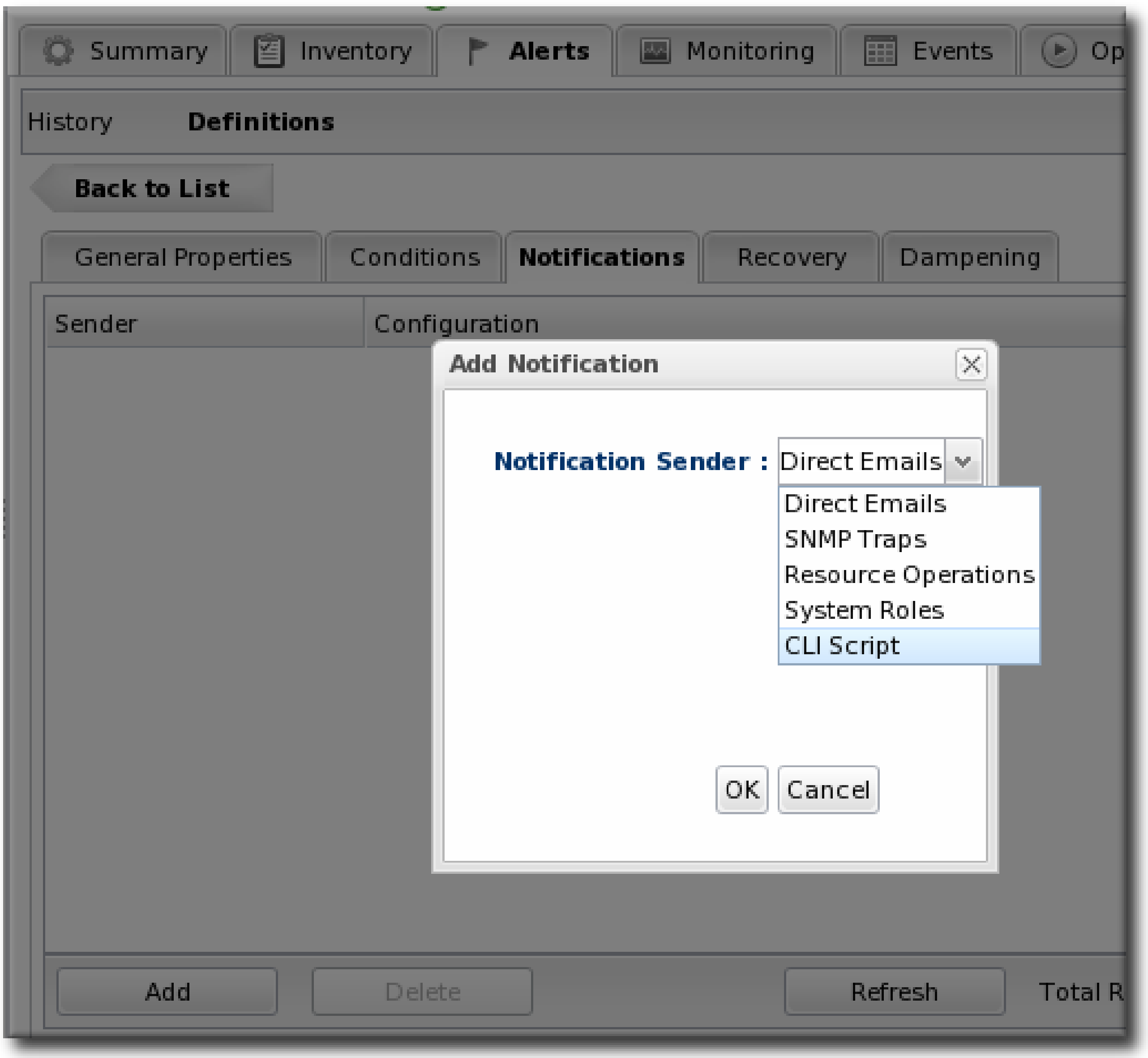


Figure 25.5. Senders

There are two parts to the operation. First is selecting which resource the operation will run on. The resource type determines what operations are available.

The default is the resource that the alert is set for; it is also possible to set it on another specific resource or on the results of a search.

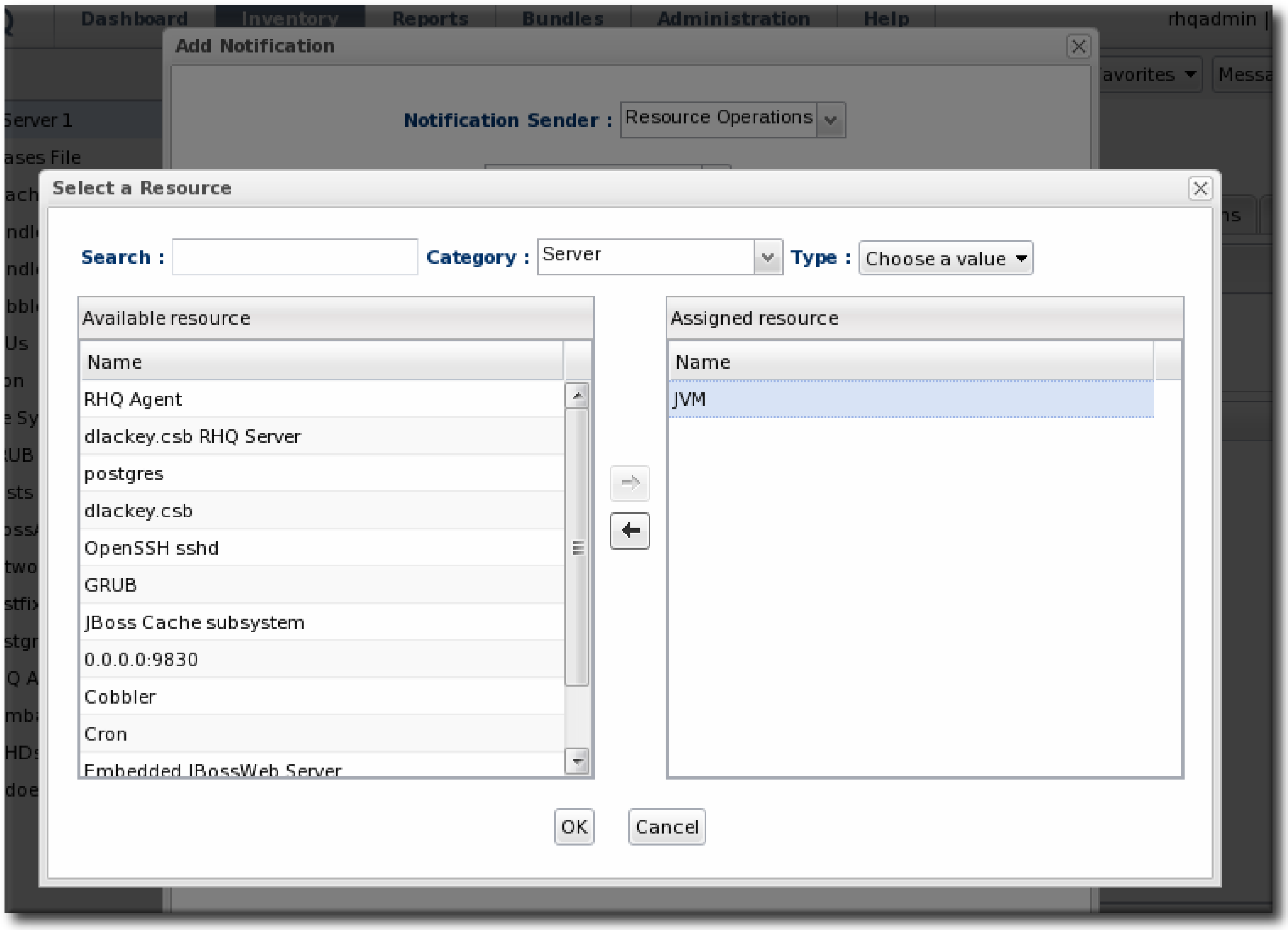


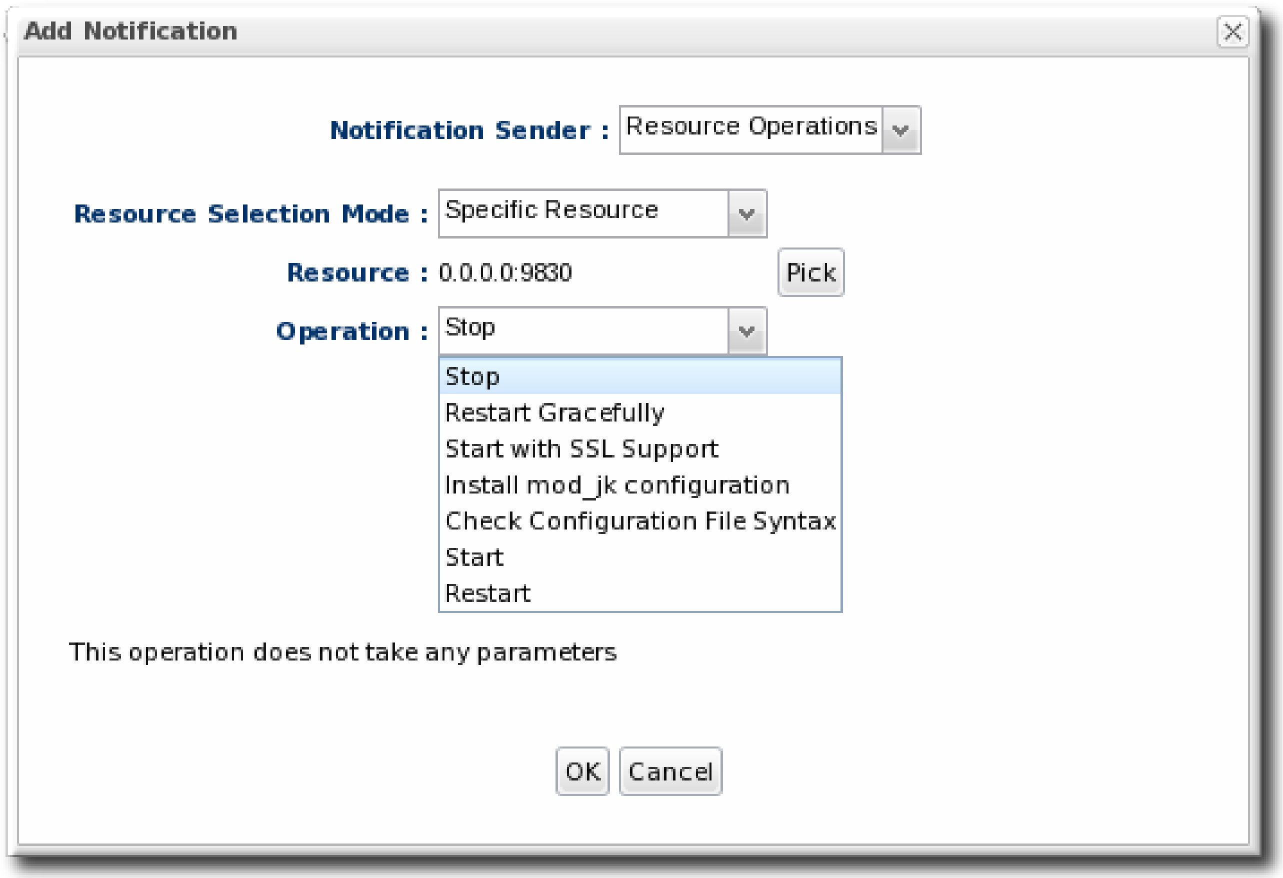
Figure 25.6. Resource Selection

IMPORTANT

If you select a **relative** resource and *do not* enter a specific resource name, then the operation will run on *every* resource which matches that resource type in the relative path. If no resource matches, then it is logged into the audit trail and the alert process proceeds.

For a relative resource, the resource name is not required. For a specific resource, it is.

The second half is selecting the operation type. The available operations and their configuration parameters depend on the type of resource selected as the target of the operation.


Add Notification

Notification Sender : Resource Operations

Resource Selection Mode : Specific Resource

Resource : 0.0.0.0:9830

Operation : Stop

- Stop
- Restart Gracefully
- Start with SSL Support
- Install mod_jk configuration
- Check Configuration File Syntax
- Start
- Restart

This operation does not take any parameters

Figure 25.7. Operation Settings

25.3.3. *Detailed Discussion:* Initiating Resource Scripts

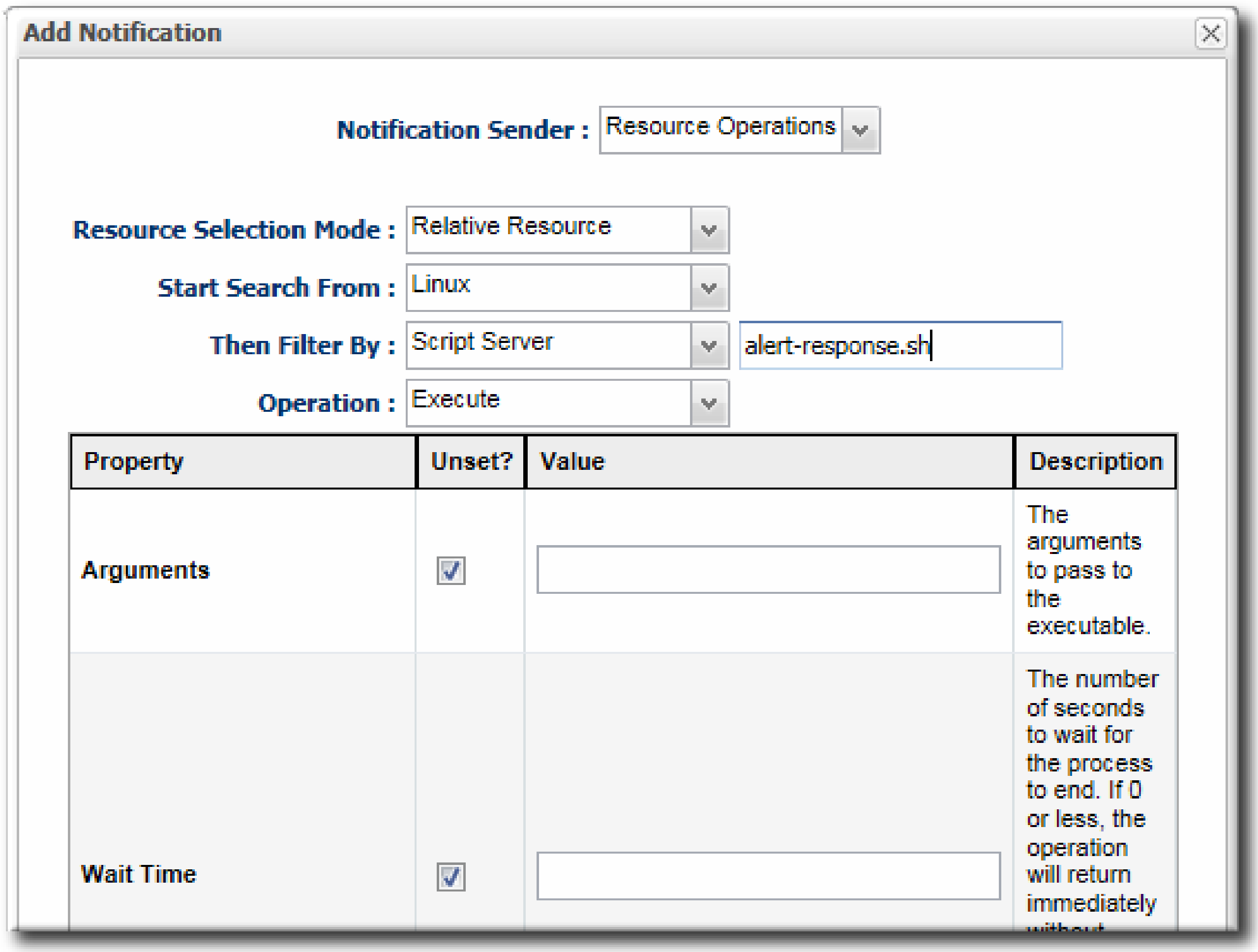
Scripts, such as shell and bat scripts, can be imported into the JBoss ON inventory and managed as resources (Section 5.4, “Importing New Resources Manually”). These scripts can include start scripts, configuration scripts, or diagnostic scripts.

As described in Section 25.3.2.2, “Setting Alert Operations”, an alert notification can run an operation on a resource (even a different resource than the one which triggered the alert). A specific usage scenario is to run a resource script as an operation.

NOTE

The script must be uploaded to the resource and added into the JBoss ON inventory before it can be used in an alert operation. This is covered in Section 5.4, “Importing New Resources Manually”.

Running a resource script is the same as running an operation. The resource script is selected as the resource for the operation, and then the start or restart operation for the script is set and, optionally, any command-line arguments to pass to the script.



Add Notification

Notification Sender :	Resource Operations
Resource Selection Mode :	Relative Resource
Start Search From :	Linux
Then Filter By :	Script Server / alert-response.sh
Operation :	Execute

Property	Unset?	Value	Description
Arguments	✓		The arguments to pass to the executable.
Wait Time	✓		The number of seconds to wait for the process to end. If 0 or less, the operation will return immediately

Figure 25.8. Resource Script Settings

IMPORTANT

If you select a **relative** resource and *do not* enter a specific script name in the name filter field, then the operation will run on *every* script resource that is in the relative path with the command arguments that are given. If no script matches, then it is logged into the audit trail and the alert process proceeds.

For a relative resource, the resource name is not required. For a specific resource, it is. To limit script execution to a single specific script, select the specific resource option and select the precise script from the list.

25.3.4. *Detailed Discussion:* Launching JBoss ON CLI Scripts from an Alert

JBoss ON has its own command-line client that can be used to manage server instances in the same way that the web UI manages servers. Much like running a script resource or launching an operation in response to an alert condition, a server CLI script can be run in response to an alert condition.

25.3.4.1. Notes for Using CLI Script Notifications

CLI Scripts Are Content

Unlike resource scripts, CLI scripts are not treated as resources in the inventory. These are tools available to and used by the JBoss ON server itself (not limited or associated with any given resource).

Server CLI scripts must be uploaded to the server as content within a repository before it can be run.

CLI Scripts and the Remote API

The CLI script must use the proper API to perform the operation on the server. JBoss ON has several different API sets, depending on the task being performed. To connect to a server and run a script requires the remoting API, which allows commands to be executed on the server remotely.

Writing CLI scripts is covered more in Writing JBoss ON Command-Line Scripts.

Referencing Alert Objects in a CLI Script

The CLI script can actually reference an alert object for the alert which triggers the script by using a pre-defined *alert* variable.

The **`alert`** variable implicitly identifies the alert definition and specific alert instance which has been fired. This allows you to create a proxy resource definition in the script that could be applied to any resource which uses that alert script.

```
var myResource =
ProxyFactory.getResource(alert.alertDefinition.resource.id)
```

The resource ID is pulled from the fired alert, and the script can reference the resource from there.

Limits on CLI Scripts

The resources themselves may impose limits on where a CLI script can be run or what operations it can perform. For example, for security reasons, a CLI script cannot perform a JNDI lookup on a local resource (performing a lookup on the server running the CLI script), but it can perform a remote JNDI lookup. Another common issue is that a JBoss ON server cannot run a restart operation on itself.

Be aware of potential security implications or resource constraints when writing a CLI script to be used as an alert response.

25.3.4.2. Writing Alert-Relevant CLI Scripts

Server-side scripts are both powerful and versatile. They can touch almost any server functionality and any resource, group, or other object and run a series of commands. This versatility makes CLI scripts very useful to responding to alerts in proactive ways, but it means that the alert should be planned to notify on specific conditions and the script should be designed to respond to that specific set of circumstances.

In other words, make the alert specific and the CLI script relevant.

This script checks the recent monitoring statistics for a web application and restarts the web server database if there are connection problems:

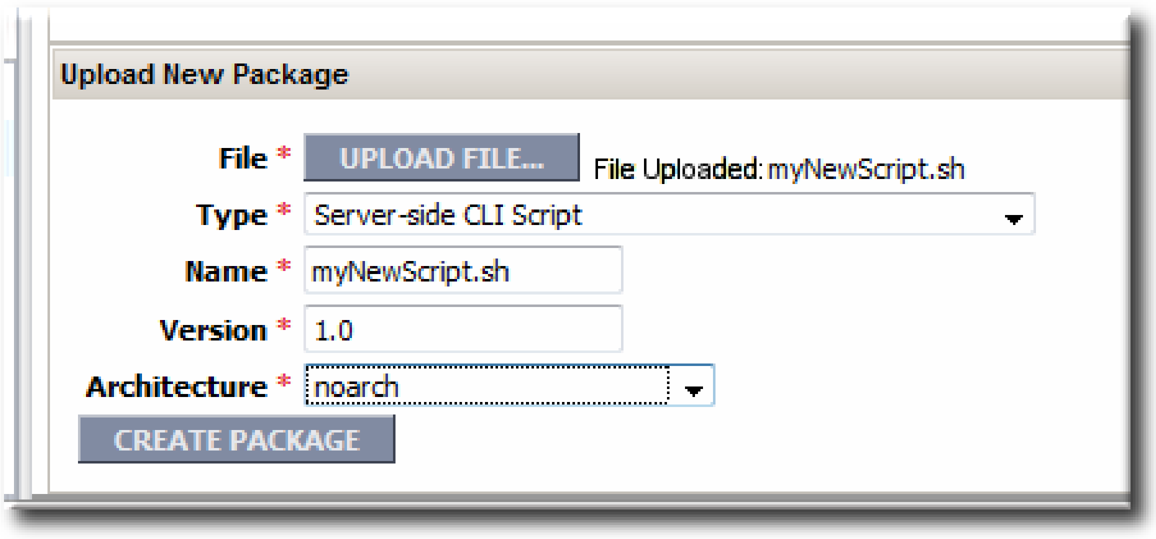
```
var myResource =
ProxyFactory.getResource(alert.alertDefinition.resource.id)

var definitionCriteria = new MeasurementDefinitionCriteria()
definitionCriteria.addFilterDisplayName('Sessions created per Minute')
definitionCriteria.addFilterResourceTypeId(myResource.resourceType.id)

var definitions =
MeasumentDefinitionManager.findMeasurementDefinitionsByCriteria(definition
Criteria)

if (definitions.empty) {
   throw new java.lang.Exception("Could not get 'Sessions created per
Minute' metric on resource "
      + myResource.id)
}

var definition = definitions.get(0)

var startDate = new Date() - 8 * 3600 * 1000 //8 hrs in milliseconds
var endDate = new Date()

var data = MeasurementDataManager.findDataForResource(myResource.id, [
definition.id ], startDate, endDate, 60)

exporter.setTarget('csv', '/the/output/folder/for/my/metrics/' + endDate +
'.csv')

exporter.write(data.get(0))

var dataSource = ProxyFactory.getResource(10411)

connectionTest = dataSource.testConnection()

if (connectionTest == null || connectionTest.get('result').booleanValue ==
false) {
    //ok, this means we had problems connecting to the database
    //let's suppose there's an executable bash script somewhere on the
server that
    //the admins use to restart the database
    java.lang.Runtime.getRuntime().exec('/somewhere/on/the/server/restart-
database.sh')
}
```

Commands, options, and variables for the JBoss ON CLI are listed in Writing JBoss ON Command-Line Scripts.

An example alert script is included with the server files in *serverInstallDir/*`alert-scripts/`.

25.3.4.3. Configuring a CLI Script Notification

1. Upload the script to a content repository.


Upload New Package

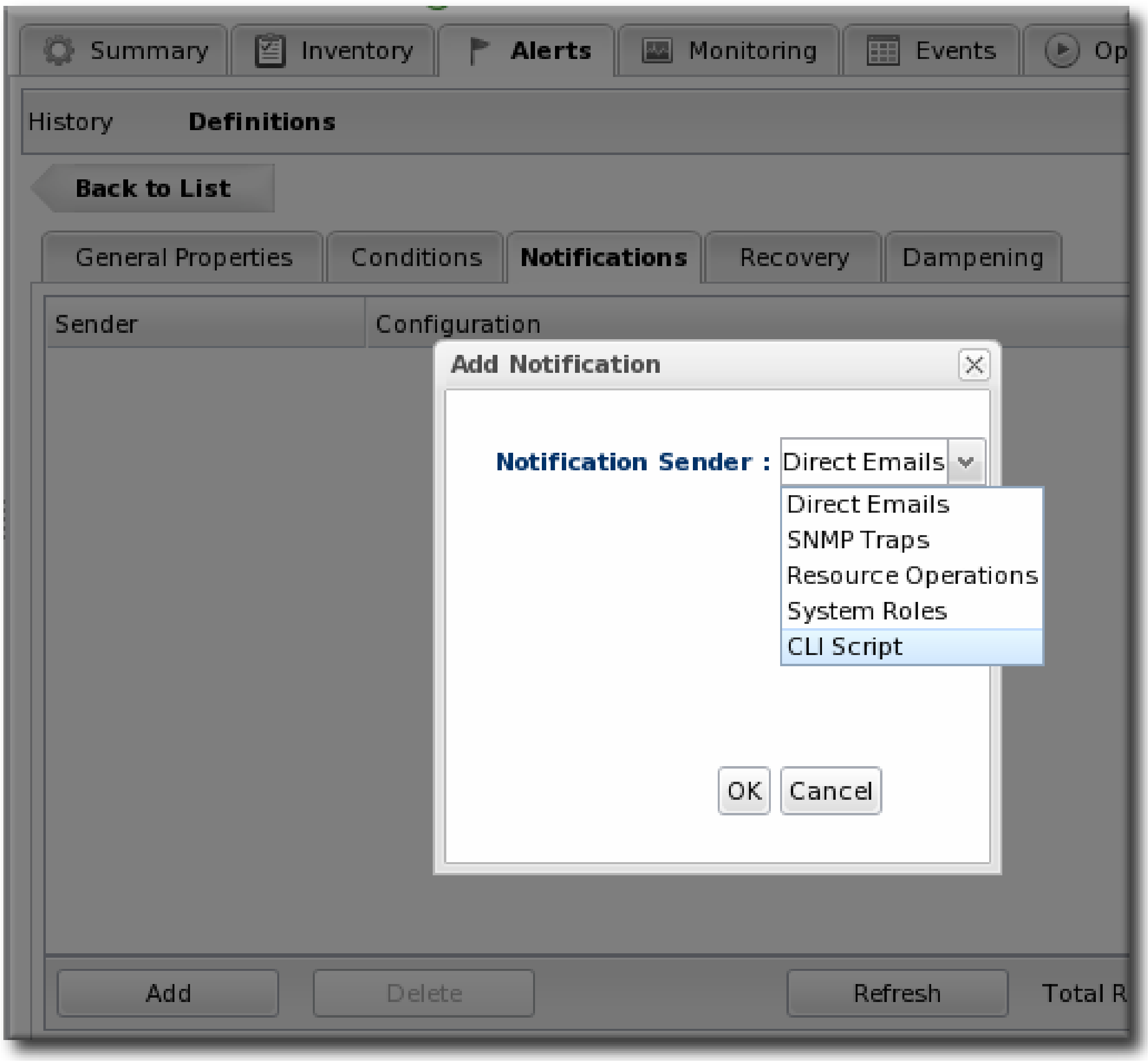
File *	UPLOAD FILE... File Uploaded: myNewScript.sh
Type *	Server-side CLI Script
Name *	myNewScript.sh
Version *	1.0
Architecture *	noarch

NOTE

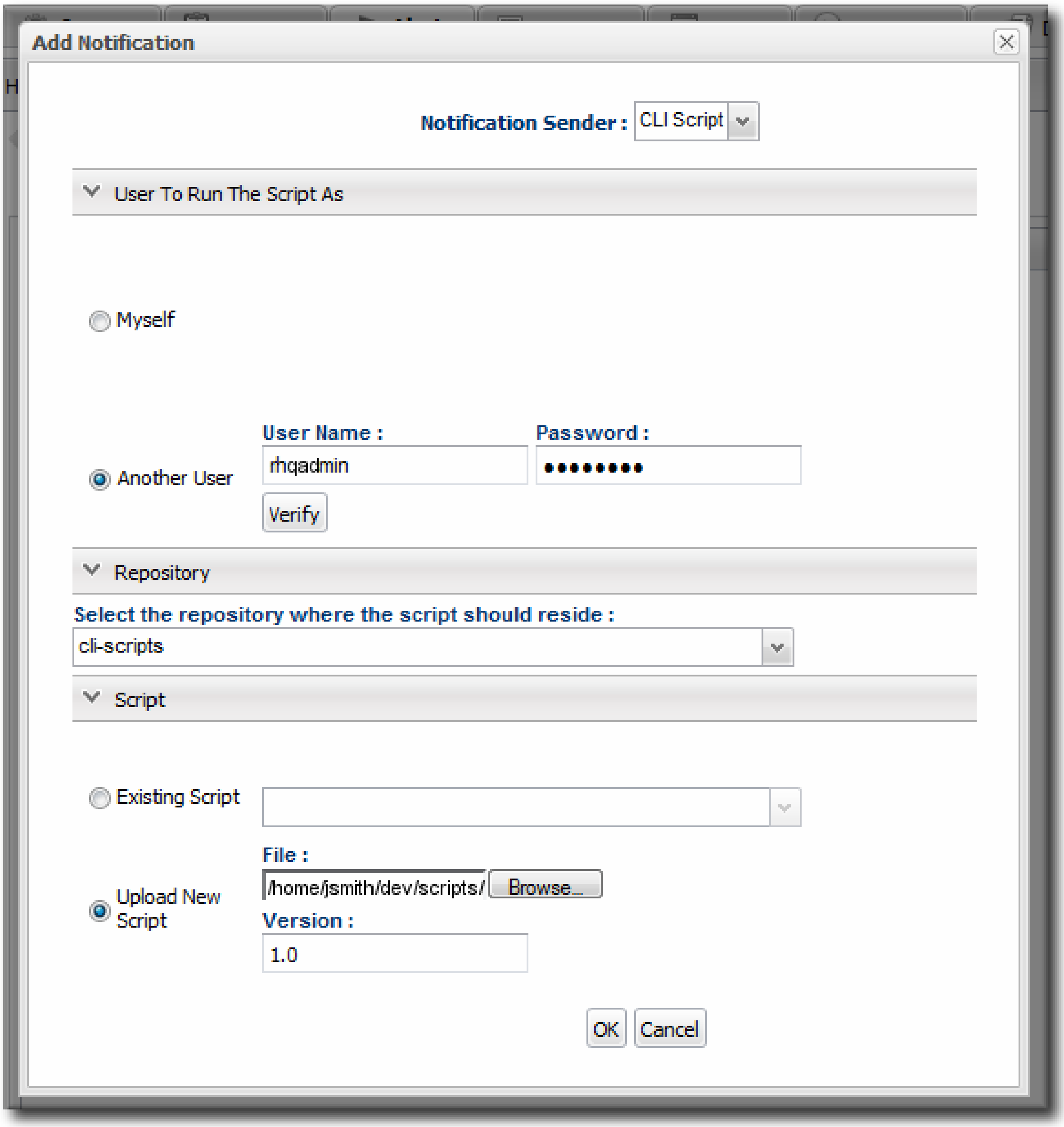
Create a separate repository for alert CLI scripts.

2. Search for the resource, and configure the basic alert definition, as in Section 25.1.2, “Basic Procedure for Setting Alerts for a Resource”.

3. In the `Notifications` tab for the alert definition, give the notification method a name, and select the `CLI Script` method from the **Alert Senders** drop-down menu.



4. First, select the JBoss ON user as whom to run the script. The default is as the user who is creating the notification.


Add Notification

Notification Sender : CLI Script

User To Run The Script As

Myself

[✓ Another User] User Name : rhqadmin Password : ●●●●●●●●●

Verify

Repository

Select the repository where the script should reside : cli-scripts

Script

Existing Script

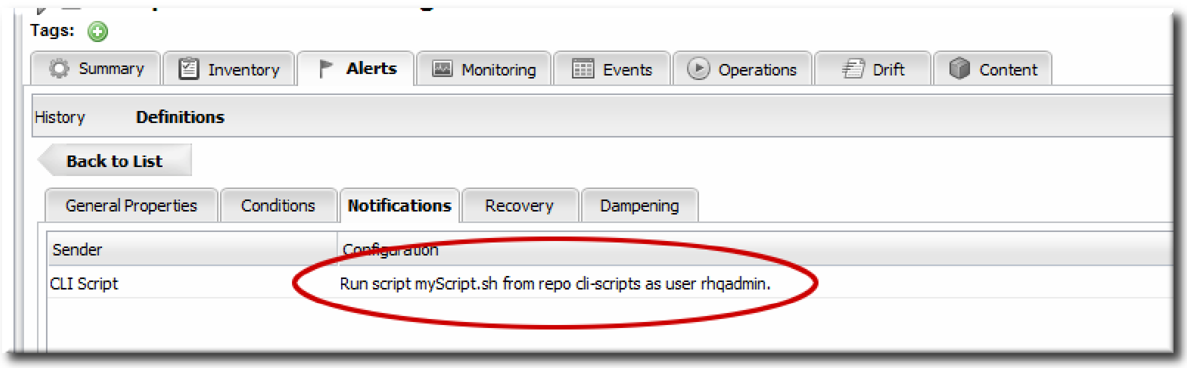
[✓ Upload New Script] File : /home/jsmith/dev/scripts/ Browse... Version : 1.0

OK Cancel

5. Select the repository which contains the CLI script. If you are uploading a new script, this is the repository to which the script will be added.

6. Select the CLI script to use from the drop-down menu, which lists all of the scripts in the specified repository. Alternatively, click the **`Upload`** button to browse to a script on the local machine.

7. Click **OK** to save the notification. The line in the `Notifications` tab shows the script, the repository, and the user as whom it will run.



Sender	Configuration
CLI Script	Run script myScript.sh from repo cli-scripts as user rhqadmin.

25.3.5. Configuring SNMP for Notifications

Configuring JBoss ON to send SNMP alerts has two parts:

- Configuring the SNMP alert plug-in for the server.
- Configuring the actual alert with an SNMP notification.

25.3.5.1. JBoss ON SNMP Information

JBoss ON can send SNMP traps to other management stations and systems as part of alerting notifications. The data transmitted contain details about the alert, such as the name of the alert that was triggered and the resource name.

The data to include in the traps, as with other SNMP notifications, are defined in the JBoss ON MIB file, in *serverRoot*/**etc/RHQ-mib.txt**. The default configuration for the MIB is shown in Example 25.1, "Default Alert Object in JBoss ON MIB". The base OID for the JBoss ON alert is **1.3.6.1.4.1.18016.2.1** (**org.dod.internet.private.enterprise.jboss.rhq.alert**).

Example 25.1. Default Alert Object in JBoss ON MIB

```
--internet(1.3.6.1)
+--private(4)
| +--enterprises(1)
| +--jboss(18016)
| +--rhq(2)
| +--alert(1)
| | +-- r-n DisplayString alertName(1)
| | +-- r-n DisplayString alertResourceName(2)
| | +-- r-n DisplayString alertPlatformName(3)
| | +-- r-n DisplayString alertCondition(4)
| | +-- r-n DisplayString alertSeverity(5)
| | +-- r-n DisplayString alertUrl(6)
| | +-- r-n DisplayString alertHierarchy(7)
| +--alertNotifications(2)
| | +--alertNotifPrefix(0)
| | +--alertNotification(1)
[alertName,alertResourceName,alertPlatformName,alertCondition,alertSever
ity,alertUrl,alertHierarchy]
| +--rhqServer(3)
+--snmpV2(6)
+--snmpModules(3)
```

```
+--rhqMIB(1)
+--rhqTraps(3)
+--rhqTrapPrefix(0)
```

NOTE

With the default MIB file, each trap sends the alert definition name, resource name, platform, alert conditions, severity, and a URL to the alert details page. The location of the MIB can be passed to the SNMP trap server to view the translated MIBs. For example, using NetSNMP:

```
[root@server ~]# snmptrapd -m RHQ-MIB -
M/opt/local/share/mibs/ietf
```

The `-M` gives the path to the SNMP server's MIB directory.

In that case, an alert notification is recorded in the trap like this:

```
Jul 8 15:13:31 snert snmptrapd[42372]: 127.0.0.1: Enterprise
Specific Trap (.0) Uptime: 0:00:00.00, RHQ-MIB::alertName =
STRING: test, RHQ-MIB::alertResourceName = STRING: snert, RHQ-
MIB::alertPlatformName = STRING: snert, RHQ-MIB::alertCondition
= STRING:
- Condition 1: Free Memory < 1.024,0MB
- Date/Time: 2013/07/08 15:13:05 MESZ
- Details: 12,9MB
, RHQ-MIB::alertSeverity = STRING: medium, RHQ-MIB::alertUrl =
STRING:
http://localhost:7080/coregui/CoreGUI.html#Resource/10001/Alert
s/History/12204, RHQ-MIB::alertHierarchy = STRING: snert
```

The output is similar in **tcpdump**, just all on the same line:

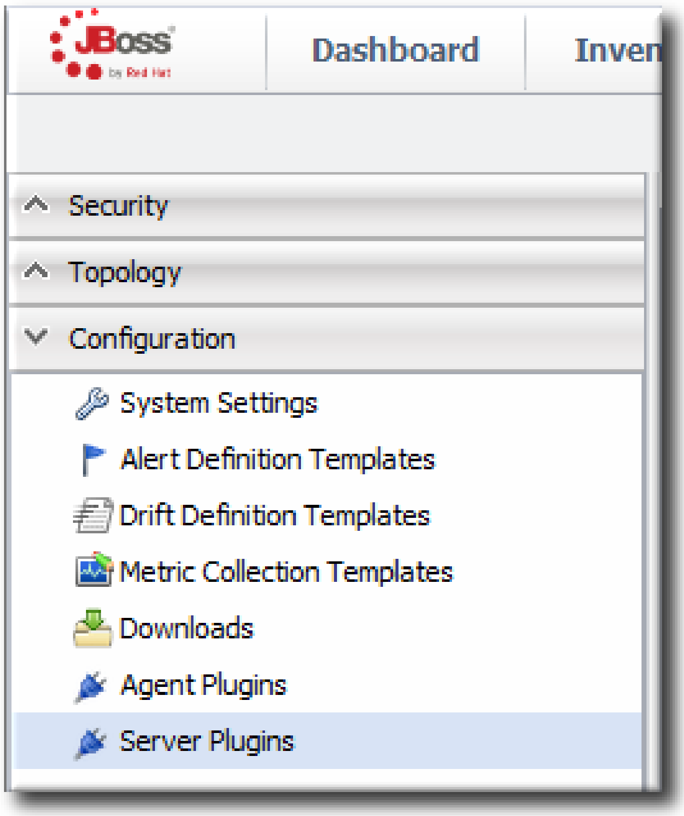
```
22:06:19.043208 IP localhost.56445 > localhost.snmptrap:
Trap(352) E:18016.2.3 0.0.0.0 enterpriseSpecific s=0 0
E:18016.2.1.1="test" E:18016.2.1.2="snert"
E:18016.2.1.3="snert" E:18016.2.1.4="^J- Condition 1: Free
Memory < 4,0GB^J- Date/Time: 2013/07/10 22:06:01 MESZ^J -
Details: 179,2MB^J" E:18016.2.1.5="medium"
E:18016.2.1.6="http://localhost:7080/coregui/CoreGUI.html#Resou
rce/10001/Alerts/History/10038" E:18016.2.1.7="snert"
```

25.3.5.2. Configuring the SNMP Alert Plug-in

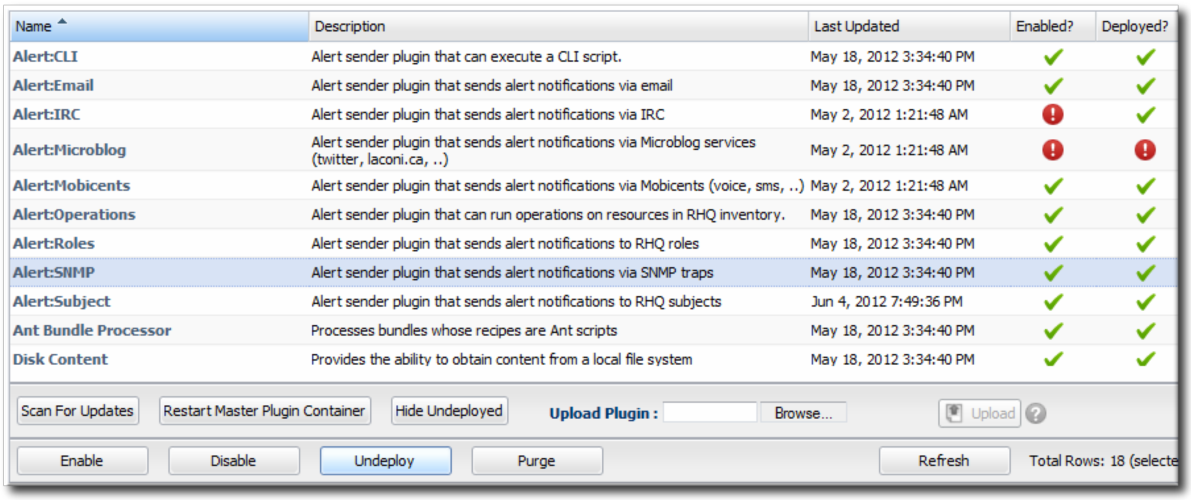
The SNMP alert sender plug-in is the only alert notification plug-in that requires additional configuration before the notification method can be used. The SNMP plug-in has to be configured with the appropriate SNMP version and SNMP agent information.

1. In the top menu, select the **Administration** tab.

2. In the **System Configuration** menu, select the **ServerPlugins** item.



3. Click the name of the SNMP plug-in in the list.

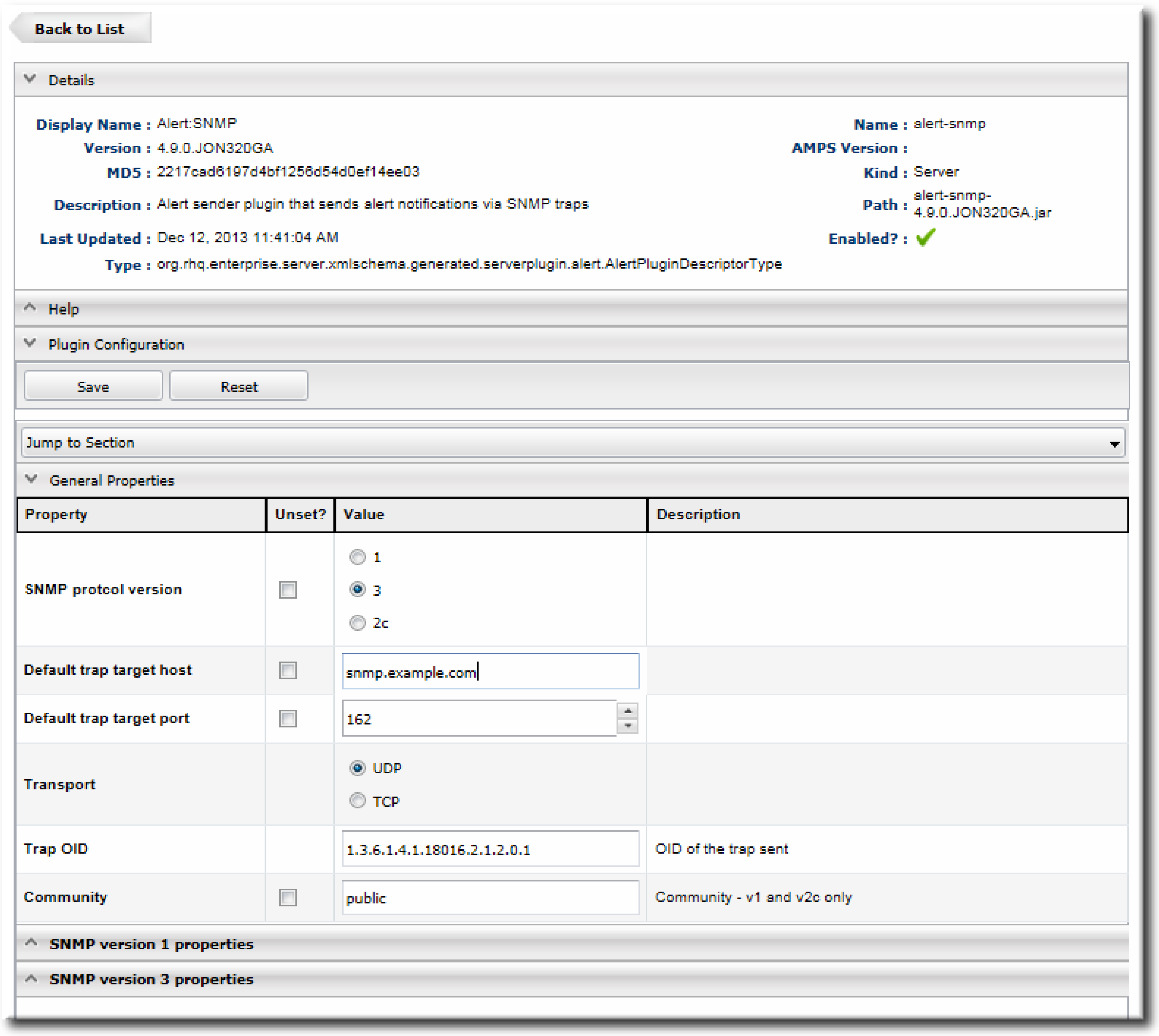


Name	Description	Last Updated	Enabled?	Deployed?
Alert:CLI	Alert sender plugin that can execute a CLI script.	May 18, 2012 3:34:40 PM	✓	✓
Alert:Email	Alert sender plugin that sends alert notifications via email	May 18, 2012 3:34:40 PM	✓	✓
Alert:IRC	Alert sender plugin that sends alert notifications via IRC	May 2, 2012 1:21:48 AM	❗	✓
Alert:Microblog	Alert sender plugin that sends alert notifications via Microblog services (twitter, laconi.ca, ..)	May 2, 2012 1:21:48 AM	❗	❗
Alert:Mobicents	Alert sender plugin that sends alert notifications via Mobicents (voice, sms, ..)	May 2, 2012 1:21:48 AM	✓	✓
Alert:Operations	Alert sender plugin that can run operations on resources in RHQ inventory.	May 18, 2012 3:34:40 PM	✓	✓
Alert:Roles	Alert sender plugin that sends alert notifications to RHQ roles	May 18, 2012 3:34:40 PM	✓	✓
Alert:SNMP	Alert sender plugin that sends alert notifications via SNMP traps	May 18, 2012 3:34:40 PM	✓	✓
Alert:Subject	Alert sender plugin that sends alert notifications to RHQ subjects	Jun 4, 2012 7:49:36 PM	✓	✓
Ant Bundle Processor	Processes bundles whose recipes are Ant scripts	May 18, 2012 3:34:40 PM	✓	✓
Disk Content	Provides the ability to obtain content from a local file system	May 18, 2012 3:34:40 PM	✓	✓

4. In the plug-in details page, expand the **`Plugin Configuration`** section.

5. Fill in the required SNMP configuration:

   - Select the appropriate SNMP version.
   - Give the hostname, port number, and transport protocol for the SNMP trap server. The default settings point to the localhost for the JBoss ON server and port 162.
   - Set the trap OID. This is, by default, the RHQ OID.
   - *For SNMP 1 and 2.* Give the name of the community.



Back to List

Details

Display Name :	Alert:SNMP	Name :	alert-snmp
Version :	4.9.0.JON320GA	AMPS Version :	
MD5 :	2217cad6197d4bf1256d54d0ef14ee03	Kind :	Server
Description :	Alert sender plugin that sends alert notifications via SNMP traps	Path :	alert-snmp-4.9.0.JON320GA.jar
Last Updated :	Dec 12, 2013 11:41:04 AM	Enabled? :	✓
Type :	org.rhq.enterprise.server.xmlschema.generated.serverplugin.alert.AlertPluginDescriptorType		

Help

Plugin Configuration

Save Reset

Jump to Section

General Properties

Property	Unset?	Value	Description
SNMP protcol version	☐	1 [✓ 3] 2c	
Default trap target host	☐	snmp.example.com	
Default trap target port	☐	162	
Transport		[✓ UDP] TCP	
Trap OID		1.3.6.1.4.1.18016.2.1.2.0.1	OID of the trap sent
Community	☐	public	Community - v1 and v2c only

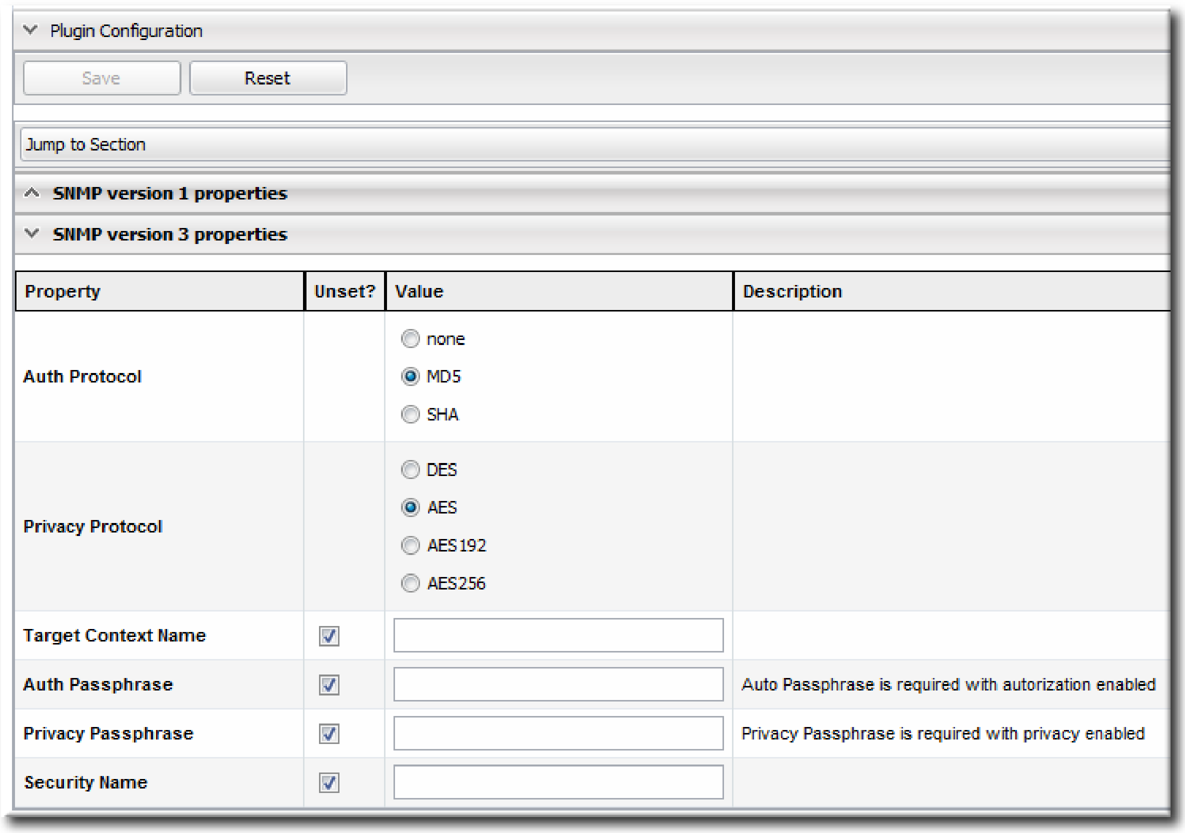
SNMP version 1 properties

SNMP version 3 properties

NOTE

This sets global defaults for SNMP alert notifications. Different settings can be given to individual alert notifications when an alert definition is created.

6. SNMP version 1 and version 3 both require additional configuration, as listed in Table 25.3, "SNMP v1 Configuration Settings" and Table 25.4, "SNMP v3 Configuration Settings". Expand the version-specific configuration section and fill in the information about the SNMP agent.



Property	Unset?	Value	Description
Auth Protocol		none [✓ MD5] SHA	
Privacy Protocol		DES [✓ AES] AES192 AES256	
Target Context Name	✓		
Auth Passphrase	✓		Auto Passphrase is required with autorization enabled
Privacy Passphrase	✓		Privacy Passphrase is required with privacy enabled
Security Name	✓		

It may be necessary to unselect the **Unset** checkbox to allow the fields to be edited.

Table 25.3. SNMP v1 Configuration Settings

Field	Description
Generic ID	The ID of the trap. The SNMP standards defines numbers from 0-6, with 6 meaning "enterprise specific," which is the default.
Enterprise OID	The OID of the JBoss ON server itself. The default value is taken from the RHQ MIB as *SMIv2.enterprise.jboss.rhq.rhqServer.*
Specific OID	A custom OID to use with the trap notification. This can be empty.
Agent address	The IP address of the alert sender, which means the IP address of the JBoss ON server.

Table 25.4. SNMP v3 Configuration Settings

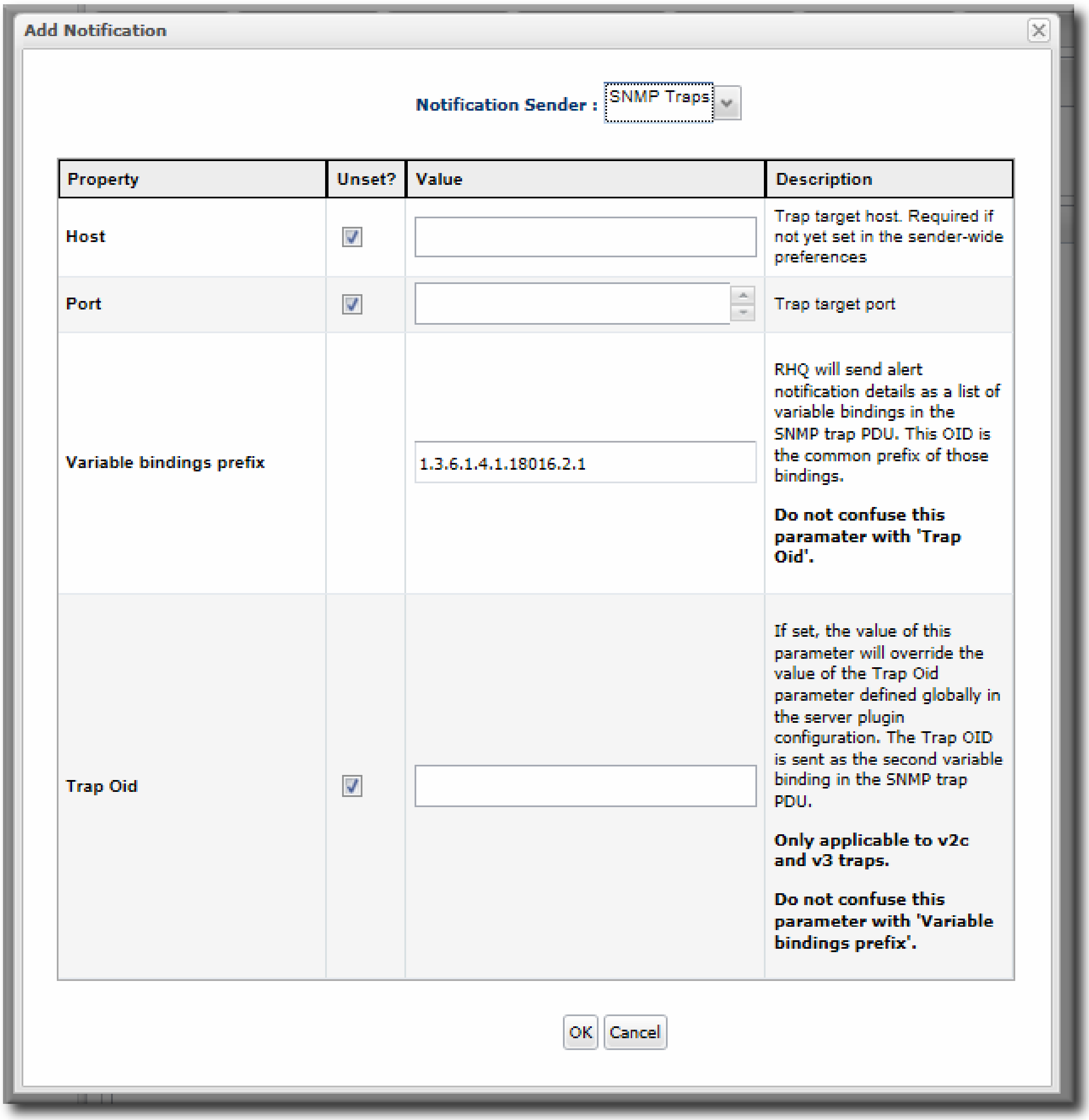
Field	Description
Auth protocol	The encryption algorithm for authentication requests. Setting this requires a corresponding authentication passphrase to be set. If there is no passphrase, this value must be **none**.

Field	Description
Privacy protocol	Sets the encryption method used with trap messages. This is used with the authentication proocol.
Engine Id	
Target Context name	
Auth passphrase	Sets the password used for authentication; this has a miminum length of eight (8) characters. This is required if an `Auth Protocol` value is set.
Privacy passphrase	Sets the password used for managing encrypted communication. This is required if authentication is used.
Security name	Gives the username to use for authentication to the trap receiver.

25.3.5.3. Configuring the SNMP Alert Notification

Before JBoss ON can send any SNMP notifications, SNMP traps have to be configured for the server.

When configuring the alert, select the SNMP Trap notification type, and fill in the JBoss ON SNMP information.



Notification Sender : SNMP Traps

Property	Unset?	Value	Description
Host	✓		Trap target host. Required if not yet set in the sender-wide preferences
Port	✓		Trap target port
Variable bindings prefix		1.3.6.1.4.1.18016.2.1	RHQ will send alert notification details as a list of variable bindings in the SNMP trap PDU. This OID is the common prefix of those bindings. **Do not confuse this paramater with 'Trap Oid'.**
Trap Oid	✓		If set, the value of this parameter will override the value of the Trap Oid parameter defined globally in the server plugin configuration. The Trap OID is sent as the second variable binding in the SNMP trap PDU. **Only applicable to v2c and v3 traps.** **Do not confuse this parameter with 'Variable bindings prefix'.**

Figure 25.9. JBoss ON SNMP Trap Information

- The hostname for the SNMP manager. If this is not set, the default from the global configuration is used.
- The port number for the SNMP manager. If this is not set, the default from the global configuration is used.
- A variable binding prefix. *Optional.* This prepends the given string to the beginning of the individual variables sent by the trap. This can be a way to identify the JBoss ON server, resource, or alert which is sending the trap. The default is set in the RHQ MIB, `SMIv2.enterprise.jboss.rhq.alert`.
- The trap OID. This is the specific OID to use with the trap definition. If this is not set, the default from the global configuration is used; by default, this is the RHQ-MIB, `1.3.6.1.4.1.18016.2.1`.

25.4. VIEWING ALERT DATA

Once an alert notification is triggered, JBoss ON records it in an *alert history* for the resource. The history includes the time the alert was sent and the condition which triggered it. Crucially, JBoss ON also shows whether an alert has been acknowledged and, if so, what user addressed it.

While monitoring and alerting work together to make administrators immediately aware of a problem with a resource, the alert history gives a much broader view into resource performance *and* IT staff responses. Knowing, historically, what kind of alerts have been triggered and when can help with root cause analysis, incident response, and maintenance policies.

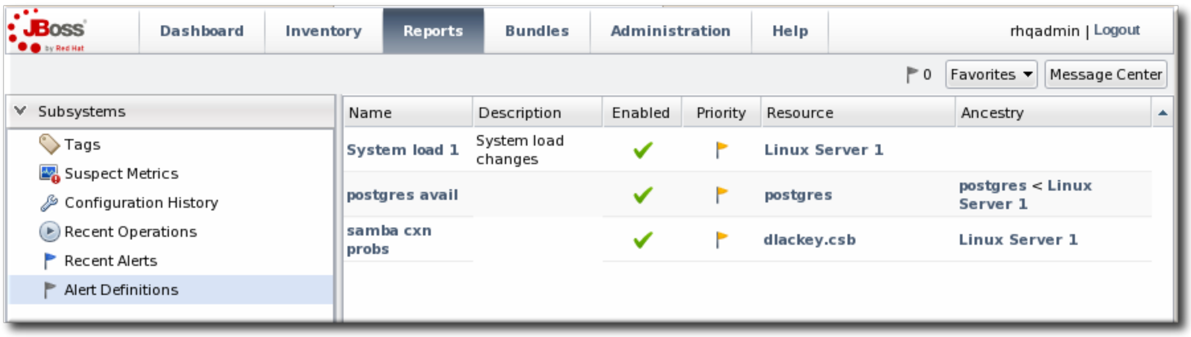
25.4.1. Viewing the Alert Definitions Report

While the alert definitions for a specific resource are always available by viewing that resource entry, it is also possible to view all of the alert definitions configured in JBoss ON in the Alert Definitions Report.

1. Select the **`Reports`** tab in the top navigation bar.

2. In the **`Subsystems`** menu box on the left, select **`Alert Definitions`**.

3. The definitions report shows a list of all configured definitions, for all resources in the inventory.



Name	Description	Enabled	Priority	Resource	Ancestry
System load 1	System load changes	✓	⚑	Linux Server 1	
postgres avail		✓	⚑	postgres	postgres < Linux Server 1
samba cxn probs		✓	⚑	dlackey.csb	Linux Server 1

The results table provide the most basic information for the definitions:

- The resource (**`Name`**).

- The parent or *ancestry*. Since resources are arranged hierarchically, sorting by the parent is very useful for finding all alert definitions for all services and applications that relate to a high-level resource like a server.

- The description of the alert.

- Whether it is active (enabled).

NOTE

A user may have the write to create and edit an alert definition, but that does not mean that the user has the right to delete an item from the alert history.

Deleting elements in the history requires the manage inventory permission.

NOTE

Reports can be exported to CSV, which can be used for office systems or further data manipulation.

To export a report, simply click the **`Export`** button. The report will automatically be downloaded as **`alertDefinitions.csv`**.

25.4.2. Viewing Alerts

The alert history can be reviewed for a resource, a group of resources, a parent, or the whole JBoss ON server.

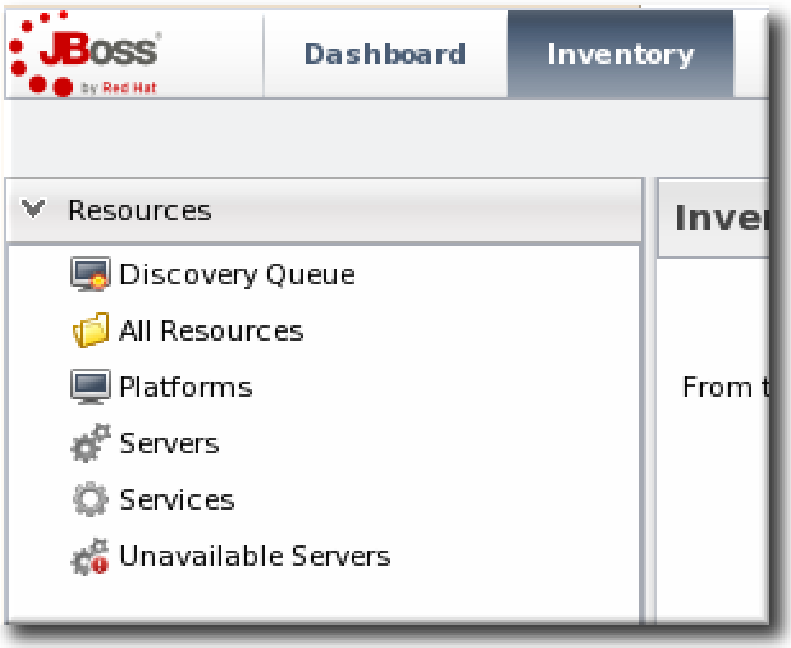
25.4.2.1. Viewing Alert Details for a Specific Resource

NOTE

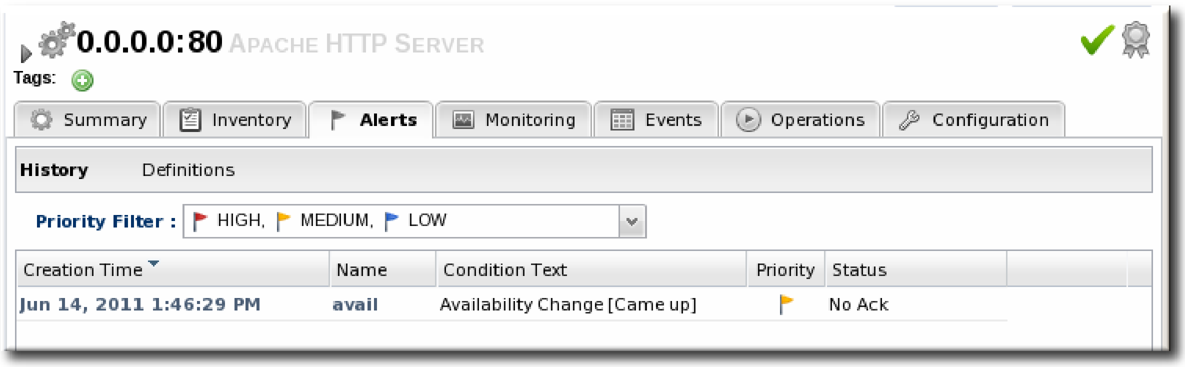
A user may have the write to create and edit an alert definition, but that does not mean that the user has the right to delete an item from the alert history.

Deleting elements in the history requires the manage inventory permission.

1. Click the **`Inventory`** tab in the top menu.
2. Select the resource type in the **`Resources`** menu table on the left, and then browse or search for the resource.

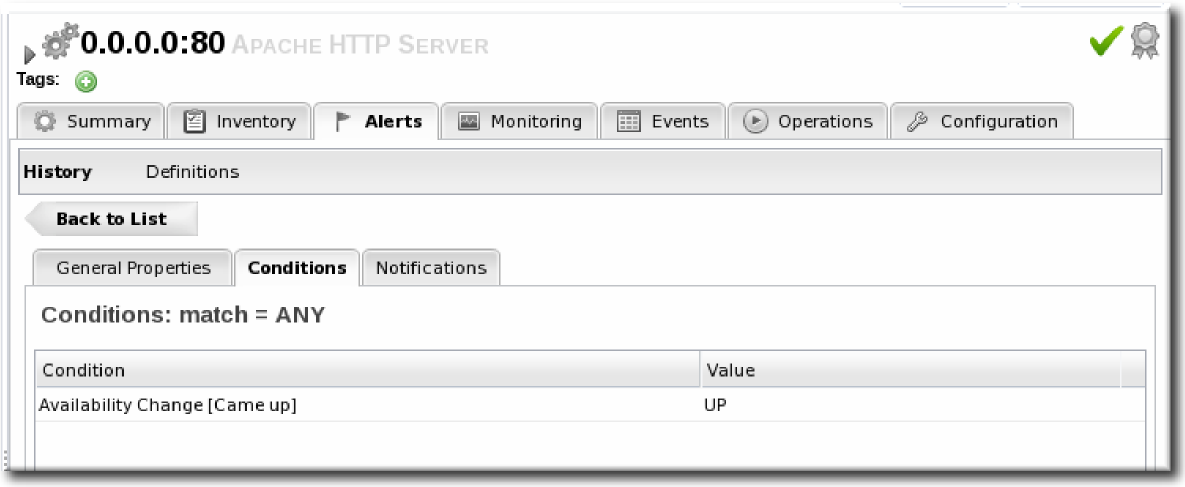


3. Click the resource in the list.
4. Click the **`Alerts`** tab, and make sure that the **`History`** subtab is selected.
5. In the list, click the timestamp or alert definition name for the fired alert.



Creation Time	Name	Condition Text	Priority	Status
Jun 14, 2011 1:46:29 PM	avail	Availability Change [Came up]		No Ack

6. The alert page has tabs for each detail for the alert, including which alert definition was triggered, the conditions that triggered, and any operations that were launched as a result.



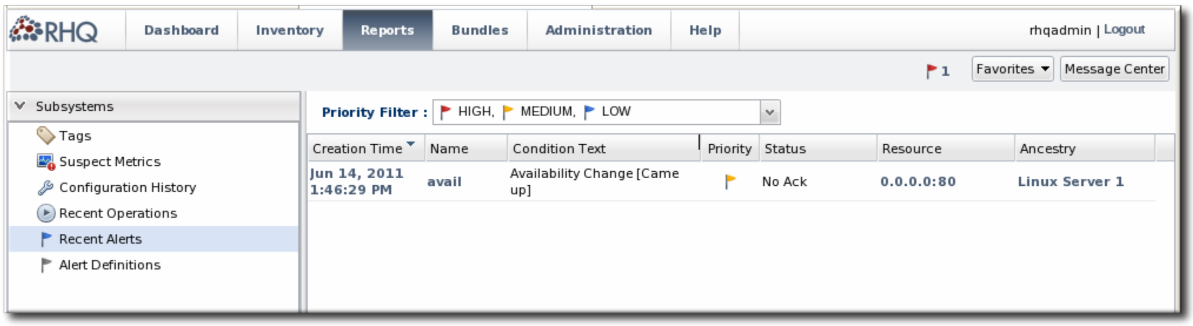
Conditions: match = ANY

Condition	Value
Availability Change [Came up]	UP

25.4.2.2. Viewing the Fired Alerts Report

1. Select the **`Reports`** tab in the top navigation bar.

2. In the **`Subsystems`** menu box on the left, select **`Recent Alerts`**.



Creation Time	Name	Condition Text	Priority	Status	Resource	Ancestry
Jun 14, 2011 1:46:29 PM	avail	Availability Change [Came up]		No Ack	0.0.0.0:80	Linux Server 1

All of the alerts for all resources in JBoss ON are listed in the results table. Several elements are useful for analysis:

- The resource (**`Name`**)
- The parent (ancestor)
- The name of the definition which triggered the alert
- The condition which triggered the alert

- The value of the resource at the time the alert was sent
- The date, which is very useful for correlating the alert notification to an external event

NOTE

A user may have the write to create and edit an alert definition, but that does not mean that the user has the right to delete an item from the alert history.

Deleting elements in the history requires the manage inventory permission.

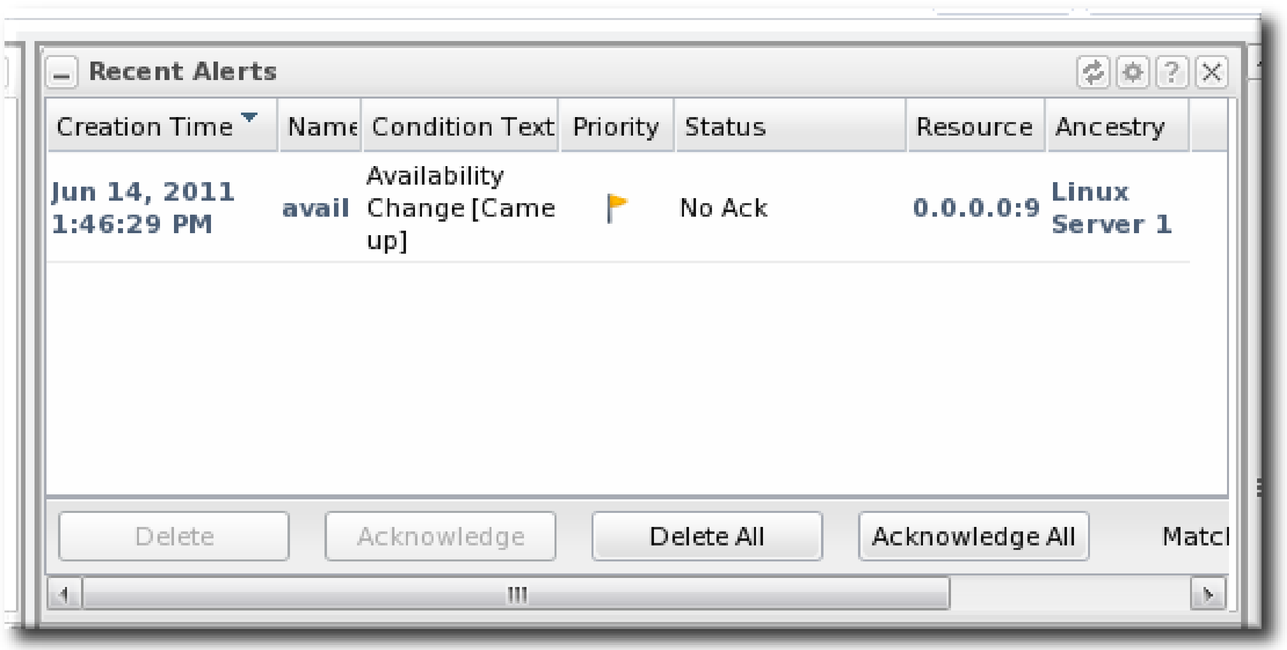
NOTE

Reports can be exported to CSV, which can be used for office systems or further data manipulation.

To export a report, simply click the **`Export`** button. The report will automatically be downloaded as **`recentAlerts.csv`**.

25.4.2.3. Viewing Alerts in the Dashboard

All of the recently-fired alerts, by default, are listed on the **`Dashboard`** page of JBoss ON in the recent alerts portlet.



Recent Alerts

Creation Time	Name	Condition Text	Priority	Status	Resource	Ancestry
Jun 14, 2011 1:46:29 PM	avail	Availability Change [Came up]	⚑	No Ack	0.0.0.0:9	Linux Server 1

Figure 25.10. Recent Alerts Portlet

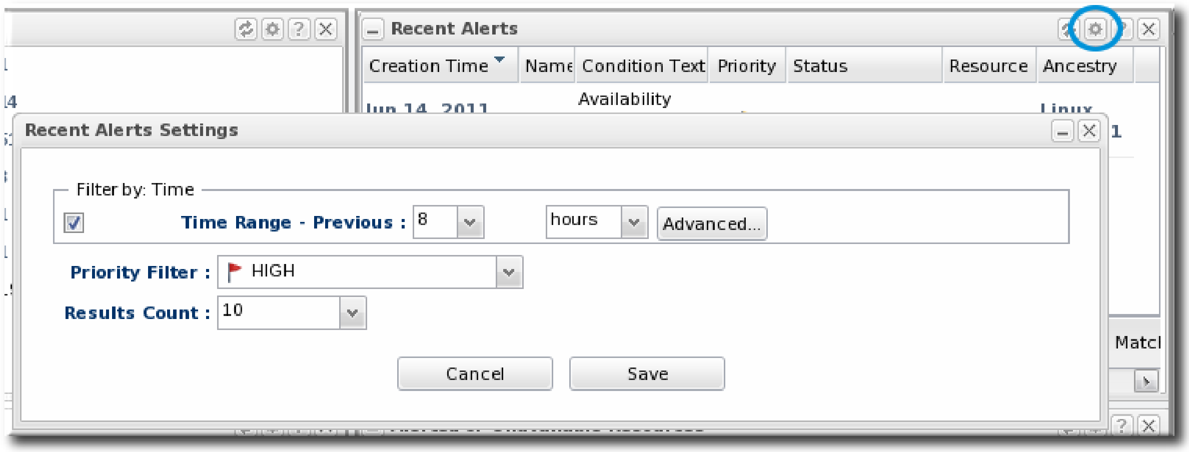
The alerts displayed in the portlet can be filtered for different conditions:

- A time range for when the alert was fired
- The alert priority (which is initially configured in the alert definition)

These conditions are evaluated *in order*, meaning that alerts are filtered first based on time, then priority.

To set the conditions for the alerts portlet in the **`Dashboard`** page:

1. In the top menu, click **Dashboard**.
2. In the **Recent Alerts** portlet, click the gear icon to open the portlet configuration page.


Recent Alerts Settings

Filter by: Time
Time Range - Previous : 8 hours Advanced...
Priority Filter : HIGH
Results Count : 10

Cancel Save

3. Change the display criteria as desired.

25.4.3. Acknowledging an Alert

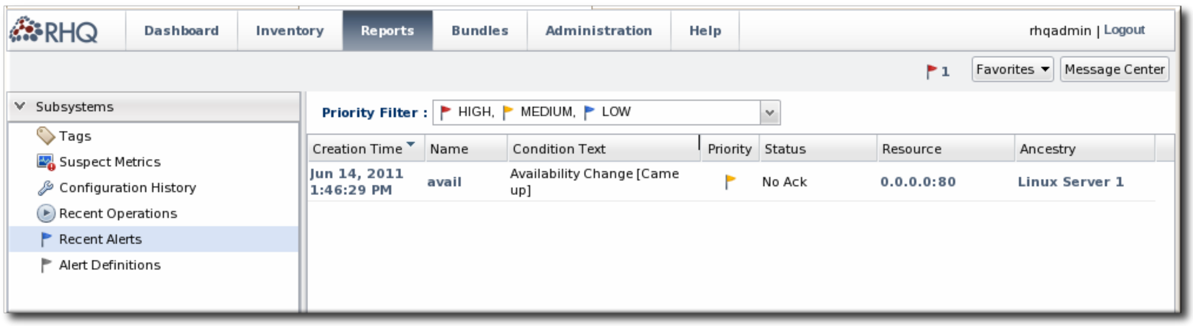
Acknowledging an alert is a way of identifying that the condition which triggered the alert has been addressed in some way. When an alert is acknowledged, the name of the user who acknowledged the alert is recorded. Recording the acknowledger's name allows the action to be audited later if necessary.

There are several different ways to acknowledge an alert:

- Through the Recent Alerts Report
- Through a group
- Through the resource entry

Using the Recent Alerts Report is useful because you can acknowledge multiple alerts at the same time and for multiple resource types, which could be simpler if a known outage triggered many alerts. Acknowledging an alert is not a requirement to close the alert, but it can be useful as part of auditing an incident response or making sure that issues have been addressed.

1. Select the **Reports** tab in the top navigation bar.
2. In the **Subsystems** menu box on the left, select **Recent Alerts**.

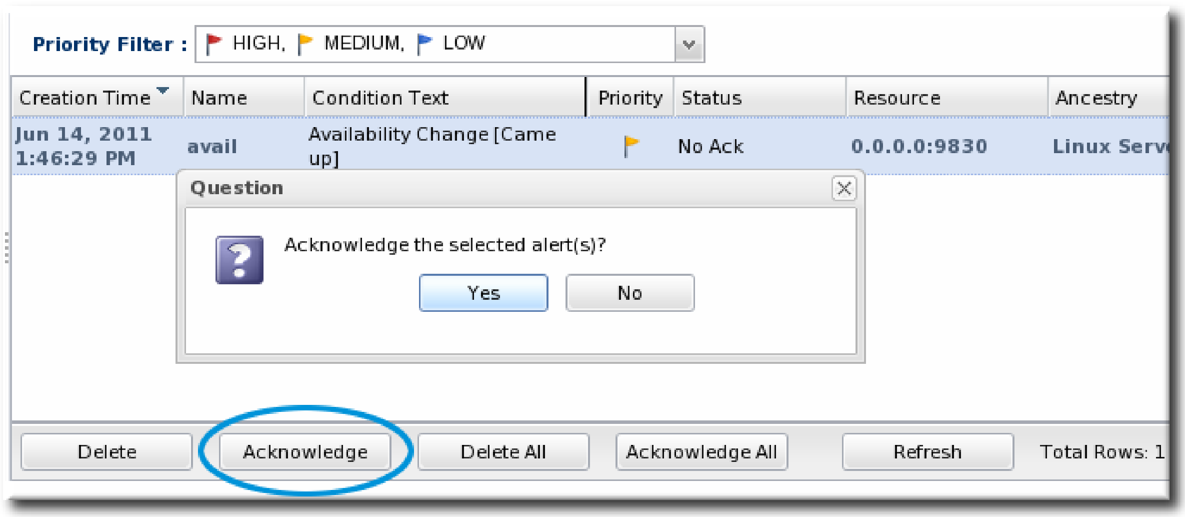


Priority Filter : HIGH, MEDIUM, LOW

Creation Time	Name	Condition Text	Priority	Status	Resource	Ancestry
Jun 14, 2011 1:46:29 PM	avail	Availability Change [Came up]		No Ack	0.0.0.0:80	Linux Server 1

3. Select the alert to acknowledge.

4. Click the **Acknowledge** button, and, when prompted, confirm the action.

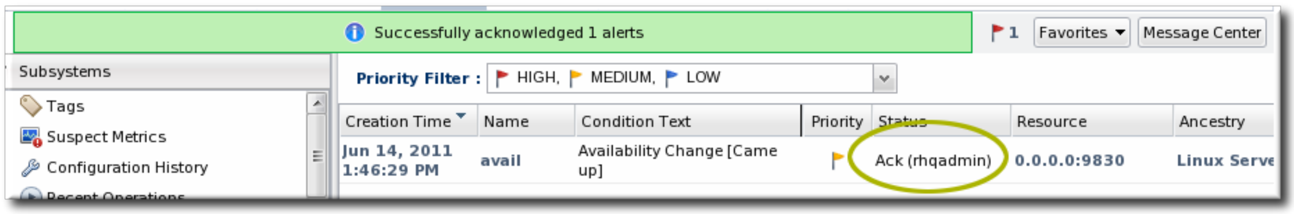


Creation Time	Name	Condition Text	Priority	Status	Resource	Ancestry
Jun 14, 2011 1:46:29 PM	avail	Availability Change [Came up]		No Ack	0.0.0.0:9830	Linux Serv

NOTE

It is also possible to acknowledge a single alert through the alert details page.

When the alert is acknowledged, the **Status** shows the name of the user who acknowledged (and presumably resolved) the alert.



Creation Time	Name	Condition Text	Priority	Status	Resource	Ancestry
Jun 14, 2011 1:46:29 PM	avail	Availability Change [Came up]		Ack (rhqadmin)	0.0.0.0:9830	Linux Serve

Figure 25.11. Alert Acknowledgment

PART IV. DEPLOYING APPLICATIONS AND CONTENT

CHAPTER 26. SUMMARY: USING JBOSS ON TO DEPLOY APPLICATIONS AND UPDATE CONTENT

One of the core management tools for JBoss Operations Network is to create, update, or remove content from its managed resources. *Content* can be anything associated with a resource or configuration, such as text files, binary files like JARs, EARs, and WARs, patches, and XML files. That content can be deployed on a managed resource to update that resource's configuration, to create a child resource, or to deploy an entirely new application.

There are two ways to manage content for resources:

- Resource-level content through the `Content` tabs
- Provisioning applications through *bundles*

Resource-level content allows a specific managed resource, usually a JBoss application server or a web server, to be associated with stored and versioned packages in named repositories. These packages can be uploaded into JBoss ON (so JBoss ON is essentially the content repository), they can be pulled from an external repository, or they can be discovered through agent plug-ins. In other words, there are three actions that resource-level content management can perform:

- It can deliver packages, updates, and patches to a resource.
- It can deploy content to a resource and even create a new child resource. This is particularly useful with web and application servers which can have different contexts as children.
- It can discover the current packages installed on a resource, creating a package digest that administrators can use to manage that asset.

Resource-level content management is limited how far it can be used to create resources. That is why JBoss ON has another system of deploying content, one that allows it to deploy full application servers or to consistently apply content across multiple resources: *provisioning through bundles*.

Bundles are added to the JBoss ON server, so they are not restricted to a single resource. They are deployed to compatible groups of resources, either platforms or JBoss servers (or other resource types which define a bundle target in their plug-in descriptor). This allows multiple resources to be updated at once, using the same content.

Bundle provisioning also allows more flexible and complex deployment options:

- Use Ant calls to perform operations before or after deploying the bundle
- Allow user-defined values or edits to configuration at the time a bundle is provisioned
- Have multiple versions of the same content bundle deployed and deployable to resources at the same time
- Revert to an earlier bundle version

CHAPTER 27. DEPLOYING CONTENT AND APPLICATIONS THROUGH BUNDLES

27.1. AN INTRODUCTION TO PROVISIONING CONTENT BUNDLES

Provisioning is a way that administrators can define and control applications, from development to production. The ultimate effect of the provisioning system is simplifying how applications are deployed. Administrators can control which versions of the same application are deployed to different resources, from different content sources, within the same application definition (the *bundle definition*). Resources can be reverted to different versions or jump ahead in deployment.

Provisioning takes one set of files (*a bundle*) and then pushes it to a platform or an application server (*the destination*). There are more complex ways of defining the content, the destinations, and the rules for that deployment, but the core of the way that provisioning handles content is to take versioned bundles and send it to the designated resource.

Provisioning works with compatible groups, not individual resources. Administrators can define groups based on disparate environments and consistently apply application changes (upgrades, new deployments, or reversions) across all group members, simultaneously.

And the type of content which can be deployed, itself, is flexible. A bundle can contain raw configuration files, scripts, ZIP archives, JAR files, or full application servers — the definition of *content* is fairly loose.

This is in contrast to the resource-level content management in JBoss ON. The type of content is relatively limited. Patches or configuration is applied per-resource. New applications can only be deployed as children of existing resources and it has to be another resource type.

Provisioning focuses on *application management*, not purely resource management.

27.1.1. Bundles: Content and Recipes

A *bundle* is a set of content, packaged in an archive. In real life, a bundle is usually an application, but it can also contain a set of configuration files, scripts, libraries, or any other content required to set up an application or a resource.

The purpose of a bundle is to take that defined set of content and allow JBoss ON to copy it onto a remote resource. The provisioning process basically builds the application on the targeted resource, so in that sense, the bundle is an application distribution. Each bundle version has its own *recipe* which tells JBoss ON what files exist in the bundle, any *tokens* which need to have real values supplied at deployment, and how to handle the bundle and existing files on the remote machine.

The recipe, configuration files, and content are all packaged together into the bundle. This is usually a ZIP file, which the agent unpacks during provisioning.

As with other content managed in JBoss ON, the bundle is versioned. Different versions can be deployed to different resources, which is good for handling different application streams in different environments (say, QA and production). Versioning bundles also allows JBoss ON to revert or upgrade bundles easily.

The bundle can contain almost any kind of content, but it has to follow a certain structure for it to be properly deployed by JBoss ON. The recipe is an Ant build file called **`deploy.xml`**; this must always be located in the top level of the bundle archive.

Past the placement of the recipe, the files and directories within the bundle can be located anywhere in the archive. In fact, the files do not necessarily need to be included in the bundle file at all; when the bundle is created, any or all files for the bundle can be pulled off a URL, which allows the content to be taken from an SVN or GIT repository, FTP server, or website.

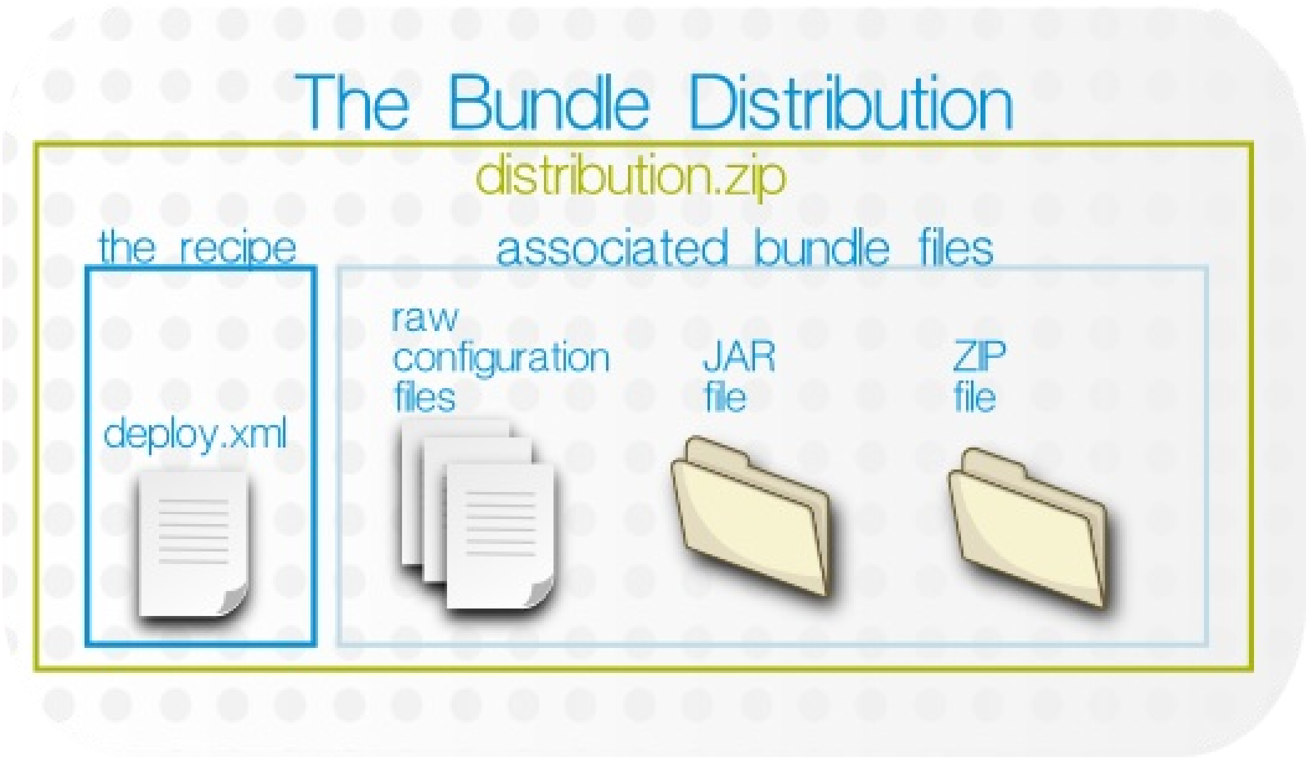


Figure 27.1. Bundle Layout

The bundle archive can contain other archives, such as JAR, WAR, and ZIP files. Provisioning uses Ant to build out bundles on the target machine, so any files which Ant can process can be processed as part of the bundle. The Ant provisioning system can process WAR, JAR, and ZIP archive files.

27.1.2. Destinations (and Bundle Deployments)

Uploading a bundle to JBoss ON does not push the bundle anywhere, so it is not automatically associated with a resource or group. (Bundles, unlike content, is resource-independent. It exists as its own definition in JBoss ON, apart from the inventory.) When the bundle is actually provisioned, then the provisioning wizard prompts for the administrator to define the *destination*.

A destination is the place where bundles get deployed. The destination is the combination of three elements:

- A compatible resource group (of either platforms or JBoss servers)
- A *base location*, which is the root directory to use to deploy the bundle. Resource plug-ins define a base location for that specific resource type in the **`<bundle-target>`** element. This can be the root directory or, for JBoss servers, common directories like the profile directory. There may be multiple available base locations.
- The *deployment directory*, which is a subdirectory beneath the base directory where the bundle content is actually sent.

For example, an administrator wants to deploy a web application to a JBoss EAP 5 server, in the **`deploy/myApp/`** directory. The JBoss AS5 plug-in defines two possible base locations, one for the installation directory and one for the profile directory. The administrator chooses the profile directory, since the application is an exploded JAR file. The agent then derives the real, absolute path of the application from those three elements:

```
JBoss AS group + {$PROFILE_DIR} + deploy/myApp/
```

If the *PROFILE_DIR* is **`/opt/jbossas/default/server/`**, then the destination is:

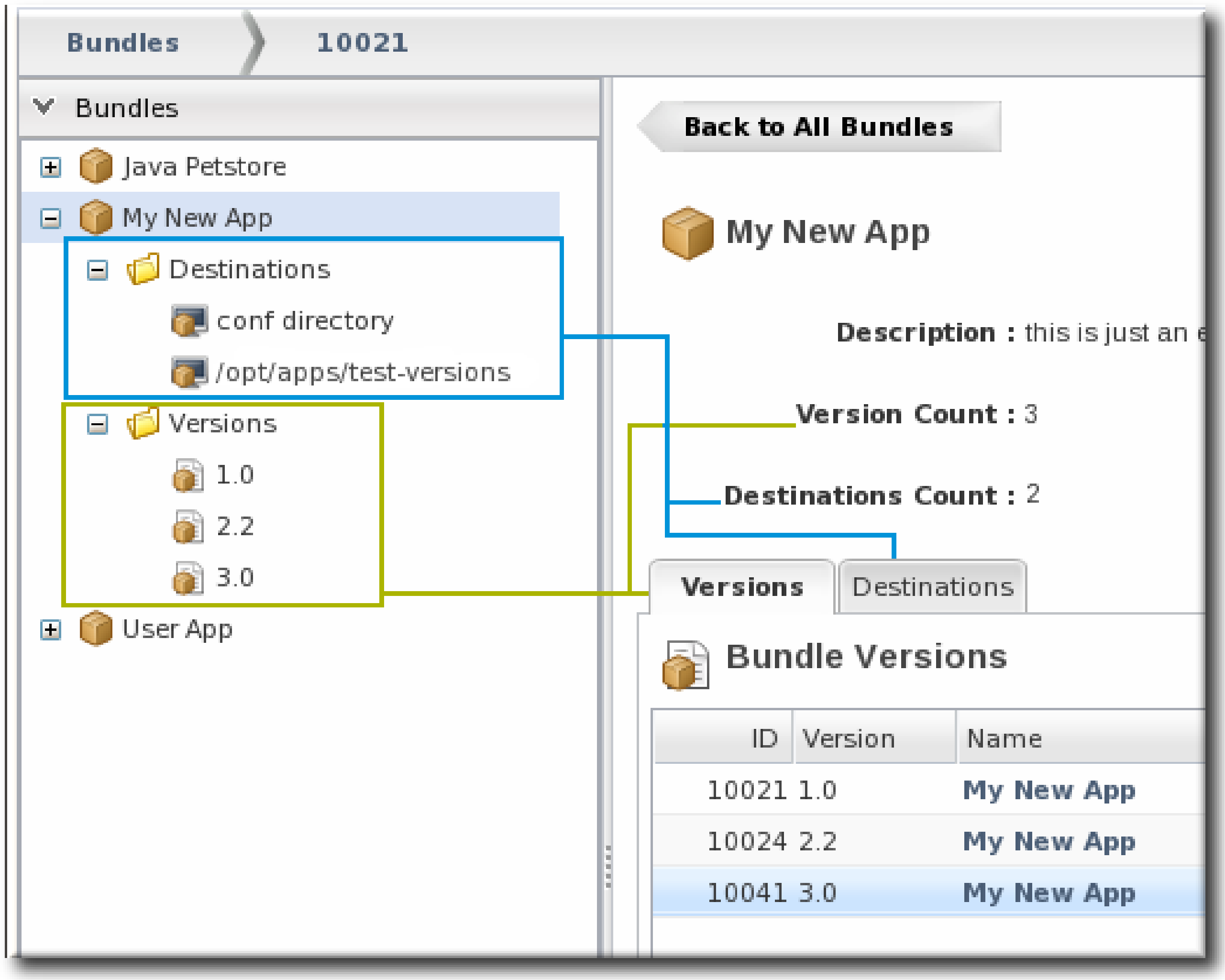
```
/opt/jbossas/default/server/deploy/myApp/
```

If the same resource group contains a JBoss EAP instance running on a Windows server, with a *PROFILE_DIR* of **`C:\jbossas\server\`**, then the path is derived slightly differently, appropriate for the platform:

```
C:\jbossas\default\server\deploy\myApp
```

It is up to the agent, based on the platform and resource information for its inventory, to determine the absolute path for the destination to which the bundle should be deployed.

Once a bundle is actually deployed to a destination, then that association — bundle version and destination — is the *bundle deployment*.



Description : this is just an

Version Count : 3

Destinations Count : 2

ID	Version	Name
10021	1.0	My New App
10024	2.2	My New App
10041	3.0	My New App

Figure 27.2. Bundles, Versions, and Destinations

27.1.3. File Handling During Provisioning

Behavior at Deployment

A bundle file contains a set of files and directories that should be pushed to a resource. However, the provisioning process does not merely copy the files over to the deployment directory. Provisioning treats a bundle as, essentially, a template that defines the entire content structure for the target (root) directory.

For example, a bundle contains these files:

```
app.conf
lib/myapp.jar
```

If the root directory is **`deploy/myApp/`**, then the final configuration is going to be:

```
deploy/myApp/app.conf
deploy/myApp/lib/myapp.jar
```

By default, if there are any files in **`deploy/myApp/`**, then they will be removed before the bundle is copied over, so that the deployment directory looks exactly the way the bundle is configured.

For an application-specific destination, like **`deploy/myApp/`**, then that behavior is totally acceptable because the defined application content should be the only content in that directory. However, bundles can contain a variety of content and can be deployed almost anywhere on a platform or within a JBoss server. In a lot of real life infrastructures, the directory where the bundle is deployed may have existing data that should be preserved.

The deployment directory is the *root directory* for the bundle. The recipe for the bundle can define a parameter that tells the provisioning process how to handle data in that root directory — specifically, should it ignore (preserve) existing files and subdirectories in the root directory, or should it *manage* the root directory and make it match the bundle structure.

The *compliance* option in the recipe tells the provisioning system whether to delete everything and force the directory to match the bundle content. The default is for that to be *full*, for the bundle to define the content and structure of the root directory. If the data in that directory must be saved, the **`compliance`** option can be set to *filesAndDirectories*, which means that provisioning will copy over the bundle and create the appropriate files and subdirectories, but it will not manage (remove) the existing content in the directory.

WARNING

If you deploy a bundle to a high level directory like **`/etc`** on a platform or a critical directory like **`deploy/`** or **`conf/`**, then all of the existing files and subdirectories in that directory are deleted. This can cause performance problems or data loss.

Bundles and Subdirectories

Even if the **`compliance`** option is set to *filesAndDirectories*, subdirectories and files contained in the bundle are always managed by the bundle, even if they existed before the bundle was deployed.

For example, the **`deploy/`** directory has this layout before any bundle is deployed:

```
deploy/
deploy/notes.txt
deploy/myApp1/
deploy/myApp2/
deploy/myApp2/first.txt
```

A bundle is created with this layout:

```
myApp2/
myApp2/foo.txt
myApp2/bar.txt
```

If **`compliance`** is set to *filesAdDirectories*, the existing files in the **`deploy/`** remain untouched, *except* for what is in the **`myApp2/`** subdirectory, because that directory is defined by the bundle. So, the final directory layout is this:

```
deploy/ (ignored)
deploy/notes.txt (ignored)
deploy/myApp1/ (ignored)
deploy/myApp2/ (managed)
myApp2/foo.txt (managed)
myApp2/bar.txt (managed)
```

NOTE

Any existing content in the root directory is backed up before it is deleted, so it can be restored later.

The backup directory is **`/home/.rhqdeployments/`***resourceID***`/backup`**.

Files Added After the Bundle Is Deployed

After the initial deployment, there can be instances where files are added to the deployment directory, such as log files or additional data.

Within the deployment directory, the provisioning process overwrites any bundle-associated files with the latest version and removes any files that are not part of the bundle. Log files and other data — as with the root directory — need to be preserved between upgrades. Those known files and directories can be called out in the recipe using the **`<rhq:ignore>`** element, which tells the provisioning process to ignore those files *within* the deployment directory.

Setting these options in the recipe is described in Section 27.5.7.3, "WARNING: The Managed (Target) Directory and Overwriting or Saving Files".

27.1.4. Requirements and Resource Types

By default, three resource types support bundles:

- Platforms, all types
- JBoss EAP 6 (AS 7) standalone servers[5]
- JBoss EAP 5 and any server which uses the JBoss AS 5 plug-in
- JBoss EAP 4 (deprecated)

Bundle support is defined in the plug-in descriptor, so custom plug-ins can be created that add bundle support for those resource types. For examples of writing agent plug-ins with bundle support, see *Writing Custom JBoss ON Plug-ins*

27.1.5. Provisioning and Agent User System Permission

When a bundle is deployed, the provisioning process can write the bundle files to whatever directory on the system is specified, so long as the system user as which the agent is running has permissions to write to that directory.

As discussed in Chapter 4, *Interactions with System Users for Agents and Resources*, the system user which the agent uses is critical to the overall performance of JBoss ON. One the one hand, the system user must be joined to the appropriate groups and granted appropriate write access to manage resources effectively. On the other hand, especially when dealing with system content and configuration through bundles, it is critical that the agent user not be given excessive permissions, or the provisioning process itself could be used maliciously.

Whatever permissions the agent user has, the provisioning system and the deployed bundle can use. Every deployed bundle has the full system rights of the agent user.

Evaluating the required system permissions is critical when an agent is installed. It is also critical to evaluate what rights the agent user has and to plan the deployment path of a bundle — and even the content of the bundle itself — accordingly to prevent the deployed application or configuration files from making undesirable system changes or being used maliciously.

27.1.6. Provisioning and Roles

There are a set of bundle group-level and global permissions that are required for different management points for a bundle. The appropriate permissions must be granted to a role for a user to be able to manage bundles and deploy applications.

Bundle management breaks down into two types of tasks: creating bundle versions and deploying bundles to resources. The workflow for each of those processes is very much defined by the organizational structure, development process, and production rules. For example, in some environments, it may be permissible for anyone to create and delete new bundles and to deploy content to any resources which they have access to. In other organizations, it may be necessary to restrict bundle management to specific users or to divide tasks so that one group creates bundles while another user group deploys them.

The bundle permissions are listed in Table 9.1, “JBoss ON Access Control Definitions” , while a general example for planning permissions for different groups is covered in Section 9.4, “*Extended Example: View All Resources, Edit Some Resources*”.

The critical thing is to create a set of roles which parallel and reinforce the desired bundle deployment workflow — and this means that the workflow must be clearly defined and understood. The different parts of the provisioning process encompass two main relationships. Creating bundles (uploading them to the JBoss ON database) defines a bundle/user relationship. Deploying bundles (copying the content over to a resource) defines a bundle/resource relationship. The defined roles and permissions should reflect that.

Much like resources, bundles are defined in roles first by being members of *bundle groups*, and then the group itself is added to a role. By default, all bundles must be added to at least one group when they are created; they cannot be viewed and deployed until they belong to a group which belongs to a role with appropriate permissions to access it. There is one exception: users can be granted a global *view bundles* permission; this allows that user to create a bundle that is not in any group. The bundle remains in a holding state called *unassigned*, and only users with the *view bundles* permission can see it (which means, only they can add it to a group, assuming they have the appropriate permission). Creating bundle groups is covered in Section 27.7.1, "Managing Bundle Groups" .

Section 27.8, "*Extended Example*: Using Bundle Groups and Access Control within the Provisioning Process" covers some basic permissions models, defining different potential bundle/user and bundle/resource relationships.

27.1.7. Space Considerations for Bundles

Bundles can have a significant impact on disk space requirements.

JBoss ON stores all versions of content. This is part of versioning control, allowing changes to bundles to be reverted and managed and for different versions to be deployed at different times.

Therefore, the system which hosts the backend database (Oracle or PostgreSQL) must have enough disk space to store all versions of all bundles. Additionally, the database itself must have adequate tablespace for the content.

When calculating the required amount of space, estimate the size of every artifact, and then the number of versions for each artifact. At a minimum, **have twice that amount of space available** ; both PostgreSQL and Oracle require twice the database size to perform cleanup operations like vacuum, compression, and backup and recovery.

27.1.8. Bundles and JBoss ON Server and Agent Plug-ins

27.1.8.1. Resource Support and the Agent Resource Plug-in

Whether provisioning is supported is defined in the resource type. For a resource type to allow provisioning, the resource plug-in descriptor must define a *bundle target*. That is the indication to the agent the provisioning is supported.

The **`<bundle-target>`** element simply defines allowed base directories for the resource which can be used as base directories in the bundle definition.

```
<server name="JBossAS:JBossAS Server" ...>
   <bundle-target>
      <destination-base-dir name="Library Directory" description="Where
the jar libraries are">
         <value-context>pluginConfiguration</value-context>
         <value-name>lib.dir</value-name>
      </destination-base-dir>
      <destination-base-dir name="Deploy Directory" description="Where the
deployments are">
         <value-context>pluginConfiguration</value-context>
         <value-name>deploy.dir</value-name>
```

```
        </destination-base-dir>
    </bundle-target>
</server>
```

Every resource plug-in descriptor defines a *base directory*, the root for all deployments, apart from provisioning configuration. For platforms, this is the root directory. For servers, it is usually the installation directory. The **`<bundle-target>`** can use the already-configured base directory or it can set different directories to use. In the example, two directories — the **`deploy/`** and **`lib/`** directories — are given as supported base directories. When a bundle definition is created, the wizard offers the choice of which directory to use.

27.1.8.2. Server-Side and Agent Plug-ins for Recipe Types

By default, JBoss ON supports one type of recipe, an Ant build file. However, other types of recipes could be supported because the recipe type is defined in a pair of plug-ins, one for the server and one for the agent.

The server-side plug-in tells the JBoss ON server how to manage bundles and destinations for that type of recipe.

The agent plug-in creates a child resource for the platform which is used to perform provisioning operations on the platform or target resource. For example, Ant bundles are actually deployed by the special JBoss ON resource, *Ant Bundle Handler*. This resource is added automatically to platforms as a child resource to enable Ant-based provisioning.

NOTE

Since recipe type support is implemented on the agent side through a special resource, that resource must exist in the JBoss ON inventory for it to perform provisioning. For example, without the Ant bundle handler in the inventory for a platform, JBoss ON cannot perform provisioning on that platform.

Administrators do not have to interact directly with the Ant bundle handler resource, but that child resource must be present in the platform's inventory for Ant provisioning to work.

27.1.9. Managing and Deploying Bundles with the JBoss ON CLI

Both uploading bundles to JBoss ON and deploying bundles to resources can be performed using the JBoss ON CLI.

The ability to script bundle deployments is very powerful, because it allows content or configuration updates, even new application servers, to be deployed automatically based on activity in other resources across JBoss ON. This is particularly useful with using JBoss ON CLI scripts in response to an alert:

- A new JBoss application server can be deployed when an existing JBoss server experiences a heavy load or decreased performance.
- Configuration files for a selected snapshot image can be immediately deployed to a platform or JBoss server to remedy configuration drift, in response to a drift alert.
- A new web context can be deployed when another web is disabled within a **`mod_cluster`** domain.

Scripting also allows updates to be applied on schedule, such as having daily or weekly scheduled updates to a QE environment — which is also useful because the bundle content can be pulled from a GIT or SVN repository used by a build system first, and then deployed for testing.

The bundles API is in the Javadocs at http://docs.redhat.com/docs/en-US/JBoss_Operations_Network/100/html/API_Guides/index.html.

27.2. *EXTENDED EXAMPLE*: COMMON PROVISIONING USE CASES (AND HOW THEY HANDLE FILES)

Section 27.1, "An Introduction to Provisioning Content Bundles" describes the different elements in the bundles and provisioning system — but what is provisioning really for? There are a handful of common use cases that illustrate how provisioning works in a real environment.

- Deploying a full application server
- Deploying a web application to an application server
- Deploying configuration files

IMPORTANT

One thing to remember for all of this is that bundles take content from point A (JBoss ON) and send it to point B (the destination). This is not a simple copy operation (Section 27.1.3, "File Handling During Provisioning"). The provisioning system, by default, makes the destination, point B, match exactly how the directory in the bundle is laid out. This means adding or editing files, creating or deleting subdirectories, and deleting any existing content from the destination.

That concept of managing the layout and content of the destination influences the ways and effects of deploying content.

27.2.1. Deploying A Full Application Server

Bundle Details

This is the core use for the provisioning system, deploying an entire application server. This bundle contains the complete configuration stack for a server like JBoss EAP (or Tomcat or Apache). The bundle contains all of the files used by the application — the JAR and configuration files and scripts for the EAP server, and all EAR or WAR web applications that are to be deployed on that EPA instance. All of the application server and web app files and directories are zipped into an archive, with the **`deploy.xml`** which defines the Ant recipe.

File Handling Details

Because this is a complete application server, it will be deployed into its own, new (or empty) directory, such as **`/opt/my-application`**. That directory will be dedicated to the application server, so it will be entirely *managed* by the bundle.

There is an attribute in the recipe called *compliance* which tells the bundle system how to deploy content. For deploying a full application server, the bundle system should have complete control over the directory, so **`compliance`** is set to *full*. This means:

- Only files and subdirectories in the bundle distribution file will be in the root directory.

- Any existing files or subdirectories will be deleted.
- If files or subdirectories are added to the root directory, then they will be deleted when the bundle is updated or redeployed, unless those files are explicitly ignored (a setting in the recipe).

27.2.2. Deploying A Web Application

Instead of deploying a full application server, it is possible to deploy a web application to the application server. However, this takes an understanding of the directory layout of both the application server and the web application.

For example, this is the deployment directory path for the application server:

```
/opt/my-application/deploy
```

The goal is to deploy a new web app, **`myApp1.war`** to the **`deploy/`** directory.

```
/opt/my-application/deploy/myapp1.war/
```

Bundle Details

In this case, the bundle file contains only the WAR file itself and the **`deploy.xml`** recipe.

File Handling Details

Unlike the application server, when deploying the web app, there are or could be other web apps also in the **`deploy/`** directory. The bundle system, then, should not manage the root directory, meaning existing or new files should be allowed within the root directory even if they are not included in the bundle.

The intent here is not to manage all of the content in the directory but to add to that content by deploying the WAR file. So, the recipe should specify **`compliance=fileAndDirectories`**, which tells the provisioning system to leave alone any existing files in the directory that are outside the bundle.

NOTE

Setting **`compliance=filesAndDirectories`** only preserves files outside the bundle deployment. If the bundle directory is **`deploy/myApp/`**, then any files in **`deploy/myApp/`** or subdirectories like **`deploy/myApp/WEB-INF/`** will be overwritten or removed when the bundle is deployed. The subdirectories defined *in the bundle distribution* are still entirely managed by the bundle system.

One other consideration is that only one bundle can be deployed to a root directory. If there are multiple web applications to be deployed to the same EAP server and all of them will be managed by the provisioning system, then there are two options:

- Include all of the web applications in the same bundle distribution. For example, to deploy **`myApp1.war`** and **`myApp2.war`** to the **`deploy/`** directory, both WAR files can be included in the same bundle and deployed to **`deploy/myApp1.war/`** and **`deploy/myApp2.war/`** simultaneously.
- It may not be possible to roll all of the web apps into the same bundle. Instead of using **`deploy/`** as the root directory, there could be two different bundle distributions that use a subdirectory as the root directory. For example, the first web app could use **`deploy/myApp1/`**

so that the final deployment is `deploy/myApp1/myApp1.war/`, while the second app uses `deploy/myApp2/`, resulting in `deploy/myApp2/myApp2.war/`.

This allows the two web applications to be deployed, updated, and reverted independently of each other.

27.2.3. Deploying Configuration Files

Another common scenario is deploying configuration files to an application server (or even another resource, like a platform), using bundles.

This is very similar to deploying a web application. **If JBoss ON deploys a bundle to a given directory, it expects to manage the content within that directory and all content within any subdirectories defined in the bundle.** With configuration files, it is critical that you understand and include *all* of the configuration files in the bundle or critical files could be removed.

For example, an administrator creates a bundle to deploy two configuration files:

- New login configuration, in `server/default/conf/login-config/xml`
- New JMX console users, in `server/default/conf/props/jmx-console.properties`

The root directory is the `conf/` directory for the application server.

Bundle Details

The bundle must contain *all* of the files that are expected to be in the `conf/login-config/` and `conf/props/` subdirectories, not just the two new files that the administrator wants to use. Additionally, the `compliance` parameter in the recipe must be set to *filesAndDirectories* so that all of the existing configuration files in the root directory, `conf/`, are preserved.

File Handling Details

As with deploying a web app, **the intent is to add new content, not to replace existing content.** Setting `compliance=filesAndDirectories` only preserves files outside the bundle deployment. However, because there are two subdirectories defined in the bundle, JBoss ON will manage all of the content in those subdirectories. This means:

- The recipe has to have `compliance=filesAndDirectories` set for the bundle to preserve the other files in the `conf/` root directory.
- Any files in the subdirectories of the bundle — `conf/log-config/` and `conf/props/` — will be overwritten when the bundle is deployed. The provisioning process can ignore files in the root directory, but it always manages (meaning, updates, adds, or deletes) files in any subdirectories identified in the bundle so that they match the content of the bundle.
- Existing files in the `conf/log-config/` and `conf/props/` subdirectories must be included in the bundle distribution.

NOTE

There is an alternative to including all of the configuration files in the bundle distribution.

The bundle, containing only the new files, could be deployed to a separate directory, like **`/opt/bundle/`**. Then, an Ant post-install task can be defined in the recipe that copies the configuration files from the root directory into the application server's **`conf/`** directory.

27.3. *EXTENDED EXAMPLE*: PROVISIONING APPLICATIONS TO A JBOSS EAP SERVER (PLANNING)

The Setup

Tim the IT Guy at Example Co. has to manage the full application lifecycle for Example Music's online band management application, MusicApp. There are two environments: one for QA and one for the live site. Both environments contain a mix of Windows and Linux servers.

Tim wants to deploy the latest development version weekly to the QA environment, based on the most current build in their development GIT repo. He wants the most stable version of the application to be deployed to the production environment, based on a static package.

What to Do

The best plan for Tim is to work backwards, starting with the way he wants his ideal QA and production environments to be configured.

Tim's first step is to identify his destinations, based on his environments. Because he has two separate environments, he wants to create two separate groups, one for QA and one for production. MusicApp runs on a JBoss server, so his compatible groups will be for the JBoss AS/EAP resources rather than platforms.

Additionally, the needs for each of his environments is different:

- The QA environment needs ...
    - New builds directly from the GIT repository, every week.
    - A completely clean directory to begin from with every deployment.
    - There is a separate QA environment for each of Example Co.'s web applications, so MusicApp is the only application running on those specific servers.
- The production environment needs ...
    - A stable build that can be safely stored in JBoss ON.
    - To save historic data. The production environment has both log directories and user-supplied data directories that need to be preserved between application upgrades.
    - A couple of different web applications run on the same production servers.

The application itself is the same for both environments. Instance-specific configuration — port numbers, the application name, the machine IP address — are based on tokens that are realized when the application is deployed. The JAR files contained in the bundle should be extracted at the time the application is deployed, with the exception of one client which site members can either install or launch locally.

Tim decides to use different versions of the same bundle, labeling the QA versions as *devel* and the production versions as *stable*.

There are some similarities between the devel and stable bundle recipes:

- MusicApp should be deployed to the **`deploy/`** directory, but because it is not the only application that they run, it will have its own webapp context subdirectory. While this is not strictly necessary in the devel environment (where MusicApp is the only application), this maintains consistency with the final deployment destination.
- Both recipes will configure the application JAR file, **`MusicApp.jar`**, to be exploded when it is deployed.
- The client archive file, **`MyMusic.jar`**, will not be exploded (**`<rhq:file ... exploded="false">`**).
- Tokens are defined in the raw configuration files and the recipe for the port numbers, IP addresses, and application names.

And then there are differences in the recipes, related to how the devel and stable versions should handle existing files.

- The QA environment always requires a pristine deployment. This requires three settings:
  - The **`compliance`** value is always *full*, so no existing files are preserved during the initial deployment.
  - No **`<rhq:ignore>`** elements are set, so no generated files are preserved during an upgrade.
  - The **`cleanDeployment`** option is always set in the JBoss ON CLI script that automates deployments. This removes all bundle-associated files in the directory before deploying the new bundle.
- The production environment needs to preserve its existing data between upgrades, which requires two settings:
  - The **`compliance`** value is always *filesAndDirectories*, which preserves existing files during the initial deployment.
  - Two **`<rhq:ignore>`** elements are set, one for the log directory and one for the data directory containing the site member uploads.

The last significant action comes when the bundles are actually uploaded to JBoss ON.

Version 1 of the application is already stable and complete, so Tim creates the first bundle as a stable version. He packages the **`deploy.xml`** with the other application files in a ZIP file and uploads the entire bundle directly to JBoss ON, so it is stored in the JBoss ON database.

Version 2 is a devel version. The QA environment requires frequent updates based on the latest build in GIT. Tim uploads the **`deploy.xml`** separately, but he points the provisioning wizard to the GIT URL for all of the associated packages. When the bundle is deployed, JBoss ON takes the packages from the repository.

The Results

Tim deployed version 1 of the bundle to the production environment, and he deployed version 2 to the QA environment.

This means that Tim has deployed different versions of the same application, pulled from different sources, to different resources. If he ever has a problem with the production server, he can simply revert it to the last stable version.

Additionally, he can script bundle deployments to the QA environment, so his tests can be fully automated.

27.4. THE WORKFLOW FOR CREATING AND DEPLOYING A BUNDLE

1. Identify what files belong in the archive, such as an application server, an individual web application, configuration files for drift management, or other things.

2. Determine how the location where the bundle will be deployed will be handled. Existing files and directories can be overwritten or preserved, depending on the definitions in the recipe. This is covered in Section 27.1.3, “File Handling During Provisioning” and Section 27.5.7.3, “WARNING: The Managed (Target) Directory and Overwriting or Saving Files”.

3. Identify what information will be deployment-specific, such as whether the deployed content will require a port number, hostname, or other setting. Some of these values can use tokens in the configuration file and the provisioning process can interactively prompt for the specific value at deployment time.

   Tokens are covered in Section 27.5.5, “Using Templatized Configuration Files” .

4. Create the content which will be deployed.

5. Create an Ant recipe, named **`deploy.xml`**. The recipe defines what content and configuration files are part of the bundle and how that content should be deployed on the bundle destination. Pre- and post-install targets are supported, so there can be additional processing on the local system to configure or start services as required.

   Ant recipes and tasks are covered in Section 27.5.3, “Breakdown of an Ant Recipe” and Section 27.5.4, “Using Ant Tasks”.

6. After the bundle content, configuration file, and recipe are created, compress all of those files into a bundle archive. This must have the **`deploy.xml`** recipe file in the top level of the directory and then the other files in the distribution, relative to that **`deploy.xml`** file. This is illustrated in Section 27.1.1, “Bundles: Content and Recipes”.

   NOTE

   JBoss ON allows JAR and ZIP formats for the bundle archive.

7. Optionally, verify that the bundle is correctly formatted by running the bundle deployer tool. This is covered in Section 27.6, “Testing Bundle Packages”.

8. Create the groups of resources to which to deploy the bundle.

9. Upload the bundle to the JBoss ON server, as described in Section 27.7.2, “Uploading Bundles to JBoss ON”.

10. Deploy the bundle to the target groups, as described in Section 27.7.3, “Deploying Bundles to a Resource”.

27.5. CREATING ANT BUNDLES

Bundles are archive files that is stored on the server and then downloaded by an agent to deploy to a platform or resource. A bundle distribution is comprised of two elements:

- An Ant recipe file named **`deploy.xml`**
- Any associated application files. These application files can be anything; commonly, they fall into two categories:
  - Archive files (JAR or ZIP files)
  - Raw text configuration files, which can include tokens where users define the values when the bundle is deployed

NOTE

The process and guidelines for actually creating an Ant recipe are outside the scope of this documentation. This document outlines the options and requirements for using Ant recipes *specifically* to work with the JBoss ON provisioning system.

For basic instructions, options, and tutorials for writing Ant tasks, see the Apache Ant documentation at http://ant.apache.org/manual/index.html.

27.5.1. Supported Ant Versions

Table 27.1, “Ant Versions” lists the supported Ant and Ant Task versions.

Table 27.1. Ant Versions

Software	Version
Ant	• 1.8.0
Ant-Contrib	• 1.0b3

27.5.2. Additional Ant References

Provisioning relies on Ant configuration and tasks, so a good understanding of the Ant build process is beneficial. There are several resources for additional Ant information:

- Apache Ant documentation main page
- Apache Ant documentation for the build file
- Liquibase Database Schema Tasks
- Ant Contrib Tasks

27.5.3. Breakdown of an Ant Recipe

The Ant recipe for JBoss ON bundles is the same basic file as a standard Apache Ant file and is processed by an integrated Ant build system in JBoss ON. This Ant recipe file must be bundled in the top directory of the distribution ZIP file and be named **`deploy.xml`**.

The JBoss ON Ant recipes allows all of the standard tasks that are available for Ant builds, which provides flexibility in scripting a deployment for a complex application. The JBoss ON Ant recipe *must* also provide additional information about the deployment that will be used by the provisioning process; this includes information about the destination and, essentially, metadata about the application itself.

Example 27.1. Simple Ant Recipe

For provisioning, the Ant recipe is more of a definition file than a true script file, although it can call Ant targets and do pre- and post-provisioning operations. As with other Ant scripts, the JBoss ON Ant recipe uses a standard XML file with a **`<project>`** root element and defined targets and tasks. The elements defined in the **`<rhq:bundle>`** area pass metadata to the JBoss ON provisioning system when the project is built.

The first part of the **`deploy.xml`** file simply identifies the file as an an script and references the provisioning Ant elements.

```
<?xml version="1.0"?>
<project name="test-bundle" default="main"
 xmlns:rhq="antlib:org.rhq.bundle">
```

The next element identifies the specific bundle file itself. The provisioning system can manage and deploy multiple versions of the same application; the **`<rhq:bundle>`** element contains information about the specific version of the bundle (including, naturally enough, an optional version number).

```
    <rhq:bundle name="Example App" version="2.4" description="an example
bundle">
```

All that is *required* for a recipe is the **`<rhq:bundle>`** element that defines the name of the application. However, the bundle element contains all of the information about the application and, importantly, how the provisioning system should handle content contained in the application.

The first item to address is any *templatized property* that is used in a configuration file. This is covered in Section 27.5.5, “Using Templatized Configuration Files” . Any token used in a configuration file must be defined in the recipe for it to be realized (to have a value supplied) during provisioning. For the ***`port`*** token defined in Section 27.5.5, “Using Templatized Configuration Files” , the **`<rhq:input-property>`** element identifies it in the recipe. The ***`name`*** argument is the *input_field* value in the token, the ***`description`*** argument gives the field description used in the UI and the other arguments set whether the value is required, what its allowed syntax is, and any default values to supply. (This doesn't list the files which use tokens, only the tokens themselves.)

```
 <rhq:input-property
            name="listener.port"
            description="This is where the product will listen for
incoming messages"
            required="true"
            defaultValue="8080"
            type="integer"/>
```

There is a single element which identifies all of the content deployed by the bundle, the

`<rhq:deployment-unit>` element. The entire application — its name, included ZIP or JAR files, configuration files, Ant targets — are all defined in the **`<rhq:deployment-unit>`** parent element.

The name and any Ant targets are defined as arguments on **`<rhq:deployment-unit>`** directly. In this, the name is **`appserver`**, and one preinstall target and one postinstall target are set.

```
        <rhq:deployment-unit name="appserver"
preinstallTarget="preinstall" postinstallTarget="postinstall"
compliance="filesAndDirectories">
```

There is one other critical element on the **`<rhq:deployment-unit>`** element: the **`compliance`** argument. Provisioning doesn't simply copy over files; as described in Section 27.1.3, “File Handling During Provisioning”, it remakes the directory to match what is in the bundle. If there are any existing files in the deployment directory when the bundle is first deployed, they are deleted by default. Setting **`compliance`** to *filesAndDirectories* means that the provisioning process does *not* manage the deployment directory — meaning any existing files are left alone when the bundle is deployed.

Any existing content in the root directory is backed up before it is deleted, so it can be restored later. The backup directory is **`/home/.rhqdeployments/`***resourceID***`/backup`**.

Any configuration file is identified in an **`<rhq:file>`** element. The ***name*** is the name of the configuration file *within the bundle*, while the **`destinationFile`** is the relative (to the deployment directory) path and filename of the file after it is deployed.

```
 <rhq:file name="test-v2.properties"
destinationFile="conf/test.properties" replace="true"/>
```

Bundles can contain archive files, either ZIP or JAR files. Every archive file is identified in an **`<rhq:archive>`** element within the deployment-unit. The **`<rhq:archive>`** element does three things:

- Identify the archive file by name.
- Define how to handle the archive. Simply put, it sets whether to copy the archive over to the destination and then leave it as-is, still as an archive, or whether to extract the archive once it is deployed. This is called *exploding* the archive. If an archive is exploded, then a postinstall task can be called to move or edit files, as necessary.
- Identify any files within the archive which contain tokens that need to be realized. This is a child element, **`<rhq:fileset>`**. This can use wildcards to include types of files or files within subdirectories or it can explicitly state which files to process.

```
            <rhq:archive name="MyApp.zip" exploded="true">
                <rhq:replace>
                    <rhq:fileset>
                        <include name="**/*.properties"/>
                    </rhq:fileset>
                </rhq:replace>
            </rhq:archive>
```

Another possible child element sets how to handle any files within the deployment directory that are not part of the bundle. For example, the application may generate log files or it may allow users to upload content. By default, the provisioning process cleans out a directory from non-bundle content every time a bundle is provisioned. However, logs, user-supplied data, and other types of files are data that should remain intact after provisioning. Any files or subdirectories which should be ignored by the provisioning process (and therefore preserved) are identified in the **`<rhq:ignore>`** element. In this case, any **`*.log`** files within the **`logs/`** directory are saved.

```
            <rhq:ignore>
                <rhq:fileset>
                    <include name="logs/*.log"/>
                </rhq:fileset>
            </rhq:ignore>
        </rhq:deployment-unit>
    </rhq:bundle>
```

This only applies to upgrading a bundle, meaning *after* the initial deployment.

The last elements set the Ant tasks to run before or after deploying the content, as identified initially in the **`<rhq:deployment-unit>`** arguments. Most common Ant tasks are supported (as described in Section 27.5.4, "Using Ant Tasks"). This uses a preinstall task to print which directory the bundle is being deployed to and whether the operation was successful. The postinstall task prints a message when the deployment is complete.

```
<target name="main" />

    <target name="preinstall">
        <echo>Deploying Test Bundle v2.4 to ${rhq.deploy.dir}...</echo>
 <property name="preinstallTargetExecuted" value="true"/>
 <rhq:audit status="SUCCESS" action="Preinstall Notice"
info="Preinstalling to ${rhq.deploy.dir}" message="Another optional
message">
  Some additional, optional details regarding
  the deployment of ${rhq.deploy.dir}
 </rhq:audit>
    </target>

    <target name="postinstall">
        <echo>Done deploying Test Bundle v2.4 to
${rhq.deploy.dir}.</echo>
        <property name="postinstallTargetExecuted" value="true"/>
    </target>
</project>
```

Section 27.5.8, "A Reference of JBoss ON Ant Recipe Elements" lists the different JBoss ON elements in the Ant recipe file. For information on standard Ant tasks, see the Apache Ant documentation.

27.5.4. Using Ant Tasks

An Ant bundle distribution file is just an Ant recipe and its associated files. As Example 27.1, "Simple Ant Recipe" shows, the Ant recipe is the expected **`deploy.xml`** file with some JBoss ON-specific elements. An Ant bundle distribution file supports more complex Ant configuration, including Ant tasks and targets.

27.5.4.1. Supported Ant Tasks

Any standard Ant task can be run as part of the Ant bundle provisioning (with the exception of **<antcall>** and **<macrodef>**). This includes common commands like **echo**, **mkdir**, and **touch** – whatever is required to deploy the content fully.

IMPORTANT

The **<antcall>** element *cannot* be used with the Ant recipe. **<antcall>** calls a target within the **deploy.xml** file, which loops back to the file, which calls the **<antcall>** task again, which calls the **deploy.xml** file again. This creates an infinite loop.

To perform the same operations that would be done with **<antcall>**, use the **<ant>** task to reference a separate XML file which contains the custom Ant targets. This is described in Section 27.5.4.3, "Calling Ant Targets".

IMPORTANT

The **macrodef** call, and therefore macro definitions, are not supported with Ant bundles.

Along with the standard Ant tasks, Ant bundle recipes can use optional Ant tasks:

- Liquibase Database Schema Tasks
- Ant Contrib Tasks

27.5.4.2. Using Default, Pre-Install, and Post-Install Targets

As with other Ant tasks, the **<project>** allows a default target, which is required by the provisioning system. This is a no-op because the Ant recipe mainly defines the metadata for and identifies files used by the provisioning process. Other operations aren't necessary. This target is required by Ant, even though it is a no-op target. Use pre- and post-install targets to perform tasks with the bundle before and after it is unpacked.

For example:

```
<target name="main" />
```

Additionally, JBoss ON provisioning tasks can define both pre- and post-install targets. This allows custom tasks, like simple progress messages or setting properties.

27.5.4.3. Calling Ant Targets

As mentioned in Section 27.5.4.1, "Supported Ant Tasks", using **<antcall>** does not work in an Ant bundle recipe; it self-referentially calls the **<rhq:bundle>** task in an infinite loop. However, it is possible to process tasks that are *outside* the default target. This can be done using pre- and post install targets (Section 27.5.4.2, "Using Default, Pre-Install, and Post-Install Targets").

1. In **deploy.xml** for the Ant recipe, add a **<rhq:deployment-unit>** element which identifies the Ant target.

   ```
   <rhq:deployment-unit name="jar" postinstallTarget="myExampleCall">
   ```

2. Then, define the target.

```
    <target name="myExampleCall">
       <ant antfile="another.xml" target="doSomething">
          <property name="param1" value="111"></property>
       </ant>
    </target>
```

3. Create a separate **another.xml** file in the same directory as the **deploy.xml** file. This file contains the Ant task.

```
<?xml version="1.0"?>
<project name="another" default="main">
    <target name="doSomething">
       <echo>inside doSomething. param1=${param1}</echo>
    </target>
</project>
```

27.5.5. Using Templatized Configuration Files

A bundle can contain configuration files for an application. These configuration files can use hard-coded values or they can use *tokens* that are filled in (automatically or with user-supplied values) when the bundle is actually deployed.

NOTE

For a user-defined token to be realized, it must be referenced in the recipe so that the bundle deployment wizard will prompt for the value, using the **<rhq:input-property>** key in the Ant recipe. For examples, see Section 27.5.8.2, “rhq:input-property” and Example 27.1, “Simple Ant Recipe”.

User-defined tokens can be any property; the values are supplied through the provisioning UI and inserted into the templatized configuration file.

The token key is a simple attribute-value assertion, with the *input_field* as the element in the UI and the *property* being the value in the configuration file. The property of user-defined tokens must contain only alphanumeric characters, an underscore (_), or a period (.); no other characters are allowed.

```
input_field=@@property@@
```

For example, to set a port number token in a configuration file, define the property:

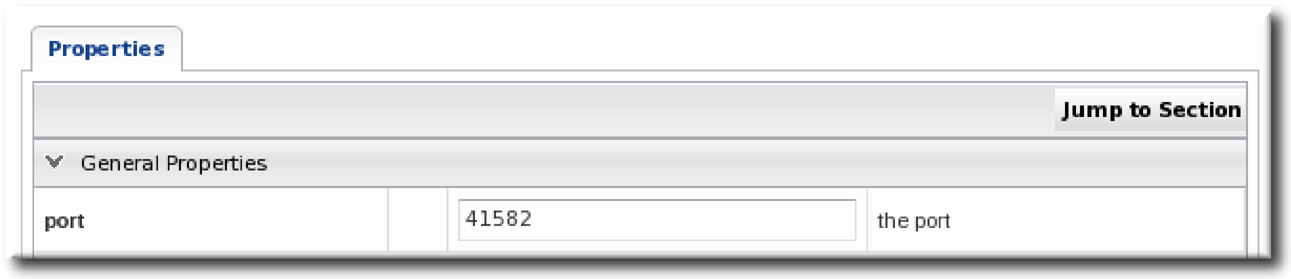
```
port=@@listener.port@@
```

The user-defined token then must be noted in the recipe, so that the provisioning process knows to realize the phrase. To configure a property in an Ant recipe, add a **<rhq:input-property>** key in the Ant XML file.

For example:

```
<rhq:input-property
    name="listener.port"
    ... />
```

The provisioning wizard prompts for a value for all of the user-defined tokens referenced in the recipe.



port	41582	the port

Figure 27.3. Port Token During Provisioning

Along with user-defined variables that can be specified in the recipe file, there are variables that are made implicitly available to recipes. These tokens can be used in a templatized file as a user-defined variable without having to define the token template in the recipe itself.

Table 27.2. Variables Defined by JBoss ON

Token	Description
rhq.deploy.dir	The directory location where the bundle will be installed.
rhq.deploy.id	A unique ID assigned to the specific bundle deployment.
rhq.deploy.name	The name of the bundle deployment.

Additionally, some tokens can be realized by the provisioning process pulling information from the local system. These values, listed in Table 27.3, “System-Defined Tokens”, are taken either from the Java API or from Java system properties. They can be inserted directly in the templatized configuration file without having to put a corresponding entry in the recipe. For example:

```
@@rhq.system.hostname@@
```

Table 27.3. System-Defined Tokens

Token Name	Taken From...	Java API
rhq.system.hostname	Java API	SystemInfo.getHostname()
rhq.system.os.name	Java API	SystemInfo.getOperatingSystemName()
rhq.system.os.version	Java API	SystemInfo.getOperatingSystemVersion()
rhq.system.os.type	Java API	SystemInfo.getOperatingSystemType().toString()

Token Name	Taken From...	Java API
rhq.system.architecture	Java API	SystemInfo.getSystemArchitecture()
rhq.system.cpu.count	Java API	SystemInfo.getNumberOfCpus()
rhq.system.interfaces.java.address	Java API	InetAddress.getByName(SystemInfo.getHostname()).getHostAddress()
rhq.system.interfaces.*network_adapter_name*.mac	Java API	NetworkAdapterInfo.getMacAddress()
rhq.system.interfaces.*network_adapter_name*.type	Java API	NetworkAdapterInfo.getType()
rhq.system.interfaces.*network_adapter_name*.flags	Java API	NetworkAdapterInfo.getAllFlags()
rhq.system.interfaces.*network_adapter_name*.address	Java API	NetworkAdapterInfo.getUnicastAddresses().get(0).getHostAddress()
rhq.system.interfaces.*network_adapter_name*.multicast.address	Java API	NetworkAdapterInfo.getMulticastAddresses().get(0).getHostAddress()
rhq.system.sysprop.java.io.tmpdir	Java system property	
rhq.system.sysprop.file.separator	Java system property	
rhq.system.sysprop.line.separator	Java system property	
rhq.system.sysprop.path.separator	Java system property	
rhq.system.sysprop.java.home	Java system property	
rhq.system.sysprop.java.version	Java system property	
rhq.system.sysprop.user.timezone	Java system property	
rhq.system.sysprop.user.region	Java system property	
rhq.system.sysprop.user.country	Java system property	

Token Name	Taken From...	Java API
rhq.system.sysprop.user.language	Java system property	

27.5.6. Processing JBoss ON Properties and Ant Properties

As described in Section 27.5.5, “Using Templatized Configuration Files” , JBoss ON defines its own recipe properties, in **`<rhq:input-property>`** tags. These are input properties; at the time the bundle is deployed, the server parses the configuration in the file and prompts for any input properties. It then deploys the bundle using those user-defined values.

JBoss ON also supports using Ant **`<property>`** tags — but in a more limited way than a normal Ant build system.

Deploying a bundle has two steps: creating the bundle entity in the JBoss ON server and then installing that bundle on a managed resource, through the JBoss ON agent. Unlike a normal Ant build system, *these steps happen on different machines*.

It is critical to understand that the systems — and, therefore, the environments — where these two build steps are run are different. It affects the overall processing workflow for bundles and limits the Ant build process.

When a bundle is uploaded (created), the JBoss ON server checks the Ant recipe to see what bundle files are required. This means that it processes the **`<rhq:bundle>`** element for the archive file **`<rhq:archive>`** and any other associated content.

At this point, the server has not processed any bundle configuration or evaluated any properties, either JBoss ON input properties or Ant proprties. This is because many JBoss ON input properties are environment-specific, and attempting to expand those properties would result in different bundle configuration (such as Java version, operating system version, and even filesystem location) than what is required when the bundle is ultimately deployed on the managed platform.

By moving all of the Ant processsing over to the JBoss ON agent platform, the way that Ant properties are processed is different than on a typical Ant build server. In a typical Ant implementation, Ant properties can be used to templatize core elements of the recipe, such as filenames and version numbers. However, since the Ant script is not run on the JBoss ON server, any templatized properties cannot (yet) be expanded. This means that required configuration — such as the archive name, other files, names, and version numbers — cannot be templatized in either Ant properties or JBoss ON input properties. They must be set explicitly or left empty so that the server prompts for the value. Another way of saying it is that properties used when the script is actually run are ignored, while required properties require a defined value up front. A templatized value at this point in the configuration is treated as a literal value.

27.5.7. Limits and Considerations for Ant Recipes

27.5.7.1. Unsupported Ant Tasks

As stated in Section 27.5.4.1, “Supported Ant Tasks” , most standard Ant tasks are supported for use in bundle recipes, but there are some tasks which are not supported:

- **`<antcall>`** (instead, use **`<ant>`** to reference a separate XML file in the bundle which contains the Ant targets)

- `<macrodef>` and macro definitions

27.5.7.2. Symlinks

The Java utilities library (`java.util.zip`) included for the bundling system *does not* support symbolic links. Therefore, bundle recipes and configuration files cannot use symlinks.

Symlinks may be an issue for an application server (such as EAP or EPP) which was installed initially from an RPM and then compressed to be used in a bundle. The RPMs available from Red Hat Network contain operating system-specific symlinks which would cause failures during provisioning.

If it is necessary to reference shared libraries or other files (which would normally be referenced with a symlink), then include the required libraries in an archive with the bundle and use an Ant task to extract the files when deploying the bundle. For example:

```
<untar src="abc.tar.gz" compression="gzip" dest="somedirectory"/>
```

27.5.7.3. WARNING: The Managed (Target) Directory and Overwriting or Saving Files

One important thing to consider with an Ant recipe is how to handle files in the deployment directory. (This is touched on in Section 27.1.3, “File Handling During Provisioning”.)

By default, deploying or updating a bundle replaces everything in the deployment directory, either by overwriting it or deleting it. The file handling rules are very similar to RPM package upgrade rules. This is very simplified, but the provisioning process responds in one of two ways to existing files the deployment directory:

1. The file in the current directory is also in the bundle. In this case, the bundle file always overwrites the current file. (There is one exception to this. If the file in the bundle has not been updated and is the same version as the local file, but the local file has modifications. In that case, the local file is preserved.)

2. The file in the current directory does not exist in the bundle. In that case, the bundle deletes the file in the current directory.

The behavior for #2, when a file is deleted, can be changed by settings in the Ant recipe.

There are three ways to manage if and how files are preserved during provisioning: **`compliance`**, **`<rhq:ignore>`**, and **`cleanDeployment`**.

compliance

All of the information about the application being deployed is defined in the **`<rhq:deployment-unit>`** element in a bundle recipe. The **`compliance`** attribute on the **`<rhq:deployment-unit>`** element sets how the provisioning process should handle existing files in the deployment directory.

The default value is *compliance=full* which means that the provisioning process deletes any other files in the root directory.

NOTE

Any existing content in the root directory is backed up before it is deleted, so it can be restored later.

The backup directory is **`/home/.rhqdeployments/`***resourceID*/**`backup`**.

Alternately, the value can be set to filesAndDirectories, which tells the provisioning process to ignore any existing files in the root directory, as long as there is not a corresponding file in the bundle.

The **`compliance`** attribute applies to both the initial deployment and upgrade operations, so this can be used to preserve files that may exist in a directory before a bundle is ever deployed.

See Section 27.5.8.3, “rhq:deployment-unit”.

NOTE

When a bundle will no longer be used on a resource, it can be entirely removed from the filesystem. This is called *purging*. The way that the provisioning system handles files when purging a bundle mirrors that way that it handles files when provisioning a system. By default, purging a bundle deletes everything in the deployment directory. If the *compliance* option is set to filesAndDirectories in the bundle, then the provisioning process removes all of the files and directories associated with the bundle and leaves unrelated files and directories intact.

<rhq:ignore>

There can be files that are used or created by an application, apart from the bundle, which need to be preserved after a bundle deployment. This can include things like log files, instance-specific configuration files, or user-supplied content like images. These files can be ignored during the provisioning process, which preserves the files instead of removing them.

To save files, use the **`<rhq:ignore>`** element and list the directories or files to preserve.

```
<rhq:ignore>
    <rhq:fileset>
        <include name="logs/*.log"/>
    </rhq:fileset>
</rhq:ignore>
```

The **`<rhq:ignore>`** element only applies when bundles are updated; it does not apply when a bundle is initially provisioned.

Also, the **`<rhq:ignore>`** element only applies to file that exist outside the bundle. Any files that are in the bundle will overwrite any corresponding files in the deployment directory, even if they are specified in the **`<rhq:ignore>`** element.

See Section 27.5.8.10, “rhq:ignore”.

Clean Deployment

Both **`compliance`** and **`<rhq:ignore>`** are set in the recipe. At the time that the bundle is actually provisioned, there is an option to run a *clean deployment*. The clean deployment option deletes everything in the deployment directory and provisions the bundle in a clean directory, regardless of the **`compliance`** and **`<rhq:ignore>`** settings in the recipe.

See Section 27.7.6, “Deploying a Bundle to a Clean Destination” .

27.5.8. A Reference of JBoss ON Ant Recipe Elements

27.5.8.1. rhq:bundle

Contains the definition for the main JBoss ON-related Ant task that is required for any Ant bundle recipe. This element defines basic information about the bundle and is the parent element for all of the specific details about what is in the bundle and how it should be provisioned.

Element Attributes

Attribute	Description	Optional or Required
name	The name given to the bundle.	Required
version	The version string for this specific bundle. Bundles can have the same name, but each bundle of that name must have a unique version string. These version strings normally conform to an OSGi style of versioning, such as **`1.0`** or **`1.2.FINAL`**.	Required
description	A readable description of this specific bundle version.	Optional

Example

```
<rhq:bundle name="example" version="1.0" description="an example bundle">
```

27.5.8.2. rhq:input-property

Adds a property to the bundle task that defines a template token that must have its value supplied by a user at the time the bundle is deployed. This is similar to standard Ant properties.

NOTE

All of the system properties listed in Table 27.3, “System-Defined Tokens” and the Ant-specific tokens in Table 27.2, “Variables Defined by JBoss ON” are available to be used as templatized tokens in bundle configuration *without* having to set a **`<rhq:input-property>`** definition.

All input properties set some parameter that must have its value defined by a user when the bundle is provisioned on a resource, and the fields to enter those values are automatically generated in the JBoss ON UI bundle deployment wizard.

Element Attributes

Attribute	Description	Optional or Required

Attribute	Description	Optional or Required
name	The name of the user-defined property. Within the recipe, this property can be referred to by this name, in the format **`${`***property_name***`}`**.	Required
description	A readable description of the property. This is the text string displayed in the JBoss ON bundle UI when the bundle is deployed.	Required
type	Sets the syntax accepted for the user-defined value. There are several different options: • string • longString • long • password • file • directory • boolean • integer • float • double	Required
required	Sets whether the property is required or optional for configuration. The default value is **`false`**, which means the property is optional. If this argument isn't given, then it is assumed that the property is optional.	Optional
defaultValue	Gives a value for the property to use if the user does not define a value when the bundle is deployed.	Optional

Example

```
<rhq:input-property
    name="listener.port"
    description="This is where the product will listen for incoming
messages"
    required="true"
    defaultValue="8080"
    type="integer"/>
```

See Also

- Section 27.5.8.4, “rhq:archive”
- Section 27.5.8.6, “rhq:file”

27.5.8.3. rhq:deployment-unit

Defines the bundle content — such as applications or configuration files — being deployed by the bundle. A deployment unit can be simple text files, archives, or a full software product, including an application server, web server, or database. A deployment unit can have multiple archive and configuration files associated with it.

Only a single deployment unit is provisioned at a time by the provisioning process, so there can be only one `<rhq:deployment-unit>` element in a bundle recipe.

Element Attributes

Attribute	Description	Optional or Required
name	The name of the application.	Required
compliance	Sets whether JBoss ON should manage all files in the top root directory (deployment directory) where the bundle is deployed. If filesAndDirectories, any unrelated files found in the top deployment directory (files *not* included in the bundle) are ignored and will not be overwritten or removed when future bundle updates are deployed. The default is full, which means that the provisioning process manages all files and directories and removes or overwrites anything not in the bundle. Any existing content in the root directory is backed up before it is deleted, so it can be restored later. The backup directory is `/home/.rhqdeployments/`*resourceID*`/backup`.	Optional

Attribute	Description	Optional or Required
preinstallTarget	An Ant target that is invoked before the deployment unit is installed.	Optional
postinstallTarget	An Ant target that is invoked after the deployment unit is installed.	Optional

Example

```
<rhq:deployment-unit name="appserver" preinstallTarget="preinstall"
postinstallTarget="postinstall">
```

See Also

- Section 27.5.4.2, “Using Default, Pre-Install, and Post-Install Targets”

27.5.8.4. rhq:archive

Defines any archive file that is associated with deploying the application. An archive can be a ZIP or JAR file. A bundle doesn't require an archive file, so this element is optional.

NOTE

This must have an explicit value for the name. It will not process an Ant **${}** property definition.

Element Attributes

Attribute	Description	Optional or Required
name	The filename of the archive file to include in the bundle. **IMPORTANT** If the archive file is packaged with the Ant recipe file inside the bundle distribution ZIP file, then the **name** must contain the *relative path* to the location of the archive file in the ZIP file.	Required

Attribute	Description	Optional or Required
exploded	Sets whether the archive's contents will be extracted and stored into the bundle destination directory (true) or whether to store the files in the same relative directory as is given in the **name** attribute (false). If the files are exploded, they are extracted starting in the deployment directory. Post-install targets can be used to move files after they have been extracted.	Optional

Example

```
<rhq:archive name="file.zip">
    <rhq:replace>
         <rhq:fileset>
               <include name="**/*.properties"/>
         </rhq:fileset>
    </rhq:replace>
</rhq:archive>
```

See Also

- Section 27.5.8.2, “rhq:input-property”
- Section 27.5.8.11, “rhq:fileset”
- Section 27.5.8.9, “rhq:replace”

27.5.8.5. rhq:url-archive

Defines remote archive to use, which is accessed through the given URL. This is similar to **rhq:archive** except that the server accesses the archive over the network rather than including the archive directly in the bundle distribution file.

Element Attributes

Attribute	Description	Optional or Required

Attribute	Description	Optional or Required
url	Gives the URL to the location of the archive file. The archive is downloaded and installed in the deployment directory. **NOTE** For the bundle to be successfully deployed, the URL must be accessible to all agent machines where this bundle is to be deployed. If an agent cannot access the URL, it cannot pull down the archive and thus cannot deploy it on the machine.	Required
exploded	If true, the archive's contents will be extracted and stored into the bundle destination directory; if false, the zip file will be compressed and stored in the top level destination directory. **NOTE** If the files are exploded, they are extracted starting in the deployment directory. Post-install targets can be used to move files after they have been extracted.	Optional

Example

```
<rhq:url-archive url="http://server.example.com/apps/files/archive.zip">
```

```
    <rhq:replace>
         <rhq:fileset>
               <include name="**/*.properties"/>
         </rhq:fileset>
    </rhq:replace>
</rhq:url-archive>
```

See Also

- Section 27.5.8.4, "rhq:archive"
- Section 27.5.8.2, "rhq:input-property"
- Section 27.5.8.11, "rhq:fileset"
- Section 27.5.8.9, "rhq:replace"

27.5.8.6. rhq:file

Contains the information to identify and process configuration files for the application which have token values that must be realized. Normally, configuration files are copied directly from the bundle package into the deployment directory. The **`<rhq:file>`** element calls out files that require processing before they should be copied to the destination. The attributes on the **`<rhq:file>`** element set the name of the raw file in the bundle distribution ZIP file and the name of the target file that it should be copied to.

Raw files can be included with the archive files that contain properties or configuration for the application. These configuration files can be templatized with user-defined or system-defined tokens, like those listed in Section 27.5.5, "Using Templatized Configuration Files" . Any templatized files that are included in the bundle distribution file that are templatized must be listed in the Ant recipe so that they are processed and the tokens are realized.

NOTE

This must have an explicit value for the name. It will not process an Ant **`${}`** property definition.

Element Attributes

Attribute	Description	Optional or Required

Attribute	Description	Optional or Required
name	The name of the raw configuration file. **IMPORTANT** If the configuration file is packaged with the Ant recipe file inside the bundle distribution ZIP file, then the `name` must contain the *relative path* to the location of the file within the ZIP file.	Required
destinationFile	The full path and filename for the file on the destination resource. Relative paths must be relative to the final deployment directory (defined in the `rhq.deploy.dir` parameter when the bundle is deployed). It is also possible to use absolute paths, as long as both the directory and the filename are specified. **NOTE** If the `destinationDir` attribute is used, the `destinationFile` attribute *cannot* be used.	Required, unless destinationDir is used

Attribute	Description	Optional or Required
destinationDir	The directory where this file is to be copied. If this is a relative path, it is relative to the deployment directory given by the user when the bundle is deployed. If this is an absolute path, that is the location on the filesystem where the file will be copied. This attribute sets the *directory* for the file to be copied to. The actual file name is set in the **`name`** attribute. If the **`destinationFile`** attribute is used, the **`destinationDir`** attribute *cannot* be used.	Required, unless destinationFile is used
replace	Indicates whether the file is templatized and requires additional processing to realize the token values.	Required

Example

```
<rhq:file name="test-v2.properties"
destinationFile="subdir/test.properties" replace="true"/>
```

If neither the **`destinationDir`** nor the **`destinationFile`** attribute is used, then the raw file is placed in the same location under the deployment directory as its location in the bundle distribution.

27.5.8.7. rhq:url-file

As with **`rhq:file`**, contains the information to identify and process configuration files for the application which have token values that must be realized. This option specifies a remote file which is downloaded from the given URL, rather than being included in the bundle archive.

Element Attributes

Attribute	Description	Optional or Required

Attribute	Description	Optional or Required
url	Gives the URL to the templatized file. The file is downloaded and installed in the deployment directory. **NOTE** For the bundle to be successfully deployed, the URL must be accessible to all agent machines where this bundle is to be deployed. If an agent cannot access the URL, it cannot pull down the archive and thus cannot deploy it on the machine.	Required
destinationFile	The full path and filename for the file on the destination resource. Relative paths must be relative to the final deployment directory (defined in the `rhq.deploy.dir` parameter when the bundle is deployed). It is also possible to use absolute paths, as long as both the directory and the filename are specified. **NOTE** If the `destinationDir` attribute is used, the `destinationFile` attribute *cannot* be used. This attribute must give both the path name and the file name.	Required, unless destinationDir is used

Attribute	Description	Optional or Required
destinationDir	The directory where this file is to be copied. If this is a relative path, it is relative to the deployment directory given by the user when the bundle is deployed. If this is an absolute path, that is the location on the filesystem where the file will be copied. This attribute sets the *directory* for the file to be copied to. The actual file name is set in the **`name`** attribute. If the **`destinationFile`** attribute is used, the **`destinationDir`** attribute *cannot* be used.	Required, unless destinationFile is used
replace	Indicates whether the file is templatized and requires additional processing to realize the token values.	Required

Example

```
<rhq:url-file url="http://server.example.com/apps/files/test.conf"
destinationFile="subdir/test.properties" replace="true"/>
```

If neither the **`destinationDir`** nor the **`destinationFile`** attribute is used, then the raw file is placed in the same location under the deployment directory as its location in the bundle distribution.

See Also

- Section 27.5.8.6, “rhq:file”

27.5.8.8. rhq:audit

Sets custom audit trail messages to use during the provisioning process. This is useful with complex recipes that perform some additional custom tasks. As the tasks are processed, the **`rhq:audit`** configuration sends information to the server about the additional processing steps and their results.

Element Attributes

Attribute	Description	Optional or Required
status	The status of the processing. The possible values are SUCCESS, WARN, and FAILURE. The default is SUCCESS.	Optional

Attribute	Description	Optional or Required
action	The name of the processing step.	Required
info	A short summary of what the action is doing, such as the name of the target of the action or an affected filename.	Optional
message	A brief text string which provides additional information about the action.	Optional

Example

```
<rhq:audit status="SUCCESS" action="Preinstall Notice" info="Preinstalling
to ${rhq.deploy.dir}" message="Another optional message">
 Some additional, optional details regarding
 the deployment of ${rhq.deploy.dir}
</rhq:audit>
```

27.5.8.9. rhq:replace

Lists templatized files, in children **`<rhq:fileset>`** elements, contained in the archive which need to have token values realized when the archive is deployed.

Any file which uses a token that must be replaced with a real value is a templatized file. When the provisioning process runs, the token value is substituted with the defined value. This element lists all of the files which are templatized; the only files which are processed by the provisioning system for token substitution are the ones listed in the **`<rhq:replace>`** element.

Example

```
<rhq:archive name="file.zip">
    <rhq:replace>
        <rhq:fileset>
            <include name="**/*.properties"/>
        </rhq:fileset>
    </rhq:replace>
</rhq:archive>
```

See Also

- Section 27.5.8.11, “rhq:fileset”
- Section 27.5.8.4, “rhq:archive”

27.5.8.10. rhq:ignore

Lists files in the deployment directory which should not be deleted when a new bundle is deployed. **This only applies to upgrade operations, not to the initial deployment of a bundle.**

Once an application is deployed, instance-specific files — like data files or logs — can be created and should be retained if the application is ever upgraded. This element, much like **`<rhq:replace>`**, contains a list of files or directories in the instance to save.

NOTE

If a file is ignored in the recipe, then the file is not deleted when the bundle is deployed. However, if a file of the same name exists in the bundle, then the local file is overwritten.

Do not attempt to ignore files that are packaged in the bundle. Only files generated by the applications, such as log and data files, should be ignored by the provisioning process since they should be preserved for the upgraded instance.

IMPORTANT

It is possible to deploy one bundle to a subdirectory of another bundle (such as Bundle A is deployed to **`/opt/myapp`** and Bundle B to **`/opt/myapp/webapp1`**).

In that case, set the recipe in Bundle A to ignore the directory to which Bundle B will be deployed. This prevents updates or reversions for Bundle A from overwriting the configuration from Bundle B.

Example

```
<rhq:ignore>
    <rhq:fileset>
        <include name="logs/*.log"/>
    </rhq:fileset>
</rhq:ignore>
```

See Also

- Section 27.5.8.11, “rhq:fileset”

27.5.8.11. rhq:fileset

Provides a list of files.

Two JBoss ON elements — **`<rhq:replace>`** and **`<rhq:ignore>`** — define file lists in either the archive file or the deployment directory. This element contains the list of files.

Child Element

Child Element	Description
<include name=*filename* />	The filename of the file. For **`<rhq:replace>`**, this is a file within the archive (JAR or ZIP) file which is templatized and must have its token values realized. For **`<rhq:ignore>`**, this is a file in the application's deployment directory which should be ignored and preserved when the bundle is upgraded.

Example

```
<rhq:replace>
     <rhq:fileset>
         <include name="**/*.properties"/>
     </rhq:fileset>
</rhq:replace>
```

See Also

- Section 27.5.8.10, “rhq:ignore”
- Section 27.5.8.9, “rhq:replace”

27.5.8.12. rhq:system-service

Points to a script file to launch as part of the provisioning process. This is usually an init file or similar file that can be used by the deployed application to set the application up as a system service.

Element Attributes

Attribute	Description	Optional or Required
name	The name of the script.	Required
scriptFile	The filename of the script. If the script file is packaged with the Ant recipe file inside the bundle distribution ZIP file, then the `scriptFile` must contain the *relative path* to the location of the file in the ZIP file.	Required
configFile	The name of any configuration or properties file used by the script. If the configuration file is packaged with the Ant recipe file inside the bundle distribution ZIP file, then the `configFile` must contain the *relative path* to the location of the file in the ZIP file.	Optional
overwriteScript	Sets whether to overwrite any existing init file to configure the application as a system service.	Optional
startLevels	Sets the run level for the application service.	Optional
startPriority	Sets the start order or priority for the application service.	Optional

Attribute	Description	Optional or Required
stopPriority	Sets the stop order or priority for the application service.	Optional

Example

```
<rhq:system-service name="example-bundle-init" scriptFile="example-init-script"
      configFile="example-init-config" overwriteScript="true"
      startLevels="3,4,5" startPriority="80" stopPriority="20"/>
```

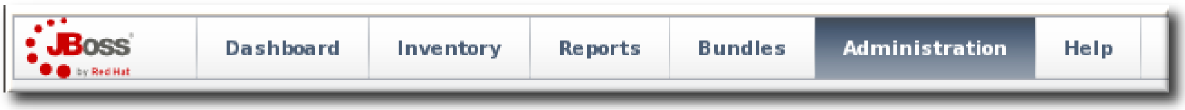
27.6. TESTING BUNDLE PACKAGES

Ant recipes can be complex, so it's important (and useful) to test a bundle before deploying it. JBoss ON includes a command-line tool that can be used to test Ant provisioning bundles quickly.

27.6.1. Installing the Bundle Deployer Tool

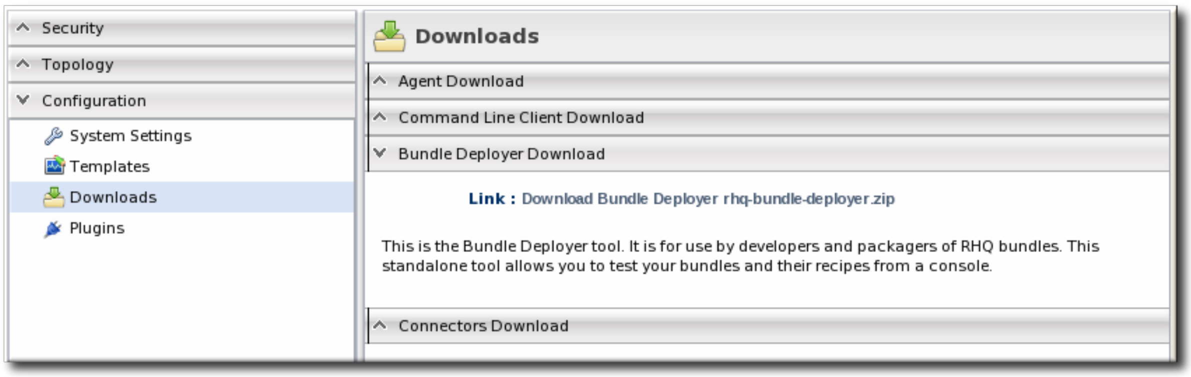
This tool can be downloaded and installed on any machine, independent of any JBoss ON server or agent.

1. Click the **`Administration`** tab in the top menu.



2. Select the **`Downloads`** in the left menu table.

3. Scroll to the **`Bundle Deployer Download`** section, and click the package download link.



This is the Bundle Deployer tool. It is for use by developers and packagers of RHQ bundles. This standalone tool allows you to test your bundles and their recipes from a console.

4. Save the **`.zip`** file into the directory where the bundle tool should be installed, such as **`/opt/`**.

5. Unzip the packages.

```
cd /opt/

unzip rhq-bundle-deployer-version.zip
```

27.6.2. Using the Bundle Deployer Tool

IMPORTANT

This bundle deployment tool is *only* to test the provisioning process and deployed application. This tool does not interact with the JBoss ON server or agent, so JBoss ON is unaware of any applications deployed with this tool and cannot manage them.

1. Unzip the bundle distribution package to check (or copy an unzipped directory that contains the application files). For example:

   ```
   mkdir /tmp/test-bundle
   cd /tmp/test-bundle
   unzip MyBundle.zip
   ```

2. Open the top directory of the bundle distribution, where the **`deploy.xml`** Ant recipe file is.

3. Set the bundle deployer tool location in the PATH.

   ```
   PATH="/opt/rhq-bundle-deployer-3.0.0/bin:$PATH"
   ```

4. Run the bundle deploy tool, and use the format **`-D`***input_properties* to pass the values to user-defined tokens in the templatized files. For example:

   ```
   rhq-ant -Drhq.deploy.dir=/opt/exampleApp -Dlistener.port=7081
   ```

   This installs the application in **`/opt/exampleApp`** and sets a port value of 7081.

   NOTE

   Optionally, use the **`rhq.deploy.id`** attribute to set an identifier for the deployment. The default is 0, which means a new deployment. When bundles are deployed in the UI, the server assigns a unique ID to the deployment. Using the **`rhq.deploy.id`** attribute on a new deployment simulates the server's ID assignment.

   Using the **`rhq.deploy.id`** attribute if there is already a previous deployment allows you to test the upgrade performance of the bundle. Performing an upgrade requires a new, unique ID number.

27.7. PROVISIONING BUNDLES

27.7.1. Managing Bundle Groups

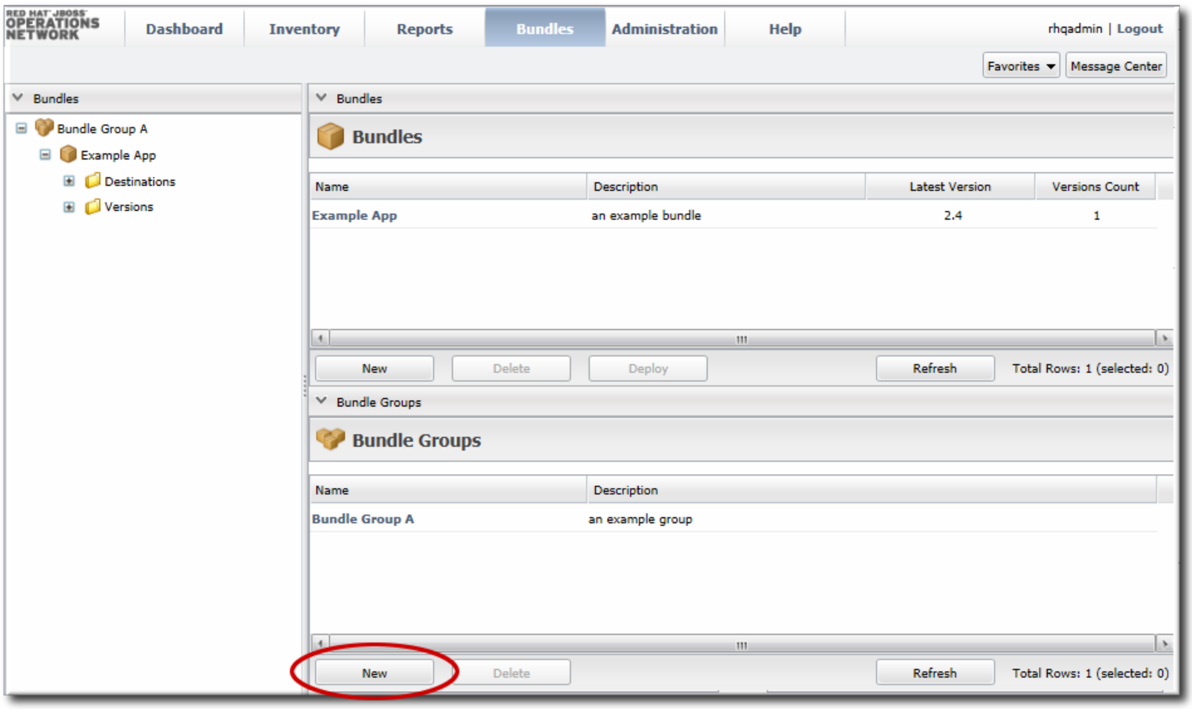
Bundle groups are an integral part of defining access controls for JBoss ON. Bundles cannot be deployed until they belong to a bundle group, which is assigned to an appropriately configured role.

27.7.1.1. Creating Bundle Groups

Resource groups are covered in Chapter 6, *Managing Groups*. Resource groups are used to organize individual resources to simplify overall management by consolidating maintenance operations or to maintain access controls.

Bundle groups are analogous to resource groups. Bundle groups consolidate some deployment operations to make it easier to control application lifecycles. More important, bundle groups provide a level of access control in two facets: controlling user access to both create and deploy bundles and resource limits to control where bundles are deployed.

1. Click the **`Bundles`** tab in the top menu.
2. In the **`Bundle Groups`** area at the bottom, click the **`New`** button.



Bundles

Name	Description	Latest Version	Versions Count
Example App	an example bundle	2.4	1

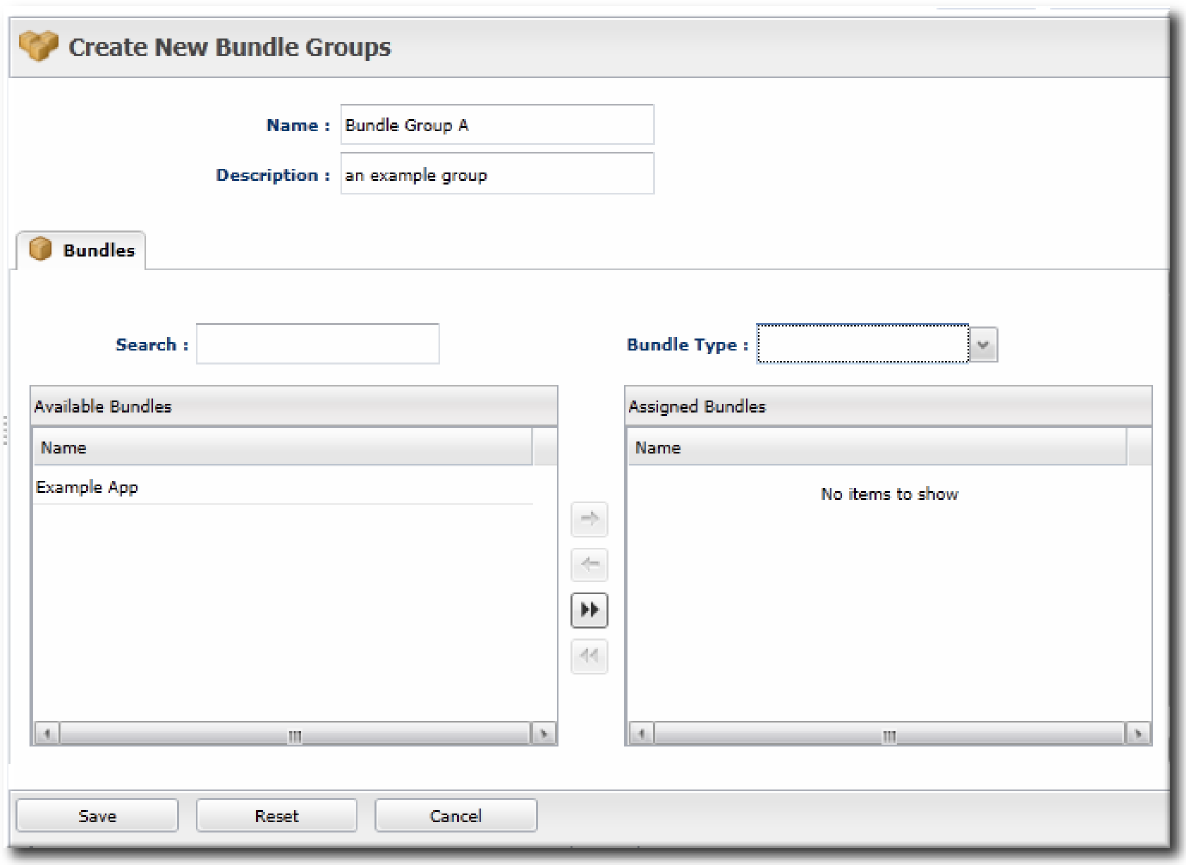
Total Rows: 1 (selected: 0)

Bundle Groups

Name	Description
Bundle Group A	an example group

Total Rows: 1 (selected: 0)

3. Enter a name and description for the group.



Name : Bundle Group A
Description : an example group
Search :
Bundle Type :

Available Bundles
Name
Example App

Assigned Bundles
Name
No items to show

4. Select the group members. If any bundles have been uploaded, then they can be added to the group when it created.

   Alternatively, bundles can be added to a bundle group when they are created, updated, or by editing the group.

5. Click the **Save** button.

27.7.1.2. Assigning Bundles to Bundle Groups

The bundle upload wizard (both for new bundles and for update) includes a step to add or remove the bundle to a bundle group. A bundle can also be added or removed from a bundle group at any time.

When a bundle belongs to a bundle group, it is listed under that bundle group in the hierarchy. This organization helps define the development and deployment lifecycle for bundles.

A user can only see bundles which belong to a bundle group which the user has rights to. The only exception is for users which have the global *view bundles* permission. Those users can see all bundles, regardless of group assignments — including bundles which do not belong to any group. When a bundle belongs to no group, it is in an *unassigned* group. As soon as it is assigned to a group, then its location in the hierarchy is updated.

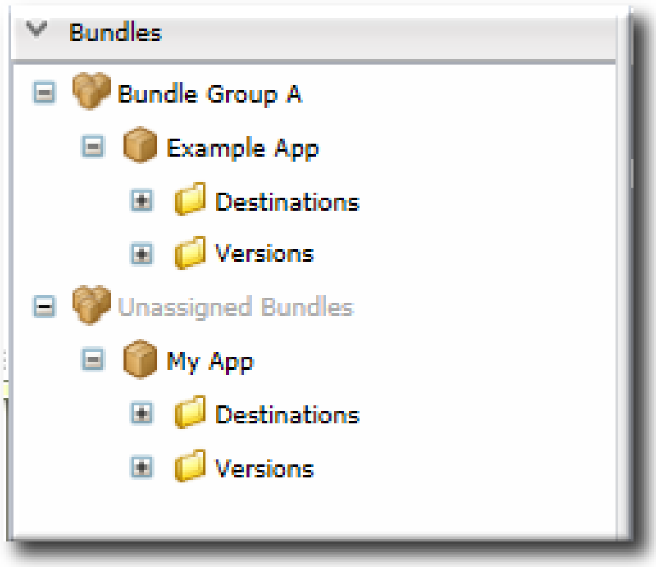
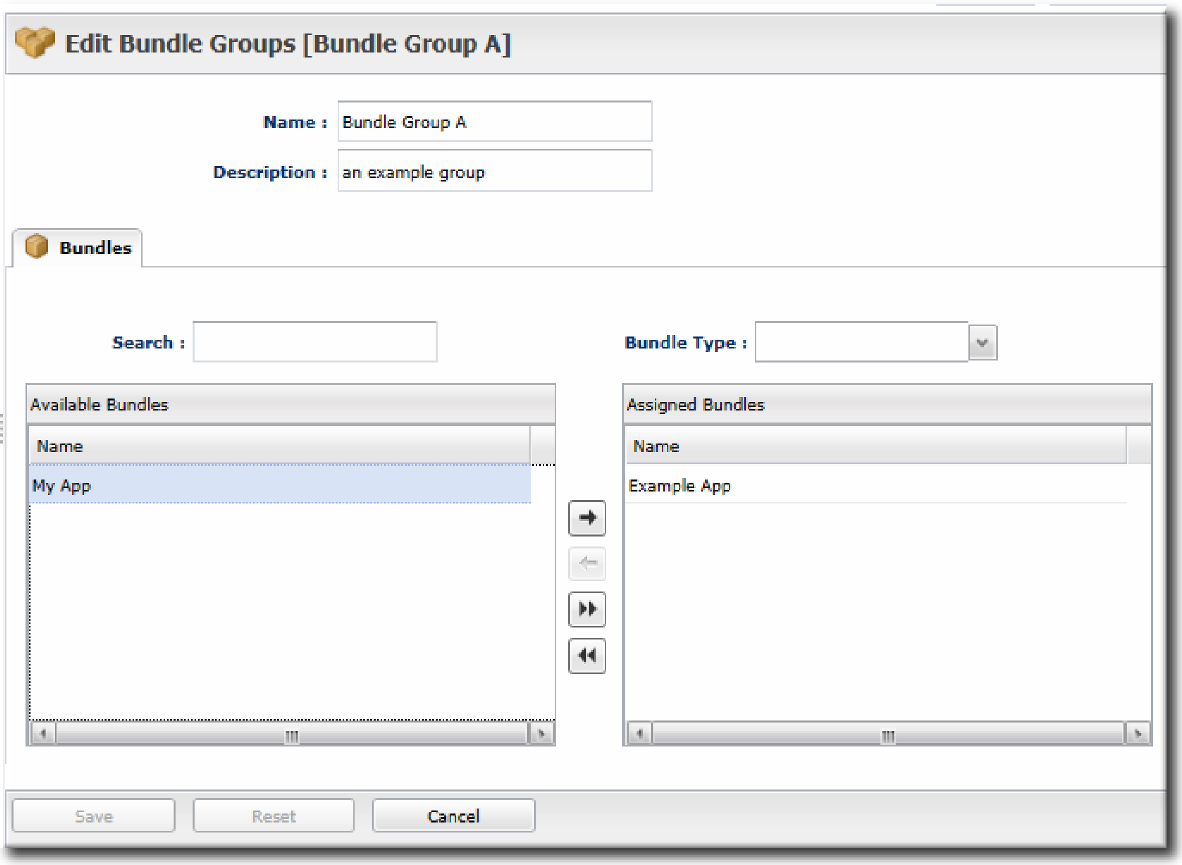


Figure 27.4. Bundle Groups and Unassigned Bundles in the Bundle Hierarchy

IMPORTANT

A bundle must belong to a bundle group for access controls to apply. In other words, a bundle must belong to a bundle group before it can be viewed by users or deployed to a resource.

1. Click the **`Bundles`** tab in the top menu.
2. In the **`Bundle Groups`** area at the bottom, click the name of the bundle group to edit.
3. Select the group members to add or remove from the group. When a bundle is selected in a box, the corresponding arrow becomes active to move it to the other box.



4. Click the **Save** button.

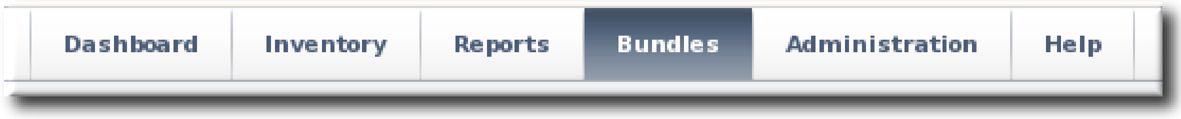
27.7.2. Uploading Bundles to JBoss ON

All of the files associated with a distribution — the recipe, any JARs or ZIPs, and any configuration files — have to be accessible to JBoss ON. Either the files need to be uploaded and stored in the JBoss ON database or a URL needs to be configured, where the server can download the packages.

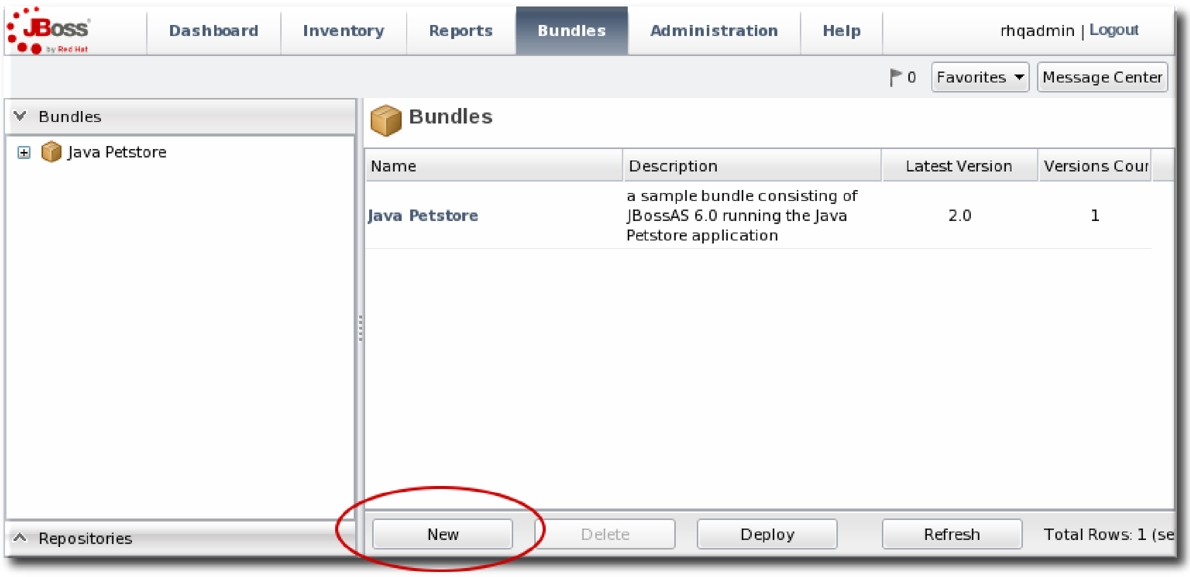
NOTE

If the files are all combined in a single ZIP file to upload, then the recipe file must be in the top level of the package.

1. In the top menu, click the **Bundles** tab.



2. At the bottom of the **Bundles** section, click the **New** button.

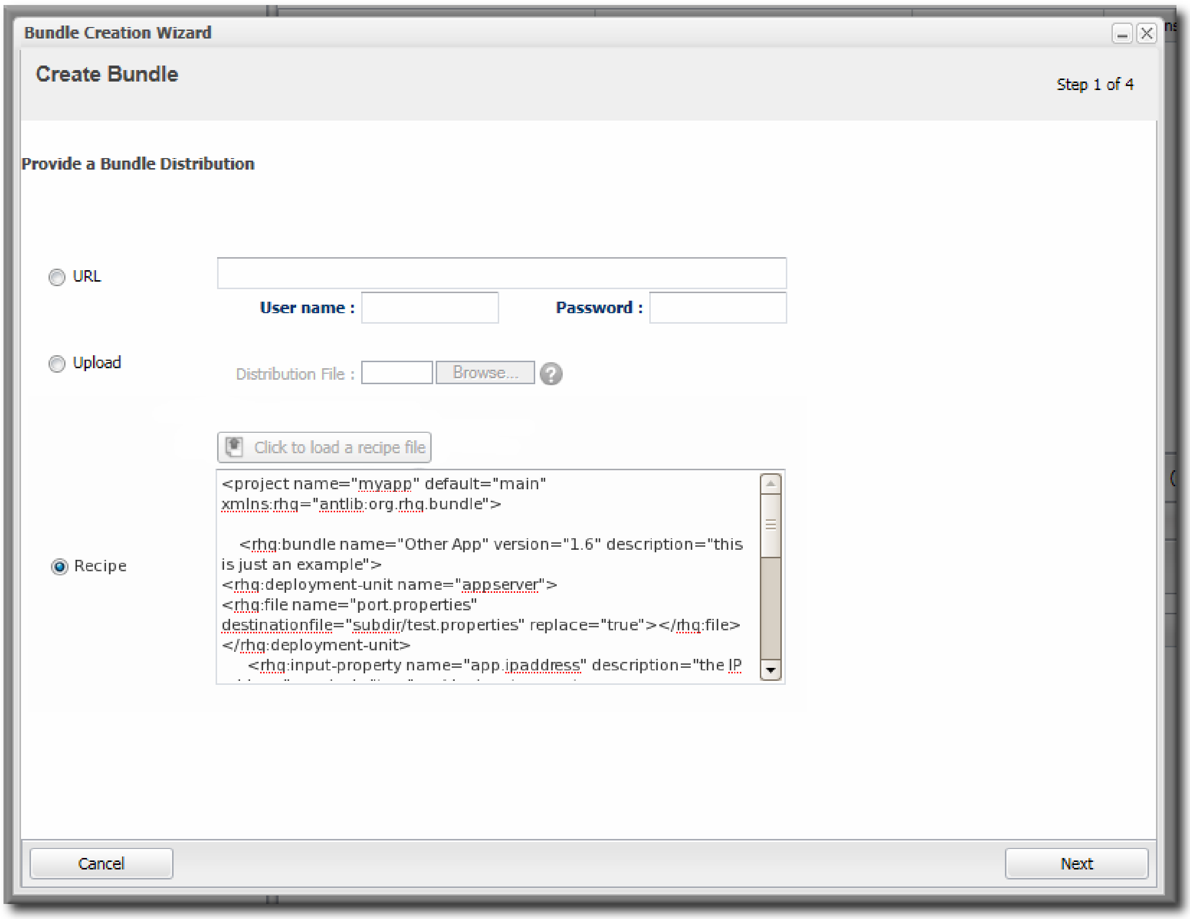


Bundles

Name	Description	Latest Version	Versions Cour
Java Petstore	a sample bundle consisting of JBossAS 6.0 running the Java Petstore application	2.0	1

Total Rows: 1 (se

3. Upload the distribution package or the recipe file.



Recipe [✓]

```
<project name="myapp" default="main"
xmlns:rhq="antlib:org.rhq.bundle">

   <rhq:bundle name="Other App" version="1.6" description="this
is just an example">
<rhq:deployment-unit name="appserver">
<rhq:file name="port.properties"
destinationfile="subdir/test.properties" replace="true"></rhq:file>
</rhq:deployment-unit>
    <rhq:input-property name="app.ipaddress" description="the IP
```

There are three options on how the bundle distribution is made available to the JBoss ON server:

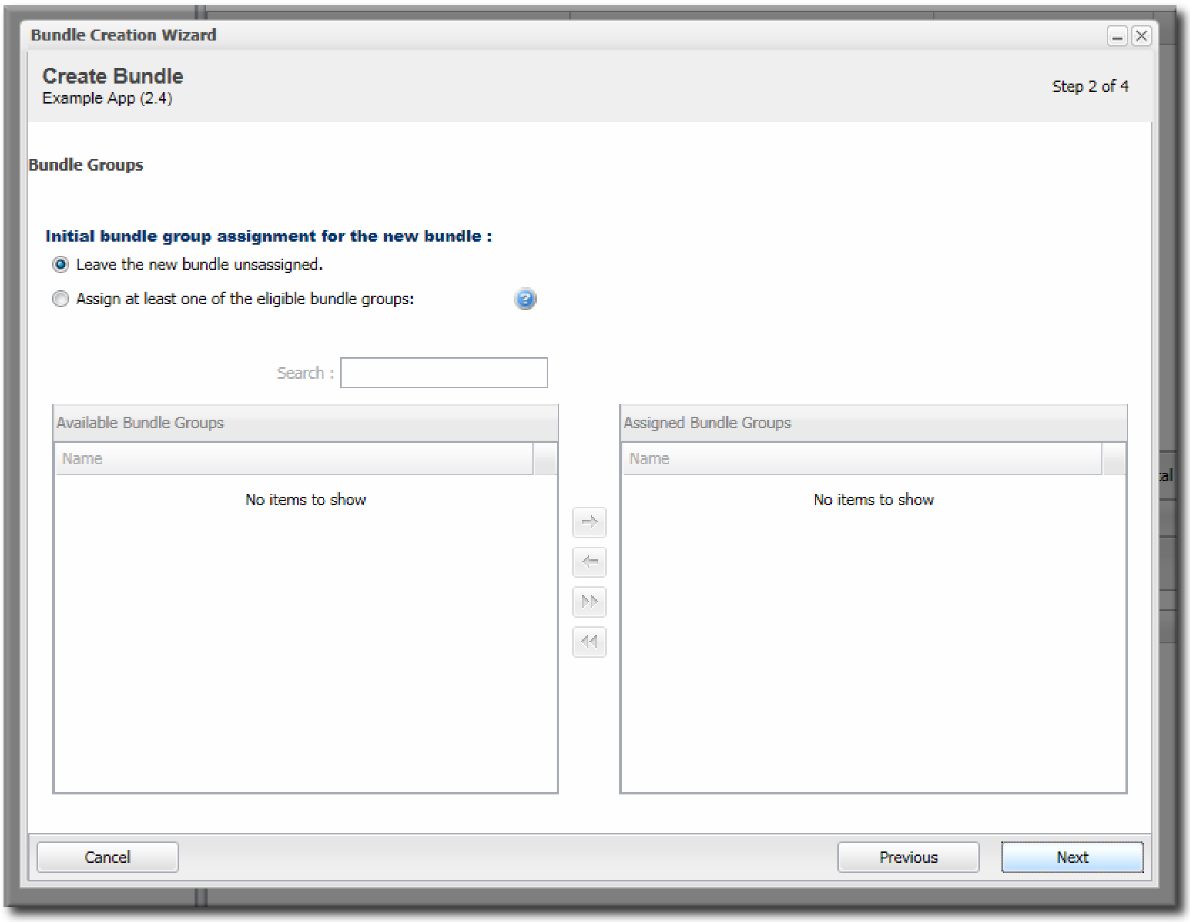
- **URL** points to any URL, such as an FTP site or SVN or GIT repo, where there is a complete bundle distribution file available. If the repository requires authentication, then the username and password for a user account can be given to allow the server to authenticate to the site.

NOTE

Using an SVN or GIT repo allows you to pull the packages directly from a build system.

- **`Upload`** uploads a single bundle distribution file (which includes both the recipe an all associated files) from the local system to the JBoss ON server.
- **`Recipe`** uploads a recipe file only, and then any additional files required for the bundle are uploaded separately. This option includes an edit field where the uploaded recipe can be edited before it is sent to the server.

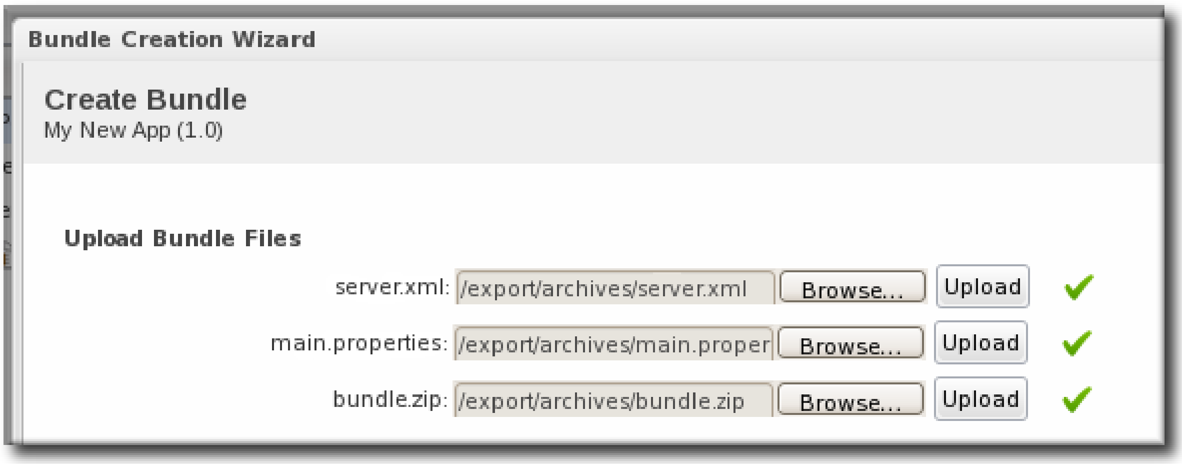
4. Select any groups in the left box to which to assign the bundle, and click the arrow to move it to the **`Assigned`** box.



Initial bundle group assignment for the new bundle :
[✓ Leave the new bundle unassigned.]
Assign at least one of the eligible bundle groups:

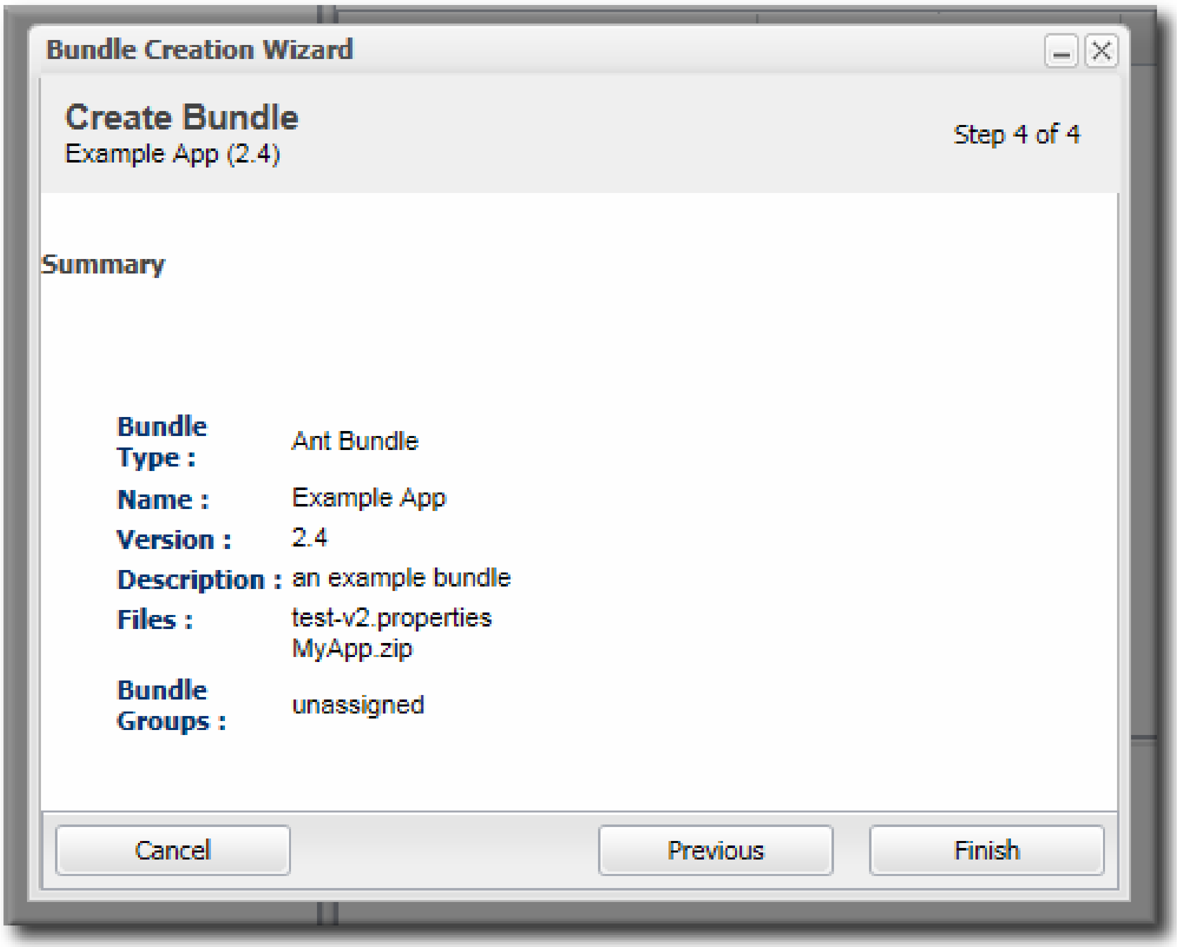
Bundle groups are required for access control. Without a group assignment, users are unable to view bundles (unless they have the global view bundles permission) or to deploy the bundles.

5. Upload any associated files that were not uploaded previously. For the **`URL`** and **`Upload`**, all of the files are usually uploaded in a single file, so there is nothing to do on this screen. For the **`Recipe`** option, all of the files listed in the recipe must be uploaded manually at this step.



server.xml: /export/archives/server.xml ✓
main.properties: /export/archives/main.proper ✓
bundle.zip: /export/archives/bundle.zip ✓

6. The final screen shows all of the information for the new bundle. Click **Finish** to save the new bundle.



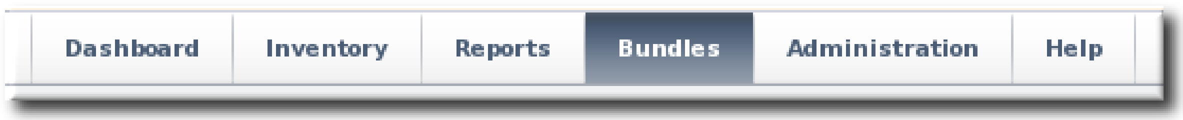
Bundle Type :	Ant Bundle
Name :	Example App
Version :	2.4
Description :	an example bundle
Files :	test-v2.properties MyApp.zip
Bundle Groups :	unassigned

27.7.3. Deploying Bundles to a Resource

Bundles are deployed to *resources* by deploying the bundle to a *JBoss ON group*. Any compatible group that contains resources which support bundles (platforms and JBoss AS resources by default) is automatically listed as an option for the destination.

For platforms, the groups cannot contain different operating systems and architectures. However, the same bundle distribution file and properties can be used for any platform because the provisioning process will automatically format the deployment directory and provisioned files to match the platform's architecture.

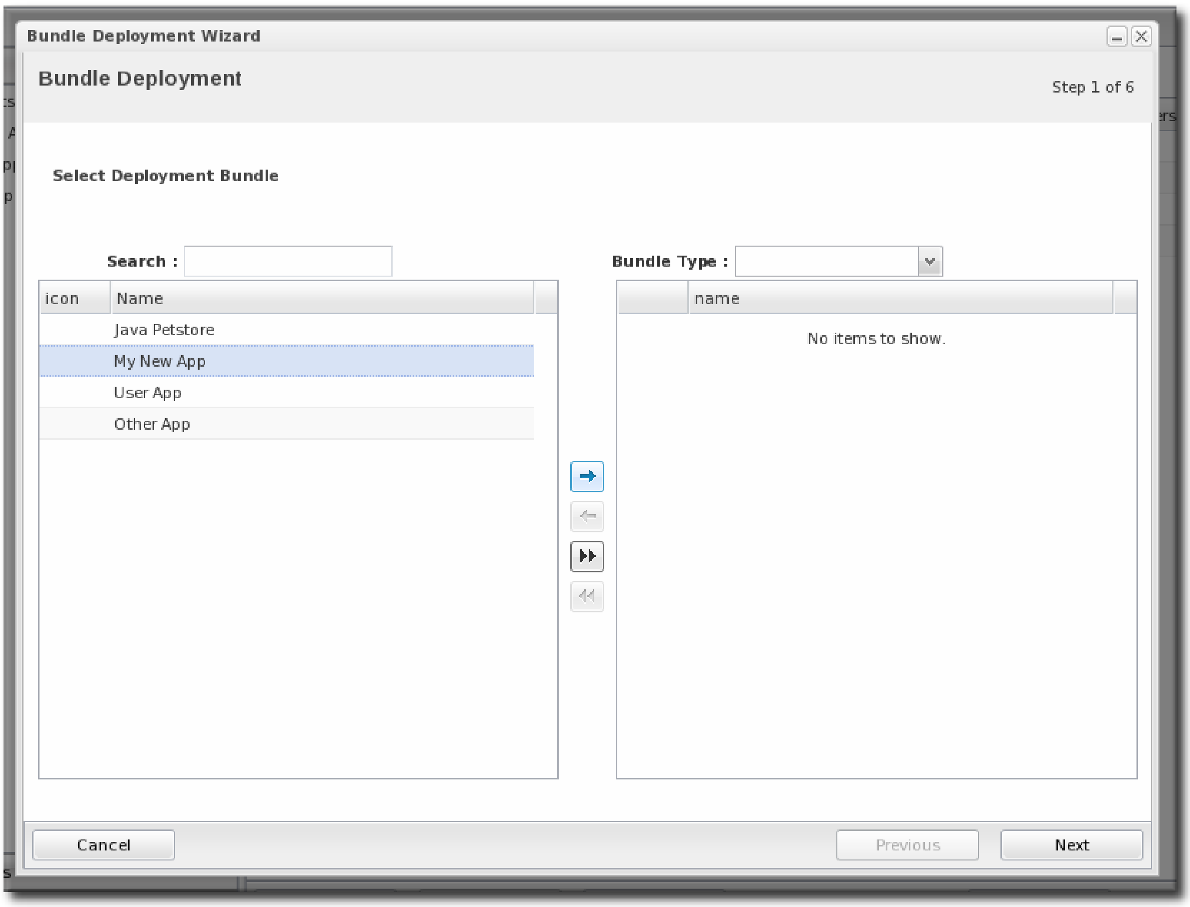
1. In the top menu, click the **Bundles** tab.



2. Scroll to the bottom of the window and click the **Deploy** button.

   Alternatively, click the name of the bundle in the list, and then click the deploy button at the top of the bundle page.

3. Select the bundles to deploy from the list on the left and use the arrows to move them to the box on the right.



4. Once the bundles are selected, define the destination information.

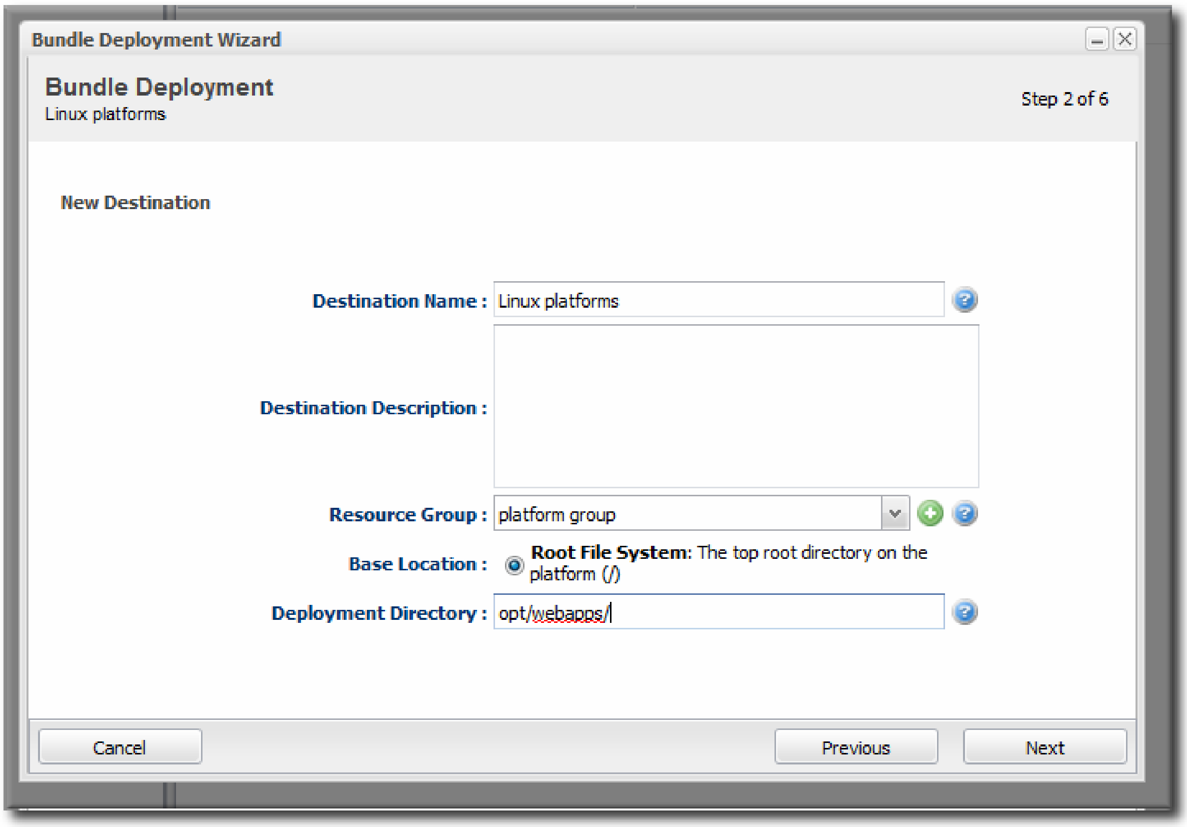
   The destination is a combination of the resources the bundle is deployed on and the directory to which is it deployed. Each destination is uniquely defined for each bundle.

   To define the destination, first select the resource group from the **Resource** drop-down menu. The resource group identifies the type of resource to which the bundle is being deployed, and the resource type defines other deployment parameters. When the group is selected, then the *base location* is defined. For a platform, this is the root directory. For a JBoss AS instance, it is the installation directory. For custom resources, the base location is defined in the plug-in descriptor.

NOTE

If you haven't created a compatible group or if you want to create a new group specifically for this bundle deployment, click the + icon to create the group. Then, continue with the provisioning process.

Set the actual deployment directory to which to deploy the bundle. This directory is a relative path to the plug-in-defined base location.



Destination Name : Linux platforms

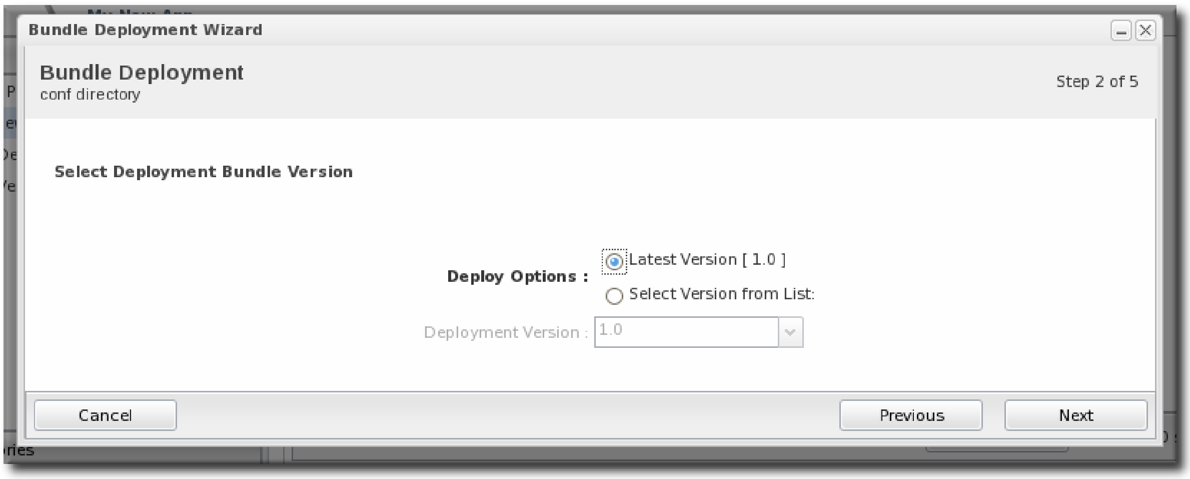
Destination Description :

Resource Group : platform group

Base Location : (●) **Root File System**: The top root directory on the platform (/)

Deployment Directory : opt/webapps/

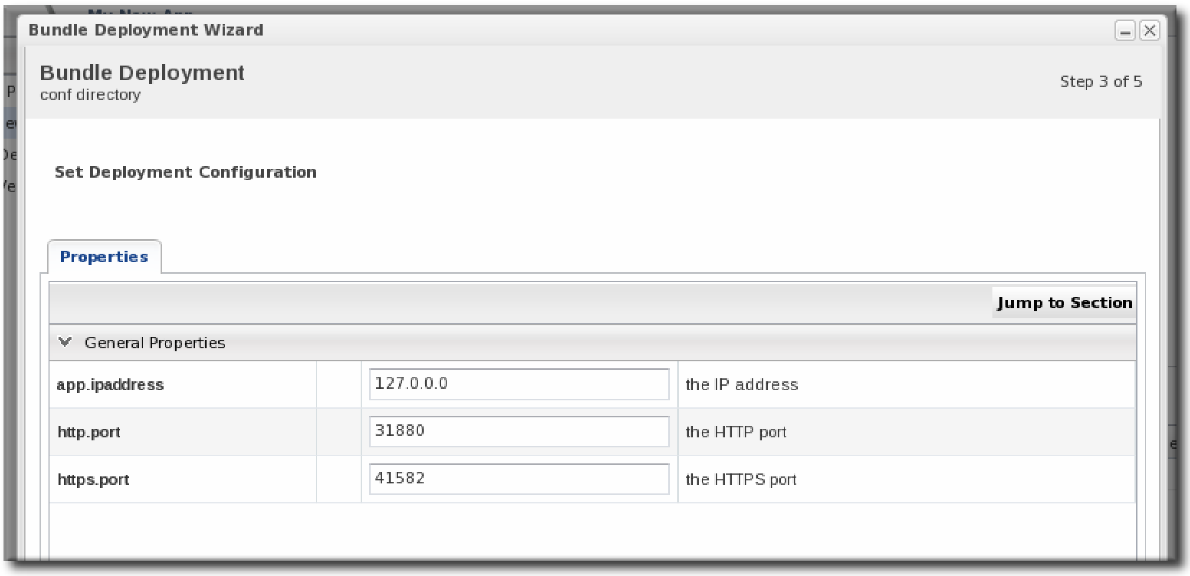
5. Select the version of the bundle to deploy. If there are multiple versions of a bundle available, then any of those versions can be selected. There are also quick options to deploy the latest version or the currently deployed version.



Deploy Options : [✓ Latest Version [1.0]] Select Version from List:

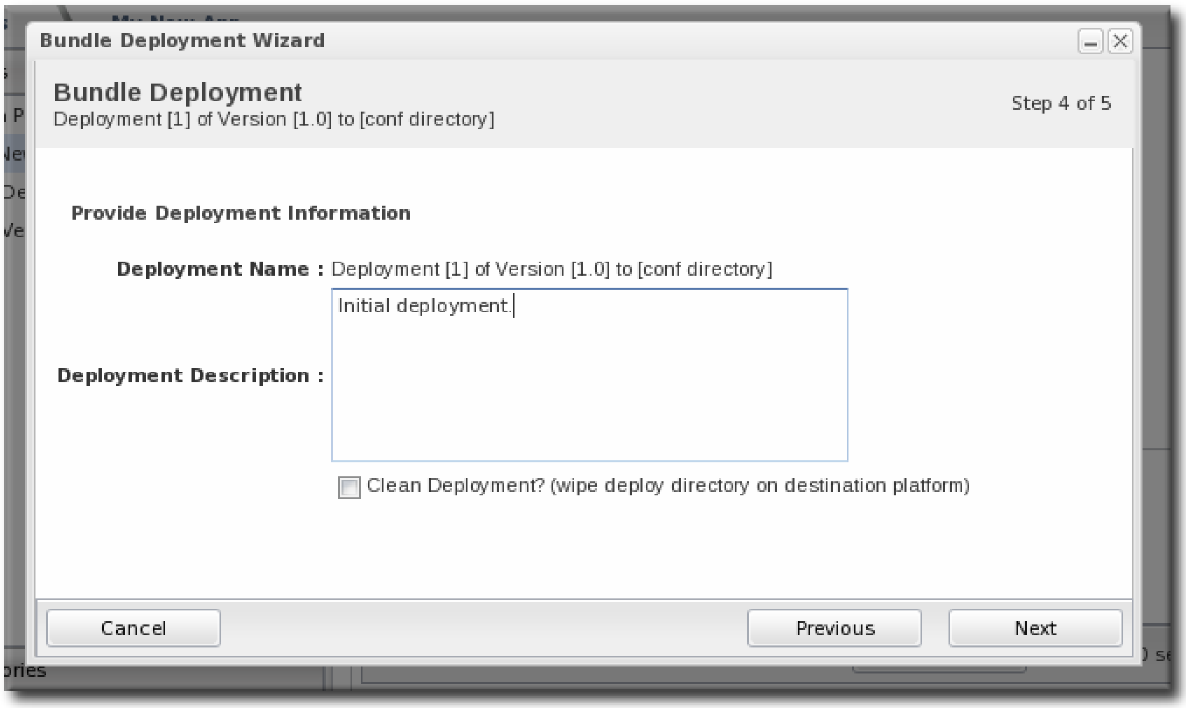
Deployment Version : 1.0

6. If there are any user-defined properties, then they are entered in the fields in the next page. User-defined properties are configured in the bundle recipe using tokens.



app.ipaddress	127.0.0.0	the IP address
http.port	31880	the HTTP port
https.port	41582	the HTTPS port

7. Fill in the information about the specific deployment instance. The checkbox sets the option on whether to overwrite anything in the existing deployment directory or whether to preserve any existing files.


Bundle Deployment Wizard
Bundle Deployment
Deployment [1] of Version [1.0] to [conf directory]
Step 4 of 5
Provide Deployment Information
Cancel
Previous
Next

Deployment Name : Deployment [1] of Version [1.0] to [conf directory]

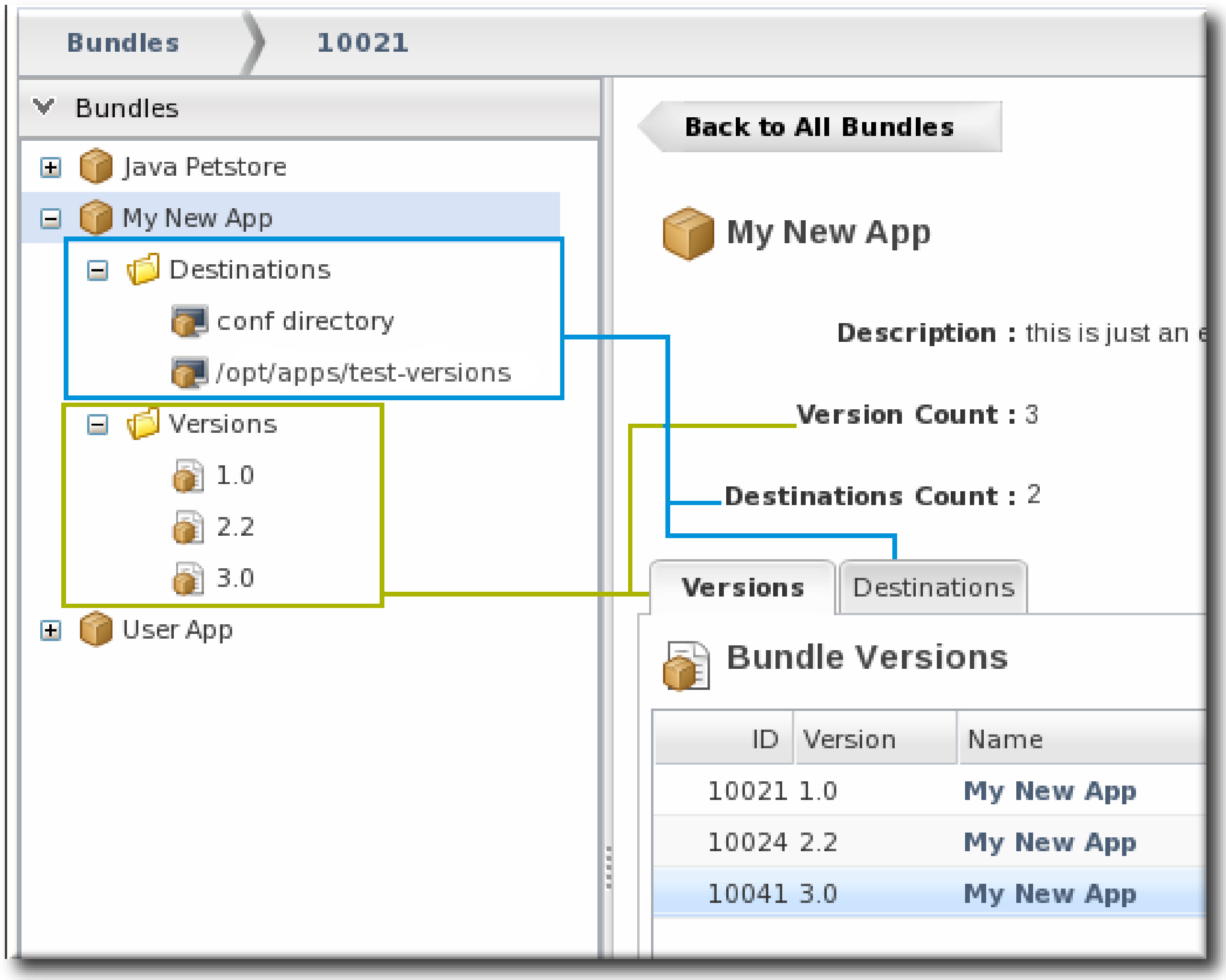
Deployment Description : Initial deployment.

☐ Clean Deployment? (wipe deploy directory on destination platform)

8. The final screen shows the progress for deploying the packages. Click **`Finish`** to complete the deployment.

27.7.4. Viewing the Bundle Deployment History

A bundle has two areas of information: one for its versions and one for its destinations (places where it is deployed). The main bundle entry shows only those two things, the versions and the destinations. The version area is a way to track and control the *content of the bundle*, while the destinations area is a way to track and control *the process of deploying bundles*



My New App

Description : this is just an e

Version Count : 3

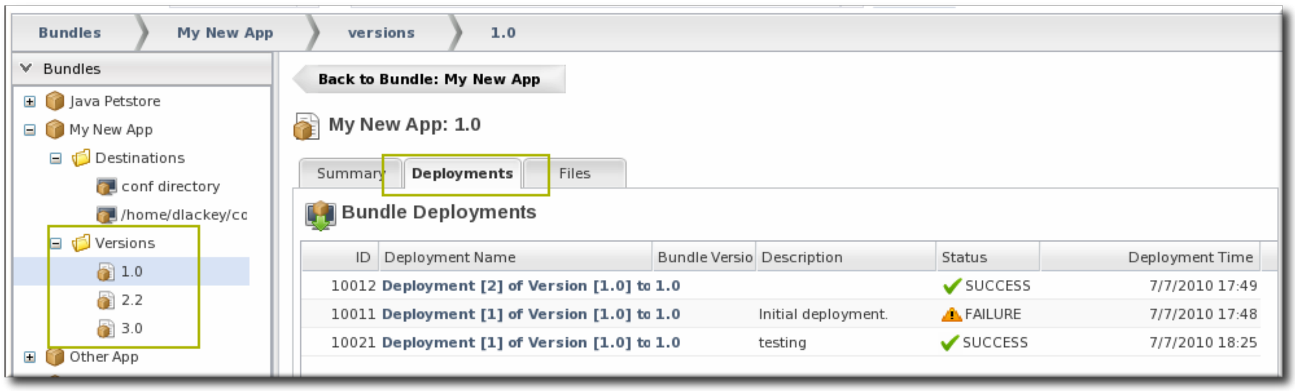
Destinations Count : 2

Bundle Versions

ID	Version	Name
10021	1.0	My New App
10024	2.2	My New App
10041	3.0	My New App

Figure 27.5. Bundles, Versions, and Destinations

Selecting a version under the main bundle entry shows its recipe (on the **`Summary`** tab) and a list of all of the files associated with that particular version (on the **`Files`** tab). The **`Deployments`** tab shows every destination, with timestamps and comments, that that particular version of the bundle has been deployed to.



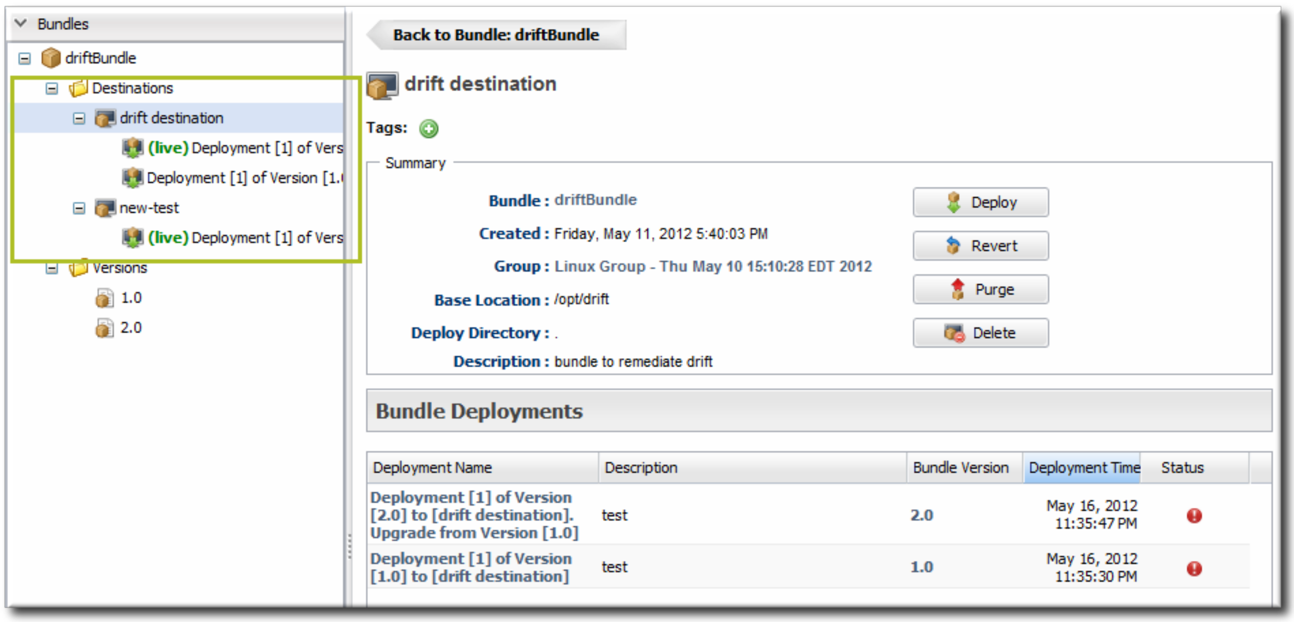
My New App: 1.0

Bundle Deployments

ID	Deployment Name	Bundle Versio	Description	Status	Deployment Time
10012	Deployment [2] of Version [1.0] to	1.0		✓ SUCCESS	7/7/2010 17:49
10011	Deployment [1] of Version [1.0] to	1.0	Initial deployment.	⚠ FAILURE	7/7/2010 17:48
10021	Deployment [1] of Version [1.0] to	1.0	testing	✓ SUCCESS	7/7/2010 18:25

Figure 27.6. Deployment Information for a Version

A destination entry shows only a list of versions that have been deployed to that destination. In a sense, the destination area is the best areas to track the audit history of an application. Each deployment of a bundle version to a destination is listed below the destination, with the live version marked. Reversions are also marked, showing what version the deployment was downgraded to.


Bundles
driftBundle
Destinations
drift destination
(live) Deployment [1] of Vers
Deployment [1] of Version [1.
new-test
(live) Deployment [1] of Vers
Versions
1.0
2.0
Back to Bundle: driftBundle
Tags:
Summary
Deploy
Revert
Purge
Delete

drift destination

Bundle : driftBundle
Created : Friday, May 11, 2012 5:40:03 PM
Group : Linux Group - Thu May 10 15:10:28 EDT 2012
Base Location : /opt/drift
Deploy Directory : .
Description : bundle to remediate drift

Bundle Deployments

Deployment Name	Description	Bundle Version	Deployment Time	Status
Deployment [1] of Version [2.0] to [drift destination]. Upgrade from Version [1.0]	test	2.0	May 16, 2012 11:35:47 PM	●
Deployment [1] of Version [1.0] to [drift destination]	test	1.0	May 16, 2012 11:35:30 PM	●

Figure 27.7. Deployment History for a Destination

Along with showing the history of deployments and updates, the destinations area is the place where new versions can be deployed or reverted most directly.

27.7.5. Reverting a Deployed Bundle

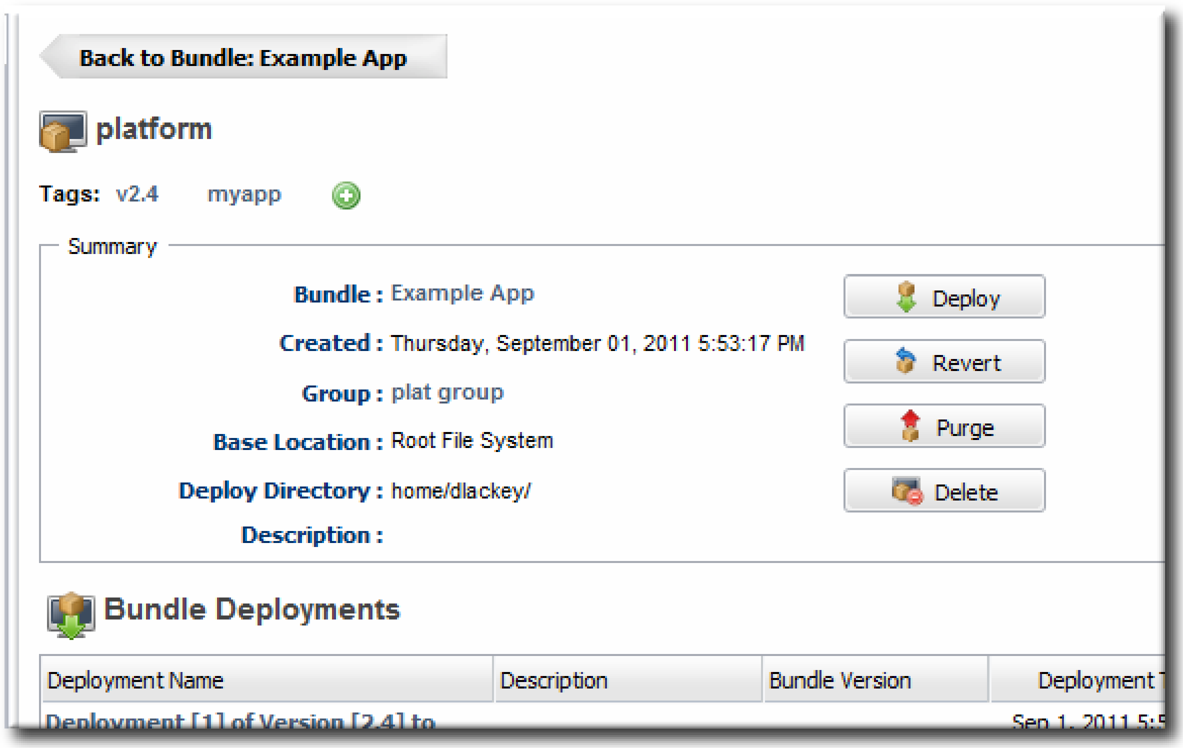
Ant bundles can be rolled back to a previous version number or a previous deployment of that bundle. This provides some extra protection and flexibility when deploying and managing applications, particularly for testing and production systems.

1. In the top menu, click the **Bundles** tab.

2. In the left navigation window, expand the bundle node, and then open the **Destinations** folder beneath it.

3. Select the destination from the left navigation.

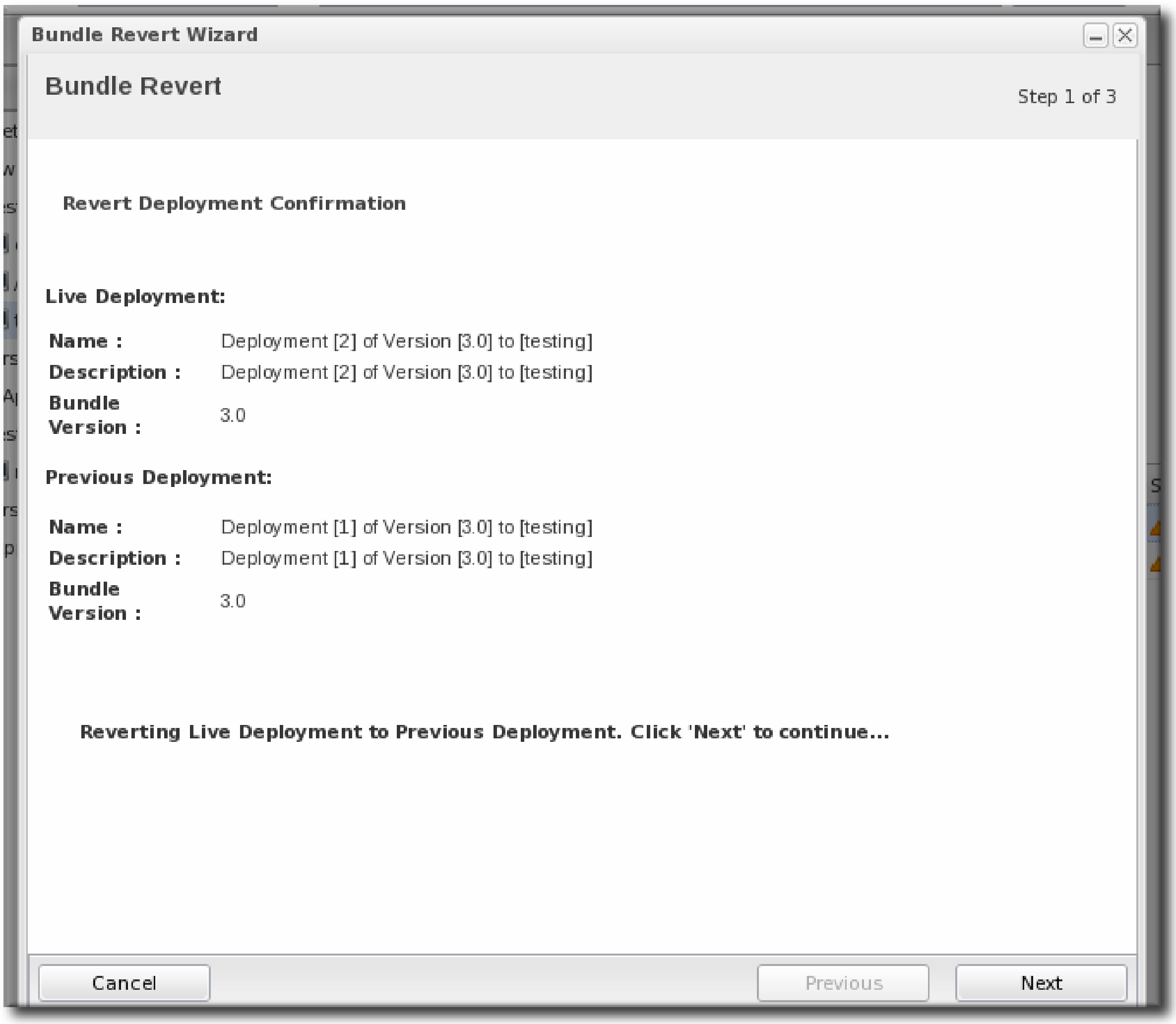
4. In the main window for the destination, click the **Revert** button.



Bundle : Example App
Created : Thursday, September 01, 2011 5:53:17 PM
Group : plat group
Base Location : Root File System
Deploy Directory : home/dlackey/
Description :

Deployment Name	Description	Bundle Version	Deployment T
Deployment [1] of Version [2.4] to			Sep 1, 2011 5:5

5. The next page shows the summary of the current deployment and the immediate previous deployment, which it will be reverted to.



Revert Deployment Confirmation

Live Deployment:

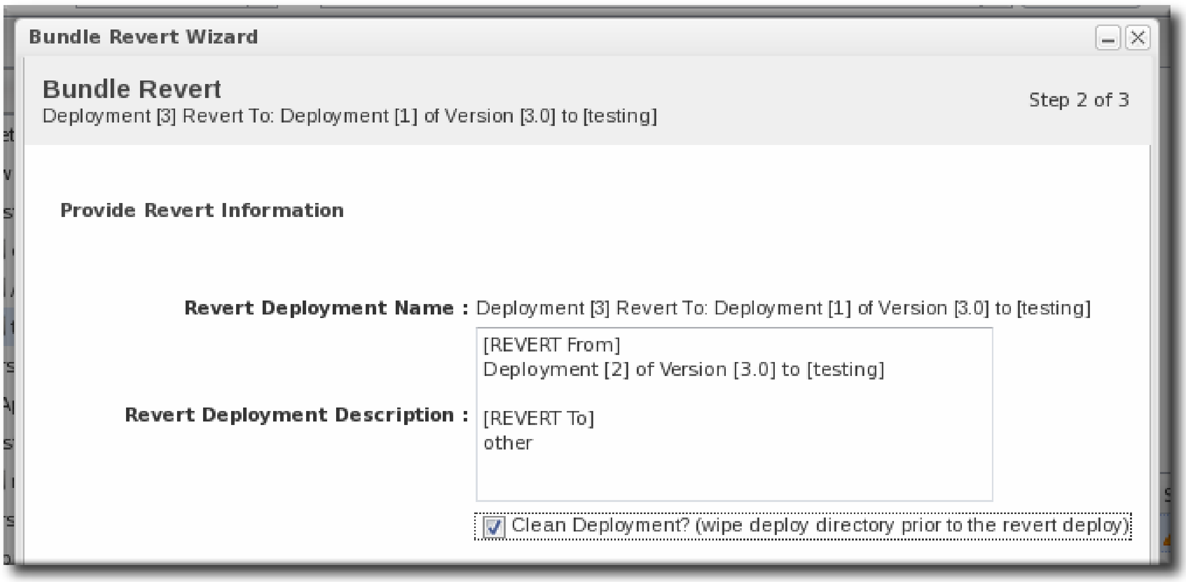
Name : Deployment [2] of Version [3.0] to [testing]
Description : Deployment [2] of Version [3.0] to [testing]
Bundle Version : 3.0

Previous Deployment:

Name : Deployment [1] of Version [3.0] to [testing]
Description : Deployment [1] of Version [3.0] to [testing]
Bundle Version : 3.0

Reverting Live Deployment to Previous Deployment. Click 'Next' to continue...

6. Add any notes to the revert action. Optionally, select the checkbox to clean the deployment directory and install the previous version fresh.


Bundle Revert Wizard

Bundle Revert
Deployment [3] Revert To: Deployment [1] of Version [3.0] to [testing]

Step 2 of 3

Provide Revert Information

Revert Deployment Name : Deployment [3] Revert To: Deployment [1] of Version [3.0] to [testing]

Revert Deployment Description : [REVERT From]
Deployment [2] of Version [3.0] to [testing]

[REVERT To]
other

[✓] Clean Deployment? (wipe deploy directory prior to the revert deploy)

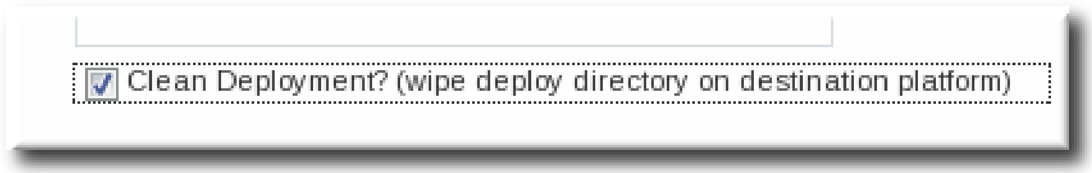
7. Click **`Finish`** on the final screen to complete the rollback.

27.7.6. Deploying a Bundle to a Clean Destination

A bundle can be deployed to a destination where there may already be an application, files, or even a previous bundle deployment. When deploying a new bundle, there are two options for how the provisioning process handles the update:

- Preserve the existing files and directories, with appropriate upgrades, according to the recipe configuration (Section 27.5.7.3, “WARNING: The Managed (Target) Directory and Overwriting or Saving Files”)
- Completely overwrite the existing files and deploy the bundle in an empty directory

To deploy the bundle in a clean directory, then select the **`Clean Deploy`** checkbox when running through the deployment wizard in Section 27.7.3, “Deploying Bundles to a Resource” .


[✓] Clean Deployment? (wipe deploy directory on destination platform)

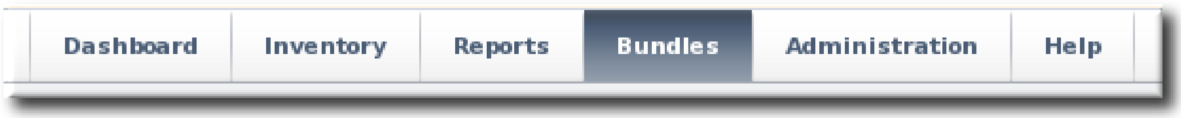
27.7.7. Purging a Bundle from a Resource

Purging a bundle removes all of the files associated with the bundle from all of the target resources. However, this does not remove the bundle from the JBoss ON database, so it can be easily re-deployed to the same resources later or to other resources.

IMPORTANT

The exact files that are purged mirrors how the bundle manages the deployment directory. By default, purging includes deleting the deployment directory (`compliance=full`). If the deployment directory is used by other applications – like an app server `deploy/` directory — then those other applications or files will also be deleted. After purging, there is no live deployment and nothing to revert.

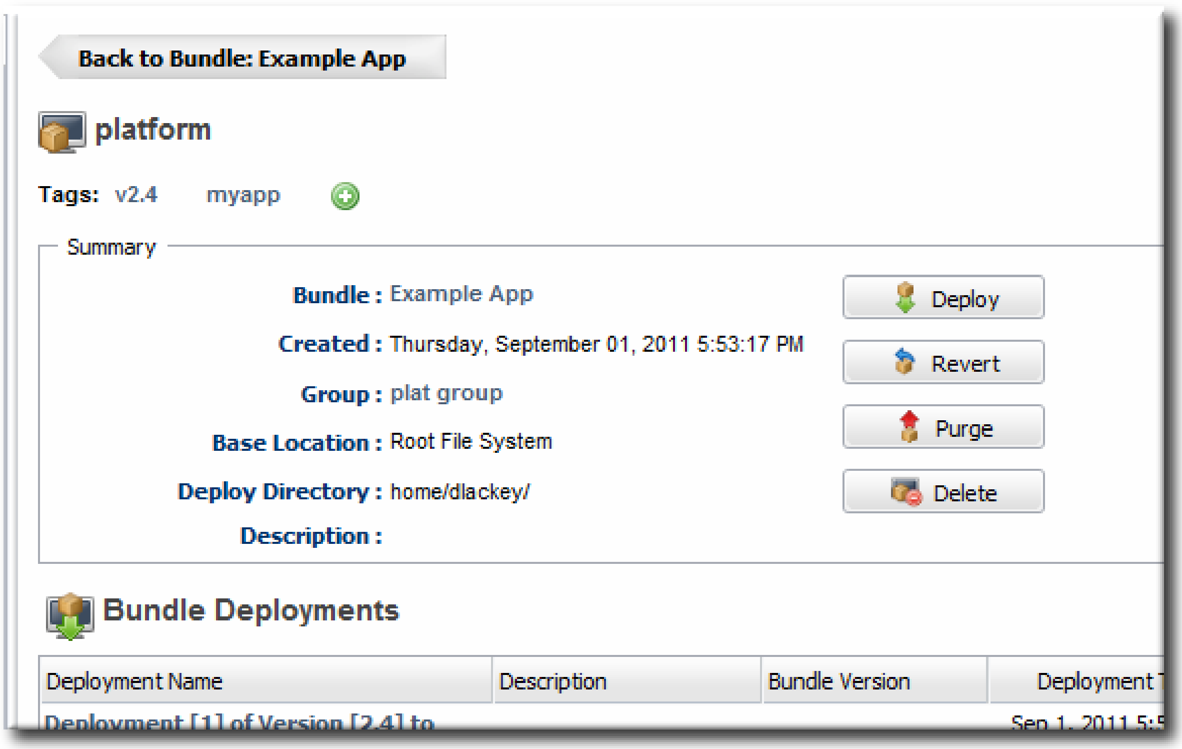
1. In the top menu, click the **`Bundles`** tab.



2. In the left navigation window, expand the bundle node, and then open the **`Destinations`** folder beneath it.

3. Select the destination from the left navigation.

4. In the main window for the destination, click the **`Purge`** button.



Bundle : Example App
Created : Thursday, September 01, 2011 5:53:17 PM
Group : plat group
Base Location : Root File System
Deploy Directory : home/dlackey/
Description :

Deployment Name	Description	Bundle Version	Deployment T
Deployment [1] of Version [2.4] to			Sep 1, 2011 5:5

5. When prompted, confirm that you want to remove the bundled application and configuration from the target resources.

27.7.8. Upgrading Ant Bundles

The bundle upgrade process decides whether to upgrade (meaning, overwrite) files within the application's deployment directory by comparing the MD5 hash codes on the files. There are several different upgrade scenarios:

- If the hash code on the new file is different than the original file and there are no local modifications, then JBoss ON installs the new file over the existing file.

- If the hash code on the new file is different than the original file and there *are* local modifications, then JBoss ON backs up the original file and installs the new file.
- If the hash code on the new file and the original file is the same and there *are* local modifications on the original file, then the provisioning process preserves the original file, in place.
- If there was no file in the previous bundle but there is one in the new bundle, then the new file is used and any file that was added manually is backed up.

Backed up files are saved to a **`backup/`** directory within the deployment's destination directory. If the original file was located outside the application's directory (like, it was stored according to an absolute location rather than a relative location), then it is saved in an **`ext-backup/`** directory within the deployment's destination directory.

NOTE

If a file is ignored in the recipe, then the file is left unchanged. *Never* ignore files packaged in the bundle. Only files generated by the applications, such as log and data files, should be ignored by the provisioning process since they should be preserved for the upgraded instance.

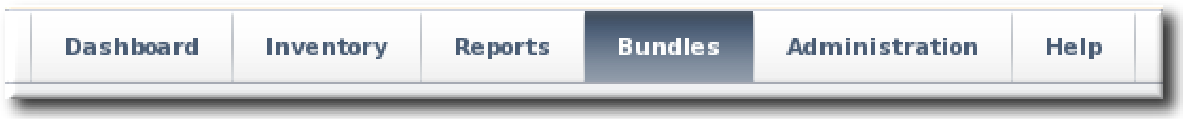
If a completely fresh installation is required, then it is possible to run a clean deployment. This is described in Section 27.7.6, "Deploying a Bundle to a Clean Destination".

27.7.9. Deleting a Bundle from the JBoss ON Server

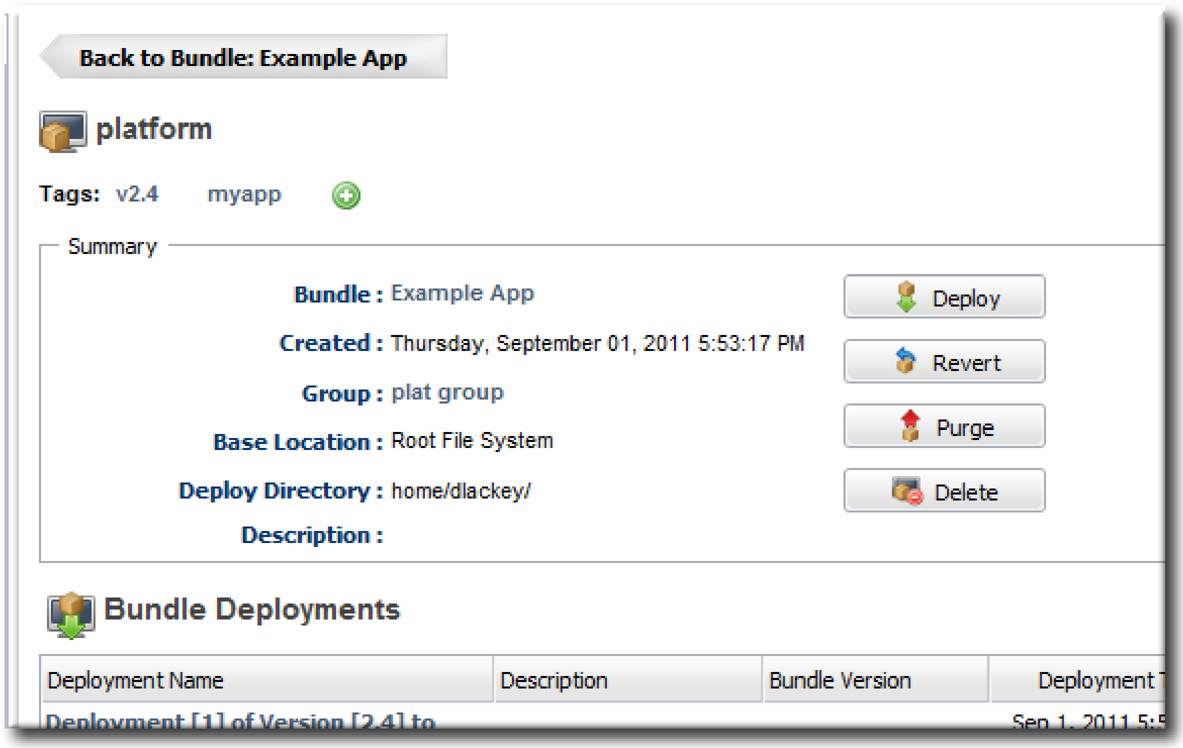
Deleting a bundle removes all of its recipes and associated files from the JBoss ON database. It also removes the bundle from any bundle groups to which it belonged.

The deployed applications or configuration remain intact on the target resources.

1. In the top menu, click the **`Bundles`** tab.



2. In the left navigation window, expand the bundle node, and then open the **`Destinations`** folder beneath it.
3. Select the destination from the left navigation.
4. In the main window for the destination, click the **`Delete`** button.



platform

Tags: v2.4 myapp

Summary

Bundle : Example App
Created : Thursday, September 01, 2011 5:53:17 PM
Group : plat group
Base Location : Root File System
Deploy Directory : home/dlackey/
Description :

Bundle Deployments

Deployment Name	Description	Bundle Version	Deployment T
Deployment [1] of Version [2.4] to			Sep 1, 2011 5:5

5. When prompted, confirm that you want to delete the bundle.

27.8. *EXTENDED EXAMPLE*: USING BUNDLE GROUPS AND ACCESS CONTROL WITHIN THE PROVISIONING PROCESS

Roles define interactions between different entities: resources (through resource groups), users, and bundles (through bundle groups). As Section 27.1.5, “Provisioning and Agent User System Permission” notes, with provisioning, these relationships breakdown along the lines of the different parts of the provisioning lifecyle: bundle/user relationships are most critical for creating bundle versions, while bundle/resource relationships are most critical for deploying bundles.

Roles can be layered; the access controls that are in effect are the *cumulative* rules for all roles which a user is assigned to. This means that if a single user belongs to multiple roles which each allow access to different resource groups, the user has access to *all* of the resource groups listed in those roles. The same is true for bundle groups.

This means that the resource groups and bundle groups defined in roles work in tandem to define the allowed provisioning paths for a user. Section 9.1.3, “Access and Groups” touches on this relationship. For instance, if a user belongs to one role which grants access to Resource Group A and another role which grants him access to Bundle Group B, the user can deploy the bundles in Bundle Group B to the resource in Resource Group A — but not other bundles or other resources.

This building-block approach to roles allws administrators to define very complex relationships between resources, bundles, and other users in very manageable ways.

NOTE

Limit the configuration of a single role as much as possible. It may be beneficial to use entirely separate roles for resource groups/permissions and bundle groups/permissions. This can make it more clear what access a particular role is granting and in what area, which makes it easier to maintain for users.

There are many different permutations of access controls for deploying bundles. The roles defining bundle permissions should echo and reinforce the provisioning workflow, following who creates bundles, who deploys bundles, and in what environments.

These examples outline simple, representative workflows. These can be adapted to the infrastructure, application lifecycle, and user groups within your organization.

27.8.1. Global v. Group Permissions for Creating and Deploying

Global bundle permissions grant the user access to all bundles on the system, regardless of associated groups.

The global view and create permissions allows the user to create bundles independent of bundle groups. The global deploy permission allows the user to deploy bundles to any resource groups that the user can view, even if they are not associated with the role.

Example 27.2. A User Deploys Bundles Anywhere

```
Role R1 <--- User U
   |
   v
View Bundles
Create Bundles
Deploy Bundles
```

With global deploy permissions, the user must still have roles which define what resource groups the user can see. However, those resource groups do not need to be specified in the bundle role.

Group permissions restrict the given action to groups specified within the role itself. This is true for both bundle creation permissions (create bundles in group) and bundle deploy permissions (deploy bundles to resource group).

To restrict deployment to specified resource groups, use the *deploy to group* permission, which requires an explicit resource group in the role.

Example 27.3. Deploy Bundles to a Specific Resource Group: Single Role

```
Resource Group X ---> Role R <--- User U
                        ^
                        |
                 View Bundles
   Deploy Bundles To Group
```

The global permission *view bundles* allows the user to see any bundles on the system; it simply restricts the ability to deploy those bundles to a single resource group. The global permission could be set in a different, but the resource group must still be specified in the same role as the *deploy bundles to group* permission.

Example 27.4. Deploy Bundles to a Specific Resource Group: Two Roles

```
Resource Group X ---> Role R1 <--- User U    -->  Role R2
                         ^                            ^
```

```
                          |                            |
                Deploy Bundles To Group        View Bundles
```

27.8.2. Permissions and the Application Development Workflow

There are two parts to the application provisioning workflow: creating bundle versions in JBoss ON and then deploying those versions on selected resources.

Those two tasks can be divided between different groups within an organization. For example, a development group could created bundles, but the production group is responsible for deploying them. Or a QE lead could create the bundles in JBoss ON, while the QE team is responsible for deploying them to test systems.

While global deploy and create permissions can be set, more than likely, a real environment will restrict most users to actions within specified groups.

The first conceptual part is to create a permission that allows a user to create bundles in a specific group. A global permission to view all bundles can be set in a second role, since that permission could logically be applied to multiple users who all have different bundle groups to manage. (This falls uder the principle of keeping each single role as simple as possible.)

Example 27.5. Creating Bundles in a Specific Bundle Group

```
Bundle Group A ---> Role R1 <--- User U --->  Role R2
                          ^                         ^
                          |                         |
                Create Bundles In Group     View Bundles
```

If multiple team members are responsible for updating each others' bundles or of maintaining a community bundle, then those users can simply be added to the same create role for the given bundle group. This example also specifies a delete permission.

Example 27.6. Multiple Users Updating Each Others' Bundles

```
Bundle Group A
     |
     v
  Role R1 <--- User U1, User U2, User U3
     ^
     |
Create Bundles In Group
Delete Bundles From Group
```

This role arrangement is egalitarian, with each member of the role having equal rights over the other role users' bundles, within the same bundle group.

In many workflows, there will one person or set of people responsible for maintaining bundles, while another group is responsible for deploying those bundles. In that case, one group belongs to a role with *create bundles in group* permissions to a given bundle group, while a separate set of users belong to a role with *deploy bundles to group* permissions for that bundle group and a specified resource group.

Example 27.7. Team Lead Creates Bundles - Team Members Deploy Bundles

```
Bundle Group A                                    Bundle Group A
     |                                                   |
     v                                                   v
  Role R1 <--- User TeamLeader   Resource Group X ---> Role R2 <---
Users TeamMember1, TeamMember2
     ^                                                   ^
     |                                                   |
Create Bundles In Group                         Deploy Bundles To
Group
```

This arrangement divides the create and deploy aspects of the application provisioning cycle.

The *arrangement* of the users in separate roles and the division of the types of permissions is crucial. The same arrangement could be made with global permissions, rather than restricting either or both roles to a single bundle group or resource group, but that division of responsibility would still exist.

[5] There is no defined deployment directory for servers in an EAP 6 domain. Deployments are handled centrally, through other mechanisms.

CHAPTER 28. MANAGING RESOURCE-LEVEL CONTENT UPDATES

JBoss Operations Network can be used to store and deploy content to resources. This can be done to apply updates and patches (as with JBoss AS servers) or to set up repositories used for provisioning applications and deploying custom software.

28.1. ABOUT CONTENT

Content for a resource can be almost anything, such as WAR and EAR files, configuration files, or scripts. JBoss ON provides a central framework to associate content, repositories, and resources in the inventory.

28.1.1. What Content Is: Packages

A *package* is anything that is installed on a platform or for a server or application. This can be a JAR file or even a configuration file. A package simply provides some form of content for a resource. Packages can be sent to a resource through a JBoss ON-recognized repository or simply by uploading the package to the JBoss ON server and then sending it to the resource.

A resource can only be associated with or manage content if the resource plug-in identifies that content is available and the type of content that is supported. For example, application and web servers like JBoss AS/EAP and Tomcat support EAR, WAR, and JAR files as content; but a database like PostgreSQL does not support any content types.

In a sense, content is both the software bits, scripts, or configuration files associated with a resource and also a resource itself. When content is added to a resource, it becomes a child resource in the JBoss ON hierarchy — but it can be managed, reverted, updated, or replaced by uploading new software bits. The parent resource (such as the application server) supports content; the child resource is a *content backed resource*.

Content can be added to a resource either by manually creating a child resource (and uploading the packages) or by adding the package to a content source and deploying it to the parent resource. The agent can also actively *check* for new content as part of its discovery scan and add any discovered content to its inventory. The agent's recurring package discovery scan has a default interval of 24 hours, as with the services scan.

28.1.2. Where Content Comes From: Providers and Repositories

Content sources are developers and distributors of content. Sources can be external third party software developers or internal development teams that create custom content. The type of content available from sources includes both software packages (such as configuration scripts) and updates (version upgrades, patches, and errata).

A *repository* is a user-defined collection of software packages, which can come from one or multiple content sources. A repository may contain packages for an application or family of applications or for a specific purpose, like repositories for laptop configuration and repositories for installing web servers.

Repositories aren't siloed, separate containers for packages; they are essentially views that show a subset of available packages. All packages are stored in the JBoss ON database. A JBoss ON repository is a way of grouping those packages, both to make it easier to administer with resources and to provide a mechanism of access control for users (Section 28.1.4, “Authorization to Repositories and Packages”).

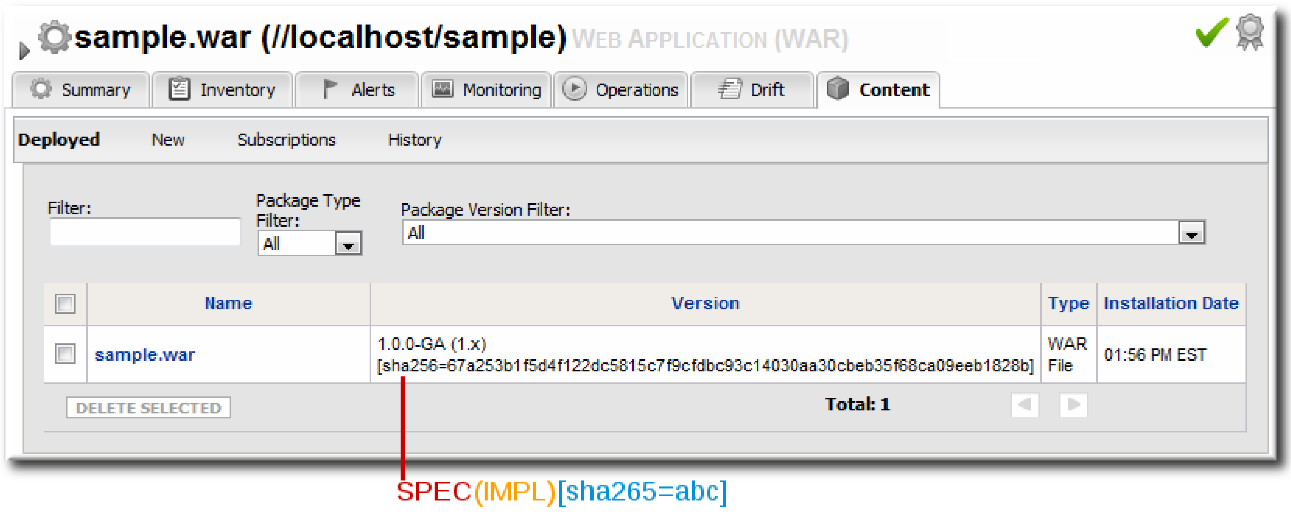
Resources can be subscribed to content repositories that are configured in JBoss ON, which provides a smooth and reliable mechanism for delivering consistent, administrator-configured content to resources.

28.1.3. Package Versions and History

Packages are *versioned* within JBoss ON itself. When a package is added to a resource or content source, the installer prompts for a version number; this is used as the UI display number.

This display version number is not required; if it is not given, then the JBoss ON server derives a number based on a calculated SHA-256 checksum for the package and the specification version and the implementation version in the **META-INF/MANIFEST.MF** file (for EARs and WARs).

```
SPEC(IMPLEMENTATION)[sha256=abcd1234]
```



Filter:

Package Type Filter: All

Package Version Filter: All

	Name	Version	Type	Installation Date
☐	sample.war	1.0.0-GA (1.x) [sha256=67a253b1f5d4f122dc5815c7f9cfdbc93c14030aa30cbeb35f68ca09eeb1828b]	WAR File	01:56 PM EST

Total: 1

Figure 28.1. Package Version Numbers

For example, for a **META-INF/MANIFEST.MF** file with these version numbers:

```
Manifest-Version: 1.0
Created-By: Apache Maven
Specification-Title: My Example App
Specification-Version: 1.0.0-GA
Specification-Vendor: Example, Corp.
Implementation-Title: My Example App
Implementation-Version: 1.x
Implementation-Vendor-Id: org.example
Implementation-Vendor: Example, Corp.
...
```

This creates a version number for the package like this:

```
1.0.0-GA(1.x)[sha256=abcd1234]
```

If the **META-INF/MANIFEST.MF** file does not contain one of the specification version or the implementation version, then only one is used. For example, if only the implementation version is given:

```
(1.x)[sha256=abcd1234]
```

If no version number is given, then the SHA is used as the identifier. (The SHA is used as the identifier internally, anyway.)

```
[sha256=abcd1234]
```

For exploded WARs and EARs, the calculated SHA-256 checksum is added to the **MANIFEST.MF** file. This allows the agent to check the file during discovery scans to verify the version of the package quickly.

```
Manifest-Version: 1.0
Created-By: Apache Maven
RHQ-Sha256: 570f196c4a1025a717269d16d11d6f37
...
```

For unexploded (archived) content, the checksum is recalculated with every package discovery scan and compared to the checksum in inventory.

NOTE

Exploded WARs and EARs can be deployed on JBoss and Tomcat servers. Because the content deployment process edits the **META-INF/MANIFEST.MF** file, the deployed content is not exactly identical to the content packages that were uploaded.

A clear versioning system makes it possible to handle package lifecycles in a clear and effective way. Updated content can be tracked as it is deployed, updates can be applied consistently, and packages can be reverted to a previous version. The same repository can also contain different versions of the same package, making it possible to apply different versions to different resources.

NOTE

Package versions from different content sources can be associated with the same repository.

Whenever a package is installed on a resource, it is recorded in the content history for the resource and the package. Since there can be multiple files associated with a single package, then there can be multiple files recorded in the content history, all associated with that package version.

NOTE

Versioning only matters to content knit with a resource, like EARs and WARs. Other types of content stored in content sources (like CLI scripts used for alerting) do not track versions. Content deployed in bundles handles versioning through the bundle definition, not the content system.

28.1.4. Authorization to Repositories and Packages

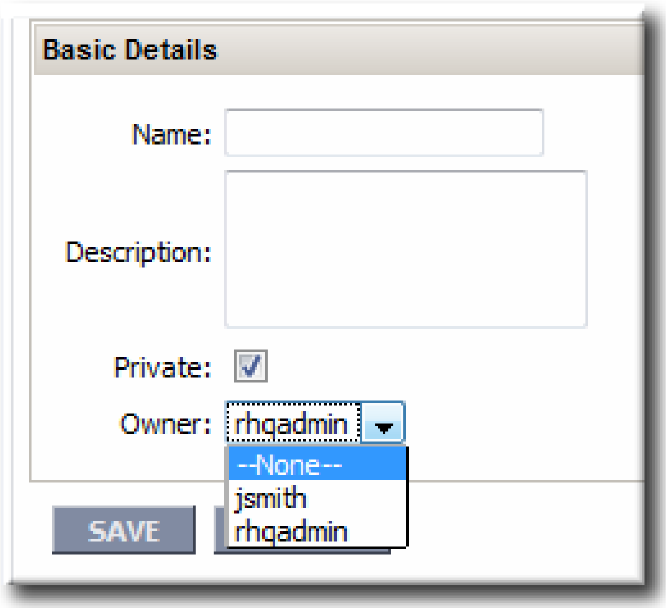
There are a lot of reasons that users need to be able to access content in repositories. The most common is to manage packages on resources, but there are other reasons, too, like using the server CLI scripts in a repository to respond to alerts.

JBoss ON provides a way to balance the need for clear and simple access to content with the need to protect private or sensitive information. JBoss ON defines clear *authorization* rules for content repositories.

Every user has the ability to create repositories and to upload packages to them — regardless of the permissions for that user.

When a repository is created, there are settings which control access to them:

- *Owner* sets write access to a repository. It assigns the repository to belong to a specific user. If no user is specified, then only users with the manage repositories permission have the right to access those repositories.
- *Private* sets read (download) access to the repository. It sets whether the repository can be viewed by anyone or only by the owner and users with the manage repositories permission. Public repositories are viewable by everyone, regardless of the owner.


Basic Details

Name:

Description:

Private: ✓

Owner: rhqadmin

Figure 28.2. Repository Ownership and Access Settings

Repo managers (users with the manage repositories permission) can change the ownership and privacy settings of a repository. Users without the manage repositories permission can change the privacy settings but they cannot change the ownership; the repository is always owned by them or managed by the repo manager.

NOTE

Be very careful when switching public repositories to private. Any operations which relied on those repositories, such as running server CLI scripts in response to alerts, will no longer work if the privileges of the user are insufficient to access the repository.

JBoss ON uses the *repositories* access control permission to define users with administrative access to repositories. Any user with that permission can manage any configured repository, regardless of who the repository's owner is. Repositories without an owner can only be managed by users with the repositories permission. Lastly, only users with this permission can associate a content source with a repository; all other users must add packages to the repository manually.

28.1.5. Space Considerations for Content

Content can have a significant impact on disk space requirements.

JBoss ON stores all versions of content. This is part of versioning control, allowing changes to content-backed resources to be reverted and managed and for different versions to be deployed at different times.

Therefore, the system which hosts the backend database (Oracle or PostgreSQL) must have enough disk space to store all versions of all bundles. Additionally, the database itself must have adequate tablespace for the content.

When calculating the required amount of space, estimate the size of every artifact, and then the number of versions for each artifact. At a minimum, **have twice that amount of space available**; both PostgreSQL and Oracle require twice the database size to perform cleanup operations like vacuum, compression, and backup and recovery.

28.2. CREATING A CONTENT SOURCE

A content source is whatever mechanism supplies software packages to JBoss ON and, through JBoss ON's content management, to resources. JBoss ON supports several different types of content sources.

Table 28.1. Types of Content Sources

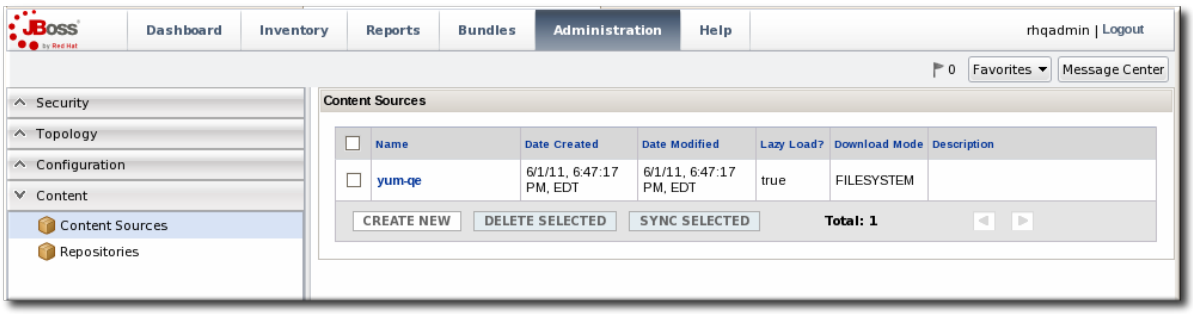
Source Type	Description	Requires Credentials?
Remote URL	Downloads from a remote URL. This can use a couple of different protocols, such as FTP.	No
HTTP	Similar to the Remote URL content source, connects over a network connection to the source. This uses specifically the HTTP protocol. The HTTP content source can also connect to an HTTP proxy. HTTPS is not supported.	Optionally allows credentials to log into the given HTTP site or the proxy server[a]
JBoss Customer Portal Feed (RSS)	Similar to the Remote URL content source, except that it works specifically with the Customer Portal RSS feed for JBoss cumulative patches.	Yes[a]
Local Disk	Connects to a single local directory (on the local system or NFS-mounted) and looks for packages of the specified type and architecture to download.	No

Source Type	Description	Requires Credentials?

[a] Any passwords given in the content source configuration are obfuscated in the JBoss ON database.

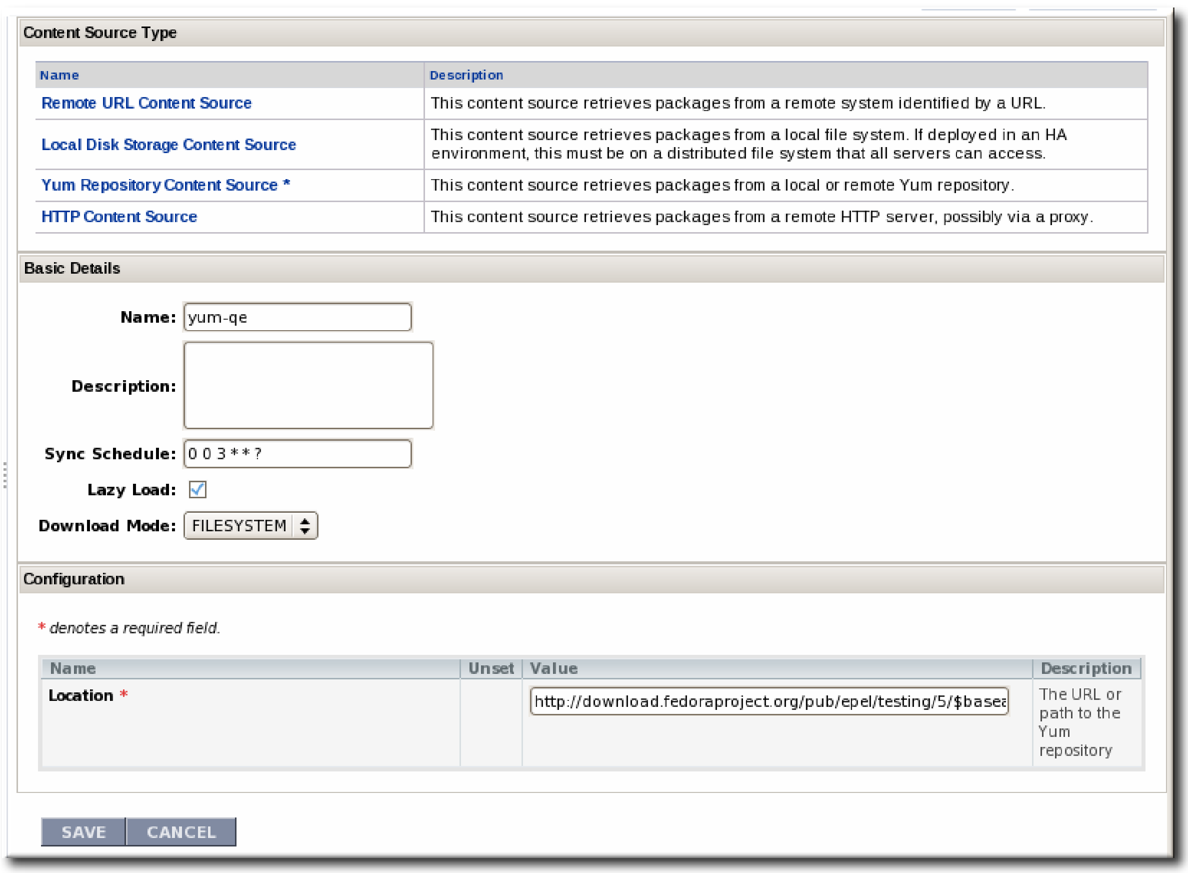
28.2.1. Creating a Content Source (General)

1. In the top menu, click the **`Administration`** tab.
2. In the **`Content`** menu table on the left, select the **Content Sources** item.
3. Below the list of current content sources, click the **`CREATE NEW`** button.



Name	Date Created	Date Modified	Lazy Load?	Download Mode	Description
yum-qe	6/1/11, 6:47:17 PM, EDT	6/1/11, 6:47:17 PM, EDT	true	FILESYSTEM	

4. Select the content source type, which defines how the content is delivered from the source. Table 28.1, “Types of Content Sources” describes the different content sources.
5. When the content source type is selected, a form automatically opens to fill in the basic details and configuration for the resource. These basic details identify the content source in the JBoss ON server and are the same for each content source type, while the configuration is specific to the content source type.


Content Source Type

Name	Description
Remote URL Content Source	This content source retrieves packages from a remote system identified by a URL.
Local Disk Storage Content Source	This content source retrieves packages from a local file system. If deployed in an HA environment, this must be on a distributed file system that all servers can access.
Yum Repository Content Source *	This content source retrieves packages from a local or remote Yum repository.
HTTP Content Source	This content source retrieves packages from a remote HTTP server, possibly via a proxy.

Basic Details

Name:	yum-qe
Description:	
Sync Schedule:	0 0 3 * * ?
Lazy Load:	✓
Download Mode:	FILESYSTEM

Configuration

* denotes a required field.

Name	Unset	Value	Description
Location *		http://download.fedoraproject.org/pub/epel/testing/5/$basea	The URL or path to the Yum repository

- Give a unique name and optional description for the content source provider.
- The schedule sets how frequently the content in the JBoss ON database is updated by the content source; this uses a standard Quartz Cron Syntax.
- The lazy load setting sets whether to download packages only when they are installed (**Yes**) or if all packages should be download immediately.
- The download mode sets how the content is stored in JBoss ON. The default is **DATABASE**, which stores all packages in the JBoss ON database instance. The other options are to store the packages on a network filesystem or not to store them at all.

6. Fill in the other configuration information for the content source. The required information varies depending on the content source type. This is going to require some kind of connection information, such as a URL or directory path, and possibly authentication information, like a username and password.

NOTE

Any passwords stored for content sources are obfuscated in the JBoss ON database.

28.2.2. Creating a Content Source (Local Disk)

A local disk content source is set up more or less as described in Section 28.2.1, “Creating a Content Source (General)”, but the values for both the content source setup *and* the repository setup are critical for content synchronization to work.

A single content source can be associated with multiple repositories, and this is true for local disk configuration. For local disk providers, the content source defines a root directory, and then the repository name identifies the subdirectory which contains the packages.

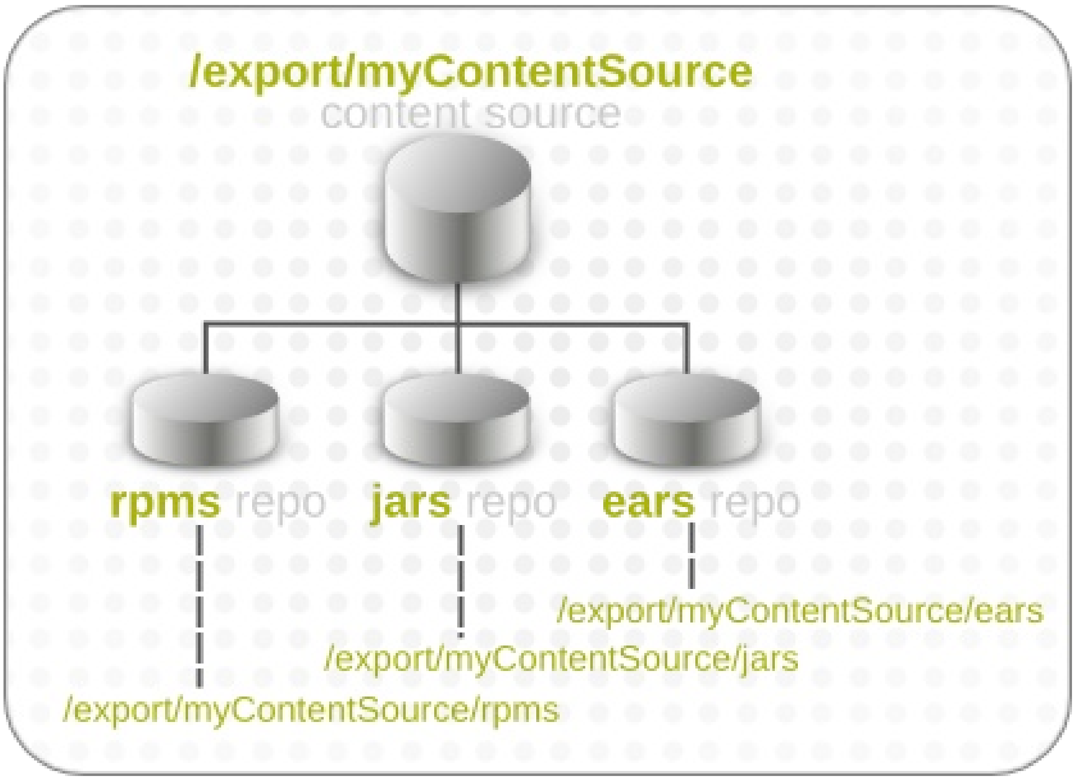

/export/myContentSource
content source
rpms repo
jars repo
ears repo
/export/myContentSource/ears
/export/myContentSource/jars
/export/myContentSource/rpms

Figure 28.3. Local Disk Structure

This structure allows multiple repositories to use the same base directory in the content provider.

JBoss ON derives the information for the local disk based on the combination of the content source configuration (root directory) and the repository configuration (subdirectory). **For the sync to work, the repository must have the identical name as the subdirectory which contains the packages.**

IMPORTANT

Each subdirectory name must be unique through the hierarchy of the root directory tree. For example, there should not be directories named `/export/myContentSource/test` and `/export/myContentSource/subdir/test`.

Having two directories, even at different levels, with the same name can result in unpredictable package sync behavior.

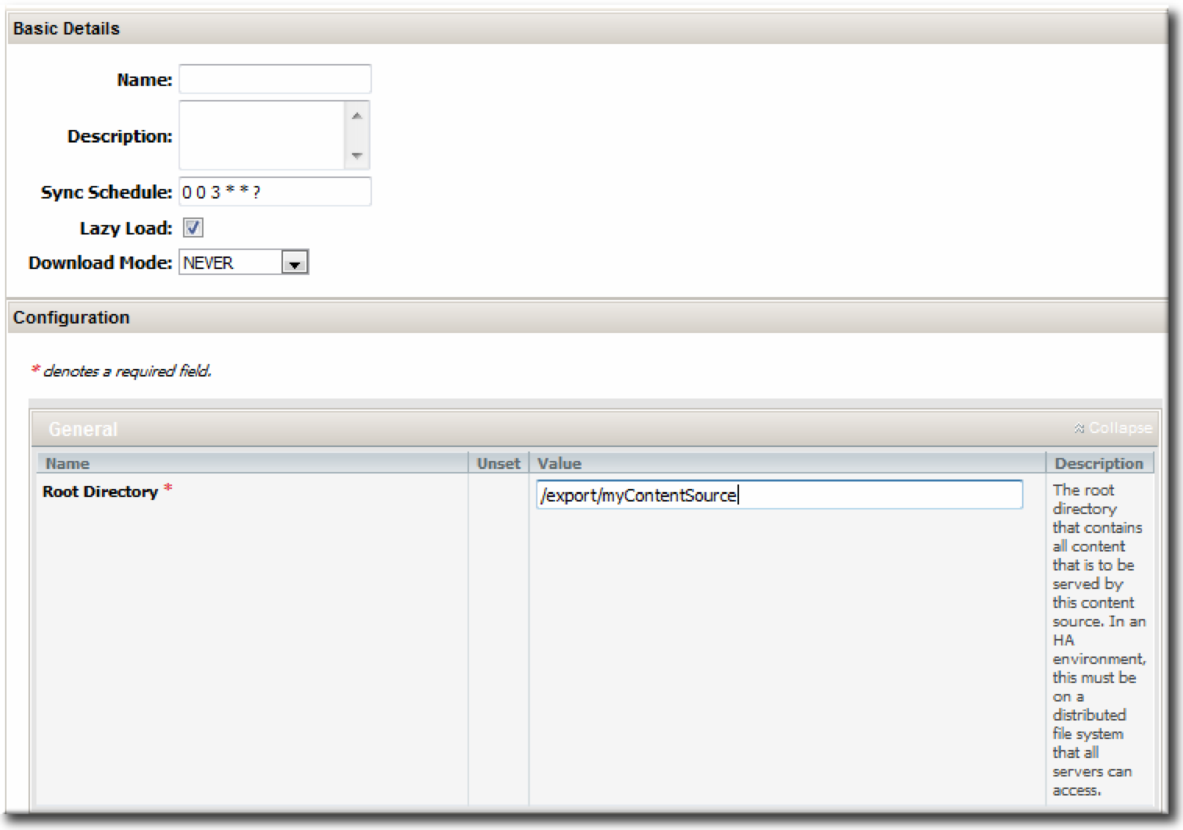
To set up the local disk provider:

1. Set up the content source as in Section 28.2.1, “Creating a Content Source (General)”.

   NOTE

   If the subdirectories to sync already exist, then the content source configuration prompts for possible repositories to associate with the local disk provider based on the subdirectory names.

2. Enter the root directory path.



Basic Details

Name:	
Description:	
Sync Schedule:	0 0 3 * * ?
Lazy Load:	✓
Download Mode:	NEVER

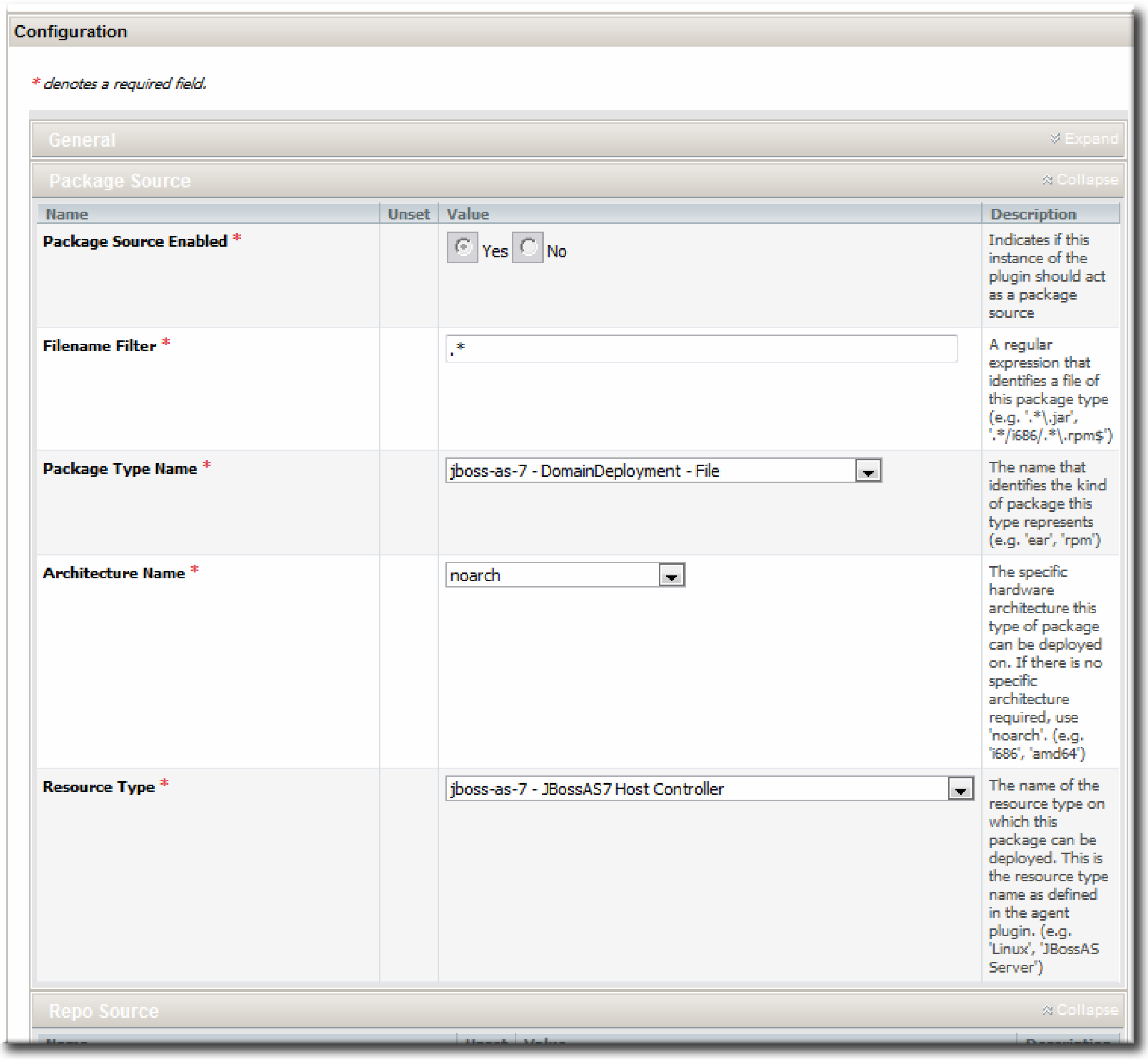
Configuration

* denotes a required field.

General

Name	Unset	Value	Description
Root Directory *		/export/myContentSource	The root directory that contains all content that is to be served by this content source. In an HA environment, this must be on a distributed file system that all servers can access.

3. Enter the content package information, which the JBoss ON server uses to identify the packages to pull into the content storage.



Configuration

* denotes a required field.

General — Expand

Package Source — Collapse

Name	Unset	Value	Description
Package Source Enabled *		[✓ Yes] No	Indicates if this instance of the plugin should act as a package source
Filename Filter *		.*	A regular expression that identifies a file of this package type (e.g. '.*\.jar', '.*/i686/.*\.rpm$')
Package Type Name *		jboss-as-7 - DomainDeployment - File	The name that identifies the kind of package this type represents (e.g. 'ear', 'rpm')
Architecture Name *		noarch	The specific hardware architecture this type of package can be deployed on. If there is no specific architecture required, use 'noarch'. (e.g. 'i686', 'amd64')
Resource Type *		jboss-as-7 - JBossAS7 Host Controller	The name of the resource type on which this package can be deployed. This is the resource type name as defined in the agent plugin. (e.g. 'Linux', 'JBossAS Server')

Repo Source — Collapse

4. Create the repository, as in Section 28.3.1, “Creating a Repository” , and give it the name of the subdirectory to use.

IMPORTANT

Each subdirectory name must be unique through the hierarchy of the root directory tree. For example, there should not be directories named `/export/myContentSource/test` and `/export/myContentSource/subdir/test`.

Having two directories, even at different levels, with the same name can result in unpredictable package sync behavior.

5. Create the subdirectory on the local system, and copy in the packages which should be added to the JBoss ON content system.

28.3. MANAGING REPOSITORIES

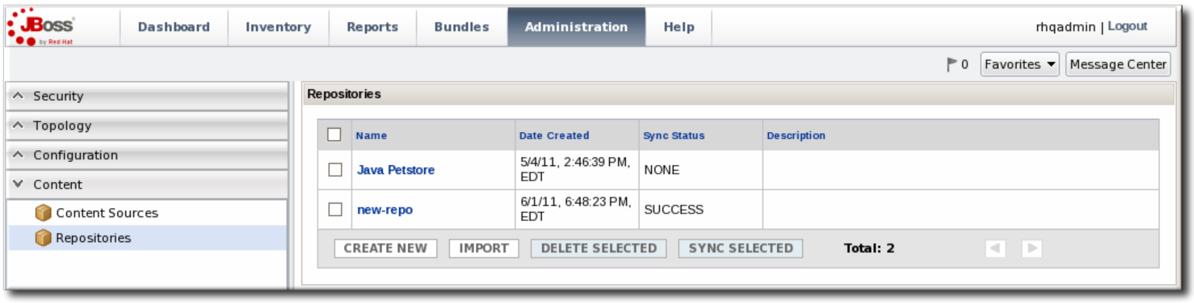
A repository is essentially a mapping between the data in a content source and specific resources in the JBoss ON inventory.

28.3.1. Creating a Repository

1. In the top menu, click the **`Administration`** tab.

2. In the **`Content`** menu table on the left, select the **Repositories** item.

3. Below the list of current repositories, click the **CREATE NEW** button.



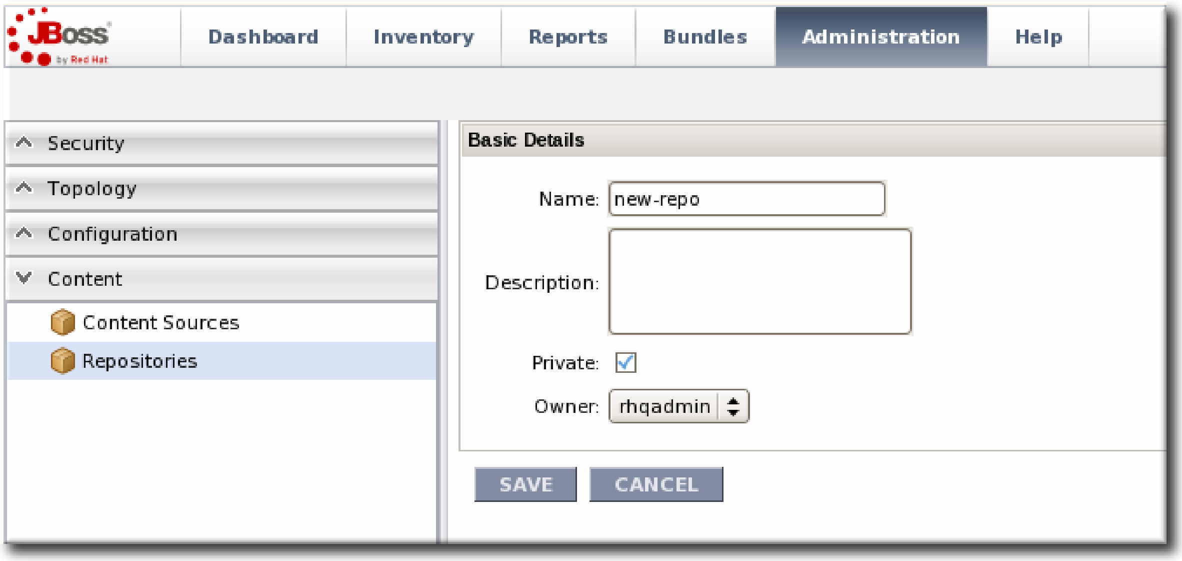
Repositories

	Name	Date Created	Sync Status	Description
☐	Java Petstore	5/4/11, 2:46:39 PM, EDT	NONE	
☐	new-repo	6/1/11, 6:48:23 PM, EDT	SUCCESS	

Total: 2

4. Fill in the name and a description. Additionally, set the authorization restrictions for the repository by setting an owner for the repo and whether it is public or private.

   Only users with the repositories permission can set an owner. All repositories created by users without the repositories permission automatically belong to that user.



Basic Details

Name:	new-repo
Description:	
Private:	✓
Owner:	rhqadmin

5. Click **Save**.

6. On the **`Repositories`** page, click the name of the new repository in the list.

7. *Optional*. To change the default synchronization schedule, click the **`Edit`** button and enter a new schedule, in a cron format, in the **`Sync Schedule`** field.

8. Add content sources to supply content to the repository, as in Section 28.3.2.1, "Associating Content Sources with a Repository".

   More than one content source can supply content to a repository.

9. Associate resources with the repository, as in Section 28.3.3, "Associating Resources with the Repository". A resource can only receive packages from a repository if it is associated with the repository.

NOTE

You can search for specific resources or types of resources and subscribe multiple resources at once.

28.3.2. Linking Content Sources to Repositories

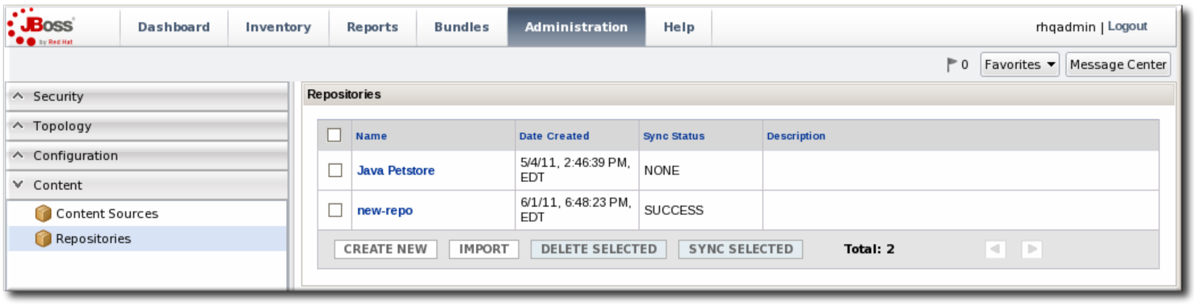
There are a couple of ways to map the repositories to the right content sources. A repository can be subscribed to multiple content sources by editing the repository configuration. A content source can be added to multiple repositories simultaneously by importing the content source.

28.3.2.1. Associating Content Sources with a Repository

1. In the top menu, click the **`Administration`** tab.

2. In the **`Content`** menu table on the left, select the **Repositories** item.

3. On the **`Repositories`** page, click the name of the repository in the list.

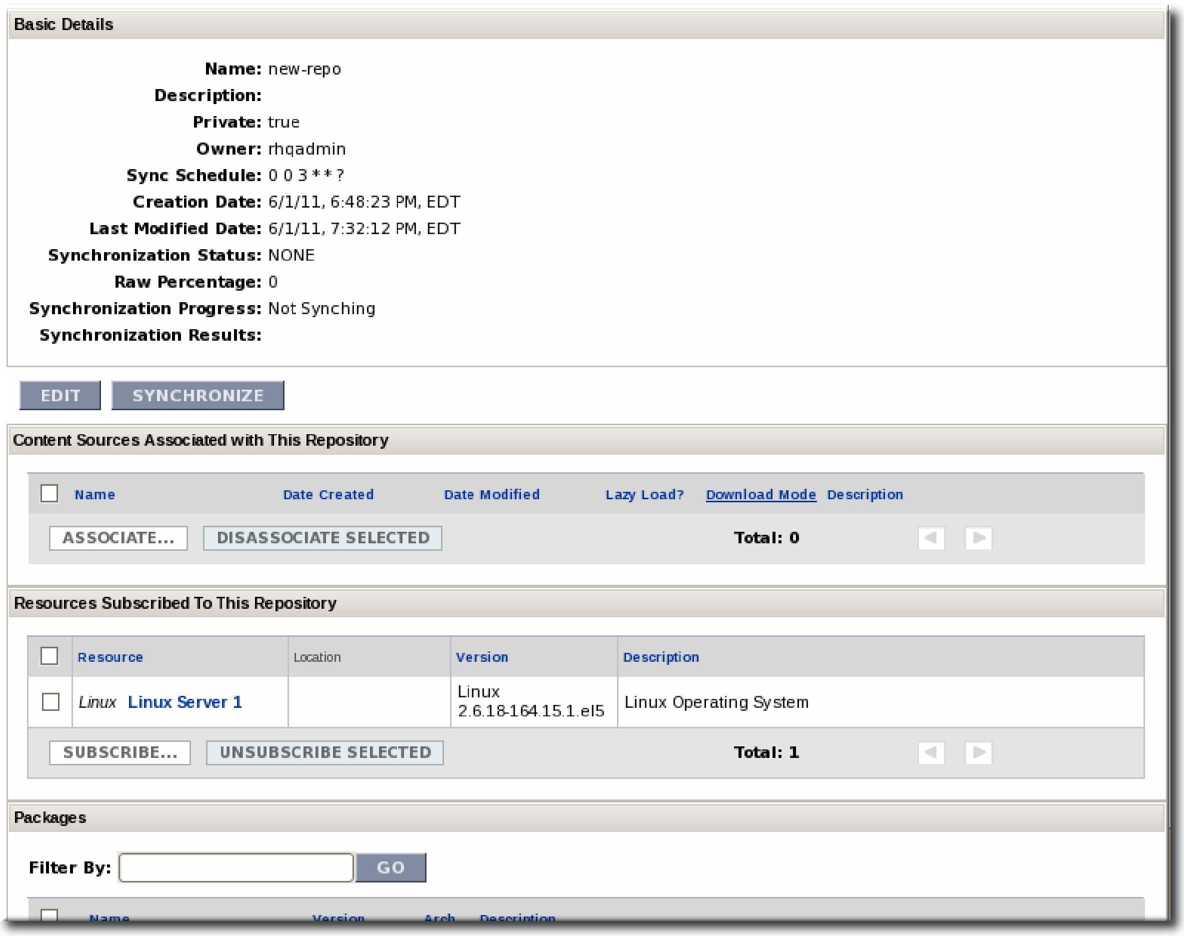


Repositories

	Name	Date Created	Sync Status	Description
☐	Java Petstore	5/4/11, 2:46:39 PM, EDT	NONE	
☐	new-repo	6/1/11, 6:48:23 PM, EDT	SUCCESS	

Total: 2

4. In the **`Content Sources`** section of the repository's details page, click the **`Associate`** button to add existing content sources to the repository.



Basic Details

Name:	new-repo
Description:	
Private:	true
Owner:	rhqadmin
Sync Schedule:	0 0 3 * * ?
Creation Date:	6/1/11, 6:48:23 PM, EDT
Last Modified Date:	6/1/11, 7:32:12 PM, EDT
Synchronization Status:	NONE
Raw Percentage:	0
Synchronization Progress:	Not Synching
Synchronization Results:	

Content Sources Associated with This Repository

	Name	Date Created	Date Modified	Lazy Load?	Download Mode	Description

Total: 0

Resources Subscribed To This Repository

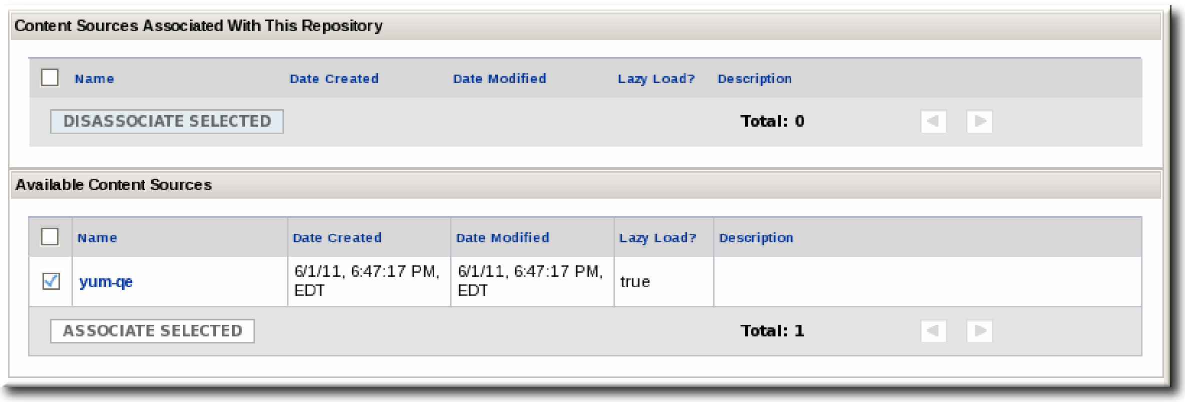
	Resource	Location	Version	Description
☐	Linux Linux Server 1		Linux 2.6.18-164.15.1.el5	Linux Operating System

Total: 1

Packages

Filter By:

5. Select checkboxes next to the content sources to associate with the repository.



Content Sources Associated With This Repository

	Name	Date Created	Date Modified	Lazy Load?	Description

Total: 0

Available Content Sources

	Name	Date Created	Date Modified	Lazy Load?	Description
✓	yum-qe	6/1/11, 6:47:17 PM, EDT	6/1/11, 6:47:17 PM, EDT	true	

Total: 1

6. Click the **ASSOCIATE SELECTED** button.

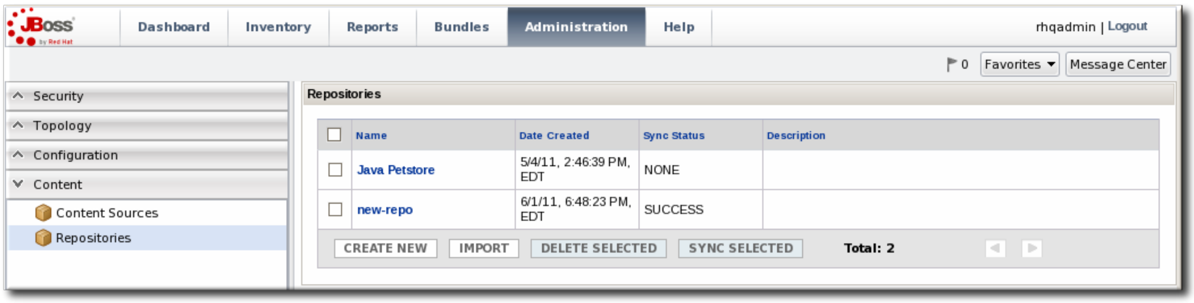
28.3.2.2. Importing a Content Source into Repositories

If the same content source will be associated with multiple repositories, the content source can be imported into all of them simultaneously.

1. In the top menu, click the **Administration** tab.

2. In the **Content** menu table on the left, select the **Repositories** item.

3. On the **Repositories** page, click the **IMPORT** button.



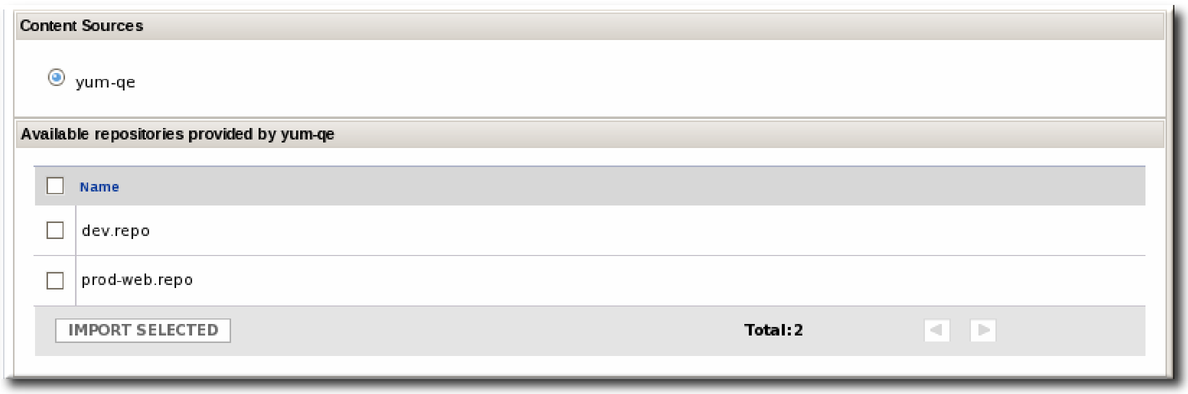
Repositories

	Name	Date Created	Sync Status	Description
☐	Java Petstore	5/4/11, 2:46:39 PM, EDT	NONE	
☐	new-repo	6/1/11, 6:48:23 PM, EDT	SUCCESS	

Total: 2

4. Select the radio button by the name of the content source to import.

5. When the content source is selected, then a list of available repositories for that content source automatically opens. In the **Available repositories....** area, select the checkbox by the name of each repository to associate with the content source.



Content Sources

yum-qe

Available repositories provided by yum-qe

	Name
☐	dev.repo
☐	prod-web.repo

Total:2

6. Click the **IMPORT SELECTED** button.

NOTE

As described in Section 28.1.2, “Where Content Comes From: Providers and Repositories”, a repository is a user-defined view of a subset of packages stored in the JBoss ON database. A repository is not a separate container.

When adding a package to one repository through the UI, it may fail with an error claiming that the package already exists, even if the package isn't in the specified repository. This is because a package with the same name exists in *another* repository and it causes a collision in the database.

It is currently not possible to have the same package in two repositories or to move or share a package between repositories.

It is possible to work around this issue by using CLI scripts. The JBoss ON CLI scripts store the username of the person uploading the package in the package version data automatically. If a person has access to all of the packages one has uploaded, then it is possible to extrapolate which repository contains the package and then manage the package there.

28.3.3. Associating Resources with the Repository

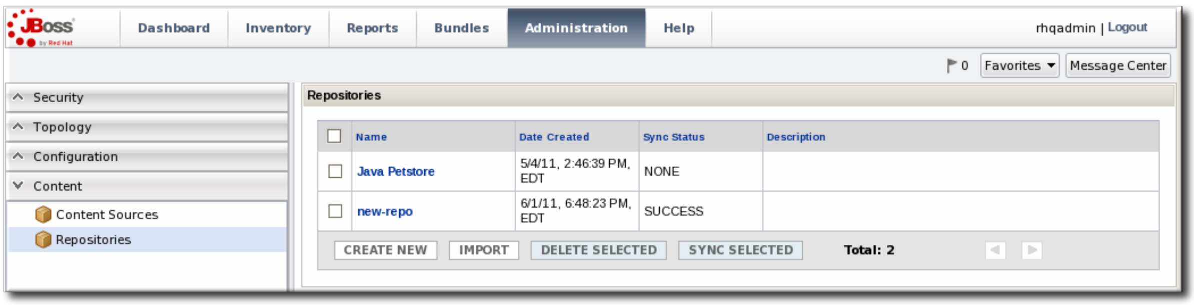
Content can only be sent to a resource if that resource is first associated with a repository. A resource-repository association can be made by editing the resource entry or by editing the repository entry.

28.3.3.1. Adding Resources to a Repository

1. In the top menu, click the **`Administration`** tab.

2. In the **`Content`** menu table on the left, select the **Repositories** item.

3. On the **`Repositories`** page, click the name of the repository to edit.

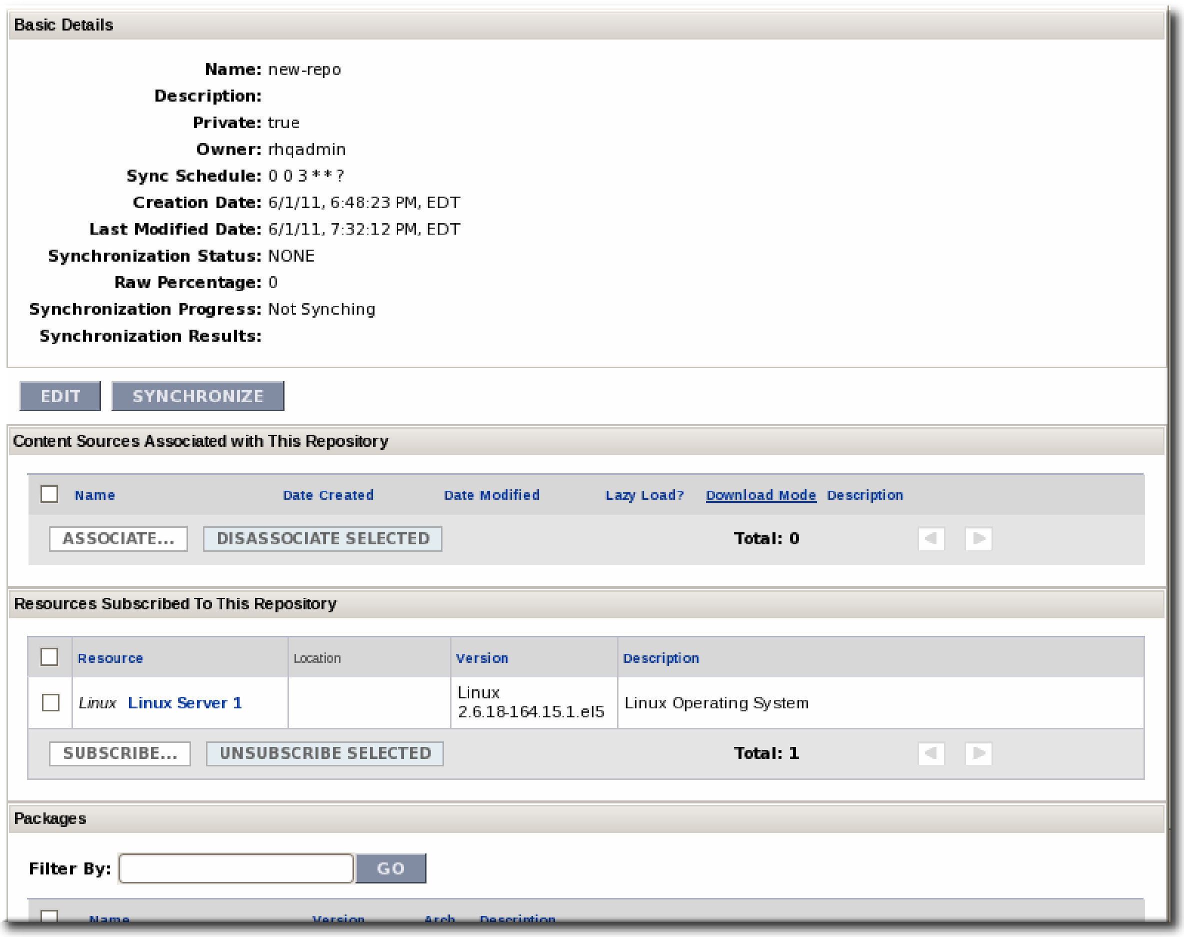


Repositories

	Name	Date Created	Sync Status	Description
☐	Java Petstore	5/4/11, 2:46:39 PM, EDT	NONE	
☐	new-repo	6/1/11, 6:48:23 PM, EDT	SUCCESS	

Total: 2

4. In the **`Resources`** section, click the **`SUBSCRIBE`** button to add resources to the repository.



Basic Details

Name: new-repo
Description:
Private: true
Owner: rhqadmin
Sync Schedule: 0 0 3 * * ?
Creation Date: 6/1/11, 6:48:23 PM, EDT
Last Modified Date: 6/1/11, 7:32:12 PM, EDT
Synchronization Status: NONE
Raw Percentage: 0
Synchronization Progress: Not Synching
Synchronization Results:

Content Sources Associated with This Repository

	Name	Date Created	Date Modified	Lazy Load?	Download Mode	Description

Total: 0

Resources Subscribed To This Repository

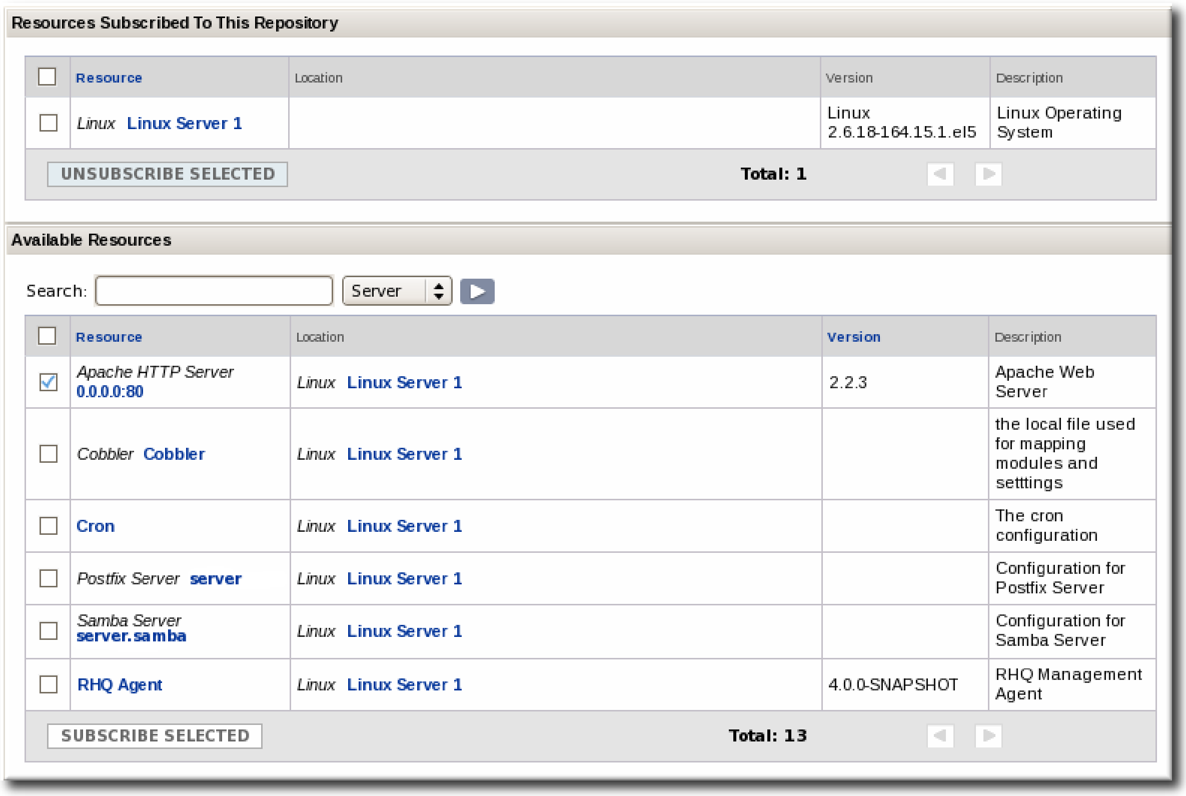
	Resource	Location	Version	Description
☐	Linux Linux Server 1		Linux 2.6.18-164.15.1.el5	Linux Operating System

Total: 1

Packages

Filter By:

5. Select checkboxes next to the resources to associate with the repository. It is possible to filter the list of resources by name or by type.



Resources Subscribed To This Repository

☐	Resource	Location	Version	Description
☐	*Linux* Linux Server 1		Linux 2.6.18-164.15.1.el5	Linux Operating System

UNSUBSCRIBE SELECTED Total: 1

Available Resources

Search: [ ] Server

☐	Resource	Location	Version	Description
✓	*Apache HTTP Server* 0.0.0.0:80	*Linux* Linux Server 1	2.2.3	Apache Web Server
☐	*Cobbler* Cobbler	*Linux* Linux Server 1		the local file used for mapping modules and setttings
☐	Cron	*Linux* Linux Server 1		The cron configuration
☐	*Postfix Server* server	*Linux* Linux Server 1		Configuration for Postfix Server
☐	*Samba Server* server.samba	*Linux* Linux Server 1		Configuration for Samba Server
☐	RHQ Agent	*Linux* Linux Server 1	4.0.0-SNAPSHOT	RHQ Management Agent

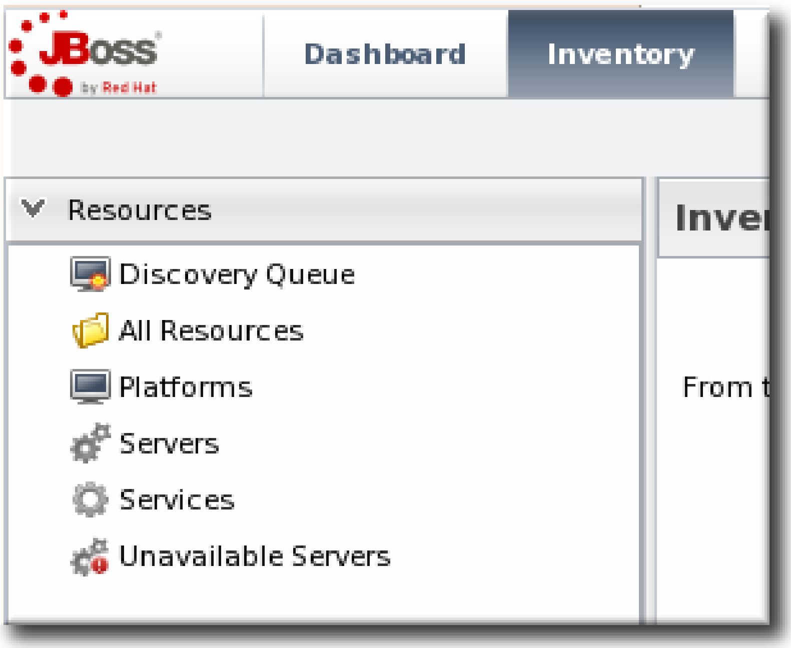
SUBSCRIBE SELECTED Total: 13

6. Click the **SUBSCRIBE SELECTED** button.

28.3.3.2. Managing the Repositories for a Resource

A few resource types, like platforms, have content tabs in their configuration which allows them to control their content subscriptions.

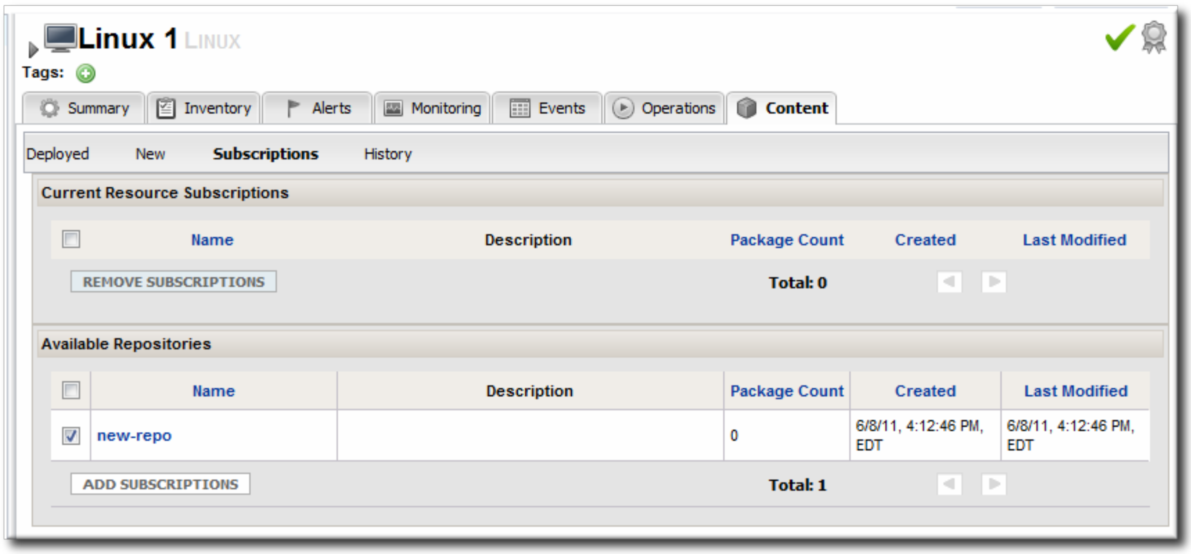
1. Select the resource type in the **Resources** menu table on the left, and then browse or search for the resource.



2. Click the **Content** tab of the resource.

3. Open the **Subscriptions** subtab.

4. The **Available Repositories** section has a list of repositories that the resource isn't subscribed to. Click the checkboxes by all of the repositories to subscribe the resource to.



Current Resource Subscriptions

☐	Name	Description	Package Count	Created	Last Modified

Total: 0

Available Repositories

☐	Name	Description	Package Count	Created	Last Modified
☑	new-repo		0	6/8/11, 4:12:46 PM, EDT	6/8/11, 4:12:46 PM, EDT

Total: 1

5. Click **ADD SUBSCRIPTIONS**.

The same process can be used to unsubscribe a resource from content repositories.

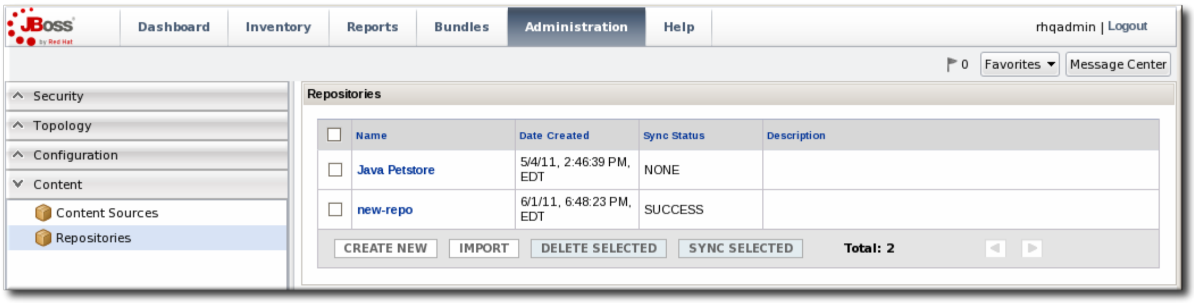
28.4. UPLOADING PACKAGES

Packages can be pulled from a content source, but individual packages can also be uploaded directly to the JBoss ON server. A variety of package types are supported, including JAR files, basic scripts, JBoss ON CLI scripts, and patches.

1. In the top menu, click the **Administration** tab.

2. In the **Content** menu table on the left, select the **Repositories** item.

3. On the **Repositories** page, click the name of the repository in the list.



Repositories

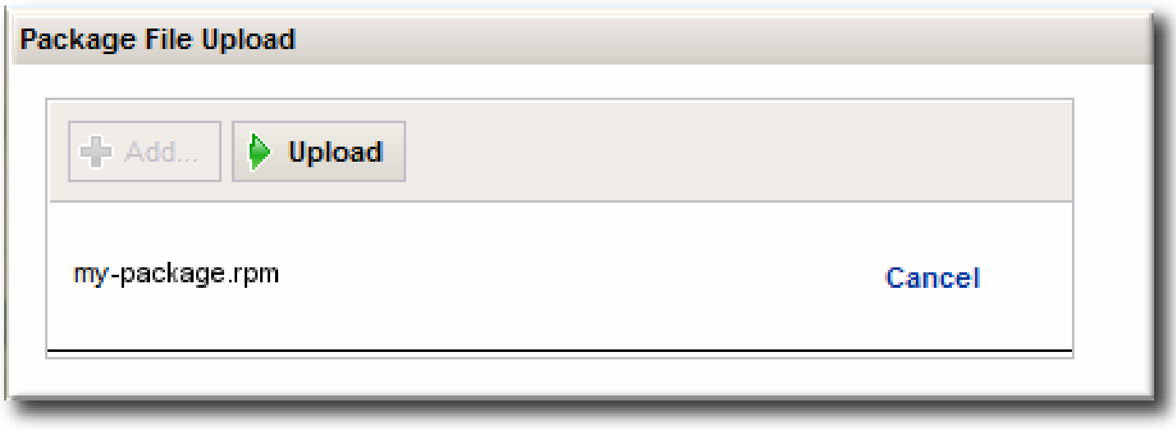
☐	Name	Date Created	Sync Status	Description
☐	Java Petstore	5/4/11, 2:46:39 PM, EDT	NONE	
☐	new-repo	6/1/11, 6:48:23 PM, EDT	SUCCESS	

Total: 2

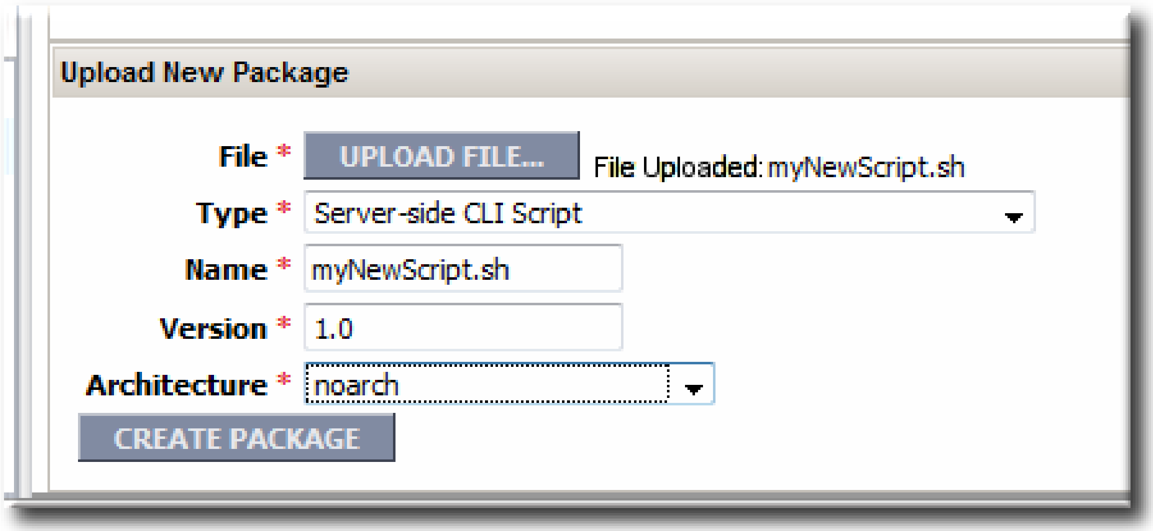
4. Scroll to the bottom of the page, to the **Upload Packages** section.

5. Click the **Upload File** button to upload the package.

6. In the pop-up window, click the **Add** button to browse to the package, then click the **Upload** button.



7. Some information about the package is automatically filled in, including the name and a default UI version number. Set the package type, architecture, and any other necessary information.



File * File Uploaded: myNewScript.sh
Type * Server-side CLI Script
Name * myNewScript.sh
Version * 1.0
Architecture * noarch

If a version number is set, then this value is displayed in the UI. If not, then a version number is calculated, based on the spec version and implementation version in **MANIFEST.MF** (for EARs and WARs) or the calculated SHA-256 value for the package itself. Internally, the package is identified by the SHA value.

```
SPEC(IMPLEMENTATION)[sha256=abcd1234]
```

NOTE

For exploded content for EARs and WARs, the calculated SHA-256 version number is written into the **MANIFEST.MF** file.

8. Click the **CREATE PACKAGE** button to finish adding the package to the repository.

28.5. SYNCHRONIZING CONTENT SOURCES OR REPOSITORIES

The original source of content is external to JBoss ON, and the content packages are pulled into JBoss ON and stored. Any changes that are made at the original content source need to be pulled into JBoss ON by *synchronizing* the two sources.

Likewise, any changes in the content source are carried over to the repository when the source and repository are synchronized.

28.5.1. Scheduling Synchronization

Synchronization is already scheduled in the content source entry in JBoss ON. This schedule has the standard cron format.

```
*     *     *     *      *  [sync-command]
-     -     -     -      -
|     |     |     |      |
|     |     |     |      +----- Day of Week (0=Sunday ... 6=Saturday)
|     |     |     +------- Month (1 - 12)
|     |     +--------- Date (1 - 31)
|     +----------- Hour (0 - 23)
+------------- Minute (0 - 59)
```

For example, to synchronize the source with JBoss ON on Tuesday and Friday at 3am:

```
0 3 * * 2,5
```

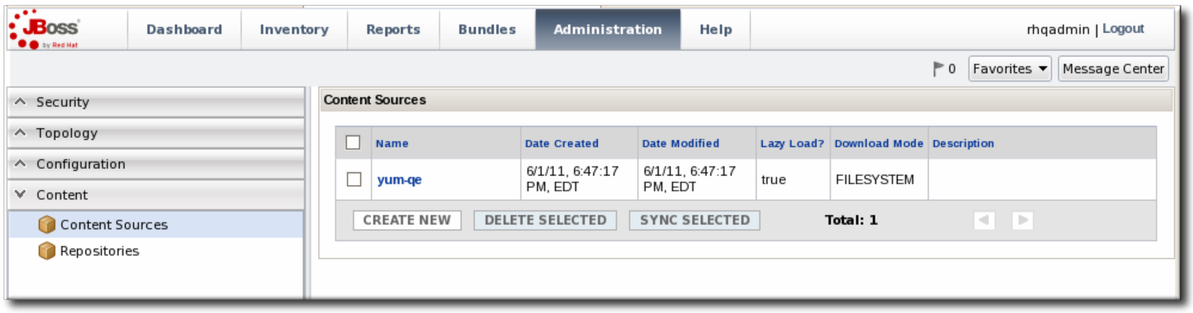
The Quartz documentation explains the cron syntax in more detail.

To edit the schedule synchronization times for a source:

1. In the top menu, click the **Administration** tab.

2. In the **Content** menu table on the left, select either the **Content Sources** or **Repositories** item.

3. Click the name of the item to edit.

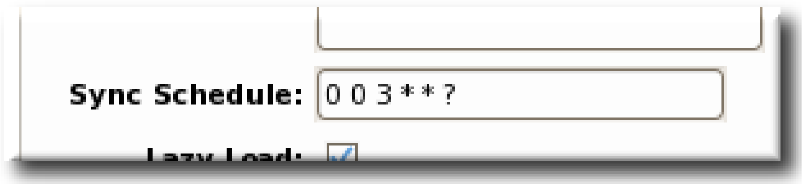


Content Sources

	Name	Date Created	Date Modified	Lazy Load?	Download Mode	Description
☐	yum-qe	6/1/11, 6:47:17 PM, EDT	6/1/11, 6:47:17 PM, EDT	true	FILESYSTEM	

Total: 1

4. Reset the cron schedule in the **Sync Schedule** field.


Sync Schedule: 0 0 3 * * ?

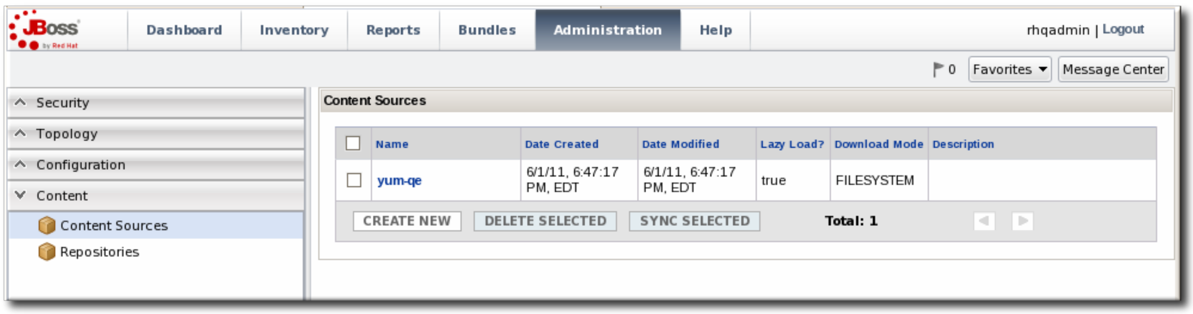
Lazy Load: ✓

5. Click **Save**.

28.5.2. Manually Synchronizing Content Sources or Resources

If a major change happens to the content source, then the changes can be manually sent over to the JBoss ON server by initiating a synchronization manually.

1. In the top menu, click the **`Administration`** tab.
2. In the **`Content`** menu table on the left, select the **Content Sources** or **Repositories** item.
3. Click the name of the item to edit.

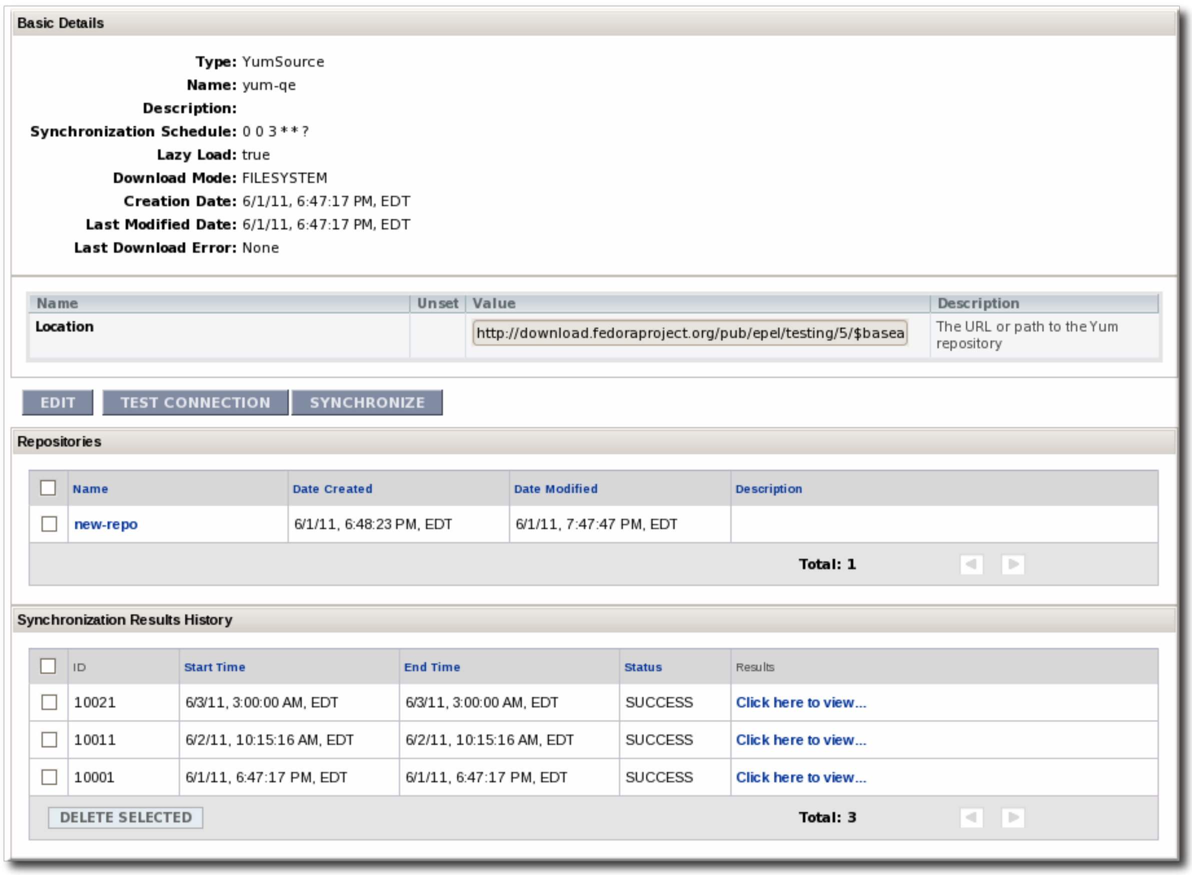


Content Sources

	Name	Date Created	Date Modified	Lazy Load?	Download Mode	Description
☐	yum-qe	6/1/11, 6:47:17 PM, EDT	6/1/11, 6:47:17 PM, EDT	true	FILESYSTEM	

Total: 1

4. Click the **`Synchronize`** button. All of the synchronization attempts, with the outcome of the operation, are listed at the bottom of the screen.



Basic Details

Type:	YumSource
Name:	yum-qe
Description:	
Synchronization Schedule:	0 0 3 * * ?
Lazy Load:	true
Download Mode:	FILESYSTEM
Creation Date:	6/1/11, 6:47:17 PM, EDT
Last Modified Date:	6/1/11, 6:47:17 PM, EDT
Last Download Error:	None

Name	Unset	Value	Description
Location		http://download.fedoraproject.org/pub/epel/testing/5/$basea	The URL or path to the Yum repository

Repositories

	Name	Date Created	Date Modified	Description
☐	new-repo	6/1/11, 6:48:23 PM, EDT	6/1/11, 7:47:47 PM, EDT	

Total: 1

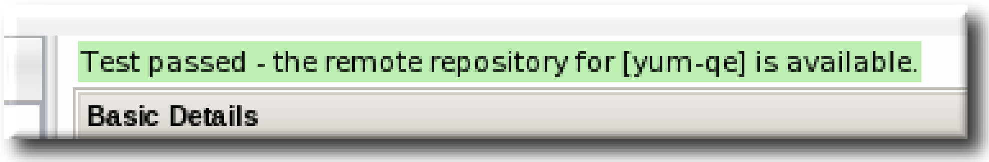
Synchronization Results History

	ID	Start Time	End Time	Status	Results
☐	10021	6/3/11, 3:00:00 AM, EDT	6/3/11, 3:00:00 AM, EDT	SUCCESS	Click here to view...
☐	10011	6/2/11, 10:15:16 AM, EDT	6/2/11, 10:15:16 AM, EDT	SUCCESS	Click here to view...
☐	10001	6/1/11, 6:47:17 PM, EDT	6/1/11, 6:47:17 PM, EDT	SUCCESS	Click here to view...

Total: 3

NOTE

You can test the connection to a source or repository by clicking the `Test Connection` button. This ensures that the JBoss ON server can connect to the content source before attempting to pull down the packages.

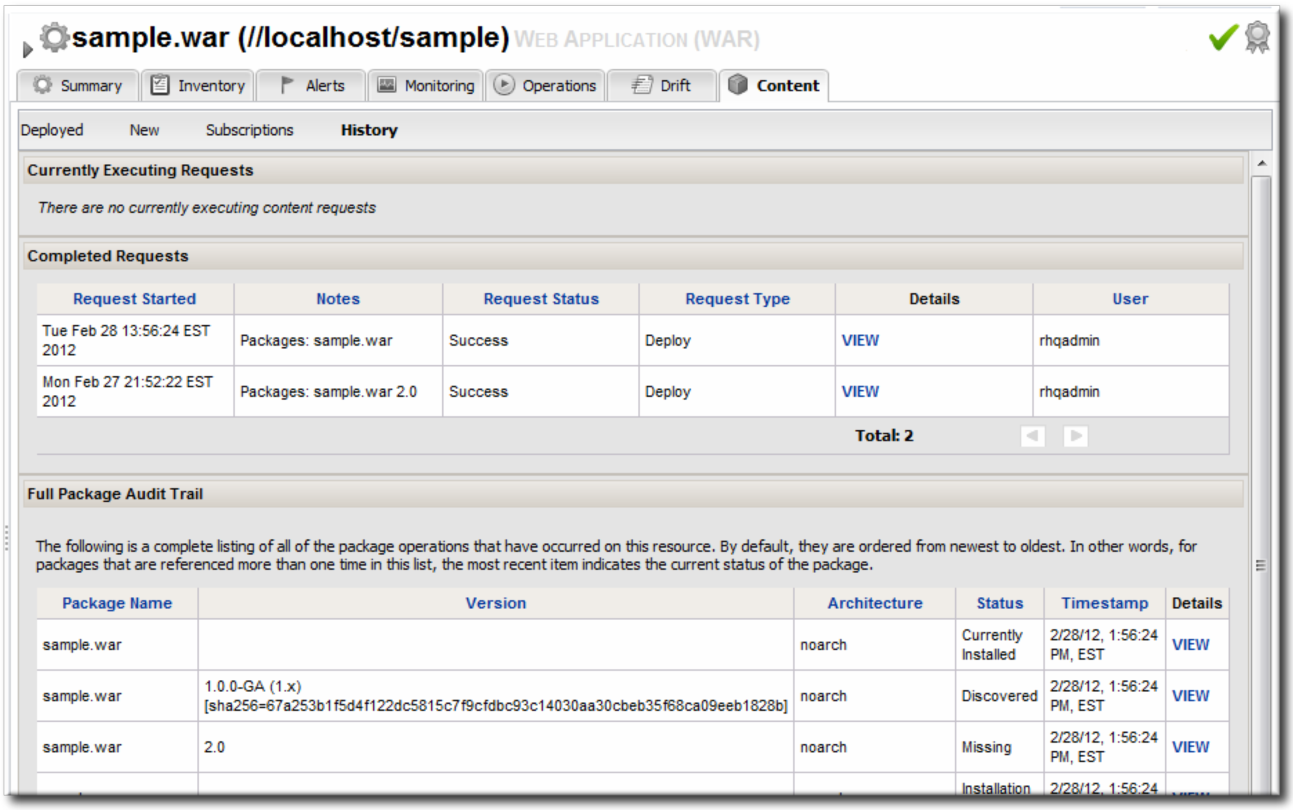

Test passed - the remote repository for [yum-qe] is available.

Basic Details

To synchronize multiple sources, stay on the main content sources or repositories page, select the checkbox by each of the content sources to synchronize, and click the `Sync Selected` button.

28.6. TRACKING CONTENT VERSIONS FOR A RESOURCE

Every time a package is installed on a resource through a repository, the resource shows the operation. This includes even installation failures. The content package history for a resource is viewable in the `Content` tab, under the `History` subtab.



Currently Executing Requests

There are no currently executing content requests

Completed Requests

Request Started	Notes	Request Status	Request Type	Details	User
Tue Feb 28 13:56:24 EST 2012	Packages: sample.war	Success	Deploy	VIEW	rhqadmin
Mon Feb 27 21:52:22 EST 2012	Packages: sample.war 2.0	Success	Deploy	VIEW	rhqadmin

Total: 2

Full Package Audit Trail

The following is a complete listing of all of the package operations that have occurred on this resource. By default, they are ordered from newest to oldest. In other words, for packages that are referenced more than one time in this list, the most recent item indicates the current status of the package.

Package Name	Version	Architecture	Status	Timestamp	Details
sample.war		noarch	Currently Installed	2/28/12, 1:56:24 PM, EST	VIEW
sample.war	1.0.0-GA (1.x) [sha256=67a253b1f5d4f122dc5815c7f9cfdbc93c14030aa30cbeb35f68ca09eeb1828b]	noarch	Discovered	2/28/12, 1:56:24 PM, EST	VIEW
sample.war	2.0	noarch	Missing	2/28/12, 1:56:24 PM, EST	VIEW
			Installation	2/28/12, 1:56:24	

Figure 28.4. Package History for a Resource

The package history shows both the time the operation was initiated and completed and the user who initiated it. This is valuable for auditing changes, correlating incidents and response, and tracking resource configuration.

PART V. MANAGING JBOSS RESOURCES

CHAPTER 29. HOW JBOSS ON MANAGES JBOSS RESOURCES

JBoss Operations Network provides extra tools that help manage JBoss server instances. These management tools cover everything from manually configuring a JBoss inventory to applying JBoss patches.

29.1. HOW JBOSS ON WORKS WITH JBOSS SOFTWARE

JBoss AS/EAP is an application server, so its core goal is delivering web content. In order to manage JBoss applications effectively, JBoss ON manages both the application server itself and the content that it delivers.

At the server level, JBoss ON treats JBoss servers and services as resources, just like other types of managed resources. Each major JBoss server type — like EAP, SOA-P, Data Services, or Business Rules Management — is implemented through an agent resource plug-in; these plug-ins are available in separate, independently-installed *plug-in packs*. These plug-ins define a variety of related services and projects that support the main server type. For example, the EAP plug-in pack includes agent plug-ins for EAP, Hibernate, Cache Service, JMS, and JMX.

JBoss ON provides support for application servers and services by managing and monitoring:

- Monitoring, event logging, availability, and alerting ("Setting up Monitoring, Alerts, and Operations")
- Managing software updates and patches (Section 32.4, “Applying JBoss Patches from the Patch RSS Feed”)

In JBoss ON, there is a very close relationship between JBoss resources and JBoss content. The web content — like EARs, WARs, and web contexts — is treated as a hybrid kind of resource. They have a defined parent-child hierarchy and have management tasks like starting and stopping instances, metrics and alerting, and configurtion properties as do other types of resources.

But content-backed resources are simultaneously treated as software packages, with update histories, content repositories, and the ability to revert to previous versions.

Managing JBoss content resources, then, focuses on the relationship of the content to the application server:

- Deployed web applications (Section 32.3, “Deploying Applications”)
- Clusters and web contexts (Section 32.5, “Managing mod_cluster Deployments for JBoss EAP 5 (Tech Preview)”)

The key for management (of resources and content) is that everything is unified into a central location, across JBoss applications, across domains, and across machines.

29.2. WHAT'S COVERED IN THIS GUIDE

In a very general sense, JBoss ON handles JBoss resources the same as any other: it provides monitoring and alerting, manages configuration and drift, and displays and controls the inventory.

On another level, because of the tight integration between JBoss products and JBoss ON, JBoss ON provides unique ways to manage JBoss servers. JBoss ON can streamline common operations in ways that are easier than trying to do it directly on JBoss resources. It is also centralized and offers a focused view on JBoss resources.

So, the purpose of this guide is not to delve into how to use JBoss ON generally. This guide is focused on how to use JBoss ON to perform specific tasks with JBoss resources and, therefore, how to manage JBoss resources more effectively. This includes:

- Content management for EARs and WARs and other web applications
- Managing JBoss server clusters
- Managing JBoss server domains
- Monitoring JBoss instances

For concepts and general procedures for using JBoss ON, see the full documentation set at http://docs.redhat.com/docs/en-US/JBoss_Operations_Network/index.html.

29.3. INSTALLING JBOSS PLUG-IN PACKS

All resources are identified to the server and the agent through *agent plug-ins*. Agent plug-ins defined support resources, including supported metrics, operations, configuration propertities, and resource versions.

JBoss products are supported through their own special plug-ins which are available in separate *plug-in packs*. These plug-ins packs have to be installed before the JBoss products can be discovered and managed.

Table 29.1. Supported JBoss Products

Product	Versions	Plug-in Pack
JBoss AS or EAP	• 4.x • 5.x	EAP
JBoss AS	• 7.1	Tech Preview
mod_cluster	• 1.1.2	Tech Preview
Hibernate		EAP
JBoss Cache Service	• 1 • 3	EAP
JMX		EAP
JMS Manager Service (HornetQ)		EAP

Product	Versions	Plug-in Pack
Business Rules Management Service (BRMS/Drools)	• 5.x	BRMS
Data Services Platform (EDS-P/Teiid)	• 5.x	EDS
Developer Studio (ModeShape)	• 4.x	ModeShape
Tomcat (Web Services)	• 1.x	EWS
ESB Services (SOA-P)	• 4.x • 5.x	SOA-P

To install JBoss plug-ins:

1. Download the plug-in JAR files from the Customer Support Portal.

   In the Customer Support Portal, click **Software**, and then select the **JBoss ON for** *Plug-in* drop-down box.

2. Download the plug-in packs.

3. Unzip the additional plug-in packs. This creates a subdirectory with the name **jon-plugin-pack-*plugin_name*-3.2.GA1**.

4. Copy the new plug-ins from the **jon-plugin-pack-*plugin_name*-3.2.GA1/** directory to the JBoss ON server plug-in directory.

   ```
   [root@server rhq-agent]# cp myDirectory/jon-plugin-pack-plugin_name-
   3.2.GA1/* /opt/jon/jon-server-3.2.GA1/plugins
   ```

   The server polls this directory every few minutes to look for updated plug-ins.

5. Once the plug-ins have been deployed to the server, reload the agent plug-ins on the agents themselves.

   This can be done from the command line using the agent's **plugins** command:

   ```
   > plugins update
   ```

   This can also be done in the JBoss ON GUI by scheduling an *update plugins* operation for an agent or a group or agents.

CHAPTER 30. GENERAL TASKS

30.1. SETTING UP A CUSTOM JVM FOR DISCOVERY

The Generic JMX Plug-in detects Java processes. This is a very simple plug-in; aside from the ability to exclude some types of JVMs, the plug-in detects *any* properly-configured Java process and imports it is as a JMX server. For a generic JMX server, all of its subsystems — logging, threads, memory, and others — are also imported as children of the JMX server.

All of this background information is covered in the JMX resource documentation in the *Resource Reference: Monitoring, Operation, and Configuration Options*. Writing a custom plugin is covered in *Writing Custom Plug-ins*.

A JVM resource itself has to be configured in a certain way for JBoss ON to be able to discover the resource to add it to the inventory.

30.1.1. Required JVM Configuration for Discovery

The agent discovers and subsequently manages a resource by identifying the resource based on certain parameters and then connecting to the agent over those discovered settings. For a Java process to be discovered using the Generic JMX Plug-in, it has to be configured to be connected to in one of two ways:

- Sun JMX remoting is enabled, with a port system property specified in the command line.

  ```
  -Dcom.sun.management.jmxremote.port=12345 com.xyz.MyAppMain
  ```

- A Sun/Oracle-compatible Java process is accessible through the **com.sun.tools.attach** API, and the resource key is specified as a system property in the command line.

  ```
  -Dorg.rhq.resourceKey=KEY com.xyz.MyAppMain
  ```

30.1.2. Excluding Java Processes from Discovery

The Generic JMX Plug-in is designed to be extended so that custom plug-ins can be based off it to detect and manage specific types of Java processes. For example, the JBoss ON agent, as a Java process, would normally be detected by the Generic JMX Plug-in, but it is discovered by the Agent Plug-in, which trumps the Generic JMX Plug-in. Similarly, JBoss EAP and Tomcat servers are also discovered by server-specific plug-ins, not the generic plug-in.

Resources which can be discovered by another plug-in should be excluded from the Generic JMX Server discovery, or there could be (spurious) conflict errors recorded in the agent log.

The Java processes to exclude from the discovery scan are defined in the JBoss ON agent configuration. There is a Java option for the agent, **rhq.jmxplugin.process-filters**, which lists strings to ignore specifically from the Generic JMX Plug-in discovery scan. If a Java process contains any of the strings in the filter, it is excluded from discovery as a JMX server[6].

The default agent configuration filters out the classes for the agent, the JBoss ON server, and Tomcat servers:

```
RHQ_AGENT_ADDITIONAL_JAVA_OPTS="-Drhq.jmxplugin.process-
filters=org.rhq.enterprise.agent.AgentMain,org.jboss.Main,catalina.startup
.Bootstrap"
```

To add excludes filters:

1. Open the agent configuration file.

   ```
   [jbosson-agent@server ~]$ vim agentRoot/rhq-agent/conf/agent-
   configuration.xml
   ```

2. In the **RHQ_AGENT_ADDITIONAL_JAVA_OPTS** link, add the string to exclude to the **rhq.jmxplugin.process-filters** option. This can be the classname or any other identifying string which is in the command line for the given process.

   For example:

   ```
   RHQ_AGENT_ADDITIONAL_JAVA_OPTS="-Drhq.jmxplugin.process-
   filters=org.rhq.enterprise.agent.AgentMain,org.jboss.Main,catalina.s
   tartup.Bootstrap,com.abc.OtherAppMain"
   ```

   The **rhq.jmxplugin.process-filters** value is a comma-separated list of strings.

3. Restart the agent with the **--config** option to load the new configuration.

   ```
   [jbosson-agent@server ~]$ agentRoot/rhq-agent/bin/rhq-agent.sh --
   config
   ```

30.1.3. Manually Importing a JVM Resource

As Section 30.1.1, “Required JVM Configuration for Discovery” describes, there are only two connection configurations that can be used for a Java process for it to be discovered automatically by the agent. Specifically, those two connection settings both use a Sun management API, for remoting or for attach.

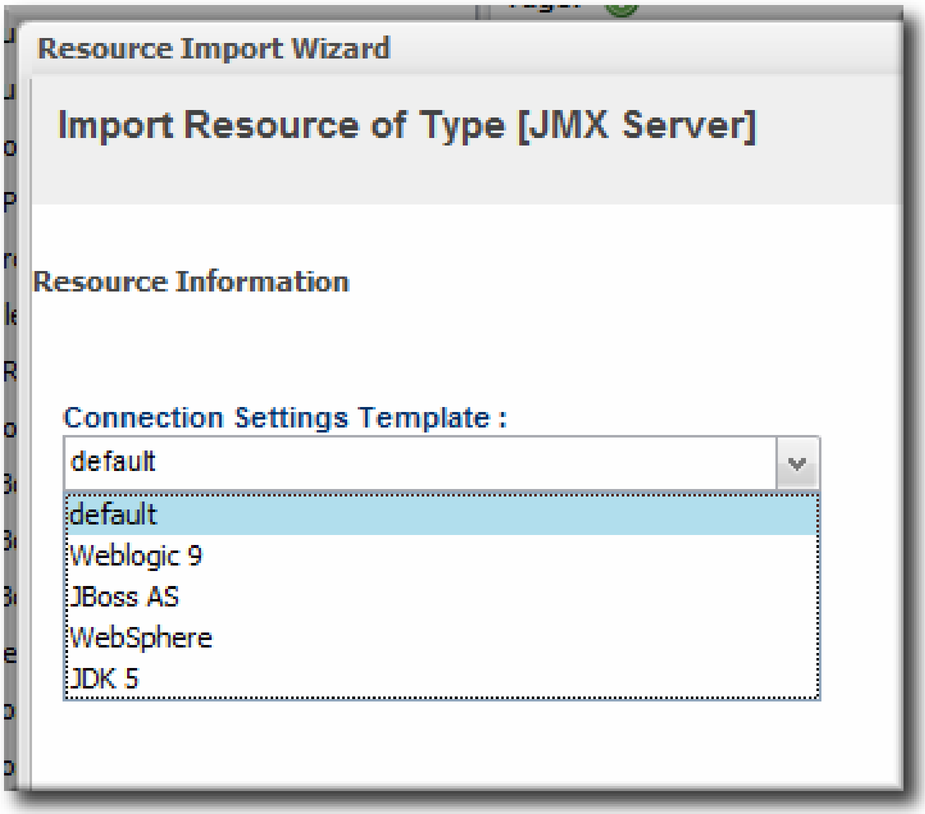
Any Java process can be imported into the inventory manually, even if it does not use the expected configuration for autodiscovery, as long as it enables JMX remoting in a supported form, meaning Sun or IBM remoting.

NOTE

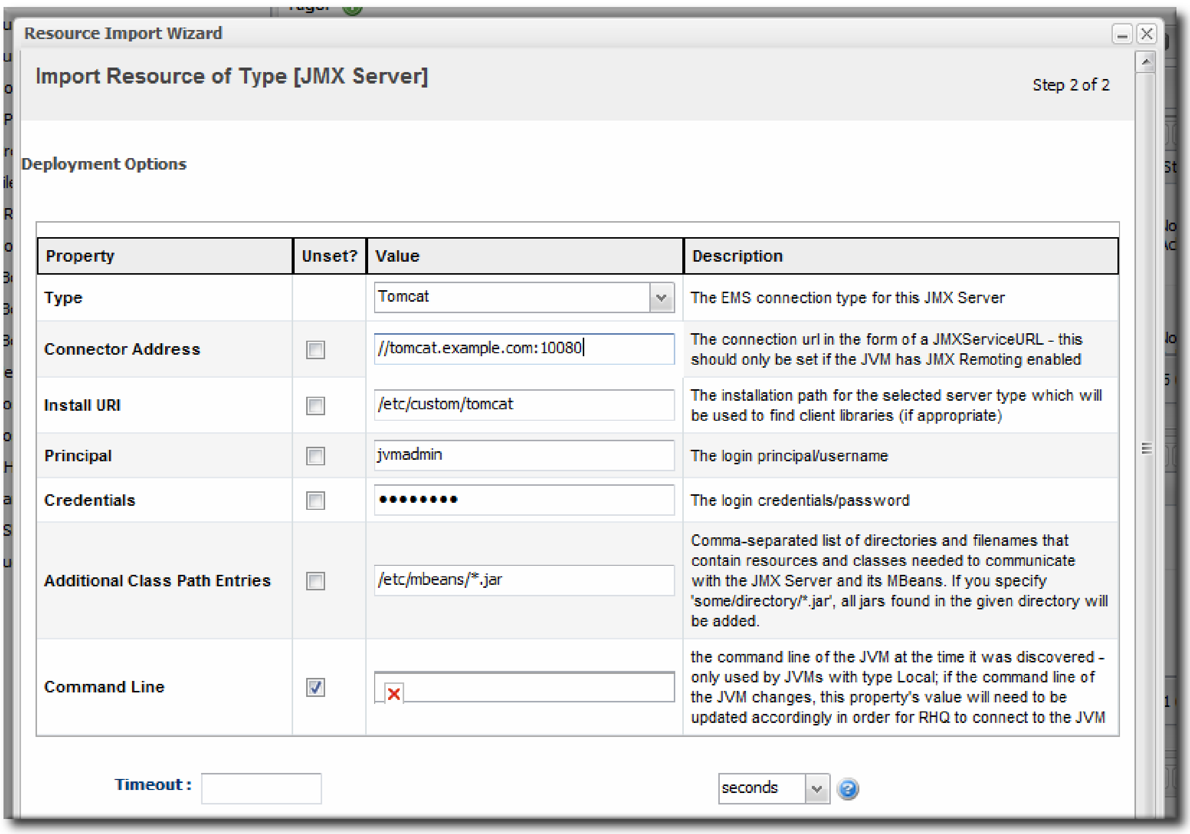
The JVM instance has to be running for the resource to be discovered and imported.

1. Click the **Inventory** tab in the top menu.
2. Select the platform resource.
3. Click the **Inventory** tab of the platform.
4. Click the **Import** button in the bottom of the **Inventory** tab, and select the JMX server resource type.

5. Select the type of JVM, and set all of the connection properties correctly, depending on the type of JVM selected.



6. Fill in the connection information for the JVM. This varies depending on the JVM type, but it includes options like a URL and port, directory paths for client libraries, directory paths for classes, and login credentials.



Property	Unset?	Value	Description
Type		Tomcat	The EMS connection type for this JMX Server
Connector Address	☐	//tomcat.example.com:10080	The connection url in the form of a JMXServiceURL - this should only be set if the JVM has JMX Remoting enabled
Install URI	☐	/etc/custom/tomcat	The installation path for the selected server type which will be used to find client libraries (if appropriate)
Principal	☐	jvmadmin	The login principal/username
Credentials	☐	••••••••	The login credentials/password
Additional Class Path Entries	☐	/etc/mbeans/*.jar	Comma-separated list of directories and filenames that contain resources and classes needed to communicate with the JMX Server and its MBeans. If you specify 'some/directory/*.jar', all jars found in the given directory will be added.
Command Line	☑		the command line of the JVM at the time it was discovered - only used by JVMs with type Local; if the command line of the JVM changes, this property's value will need to be updated accordingly in order for RHQ to connect to the JVM

Timeout : seconds

7. Click the **`Finish`** to import the instance.

30.2. ENABLING THE AGENT TO CONNECT TO SECURED JMX SERVERS

By default, JBoss EAP has its JMX server running in secure mode. However, while the agent can *discover* a JMX server in secure mode, it cannot *connect* to that secured JMX server because it cannot detect the proper credentials.

For example, the JMX server has these system properties:

```
-Dcom.sun.management.jmxremote
-Dcom.sun.management.jmxremote.port=5222
-Dcom.sun.management.jmxremote.ssl=false
-Dcom.sun.management.jmxremote.authenticate=true
-Dcom.sun.management.jmxremote.password.file=/jmxremote.password
-Dcom.sun.management.jmxremote.access.file=/jmxremote.access
```

The agent's JMX plug-in examines the command line for the JMX server's process. It detects the port to use to connect to the JMX server, but it cannot read the password and access files to get the JMX server's credentials.

NOTE

Because the agent cannot connect to the JMX server, it assigns the resource a DOWN availability state, even if the server is running fine.

There are several ways to enable the agent to connect to the JMX server.

Edit the jmx-console-users.properties File

The agent generally reads the connection credentials from the **`jmx-console-*.properties`** file in the *JbossASInstallDir*/**`server/default/conf/props/`** directory.

When the JMX server is running secured, there are no entries in the **`jmx-console-users.properties`** file, so there is no way for the agent to get the credentials.

1. Open the **`jmx-console-*.properties`** file for editing. For example:

   ```
   [root@server ~]# vim
   JbossASInstallDir/server/default/conf/props/jmx-console-
   users.properties
   ```

2. Uncomment or add a line for the admin user.

   ```
   admin=admin
   ```

If that does not work, then edit the connection settings for the resource.

Edit the Connecting Settings to Use the Remote Access Files

By default, the agent uses the **`jmx-console-*.properties`** file for a username, not the access files. It is possible to change the connection settings for the resource so that the agent uses the access files, going through the remote endpoint, which were specified in the JMX server's command line.

1. Click the **Inventory** tab in the top menu.

2. Search for the JMX server in the **Servers** area of the **Inventory**, or open the JBoss EAP instance and navigate through its children to find the JMX server instance.

3. On the JMX server's entry page, open the **Inventory** tab, and select the **Connection Settings** subtab.

4. Enter the user name and password to set in the JMX remote access files.

5. Click the **Save** button.

Edit the Connection Settings to Connect Through the Parent Resource

JBoss ON can connect to the parent resource, and then use that to connect to the JMX server, rather than connecting through the remoting endpoint. This does not require using any user credentials, since the parent can connect to the child resource using internal authentication.

1. Click the **Inventory** tab in the top menu.

2. Search for the JMX server in the **Servers** area of the **Inventory**, or open the JBoss EAP instance and navigate through its children to find the JMX server instance.

3. On the JMX server's entry page, open the **Inventory** tab, and select the **Connection Settings** subtab.

4. Unset all of the connection properties *except* for the **Type** property.

5. For the **Type** property, select the **Parent** value.

6. Click the **Save** button.

[6] The excluded process can still be discovered by another agent plug-in.

CHAPTER 31. MANAGING JBOSS EAP 6 (AS 7)

JBoss Enterprise Application Platform 6 is modular and flexible, with an unlimited number of configuration and deployment options depending on your environment.

These chapters do not cover every single management options or task possible through JBoss ON. Rather, they cover tasks or configuration that are unique to EAP 6 or that require special consideration when used to manage EAP 6.

31.1. THE STRUCTURE OF JBOSS EAP 6

The key features of JBoss EAP 6 are a modular class-loading structure and a domain framework which centralizes configuration in a single controller. These two features have one big asset in common: flexibility. The idea is to start with a small, lightweight application server and then add in the modules for the desired functionality and the server instances for a naturally distributed system.

JBoss ON works within the changeability of that system to bring clarity into how those modular, distributed resources are related and to provide an extra skin of both administrative control and information visibility.

JBoss ON imposes a hierarchy and structure on JBoss EAP 6 installation, but before you can have an idea of how that structure is applied, it helps to have an understanding of the order that EAP 6 itself has.

NOTE

JBoss EAP 6 topologies for standalone and domain servers are described in the JBoss AS 7 project documentation.

31.1.1. "Classic" Structure: Standalone Servers

Most of the functionality for the JBoss EAP 6 server is implemented through independent classes or *subsystems*. Which subsystems a server uses, and their configuration properties or configured instances, are defined in a *profile*.

A standalone instance for EAP 6, much as in EAP 5, is a collection of those defined subsystems, a set of socket bindings to define communications, network interfaces, and other settings.

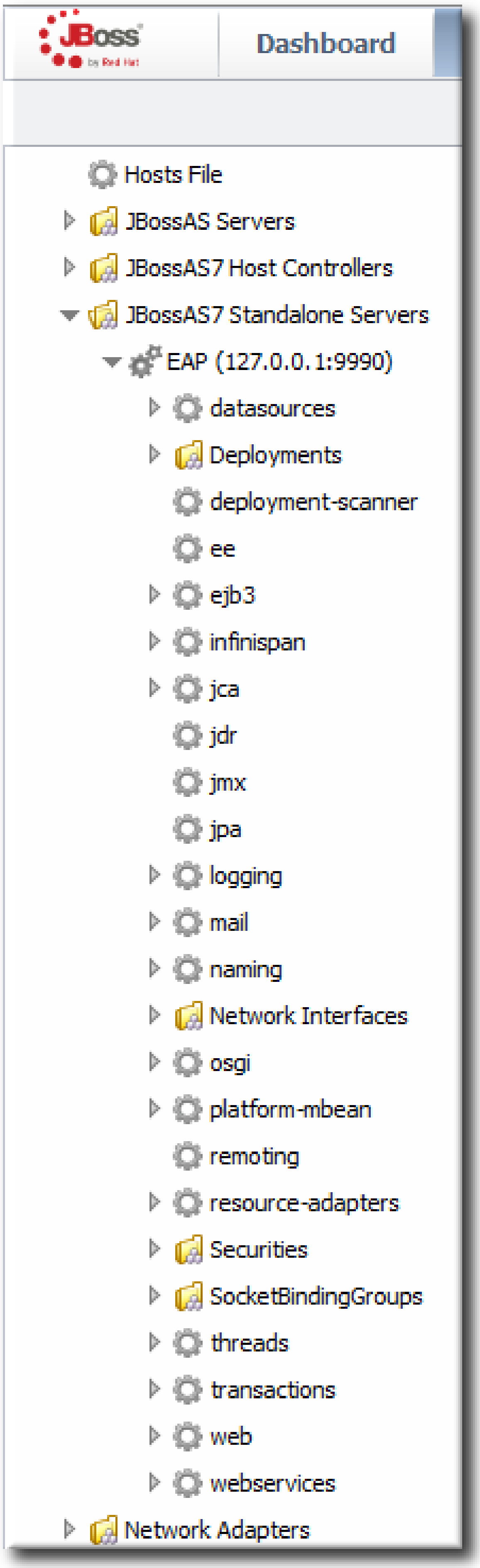
The entire server instance is defined in its **standalone.xml** file. Almost every entry in that file is identified as a child of the server. For example, these subsystem entries create the **ee** and **jmx** child resources for the server:

```
<subsystem xmlns="urn:jboss:domain:ee:1.1"/>
<subsystem xmlns="urn:jboss:domain:jmx:1.1">
    <show-model value="true"/>
    <remoting-connector/>
</subsystem>
```

Likewise, socket bindings, network interfaces, and deployed content, which are defined in the configuration file, are discovered as resources.

Subsystems like datasources and logging configuration can have multiple instances defined; these are all configured as child resources of the subsystem. Because of the modular nature of the EAP 6 server, the exact subsystems — and therefore the exact children in JBoss ON — vary with the subsystem and other definitions in the configuration file.

Overall, because of the number of subsystems and other configuration settings, the standalone server has a very wide, flat, and shallow inventory tree. JBoss ON imposes little additional hierarchy on the standalone server structure. It reflects the way the server is configured in `standalone.xml`.



Hosts File
JBossAS Servers
JBossAS7 Host Controllers
JBossAS7 Standalone Servers
EAP (127.0.0.1:9990)
datasources
Deployments
deployment-scanner
ee
ejb3
infinispan
jca
jdr
jmx
jpa
logging
mail
naming
Network Interfaces
osgi
platform-mbean
remoting
resource-adapters
Securities
SocketBindingGroups
threads
transactions
web
webservices
Network Adapters

Figure 31.1. Standalone Server Configuration

Management and operations for the standalone server are self-contained. **Every child node for the standalone server combines both its configuration and its runtime environment (monitoring, alerting, and operations) in the same location. To change the configuration for a datasource, for example, edit the datasource entry directly.** Every standalone server is managed independently of any other server, even in the same cluster or the same machine.

All of this is comparable to the structure and functionality of a JBoss EAP 5 server, with some slight differences in the organization of the inventory.

31.1.2. Separating Configuration and Real-Time Operations: Domains

The structure of the domain is entirely different, in a way that embraces a more modular design.

A domain divides the application server into two conceptual halves: server configuration and runtime operations.

Configuration is centralized into the domain controller. The domain controller defines the profile and subsystem configurations, JVM settings, interfaces, and other settings in a single place.

The functions of the application server are carried out by *managed* servers, which can be installed on multiple machines. These managed servers do not define their own configuration (with a handful of exceptions); they accept the profile structures from the domain and serve the web applications deployed through the domain.

There is a secondary level of organization domains and managed servers called *server groups*. Server groups are defined as part of the domain configuration. They create environments for managed servers; they are configuration nodes, in a sense. A domain can have multiple profiles, JVM definitions, and web applications; it is a central repository of all possible configuration. Only specific configuration and content is assigned to a server group, and that specific configuration is what is applied to the managed servers.

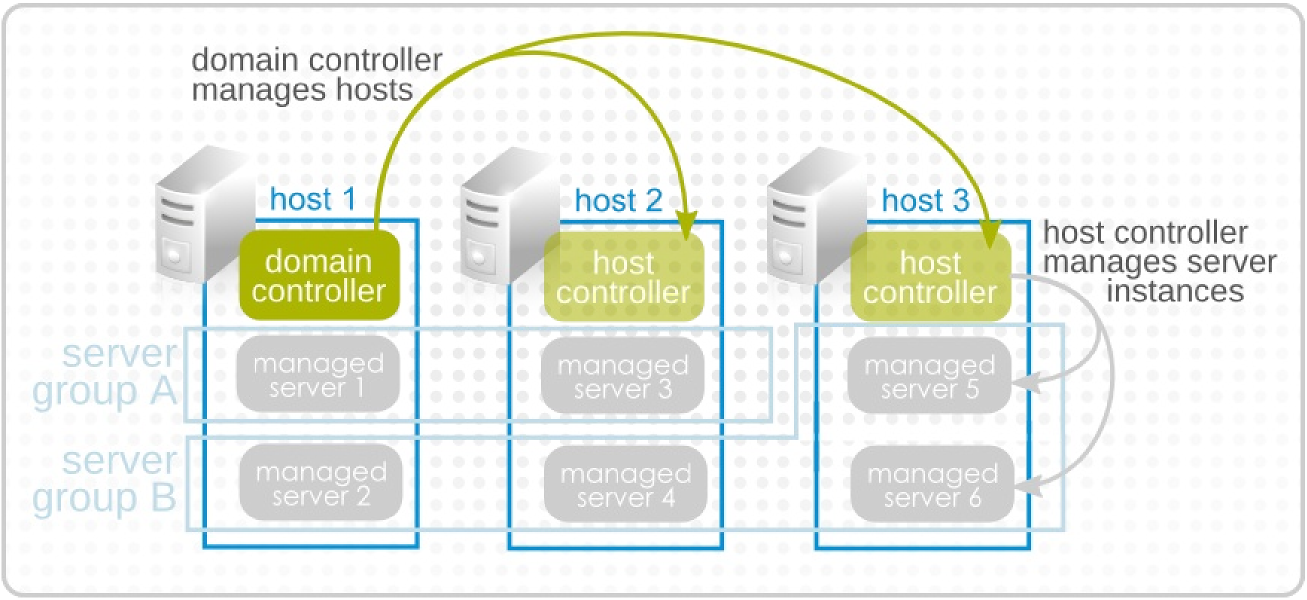


Figure 31.2. Simple Domain Structure

The domain configuration is defined in the `domain.xml` file. This lists all of the configured profiles and subsystems, server groups, socket binding configuration, system properties, deployments, and other settings. As with the standalone server, almost every entry is discovered and added to the inventory as a resource. For example, this creates a server group resource, with a child deployment resource and a child JVM resource for the server group.

-

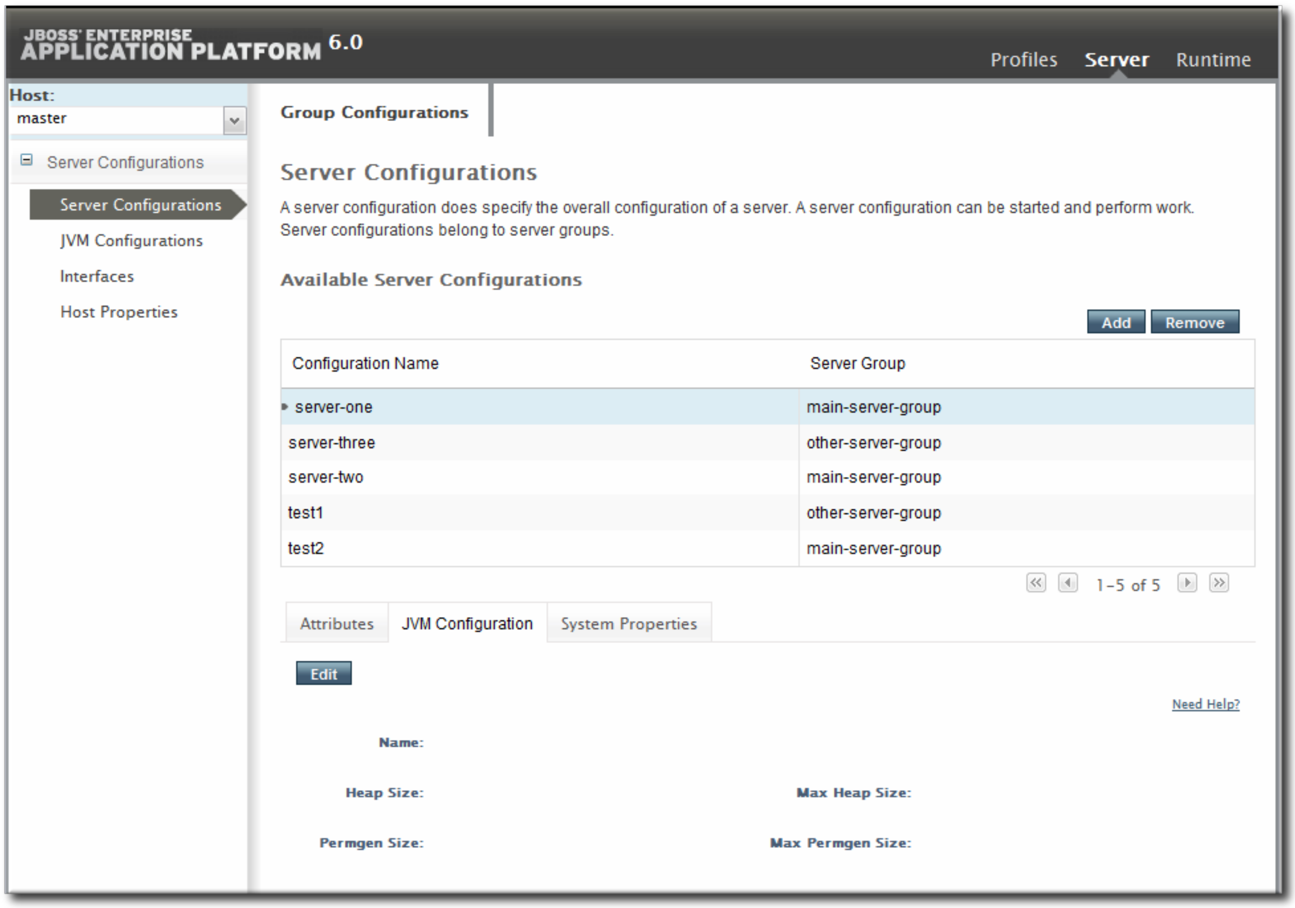
Example 31.1. Server Group domain.xml Entry

```
          <server-group name="main-server-group" profile="full">
              <jvm name="default">
                  <heap size="1303m" max-size="1303m"/>
                  <permgen max-size="256m"/>
              </jvm>
              <socket-binding-group ref="full-sockets"/>
              <deployments>
                  <deployment name="sample2.war" runtime-
name="sample2.war"/>
              </deployments>
 </server-group>
```

The instances that carry out the operations are identified in the **`host.xml`** file. These managed servers have virtually no configuration of their own; they simply point back to the original server group configuration to use in the domain.

```
    <servers>
        <server name="server-one" group="main-server-group"/>
        <server name="server-two" group="other-server-group">
            <!-- server-two avoids port conflicts by incrementing the
ports in
                 the default socket-group declared in the server-group --
>
            <socket-bindings port-offset="150"/>
        </server>
    </servers>
```

The domain, through its profiles and its server groups, defines the overall configuration. The managed servers are workers under that configuration. This division between configuration and operation, particularly the emphasis on the functionality of the domain as a whole rather than any individual member, is reflected in the EAP 6 management console. The **`Profile`** and **`Servers`** areas manage domain configuration, while **`Runtime`** displays current statistics and deployed web applications. The focus is always on domain configuration and domain information.



Server Configurations

A server configuration does specify the overall configuration of a server. A server configuration can be started and perform work. Server configurations belong to server groups.

Available Server Configurations

Configuration Name	Server Group
server-one	main-server-group
server-three	other-server-group
server-two	main-server-group
test1	other-server-group
test2	main-server-group

Name:

Heap Size: Max Heap Size:

Permgen Size: Max Permgen Size:

Figure 31.3. EAP 6 Console

Managing domain resources in JBoss ON is less about managing the configuration than it is managing information. One of the strengths of the JBoss ON inventory structure is that is exposes the components of the domain very clearly and helps delineate the relationships between those resources.

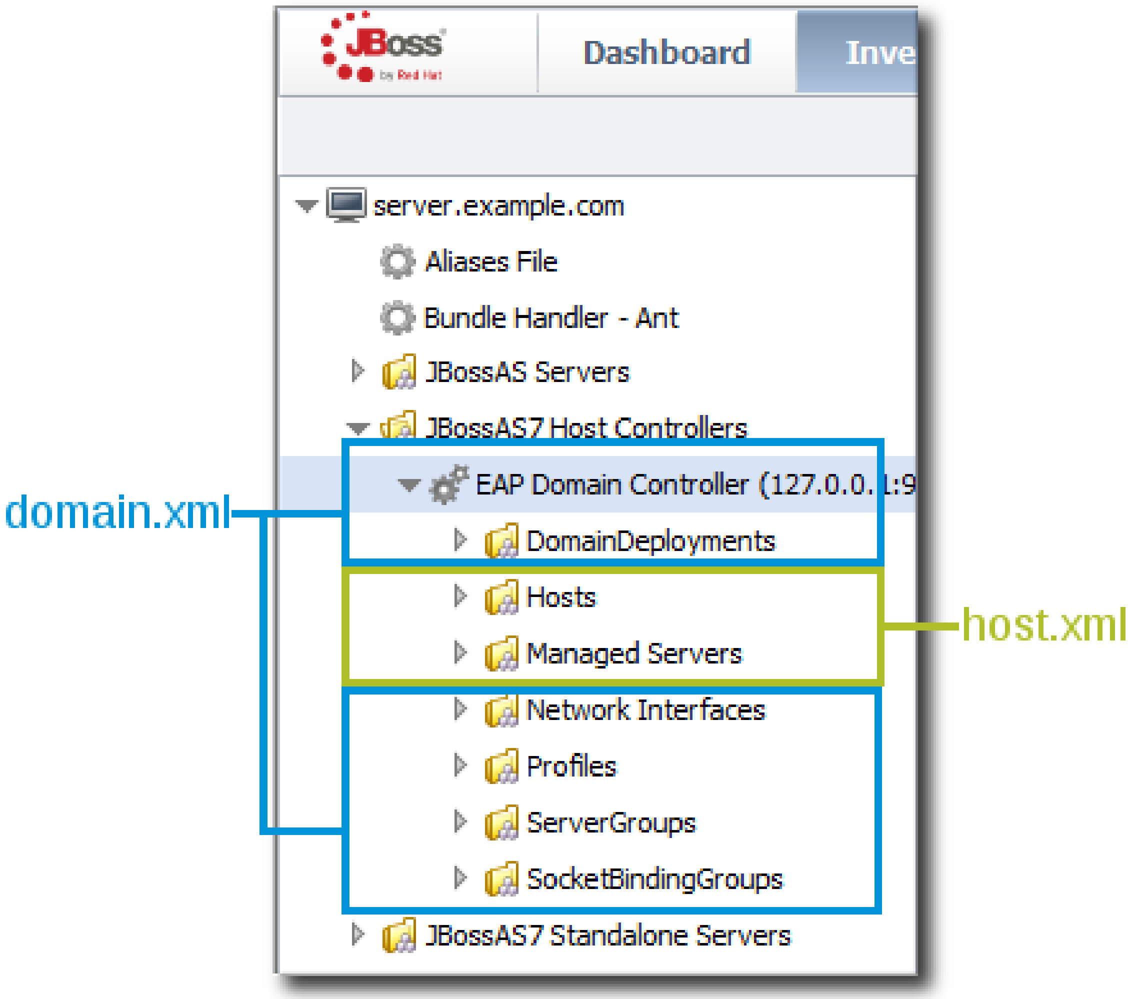


Figure 31.4. Domain Resources

For example, the entry for a managed server lists all of the subsystems defined in its server group's profile, it lists its host controller definition, and it lists managed web applications deployed on it (through the server group). This consolidates all of the configuration information that is applied to that one worker instance, which makes the monitoring information on that managed server more valuable.

There is a shade of artificiality in the JBoss ON inventory in order to maintain the domain relationships. The domain is a single, overarching entity, but the components of the domain can span multiple platforms and resources on those platforms. The domain controller is considered the parent resource for all domain resources, regardless of their platform. This is hinted at in Figure 31.5, “Domain Components in the JBoss ON Inventory”. The domain controller is on Platform 1, so even though a host controller and two managed servers are on Platform 2, the host controller and managed resources are shown as children of the domain resource (on Platform 1).

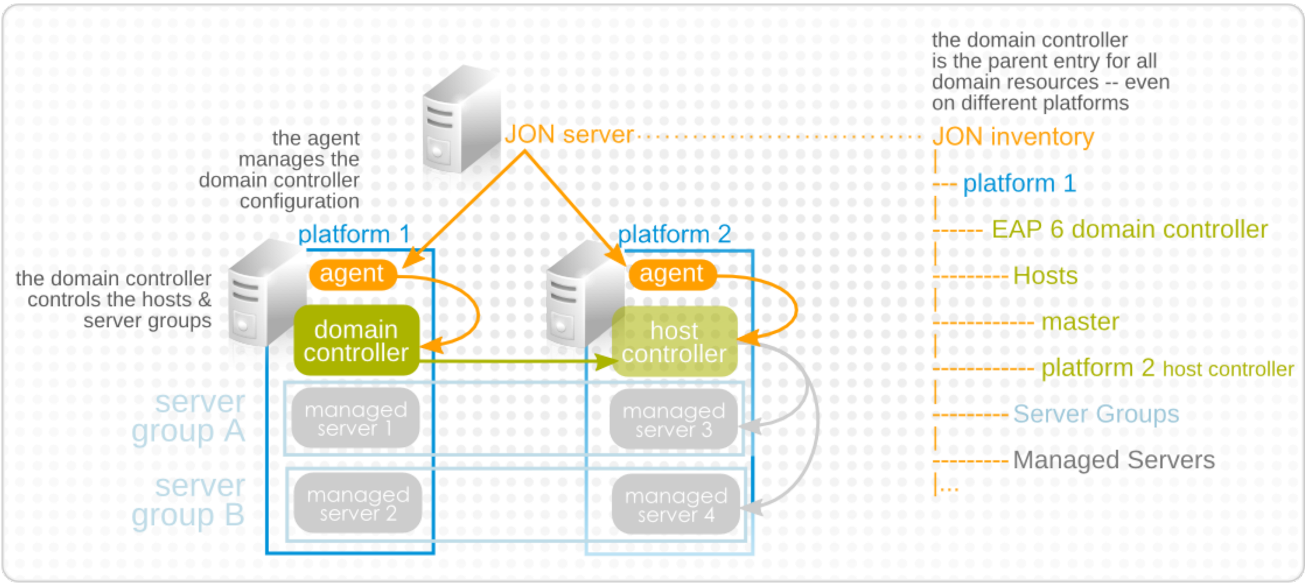

the domain controller is the parent entry for all domain resources -- even on different platforms
JON server
JON inventory
the agent manages the domain controller configuration
--- platform 1
------ EAP 6 domain controller
--------- Hosts
------------ master
------------ platform 2 host controller
--------- Server Groups
--------- Managed Servers
|...
platform 1
platform 2
the domain controller controls the hosts & server groups
agent
agent
domain controller
host controller
server group A
server group B
managed server 1
managed server 2
managed server 3
managed server 4

Figure 31.5. Domain Components in the JBoss ON Inventory

31.1.3. EAP 6 Resources in JBoss ON

Every EAP 6 configuration area in `domain.xml`, `host.xml`, or `standalone.xml` is interpreted as a resource type. Because EAP 6 draws a distinction between configuration components and operational components, the EAP 6 resource types in JBoss ON do different things; some resources define configuration, while others are used for monitoring, alerting, and operations.

In domains, profiles, socket bindings, and network interfaces for the domain are all configuration entries. Therefore, in JBoss ON, the corresponding resource types define configuration settings.

In domains, the managed servers are the functional instances. These resources are active, so, in JBoss ON, the managed server resources can run operations, monitor metrics, and trigger alerts.

There is no overlap between the configuration resources and the operational resources in functionality. For example, a datasource resource under a profile has configuration, but no monitoring or alerting. A datasource under a managed server collects the monitoring metrics, defines alerts, and runs operations. These resources are related — the managed server datasource resource references the configuration in the profile datasource — but they are different.

In standalone servers, configuration and management are enabled in the same resource. A standalone server is basically a combination of a profile resource (configuration) and a managed server resource (management). A datasource resource for a standalone server, then, has both configuration and monitoring, alerts, and operations on the same resource.

31.1.4. The Purpose of Managing EAP 6 Resources with JBoss ON

The JBoss ON UI maintains parity with the EAP 6 console, so it can be used to edit most configuration settings, create child entries, and deploy content, same as the EAP 6 management console.

The purpose of JBoss ON management, though, is not to have yet another web UI to use. JBoss ON maintains a larger perspective on the resources within the domain. This infrastructure-wide perspective reveals the relationship of a resource within the domain, the larger IT structure, and even itself and its own history aside from simply making a current configuration change.

Standalone servers and domains in JBoss ON are organized in a way that highlights the information that is available to administrators:

- Additional monitoring metrics for resources, with stored historical data, resource-specific operational baselines, and alternate display types
- Configuration and connection setting histories
- Inventory change histories when children are added or removed through JBoss ON
- Package versioning and repositories for controlled package updates
- Configuration, monitoring, and alerting for underlying and associated resources, like the platform, Apache and Tomcat web servers, and plug-ins like `mod_cluster`

31.2. UPGRADING THE JBOSS EAP 6 RESOURCE PLUG-IN

JBoss Operations Network 3.0 and 3.0.1 had a tech preview version of the JBoss Enterprise Application Platform (EAP) 6 agent plug-in. With normal upgrade processes, agent plug-ins are updated when the JBoss ON server is updated. **However, tech preview plug-ins are not upgraded. The tech preview version of the plug-in must be deleted, and then the new plug-in installed.**

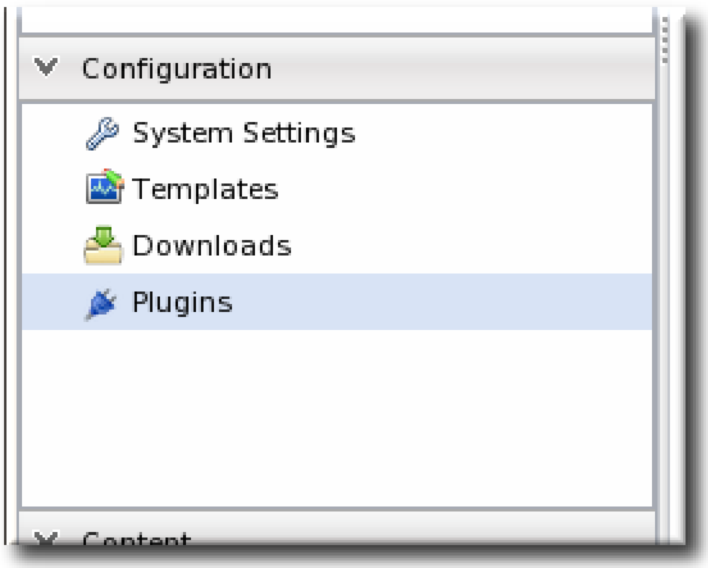
The EAP 6 tech preview plug-in and the EAP 6 GA plug-in are treated as two separate plug-ins by JBoss ON. If the tech preview version is not deleted, then any EAP 6 resource will be discovered and listed twice in the inventory, once discovered by the tech preview plug-in and discovered again by the GA plug-in.

NOTE

Any configuration, monitoring data and baselines, and resource histories will be lost after upgrading to the new EAP 6 plug-in.

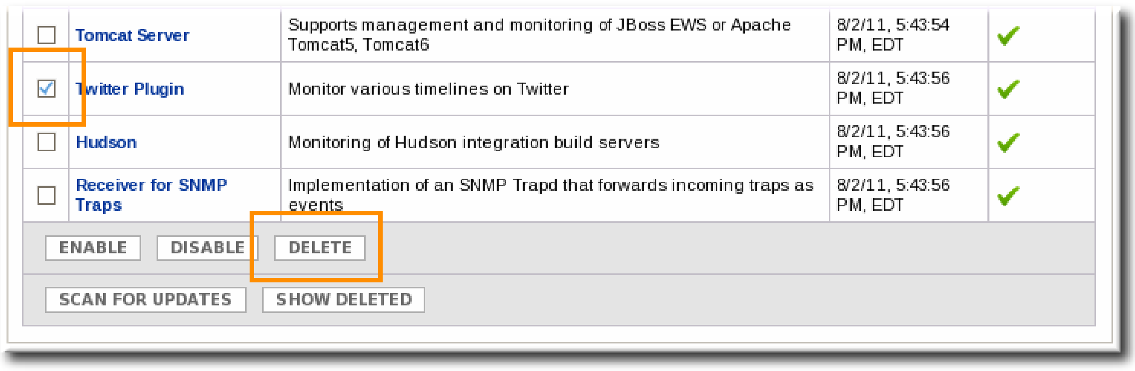
To load the new EAP 6 plug-in:

1. Delete the original tech preview plug-in and purge it from the JBoss ON database. Purging the plug-in allows the server to deploy the new plug-in in its place.
   1. In the top menu, click the **`Administration`** tab.
   2. In the **`Configuration`** box on the left navigation bar, click the **`Agent Plugins`** link.



3. Select the EAP 6 tech preview plug-in.

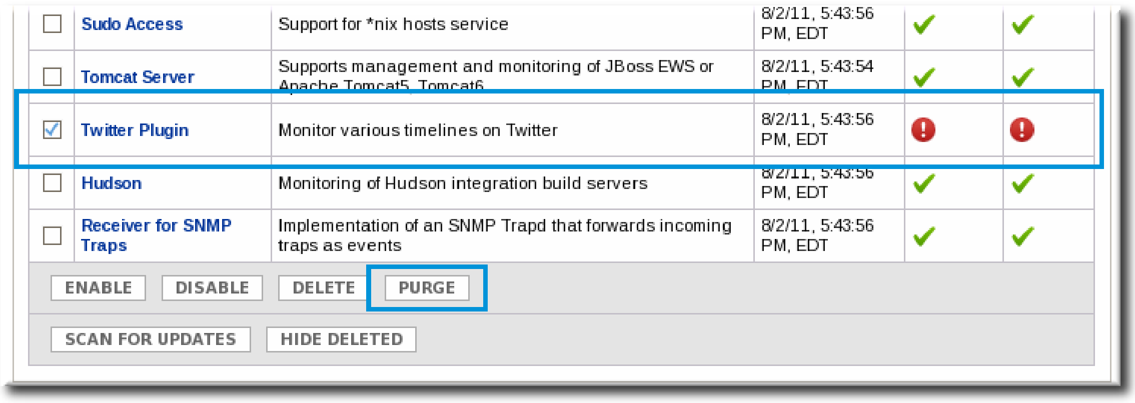
4. Click the **`Delete`** button.



☐	Tomcat Server	Supports management and monitoring of JBoss EWS or Apache Tomcat5, Tomcat6	8/2/11, 5:43:54 PM, EDT	✓
☑	Twitter Plugin	Monitor various timelines on Twitter	8/2/11, 5:43:56 PM, EDT	✓
☐	Hudson	Monitoring of Hudson integration build servers	8/2/11, 5:43:56 PM, EDT	✓
☐	Receiver for SNMP Traps	Implementation of an SNMP Trapd that forwards incoming traps as events	8/2/11, 5:43:56 PM, EDT	✓

5. Click the **`SHOW DELETED`** button at the bottom of the plug-ins list.

6. Select the EAP 6 plug-in, and then click the **PURGE** button. This removes the entry in the JBoss ON database that tells the servers to ignore that original tech preview plug-in and any updates to it.



☐	Sudo Access	Support for *nix hosts service	8/2/11, 5:43:56 PM, EDT	✓	✓
☐	Tomcat Server	Supports management and monitoring of JBoss EWS or Apache Tomcat5, Tomcat6	8/2/11, 5:43:54 PM, EDT	✓	✓
☑	Twitter Plugin	Monitor various timelines on Twitter	8/2/11, 5:43:56 PM, EDT	❗	❗
☐	Hudson	Monitoring of Hudson integration build servers	8/2/11, 5:43:56 PM, EDT	✓	✓
☐	Receiver for SNMP Traps	Implementation of an SNMP Trapd that forwards incoming traps as events	8/2/11, 5:43:56 PM, EDT	✓	✓

IMPORTANT

Wait for the purge operation to complete before continuing with the upgrade process.

2. If the server has not already been upgraded, upgrade the JBoss ON server, as described in the server upgrade section of the Installation Guide.

3. Install the EAP 6 plug-in pack, as described in the installing plug-in packs section of the Installation Guide.

4. Import the EAP 6 server and its children, and manage the resources as normal.

31.3. SETTING UP JBOSS EAP 6 INSTANCES

31.3.1. Configuring the Agent to Discover EAP 6 Instances

As covered in Chapter 4, *Interactions with System Users for Agents and Resources*, the system user as which the agent runs has a direct effect on how the agent can manage certain resource types. For EAP instances, the agent's system user must have the appropriate permissions to be able to manage EAP resources:

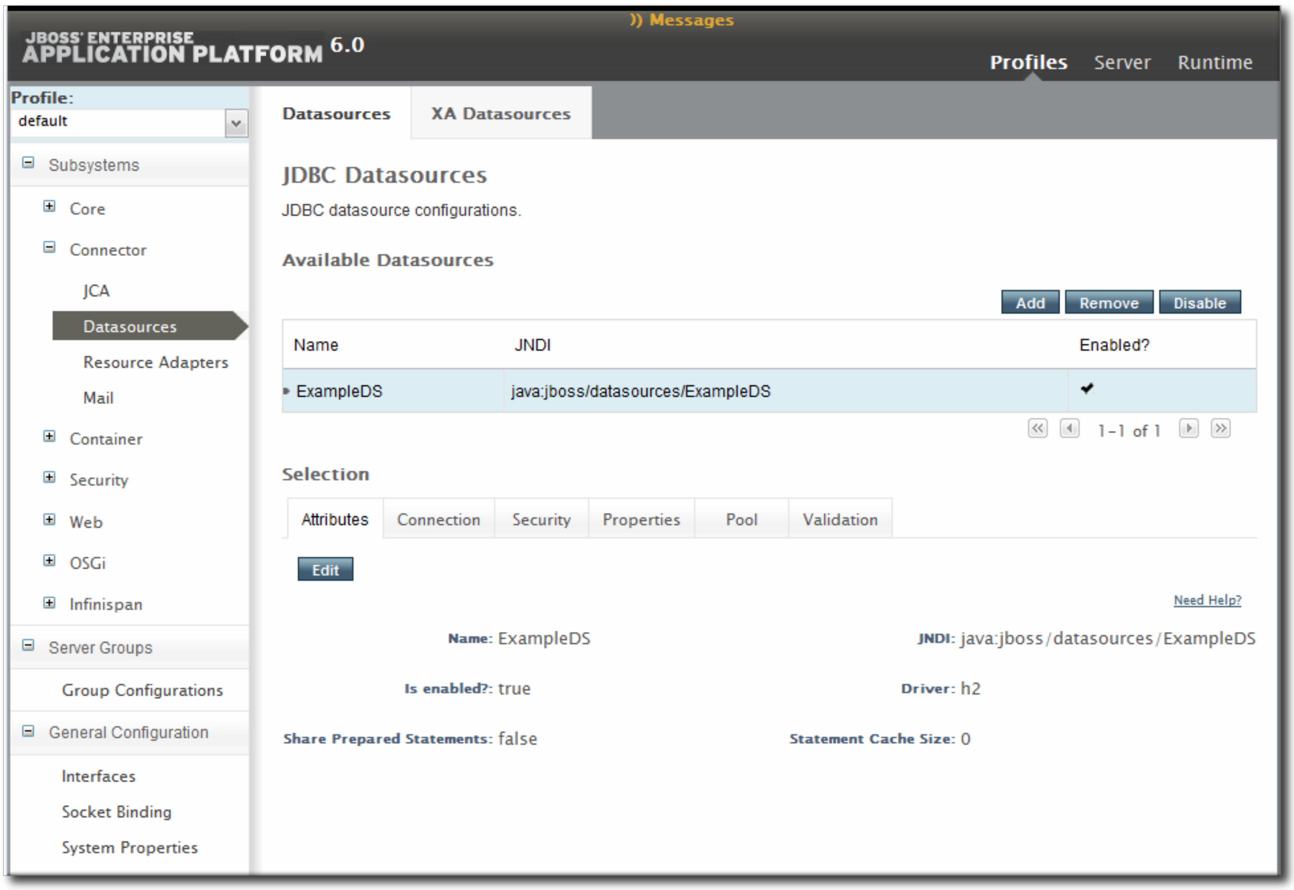
- The agent must have read permissions to the **run.jar** file, plus execute and search permissions for every directory in the path to the **run.jar** file.
- When a JBoss EAP 6 instance is installed from an RPM, the agent user must belong to the same system group which runs the EAP instance. This is **jboss**, by default.

31.3.2. Configuration for Servers and Profiles

31.3.2.1. Differences for Standalone Servers and Domains

Section 31.1, "The Structure of JBoss EAP 6" goes over some of the differences between standalone servers and domain structures. The crucial difference for configuration is that **with a standalone server, all of the configuration is performed directly on the child entries.** With a domain, configuration is divided, with almost all configuration centralized in the domain-managed profiles and server groups.

This is reflected in the EAP 6 management console. Almost all configuration is under the **Profiles** area. The domain profiles define individual subsystem configuration, system properties, global JVM settings, and server group configuration.



JDBC Datasources

JDBC datasource configurations.

Available Datasources

Name	JNDI	Enabled?
ExampleDS	java:jboss/datasources/ExampleDS	✓

1-1 of 1

Selection

Name: ExampleDS
JNDI: java:jboss/datasources/ExampleDS
Is enabled?: true
Driver: h2
Share Prepared Statements: false
Statement Cache Size: 0

Figure 31.6. Profiles Area in the EAP 6 Console

The **Servers** area covers the limited amount of configuration that is set on a managed server, mainly a local JVM definition and operations like stopping and starting the server.

In JBoss ON, the major configuration areas for the domain — profiles and their subsystems, socket bindings, server groups — are broken out as separate resource types. This configuration is applied to the managed server (through the server group), much like a template.

Always edit the resource which originates the configuration settings. For most settings, that means editing the related child resource of the domain controller.

- Subsystem configuration is located in the profile resources within the `Profiles` autogroup for the domain controller.
- JVM definitions are configured under the domain controller (domain-wide defaults), server group (group-wide settings), or the managed server (local settings).
- Network interfaces are configured under the domain controller.
- Socket bindings themselves are configured as part of the domain controller configuration, in the entries under the `SocketBindings` autogroup for the domain controller. Each server group and managed server has an offset, a number that is added to the socket bindings value, which is used to give the managed servers unique port numbers in the domain; these offsets are set on the server group and managed server connection settings.
- System properties can be set on almost any server resource: the domain controller, host controller, server group, managed server.

Some configuration settings (JVM definitions and system properties) can be defined at different levels: domain, server group, or managed server. In that case, the configuration works in a cascade, with the lowest-level configuration taking precedence. So, server group configuration trumps domain settings, and managed server settings supersede server group settings. For those configuration settings, **be sure to set the configuration at the appropriate level in the domain hierarchy**. To apply settings to the entire domain, edit the relevant domain entry; to set it at the server group or server level, create or edit the configuration at that entry level.

31.3.2.2. Requried Management Interfaces on EAP 6

The EAP 6 plug-in in JBoss ON connects to the default HTTP management interface for the EAP 6 domain controller. This management interface is used to manage and monitor the EAP 6 domain instance.

If the HTTP management interface has been removed or disabled, the agent (using the EAP 6 plug-in) will not be able to connect to the EAP 6 domain instance. Therefore, it cannot manage or monitor the EAP 6 domain resource and the resource will appear to be unavbailable, even if it is running.

If necessary, enable the HTTP management interface for the JBoss ON agent to connect to, using the EAP 6 CLI:

```
/host=instanceName/core-service=management/management-interface=http-
interface:add(interface=http,port="\${jboss.management.http.port:9990}",se
curity-realm=ManagementRealm
```

31.3.2.3. Configuration Features in JBoss ON

JBoss ON tracks all configuration changes that are made to JBoss ON resources (through the JBoss ON UI or CLI). The emphasis on JBoss ON is not only on making configuration changes; it is on managing configuration. As with other management areas in JBoss ON, the configuration maintains its history. This allows administrators to manage configuration in the context of changes and their performance or maintenance implications:

- View the change history, including diffs between versions
- Rollback changes to any previous version, simply by clicking a button
- Track which users made changes, as part of an audit trail
- Use alerting to notify administrators of any configuration changes
- Define drift monitoring to track configuration changes against a defined baseline and to control unexpected configuration changes

For each resource, JBoss ON breaks out two types of configuration: general resource properties (in the **`Configuration`** tab) and connection properties that the agent uses to connect to the resource (in the **`Inventory`** tab). Both types of configuration have configuration histories, can be reverted to previous versions, and can be used for alerting and monitoring. In reality, editing either type of configuration could end up editing the same configuration file on the resource. These two configuration areas are separated in JBoss ON to help differentiate between the configuration that affects resource behavior and the configuration that affects connections to the resource.

Even in cases when the configuration or connection settings cannot be edited, JBoss ON will still let administrators view what configuration is being applied to that resource. This is particularly useful for managed servers, which use configuration defined in the profile resources.

31.3.3. Creating Management Users

The JBoss ON agents connect to the EAP 6 server as a management user. This enables JBoss ON to perform tasks like changing the configuration or starting and stopping resources.

Having a management user is a requirement for JBoss ON to be able to manage EAP 6 instances.

There are several different ways to create a management user:

- Using an LDAP directory or external data store. This is the most secure implementation for EAP 6 and is recommended.
- Creating a management user through JBoss ON.
- Creating a local EAP account through the EAP **`add-user`** script.

Any of these are valid, according to the security implementations and needs of the local EAP 6 instance, as long as a username and password is set in the EAP 6 resource connection properties.

31.3.3.1. Setting the Credentials for a Management User

JBoss ON requires a username and password to connect to an EAP 6 server. Any EAP 6 user can be used, so long as that user has the appropriate permissions to carry out operations, application deployments, configuration changes, and other maintenance for the resource.

JBoss ON is agnostic about the EAP user configuration; it only sends the credentials to EAP, and then the EAP server processes the authentication request. Therefore, even if EAP is configured to use an LDAP directory as a user store or other security realms, JBoss ON does not need that information. It only needs the username and password.

The management user is defined in the connection properties for the EAP resource.

1. Click the **`Inventory`** tab in the top menu.

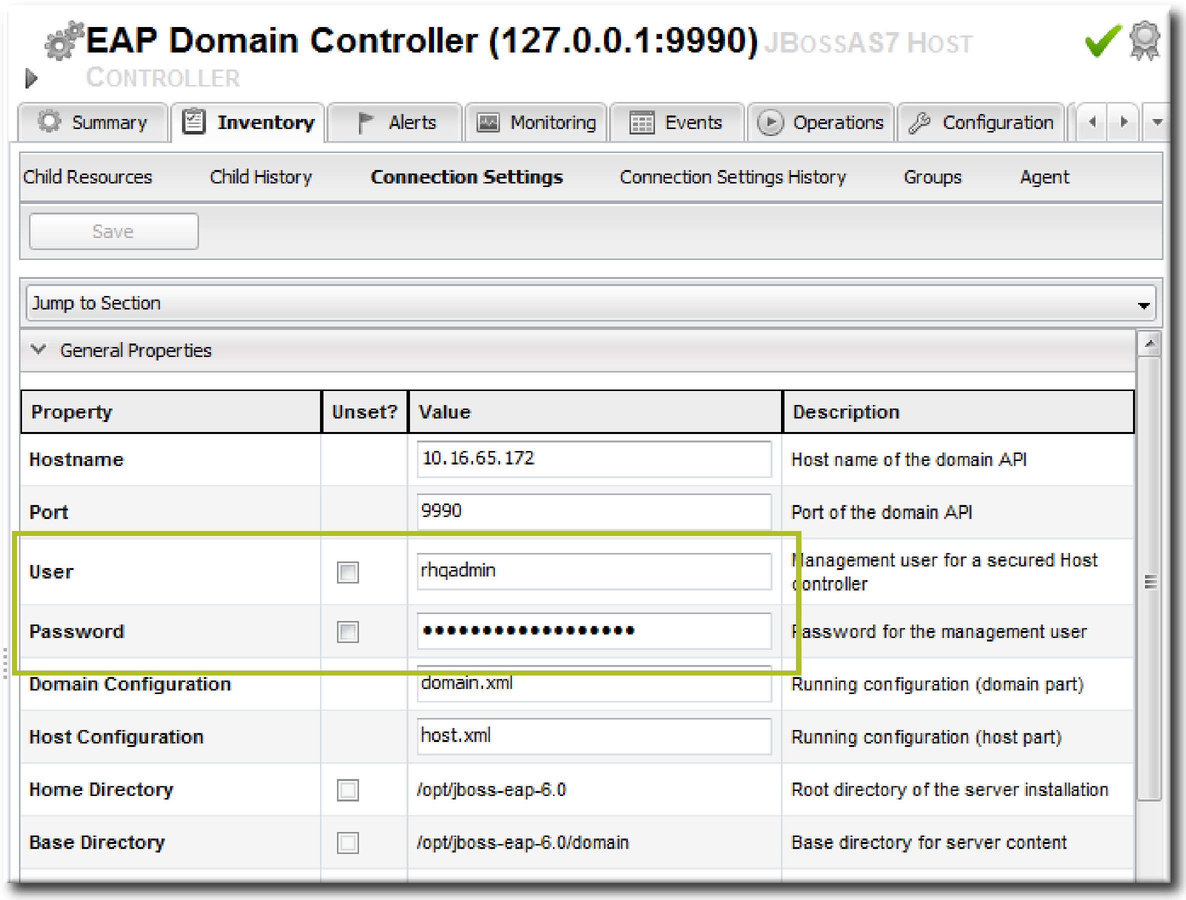
2. Select **Servers - Top Level Imports** in the **Resources** menu table on the left. Select the JBoss EAP 6 server, either the standalone server or the domain controller.

3. In the inventory tree, select the top resource entry for the server.

4. Open the **Inventory** tab.

5. Select the **Connection Settings** subtab.

6. Fill in the username and password for the management user that was created in EAP 6.



Property	Unset?	Value	Description
Hostname		10.16.65.172	Host name of the domain API
Port		9990	Port of the domain API
User	☐	rhqadmin	Management user for a secured Host controller
Password	☐	••••••••••••••••••	Password for the management user
Domain Configuration		domain.xml	Running configuration (domain part)
Host Configuration		host.xml	Running configuration (host part)
Home Directory	☐	/opt/jboss-eap-6.0	Root directory of the server installation
Base Directory	☐	/opt/jboss-eap-6.0/domain	Base directory for server content

7. Click the **Save** button at the top of the page.

31.3.3.2. Creating a Management User Through JBoss ON

There is a resource operation which creates and configures a local EAP user which the agent can use to connect to the EAP 6 instance. This is a wrapper for the EAP 6 **add-user** utility, with predefined configuration for the user and additional configuration for JBoss ON.

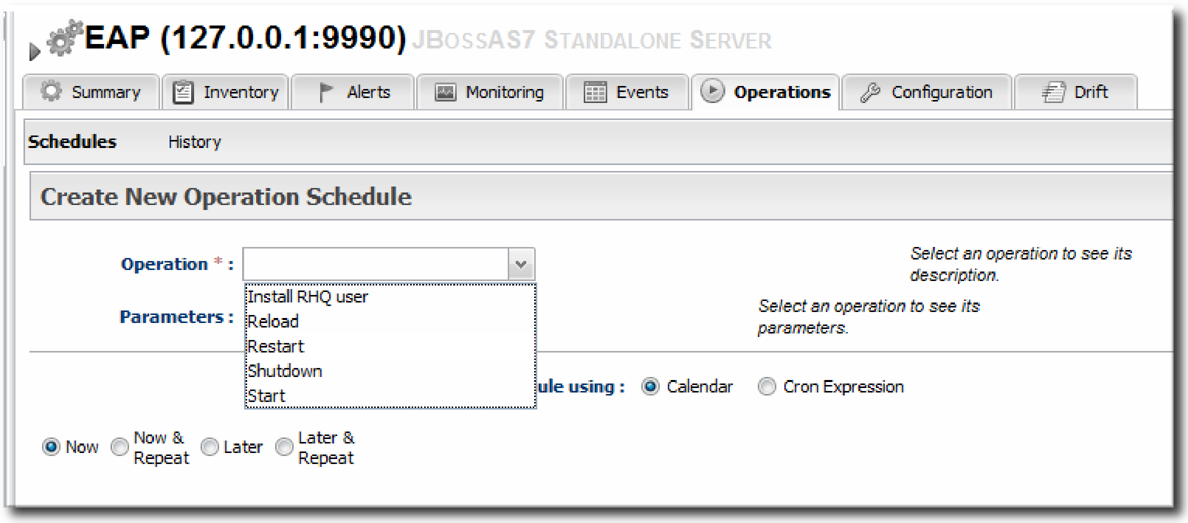
The operation creates the **rhqadmin** user in the **ManagementRealm** for the EAP 6 server. This is the default security realm; in production environments, it is strongly recommended that you use a different management realm.

The user is created in the *secure* management realm. If the EAP 6 server is running in **unsecure** mode, then the *install RHQ user* operation fails because JBoss ON cannot connect to the management realm. EAP 6 servers are secured by default.

NOTE

This operation is convenient because it sets the configuration on both EAP 6 and JBoss ON in a single step, but it has some security limitations: **It only creates the user in the default management realm (ManagementRealm). If any other security realm is used — which is recommended for production environments — then this operation, like the add-user.sh script, cannot be used.**

1. Click the `Inventory` tab in the top menu.
2. Select `Servers - Top Level Imports` in the `Resources` menu table on the left. Select the JBoss EAP 6 server, either the standalone server or the domain controller.
3. In the inventory tree, select the top resource entry for the server.
4. Open the `Operations` tab.
5. Click the **New** button at the bottom of the page.
6. Select the `Install RHQ User` option from the drop-down menu.



7. Click the `Schedule` button.

31.3.3.3. Creating a Management User in EAP 6

The EAP 6 **add-user.sh** utility creates and configures a local EAP user which the agent can use to connect to the EAP 6 instance. After creating the user in EAP, that user's credentials must be supplied to the resource's connection properties configuration in JBoss ON.

NOTE

Creating a user through the **add-user.sh** script has some security limitations: **It only creates the user in the default management realm (ManagementRealm). If any other security realm is used — which is recommended for production environments — then this script cannot be used to create a user for the JBoss ON agent.**

1. Run the **add-user** utility to create the user.

```
[root@server ~]# cd /opt/jboss-eap-6.0

[root@server bin]# ./add-user.sh
What type of user do you wish to add?
 a) Management User (mgmt-users.properties)
 b) Application User (application-users.properties)
(a): a

Enter the details of the new user to add.
Realm (ManagementRealm) :
Username : jonadmin
Password :
Re-enter Password :
About to add user 'jonadmin' for realm 'ManagementRealm'
Is this correct yes/no? yes
```

2. Set that user in the connection settings for the EAP 6 server resource in JBoss ON.

   1. Click the **Inventory** tab in the top menu.

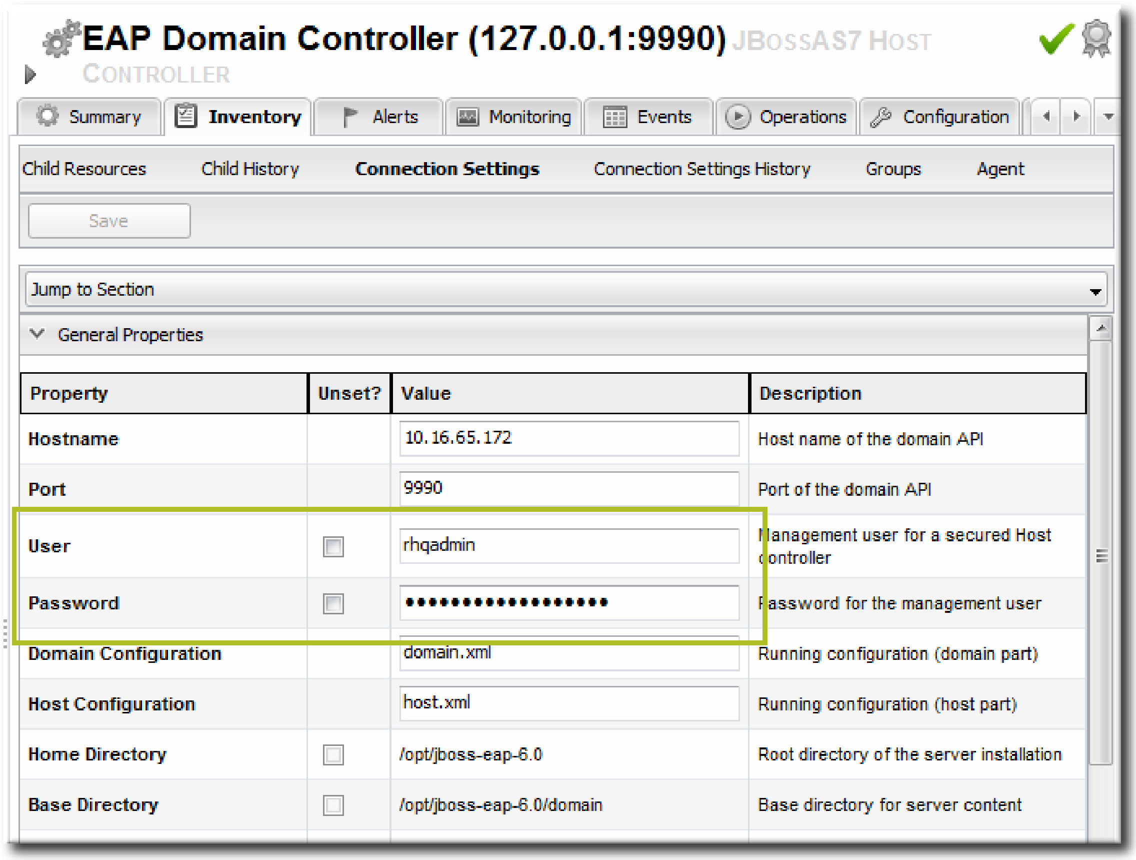
   2. Select **Servers - Top Level Imports** in the **Resources** menu table on the left. Select the JBoss EAP 6 server, either the standalone server or the domain controller.

   3. In the inventory tree, select the top resource entry for the server.

   4. Open the **Inventory** tab.

   5. Select the **Connection Settings** subtab.

   6. Fill in the username and password for the management user that was created in EAP 6.



Property	Unset?	Value	Description
Hostname		10.16.65.172	Host name of the domain API
Port		9990	Port of the domain API
User	☐	rhqadmin	Management user for a secured Host controller
Password	☐	••••••••••••••••••	Password for the management user
Domain Configuration		domain.xml	Running configuration (domain part)
Host Configuration		host.xml	Running configuration (host part)
Home Directory	☐	/opt/jboss-eap-6.0	Root directory of the server installation
Base Directory	☐	/opt/jboss-eap-6.0/domain	Base directory for server content

7. Click the **Save** button at the top of the page.

31.3.4. Creating a Dynamic Group for EAP 6 Resources

For management, particularly for monitoring or for editing configuration, it can be very convenient to have related resources grouped together in a compatible group. Compatible groups let administrators set alert definitions, change metrics settings, and change configuration for all group members simultaneously.

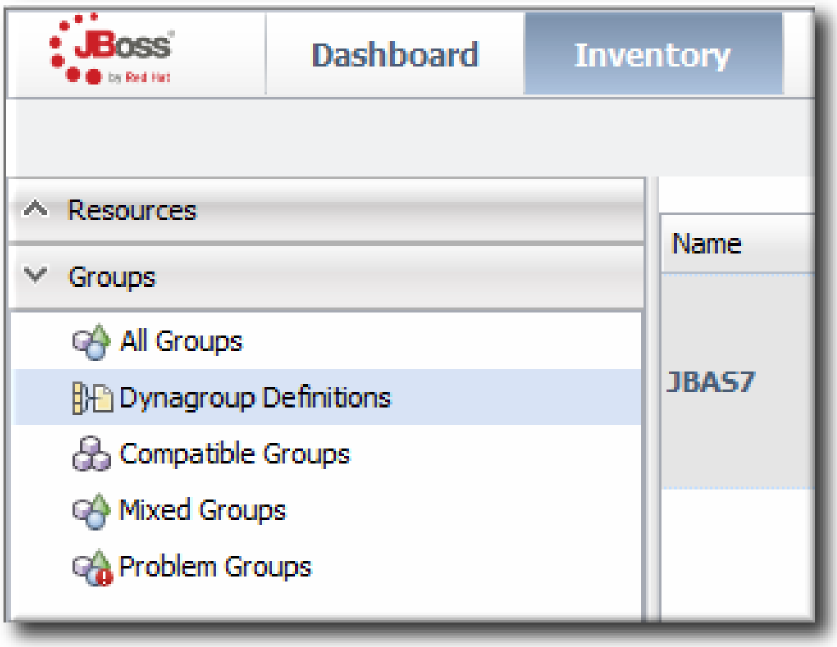
Dynagroups use search expressions to search for group members and then create groups based on any user-defined criteria. This assures that group membership is always current, since members are added and dropped automatically as the JBoss ON inventory changes.

The dynagroup syntax is described in much more detail in Initial Setup for the Resource Inventory, Groups, and Users. This specific dynagroup expression first searches for resources based on the resource plug-in, category (server), and parent. It then creates groups based on the product type and name.

This creates a separate compatible group for every JBoss product associated with JBoss EAP 6, such as SOA-P, Data Grid (JDG), and host controllers and standalone servers.

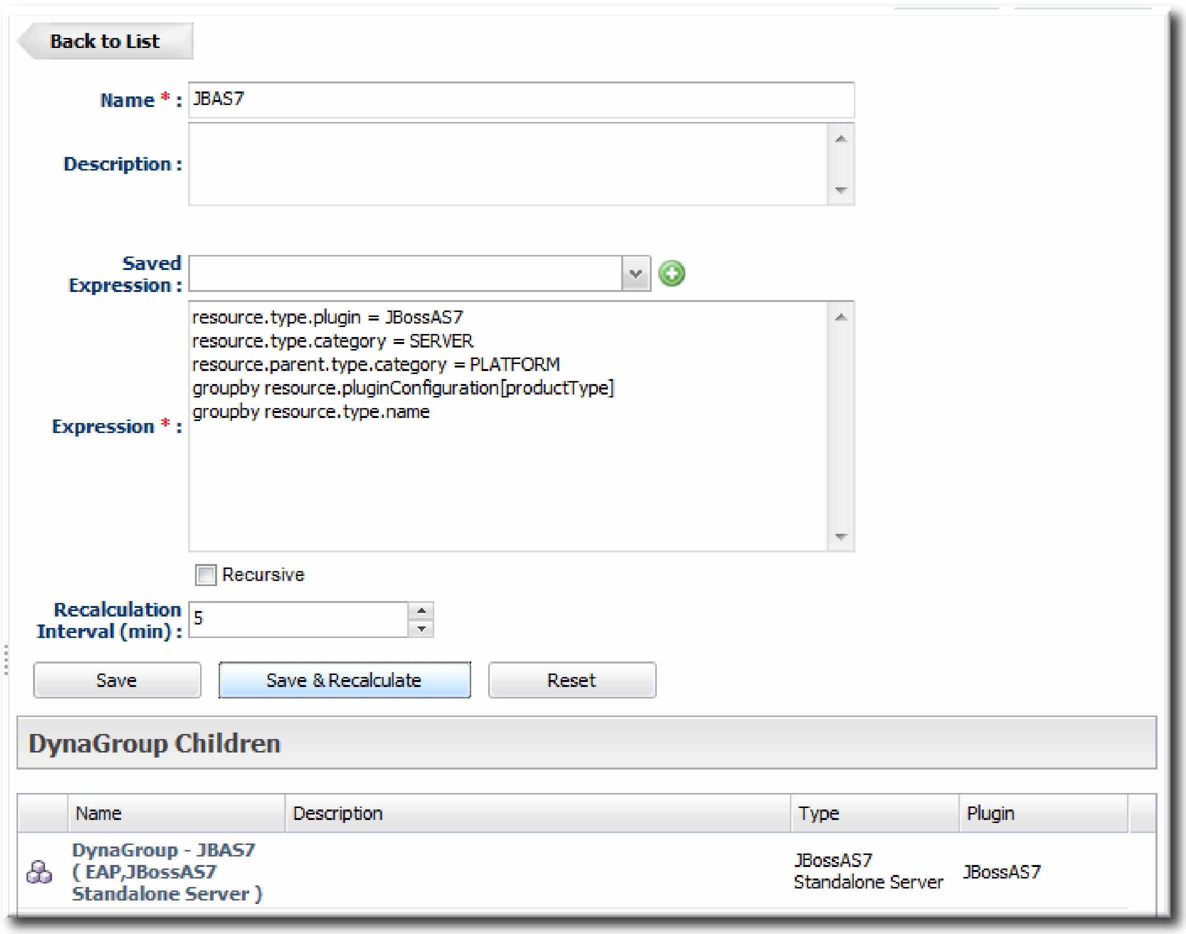
1. Click the **`Inventory`** tab in the top menu.

2. In the **`Groups`** area on the left, click the **`Dynagroup Definitions`** link.



3. Enter the expression to create compatible groups for each EAP 6 server type.

```
resource.type.plugin = JBossAS7
resource.type.category = SERVER
resource.parent.type.category = PLATFORM
groupby resource.pluginConfiguration[productType]
groupby resource.type.name
```



Name * : JBAS7

Description :

Saved Expression :

Expression * :

```
resource.type.plugin = JBossAS7
resource.type.category = SERVER
resource.parent.type.category = PLATFORM
groupby resource.pluginConfiguration[productType]
groupby resource.type.name
```

☐ Recursive

Recalculation Interval (min) : 5

	Name	Description	Type	Plugin
	DynaGroup - JBAS7 (EAP,JBossAS7 Standalone Server)		JBossAS7 Standalone Server	JBossAS7

4. Click the **Save** button in the middle of the page.

31.3.5. Setting Start Script Arguments, Environment Variables, and JAVA_OPTS

31.3.5.1. Start Script Discovery and Settings

As part of discovering a JBoss EAP 6 server (domain controller or standalone server), JBoss ON attempts to discover the environment that the server is running in. Specifically, the discovery process attempts to identify and recreate the runtime environment:

- The discovery process identifies, or attempts to identify, the start script used, including custom start scripts.
- Discovery detects a subset of environment variables set in the **run.conf** file or parent process that are required for the start script to work.

NOTE

Although the discovery process does detect some environment variables, **the discovery scan does not detect JAVA_OPTS values.**

The connection properties for the start script intentionally defer to the **run.conf** file for **JAVA_OPTS** values.

- Discovery attempts to detect any arguments passed with the start script itself.
- Discovery attempts to detect what user the script is running as and assign a *prefix* command

to use with the start script. For example, if the start script is running as the **`jboss`** user and the JBoss ON agent is running as **`jonagent`**, then the discovery script automatically assigns a **`sudo`** command, **`sudo -u jboss -g jboss`**, to pass with the start script.

The discovered settings are stored in the **`Start Script Environment Variables`** and **`Start Script Arguments`** connection settings for the EAP 6 resource.

If the discovery scan cannot detect some of the script settings (such as the agent is running as a different user than the EAP 6 server and cannot read the parent process), it fails gracefully. It simply gathers whatever information it can and ignores blank values.

NOTE

The **`Start Script Environment Variables`** and **`Start Script Arguments`** connection settings are only discovered once, when the resource is initially discovered. After that, the connection settings can be changed in the connection settings, but any changes made on the local system will not be detected. For example, if the server is discovered with a particular value (like **`-XX:PermSize=256M`**), the argument value will not be updated if the server is restarted later with a different setting value.

The script arguments and environment variables are the only ones used by the agent when it runs the start script. These arguments and environment variables are not added to other configuration settings or the parent process. The start script settings in the EAP 6 connection settings configuration are deterministic.

31.3.5.2. Start Script Arguments and Drift Monitoring

JBoss ON can monitor directories or specific files to check for *configuration drift*, which means configuration changes that move away from a designated configuration state. This is described in more detail in Section 31.3.13, “Controlling Configuration Drift” and the drift chapters of "Managing Resource Configuration."

While drift monitoring is critical for administrators to manage important resources, there may be times when it is necessary or beneficial from that configuration in a few settings. The start script arguments can be used to override the defined configuration for an EAP 6 server *without* triggering a drift alert.

Since the start script options are all connection settings, every change is recorded in a change history and can be easily viewed and reverted, as in Section 31.3.14, “Tracking and Reverting Configuration Changes”. This keeps the system properties and Java settings trackable and remediable.

Example 31.2. System Properties Without Violating the Drift Definition

Tim the IT Guy creates one specific set of configuration that all production EAP 6 servers should use an environment. He then creates a drift definition template for monitoring and associates or pins that blessed configuration to the template.

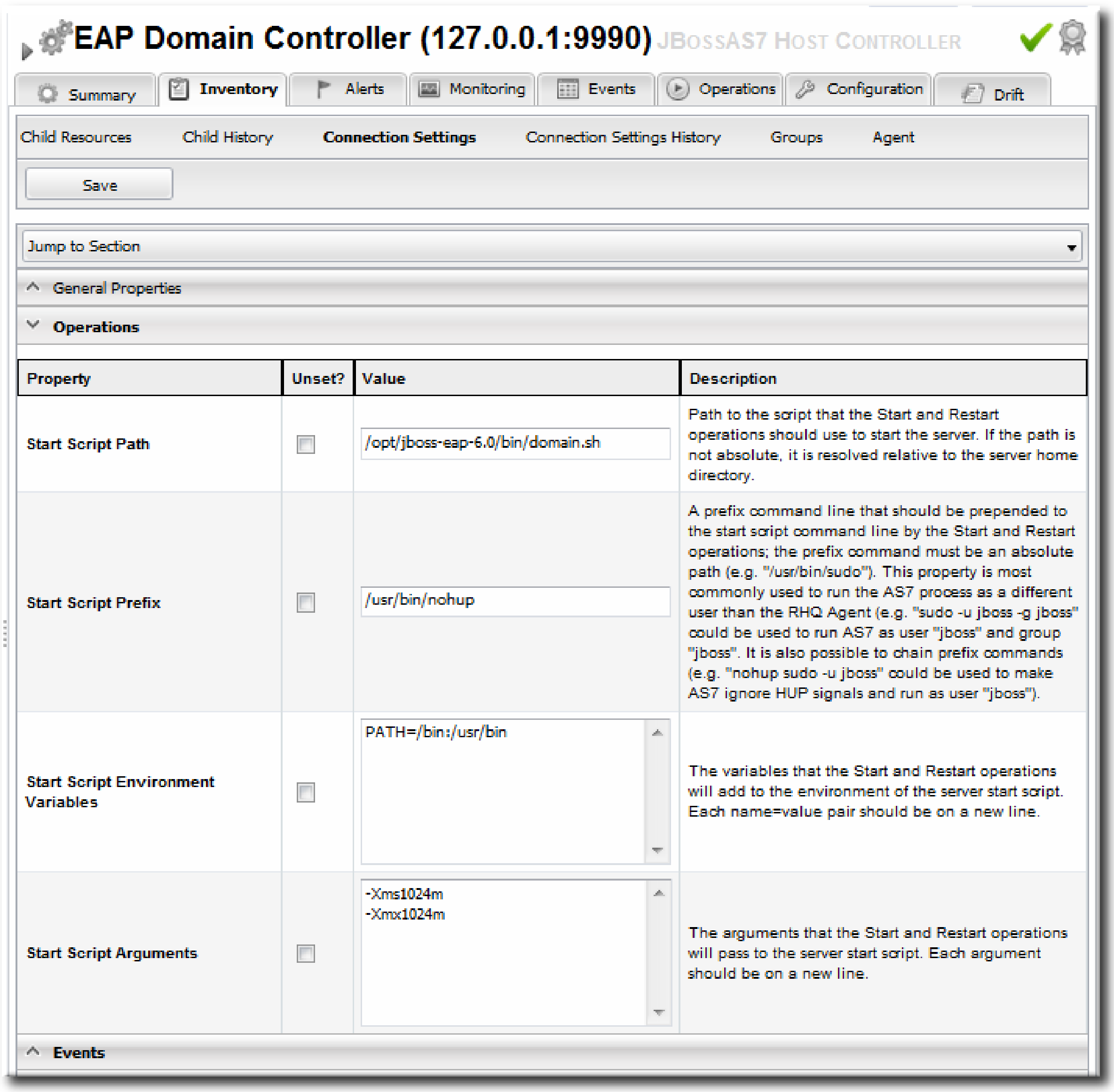
Every EAP 6 server which uses that drift template must conform to those configuration settings. Tim the IT Guy creates an alert that informs him if any production server drifts away from that specified configuration. This is an important safety measure for Example Co.'s production application servers.

However, a couple of the production servers have slightly different hardware and other applications running on them, so they require different heap sizes to run effectively.

> If Tim the TI Guy adds a system property to use a different heap size, he is going to receive constant drift alerts or his edited configuration could be overwritten if he runs an automatic server-side script to remediate drift configuration.
>
> By setting different heap settings through the start script, Tim can apply the right settings for that system without editing a configuration file, so there is no alert-able configuration drift.

31.3.5.3. Changing Start Script Configuration

1. Click the **`Inventory`** tab in the top menu.
2. Select **`Servers - Top Level Imports`** in the **`Resources`** menu table on the left. Select the JBoss EAP 6 server.
3. In the inventory tree, select the top resource entry for the server.
4. Open the **`Inventory`** tab, and select the **`Connection Settings`** subtab.
5. Expand the **`Operations`** area.
6. Change or add start script settings. These are the scripts and settings that the JBoss ON agent uses when running a start or restart operation on the EAP 6 server.



Property	Unset?	Value	Description
Start Script Path	☐	/opt/jboss-eap-6.0/bin/domain.sh	Path to the script that the Start and Restart operations should use to start the server. If the path is not absolute, it is resolved relative to the server home directory.
Start Script Prefix	☐	/usr/bin/nohup	A prefix command line that should be prepended to the start script command line by the Start and Restart operations; the prefix command must be an absolute path (e.g. "/usr/bin/sudo"). This property is most commonly used to run the AS7 process as a different user than the RHQ Agent (e.g. "sudo -u jboss -g jboss" could be used to run AS7 as user "jboss" and group "jboss". It is also possible to chain prefix commands (e.g. "nohup sudo -u jboss" could be used to make AS7 ignore HUP signals and run as user "jboss").
Start Script Environment Variables	☐	PATH=/bin:/usr/bin	The variables that the Start and Restart operations will add to the environment of the server start script. Each name=value pair should be on a new line.
Start Script Arguments	☐	-Xms1024m -Xmx1024m	The arguments that the Start and Restart operations will pass to the server start script. Each argument should be on a new line.

- To use a custom start script, one other than **`domain.sh`** or **`standalone.sh`**, enter the path and script name.

- Optionally, enter a prefix to use with the script when running the start script.

  When the start script is discovered, the agent tries to determine the user the script is running as and assign a prefix command to use with the start script. For example, if the start script is running as the **`jboss`** user and the JBoss ON agent is running as **`jonagent`**, then the discovery script automatically assigns a **sudo** command, **`sudo -u jboss -g jboss`**, to pass with the start script.

  Additionally, JBoss ON assigns the **nohup** command as a prefix so that if the JBoss Enterprise Application Platform is started by the agent and the agent process dies, the JBoss Enterprise Application Platform process continues running.

- Set any environment variables, one per line.

- Set any script arguments, one per line. For regular JAVA_OPTS, these arguments usually are **-X**, **-D**, or **-P**. Some useful **-XX** arguments are listed in the JVM options documentation from Sun. Some useful system properties for EAP 6 are listed with the JBoss AS7 project documentation.

The EAP 6 default start scripts use a `run.sh`-style script, so the arguments use that format. A custom script can use different arguments or options.

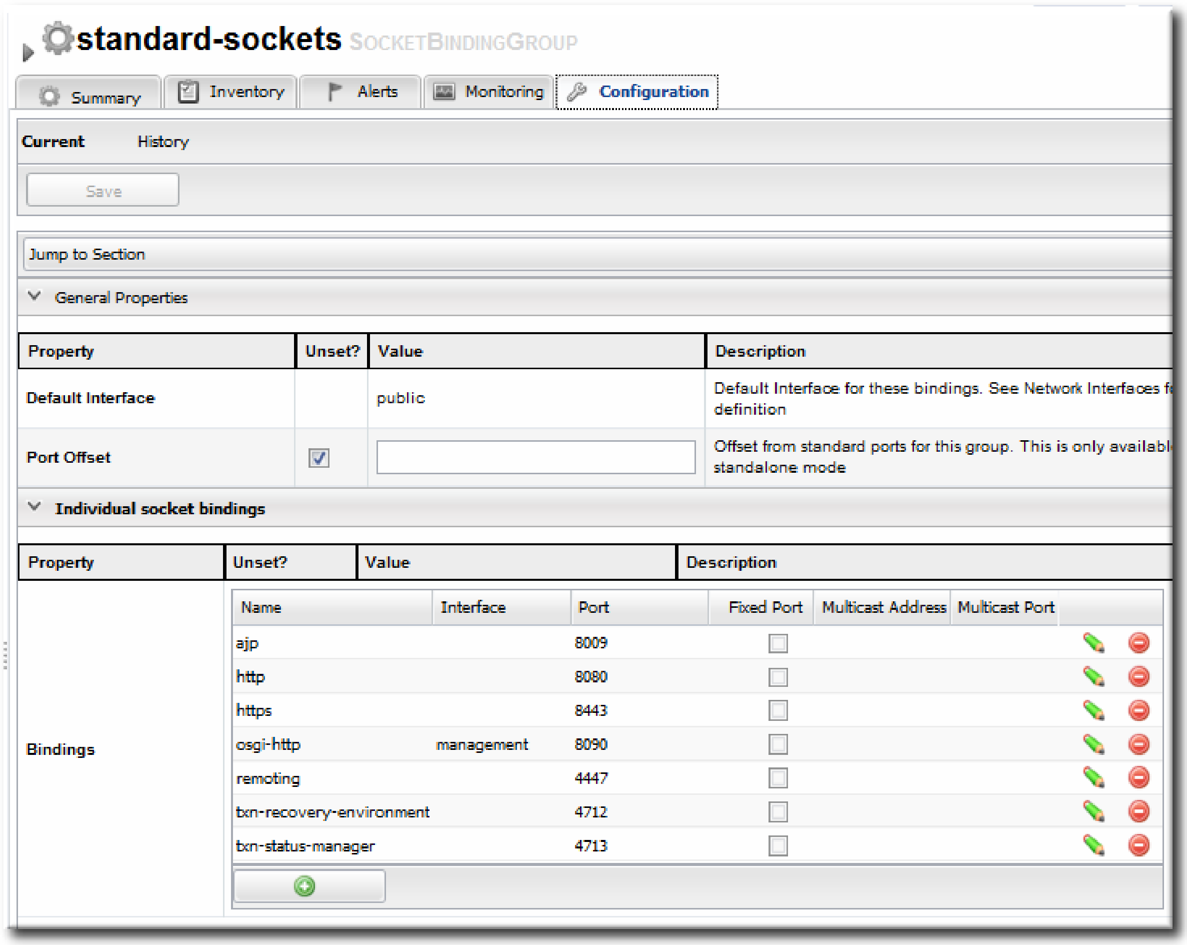
7. Click the **Save** button at the top of the page.

31.3.6. Changing Port Numbers

31.3.6.1. Changing Socket Binding Ports

The socket binding resources defines both what ports are available (such as HTTP, AJP, and HTTPS) and what those port numbers are. The socket binding configuration can also configure multicast port numbers for the sockets.

1. Click the **Inventory** tab in the top menu.
2. Select **Servers - Top Level Imports** in the **Resources** menu table on the left. Select the JBoss EAP 6 server.
3. In the inventory tree, select the **SocketBindingsGroup** compatible group, and then select the socket binding to edit.
4. Open the **Configuration** tab.
5. Click the green pencil icon to edit an existing socket definition or click the green plus sign (+) to create a new one.

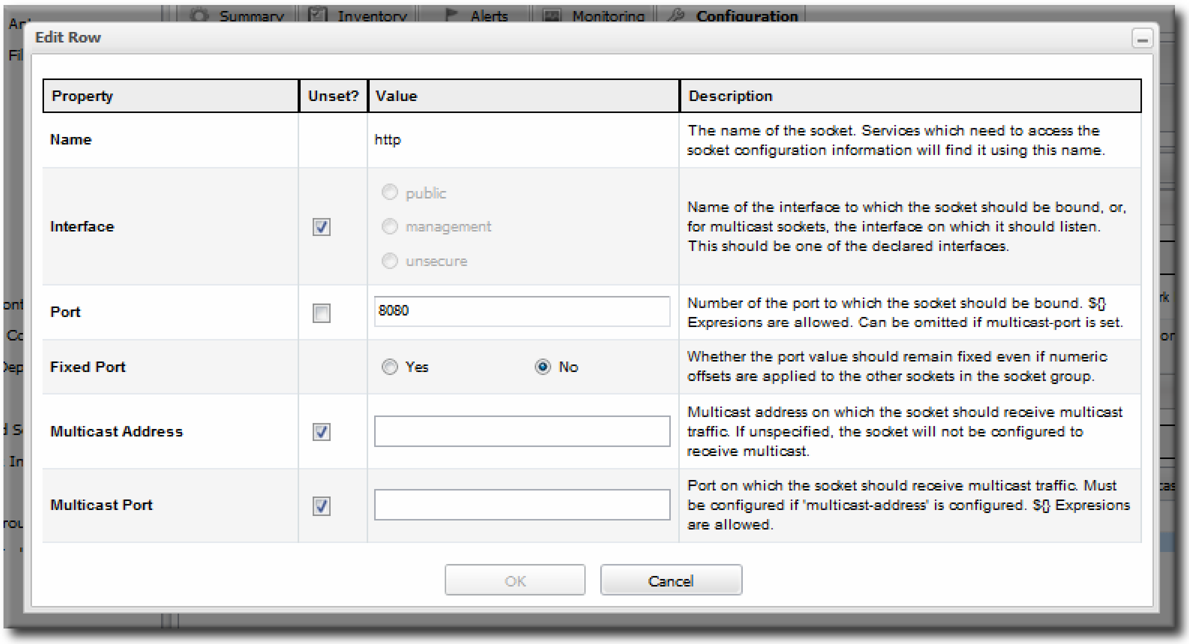


Property	Unset?	Value	Description
Default Interface		public	Default Interface for these bindings. See Network Interfaces f definition
Port Offset	✓		Offset from standard ports for this group. This is only availabl standalone mode

Property	Unset?	Value	Description
Bindings			

Name	Interface	Port	Fixed Port	Multicast Address	Multicast Port
ajp		8009	☐		
http		8080	☐		
https		8443	☐		
osgi-http	management	8090	☐		
remoting		4447	☐		
txn-recovery-environment		4712	☐		
txn-status-manager		4713	☐		

6. Change the **Port** number to any available port between 1025 and 65535. On Linux, available port numbers can be determined using `iptables`.



Property	Unset?	Value	Description
Name		http	The name of the socket. Services which need to access the socket configuration information will find it using this name.
Interface	✓	public / management / unsecure	Name of the interface to which the socket should be bound, or, for multicast sockets, the interface on which it should listen. This should be one of the declared interfaces.
Port		8080	Number of the port to which the socket should be bound. ${} Expresions are allowed. Can be omitted if multicast-port is set.
Fixed Port		Yes [✓ No]	Whether the port value should remain fixed even if numeric offsets are applied to the other sockets in the socket group.
Multicast Address	✓		Multicast address on which the socket should receive multicast traffic. If unspecified, the socket will not be configured to receive multicast.
Multicast Port	✓		Port on which the socket should receive multicast traffic. Must be configured if 'multicast-address' is configured. ${} Expresions are allowed.

Optionally, configure multicast settings for the socket. If there are multiple instances of JBoss servers on the same system or in the same cluster, then multicast may be configured for cluster communication.

7. Click the **Save** button at the top of the page.

31.3.6.2. Changing Port Offsets for Server Groups in a Domain

The port offset is a number added to each port number in a socket binding group to create the actual port numbers used by a server instance. This allows managed servers to have unique port numbers across the domain or for standalone servers within a cluster to have unique ports while using the same socket binding configuration.

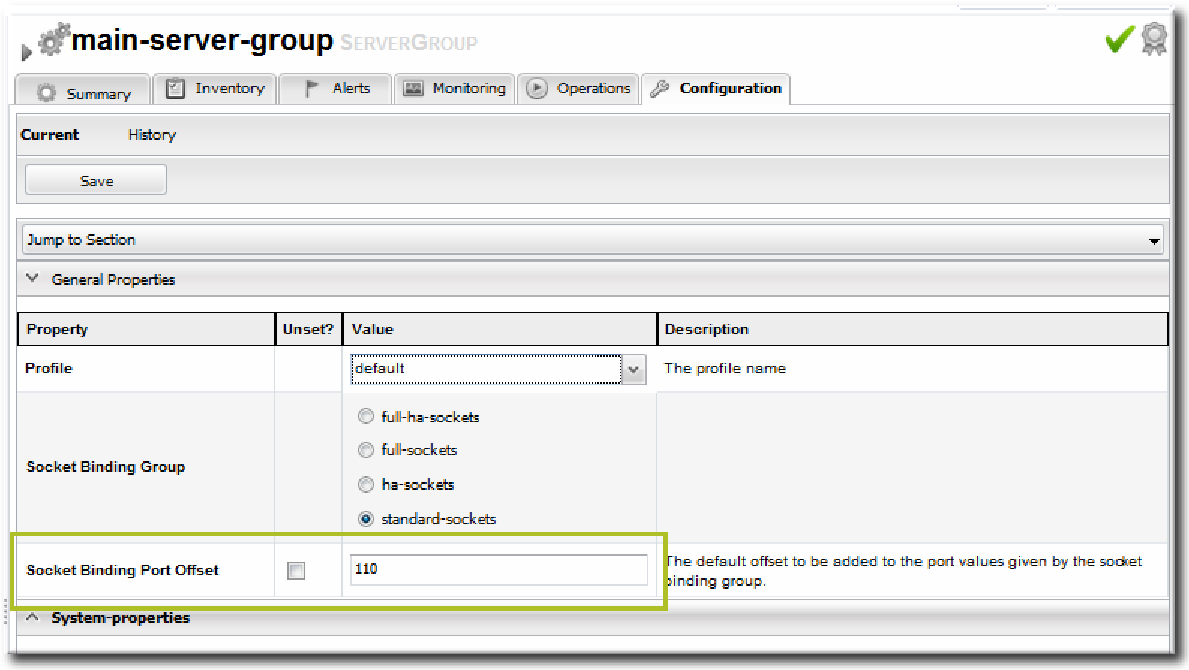
For example, if the socket binding HTTP port is 8080, and a managed server has an offset of 110, the actual port used by the managed server is 8190 (8080 + 110). Server groups can also define offsets, and the offsets are additive. So, if a server group has an offset of 15 and the managed server has an offset of 110, the port number used by the server is 8205 (8080 + 15 + 110).

NOTE

The offset for a managed server is defined in its entry in the **`host.xml`** file. This can be set when the managed server is created in JBoss ON, but it cannot be edited afterward.

It is possible to edit the port offset used by a server group:

1. Click the **`Inventory`** tab in the top menu.
2. Select **`Servers - Top Level Imports`** in the **`Resources`** menu table on the left. Select the JBoss EAP 6 server.
3. In the inventory tree, expand the **`Server Groups`** node, and select the server group.
4. Open the **`Configuration`** tab for the server group.
5. In the **`Port Offset`** field, enter the new value for the offset.



Property	Unset?	Value	Description
Profile		default	The profile name
Socket Binding Group		full-ha-sockets / full-sockets / ha-sockets / [✓ standard-sockets]	
Socket Binding Port Offset	☐	110	The default offset to be added to the port values given by the socket binding group.

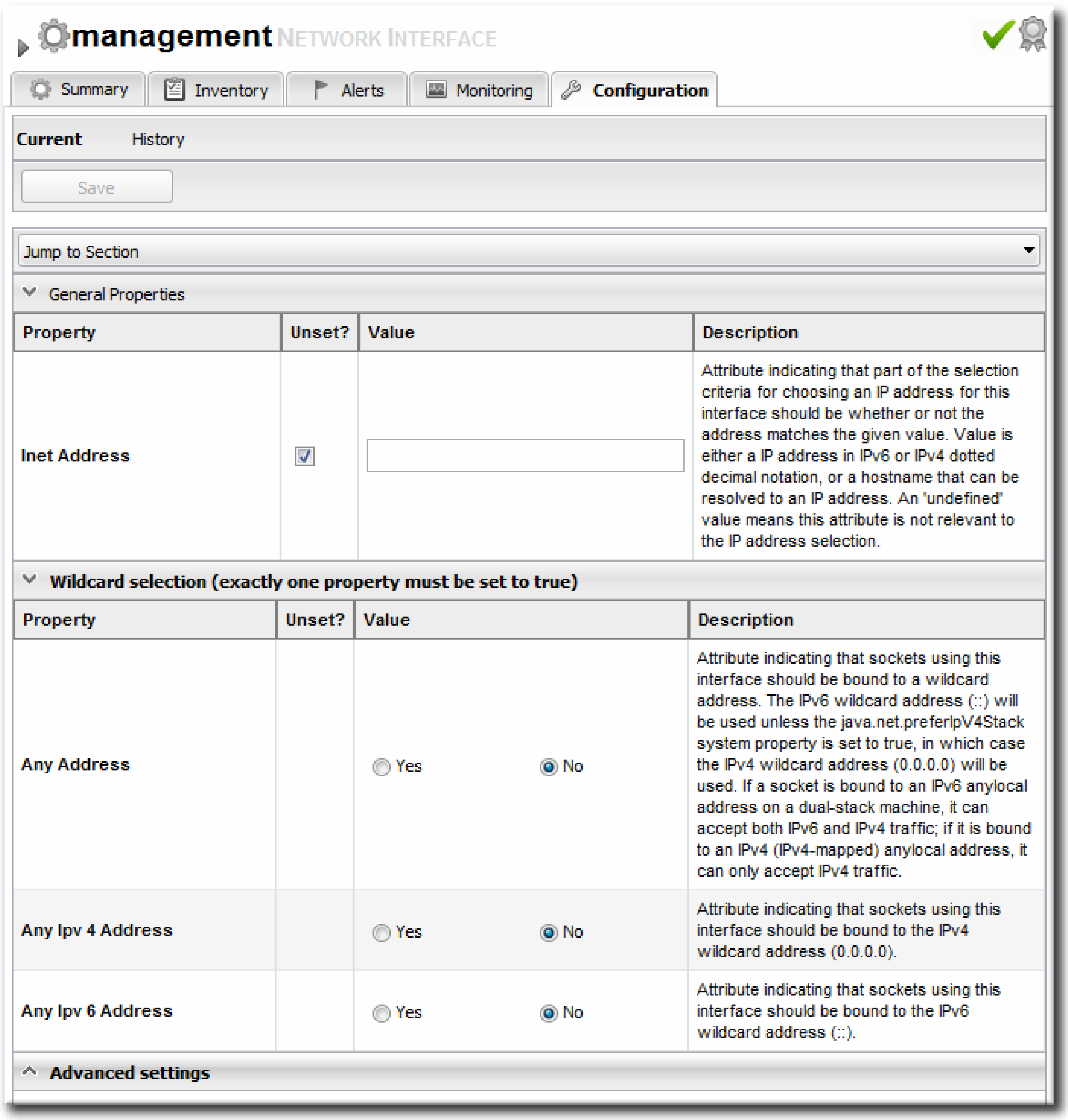
6. Click **Save** at the top of the page.

To change the offset for a standalone server, edit its connection settings, as in Section 31.3.10.1, "Changing the General Properties for an EAP 6 Server".

31.3.7. Editing Network Interfaces

Network interfaces can be configured to use a specific IP address or a type of IP address when communicating with sockets for that interface.

1. Click the **`Inventory`** tab in the top menu.
2. Select **`Servers - Top Level Imports`** in the **`Resources`** menu table on the left. Select the JBoss EAP 6 server.
3. In the inventory tree, select the **`Network Interfaces`** group under the main server entry, and select the interface (management, public, or unsecure).
4. Open the **`Configuration`** tab.
5. Set either the specific IP address for the interface to use *or* set which type of IP address to use (IPv4, IPv6, or either). **Either the IP address or the IP address type must be set.**



Property	Unset?	Value	Description
Inet Address	✓		Attribute indicating that part of the selection criteria for choosing an IP address for this interface should be whether or not the address matches the given value. Value is either a IP address in IPv6 or IPv4 dotted decimal notation, or a hostname that can be resolved to an IP address. An 'undefined' value means this attribute is not relevant to the IP address selection.

Wildcard selection (exactly one property must be set to true)

Property	Unset?	Value	Description
Any Address		Yes [✓ No]	Attribute indicating that sockets using this interface should be bound to a wildcard address. The IPv6 wildcard address (::) will be used unless the java.net.preferIpV4Stack system property is set to true, in which case the IPv4 wildcard address (0.0.0.0) will be used. If a socket is bound to an IPv6 anylocal address on a dual-stack machine, it can accept both IPv6 and IPv4 traffic; if it is bound to an IPv4 (IPv4-mapped) anylocal address, it can only accept IPv4 traffic.
Any Ipv 4 Address		Yes [✓ No]	Attribute indicating that sockets using this interface should be bound to the IPv4 wildcard address (0.0.0.0).
Any Ipv 6 Address		Yes [✓ No]	Attribute indicating that sockets using this interface should be bound to the IPv6 wildcard address (::).

Because either a specific IP address or an IP address type can be set, and which property is used is optional, the UI does not enforce that a selection is made. For the network interface to work properly, however, some kind of IP address configuration must be set.

6. Click the **Save** button at the top of the page.

31.3.8. Setting System Properties

Servers and particularly subsystems, which are integrated services, can have additional configuration properties added as system properties. The system property is a simple attribute-value pair, and it can be anything. The system properties which are available depend on the resource being edited.

When editing profiles and subsystems, extensions, server groups, and other domain components, system properties are added to the component's entry in the `domain.xml` file. When editing a host controller or a managed server, the properties are added to the server's entry in the `host.xml` file.

NOTE

System properties, as with other configuration elements in the domain, work in a cascade. If a system property is set on a server group, it applies to all members of the group. If it is set on a managed server, it applies only to that server.

Be sure to edit the entry at the appropriate level in the domain so that the configuration is applied appropriately.

System properties set default configuration; these can be overridden with the start script in the **`-D`** or **`-P`** arguments.

NOTE

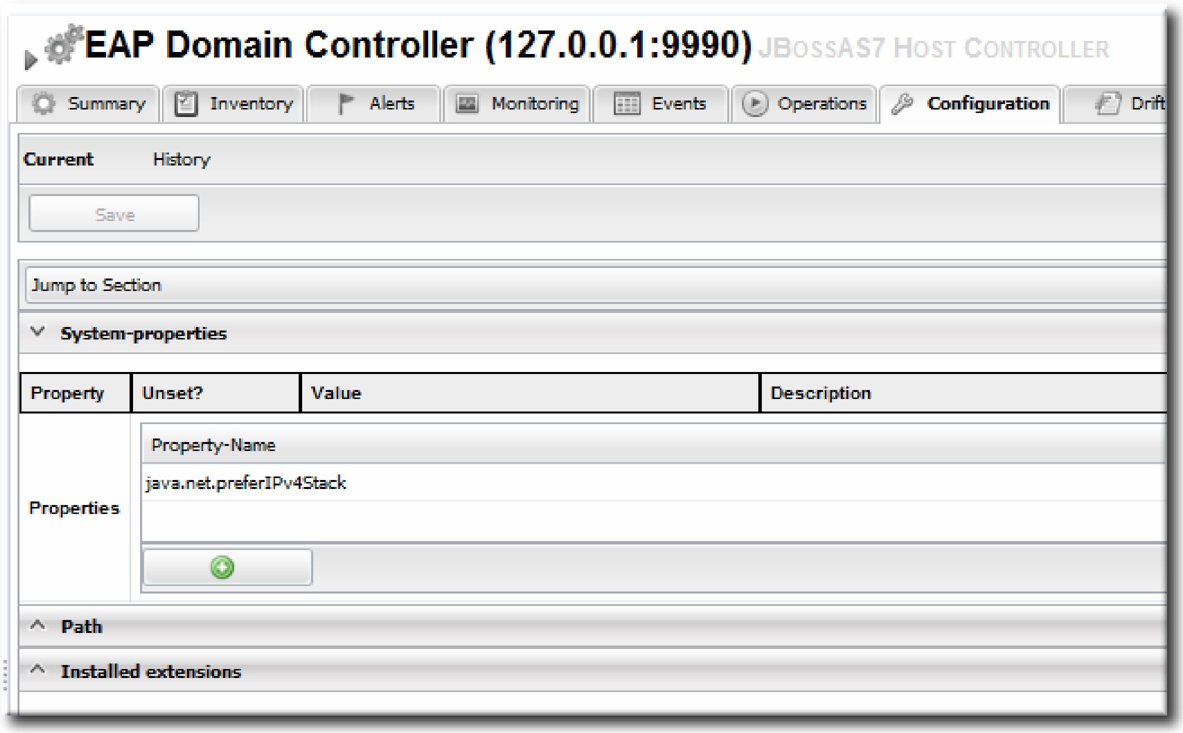
If drift monitoring is configured, then adding or editing system properties could trigger a drift alert. See Section 31.3.13, “Controlling Configuration Drift”.

Check the project documentation for information on available system properties:

- JBoss AS7
- mod_cluster
- JBossWeb

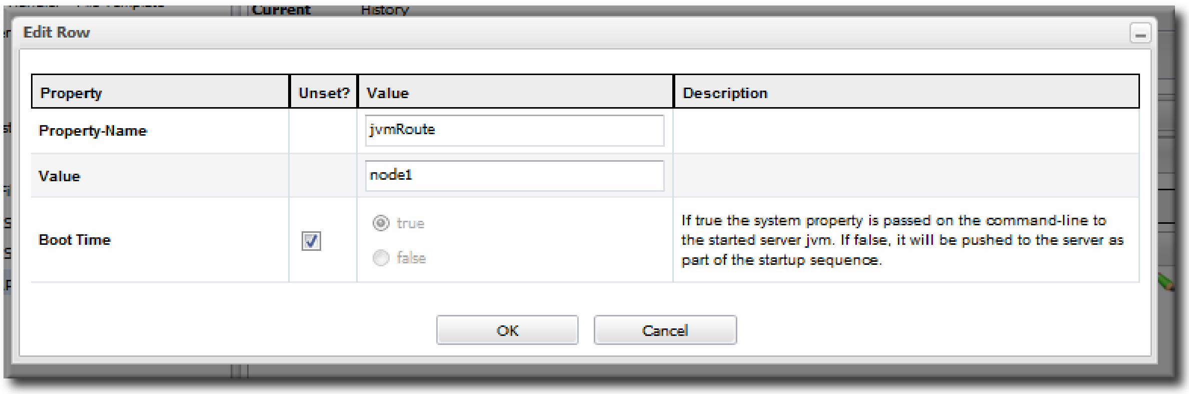
To add system properties to the configuration:

1. Click the **`Inventory`** tab in the top menu.
2. Select **`Servers - Top Level Imports`** in the **`Resources`** menu table on the left. Select the JBoss EAP 6 server.
3. In the inventory tree, select the top resource entry for the server.
4. Open the **`Configuration`** tab.
5. Expand the **`Properties`** section.



6. Click the green plus (+) icon at the bottom of the **Paths** list.

7. Fill in the new property information.



Property	Unset?	Value	Description
Property-Name		jvmRoute	
Value		node1	
Boot Time	✓	● true ○ false	If true the system property is passed on the command-line to the started server jvm. If false, it will be pushed to the server as part of the startup sequence.

   - The system property name.
   - The value of the property.
   - If the property should be loaded immediately to the running JVM or if it should be loaded when the JVM is started. The default is to load it immediately.

8. Click **OK**.

31.3.9. Adding System Paths

Some system paths are already defined for important server directory locations, like the home directory, log directory, and Java home directory. For custom applications, it may be useful or necessary to define other paths, and these can be added to the server configuration.

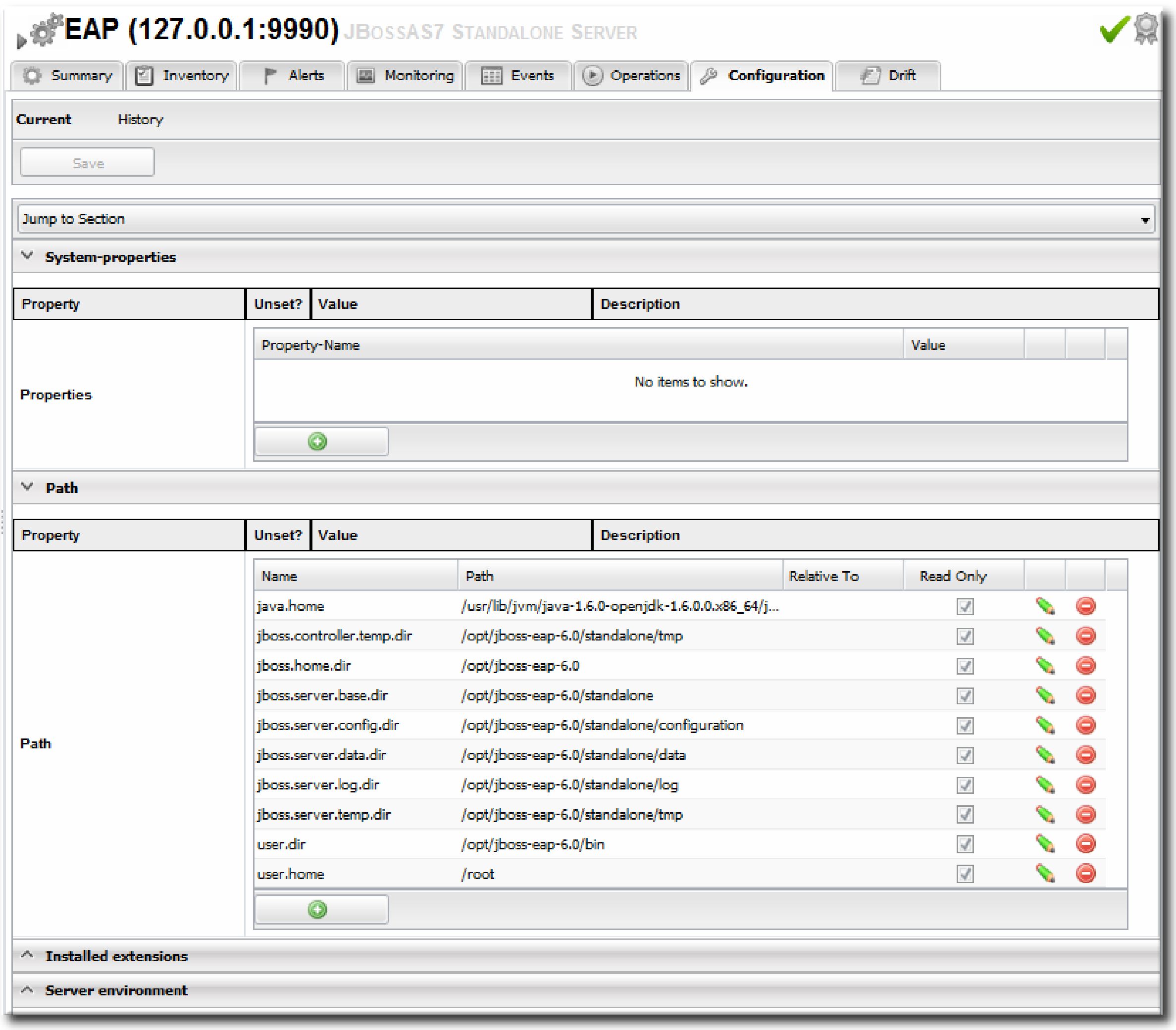
NOTE

The default system paths cannot be edited or deleted through the resource configuration. These are defined by the JBoss EAP 6 installation itself. These paths begin with the names `jboss.*`, `user.*`, and `java.*`.

Some of those default system paths, like the home directory and base directory, can be edited by editing the EAP 6 server connections settings, as in Section 31.3.10, “Editing Connection Settings”.

Edit and delete icons are displayed by these default path entries. Although an edit window comes up and the path can apparently be edited or deleted, those changes are reset immediately.

1. Click the **`Inventory`** tab in the top menu.
2. Select **`Servers - Top Level Imports`** in the **`Resources`** menu table on the left. Select the JBoss EAP 6 server.
3. In the inventory tree, select the top resource entry for the server.
4. Open the **`Configuration`** tab.
5. Expand the **`Paths`** section.



Property	Unset?	Value	Description
Properties			

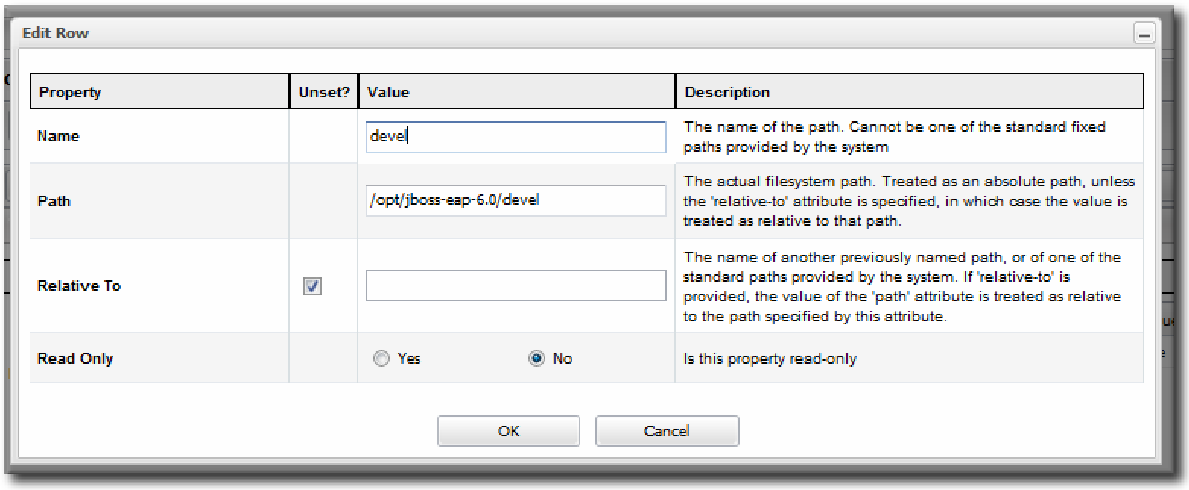
Property-Name	Value
No items to show.	

Property	Unset?	Value	Description
Path			

Name	Path	Relative To	Read Only
java.home	/usr/lib/jvm/java-1.6.0-openjdk-1.6.0.0.x86_64/j...		✓
jboss.controller.temp.dir	/opt/jboss-eap-6.0/standalone/tmp		✓
jboss.home.dir	/opt/jboss-eap-6.0		✓
jboss.server.base.dir	/opt/jboss-eap-6.0/standalone		✓
jboss.server.config.dir	/opt/jboss-eap-6.0/standalone/configuration		✓
jboss.server.data.dir	/opt/jboss-eap-6.0/standalone/data		✓
jboss.server.log.dir	/opt/jboss-eap-6.0/standalone/log		✓
jboss.server.temp.dir	/opt/jboss-eap-6.0/standalone/tmp		✓
user.dir	/opt/jboss-eap-6.0/bin		✓
user.home	/root		✓

6. Click the green plus (+) icon at the bottom of the **`Paths`** list.

7. Fill in the path information.



Edit Row

Property	Unset?	Value	Description
Name		devel	The name of the path. Cannot be one of the standard fixed paths provided by the system
Path		/opt/jboss-eap-6.0/devel	The actual filesystem path. Treated as an absolute path, unless the 'relative-to' attribute is specified, in which case the value is treated as relative to that path.
Relative To	✓		The name of another previously named path, or of one of the standard paths provided by the system. If 'relative-to' is provided, the value of the 'path' attribute is treated as relative to the path specified by this attribute.
Read Only		Yes [✓ No]	Is this property read-only

   - The name of the path to create.

   - The path (absolute or relative) to create.

   - If a relative path was given as the **`Path`** value, then de-select the **`Unset?`** checkbox for the **`Relative`** field, and enter the *name* of the system path that it is relative to.

     For example, if the new path is **`devel/`**, and this is relative to the EAP home directory, then the **`Relative`** value is *java.home.dir*. This results in a final path of **`/opt/jboss-eap-6.0/devel/`**.

   - If the property is read-only. A read-only property **cannot** be edited after it is created. Read-only paths (aside from the default paths) have to be deleted and recreated if they need to be changed.

8. Click **`OK`**.

31.3.10. Editing Connection Settings

31.3.10.1. Changing the General Properties for an EAP 6 Server

The connection settings are the properties that the JBoss ON agent uses to connect to a resource; the connection settings to use are identified in the plug-in descriptor for the resource.

Connection settings are only configurable for a domain controller or standalone server; all of the child resource connection properties are derived from the main server configuration.

1. Click the **`Inventory`** tab in the top menu.

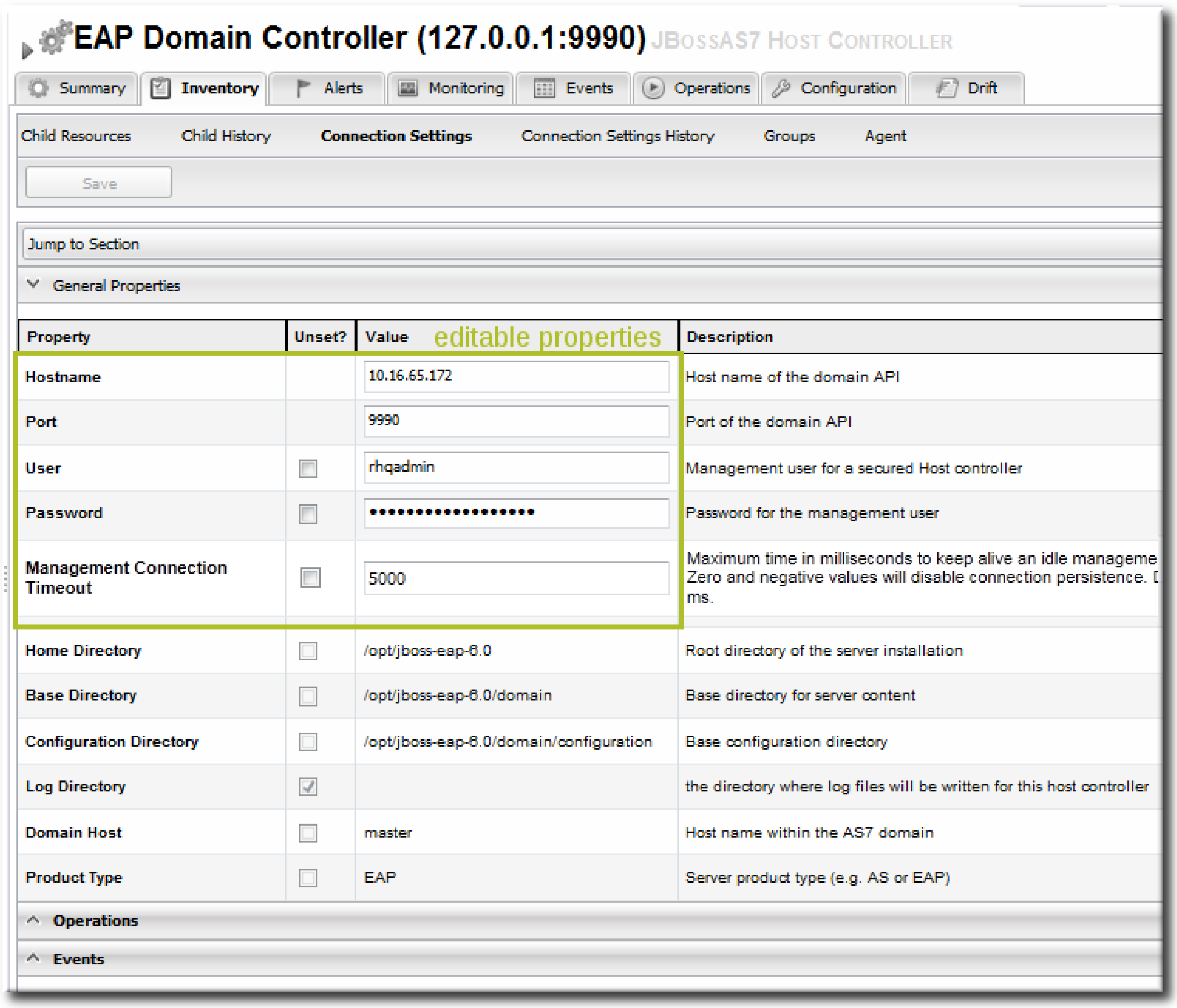
2. Select **`Servers - Top Level Imports`** in the **`Resources`** menu table on the left. Select the JBoss EAP 6 server.

3. In the inventory tree, select the top resource entry for the server.

4. Open the **`Inventory`** tab, and select the **`Connection Settings`** subtab.

5. The server connection properties are in the **`General Properties`** section. Only some of the

properties can be edited. Information that is derived from the JBoss EAP 6 installation itself, like the home directory, base directory, and server type (EAP or AS) is displayed, but is inactive.



Property	Unset?	Value	Description
Hostname		10.16.65.172	Host name of the domain API
Port		9990	Port of the domain API
User	☐	rhqadmin	Management user for a secured Host controller
Password	☐	••••••••••••••••••	Password for the management user
Management Connection Timeout	☐	5000	Maximum time in milliseconds to keep alive an idle manageme Zero and negative values will disable connection persistence. [ms.
Home Directory	☐	/opt/jboss-eap-6.0	Root directory of the server installation
Base Directory	☐	/opt/jboss-eap-6.0/domain	Base directory for server content
Configuration Directory	☐	/opt/jboss-eap-6.0/domain/configuration	Base configuration directory
Log Directory	✓		the directory where log files will be written for this host controller
Domain Host	☐	master	Host name within the AS7 domain
Product Type	☐	EAP	Server product type (e.g. AS or EAP)

- **`Hostname`** gives IP address to use to connect to the server. This is usually 127.0.0.1, but if the management interface configuration has been changed, then the IP address may be a public IP instead of the localhost.

- **`Port`** is the port of the management interface.

- **`Username`** and **`Password`** are the credentials of the JBoss EAP 6 user for the agent to use to log in. If this user was created using the *install RHQ user* operation, then the user is **`rhqadmin`**.

- *Domain Controllers Only.* With the standalone server, all of the configuration and the server instance definition are in the same file, **`standalone.xml`** or any other configuration file passed to the start script. For domains, the server configuration is defined in one file (for the domain controller), while the server instances are defined in a separate file (for the host controller). The **`Domain Configuration`** and **`Host Configuration`** fields give the names of the files within the **`domain/configuration/`** directory to reference for profile configuration and for managed server instances, respectively.

6. Click the **Save** button at the top of the page.

NOTE

Along with displaying the information in the **`Connection Settings`** area, the settings for a standalone server are also displayed (but not editable) in the **`Configuration`** tab, under the **`Server Environment`** area.

31.3.10.2. Viewing Installation Paths for EAP 6 Child Resources

Child resources for EAP 6 server such as managed servers, subsystem services, JVM definitions, and server groups are defined within the EAP 6 server's configuration files.

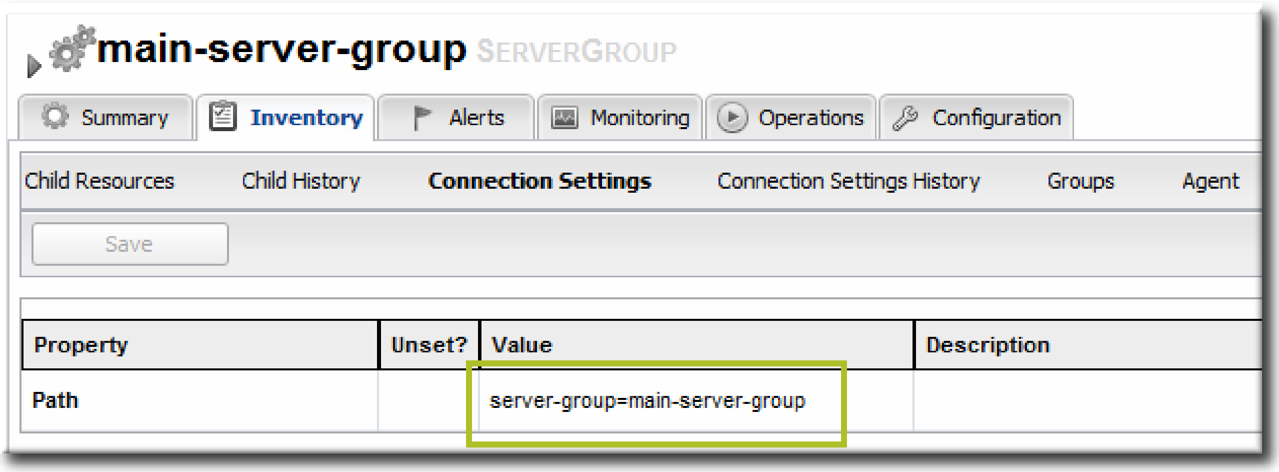
The "connection setting" for the child resource, then, is the entry in that server's configuration.

The connection setting is viewable in the child resource's **`Inventory > Connection Settings`** tab, but it cannot be edited because it is intrinsic to the resource itself. The connection setting for the child is the resource.

For example, the **`main-server-group`** definition is in the domain controller's **`domain.xml`** file:

```
    <server-groups>
        <server-group name="main-server-group" profile="full">
... 8< ...
```

In the JBoss ON GUI, the definition is given as the connection setting path, in the form *element=name*



Property	Unset?	Value	Description
Path		server-group=main-server-group	

Figure 31.7. Child Resource Connection Settings

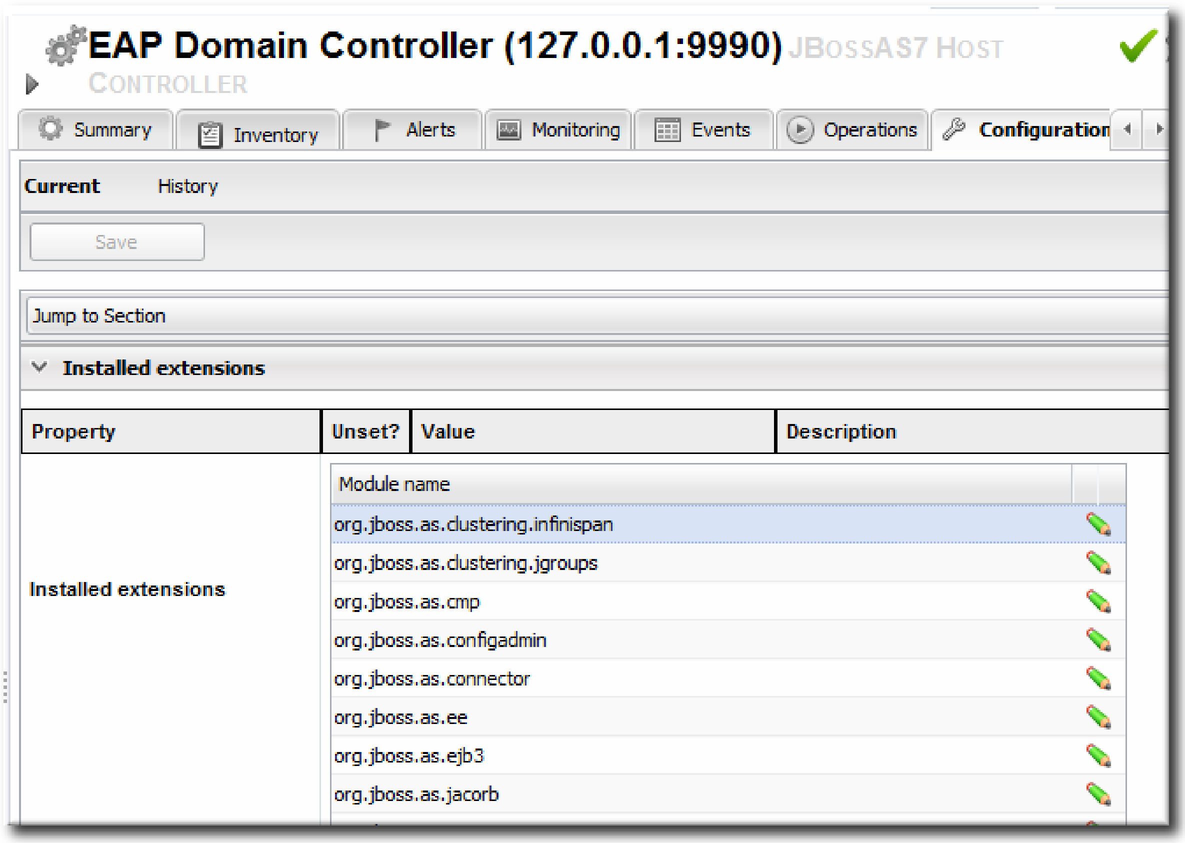
31.3.11. Viewing Installed Extensions

All of the subsystems available to JBoss EAP 6 profiles, both for standalone servers and for domains, are loaded as *extensions*. These extensions are modules, classes that are defined in the configuration file (**`domain.xml`** or **`standalone.xml`**).

```
    <extensions>
        <extension module="org.jboss.as.clustering.infinispan"/>
        <extension module="org.jboss.as.clustering.jgroups"/>
        <extension module="org.jboss.as.cmp"/>
        <extension module="org.jboss.as.configadmin"/>
        <extension module="org.jboss.as.connector"/>
        <extension module="org.jboss.as.ee"/>
        <extension module="org.jboss.as.ejb3"/>
... 8< ...
```

These extensions are not configurable through JBoss ON, but the current extension configuration can be viewed in JBoss ON as part of the standalone server or domain controller configuration.

1. Click the **Inventory** tab in the top menu.
2. Select **Servers - Top Level Imports** in the **Resources** menu table on the left. Select the JBoss EAP 6 server.
3. In the inventory tree, select the top resource entry for the server.
4. Open the **Configuration** tab.
5. Expand the **Installed extensions** section.



Property	Unset?	Value	Description
Installed extensions		Module name org.jboss.as.clustering.infinispan org.jboss.as.clustering.jgroups org.jboss.as.cmp org.jboss.as.configadmin org.jboss.as.connector org.jboss.as.ee org.jboss.as.ejb3 org.jboss.as.jacorb	

NOTE

An edit icon appears next to module names in the **Extensions** area of the EAP 6 server configuration. These properties are read-only and cannot, in fact, be edited even though an edit box pops up.

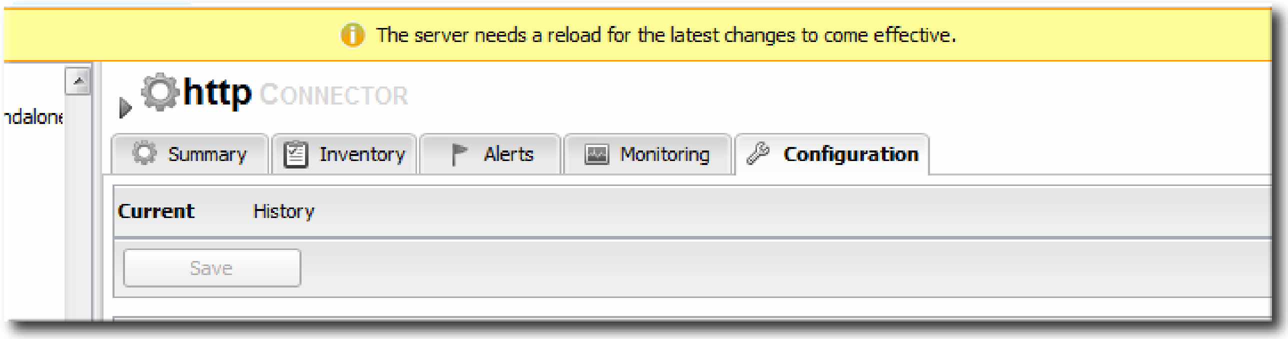
31.3.12. Reloading the Server Configuration

Some configuration changes require that the EAP 6 server be restarted before they take effect. The server is marked internally as being in a *requires-reload* state. JBoss ON does not force a reload, but it does inform administrators if changes have been made that require a reload before they can be applied.

NOTE

Because JBoss ON does not automatically reload configuration every time a configuration change is made, administrators can make multiple changes and then schedule a time for them to be applied by scheduling the reload operation during a maintenance window (through JBoss ON) or by setting up a cron job on the local system.

Clicking *any* **`Configuration > Current`** tab, for any resource within the server tree, brings up a message box that the server must be reloaded.



The server needs a reload for the latest changes to come effective.

Figure 31.8. Reload Configuration Message

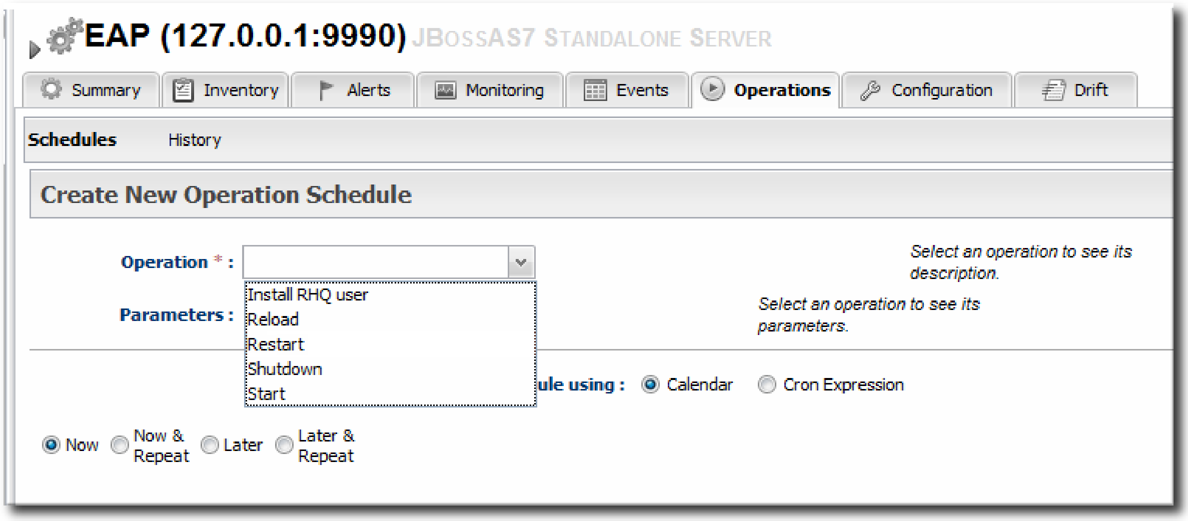
NOTE

Changes that require the configuration to be reloaded typically involve changing the way that connections are handled, such as resetting port numbers or changing connection protocols for an interface.

NOTE

If the agent goes offline and a user views a **`Configuration`** tab in the UI, the message box is displayed, even if the server has been reloaded. This is because the agent was not able to communicate to the server that the reload state has changed, so the server displays outdated information.

1. Click the **`Inventory`** tab in the top menu.
2. Select **`Servers - Top Level Imports`** in the **`Resources`** menu table on the left. Select the JBoss EAP 6 server.
3. In the inventory tree, select the top resource entry for the server.
4. Open the **`Operations`** tab.
5. Click the **New** button at the bottom of the page.
6. Select the **`Reload`**) option from the drop-down menu.



7. Click the **Schedule** button.

31.3.13. Controlling Configuration Drift

Configuration drift is the accumulation of changes, over time, to the desired or administrator-defined configuration. These changes can be trivial, accidental, or incremental, but it moves the resource outside the intended settings.

Configuration drift and JBoss ON's drift monitoring are covered in detail in the drift chapters of "Managing Resource Configuration."

For EAP 6, plan a drift strategy that covers all of the critical configuration and provides a path to remediation, possibly without requiring administrator intervention, to help preserve production systems:

1. Set drift definitions that track the critical configuration directories, such as **`domain/configuration/`** and**`standalone/configuration/`**, but that exclude directories which will have constantly changing data, such as logging, library, and data directories. Even within the configuration directories, create exclude rules for the **`host_xml_history/`**, **`domain_xml_history/`**, and **`standalone_xml_history/`** directories, since those are not proper configuration files and should not be tracked.

2. Once the desired configuration is in place, *pin* that configuration to the drift definition. This sets the desired configuration as the baseline. All changes will be compared against that baseline.

3. Create an archive of the blessed configuration.

4. Create a bundle definition that can be automatically deployed to reset the EAP 6 configuration and remediate drift.

   When creating the he destination should be the platform of the EAP 6 resource. The destination could be the standalone server or the domain controller, but using the platform allows you to deploy the bundle to an expendable directory, like **`/tmp/mybundles/holding`**, and then run a post-install task that copies the configuration files into the configuration directory.

   Deploying a bundle generally removes whatever existing files are in the target directory and replaces them with the bundle. There are ways to control that behavior, but, generally, it is safest to have the contents of the bundle match exactly what the final deployment will be.

Since it may not be feasible to have the entire configuration directory in the bundle, deploying to a separate location on the filesystem preserves the configuration directory, and only the important configuration files are updated (when they are copied by the Ant task).

For more on bundles and remedying drift, see the bundles chapter in "Deploying Applications and Content" and the drift-bundle CLI example script in "Writing JBoss ON Command-Line Scripts."

5. Set up alerts for configuration drift that do two things:

   - Send a notification email to administrators.
   - Run a CLI script on the platform that automatically deploys the bundle.

   Chapter 25, *Defining Alerts* has information on how to configure alert notifications that launch a JBoss ON server-side script or that run an operation on another resource.

NOTE

Changing the EAP 6 resource configuration, changing JVM definition settings, adding server groups and managed servers, or changing configuration settings all edit the EAP 6 configuration files, **`domain.xml`** and **`standalone.xml`**. That will trigger a drift alert, if alerting is configured.

31.3.14. Tracking and Reverting Configuration Changes

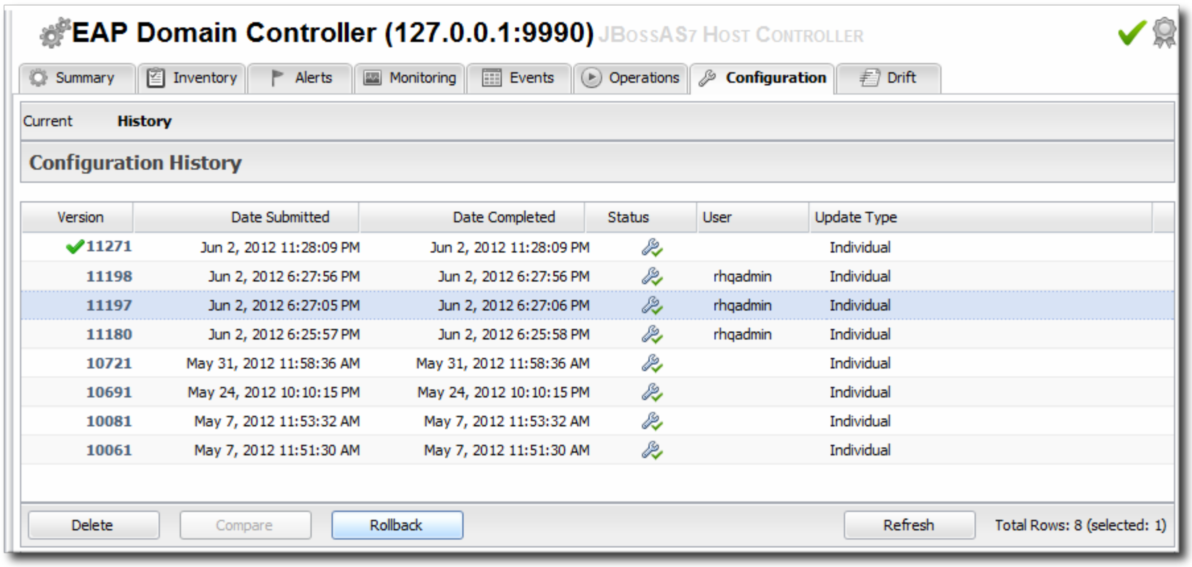
Every configuration change made through the JBoss ON UI or CLI is recorded in a change history.

1. Click the **`Inventory`** tab in the top menu.
2. Select **`Servers - Top Level Imports`** in the **`Resources`** menu table on the left. Select the JBoss EAP 6 server.
3. In the inventory tree, select the top resource entry for the server.
4. Open the **`Configuration`** tab, and select the **`History`** subtab.

   NOTE

   Change history pages are kept for resource configuration (the **`Configuration`** tab) and the connection settings (the **`Inventory > Connection Settings`** tab).

5. Clicking the change ID number opens the configuration settings that were in effect *for that version*.

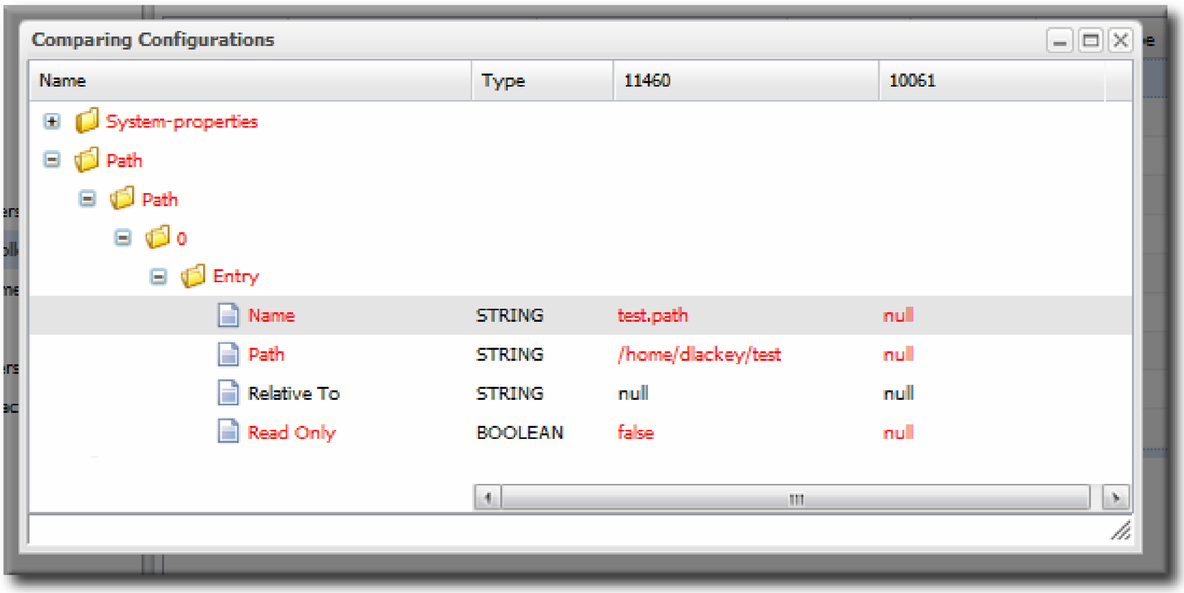


Configuration History

Version	Date Submitted	Date Completed	Status	User	Update Type
✓11271	Jun 2, 2012 11:28:09 PM	Jun 2, 2012 11:28:09 PM	🔧		Individual
11198	Jun 2, 2012 6:27:56 PM	Jun 2, 2012 6:27:56 PM	🔧	rhqadmin	Individual
11197	Jun 2, 2012 6:27:05 PM	Jun 2, 2012 6:27:06 PM	🔧	rhqadmin	Individual
11180	Jun 2, 2012 6:25:57 PM	Jun 2, 2012 6:25:58 PM	🔧	rhqadmin	Individual
10721	May 31, 2012 11:58:36 AM	May 31, 2012 11:58:36 AM	🔧		Individual
10691	May 24, 2012 10:10:15 PM	May 24, 2012 10:10:15 PM	🔧		Individual
10081	May 7, 2012 11:53:32 AM	May 7, 2012 11:53:32 AM	🔧		Individual
10061	May 7, 2012 11:51:30 AM	May 7, 2012 11:51:30 AM	🔧		Individual

Total Rows: 8 (selected: 1)

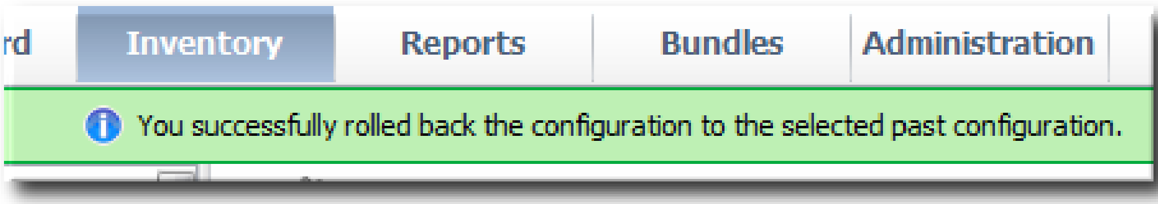
6. Changes can be compared to one another, in a standard diff format, by selecting them from the list and clicking the **Compare** button.



Comparing Configurations

Name	Type	11460	10061
System-properties			
Path			
Path			
0			
Entry			
Name	STRING	test.path	null
Path	STRING	/home/dlackey/test	null
Relative To	STRING	null	null
Read Only	BOOLEAN	false	null

7. The current, live version of the configuration can be reverted to *any* previous version by selecting the desired previous version in the list and clicking the **Rollback** button.

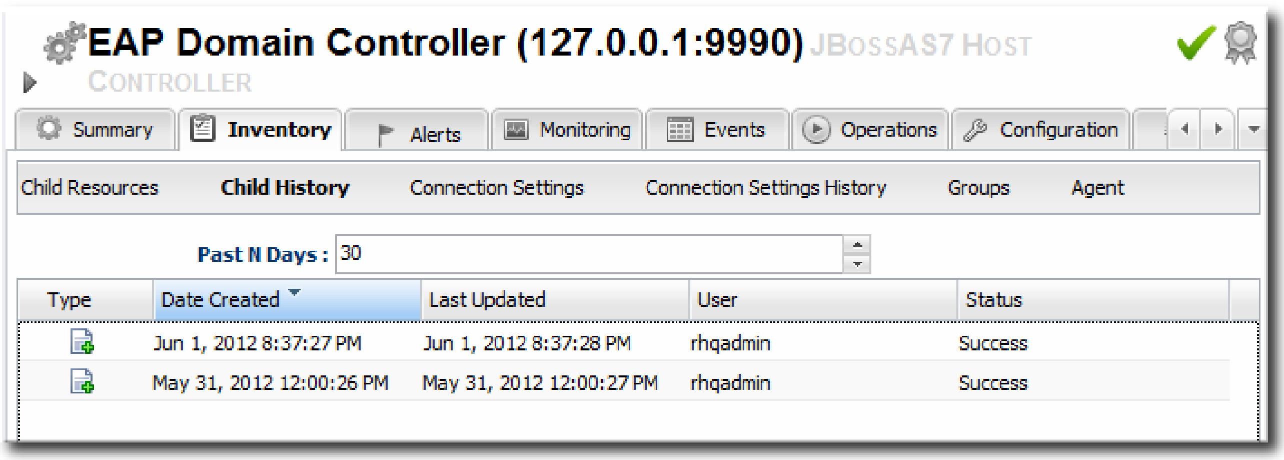


You successfully rolled back the configuration to the selected past configuration.

31.4. CREATING JBOSS EAP 6 RESOURCES

31.4.1. Tracking the Child History

Every **Inventory** tab has a **Child History** subtab. The history lists when child resources have been added or deleted, along with the date when the action occurred and which JBoss ON user initiated it.



Type	Date Created	Last Updated	User	Status
	Jun 1, 2012 8:37:27 PM	Jun 1, 2012 8:37:28 PM	rhqadmin	Success
	May 31, 2012 12:00:26 PM	May 31, 2012 12:00:27 PM	rhqadmin	Success

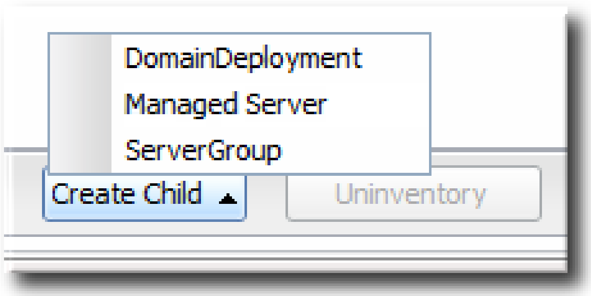
Figure 31.9. Child History

Adding or removing resources through JBoss ON helps maintain that trail of inventory changes.

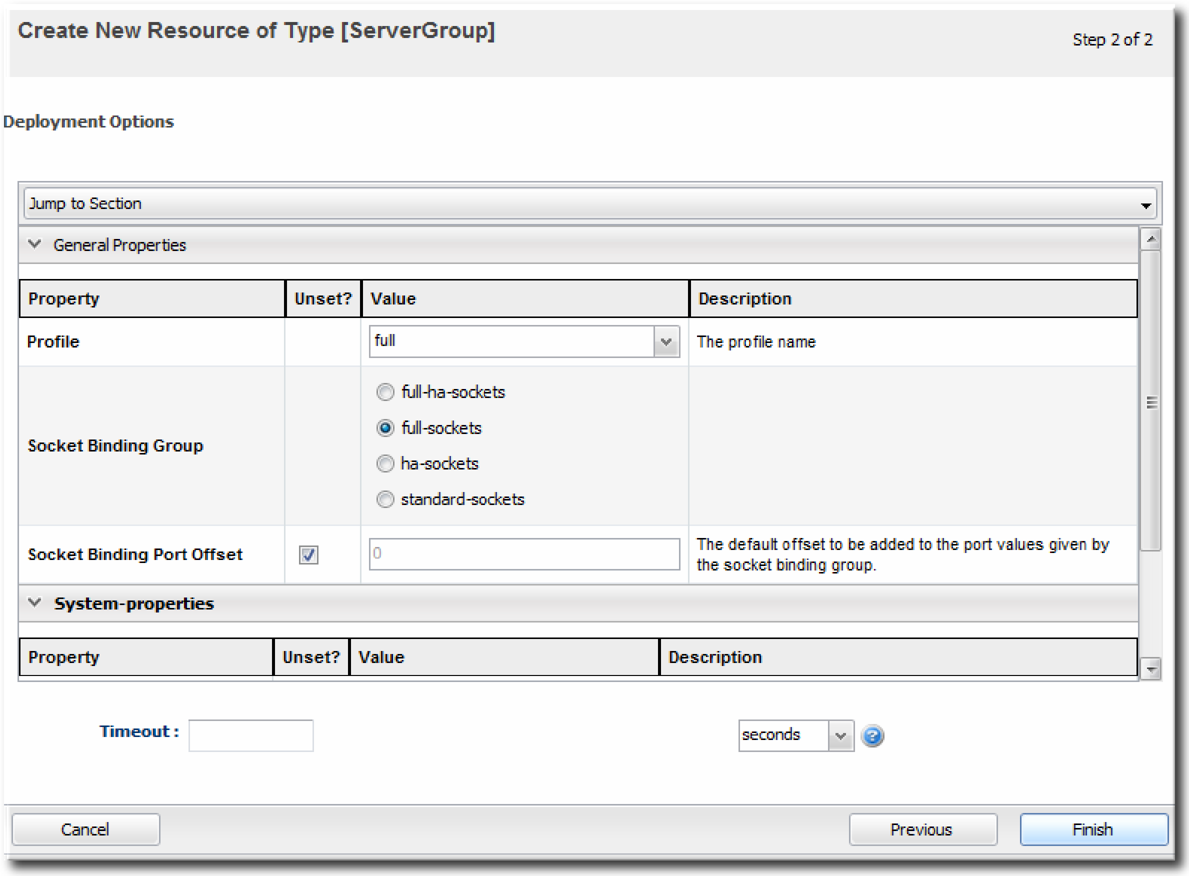
31.4.2. Creating Server Groups

Server groups are created as part of the domain setup to help manage configuration and content for managed servers.

1. Click the **`Inventory`** tab in the top menu.
2. Select **`Servers - Top Level Imports`** in the **`Resources`** menu table on the left.
3. Right-click the domain controller entry.



4. In the **`Create New`** menu, select the item for **Server Group**.
5. Enter the name for the new server group.
6. Enter the settings for the server group: the profile to use, the socket bindings group to use, and any system properties to set.


Create New Resource of Type [ServerGroup]

Step 2 of 2

Deployment Options

Jump to Section

General Properties

Property	Unset?	Value	Description
Profile		full	The profile name
Socket Binding Group		( ) full-ha-sockets (●) full-sockets ( ) ha-sockets ( ) standard-sockets	
Socket Binding Port Offset	✓	0	The default offset to be added to the port values given by the socket binding group.

System-properties

Property	Unset?	Value	Description

Timeout : seconds

7. Click **Finish**.

31.4.3. Creating Managed Servers

Managed servers are the functional servers in a domain; they perform the actual client interactions. The managed servers are children of the domain controller; the configuration and content for the managed servers is located in the domain controller configuration and is assigned by server group.

NOTE

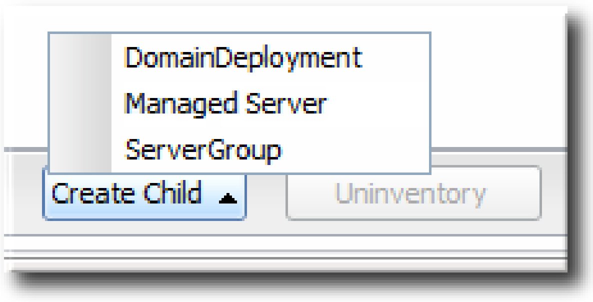
Because of the way that domain components can be arranged on physical systems (Section 31.1.2, "Separating Configuration and Real-Time Operations: Domains"), the managed server may not show up in the same configuration entry as the domain controller which is its parent. The new managed server is listed as a child of the host controller it is created on. The new managed server will show up in the platform (agent) inventory of that host controller.

NOTE

The creation process must be performed on the host controller after the job is submitted, and then the local agent must discover the new instances. Depending on where in the schedule the job runs and is completed, it can take several minutes for the new managed server to show up in the agent's discovery queue.

1. Click the **Inventory** tab in the top menu.
2. Select **Servers - Top Level Imports** in the **Resources** menu table on the left.

3. Right-click the domain controller entry.



4. In the `Create New` menu, select the item for **Managed Server**.

5. Enter the name for the new server.

6. Enter the settings for the server:

   - The EAP 6 domain host to create the server on.

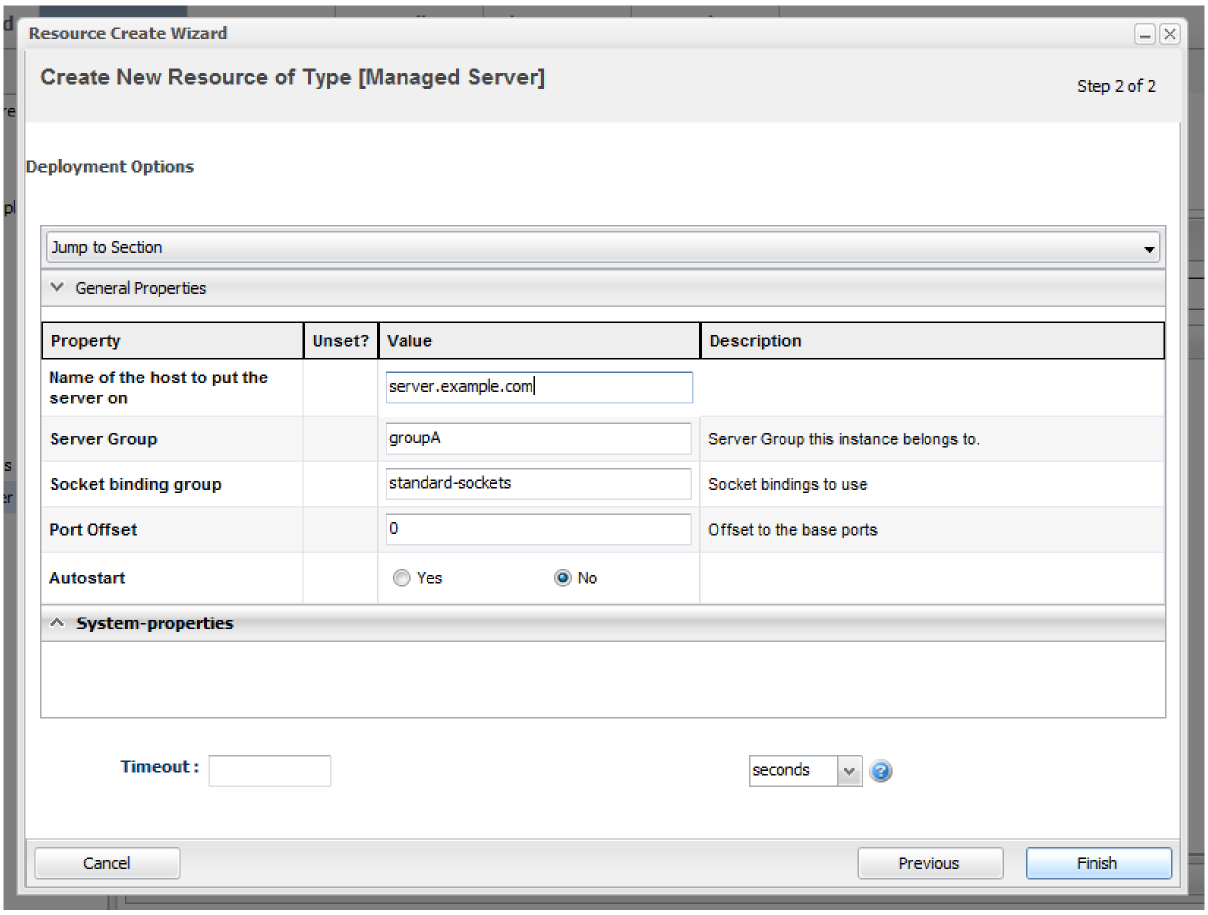
   - The server group to add the server to.

   - The socket binding group to use. This gives the base port numbers for the server instance to use.

   - The port offset. This is the number to add to the socket bindings settings to determine the actual port numbers for the server instance. If the socket binding port is 1000, and the offset is 150, then the port number used for that interface on the managed server is 1150.

   - Whether the server starts automatically when the EAP 6 host starts.

   - Any system properties for the server.



Property	Unset?	Value	Description
Name of the host to put the server on		server.example.com	
Server Group		groupA	Server Group this instance belongs to.
Socket binding group		standard-sockets	Socket bindings to use
Port Offset		0	Offset to the base ports
Autostart		Yes [✓ No]	

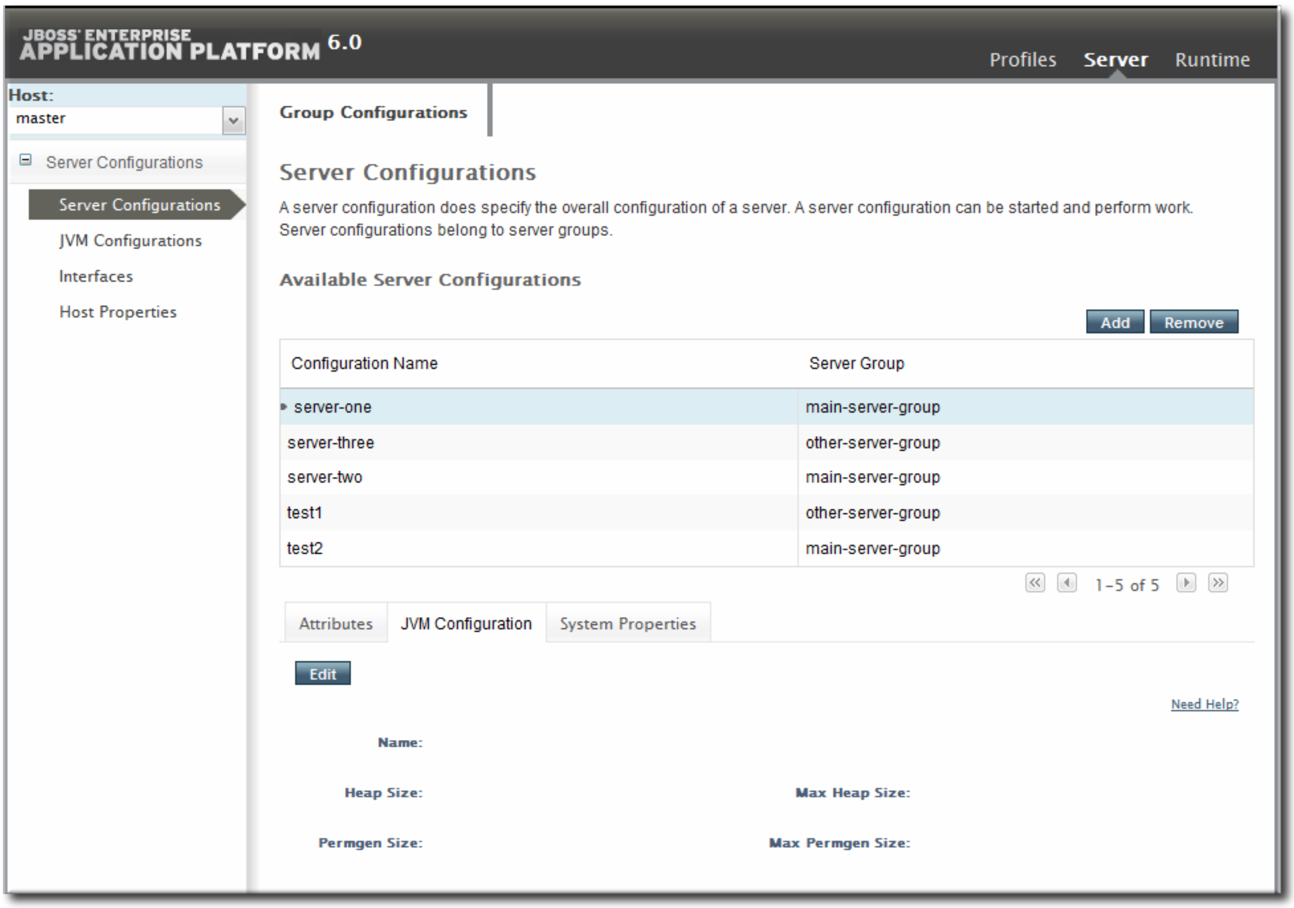
7. Click **Finish**.

31.4.4. Changing JVM Definitions

Managing the JVM settings for instances can help improve performance and better allocate system resources. Because of the nature of the EAP 6 domain structure, JVM settings work in a kind of cascade: the host controller JVM settings apply to the server group and all servers; any server group JVM settings override the host controller settings and apply to all their managed servers. Then, a JVM definition can be created for a managed resource, and that definition trumps any higher-level JVM settings.

31.4.4.1. JVM Definitions as Resources

In the EAP 6 management console, the JVM settings are treated as configuration options for server entries. Each server has a configuration tab for JVM configuration, and then there is a separate **JVM Configurations** under the **Server** page for the host controller.



Server Configurations

A server configuration does specify the overall configuration of a server. A server configuration can be started and perform work. Server configurations belong to server groups.

Available Server Configurations

Configuration Name	Server Group
server-one	main-server-group
server-three	other-server-group
server-two	main-server-group
test1	other-server-group
test2	main-server-group

Figure 31.10. JVM Definitions in the EAP 6 Console

As with the managed server configuration, the server group configuration has a tab for JVM settings.

In JBoss ON , JVMs have historically been separate resource types, with their own configuration and monitoring. For EAP 6, JVM definitions are also treated as separate resource types, children of the server or server group they apply to, even though in a structure sense the JVM *definition* is a collection of configuration properties; the JVM itself is the parent resource.

31.4.4.2. Creating a JVM Definition

A default JVM definition is already defined for the host controller, and that JVM definition is applied to every server group and, subsequently, every managed server associated with that group.

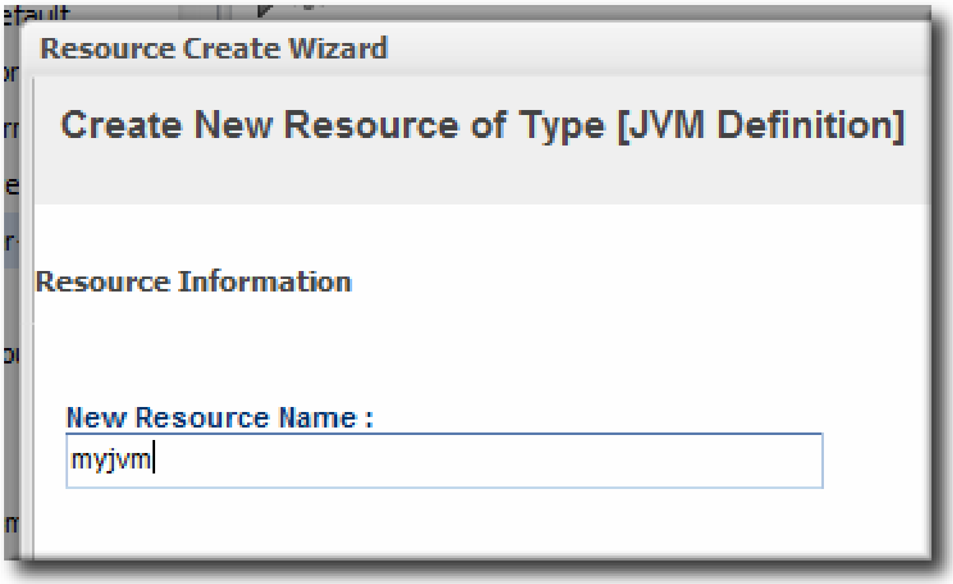
New JVM definitions can be created for server groups or managed servers to override those default settings. The managed server and server group definitions are based on the host controller JVM definition. Additional JVMs can be added to the host controller to provide a different base to use for new server group and server JVMs.

IMPORTANT

A host controller can have multiple JVM definitions; the default one is used for all managed servers and server groups, while any additional JVM definitions are available as the base definition for custom JVM definitions at the managed server and server group levels.

However, a managed server and a server group can have only one JVM definition. The UI will allow multiple JVM definitions to be created for a managed server and a server group, but the final creation step fails if a JVM definition already exists.

1. Click the **`Inventory`** tab in the top menu.
2. Select **`Servers - Top Level Imports`** in the **`Resources`** menu table on the left.
3. Right-click the entry to which to add the JVM definition.
4. In the **`Create New`** menu, select the item for **JVM Definition**.
5. Enter the name for the definition.

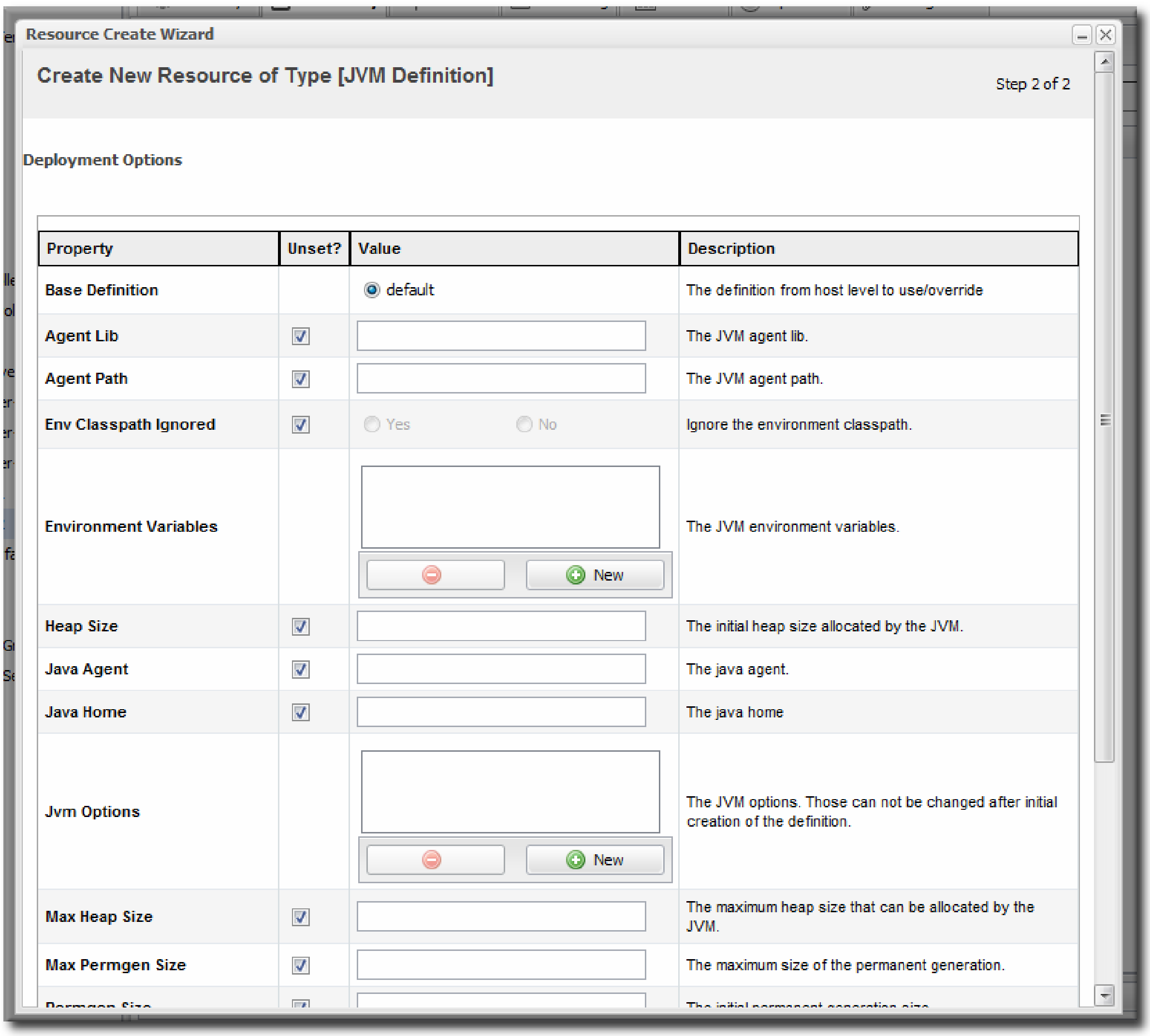


myjvm

NOTE

The name of JVM definition can be anything for a host controller. For a managed server or server group, the name of the JVM definition must be the same as the host controller JVM definition which is the base of the definition.

6. Enter the settings for the JVM. Any settings which are left blank use the values defined in the parent JVM definition.



Resource Create Wizard

Create New Resource of Type [JVM Definition]

Step 2 of 2

Deployment Options

Property	Unset?	Value	Description
Base Definition		● default	The definition from host level to use/override
Agent Lib	✓		The JVM agent lib.
Agent Path	✓		The JVM agent path.
Env Classpath Ignored	✓	○ Yes ○ No	Ignore the environment classpath.
Environment Variables		New	The JVM environment variables.
Heap Size	✓		The initial heap size allocated by the JVM.
Java Agent	✓		The java agent.
Java Home	✓		The java home
Jvm Options		New	The JVM options. Those can not be changed after initial creation of the definition.
Max Heap Size	✓		The maximum heap size that can be allocated by the JVM.
Max Permgen Size	✓		The maximum size of the permanent generation.
Permgen Size	✓		The initial permanent generation size

Most of the configuration for the JVM relates to memory and resource usage, along with options to pass environment variables or other settings to the JVM. The values of these settings can have a positive impact on both the resource performance and the overall system performance.

The configuration settings are listed in Table 31.1, “JVM Definition Properties”.

7. Click **`Finish`**.

Table 31.1. JVM Definition Properties

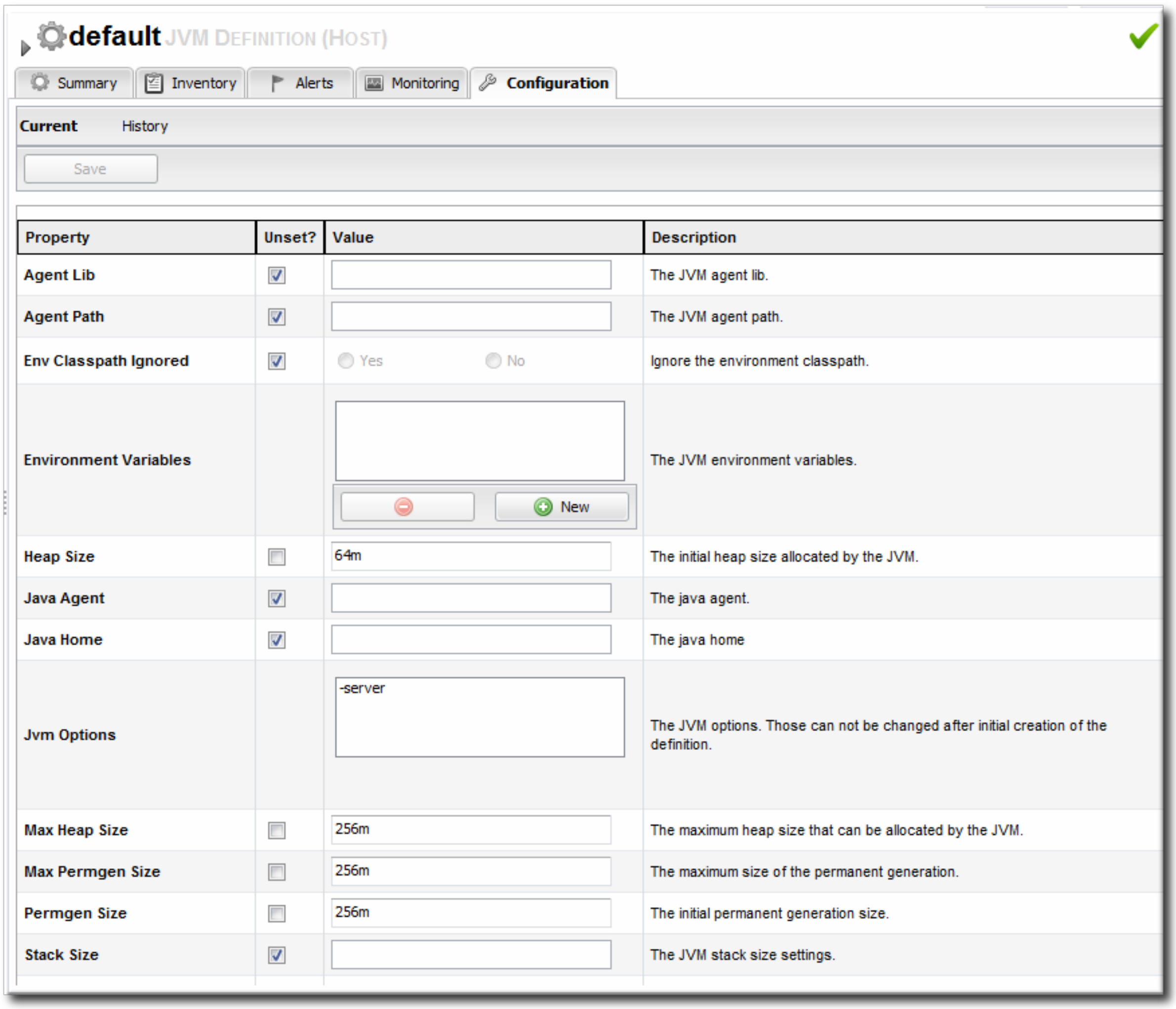
Property	Description
JVM Options	Sets any other Java option. Many of these are documented with the Java provider, such as Sun's documentation at http://www.oracle.com/technetwork/java/javase/tech/vmoptions-jsp-140102.html.
Environment variables	Sets environments variables to use with the JVM.
Heap Size	Sets the initial heap size for the JVM.

Property	Description
Max Heap Size	Sets the maximum allowed heap size for the JVM. Setting this too low can cause out of memory errors.
Permgen Size	Sets the initial size of the permanent generation.
Max Permgen Size	Sets the maximum size of the permanent generation.
Type	The type of JVM. This can be Sun or IBM, the two supported Java types for JBoss ON.

31.4.4.3. Editing a JVM Definition

The JVM definition is edited by editing the configuration properties for the JVM definition resource.

1. Click the **`Inventory`** tab in the top menu.
2. Select **`Servers - Top Level Imports`** in the **`Resources`** menu table on the left.
3. Select the EAP server and navigate to the appropriate JVM definition entry.
4. Open the **`Configuration`** tab.



Property	Unset?	Value	Description
Agent Lib	✓		The JVM agent lib.
Agent Path	✓		The JVM agent path.
Env Classpath Ignored	✓	Yes No	Ignore the environment classpath.
Environment Variables			The JVM environment variables.
Heap Size		64m	The initial heap size allocated by the JVM.
Java Agent	✓		The java agent.
Java Home	✓		The java home
Jvm Options		-server	The JVM options. Those can not be changed after initial creation of the definition.
Max Heap Size		256m	The maximum heap size that can be allocated by the JVM.
Max Permgen Size		256m	The maximum size of the permanent generation.
Permgen Size		256m	The initial permanent generation size.
Stack Size	✓		The JVM stack size settings.

5. Change any of the JVM settings. These are listed in Table 31.1, “JVM Definition Properties”.

6. Click the **Save** button at the top of the page.

31.4.5. A Short List of Parent-Child Resources

There are hundreds of resource types defined in the JBoss EAP 6 agent plug-in, covering all major servers, subsystem types, and services. The Resource Reference covers all of the different resource types and lists child types for each.

Table 31.2, “A Short List of Parent-Child Resources” lists some common JBoss EAP 6 resource types and what children resource can be created for them through the JBoss ON UI.

NOTE

Creating a child resource must be performed on the parent resource after the job is submitted. Then the local agent must discover the new instances.

Depending on where in the schedule the job runs and is completed, it can take several minutes for the new managed server to show up in the agent's discovery queue.

Table 31.2. A Short List of Parent-Child Resources

Resource	Child Resource Types
Standalone server	• Deployment
Domain Controller	• Domain deployment • Managed servers • Server groups
Host	• JVM definition
Server Groups	• Deployment • JVM definition
Datasources (under Profiles)	• Datasource • XA Datasource
Infinispan > Hibernate	• Cache
Logging	• Async handler • Console handler • Custom handler • File handler • Logger • Periodic Rotating File handler • Size Rotating File handler
Web	• Connector • Vhost

31.5. DEPLOYING WEB APPLICATIONS

Web applications are a unique type of resource. They are added as child resources of servers, so they have a place within the inventory with their own entries, possibly even their own child resources, and independent configuration for monitoring, alerting, and operations. However, web applications are *content-backed resources*. Ultimately, they are software packages, not true servers or services.

Managing a content-backed resource requires a path to make the initial deployment (creating the deployment as a child resource) and also a path for patches and updates (content updates).

IMPORTANT

Deploying content-backed resources can have a significant impact on disk space requirements.

JBoss ON stores all versions of content. This is part of versioning control, allowing changes to content-backed resources to be reverted and managed and for different versions to be deployed at different times.

Therefore, the system which hosts the backend database (Oracle or PostgreSQL) must have enough disk space to store all versions of all bundles. Additionally, the database itself must have adequate tablespace for the content.

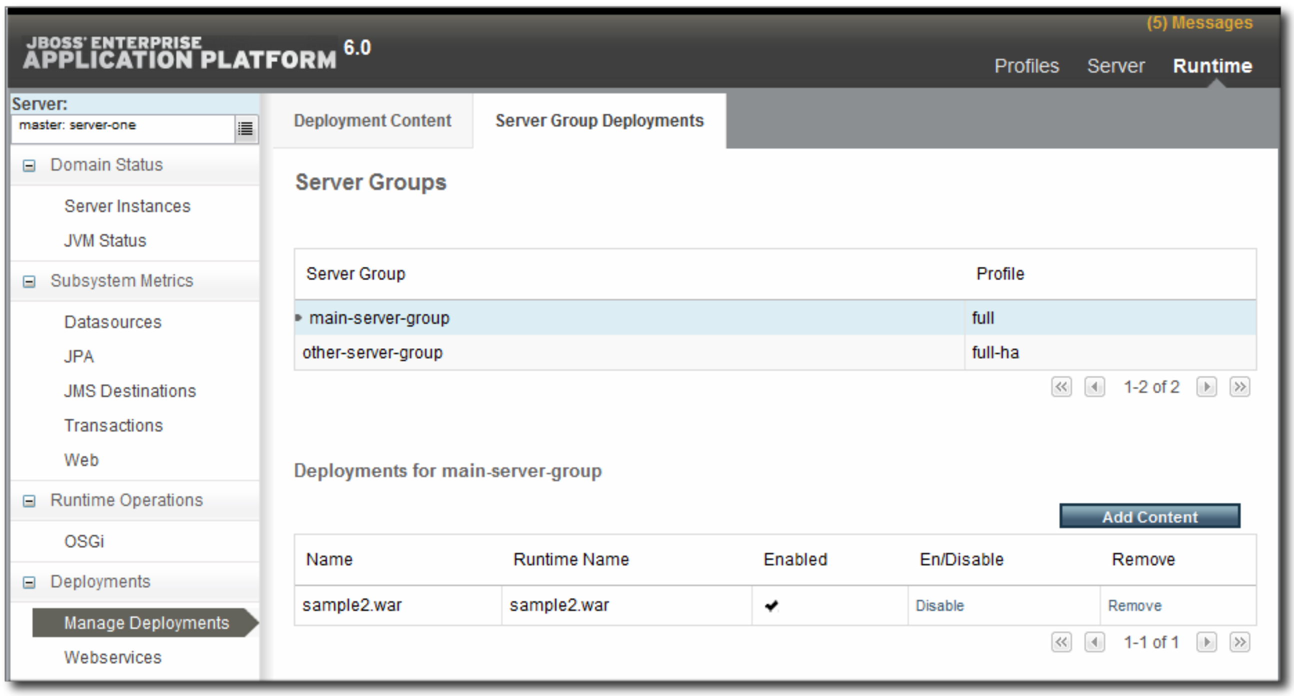
When calculating the required amount of space, estimate the size of every artifact, and then the number of versions for each artifact. At a minimum, **have twice that amount of space available**; both PostgreSQL and Oracle require twice the database size to perform cleanup operations like vacuum, compression, and backup and recovery.

31.5.1. Runtime Information and Deployment Resources

31.5.1.1. Views of Deployments

Both EAP 6 and JBoss ON offer clear workflows for adding web applications to a central content repository and then associating that content with a server group. But the perspectives they offer are very different, so deciding which console to use depends on what needs to be accomplished at that particular moment.

The EAP 6 console is focused on immediately associating content with a server group. In the EAP 6 console, deployments are treated similarly to shared configuration, like a JVM definition. A content package is deployed to a domain controller or a server group and is then associated with a managed server.



Server Groups

Server Group	Profile
main-server-group	full
other-server-group	full-ha

1-2 of 2

Deployments for main-server-group

Name	Runtime Name	Enabled	En/Disable	Remove
sample2.war	sample2.war	✓	Disable	Remove

1-1 of 1

Figure 31.11. Deployments in the Runtime Page

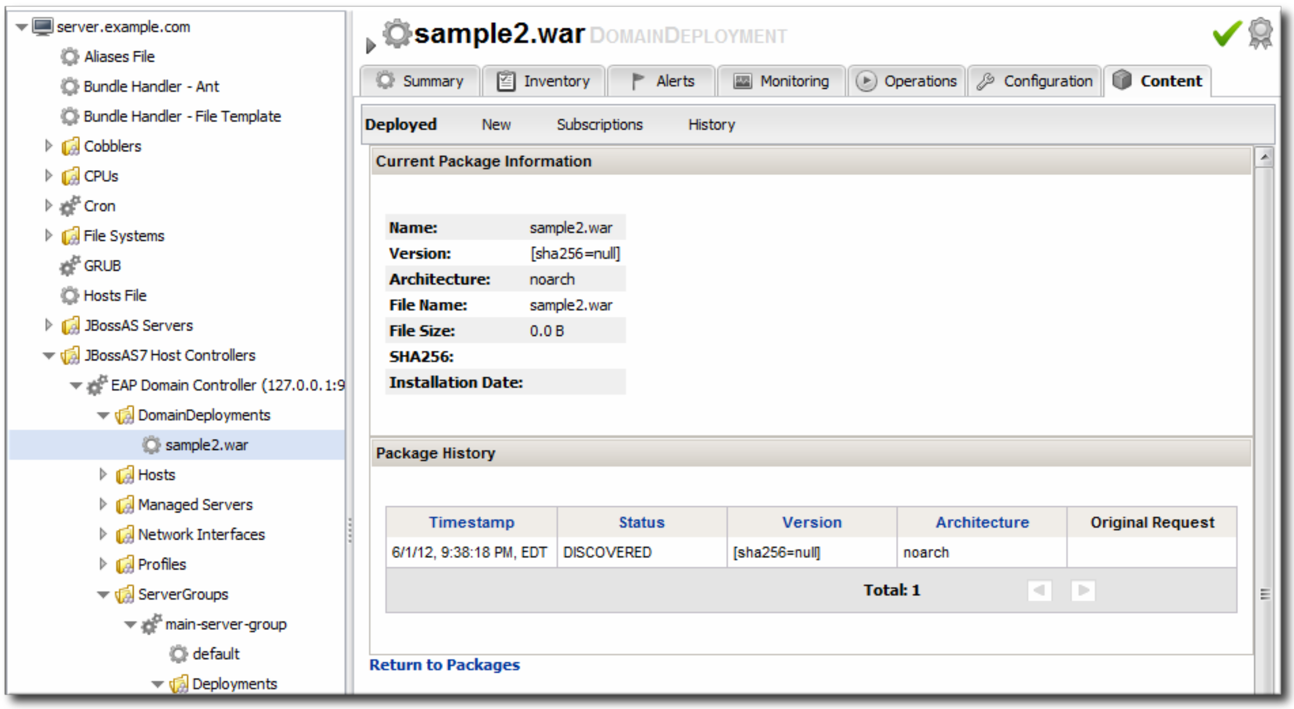
JBoss ON is focused on lifecycle management for web application, an historical and changing perspective. There are a few fundamental differences in how JBoss ON defines web applications:

- A web application is treated as a separate entity, in and of itself. It has its own place in the inventory; its association with domains and server groups is reflected as it is a child of those resources.

  JBoss ON also records package details, like its file size and an identifying SHA256 hash.

- A web application has a history. Updates to packages are recorded in a changelog which makes it possible to track how the application has been maintained.

- Web applications can be monitored. Response time metrics specifically track the performance of web applications, apart from the performance or monitoring of the underlying server.



Current Package Information

Name:	sample2.war
Version:	[sha256=null]
Architecture:	noarch
File Name:	sample2.war
File Size:	0.0 B
SHA256:	
Installation Date:	

Package History

Timestamp	Status	Version	Architecture	Original Request
6/1/12, 9:38:18 PM, EDT	DISCOVERED	[sha256=null]	noarch	

Total: 1

Figure 31.12. Deployment Resource Details in JBoss ON

The reason for this emphasis on web applications as a resource is that the core purpose is not the simple task of deploying content to a server. The EAP 6 console does that very simply and cleanly. The purpose of JBoss ON management is administration of that content of a web application:

- Creating content repositories to store patches and updates
- Tracking multiple versions of content
- Reverting software packages to a previous version (particularly for standalone servers)
- Using the same content repository for multiple EAP instances, including multiple domains and standalone server
- Tracking (and auditing) changes to content

31.5.1.2. Deployment Paths for Standalone Servers and Domains

There is a radically different structure between standalone servers and domains, and that impacts the content flow for deploying web applications.

Web application deployments in standalone servers are similar to EAP 5, in that there is a local filesystem directory which is used for deployments. Locally, applications can be added to that deployment directory and hot-deployed. Using JBoss ON, there are two paths for deploying web applications:

- Create a deployment as a child resource
- Use a bundle to provision the application in the deployment directory

Using bundles is a versatile and practical management method, because the bundle system has capabilities to store multiple versions, rollback updates, skip versions for updates, or purge deployments, all with a single button.

Unfortunately, the bundles system only works with real filesystem locations. The distributed and modular structure for JBoss EAP 6 domains means that bundles are not an option.

For EAP 6 domains, the content is deployed as a child resource to the domain and then is propagated to server groups by administrators.

It lacks the flexibility and finesse of the bundles system, but using the ordinary content system still offers ways to control content versions and updates. Content-backed resources can be associated with content repositories in JBoss ON. These repositories can store all sorts of software packages, from different content sources. Packages from the repo can be installed on the resource selectively. Additionally, a resource can be associated with multiple repos. While not as fluid as updating or reverting changes with bundles, the content repositories in JBoss ON provide a pathway to apply individual patch files to a web application or to roll out full updates.

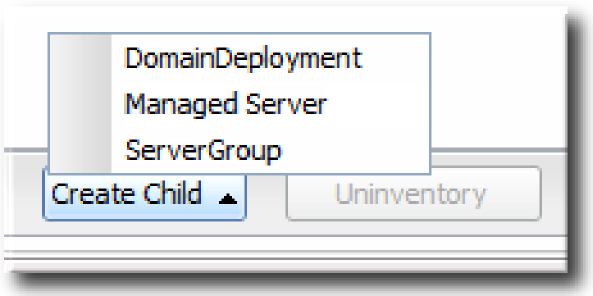
Content management in general is described in "Deploying Applications and Content" . This covers setting up content repositories, subscribing resources, and pushing updates.

31.5.2. Deploying Web Applications to a Domain

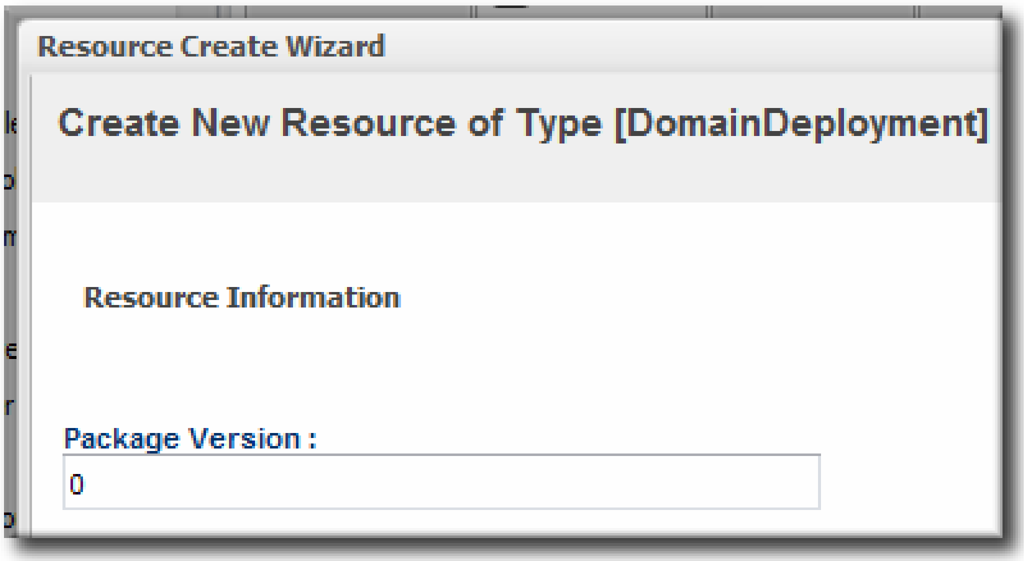
NOTE

As with creating other child resources, it can take several minutes for the new deployment to be added and visible in the JBoss ON inventory because the new resource has to be created on the local system and then discovered by the agent. If the discovery scan is running when the resource is created, then it may take until the next discovery scan to be detected, as long as 15 minutes.

1. Click the **Inventory** tab in the top menu.
2. Select **Servers - Top Level Imports** in the **Resources** menu table on the left. Select the JBoss EAP 6 domain controller.
3. Right-click the domain controller entry.



4. In the **Create New** menu, select the item for **DomainDeployment**.
5. Enter the version number.

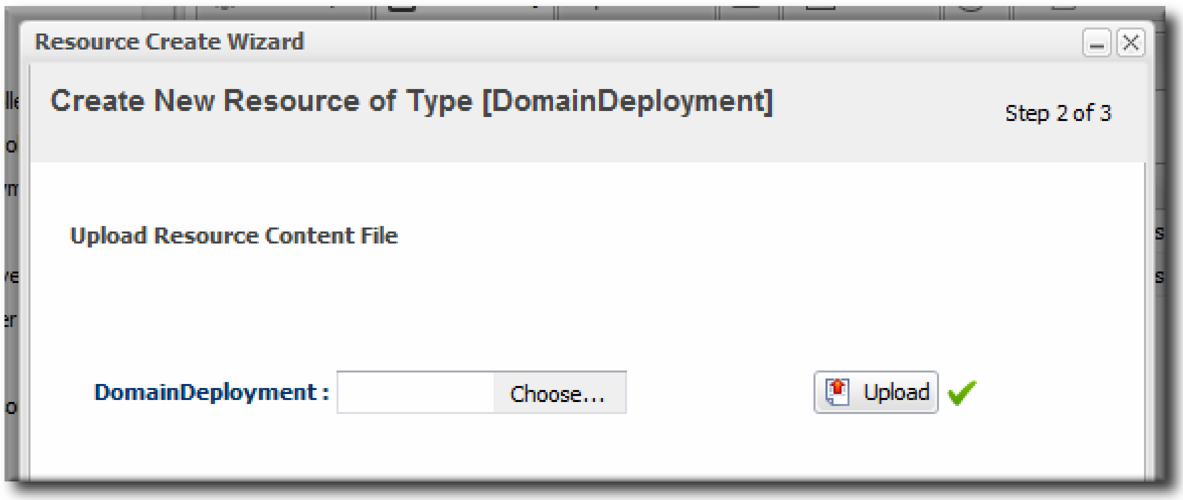


Create New Resource of Type [DomainDeployment]

Resource Information

Package Version : 0

6. Upload the EAR, WAR, or JAR file.

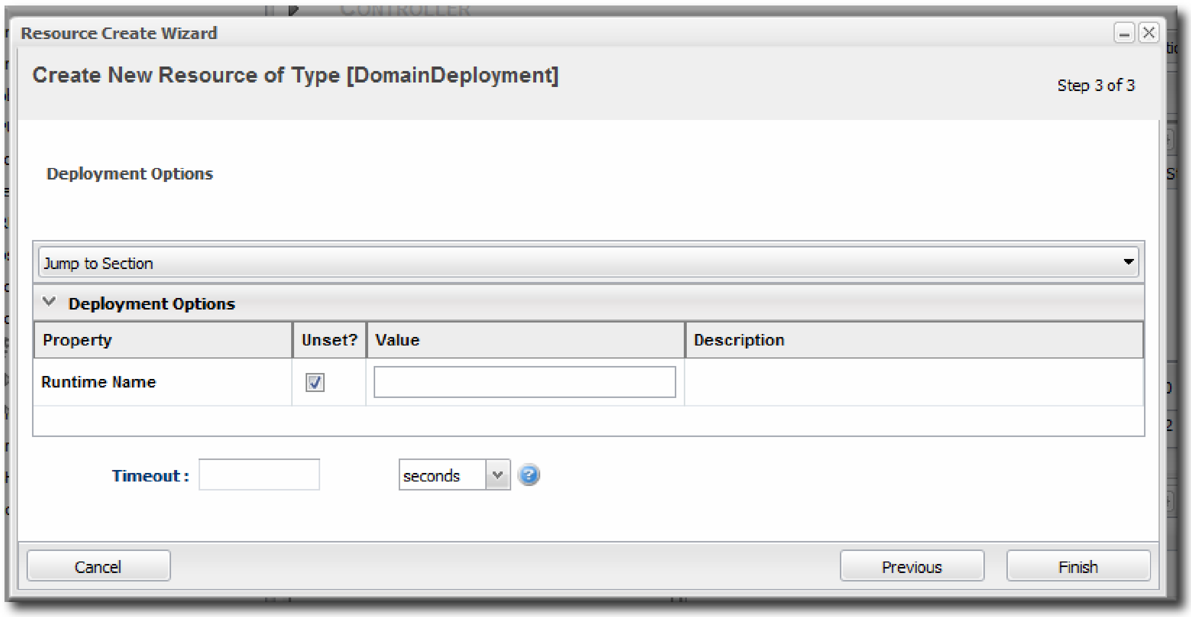


Create New Resource of Type [DomainDeployment] Step 2 of 3

Upload Resource Content File

DomainDeployment :

7. Optionally, enter a runtime name for the deployment.



Create New Resource of Type [DomainDeployment] Step 3 of 3

Deployment Options

Deployment Options

Property	Unset?	Value	Description
Runtime Name	✓		

Timeout : seconds

8. At the bottom of the wizard, set an optional timeout period. This is how long the JBoss ON server will wait during the deployment process before determining that the deployment has failed.

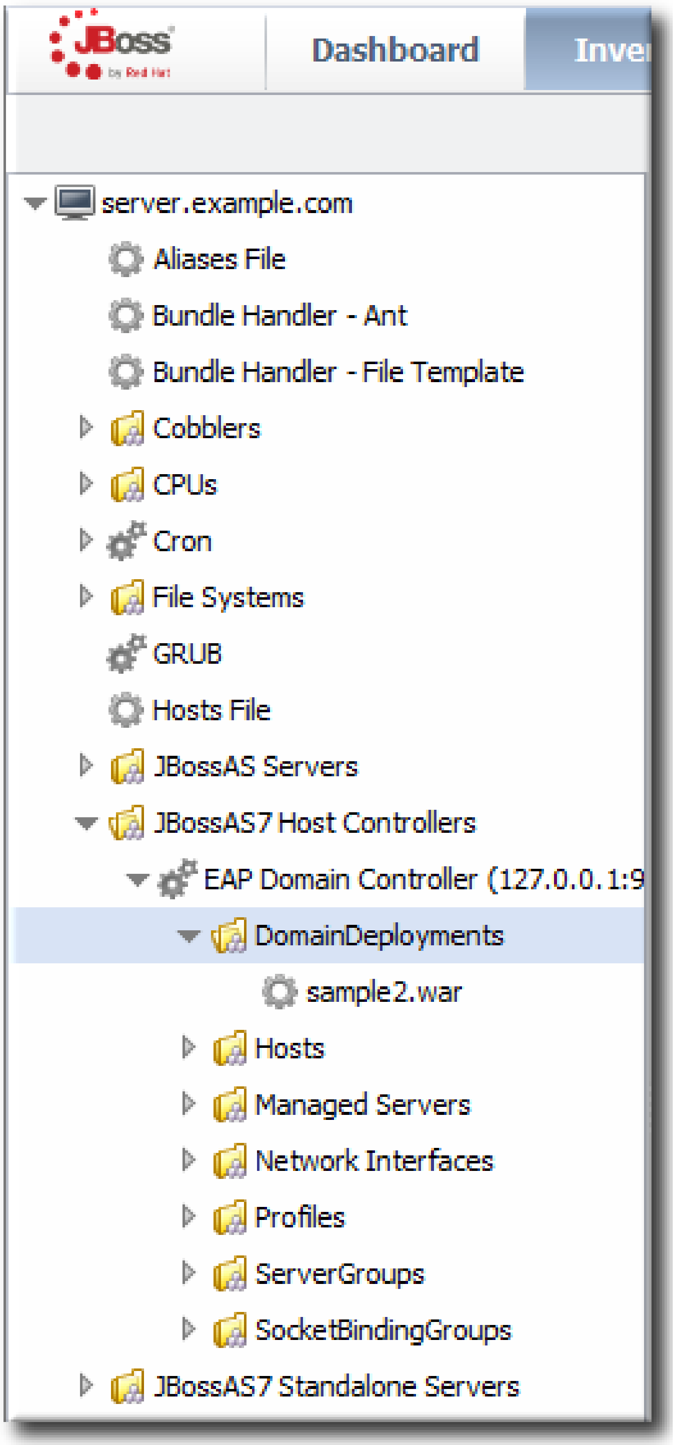
   NOTE

   The timeout period only applies to the server's reporting a result. If the operation continues running, it can still complete successfully, and the web application is deployed.

   Particularly for large application files, do not set a low timeout period, or the server will mark the deployment as having failed. If the deployment completes later, the web application must be imported into the inventory manually; it will not be discovered by the agent.

9. Click **`Finish`**.

Once the package is uploaded, a new resource entry is added to the **`DomainDeployments`** directory for the domain.



server.example.com
Aliases File
Bundle Handler - Ant
Bundle Handler - File Template
Cobblers
CPUs
Cron
File Systems
GRUB
Hosts File
JBossAS Servers
JBossAS7 Host Controllers
EAP Domain Controller (127.0.0.1:9
DomainDeployments
sample2.war
Hosts
Managed Servers
Network Interfaces
Profiles
ServerGroups
SocketBindingGroups
JBossAS7 Standalone Servers

Figure 31.13. Domain Deployments Directory

31.5.3. Assigning Web Applications to a Server Group

Assigning web applications to different groups manually allows administrators to control the lifecycle for applications, as expanded in Section 31.5.4, "*Extended Example:* Assigning Web Applications and Managing Updates". The same content is used in different operating environments.

NOTE

Web applications can be deployed directly to a server group, much like the process to deploying it to a domain, in Section 31.5.2, "Deploying Web Applications to a Domain" .

The deployment path for a domain is to upload the content to the domain controller and then to distribute that content to a server group. Creating a deployment on a server group follows the same process; it deploys it to the domain and then sends it to the server group. It just provides a shortcut.

If the content already exists on the domain controller, then the content must be deployed through the domain controller operation described here.

Deploying the content to a domain first uses the domain as a kind of repository. From the domain, the same package can be deployed to multiple groups. Additionally, deploying to a domain allows administrators to deploy multiple packages in the same operation to a group, either simultaneously or in a specified order. This can help control content streams to servers.

IMPORTANT

Newly-deployed content for a server group may not show up in the JBoss ON inventory for as long as 24 hours, even if it was successfully created. By default, discovery scans for services are only made every 24 hours. To see the new content-backed resource, run the agent's discovery scan and manually discover it.

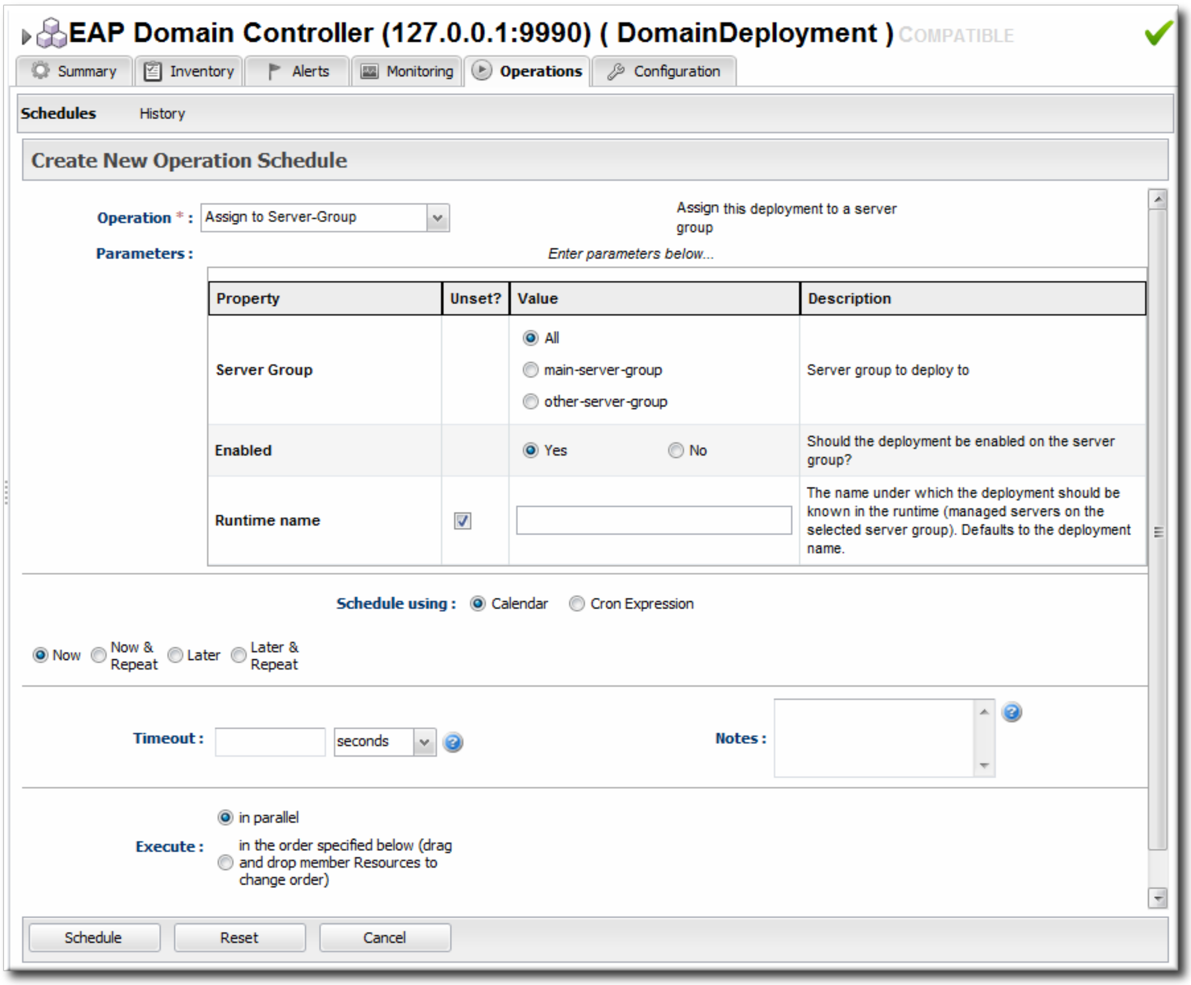
NOTE

The deployment is created in the domain as a child resource, then assigned to a server group as an operation, and finally updated through the domain's **`Content`** tab.

1. Click the **`Inventory`** tab in the top menu.
2. Select **`Servers - Top Level Imports`** in the **`Resources`** menu table on the left. Select the JBoss EAP 6 domain controller.
3. Expand the **`DomainDeployments`** folder.

   To deploy all web applications in the deployments folder, select the deployments folder itself to run the operation.

   To deploy a single web application, select the specific web application.
4. Open the **`Operations`** tab for the server.
5. Click the **New** button at the bottom of the page.
6. Select the **`Assign to Server-Group`** option from the drop-down menu.


EAP Domain Controller (127.0.0.1:9990) (DomainDeployment) COMPATIBLE

Summary | Inventory | Alerts | Monitoring | Operations | Configuration

Schedules | History

Create New Operation Schedule

Operation * : Assign to Server-Group — Assign this deployment to a server group

Parameters : Enter parameters below...

Property	Unset?	Value	Description
Server Group		[✓ All] main-server-group other-server-group	Server group to deploy to
Enabled		[✓ Yes] No	Should the deployment be enabled on the server group?
Runtime name	✓		The name under which the deployment should be known in the runtime (managed servers on the selected server group). Defaults to the deployment name.

Schedule using : [✓ Calendar] Cron Expression

[✓ Now] Now & Repeat Later Later & Repeat

Timeout : seconds

Notes :

Execute : [✓ in parallel] in the order specified below (drag and drop member Resources to change order)

Schedule Reset Cancel

7. Select which group (or all groups) to deploy the application to.

8. Enter the standard information for operations, like the schedule for when to run the operation and the timeout period.

NOTE

The timeout period sets how long the server waits before assuming that the operation has timed out and failed. This does not necessarily mean that the operation has failed or stopped running; it could complete successfully past the time out period.

9. If there are multiple web applications being deployed, then set the **`Execute`** radio button to the way set the order that the packages are deployed. All packages can be deployed at once, or they can be deployed in a specific order.

10. Click the **`Schedule`** button.

11. Optionally, run a discovery scan on the agent to discover the new content resource. By default, discovery scans for services are only made every 24 hours, so there could be a long delay in discovering new content.

    1. Open the agent entry in the UI.

    2. Open the **`Operations`** tab, and select the *execute prompt command*operation.

    3. Enter the **`discovery`** command as an operation rgument. This runs a discovery scan.

4. Click the **Schedule** button.

31.5.4. *Extended Example:* Assigning Web Applications and Managing Updates

The Setup

Tim the IT Guy wants to have a clear progression for web applications, from development through staging and production. The native structure in EAP 6 allows him to create different server groups and deploy content from his central domain controller to the appropriate server groups as it passes testing at each stage.

What Tim the IT Guy wants is to add a layer on top of that path that allows him to create an apply patches and updates. He has to be able to manage fixes and to audit when they were applied as part of his maintenance schedule.

The Plan

1. Tim first outlines what server groups he needs to maintain. For a simple environment, he just wants three groups: testing, staging, and production.

2. He creates two content repositories, one for patches and one for new versions of the web application.

3. He creates the domain deployment and then promotes the web application to the testing server group.

4. Tim configures response time monitoring for the web application. Once it meets the required performance parameters in the testing area, Tim promotes the deployment to the staging and then production server groups.

The Result

The package history for each deployment allows Tim to track when the web application was deployed, its version, and its content.

Tim can apply patches or full updates by deploying content to his repositories and installing it on every subscribed resource. These package changes, including new version numbers for the content, are included in the package history.

Because the patch and update repositories are configured in JBoss ON itself, not a specific JBoss EAP 6 domain, the packages there can be used with other domains and standalone servers, if need be.

31.5.5. Enabling and Disabling Web Applications

1. Click the **Inventory** tab in the top menu.

2. Select **Servers - Top Level Imports** in the **Resources** menu table on the left. Select the JBoss EAP 6 domain controller.

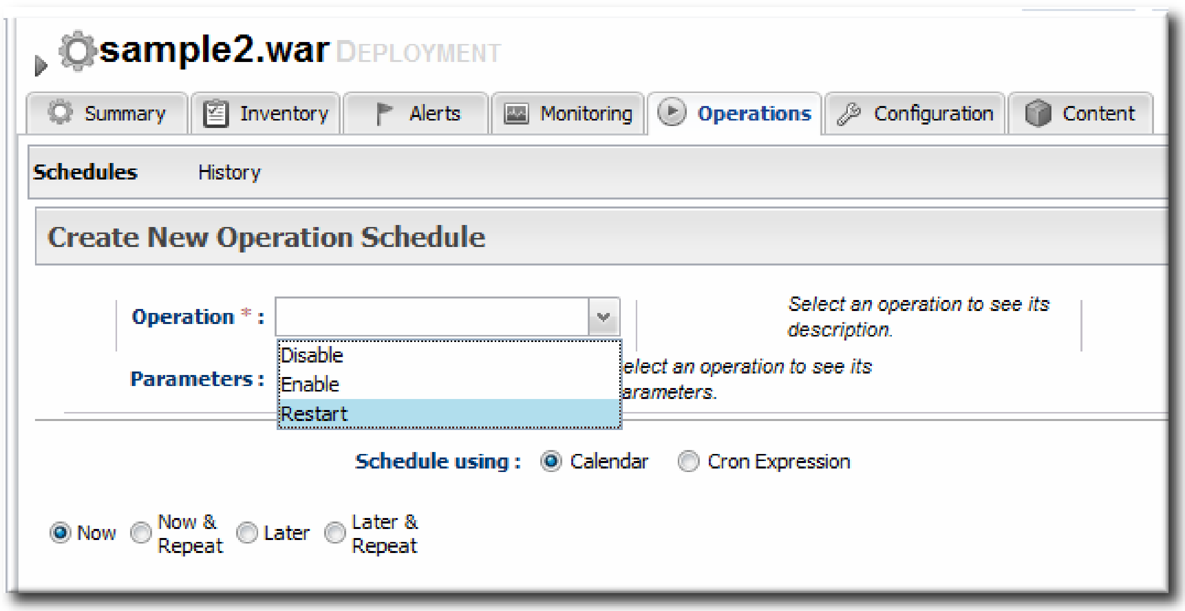
3. In the inventory tree, expand the **ServerGroups** folder, and select the server group from the list.

4. Expand the **Deployments** folder, and select the web application to enable or disable.

5. Open the **Operations** tab for the web app.

6. Click the **New** button at the bottom of the page.

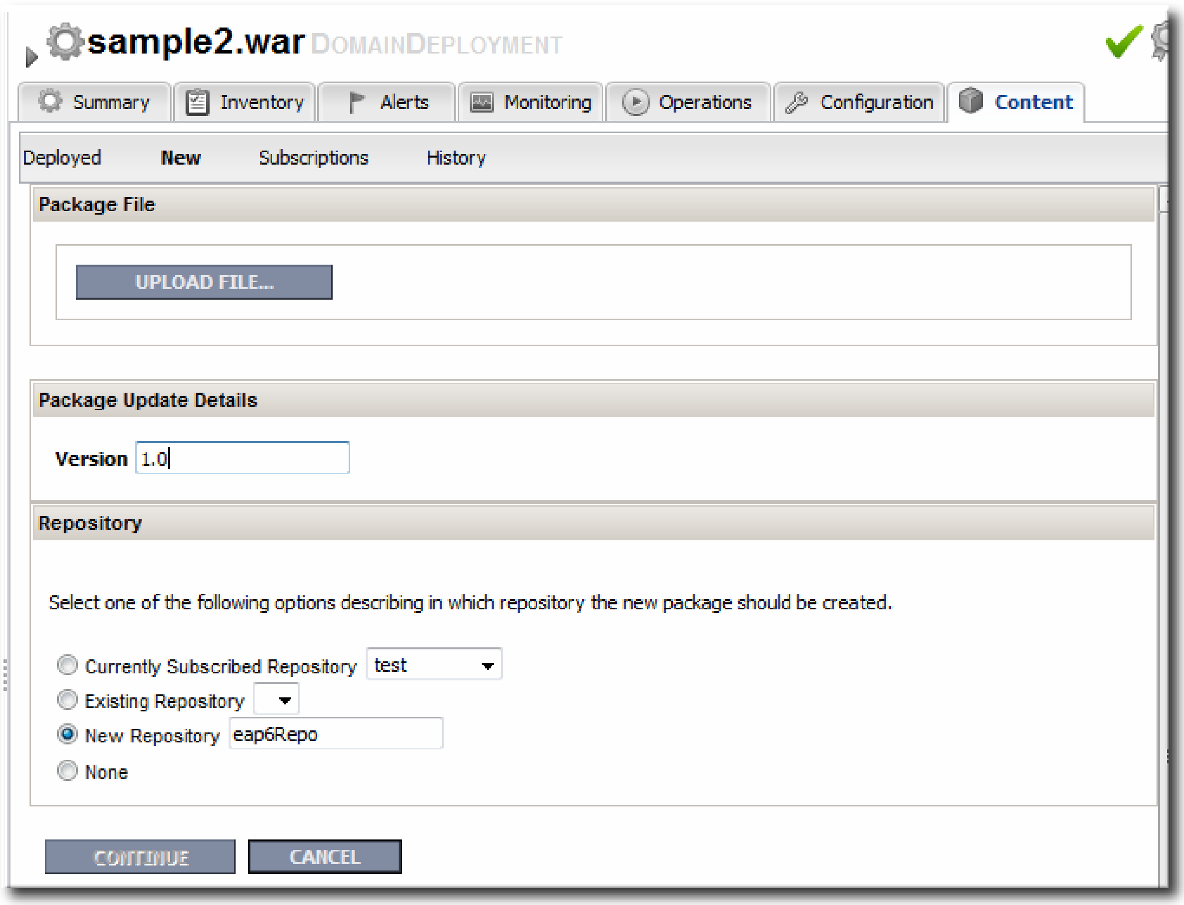
7. Select the **Enable** (or **Disable**) option from the drop-down menu.



8. Click the **Schedule** button.

31.5.6. Updating Deployment Content

1. Click the **Inventory** tab in the top menu.
2. Select **Servers - Top Level Imports** in the **Resources** menu table on the left. Select the JBoss EAP 6 server.
3. Open the **Deployments** folder (for the domain, standalone server, or server group), and select the web application to update.
4. On the web application details page, open the **Content** tab, and click the **New** subtab.
5. Click the **UPLOAD NEW PACKAGE** button.
6. Click the **UPLOAD FILE** button.
7. In the pop-up window, click the **Add** button, and browse the local filesystem to the updated content file to be uploaded.
8. Click the **UPLOAD** button.
9. Select the repository where the web application package should be stored. While this is not required, it is beneficial to store the updated package in JBoss ON so that it is available to other JBoss EAP 6 deployments.


sample2.war DOMAINDEPLOYMENT

Package File

Package Update Details

Version 1.0

Repository

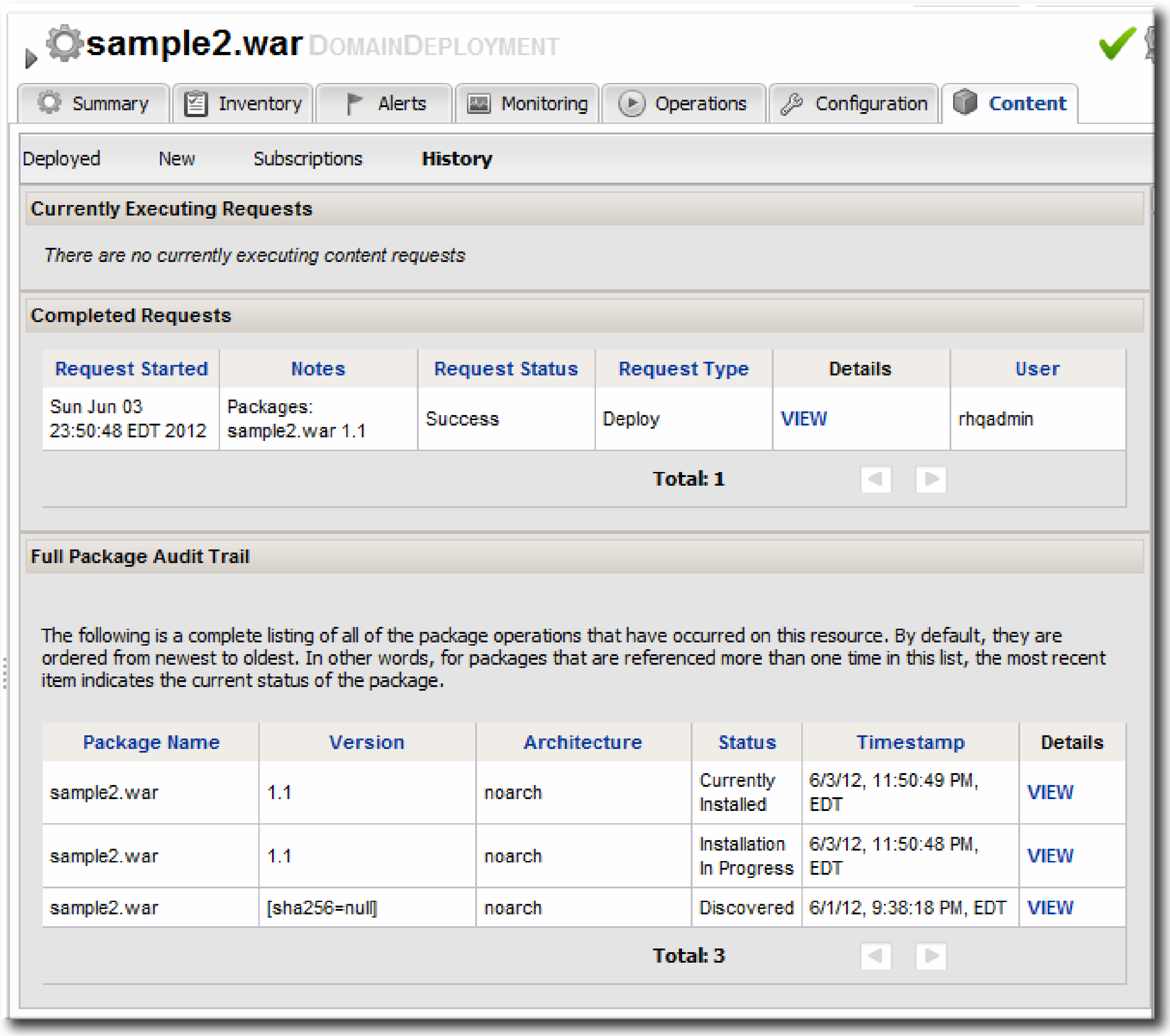
Select one of the following options describing in which repository the new package should be created.

- ○ Currently Subscribed Repository test
- ○ Existing Repository
- ● New Repository eap6Repo
- ○ None

10. Fill in the version information.

11. Confirm the details for the new package, then click **CONTINUE**.

When the package is successfully uploaded, the UI redirects to the history page on the **`Content`** tab.



Currently Executing Requests

There are no currently executing content requests

Completed Requests

Request Started	Notes	Request Status	Request Type	Details	User
Sun Jun 03 23:50:48 EDT 2012	Packages: sample2.war 1.1	Success	Deploy	VIEW	rhqadmin

Total: 1

Full Package Audit Trail

The following is a complete listing of all of the package operations that have occurred on this resource. By default, they are ordered from newest to oldest. In other words, for packages that are referenced more than one time in this list, the most recent item indicates the current status of the package.

Package Name	Version	Architecture	Status	Timestamp	Details
sample2.war	1.1	noarch	Currently Installed	6/3/12, 11:50:49 PM, EDT	VIEW
sample2.war	1.1	noarch	Installation In Progress	6/3/12, 11:50:48 PM, EDT	VIEW
sample2.war	[sha256=null]	noarch	Discovered	6/1/12, 9:38:18 PM, EDT	VIEW

Total: 3

Figure 31.14. Deployment History for a Resource

31.5.7. Deploying Web Applications to a Standalone Server

The standalone server uses a specific, filesystem directory as a deployment directory. As with domain deployments, content can be deployed by creating a child resource. However, because of the simpler structure of the standalone server, content can also be deployed through content bundles. Bundles provide simplified update and revert paths, as well as a clearer versioning system.

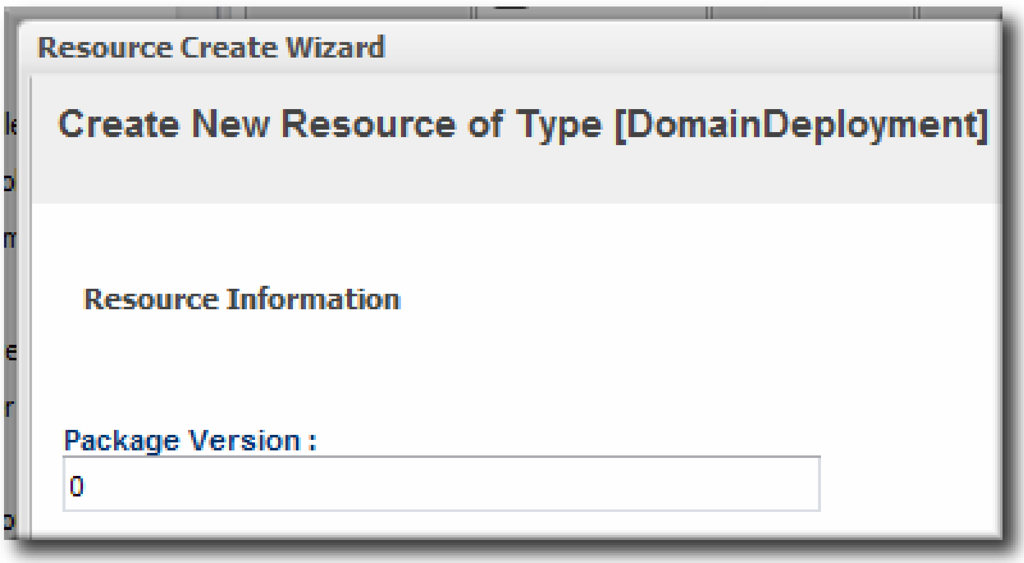
31.5.7.1. Deploying a Web Application as a Child Resource

IMPORTANT

New deployments may not show up in the JBoss ON inventory for a few minutes or for as long as 24 hours, even if it was successfully created. This is because the new resource must be discovered by the agent after it is created for it to be added to the inventory. By default, discovery scans for services are only made every 24 hours.

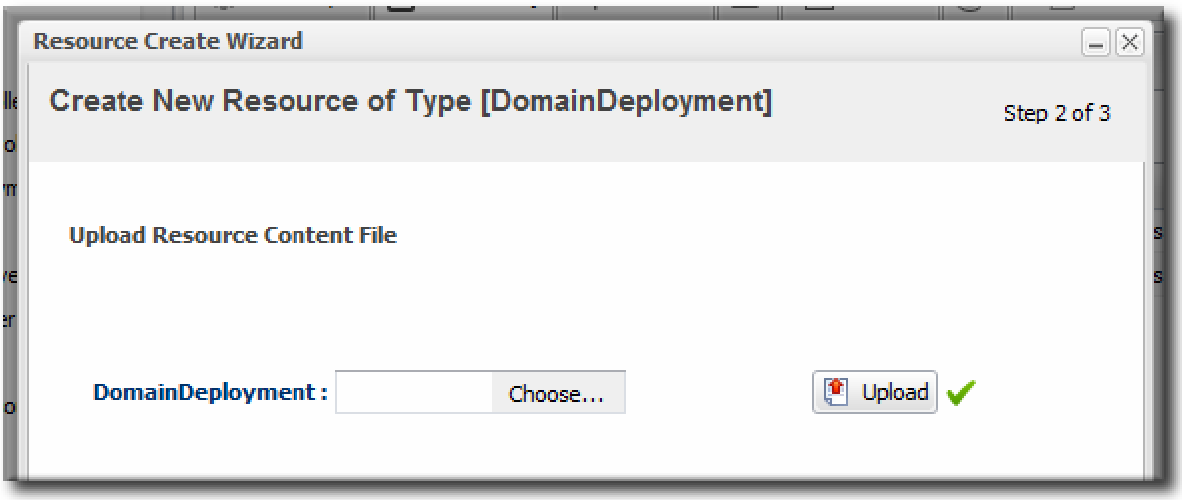
To see it immediately, run an *execute prompt command*operation on the agent and enter the `discovery` command. This runs a discovery scan.

1. Click the **`Inventory`** tab in the top menu.
2. Select **`Servers - Top Level Imports`** in the **`Resources`** menu table on the left. Select the JBoss EAP 6 standalone server.
3. Right-click the standalone server entry in the navigation tree.
4. In the **`Create New`** menu, select the item for **Deployment**.
5. Enter the version number.

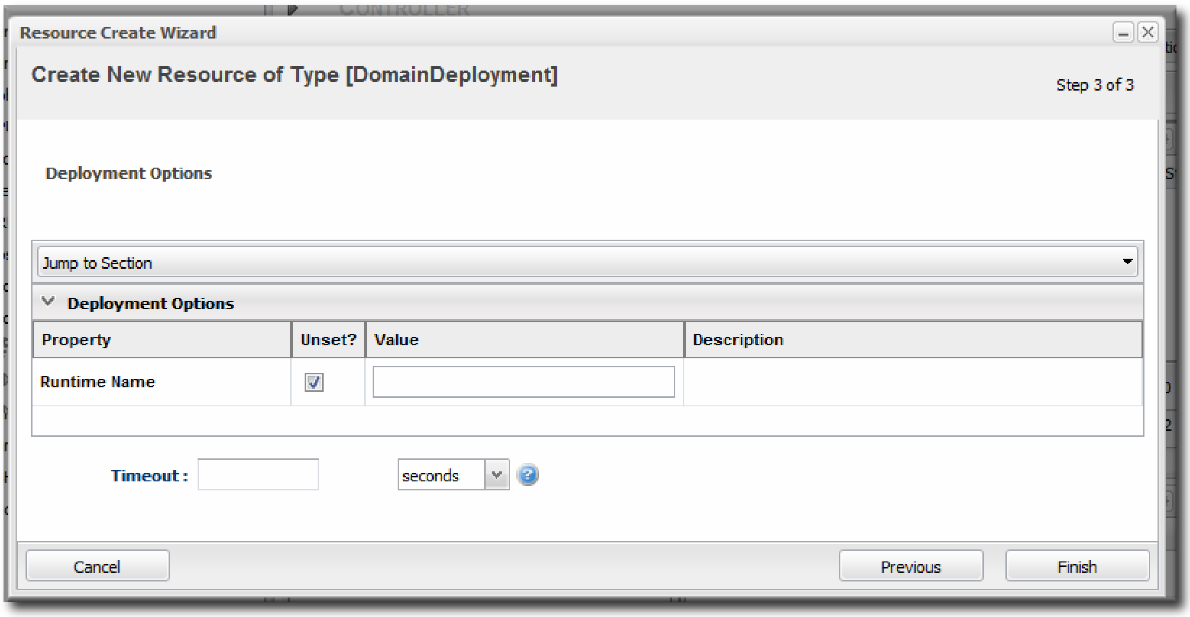


Package Version : 0

6. Upload the EAR, WAR, or JAR file.


Resource Create Wizard
Create New Resource of Type [DomainDeployment]
Step 2 of 3
Upload Resource Content File
DomainDeployment :
Choose...
Upload

7. Optionally, enter a runtime name for the deployment.


Resource Create Wizard

Create New Resource of Type [DomainDeployment]

Step 3 of 3

Deployment Options

Jump to Section

Deployment Options

Property	Unset?	Value	Description
Runtime Name	✓		

Timeout : seconds

8. At the bottom of the wizard, set an optional timeout period. This is how long the JBoss ON server will wait during the deployment process before determining that the deployment has failed.

NOTE

The timeout period only applies to the server's reporting a result. If the operation continues running, it can still complete successfully, and the web application is deployed.

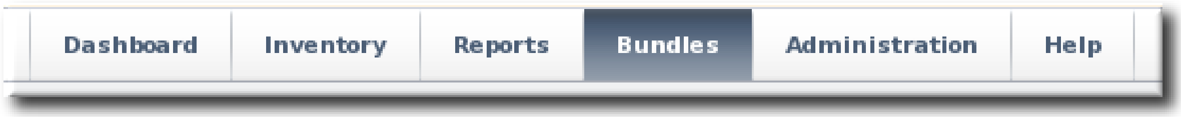
Particularly for large application files, do not set a low timeout period, or the server will mark the deployment as having failed. If the deployment completes later, the web application must be imported into the inventory manually; it will not be discovered by the agent.

9. Click **`Finish`**.

31.5.7.2. Deploying Web Applications Through Bundles

Creating bundle archives and recipes, defining destinations, and managing bundle versions are described in "Deploying Applications and Content" .

1. In the top menu, click the **`Bundles`** tab.



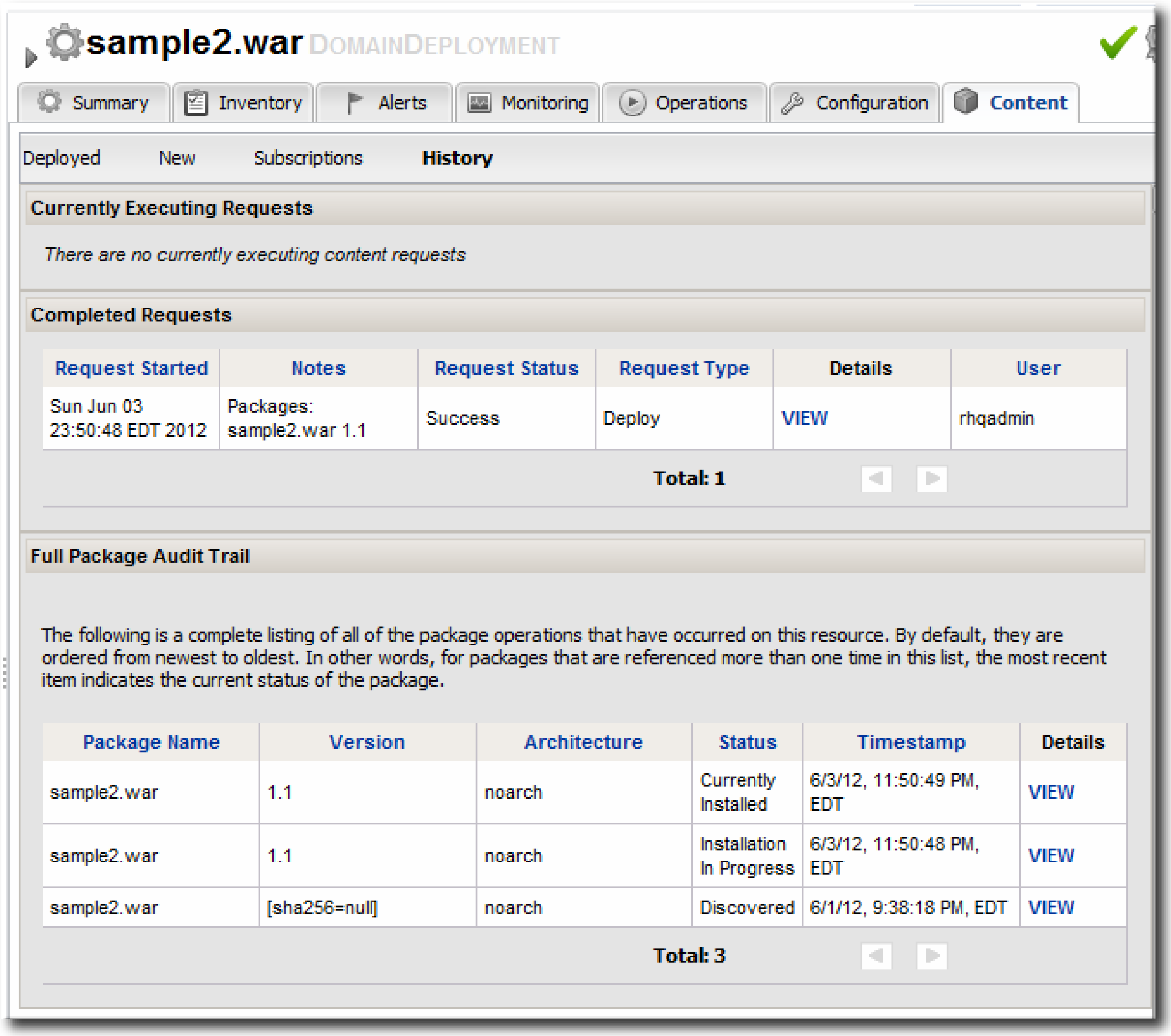
2. Upload the distribution file, and go through the deployment wizard to create the bundle version.
3. Scroll to the bottom of the window and click the **`Deploy`** button.
4. Select the bundles to deploy.

5. Define the destination information for the JBoss standalone server. A destination is a combination of a compatible group (containing the JBoss resource) and the directory to which to deploy the bundle.

6. Complete the deployment wizard, setting any require properties.

31.5.8. Tracking Content History and Reverting Changes

The deployment history page tracks changes to a content-backed resource, meaning it lists all update attempts, timestamps, and version numbers.



Currently Executing Requests

There are no currently executing content requests

Completed Requests

Request Started	Notes	Request Status	Request Type	Details	User
Sun Jun 03 23:50:48 EDT 2012	Packages: sample2.war 1.1	Success	Deploy	VIEW	rhqadmin

Total: 1

Full Package Audit Trail

The following is a complete listing of all of the package operations that have occurred on this resource. By default, they are ordered from newest to oldest. In other words, for packages that are referenced more than one time in this list, the most recent item indicates the current status of the package.

Package Name	Version	Architecture	Status	Timestamp	Details
sample2.war	1.1	noarch	Currently Installed	6/3/12, 11:50:49 PM, EDT	VIEW
sample2.war	1.1	noarch	Installation In Progress	6/3/12, 11:50:48 PM, EDT	VIEW
sample2.war	[sha256=null]	noarch	Discovered	6/1/12, 9:38:18 PM, EDT	VIEW

Total: 3

Figure 31.15. Deployment History for a Resource

Web applications deployed to standalone servers through the bundles provisioning system have a straightforward path to revert changes. There is a `Revert` button on the details page for the latest deployment which goes back to the immediate past deployed version.

Deployments to a server group or domain, however, do not have the same clear path. Any attempt to revert a content deployment ultimately means to replace it with a different version.

There are options for how to change the package when a direct reversion is not possible:

- If the domain deployment or server group deployment is associated with a content repository, then upload an updated package to the content repository and sync the change over to the associated resources.

- Upload an updated package to the domain deployment and use the *deploy to group* operation to send the updated package to the server groups.

31.5.9. Troubleshooting Deployments

Q: My deployment says it failed, but when I tried to redeploy the package, it fails because it says the resource already exists.

A: One possible cause is the timeout setting when the package was originally deployed. The timeout period sets how long the JBoss ON server waits before assuming the deploy operation failed — it does not actually stop or limit the deploy operation itself. Even if the operation hits the timeout limit, the operation can still complete successfully, which successfully deploys the web application.

If the operation times out, then the agent will not discover the new resource automatically. The web application must be imported into the inventory manually.

Q: I tried to deploy a package to a server group and it seemed to be successful, but I don't see the deployment listed.

A: There are a couple of possible causes:

The agent discovery scan hasn't run. There can be several hours between discovery runs, so it can take awhile for the application to appear in the discovery queue. To work around this, run a discovery scan on the agent.

The package has already been deployed to the *domain*. Creating a deployment child on a server group actually attempts to create the deployment on the domain (where the content is stored) and then deploys it to the server group. If the package already exists in the domain, then attempting to re-create the deployment on the server group fails as a duplicate.

In this case, use the *deploy to server group* operation on the domain controller to deploy the application.

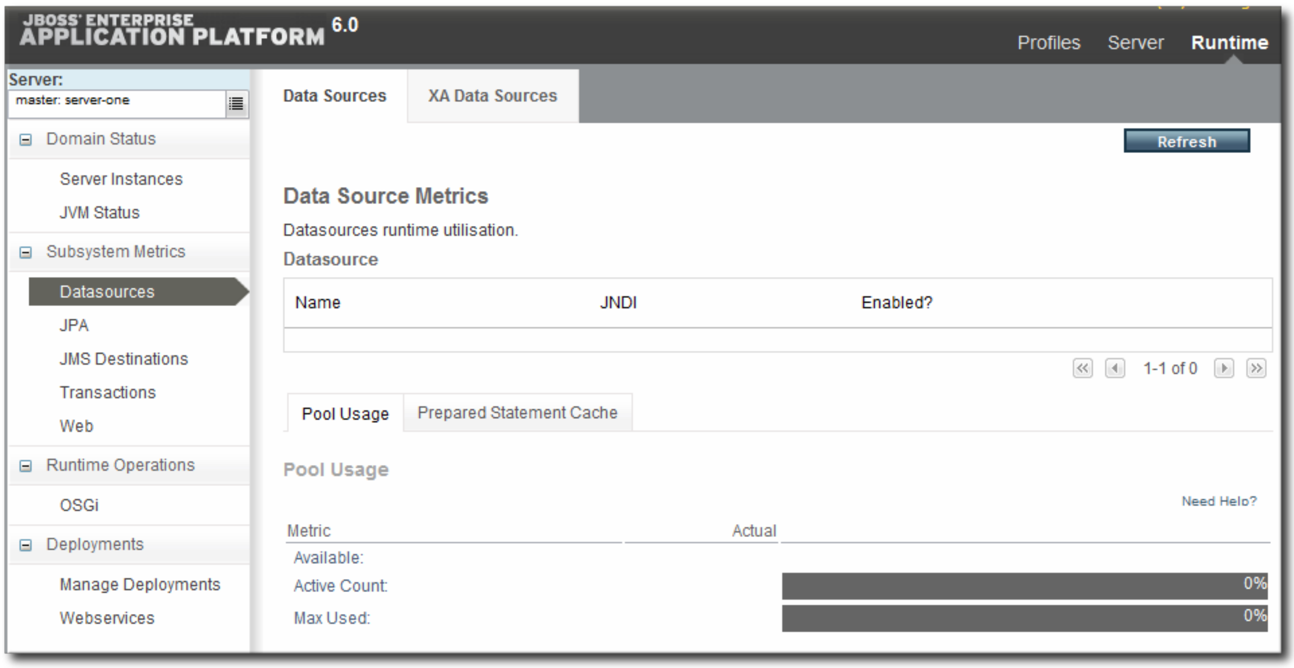
31.6. MONITORING JBOSS EAP 6 RESOURCES

31.6.1. Runtime Information and JBoss ON Monitoring

Metrics are a live-stream of information about how a resource or group of resources are performing. Metrics are vital signs, like the heart rate, blood pressure, and oxygen levels for a hospital patient. For resources, those vital statistics are elements like availability, page hits for database caches, or active connections for datasources or web applications. Those metrics show resource health, observed at a specific moment.

The JBoss EAP 6 console shows that current and immediate snapshot of information for a handful of critical metrics for a few critical resources, like datasources, transactions, and web applications.

This changing information is displayed in the **`Runtime`** tab of the EAP 6 console.



Data Source Metrics

Datasources runtime utilisation.

Datasource

Name	JNDI	Enabled?

1-1 of 0

Pool Usage

Metric	Actual
Available:	
Active Count:	0%
Max Used:	0%

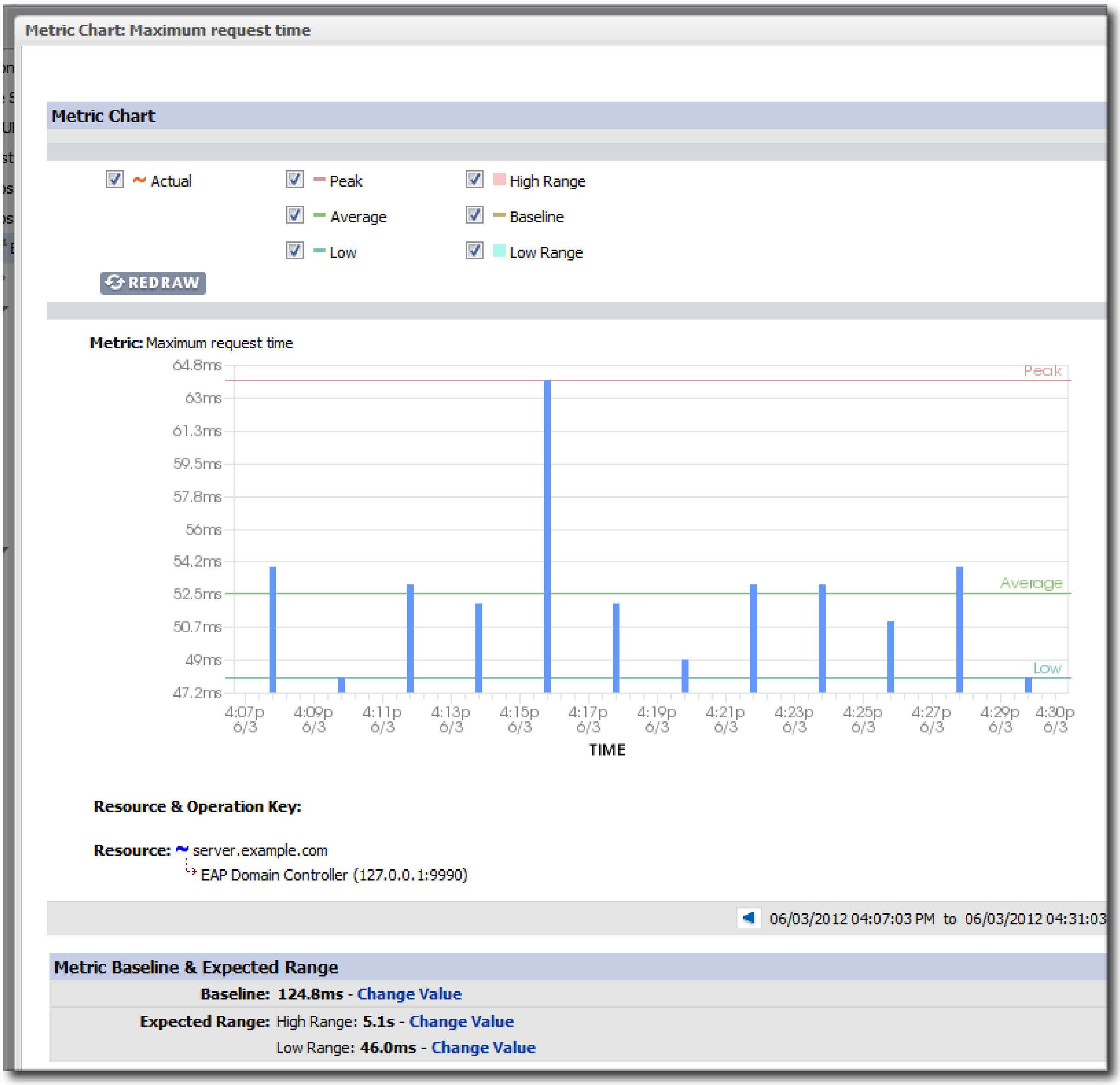
Figure 31.16. EAP 6 Runtime Metrics

Runtime information shows the health for that exact moment in time.

Looking at the runtime information *over time* gives a much better perspective of how a resource is performing, generally. Every resource in JBoss ON has areas to view and configure monitoring and to configure alerting based on that monitoring.

JBoss ON automatically processes runtime metrics in a few different ways, to help administrators process performance data:

- Time-based monitoring graphs for each metric
- Operating parameters, or *baselines* calculated on the real performance of the resource
- Availability uptime percentages, recovery times, and mean down time



Resource & Operation Key:

Resource: server.example.com
EAP Domain Controller (127.0.0.1:9990)

06/03/2012 04:07:03 PM to 06/03/2012 04:31:03

Metric Baseline & Expected Range

Baseline: **124.8ms** - Change Value

Expected Range: High Range: **5.1s** - Change Value

Low Range: **46.0ms** - Change Value

Figure 31.17. JBoss ON Baselines for a Single Metric

JBoss ON also exposes more monitoring data for individual resources through its GUI:

- More metrics are available for resources. At a minimum, every resource has availability monitoring. Some resources — like datasources and managed servers — have dozens more metrics that can be enabled for routine monitoring.

- Response time monitoring can be configured for specific URLs and pages for web applications. This allows administrators to track usability and customer experience for web applications along with internal metrics like connection counts.

- Event log monitoring can filter important messages from different error logs. This allows JBoss ON to respond (through alerts) to issues that may not have crossed a performance threshold yet, and it also makes it possible for administrators to correlate log events with observed performance metrics.

31.6.2. Setting up Monitoring for EAP 6 Resources

The **Runtime** tab in the JBoss EAP 6 management console gives an current snapshot of the state of managed servers and some common subsystems. The emphasis of the EAP 6 console is on quick access to current information. This makes it easy to perform immediate management tasks.

JBoss ON focuses on long-term performance tracking, not just an immediate view. The EAP 6 agent plug-in for JBoss ON allows administrators to monitor additional metrics for resources, to track historical measurements and establish performance baselines, and to view how the resource status has changed (gone up or down) and how long it has been in different states.

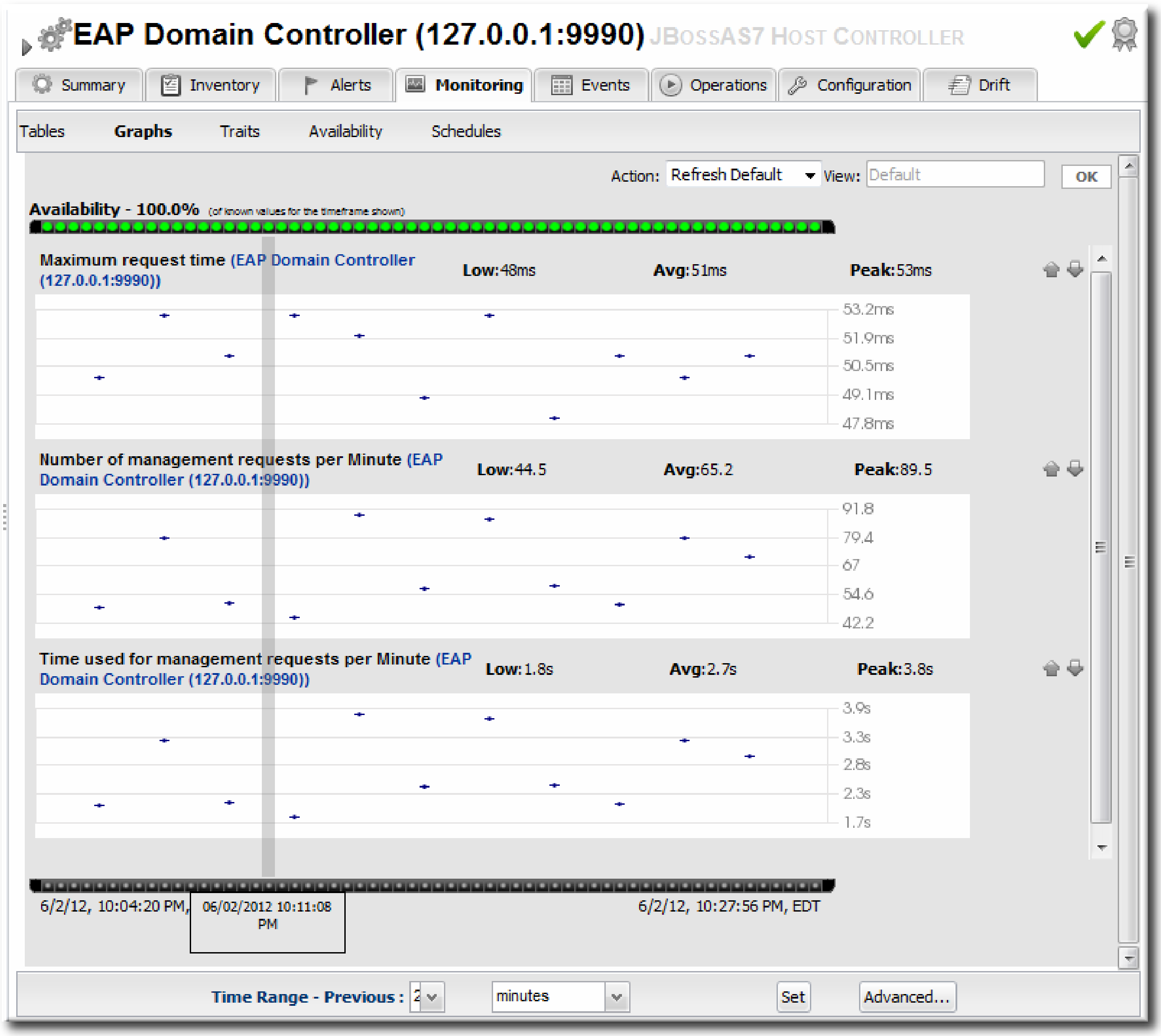

EAP Domain Controller (127.0.0.1:9990) JBOSSAS7 HOST CONTROLLER
Summary
Inventory
Alerts
Monitoring
Events
Operations
Configuration
Drift
Tables
Graphs
Traits
Availability
Schedules
Action: Refresh Default
View: Default
OK
Availability - 100.0% (of known values for the timeframe shown)
Maximum request time (EAP Domain Controller (127.0.0.1:9990))
Low:48ms
Avg:51ms
Peak:53ms
Number of management requests per Minute (EAP Domain Controller (127.0.0.1:9990))
Low:44.5
Avg:65.2
Peak:89.5
Time used for management requests per Minute (EAP Domain Controller (127.0.0.1:9990))
Low:1.8s
Avg:2.7s
Peak:3.8s
6/2/12, 10:04:20 PM,
06/02/2012 10:11:08 PM
6/2/12, 10:27:56 PM, EDT
Time Range - Previous : 2
minutes
Set
Advanced...

Figure 31.18. Monitoring Graph

Monitoring information is displayed in graphs and in tables on the **Monitoring** tab. The **Summary** tab has visually different line graphs rendered in the monitoring portlet.

NOTE

For more information on monitoring concepts and configuration, see Section 31.6, “Monitoring JBoss EAP 6 Resources”.

31.6.2.1. Enabling Additional Metrics

JBoss ON supports all of the metrics for JBoss EAP 6 resources that are displayed in the EAP 6 console, along with many additional metrics.

The metrics enabled by default in JBoss ON, however, may not be the same ones that are viewable in the EAP 6 console. For example, for data sources, the EAP 6 console displays the available connections count, active connections count, and max used connections count. In JBoss ON, the default metrics collected are the maximum and minimum pool size settings, the max wait time, and the number of time outs per minute.

NOTE

A complete list of possible metrics for each resource type is given in the Resource Reference.

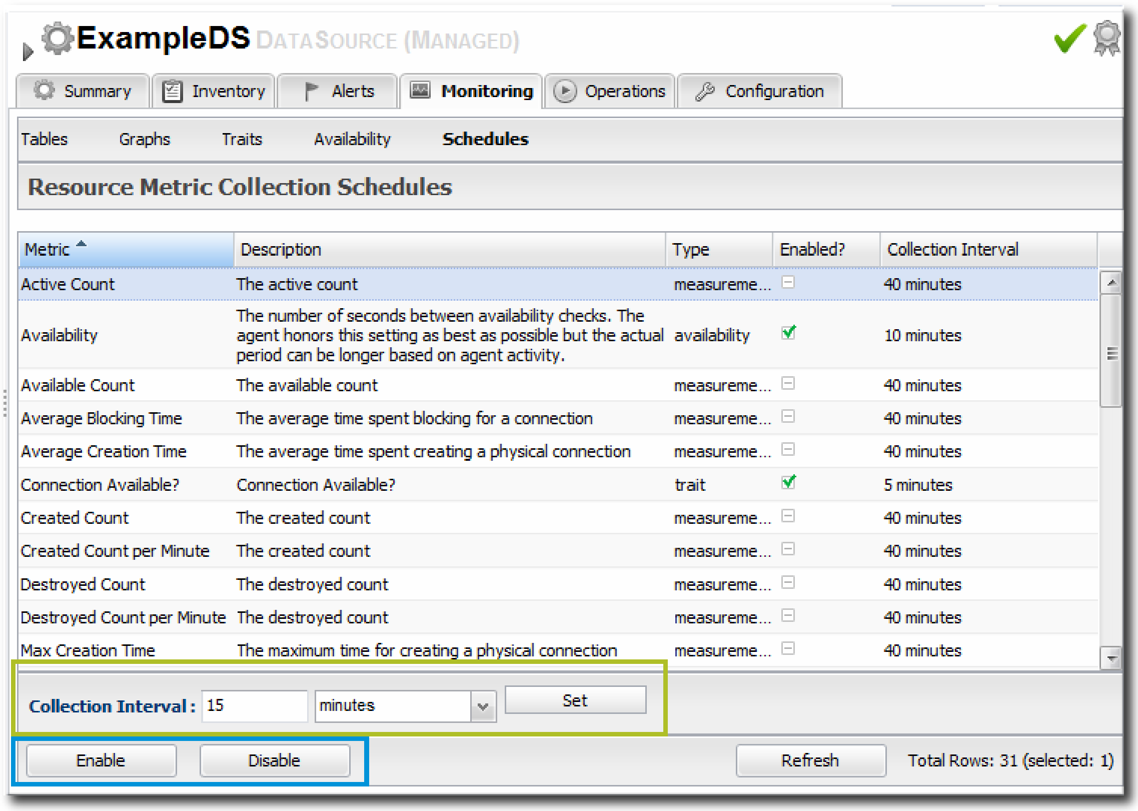
Metric collection can be enabled or disabled for a resource. Administrators can also reset how often JBoss ON checks a given metric (the *collection interval*).

NOTE

Checking less important metrics less frequently can improve platform and agent performance, while still recording that information for reference or event correlation.

To change the monitoring configuration for a resource:

1. Click the **`Inventory`** tab in the top menu.
2. Select **`Servers - Top Level Imports`** in the **`Resources`** menu table on the left. Then select the JBoss EAP 6 server and navigate to the resource to edit.
3. Click the **`Monitoring`** tab on the resource entry.
4. Click the **`Schedules`** subtab.
5. To enable a metric:
    1. Select the metric row. Multiple metrics can be selected using the **`Ctrl`** key.
    2. Click the **`Enable`** button at the bottom of the page.
6. To change a collection interval for a metric:
    1. Select the metric row. Multiple metrics can be selected using the **`Ctrl`** key, if they will all be changed to the same frequency.



ExampleDS DATASOURCE (MANAGED)

Resource Metric Collection Schedules

Metric	Description	Type	Enabled?	Collection Interval
Active Count	The active count	measureme...	☐	40 minutes
Availability	The number of seconds between availability checks. The agent honors this setting as best as possible but the actual period can be longer based on agent activity.	availability	✓	10 minutes
Available Count	The available count	measureme...	☐	40 minutes
Average Blocking Time	The average time spent blocking for a connection	measureme...	☐	40 minutes
Average Creation Time	The average time spent creating a physical connection	measureme...	☐	40 minutes
Connection Available?	Connection Available?	trait	✓	5 minutes
Created Count	The created count	measureme...	☐	40 minutes
Created Count per Minute	The created count	measureme...	☐	40 minutes
Destroyed Count	The destroyed count	measureme...	☐	40 minutes
Destroyed Count per Minute	The destroyed count	measureme...	☐	40 minutes
Max Creation Time	The maximum time for creating a physical connection	measureme...	☐	40 minutes

Collection Interval: 15 minutes

Total Rows: 31 (selected: 1)

2. Enter the desired collection period in the `Collection Interval` field, with the appropriate time unit (seconds, minutes, or hours).

3. Click **Set**.

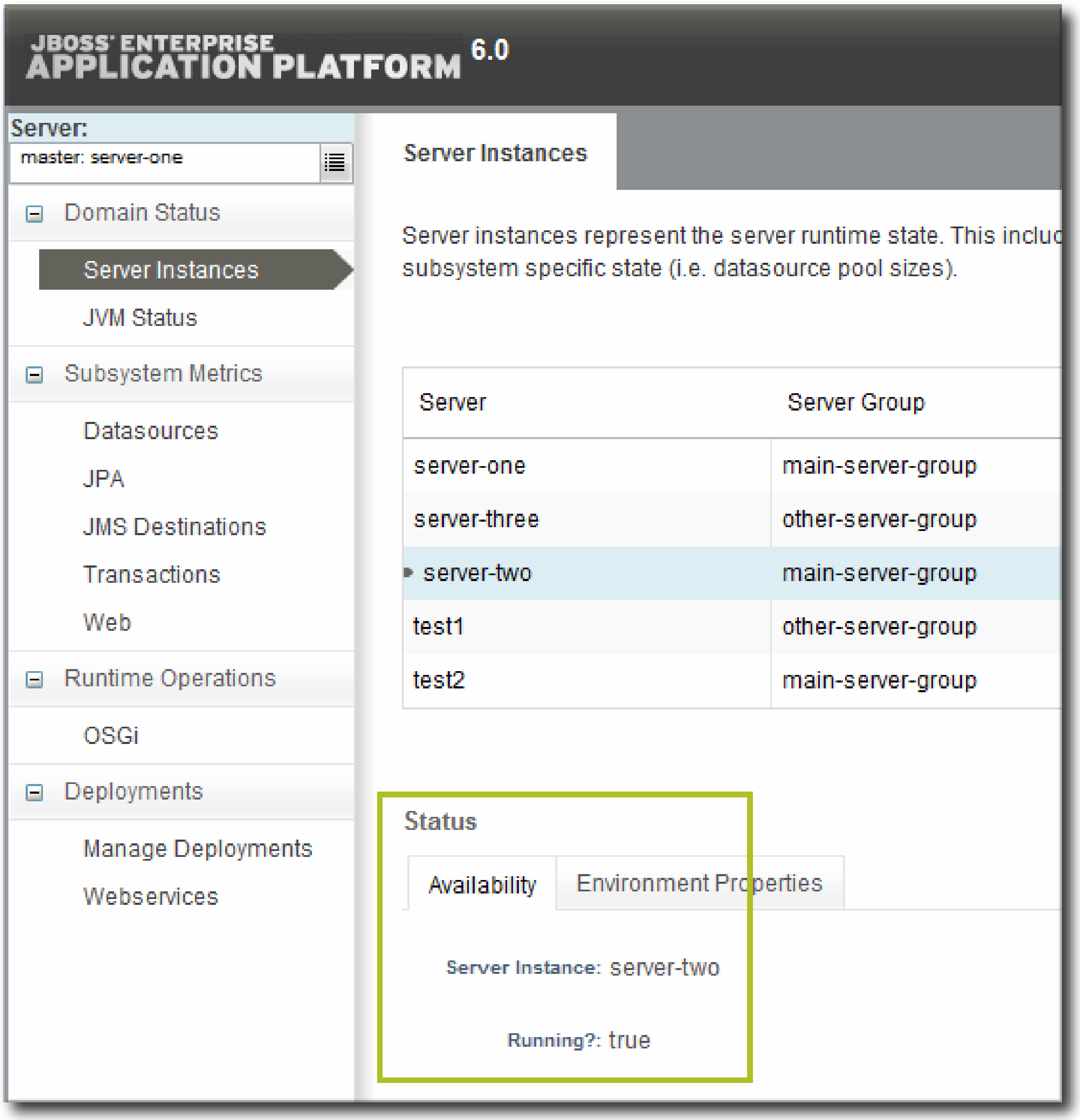
NOTE

Changing the collection interval for a disabled metric automatically enables the metric.

31.6.2.2. Availability Monitoring

Availability is a (mostly) straightforward measurement that shows whether a resource is running and minimally responsive. If the agent can connect to the resource, that resource is available.

This is how availability is displayed in the EAP 6 console for a resource: it shows whether a resource is currently running.



Server instances represent the server runtime state. This incluc
subsystem specific state (i.e. datasource pool sizes).

Server	Server Group
server-one	main-server-group
server-three	other-server-group
server-two	main-server-group
test1	other-server-group
test2	main-server-group

Server Instance: server-two

Running?: true

Figure 31.19. Availability in the EAP 6 Console

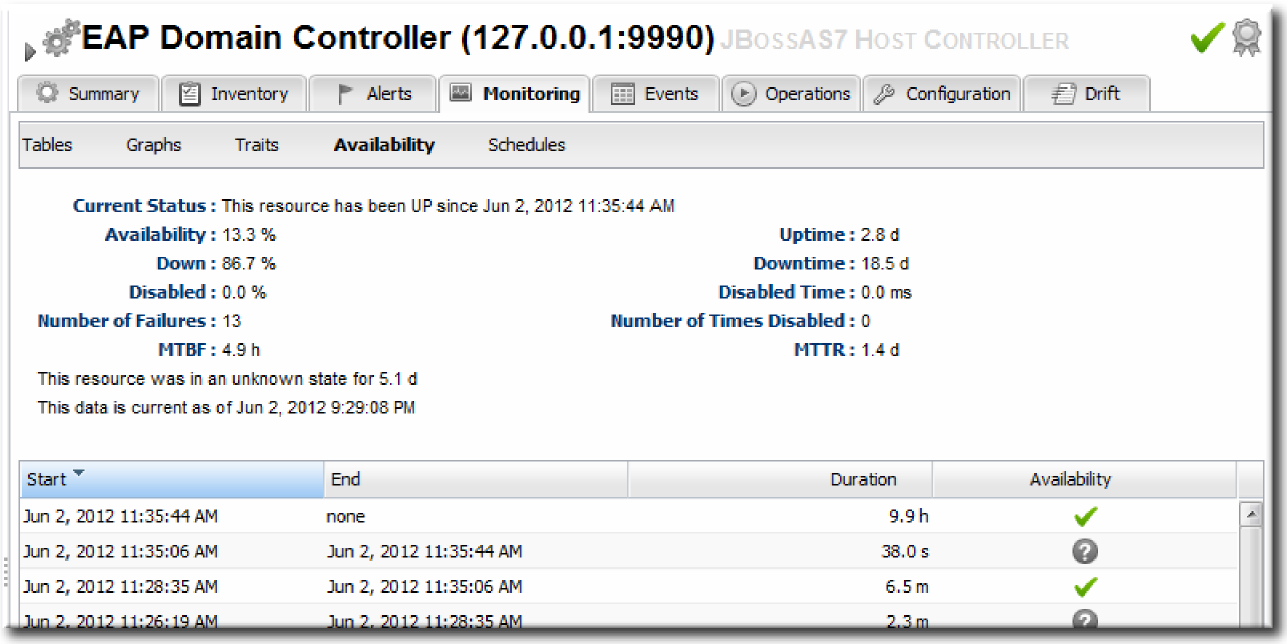
As a measure of performance, availability can acquire more shades of gray than simply *is it running*. For example, has a resource been cycling off and on repeatedly? Does it stay offline for long periods before it recovers? Is a resource unavailable because of its own failure or is a parent resource (like the platform) unreachable, while the child is running fine? JBoss ON supports four resources states — up, down, disabled, and unknown — to try to differentiate between real states for the resources.

Availability is covered in much more detail in the Chapter 18, *Availability*.

Along with showing the current status of a resource, JBoss ON also shows how long the resource has been in that state and what percentage of time the resource spends in different states. This allows JBoss ON to calculate a few different trends per resource:

- Its overall uptime percentage
- Total time spent in different states (up, down, or disabled)

- The total number of failures (times it has gone down)
- The mean time between failures, which is essentially the mean time that a resource is up and available
- The mean time to recovery, which is the mean amount of time that it takes a resource to come back up after a failure



Current Status : This resource has been UP since Jun 2, 2012 11:35:44 AM

Availability :	13.3 %	Uptime :	2.8 d
Down :	86.7 %	Downtime :	18.5 d
Disabled :	0.0 %	Disabled Time :	0.0 ms
Number of Failures :	13	Number of Times Disabled :	0
MTBF :	4.9 h	MTTR :	1.4 d

This resource was in an unknown state for 5.1 d

This data is current as of Jun 2, 2012 9:29:08 PM

Start	End	Duration	Availability
Jun 2, 2012 11:35:44 AM	none	9.9 h	✓
Jun 2, 2012 11:35:06 AM	Jun 2, 2012 11:35:44 AM	38.0 s	?
Jun 2, 2012 11:28:35 AM	Jun 2, 2012 11:35:06 AM	6.5 m	✓
Jun 2, 2012 11:26:19 AM	Jun 2, 2012 11:28:35 AM	2.3 m	?

Figure 31.20. Availability in JBoss ON

Availability is collected on a schedule, much like a monitoring metric. Controlling the frequency of availability checks can establish a resource priority.

NOTE

Servers, such as domain controllers and managed servers, have availability checks that run every minute. This can help catch resource cycling in unstable servers. Services, like datasources and JMS queues, are checked for availability every 10 minutes.

The frequency of availability checks can have an impact on agent performance, particularly for servers structured like EAP 6, with a large number of child resources. Every resource is checked for availability, by default. There may be some low-level child resource, like OSGi classes or EJBs, that really do not require an independent availability check. In that case, availability checks can be disabled for the child. The agent automatically applies the parent state to the child; if the parent is running, than the child is assumed to be running. This is called *backfilling*.

31.6.3. Configuring Events Monitoring

Event monitoring is essentially a log filter. The resource itself maintains a log file; JBoss ON checks any specified log file for messages or events that occur. JBoss ON can check for messages in a stream or only for messages that match a certain pattern or severity.

JBoss ON can issue alerts based on log messages and even take action automatically in response to events, like creating a new managed resource or restarting a server.

Since events are unpredictable, based on the real operations of the resource rather than a schedule, it is a big advantage to administrators to be able to filter messages and respond dynamically.

NOTE

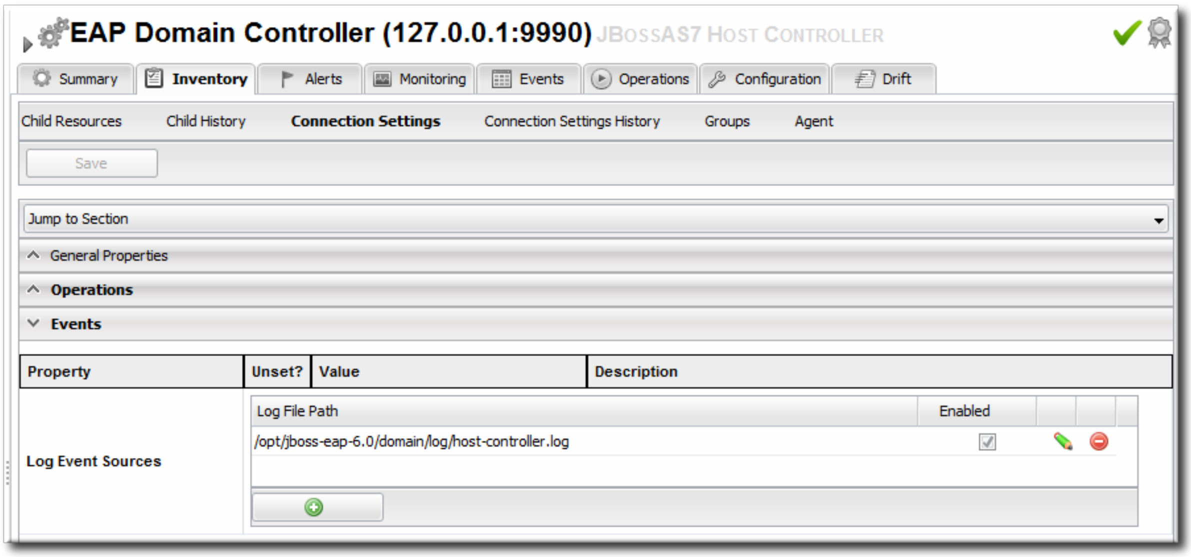
Events configuration is described more in Chapter 20, *Events*.

Three JBoss EAP 6 server types support events, and they are preconfigured with an event log in the standard location:

- Host controllers, in **`domain/log/host-controller.log`**
- Managed servers, in **`domain/servers/server-three/log/server.log`**
- Standalone servers, in **`standalone/log/server.log`**

While the event log is already configured, it is not enabled by default. The event log must be enabled before events are collected by the JBoss ON agent. Additional settings, such as specific strings to look for in log messages, can also be set.

1. Click the **`Inventory`** tab in the top menu.
2. Click the **`Servers - Top Level Imports`** item, and select the JBoss EAP 6 resource.
3. Navigate to the appropriate EAP 6 resource.
4. Click the **`Inventory`** tab on the resource entry.
5. Select the **`Connection Settings`** subtab.
6. Click green plus icon under the **`Events Log`** section.

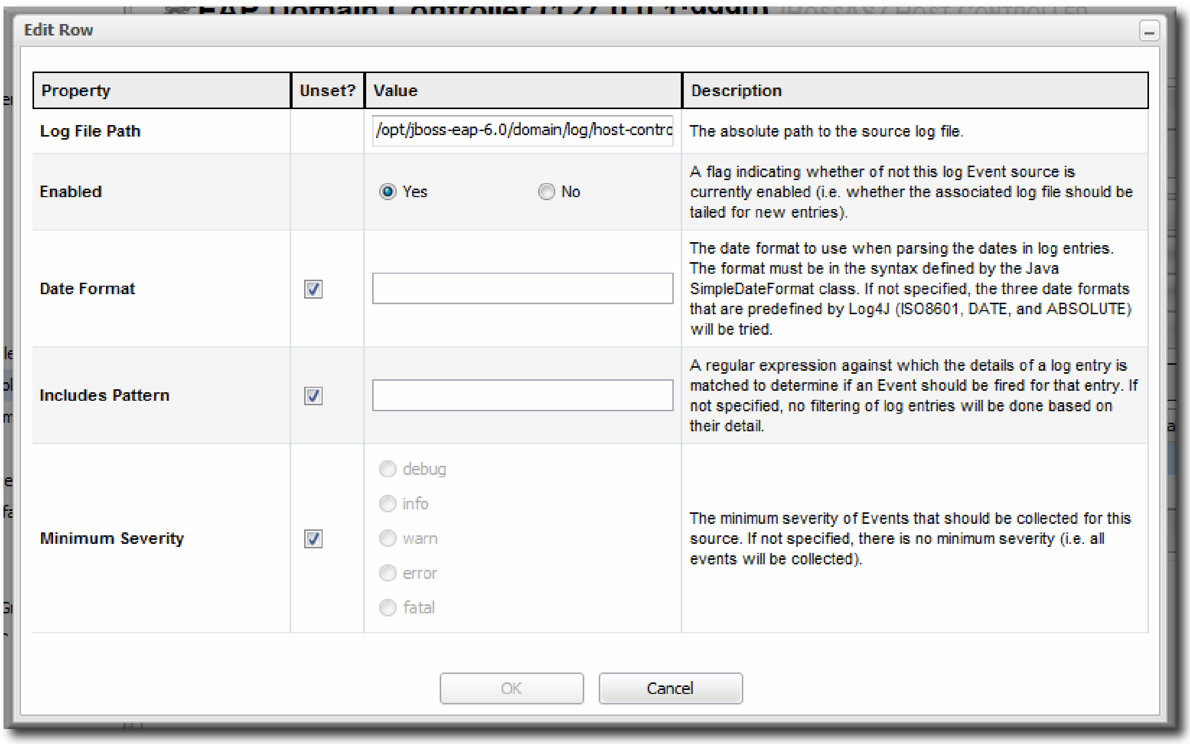


Property	Unset?	Value	Description
Log Event Sources			

Log File Path	Enabled
/opt/jboss-eap-6.0/domain/log/host-controller.log	✓

7. Enable the event entry.

   Optionally, set the date format, string filters to include messages, or the severity of the messages to include.

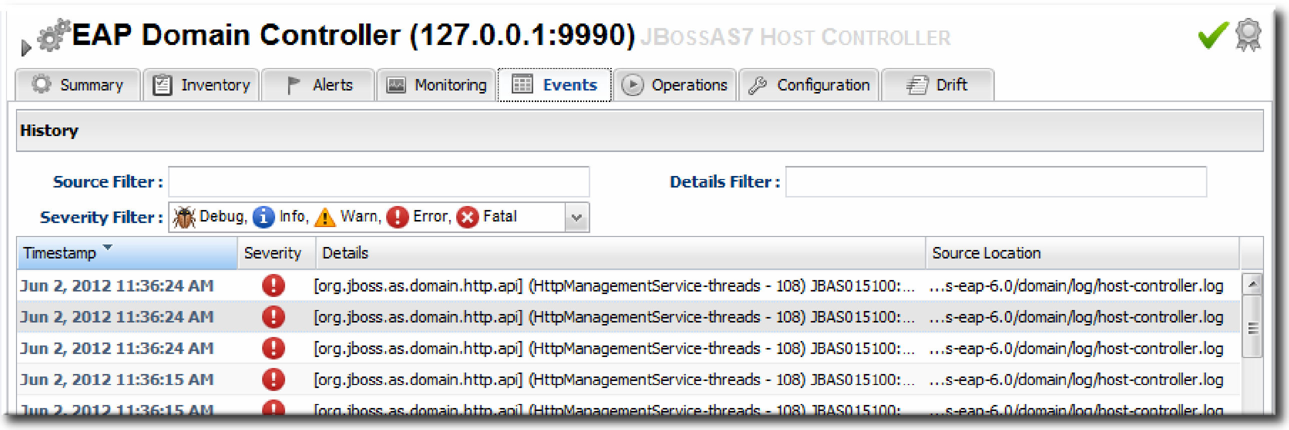


Edit Row

Property	Unset?	Value	Description
Log File Path		/opt/jboss-eap-6.0/domain/log/host-contrc	The absolute path to the source log file.
Enabled		[✓ Yes] No	A flag indicating whether of not this log Event source is currently enabled (i.e. whether the associated log file should be tailed for new entries).
Date Format	✓		The date format to use when parsing the dates in log entries. The format must be in the syntax defined by the Java SimpleDateFormat class. If not specified, the three date formats that are predefined by Log4J (ISO8601, DATE, and ABSOLUTE) will be tried.
Includes Pattern	✓		A regular expression against which the details of a log entry is matched to determine if an Event should be fired for that entry. If not specified, no filtering of log entries will be done based on their detail.
Minimum Severity	✓	debug info warn error fatal	The minimum severity of Events that should be collected for this source. If not specified, there is no minimum severity (i.e. all events will be collected).

OK Cancel

When event monitoring is enabled, the matching log messages are displayed in the **`Events`** tab for the resource.



EAP Domain Controller (127.0.0.1:9990) JBOSSAS7 HOST CONTROLLER

Summary | Inventory | Alerts | Monitoring | Events | Operations | Configuration | Drift

History

Source Filter :

Details Filter :

Severity Filter : Debug, Info, Warn, Error, Fatal

Timestamp	Severity	Details	Source Location
Jun 2, 2012 11:36:24 AM	❗	[org.jboss.as.domain.http.api] (HttpManagementService-threads - 108) JBAS015100:...	...s-eap-6.0/domain/log/host-controller.log
Jun 2, 2012 11:36:24 AM	❗	[org.jboss.as.domain.http.api] (HttpManagementService-threads - 108) JBAS015100:...	...s-eap-6.0/domain/log/host-controller.log
Jun 2, 2012 11:36:24 AM	❗	[org.jboss.as.domain.http.api] (HttpManagementService-threads - 108) JBAS015100:...	...s-eap-6.0/domain/log/host-controller.log
Jun 2, 2012 11:36:15 AM	❗	[org.jboss.as.domain.http.api] (HttpManagementService-threads - 108) JBAS015100:...	...s-eap-6.0/domain/log/host-controller.log
Jun 2, 2012 11:36:15 AM	❗	[org.jboss.as.domain.http.api] (HttpManagementService-threads - 108) JBAS015100:...	...s-eap-6.0/domain/log/host-controller.log

Figure 31.21. JBoss EAP 6 Event Logging

31.6.4. Alerting on JBoss EAP 6 Resources

Alerting is covered in detail in Chapter 25, *Defining Alerts*, and that is the best reference for details on planning what alert conditions to set and how to respond.

The key thing in planning alerts is planning automatic responses to whatever condition triggered the alert. For example, within an EAP 6 domain, one managed server may go offline or begin exhibiting extremely slow response times. In that case, JBoss ON can automatically create a new managed server instance, within the same server group, on another platform to take the failing server's place.

Taking an immediate response, without necessarily requiring administrator intervention, is invaluable for customer-facing problems, such as poor response times for a web application.

There are several types of responses that JBoss ON can take:

- Email an administrator

- Run a restart operation on the alerting resource
- Run an operation on the parent of the alerting resource
- Run a shell script
- Run a JBoss ON server-side script

  Server-side scripts are extremely powerful. A script can perform almost any management task in JBoss ON: change resource configuration, deploy updated packages, create new resources, delete existing resources, run an operation, or all of the above. For more information on writing server-side scripts, see "Writing JBoss ON Command-Line Scripts" .

31.7. USING THE MOD_CLUSTER SERVICES IN EAP 6

The **`mod_cluster`** modules provides dynamic load balancing between web contexts on a JBoss application server and an Apache web server.

There are two halves to creating a cluster with **`mod_cluster`**: configuring the module on JBoss to manage the web applications and configuring the module on Apache to manage sessions and routing.

JBoss ON can manage the embedded **`mod_cluster`** subsystem for JBoss EAP 6 for both domain servers and standalone servers.

31.7.1. About mod_cluster and JBoss ON

The **`mod_cluster`** module, a subsystem on EAP 6, communicates between the web applications on a JBoss EAP server and an Apache web server. Multiple JBoss EAP servers can be involved in a **`mod_cluster`** group, and those servers can be managed servers, standalone servers, or a mix of both.

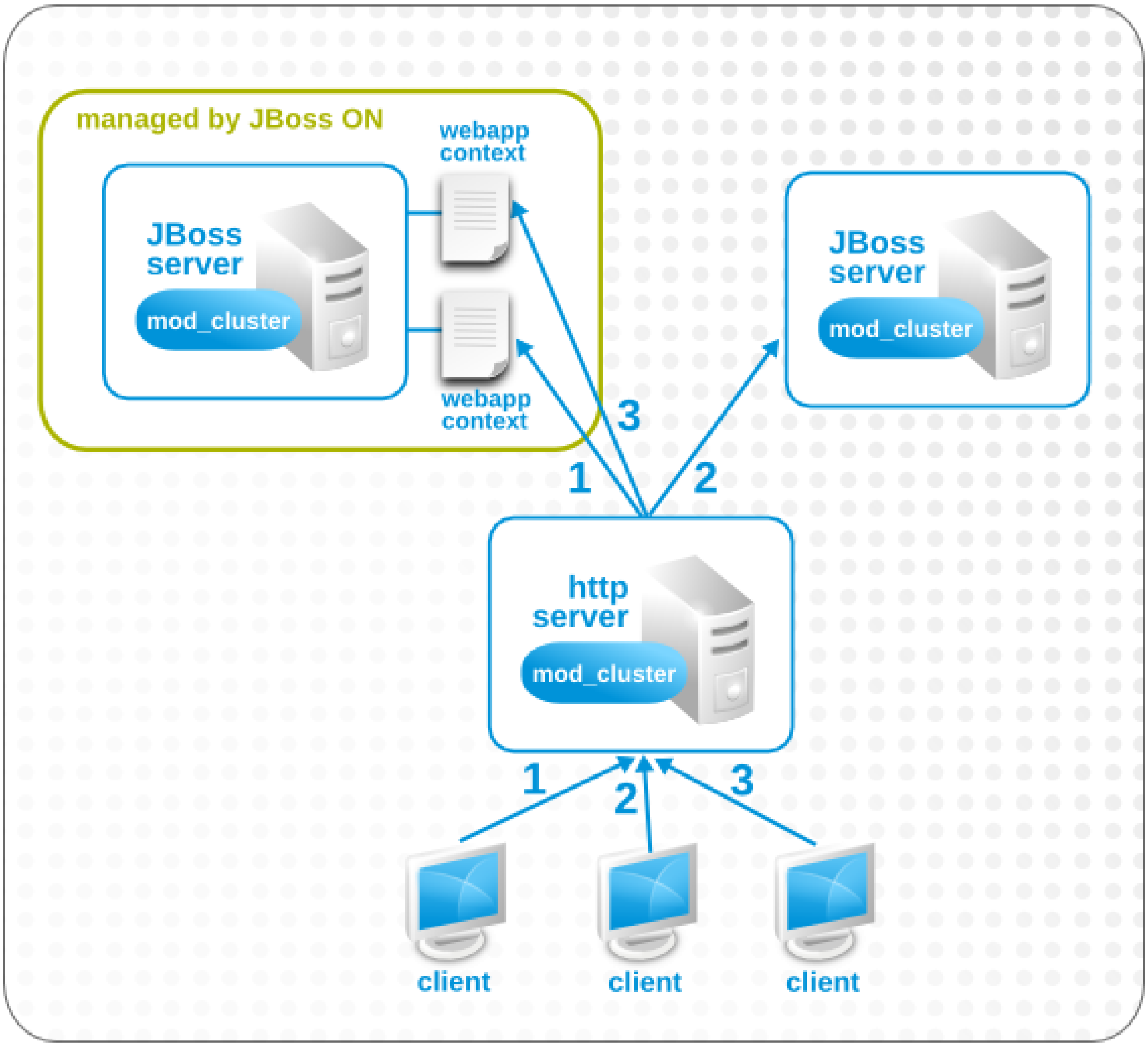


Figure 31.22. The mod_cluster Topology

Only high availability profiles contain the **`mod_cluster`** module. For domains, the **`other-server-group`** server group is configured to use the **`full-ha`** profiles, though any profile which supports high availability can be used (such as the **ha** profile or a custom profile). For standalone servers, the server must be started with the **`standalone-ha.xml`** configuration.

One EAP server within the **`mod_cluster`** server is the master node; that is the administering **`mod_cluster`** service. Every other member of the cluster is a worker node.

NOTE

Information about **`mod_cluster`** in general is available with the mod_cluster project documentation.

Whether a specific resource belongs to the **`mod_cluster`** domain depends on the profile that resource is associated with, which is largely outside of the control of JBoss ON. (Of course, a standalone server can be started with the high availability configuration with the **JAVA_OPTS** settings in JBoss ON for start scripts, or a new managed server can be created that uses a high availability profile. But the profile definitions themselves are created and maintained outside of JBoss ON.)

JBoss ON manages the **mod_cluster** configuration itself. The cluster settings include multicast (advertising), load balancing, session handling, and network settings.

31.7.2. Configuring Multicast for Load Balancing

mod_cluster can use multicast signals to inform nodes which proxy servers are available. This allows nodes to register themselves automatically to the domain.

As a subset of the registration process, nodes can be directed to a specific domain or load balancer.

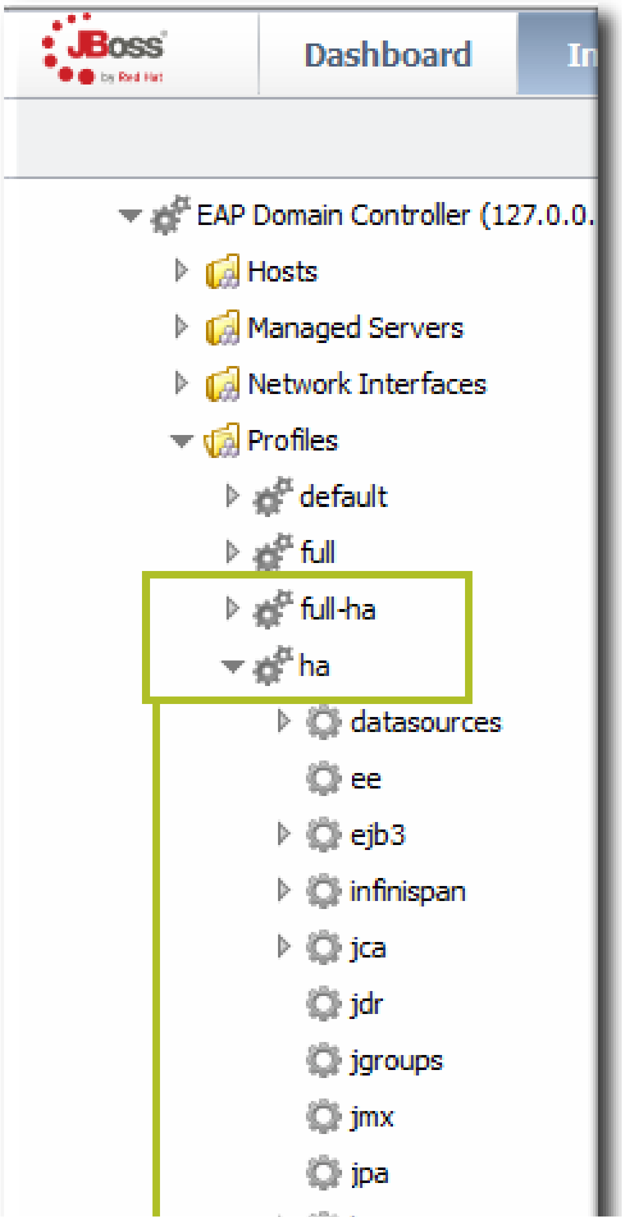
NOTE

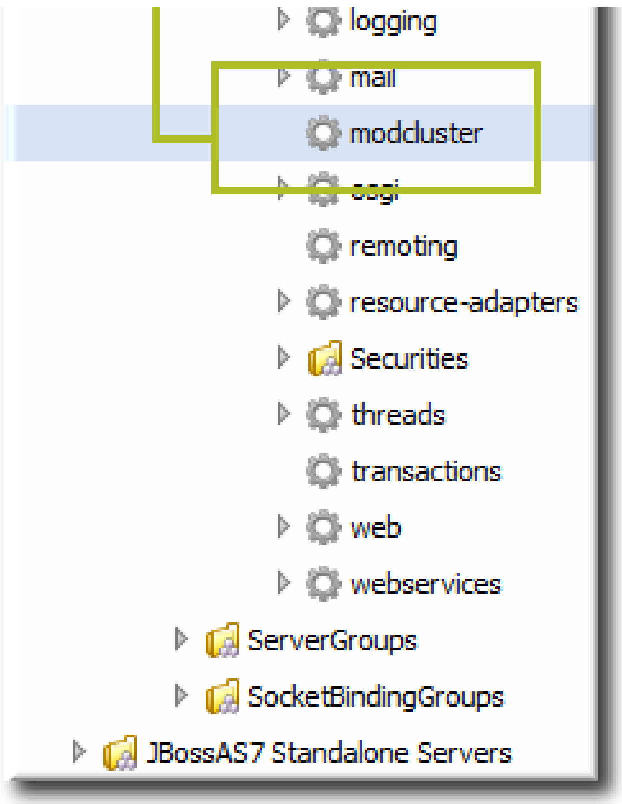
Multicast is configured by default.

1. Click the **Inventory** tab in the top menu.

2. Select **Servers - Top Level Imports** in the **Resources** menu table on the left. Then select the JBoss EAP 6 server which hosts the **mod_cluster** service (the master node).

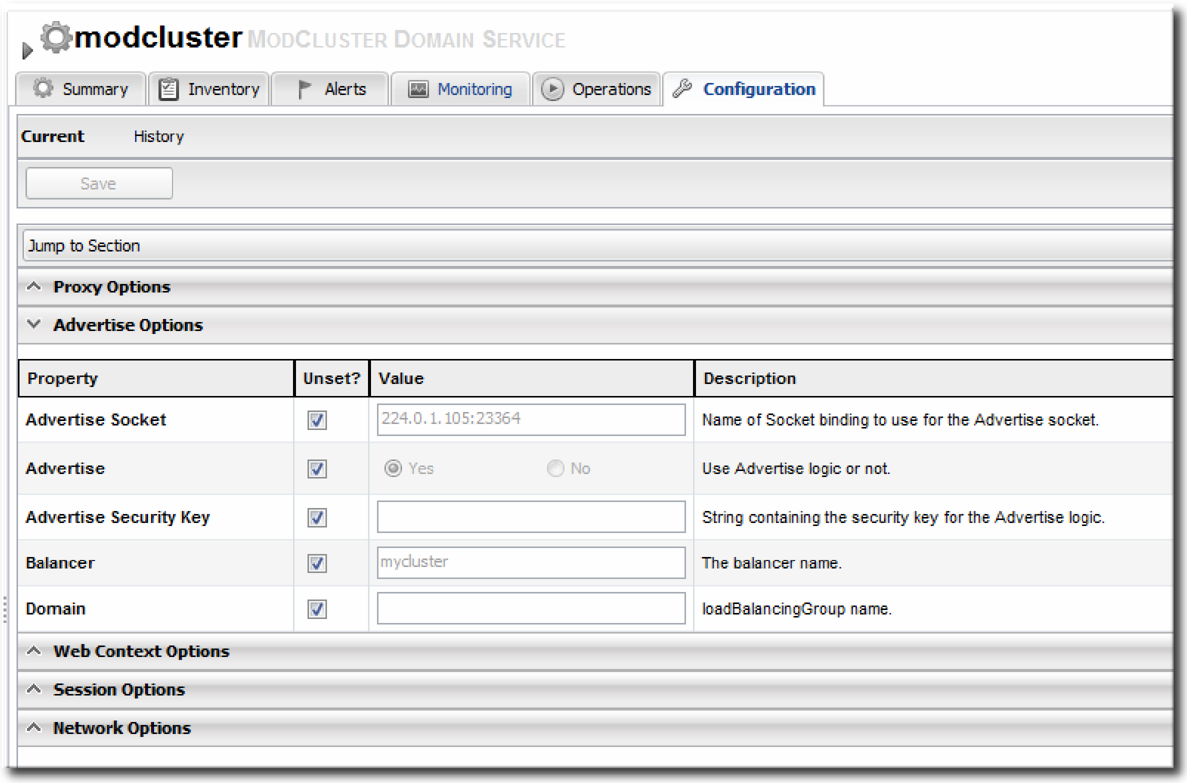
3. Navigate to the **mod_cluster** service resource.





4. Click the **`Configuration`** tab on the resource entry.

5. Go to the **`Advertise Options`** section.



Property	Unset?	Value	Description
Advertise Socket	✓	224.0.1.105:23364	Name of Socket binding to use for the Advertise socket.
Advertise	✓	[✓ Yes] No	Use Advertise logic or not.
Advertise Security Key	✓		String containing the security key for the Advertise logic.
Balancer	✓	mycluster	The balancer name.
Domain	✓		loadBalancingGroup name.

6. Change the multicast settings.

   To use a specific load balancer for the **`mod_cluster`** nodes, set the **`Balancer`** field to the load balancer name and, optionally, the **`Domain`** field to the load balancer group, which is one of the groups configured on the balancer itself.

IMPORTANT

The **Balancer** and **Domain** values must match the configuration for the corresponding Apache server.

7. Click the **Save** button at the top of the page.

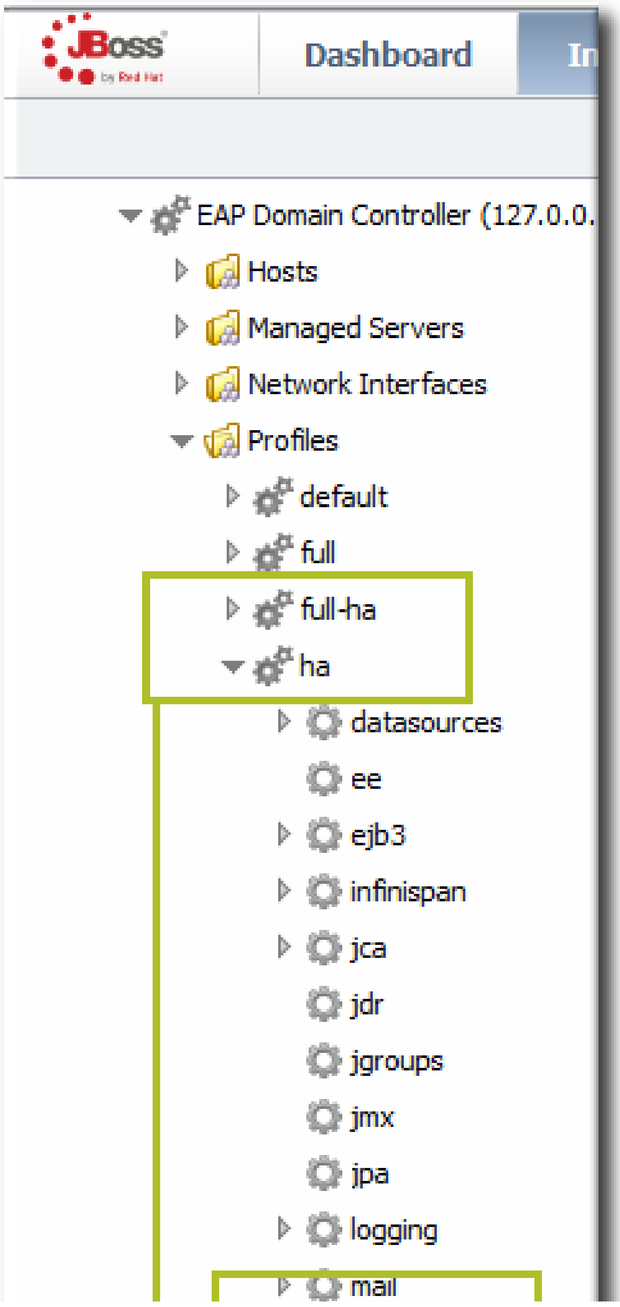
31.7.3. Excluding Web Contexts from Discovery

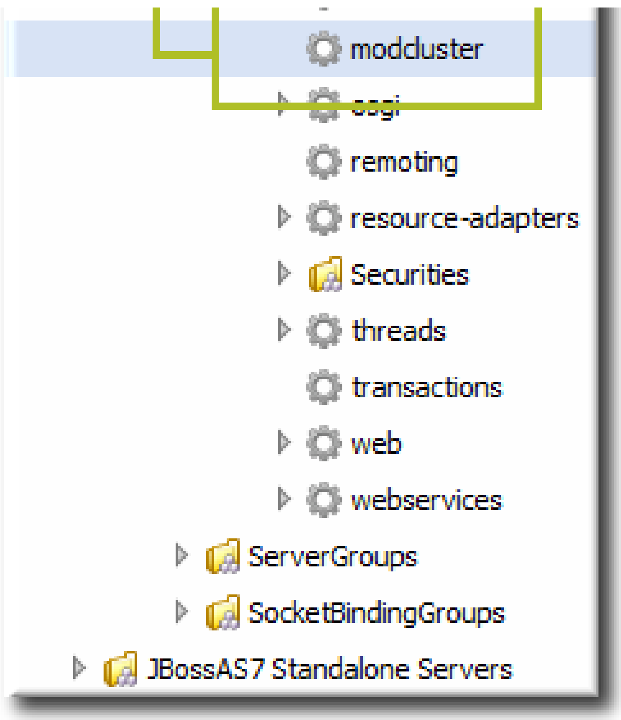
By default, any web context on a node is discovered and managed by the **mod_cluster** service. If a web context should not be included in the cluster, it can be excluded, by name.

1. Click the **Inventory** tab in the top menu.

2. Select **Servers - Top Level Imports** in the **Resources** menu table on the left. Then select the JBoss EAP 6 server which hosts the **mod_cluster** service (the master node).

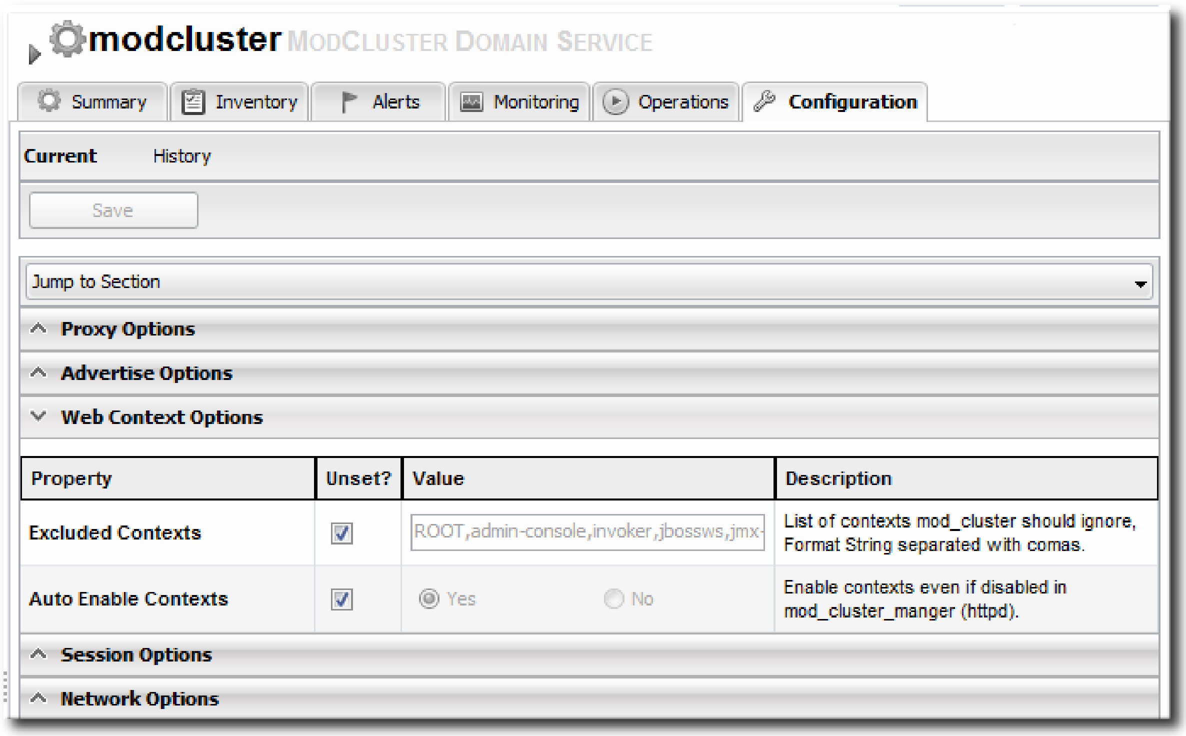
3. Navigate to the **mod_cluster** service resource.





4. Click the **`Configuration`** tab on the resource entry.

5. Go to the **`Web Context Options`** section.



Property	Unset?	Value	Description
Excluded Contexts	✓	ROOT,admin-console,invoker,jbossws,jmx-	List of contexts mod_cluster should ignore, Format String separated with comas.
Auto Enable Contexts	✓	[✓ Yes] No	Enable contexts even if disabled in mod_cluster_manger (httpd).

6. Unset the **`Excluded Contexts`** field, and add the names of any contexts to exclude.

NOTE

Some internal contexts for JBoss EAP are excluded by default. This should be kept in the excludes list; any new contexts to exclude should be added to the existing list.

7. Click the **Save** button at the top of the page.

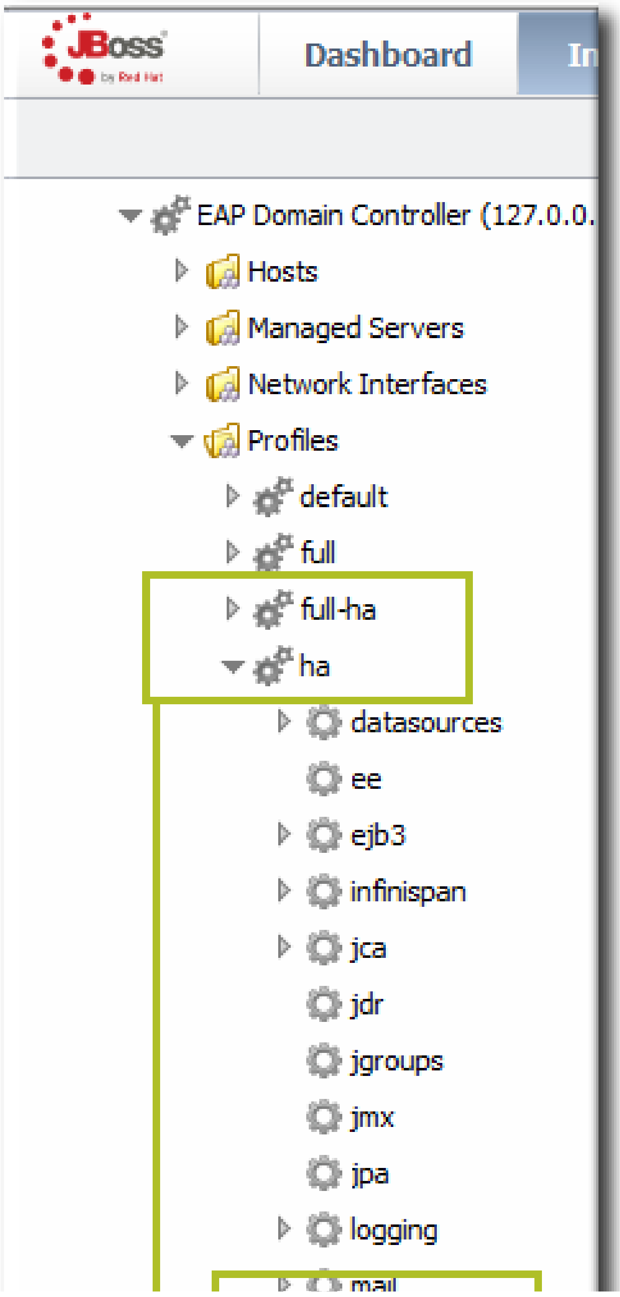
31.7.4. Configuring Web Context Metrics

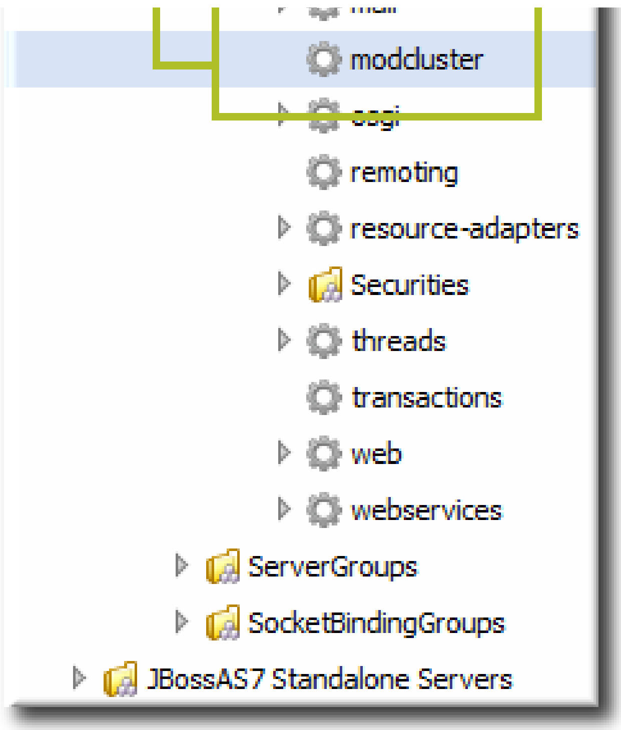
`mod_cluster` can use different types of metrics to determine how to balance traffic most efficiently. These metrics are described in the mod_cluster documentation. These metrics include things like active sessions, request counts, and traffic.

Additionally, custom metrics can be written and added to the subsystem.

These metrics can be added to the **`mod_cluster`** subsystem configuration for JBoss ON to monitor (and existing metrics can be removed).

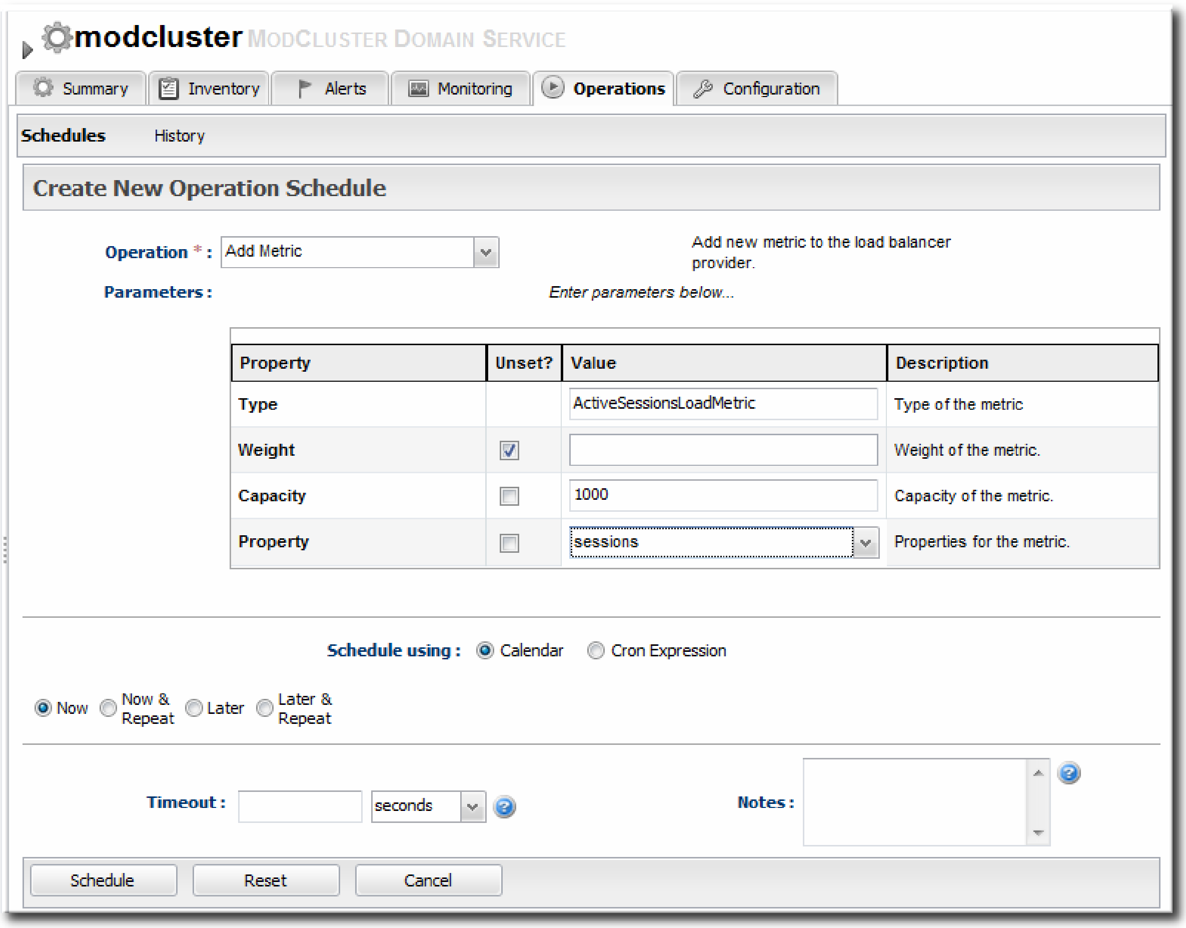
1. Click the **`Inventory`** tab in the top menu.
2. Select **`Servers - Top Level Imports`** in the **`Resources`** menu table on the left. Then select the JBoss EAP 6 server which hosts the **`mod_cluster`** service (the master node).
3. Navigate to the **`mod_cluster`** service resource.





4. Click the **`Operations`** tab on the resource entry.

5. Select the **`Add (Custom) Metrics`** operation from the drop-down menu.



Operation * : Add Metric — Add new metric to the load balancer provider.

Parameters : Enter parameters below...

Property	Unset?	Value	Description
Type		ActiveSessionsLoadMetric	Type of the metric
Weight	✓		Weight of the metric.
Capacity	☐	1000	Capacity of the metric.
Property	☐	sessions	Properties for the metric.

Schedule using : [✓ Calendar] Cron Expression

[✓ Now] Now & Repeat Later Later & Repeat

Timeout : seconds

Notes :

6. Fill in the metric information. The default metrics are listed in the mod_cluster documentation.

7. Click the **`Schedule`** button at the bottom of the page.

CHAPTER 32. MANAGING JBOSS EAP 5

32.1. DISCOVERING JBOSS EAP/AS 5 SERVERS

Most JBoss Enterprise Application Platforms are configured in a way that allows them and their child resources to be discovered and monitored without any additional configuration. When the agent attempts to discover and then manage resources, it uses certain assumptions based on common configuration to identify and connect to those resources. If for some reason the JBoss Enterprise Application Platform has configuration that differs from those assumptions, then JBoss ON may not be able to manage that system fully.

This section describes configuration settings that are required to discover or manage certain JBoss Enterprise Application Platform resource types.

32.1.1. Discovering and Managing the JBoss AS/EAP 5 JVM

For the JBoss ON agent to discover the JVM resource and its subsystems (such as memory, threading, and logging) for a JBoss EAP resource, the JBoss EAP instance must be configured to use the platform MBean server as the location where it registers its MBeans.

If necessary, edit the JBoss Enterprise Application Platform start script so that it uses the platform MBean server.

1. Open the **run.conf** file. For example, on Red Hat Enterprise Linux:

   ```
   [root@server ~]# vim jbossInstallDir/bin/run.conf
   ```

2. Add the **JAVA_OPTS** settings to use the platform MBean server.

   On Red Hat Enterprise Linux:

   ```
   JAVA_OPTS="$JAVA_OPTS -Djboss.platform.mbeanserver"
   ```

   On Windows:

   ```
   set JAVA_OPTS=%JAVA_OPTS% -Djboss.platform.mbeanserver
   ```

3. Restart the JBoss Enterprise Application Platform.

32.1.2. Enabling Remote Access to JMX and Profile Service

Management tasks like monitoring and running operations require that the agent is able to connect to the JMX server for the JBoss EAP/AS instance. This means that remote JMX access must be enabled.

1. Verify that the JBoss Naming Protocol server (JNP) is deployed. (It is deployed by default.)

   Open the **jboss-service.xml** file:

   ```
   [root@server ~]# vim jbossInstallDir/server/config/conf/jboss-
   service.xml
   ```

   Then, make sure that there is a line enabling the JNP connector.

```
<mbean code="org.jboss.naming.NamingService"> ... </mbean>
```

2. While not required, it is recommended that you enable authentication for the JNP service. (This is enabled by default.)

   1. Open the **jmx-invoker-service.xml** file.

      ```
      [root@server ~]# vim jbossInstallDir/server/config/deploy/jmx-
      invoker-service.xml
      ```

   2. Add a line for the JMX connector authentication.

      ```
      <interceptor
      code="org.jboss.jmx.connector.invoker.AuthenticationInterceptor"
                           securityDomain="java:/jaas/jmx-console"/>
      ```

   3. Make sure that the admin user is listed in the JMX users properties file. When a new JBoss EAP resource is discovered, the agent reads the JMX username and password from this file and stores them in the discovered JBoss EAP resource's connection settings. These settings are then used to connect to the EAP server's JNP service.

      ```
      [root@server ~]# vim
      jbossInstallDir/server/config/conf/props/jmx-console-
      users.properties
      ```

   4. Uncomment or add the user information. This is a simple key/value pair, *username=password*. For example:

      ```
      admin=admin
      ```

3. Restart the JBoss Enterprise Application Platform.

32.1.3. Setting Start Script Arguments, Environment Variables, and JAVA_OPTS

32.1.3.1. Start Script Discovery and Settings

As part of discovering a JBoss EAP 5 server, JBoss ON attempts to discover the environment that the server is running in. Specifically, the discovery process attempts to discover the runtime environment:

- The discovery process identifies, or attempts to identify, the start script that was used to start the EAP server, whether it was **run.sh|bat** or a custom start script.
- Discovery detects a subset of environment variables set in the **run.conf** file or parent process that are required for the start script to work.

NOTE

Although the discovery process does detect some environment variables, **the discovery scan does not JAVA_OPTS values.**

The connection properties for the start script intentionally defer to the **run.conf** file for **JAVA_OPTS** values.

- Discovery attempts to detect any arguments passed to the start script itself.

The discovered settings are stored in the `Start Script Environment Variables` and `Start Script Arguments` connection settings for the EAP 5 resource.

If the discovery scan cannot detect some of the script settings (such as the agent is running as a different user than the EAP 5 server and cannot read the parent process), it fails gracefully. It simply gathers whatever information it can and ignores blank values.

NOTE

The `Start Script Environment Variables` and `Start Script Arguments` connection settings are only discovered once, when the resource is initially discovered. After that, the connection settings can be changed in the connection settings configuration, but any changes made on the local system will not be detected. For example, if the server is discovered with a particular value (like `-XX:PermSize=256M`), the argument value will not be updated if the server is restarted later with a different setting value.

The script arguments and environment variables are the only ones used by the agent when it runs the start script. These arguments and environment variables are not added to other configuration settings or the parent process. The start script settings in the EAP 5 connection settings are deterministic.

In previous versions of the plug-in, the EAP/AS 5 server had a Java home directory setting and bind address (hostname or IP) that was used by the agent as part of identifying and running the start scripts.

If the `Script Arguments` field is empty, then the bind address and Java home directory [7] are still used for invoking the start script. If *any* script arguments are set, then the bind address and Java home values are ignored.

The script arguments value overrides any other start script connection settings.

32.1.3.2. Start Script Arguments and Drift Monitoring

JBoss ON can monitor directories or specific files to check for *configuration drift*, which means configuration changes that move away from a designated configuration state. This is described in more detail in the drift chapters of "Managing Resource Configuration."

While drift monitoring is critical for administrators to manage important resources, there may be times when it is necessary or beneficial from that configuration in a few settings. The start script arguments can be used to override the defined configuration for an EAP 5 server *without* triggering a drift alert.

Since the start script options are all connection settings, every change is recorded in a change history and can be easily viewed and reverted. This keeps the system properties and Java settings trackable and remediable.

> **Example 32.1. System Properties Without Violating the Drift Definition**
>
> Tim the IT Guy creates one specific set of configuration that all production EAP 5 servers should use an environment. He then creates a drift definition template for monitoring and associates or pins that blessed configuration to the template.
>
> Every EAP 5 server which uses that drift template must conform to those configuration settings. Tim the IT Guy creates an alert that informs him if any production server drifts away from that

specified configuration. This is an important safety measure for Example Co.'s production application servers.

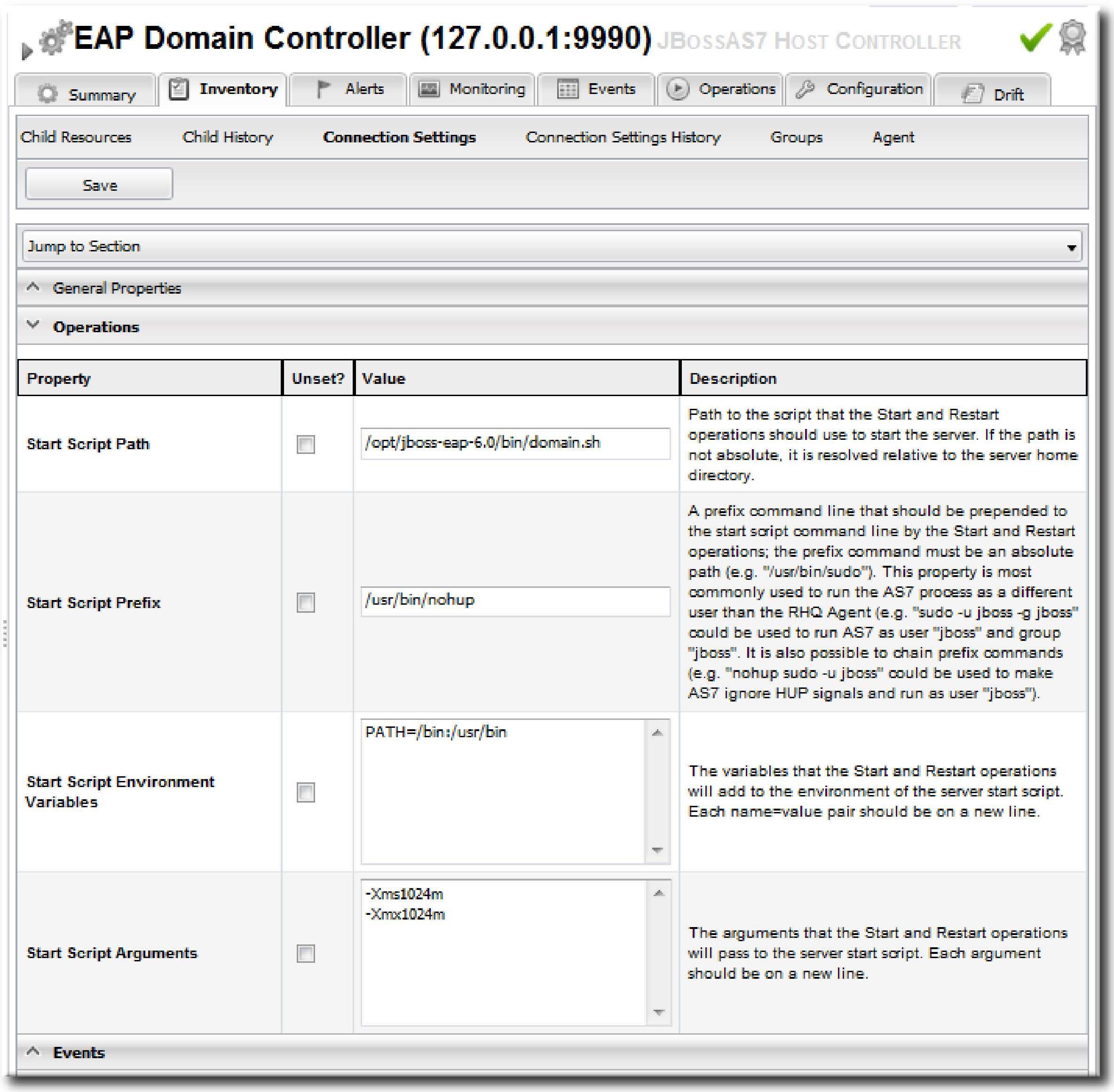
However, a couple of the production servers have slightly different hardware and other applications running on them, so they require different heap sizes to run effectively.

If Tim the TI Guy adds a system property to use a different heap size, he is going to receive constant drift alerts or his edited configuration could be overwritten if he runs an automatic server-side script to remediate drift configuration.

By setting different heap settings through the start script, Tim can apply the right settings for that system without editing a configuration file, so there is no alert-able configuration drift.

32.1.3.3. Changing Start Script Configuration

1. Click the `Inventory` tab in the top menu.
2. Select `Servers - Top Level Imports` in the `Resources` menu table on the left. Select the JBoss EAP 5 server.
3. In the inventory tree, select the top resource entry for the server.
4. Open the `Inventory` tab, and select the `Connection Settings` subtab.
5. Expand the `Operations` section.
6. Change or add start script settings. These are the scripts and settings that the JBoss ON agent uses when running a start or restart operation on the EAP 5 server.



Property	Unset?	Value	Description
Start Script Path	☐	/opt/jboss-eap-6.0/bin/domain.sh	Path to the script that the Start and Restart operations should use to start the server. If the path is not absolute, it is resolved relative to the server home directory.
Start Script Prefix	☐	/usr/bin/nohup	A prefix command line that should be prepended to the start script command line by the Start and Restart operations; the prefix command must be an absolute path (e.g. "/usr/bin/sudo"). This property is most commonly used to run the AS7 process as a different user than the RHQ Agent (e.g. "sudo -u jboss -g jboss" could be used to run AS7 as user "jboss" and group "jboss". It is also possible to chain prefix commands (e.g. "nohup sudo -u jboss" could be used to make AS7 ignore HUP signals and run as user "jboss").
Start Script Environment Variables	☐	PATH=/bin:/usr/bin	The variables that the Start and Restart operations will add to the environment of the server start script. Each name=value pair should be on a new line.
Start Script Arguments	☐	-Xms1024m -Xmx1024m	The arguments that the Start and Restart operations will pass to the server start script. Each argument should be on a new line.

- To use a custom start script, enter the path and script name.
- Optionally, enter a prefix to use with the script when running the start script.
- Set any environment variables, one per line.
- Set any script arguments, one per line. For regular JAVA_OPTS, these arguments usually are **-X**, **-D**, or **-P**. Some useful **-XX** arguments are listed in the JVM options documentation from Sun.

  The EAP 5 default start scripts use a `run.sh`-style script, so the arguments use that format. A custom script can use different arguments or options.

7. Click the **Save** button at the top of the page.

32.2. CREATING JBOSS EAP 5 CHILD RESOURCES

32.2.1. Creating Data Sources

NOTE

It can take several minutes for the new child resource to be added and visible in the JBoss ON inventory because the new resource has to be created on the agent platform and then discovered by the agent.

1. Search for the JBoss server instance to which to deploy the data source.
2. On the details page for the selected JBoss server instance, open the `Inventory` tab.
3. In the `Create New` drop-down menu, select the item for **- Data Sources**.
4. Select a template for the data source. There are three data sources templates to populate a data source with common information:
   - The default template is used with SQL databases like PostgreSQL or MySQL.
   - The Oracle Local TX is used for Oracle databases with local transactions.
   - The Oracle XA template is used for Oracle databases with XA transactions.
5. Along with the obvious settings, like the resource name, enter the information for the specific child resource to be deployed:
   - The type of data source to create, either No TX Data Sources, Local TX Data Sources or XA Data Sources
   - A unique JNDI name for the DataSource wrapper to use to bind under
   - The fully qualified name of the JDBC driver or DataSource class, such as `org.postgresql.Driver`
   - The JDBC driver connection URL string, such as `jdbc:postgresql://127.0.0.1:5432/foo`
   - The username and password to use to connect to the data source
   - The minimum and maximum connection pool sizes for this data source

   Additional settings are available under the `Advanced Settings` area.

32.2.2. Creating Connection Factories

NOTE

It can take several minutes for the new child resource to be added and visible in the JBoss ON inventory because the new resource has to be created on the agent platform and then discovered by the agent.

1. Search for the JBoss server instance to which to deploy the connection factory.
2. On the details page for the selected JBoss server instance, open the `Inventory` tab.
3. In the `Create New` drop-down menu, select the item for **- Connection Factory**.

4. Along with the obvious settings, like the resource name, enter the information for the specific child resource to be deployed:
   - The type of connection factory to create, either tx-connection-factory (transaction) or no-tx-connection-factory (no transaction)
   - A unique JNDI name for the DataSource wrapper to use to bind under
   - The username and password to use to connect to the data source
   - The minimum and maximum connection pool sizes for this data source

   Additional settings are available under the **`Advanced Settings`** area.

32.2.3. Creating JMS Queues and Topics

JMS Queues and JMS Topics are child resources of a JBossMQ service or JBossMessaging service resource, which itself is a child of a JBoss server instance.

NOTE

It can take several minutes for the new child resource to be added and visible in the JBoss ON inventory because the new resource has to be created on the agent platform and then discovered by the agent.

1. Search for the JBoss messaging service to which to deploy the JMS queue or topic.
2. On the details page for the selected JBoss messaging service, open the **`Inventory`** tab.
3. In the **`Create New`** drop-down menu, select the **- JMQ JMS Topic** or **- JMQ JMS Queue** item.
4. Aside from the obvious settings, like the resource name, the JMS Queue or JMS Topic entry requires two additional parameters:
   - The name of the queue or topic to use as the JMX object name
   - A unique JNDI name for the DataSource wrapper to use to bind under

   Additional settings are available under the **`Advanced Settings`** area.

32.3. DEPLOYING APPLICATIONS

Applications that are deployed on a JBoss server are a special type of resource. They have a place in the JBoss ON inventory and are children of the JBoss AS/EAP server. In that way, they can be discovered or created as a resource. However, some JBoss child resources — EARs and WARs — are also content, so they are treated as software packages that are stored and then deployed to JBoss servers. For these applications, the child resource is created first, by uploading a package to the JBoss server. After that, they are managed like content, with updated packages added to a content repo and then applied to the JBoss server.

32.3.1. Space Considerations for Deploying Applications

Deploying content-backed resources can have a significant impact on disk space requirements.

JBoss ON stores all versions of content. This is part of versioning control, allowing changes to content-backed resources to be reverted and managed and for different versions to be deployed at different times.

Therefore, the system which hosts the backend database (Oracle or PostgreSQL) must have enough disk space to store all versions of all bundles. Additionally, the database itself must have adequate tablespace for the content.

When calculating the required amount of space, estimate the size of every artifact, and then the number of versions for each artifact. At a minimum, **have twice that amount of space available**; both PostgreSQL and Oracle require twice the database size to perform cleanup operations like vacuum, compression, and backup and recovery.

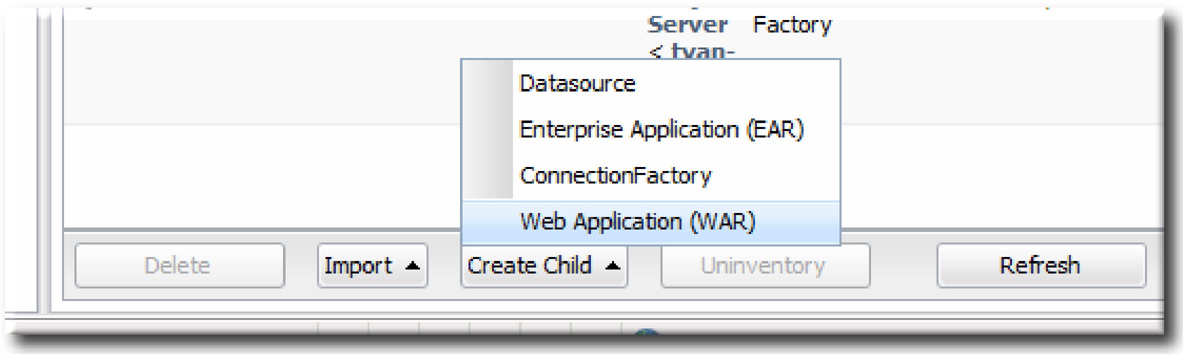
32.3.2. Deploying EAR and WAR Files

IMPORTANT

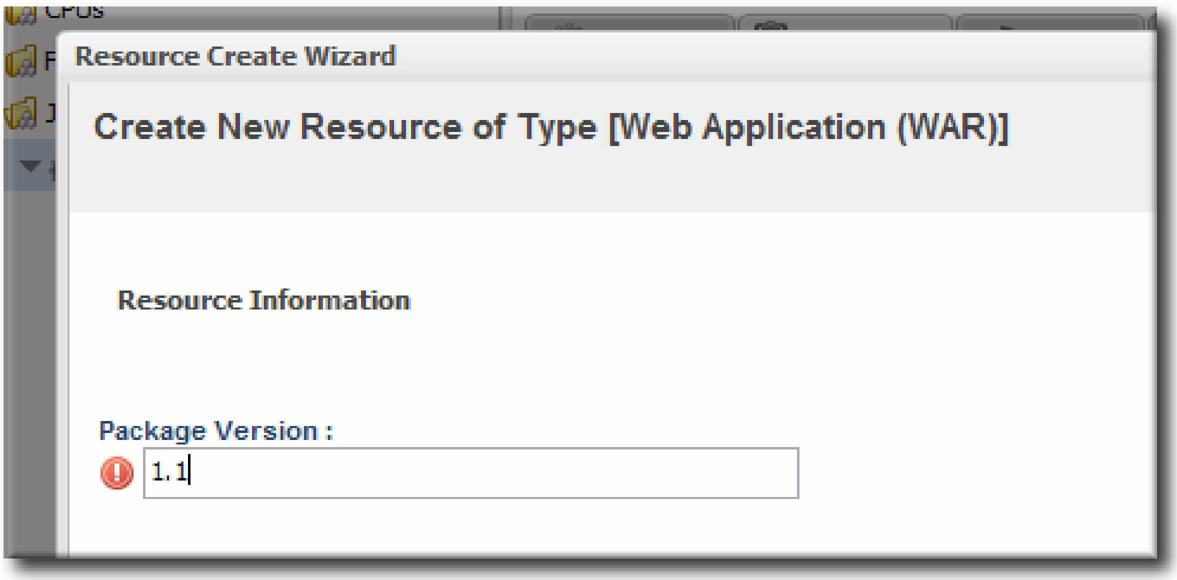
Newly-deployed content may not show up in the JBoss ON inventory for as long as 24 hours, even if it was successfully created. By default, discovery scans for services are only made every 24 hours.

To see it immediately, run an *execute prompt command*operation on the agent and enter the **`discovery`** command. This runs a discovery scan.

1. Search for the JBoss server instance to which to deploy the EAR or WAR.
2. On the details page for the selected JBoss server instance, open the **`Inventory`** tab.
3. In the **`Create New`** menu at the bottom, select the item for **- Web Application (WAR)** or **- Enterprise Application (EAR)**, as appropriate.



4. Enter the version number.



This is not used for the resource. The actual version number is calculated based on the spec version and implementation version in **MANIFEST.MF**, if any are given, or the calculated SHA-256 value for the package itself:

```
SPEC(IMPLEMENTATION)[sha256=abcd1234]
```

If no version numbers are defined in **MANIFEST.MF**, then the SHA value is used. The SHA value is always used to identify the package version internally.

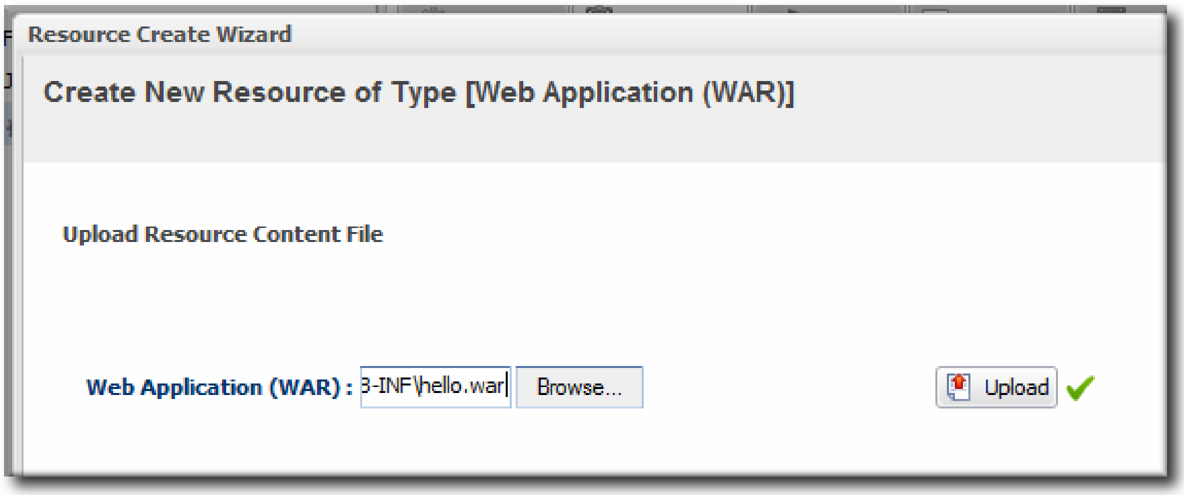
NOTE

When the EAR or WAR file is exploded after it is deployed, the **MANIFEST.MF** file is updated to include the calculated SHA version number. For example:

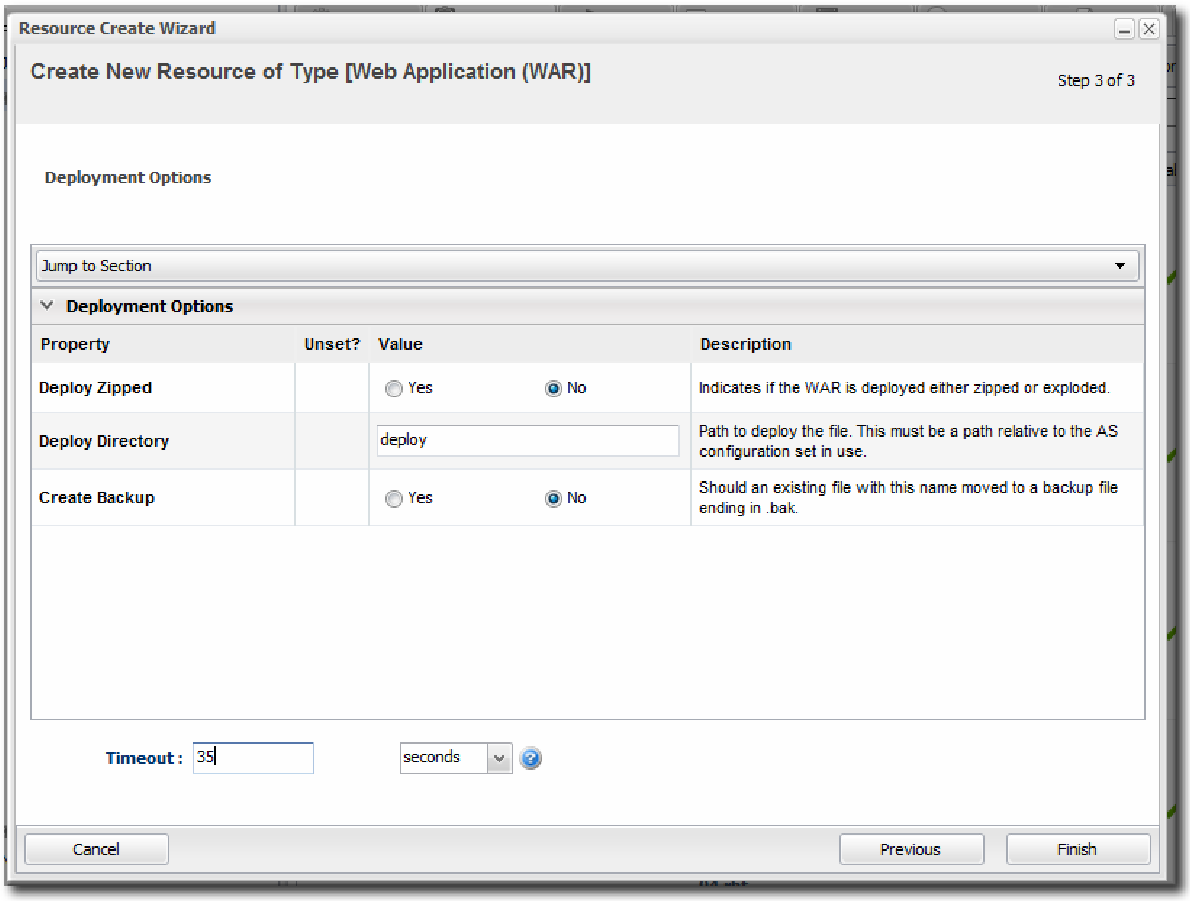
```
Manifest-Version: 1.0
Created-By: Apache Maven
RHQ-Sha256: 570f196c4a1025a717269d16d11d6f37
...
```

For more information on package versioning, see "Deploying Applications and Content".

5. Upload the EAR/WAR file.


Resource Create Wizard
Create New Resource of Type [Web Application (WAR)]
Upload Resource Content File
Web Application (WAR) :
3-INF\hello.war
Browse...
Upload

6. Enter the information for the application to be deployed.

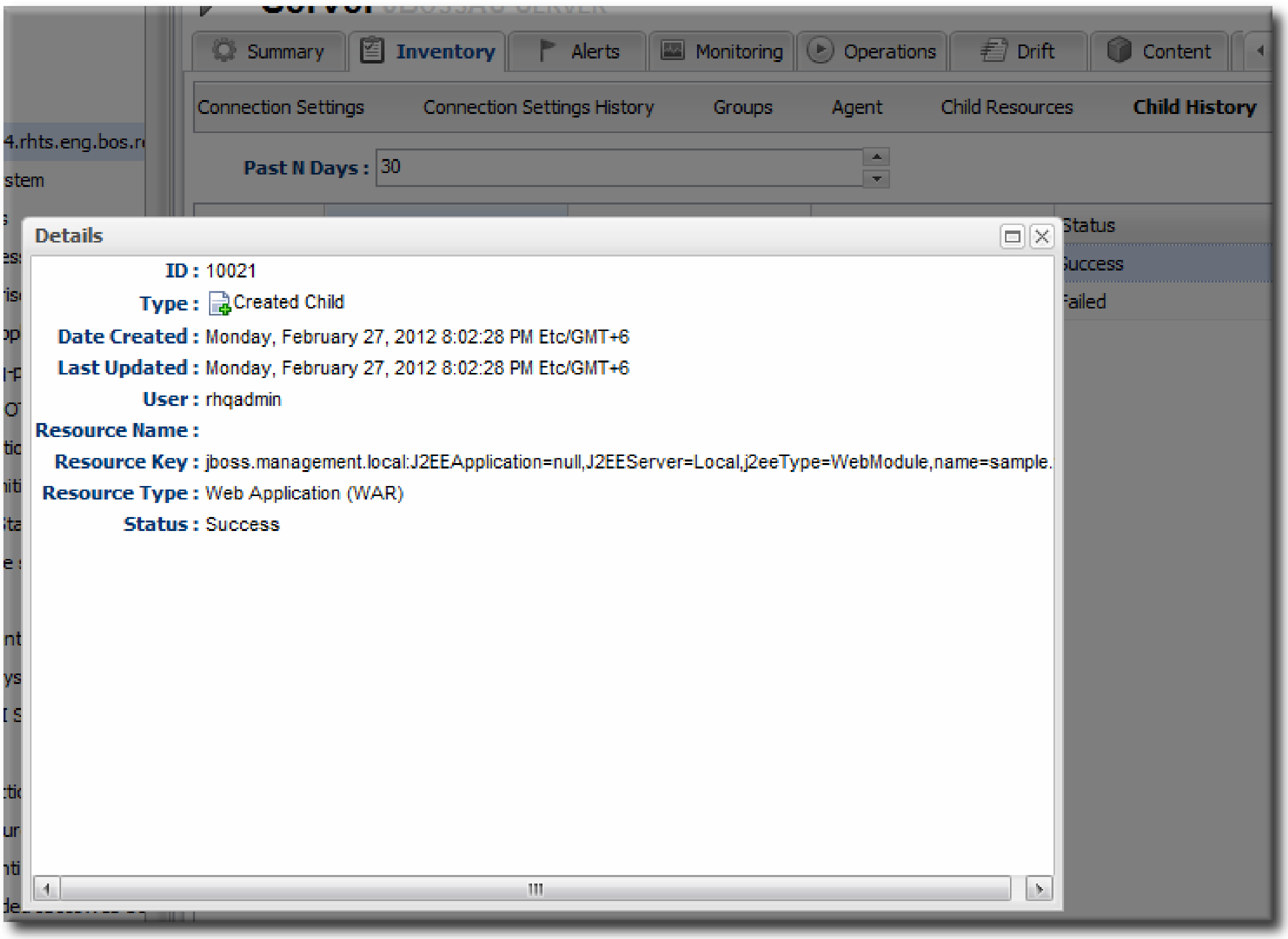


Property	Unset?	Value	Description
Deploy Zipped		Yes [✓ No]	Indicates if the WAR is deployed either zipped or exploded.
Deploy Directory		deploy	Path to deploy the file. This must be a path relative to the AS configuration set in use.
Create Backup		Yes [✓ No]	Should an existing file with this name moved to a backup file ending in .bak.

Timeout : 35 seconds

- Whether the file should be exploded (unzipped) when it is deployed.
- The path to the directory to which to deploy the EAR or WAR package. The destination directory is relative to the JBoss server instance installation directory; this cannot contain an absolute path or go up a parent directory.
- Whether to back up any existing file with the same name in the target directory.

Once the EAR/WAR file is confirmed, the new child resource is listed in the **`Child History`** subtab of the **`Inventory`** tab.



ID : 10021
Type : Created Child
Date Created : Monday, February 27, 2012 8:02:28 PM Etc/GMT+6
Last Updated : Monday, February 27, 2012 8:02:28 PM Etc/GMT+6
User : rhqadmin
Resource Name :
Resource Key : jboss.management.local:J2EEApplication=null,J2EEServer=Local,j2eeType=WebModule,name=sample.
Resource Type : Web Application (WAR)
Status : Success

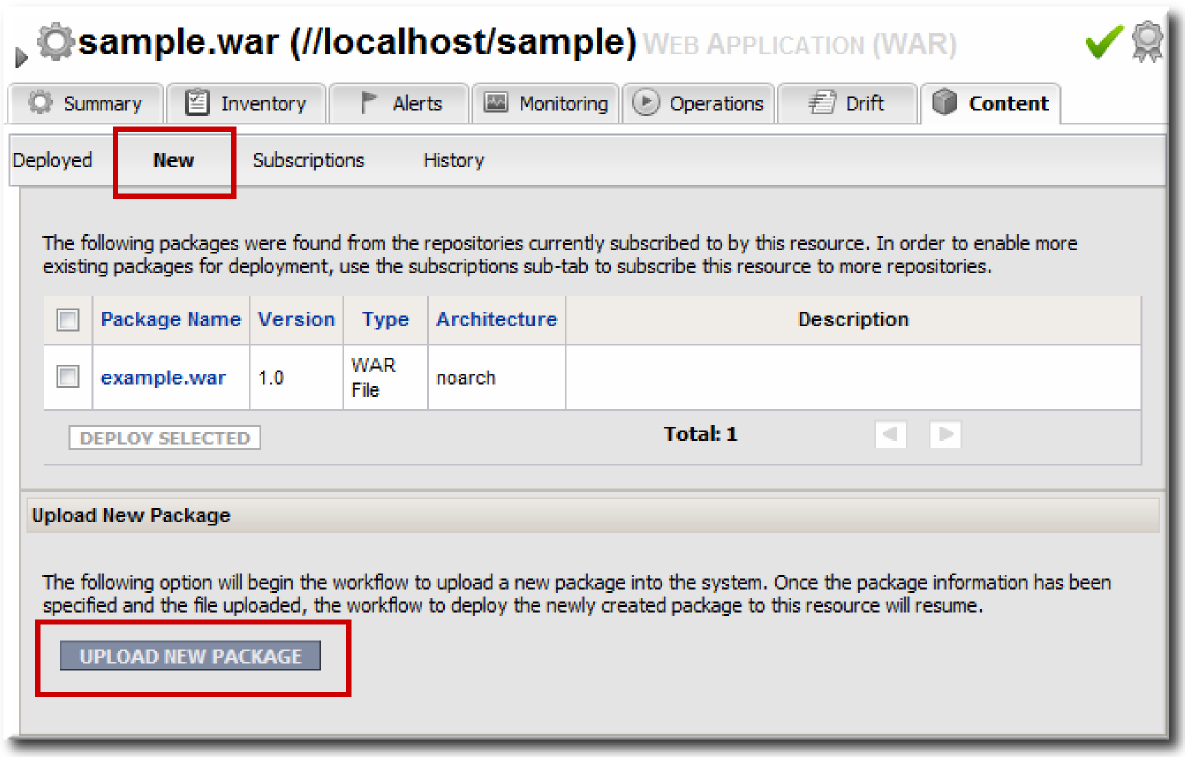
Figure 32.1. WAR Child Resource

32.3.3. Updating Applications

After the EAR or WAR resource is created, changes are treated like updated content packages. Updating the EAR/WAR resource is the same as uploading and applying new packages to that EAR/WAR resource entry.

1. Browse to the EAR or WAR resource in the JBoss ON UI.

2. In the EAR or WAR resource details page, open the `Content` tab, and click the `New` subtab.



The following packages were found from the repositories currently subscribed to by this resource. In order to enable more existing packages for deployment, use the subscriptions sub-tab to subscribe this resource to more repositories.

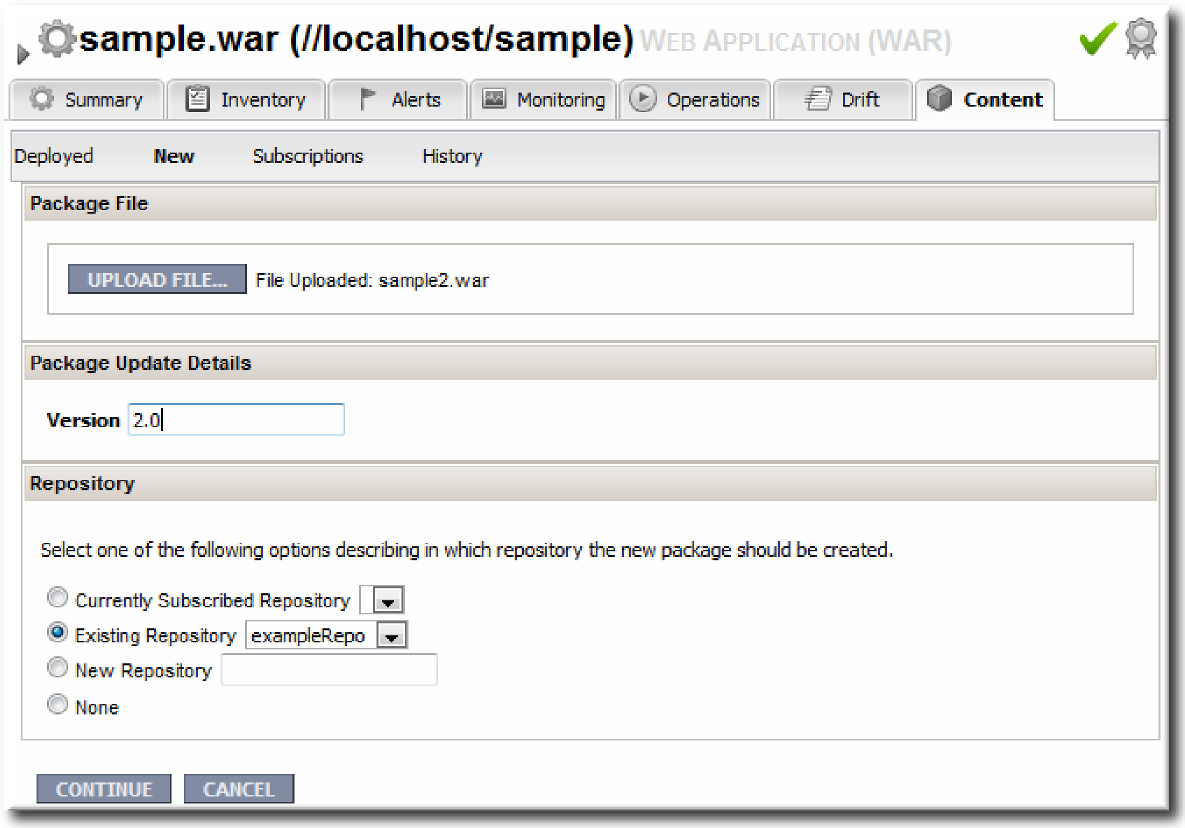
☐	Package Name	Version	Type	Architecture	Description
☐	example.war	1.0	WAR File	noarch	

Upload New Package

The following option will begin the workflow to upload a new package into the system. Once the package information has been specified and the file uploaded, the workflow to deploy the newly created package to this resource will resume.

3. Click the **UPLOAD NEW PACKAGE** button.

4. Click the **UPLOAD FILE** button.



File Uploaded: sample2.war

Package Update Details

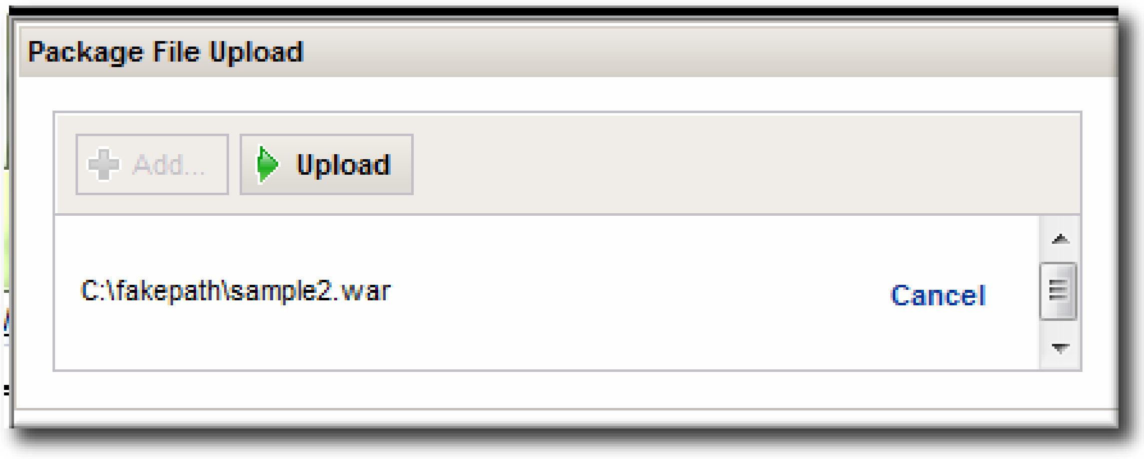
Version 2.0

Repository

Select one of the following options describing in which repository the new package should be created.

( ) Currently Subscribed Repository
(●) Existing Repository exampleRepo
( ) New Repository
( ) None

5. In the pop-up window, click the **Add** button, and browse the local filesystem to the updated WAR or EAR file to be uploaded.



C:\fakepath\sample2.war

6. Click the **UPLOAD** button to load the file and dismiss the window.

7. In the main form, select the repository where the WAR or EAR file package should be stored. If one exists, select an existing repository or a subscribed repository for the resource. Otherwise, create a new repository.

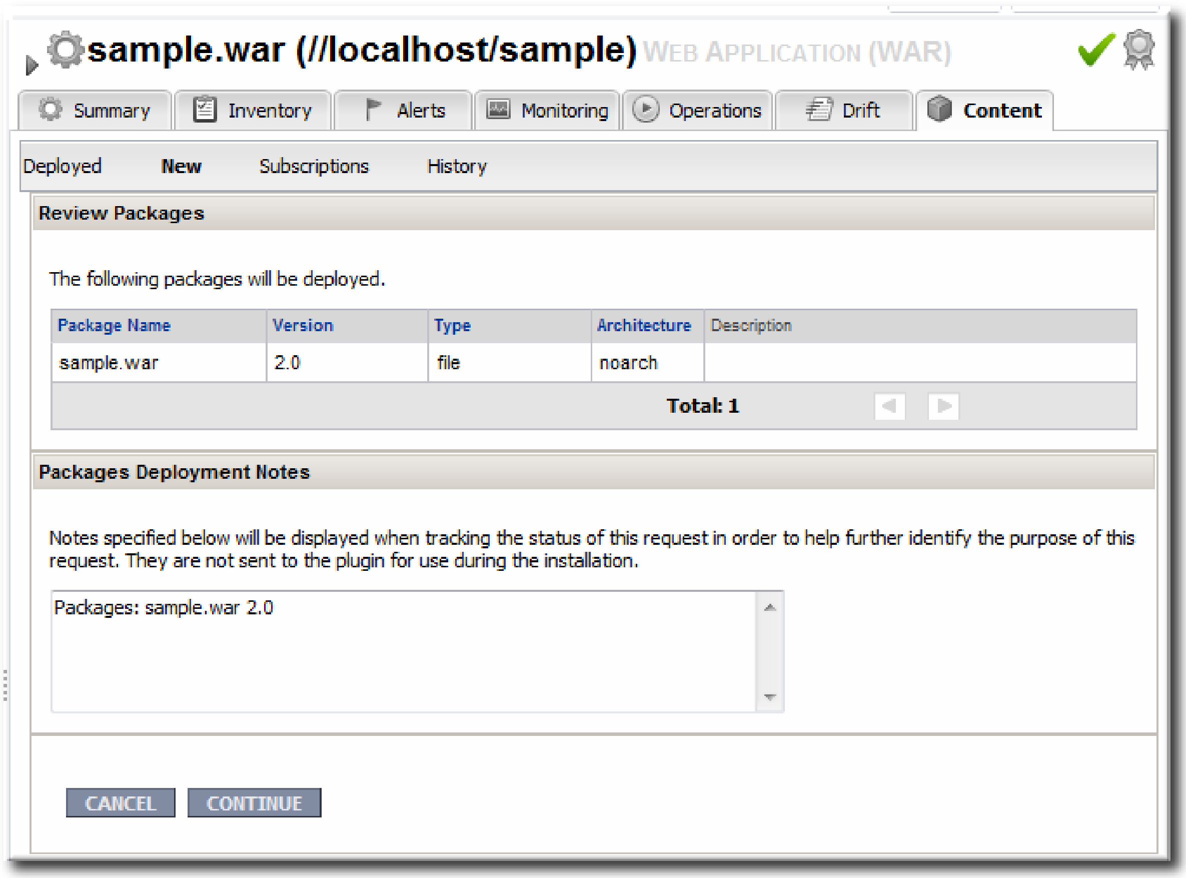
8. Optionally, set the version number for the EAR/WAR package.

   If this is set, then this value is displayed in the UI. If not, then a version number is calculated, based on the spec version and implementation version in **MANIFEST.MF**, if any are given, or the calculated SHA-256 value for the package itself. Internally, the package is identified by the SHA value.

   ```
   SPEC(IMPLEMENTATION)[sha256=abcd1234]
   ```

   For more information on package versioning, see "Deploying Applications and Content".

9. Confirm the details for the new package, then click **CONTINUE**.



Review Packages

The following packages will be deployed.

Package Name	Version	Type	Architecture	Description
sample.war	2.0	file	noarch	

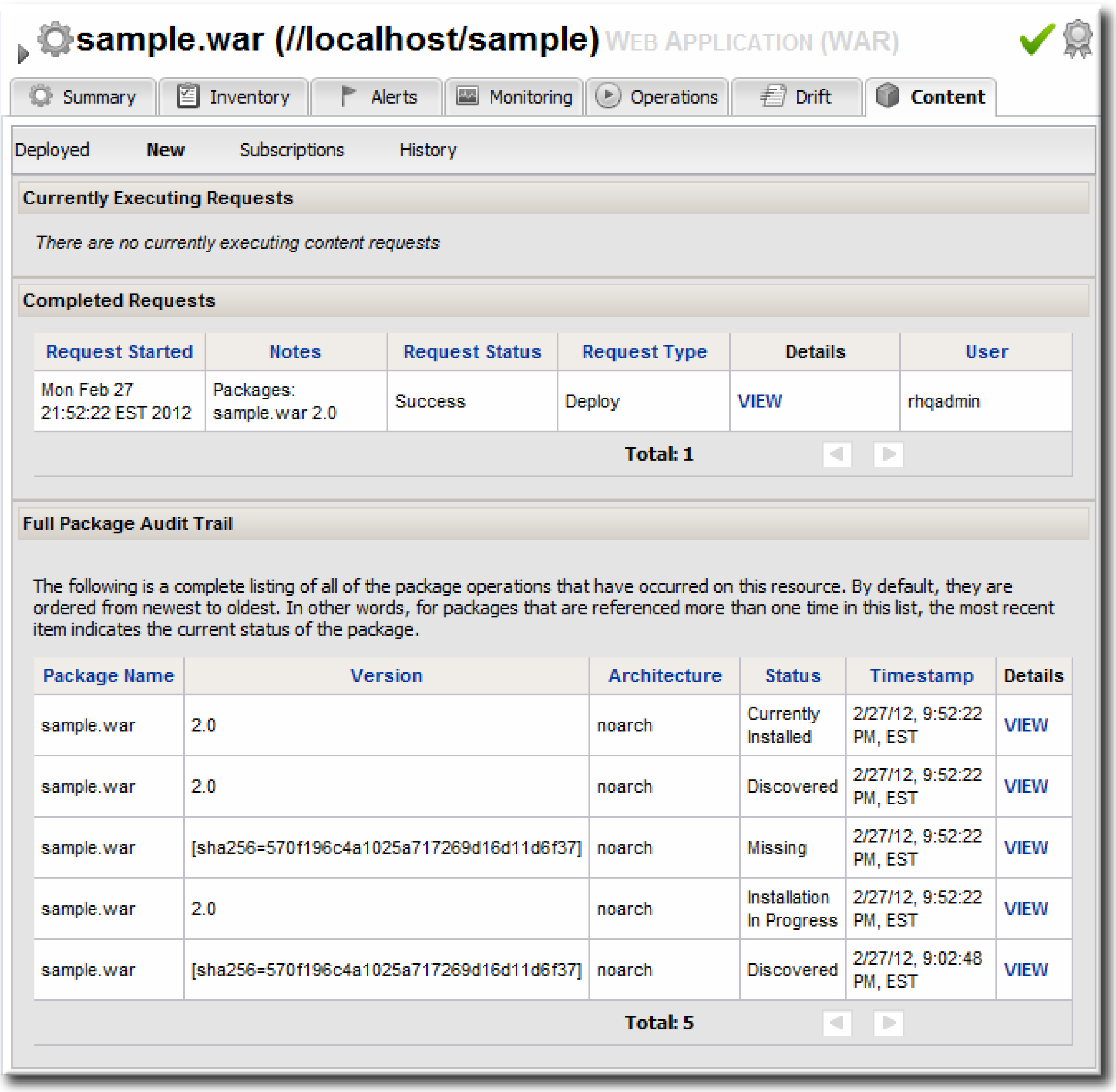
Total: 1

Packages Deployment Notes

Notes specified below will be displayed when tracking the status of this request in order to help further identify the purpose of this request. They are not sent to the plugin for use during the installation.

Packages: sample.war 2.0

When the package is successfully uploaded, the UI redirects to the history page on the **`Content`** tab.



Currently Executing Requests

There are no currently executing content requests

Completed Requests

Request Started	Notes	Request Status	Request Type	Details	User
Mon Feb 27 21:52:22 EST 2012	Packages: sample.war 2.0	Success	Deploy	VIEW	rhqadmin

Total: 1

Full Package Audit Trail

The following is a complete listing of all of the package operations that have occurred on this resource. By default, they are ordered from newest to oldest. In other words, for packages that are referenced more than one time in this list, the most recent item indicates the current status of the package.

Package Name	Version	Architecture	Status	Timestamp	Details
sample.war	2.0	noarch	Currently Installed	2/27/12, 9:52:22 PM, EST	VIEW
sample.war	2.0	noarch	Discovered	2/27/12, 9:52:22 PM, EST	VIEW
sample.war	[sha256=570f196c4a1025a717269d16d11d6f37]	noarch	Missing	2/27/12, 9:52:22 PM, EST	VIEW
sample.war	2.0	noarch	Installation In Progress	2/27/12, 9:52:22 PM, EST	VIEW
sample.war	[sha256=570f196c4a1025a717269d16d11d6f37]	noarch	Discovered	2/27/12, 9:02:48 PM, EST	VIEW

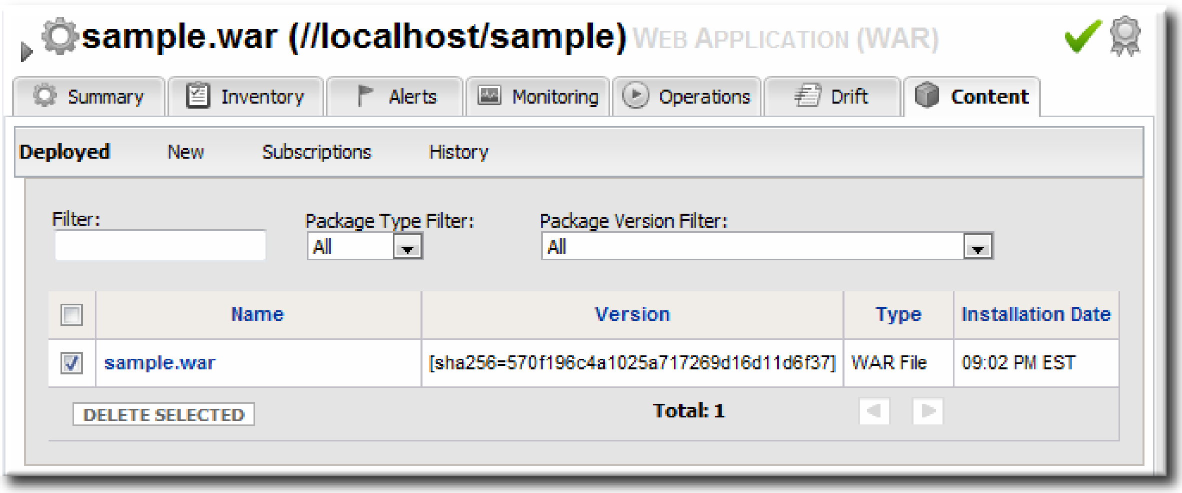
Total: 5

Figure 32.2. Deployment History for a Resource

32.3.4. Deleting an Application

Deleting an EAR/WAR application is the same as deleting the currently deployed package associated with that EAR/WAR resource entry.

1. Browse to the EAR or WAR resource in the JBoss ON UI.
2. In the EAR or WAR resource details page, open the **`Content`** tab, and click the **`Deployed`** subtab.
3. Select the checkbox by the EAR/WAR package, and click the **DELETE SELECTED** button.



Filter:

Package Type Filter: All

Package Version Filter: All

	Name	Version	Type	Installation Date
✓	sample.war	[sha256=570f196c4a1025a717269d16d11d6f37]	WAR File	09:02 PM EST

Total: 1

32.4. APPLYING JBOSS PATCHES FROM THE PATCH RSS FEED

Content management can be used to apply cumulative patches to JBoss EAP 4. By default, the JBoss ON server comes pre-configured with the JBoss Patch Content Source and JBoss Patch Repository.

NOTE

Patch updates using the Patch RSS Feed is not supported for JBoss Enterprise Application platform 5 or newer.

32.4.1. Planning How to Patch JBoss Resources

JBoss ON can automatically apply patches as they are available in the JBoss Customer Portal (CP). When a patch is released, an RSS feed on the Customer Portal is updated, and JBoss ON monitors that feed and lists the latest patches in the GUI.

There are two parts to planning how to apply patches:

- Identifying and configuring content sources
- Planning how to execute manual steps

32.4.1.1. Identifying Content Sources

A *content source* connects a JBoss ON server to some kind of content delivery mechanism. With JBoss products, the content source is the Customer Portal, which contains all patches for all JBoss products. (A content source can be another content repository, depending on your products and environment.)

A content source identifies a location. A *repository* within JBoss ON maps a content source to the resources in the inventory that the patches are applied to.

For JBoss patches, the default content provider connects the JBoss ON server to the cumulative patches provided by the JBoss Customer Service Portal. The default repository associated with the content provider is where the metadata about the patches and the patches themselves are stored within JBoss ON.

The JBoss ON agent is the entity which actually executes the patching process on a resource. The agent is informed of updates, pulls the information from the server, and then updates the local JAR and class files within the managed JBoss instance. The patching process runs independently of any server configuration profile or base configuration.

JBoss Enterprise Application Platform 4 can receive and apply patch updates through the JBoss Customer Portal feed in JBoss ON.

NOTE

Patch updates using the Patch RSS Feed is not supported for JBoss Enterprise Application platform 5 or newer.

NOTE

A Customer Portal feed is only available for a product or a specific version of a product if there is a patch in the Customer Portal for that product. JBoss ON depends on the JBoss Customer Portal to provide patch information.

32.4.1.2. Planning Manual Steps

Some patches require additional, manual changes, such as editing an XML configuration file. There are several different situations that require manual intervention:

- The file to be patched is not present in the configuration.
- There are files that need to be removed manually.
- Configuration files, such as XML or Java properties files, require patches that need to be applied manually.
- Seam is being used and must be patched manually.

Basically, admin intervention is required to resolve anything that is outside the default configuration, like merging in custom configuration or updating custom libraries.

JBoss ON performs the standard steps required to apply patches to a JBoss instance, but it does not (and should not) have any way to parse and then merge changes in the configuration. JBoss ON does not attempt to determine, value, and apply custom changes. That sort of heuristic is best performed by an administrator.

Any manual steps which are required to complete the patch are listed in the content update summary after the patch is applied.

32.4.2. Enabling the Default JBoss Patch Content Source

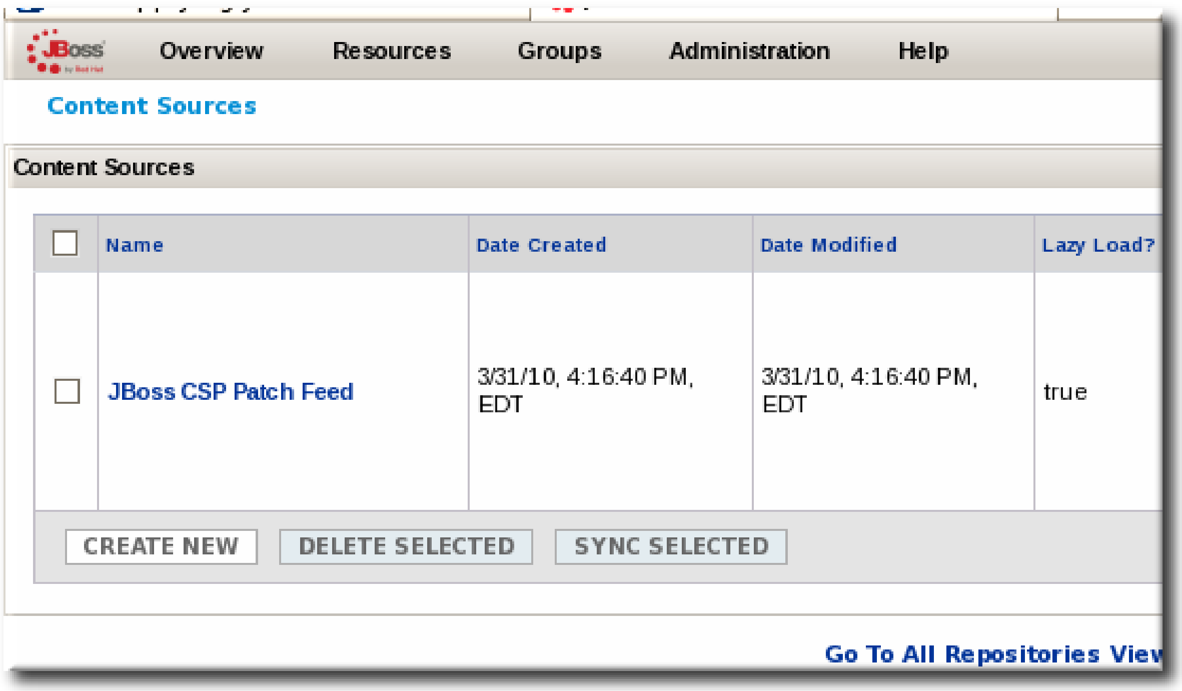
The patch stream is supported for EAP 4 servers, providing patches and content updates.

NOTE

Patch updates using the Patch RSS Feed is not supported for JBoss Enterprise Application platform 5 or newer.

1. In the `Administration` tab in the top menu, open the `Content` menu and select the `Content Sources` item.

2. Click the `JBoss Customer Portal Patch Feed` source.

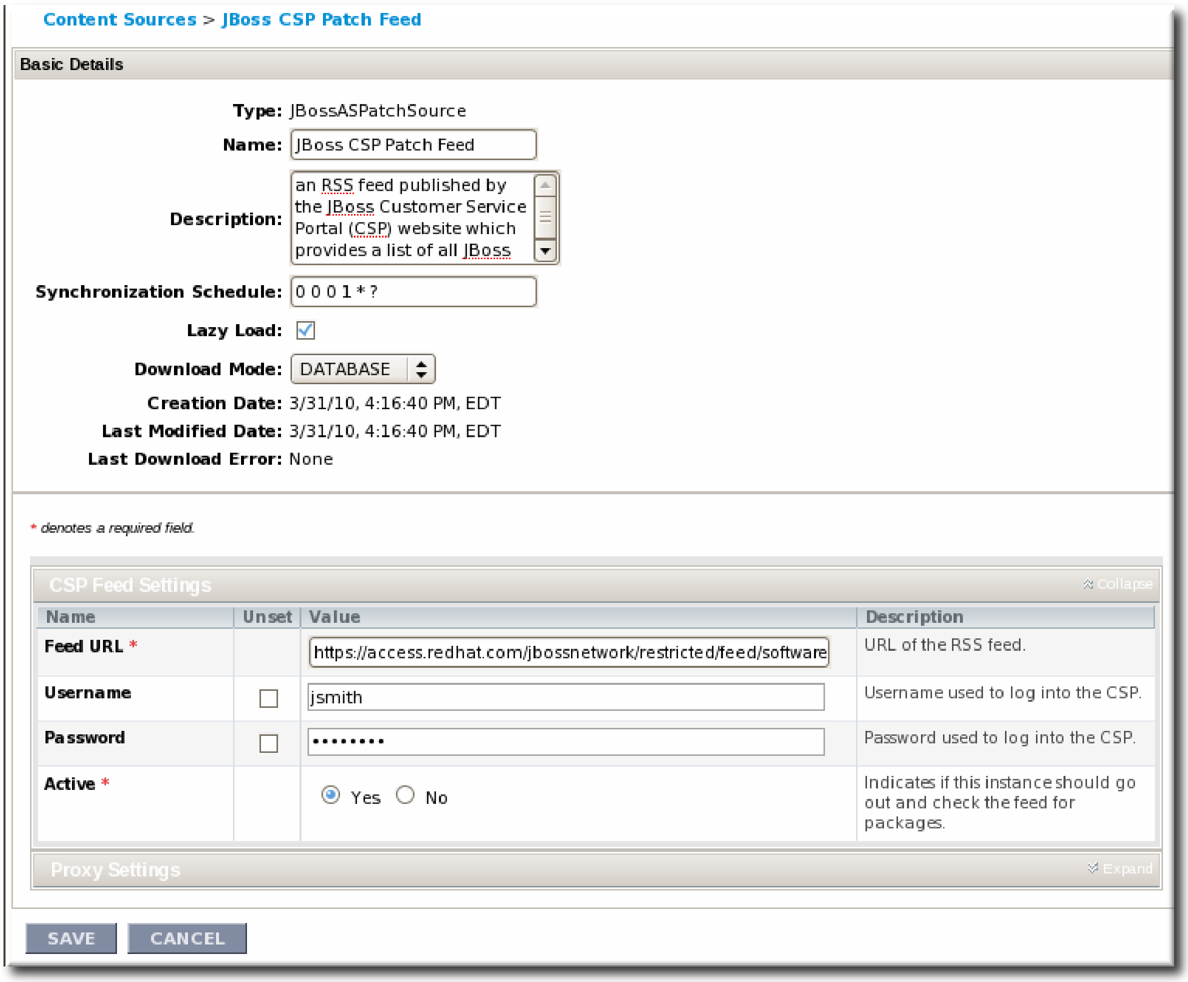


Content Sources

	Name	Date Created	Date Modified	Lazy Load?
☐	JBoss CSP Patch Feed	3/31/10, 4:16:40 PM, EDT	3/31/10, 4:16:40 PM, EDT	true

3. Click the **`Edit`** button at the bottom of the **`Customer Portal Feed Settings`** area to modify the content source.

4. Fill in the required connection information.



Basic Details

Type:	JBossASPatchSource
Name:	JBoss CSP Patch Feed
Description:	an RSS feed published by the JBoss Customer Service Portal (CSP) website which provides a list of all JBoss
Synchronization Schedule:	0 0 0 1 * ?
Lazy Load:	✓
Download Mode:	DATABASE
Creation Date:	3/31/10, 4:16:40 PM, EDT
Last Modified Date:	3/31/10, 4:16:40 PM, EDT
Last Download Error:	None

* *denotes a required field.*

CSP Feed Settings

Name	Unset	Value	Description
Feed URL *		https://access.redhat.com/jbossnetwork/restricted/feed/software	URL of the RSS feed.
Username	☐	jsmith	Username used to log into the CSP.
Password	☐	••••••••	Password used to log into the CSP.
Active *		[✓ Yes] No	Indicates if this instance should go out and check the feed for packages.

Proxy Settings

- The Customer Portal username and password.

> **NOTE**
>
> The Customer Portal password is obfuscated when it is stored in the JBoss ON database.

- The URL for the content source (https://access.redhat.com/jbossnetwork/restricted/feed/software.html?product=all&downloadType=all&flavor=rss&version=&jonVersion=2.0).
- The activation state. This should be **Yes** to enable automatic patching.

Most of the default settings, such as the schedule, can be kept.

> **IMPORTANT**
>
> Keep the **Lazy Load** checkbox activated, or all patches defined in the RSS feed, 1 GB of data, is preemptively downloaded by the JBoss ON server.

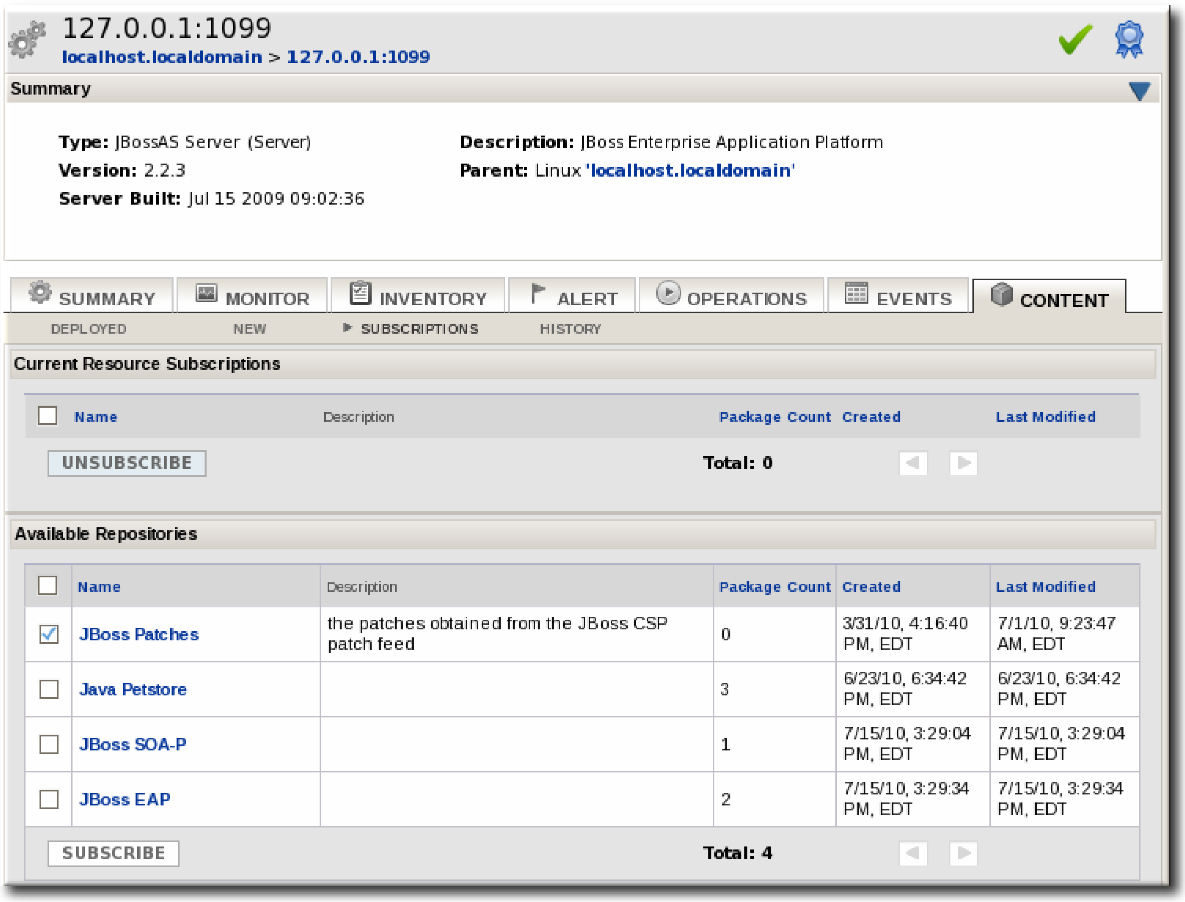
5. Click **Save**.
6. Optionally, use **Synchronize** button to force the content source to pull down the latest RSS Feed and synchronize it with the local data. The history of synchronization attempts is listed in the **Synchronization Results section**.
7. Complete any manual steps (Section 32.4.1.2, “Planning Manual Steps”) to finish applying the patches.

32.4.3. Subscribing a Specific Resource to the Default JBoss Patch Repository

The patch stream is supported for EAP 4 servers, providing patches and content updates.

> **NOTE**
>
> Patch updates using the Patch RSS Feed is not supported for JBoss Enterprise Application platform 5 or newer.

1. In the **Resources** item in the top menu, go to the **Servers** item.
2. Search for the JBoss instance to subscribe to the repository.
3. On the JBoss server resource's entry page, open the **Content** tab and select the **Subscriptions** subtab. The JBoss patches repository is listed as available for subscription.



127.0.0.1:1099
localhost.localdomain > 127.0.0.1:1099

Summary

Type: JBossAS Server (Server)
Version: 2.2.3
Server Built: Jul 15 2009 09:02:36

Description: JBoss Enterprise Application Platform
Parent: Linux 'localhost.localdomain'

SUMMARY | MONITOR | INVENTORY | ALERT | OPERATIONS | EVENTS | CONTENT

DEPLOYED | NEW | ▸ SUBSCRIPTIONS | HISTORY

Current Resource Subscriptions

☐	Name	Description	Package Count	Created	Last Modified

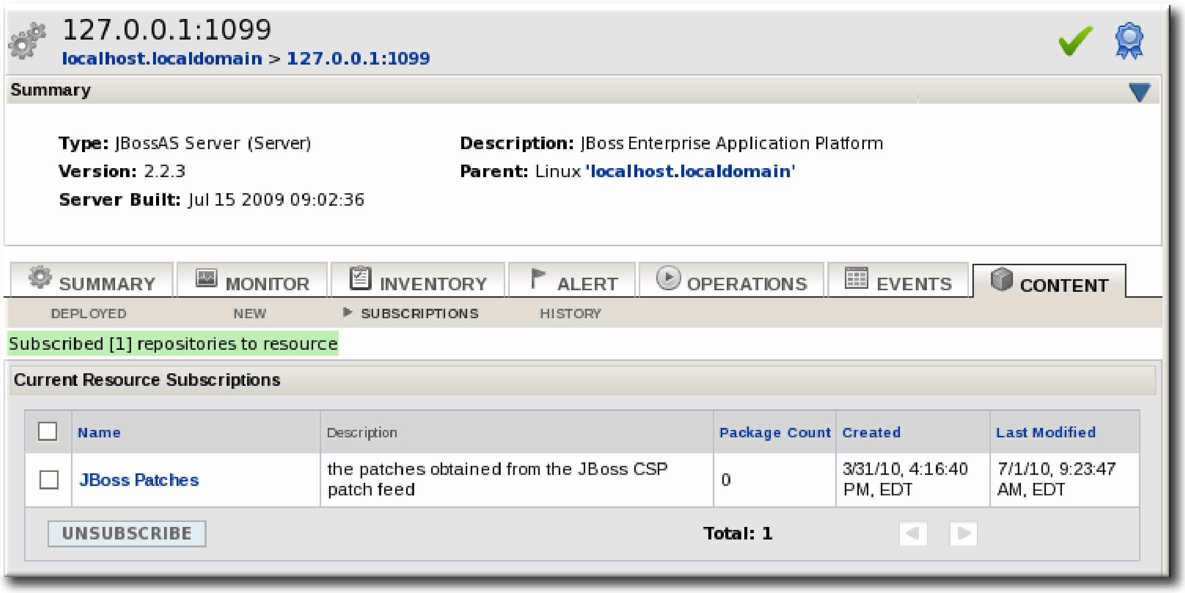
UNSUBSCRIBE   Total: 0

Available Repositories

☐	Name	Description	Package Count	Created	Last Modified
✓	JBoss Patches	the patches obtained from the JBoss CSP patch feed	0	3/31/10, 4:16:40 PM, EDT	7/1/10, 9:23:47 AM, EDT
☐	Java Petstore		3	6/23/10, 6:34:42 PM, EDT	6/23/10, 6:34:42 PM, EDT
☐	JBoss SOA-P		1	7/15/10, 3:29:04 PM, EDT	7/15/10, 3:29:04 PM, EDT
☐	JBoss EAP		2	7/15/10, 3:29:34 PM, EDT	7/15/10, 3:29:34 PM, EDT

SUBSCRIBE   Total: 4

4. Select the checkbox beside the JBoss patches repository, and click the **SUBSCRIBE** button. When the assignment is complete, the repository is listed under the `Current Resource Subscriptions` area, rather than `Available Repositories`.



127.0.0.1:1099
localhost.localdomain > 127.0.0.1:1099

Summary

Type: JBossAS Server (Server)
Version: 2.2.3
Server Built: Jul 15 2009 09:02:36

Description: JBoss Enterprise Application Platform
Parent: Linux 'localhost.localdomain'

SUMMARY | MONITOR | INVENTORY | ALERT | OPERATIONS | EVENTS | CONTENT

DEPLOYED | NEW | ▸ SUBSCRIPTIONS | HISTORY

Subscribed [1] repositories to resource

Current Resource Subscriptions

☐	Name	Description	Package Count	Created	Last Modified
☐	JBoss Patches	the patches obtained from the JBoss CSP patch feed	0	3/31/10, 4:16:40 PM, EDT	7/1/10, 9:23:47 AM, EDT

UNSUBSCRIBE   Total: 1

32.4.4. Subscribing Multiple JBoss Resources to the Default JBoss Patch Repository

The repository is associated with a content source (the patches that can be applied) and resources are then subscribed to this repository (where the patches can be applied to).

The patch stream is supported for EAP 4 servers, providing patches and content updates.

NOTE

Patch updates using the Patch RSS Feed is not supported for JBoss Enterprise Application platform 5 or newer.

1. In the `Administration` tab in the top menu, open the **Content** menu and select the `Repositories` item.
2. Click the `JBoss Patch Channel`.

   The JBoss patch repository is associated with the JBoss patch content source by default. To associate the repository to another content source, click the **ASSOCIATE...** button and assign the content source.
3. In the main page for the repository, click the **SUBSCRIBE...** button to subscribe JBoss resources to the patch repository.
4. In the search area, select `Server` in the drop-down menu.
5. Select all the JBoss server instance resources to subscribe to this repository.

32.4.5. Applying a Patch

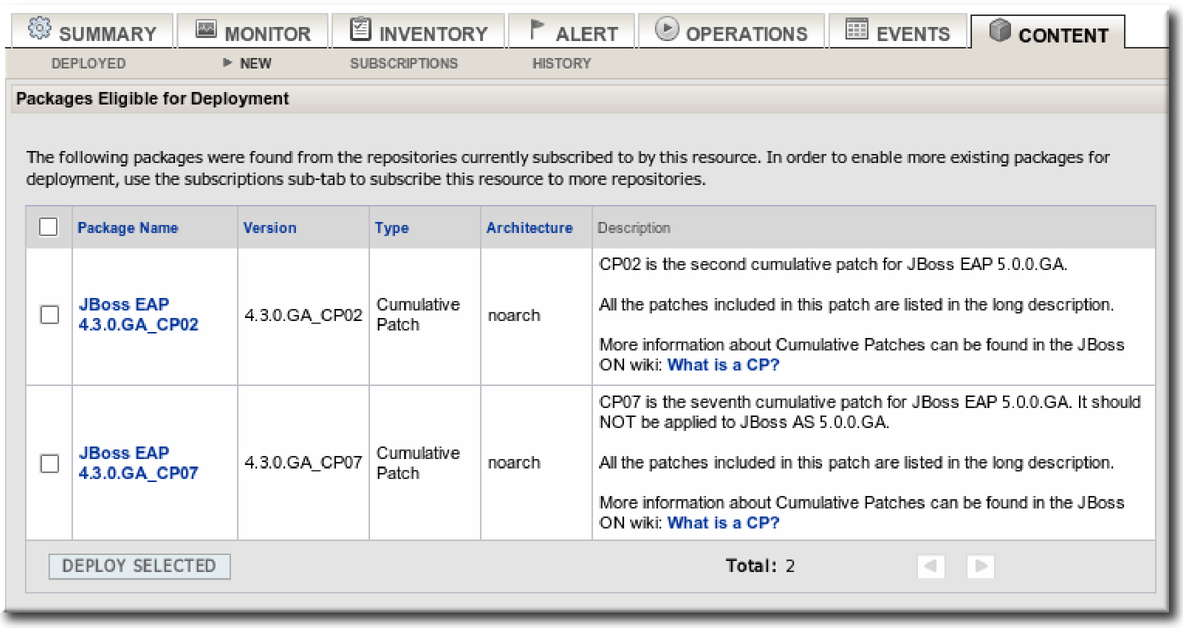
For patches to be applied to a JBoss server, the server must first be subscribed to the JBoss content repository.

The patch stream is supported for EAP 4 servers, providing patches and content updates.

NOTE

Perform patch installations during off hours or scheduled maintenance periods.

1. Stop the JBoss instance.
2. In the `Resources` item in the top menu, go to the `Servers` item.
3. Search for the JBoss instance to patch.
4. On the JBoss server resource's entry page, open the `Content` tab and select the **New** subtab. This lists all of the packages and patches which are available for that specific resource type.
5. Select the checkboxes beside the names of the patched to install, and click the **DEPLOY SELECTED** button.



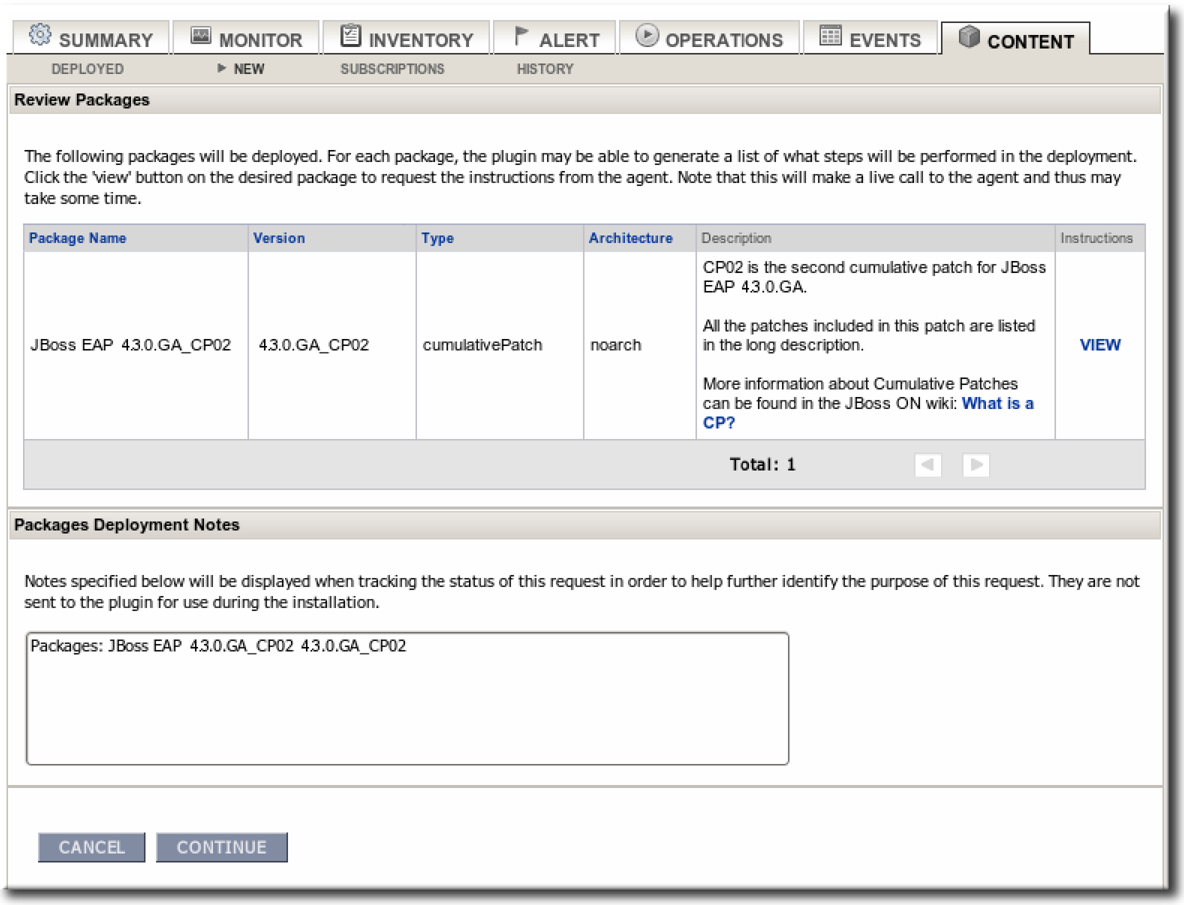
Packages Eligible for Deployment

The following packages were found from the repositories currently subscribed to by this resource. In order to enable more existing packages for deployment, use the subscriptions sub-tab to subscribe this resource to more repositories.

☐	Package Name	Version	Type	Architecture	Description
☐	JBoss EAP 4.3.0.GA_CP02	4.3.0.GA_CP02	Cumulative Patch	noarch	CP02 is the second cumulative patch for JBoss EAP 5.0.0.GA. All the patches included in this patch are listed in the long description. More information about Cumulative Patches can be found in the JBoss ON wiki: What is a CP?
☐	JBoss EAP 4.3.0.GA_CP07	4.3.0.GA_CP07	Cumulative Patch	noarch	CP07 is the seventh cumulative patch for JBoss EAP 5.0.0.GA. It should NOT be applied to JBoss AS 5.0.0.GA. All the patches included in this patch are listed in the long description. More information about Cumulative Patches can be found in the JBoss ON wiki: What is a CP?

DEPLOY SELECTED Total: 2

6. Review the information on the page and verify everything is correct. Click the **VIEW** link in the **Instructions** column to review the steps that will be performed during the package installation process.



Review Packages

The following packages will be deployed. For each package, the plugin may be able to generate a list of what steps will be performed in the deployment. Click the 'view' button on the desired package to request the instructions from the agent. Note that this will make a live call to the agent and thus may take some time.

Package Name	Version	Type	Architecture	Description	Instructions
JBoss EAP 4.3.0.GA_CP02	4.3.0.GA_CP02	cumulativePatch	noarch	CP02 is the second cumulative patch for JBoss EAP 4.3.0.GA. All the patches included in this patch are listed in the long description. More information about Cumulative Patches can be found in the JBoss ON wiki: What is a CP?	VIEW

Total: 1

Packages Deployment Notes

Notes specified below will be displayed when tracking the status of this request in order to help further identify the purpose of this request. They are not sent to the plugin for use during the installation.

Packages: JBoss EAP 4.3.0.GA_CP02 4.3.0.GA_CP02

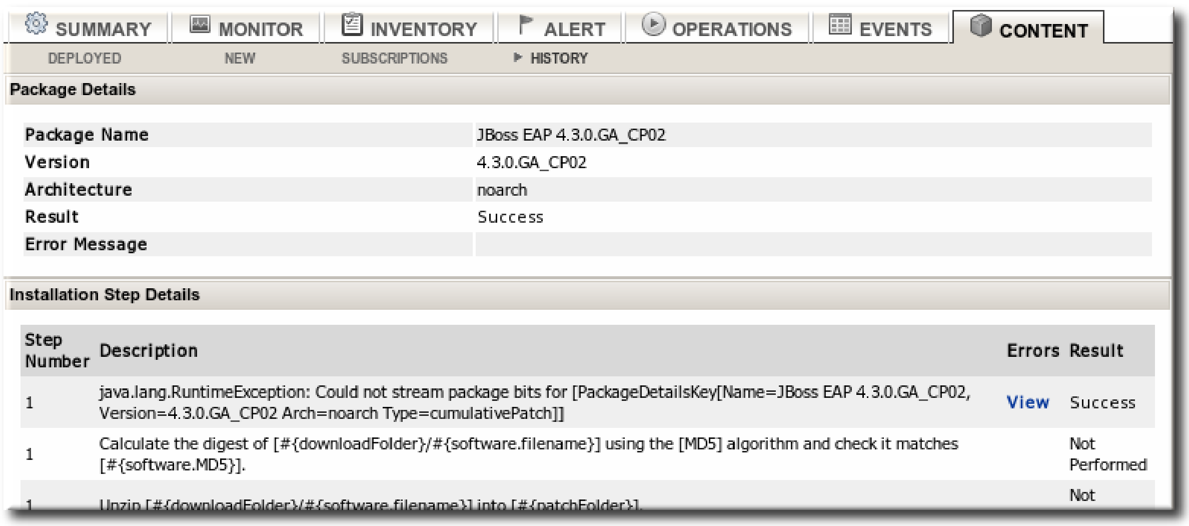
CANCEL CONTINUE

7. Optionally, enter any notes to describe the patch deployment or environment.

8. Click the **CONTINUE** button to install the updates. The patch process runs in the background; the progress can be viewed in the **History** subtab of the **Content** tab.

9. When the installation process is complete, the patch job is listed in the **Completed Requests** area. Clicking the **VIEW** button displays the list of steps that performed in the

process and whether they succeeded, failed, or were skipped.



Package Details

Package Name	JBoss EAP 4.3.0.GA_CP02
Version	4.3.0.GA_CP02
Architecture	noarch
Result	Success
Error Message	

Installation Step Details

Step Number	Description	Errors	Result
1	java.lang.RuntimeException: Could not stream package bits for [PackageDetailsKey[Name=JBoss EAP 4.3.0.GA_CP02, Version=4.3.0.GA_CP02 Arch=noarch Type=cumulativePatch]]	View	Success
1	Calculate the digest of [#{downloadFolder}/#{software.filename}] using the [MD5] algorithm and check it matches [#{software.MD5}].		Not Performed
1	Unzip [#{downloadFolder}/#{software.filename}] into [#{patchFolder}].		Not

10. When the patch process is complete, start the JBoss instance.

32.5. MANAGING MOD_CLUSTER DEPLOYMENTS FOR JBOSS EAP 5 (TECH PREVIEW)

The **`mod_cluster`** module provides intelligent, dynamic load balancing for web applications. There are two halves to **`mod_cluster`**: one in the JBoss application server (which manages the web application contexts) and one in the HTTP server (which manages sessions and routing). JBoss ON monitors and manages the **`mod_cluster`** module within the JBoss server.

32.5.1. About mod_cluster

`mod_cluster` is an HTTP load balancer that provides a level of intelligence and control over web applications that is not available with other HTTP load balancers. **`mod_cluster`** has lots of features that improve performance and management, but two are crucial:

- By using multicast (*advertising*), **`mod_cluster`** signals workers what proxy servers are available, so workers can register themselves dynamically with the cluster domain.

- Using a special communication layer between the JBoss server and the HTTP server, **`mod_cluster`** can not only register when a web context is enabled but also when it is disabled and removed from load balancing. This allows **`mod_cluster`** to handle full web application life cycles.

More detail about the features of **`mod_cluster`** is in the product documentation at http://www.jboss.org/mod_cluster.

`mod_cluster` has two modules: one for the HTTP server which handles routing and load balancing and one for the JBoss server to manage the web application contexts. Both modules must be installed and configured for the cluster to function.

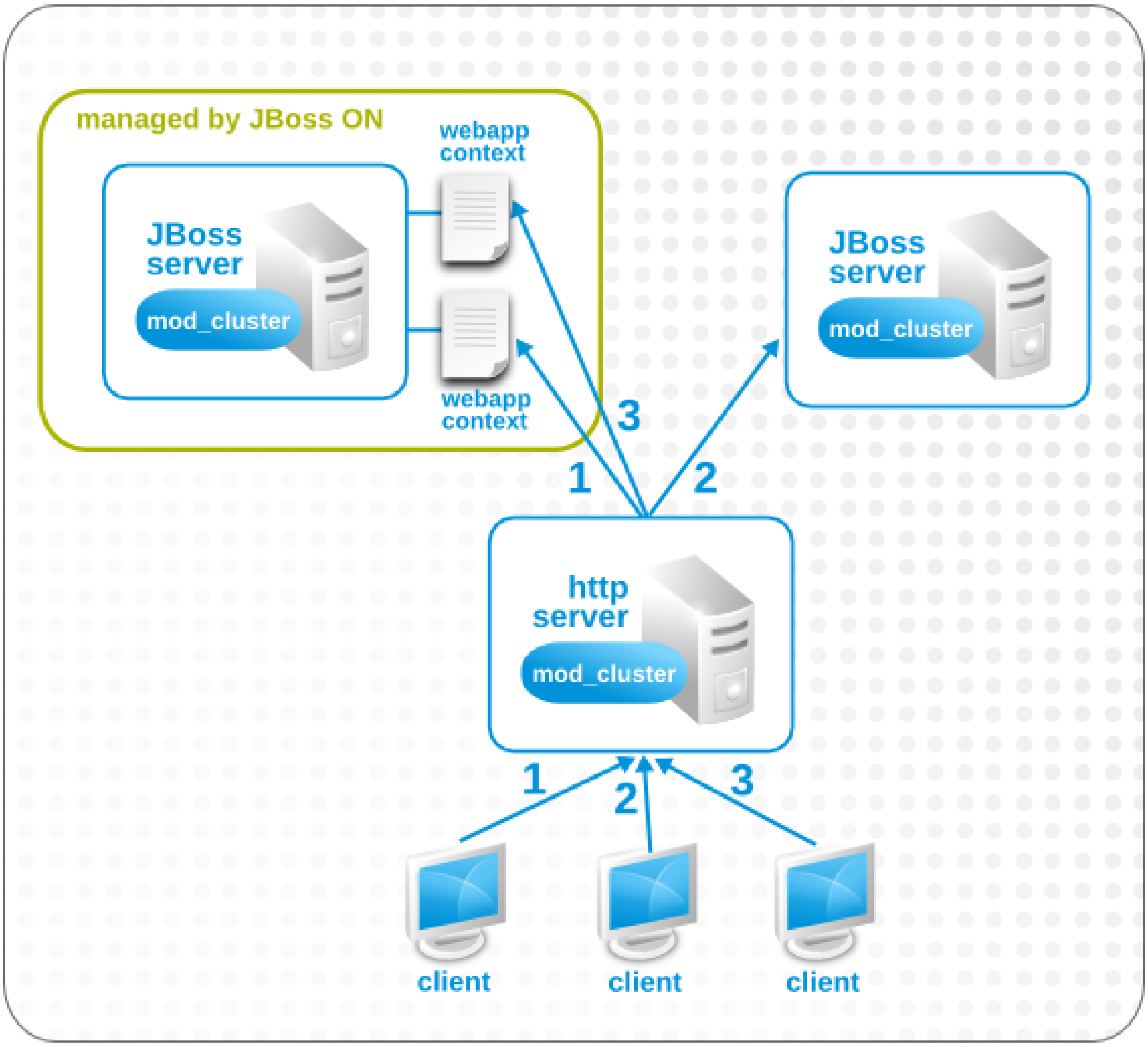
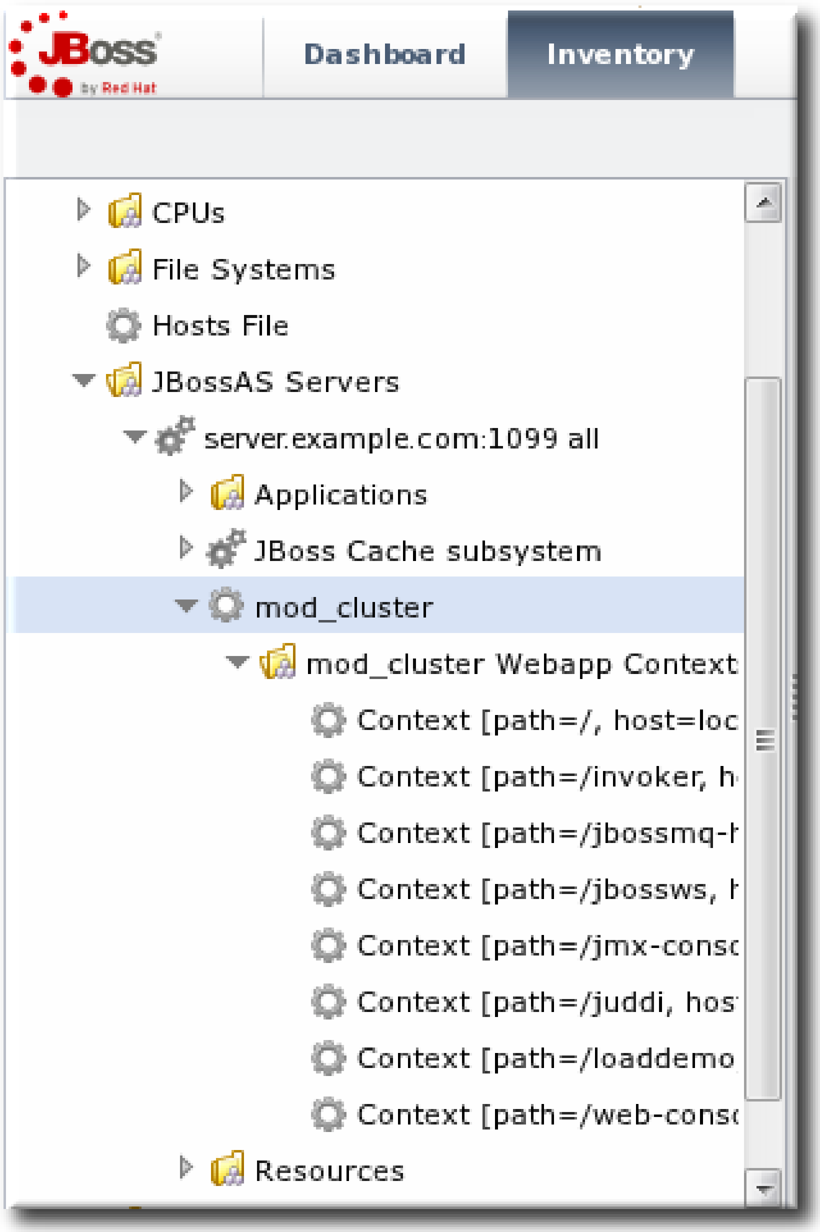


Figure 32.3. The mod_cluster Topology

In JBoss ON, the entire `mod_cluster` domain is imported as a child resource for the JBoss server. The web application contexts are listed as children resources for the cluster, with contexts as children within the `mod_cluster` module.



CPUs
File Systems
Hosts File
JBossAS Servers
server.example.com:1099 all
Applications
JBoss Cache subsystem
mod_cluster
mod_cluster Webapp Context
Context [path=/, host=loc
Context [path=/invoker, h
Context [path=/jbossmq-h
Context [path=/jbossws, h
Context [path=/jmx-conso
Context [path=/juddi, hos
Context [path=/loaddemo
Context [path=/web-conso
Resources

Figure 32.4. The mod_cluster Resource Hierarchy

IMPORTANT

The `mod_cluster` module in the HTTP server is configured externally from JBoss ON and is not managed by JBoss ON.

The `mod_cluster` module in the JBoss server can be managed by JBoss ON, and it is critical that the cluster is properly configured in order for JBoss ON to manage its resources. JBoss ON detects `mod_cluster` like any other JMX resource (such as Hibernate).

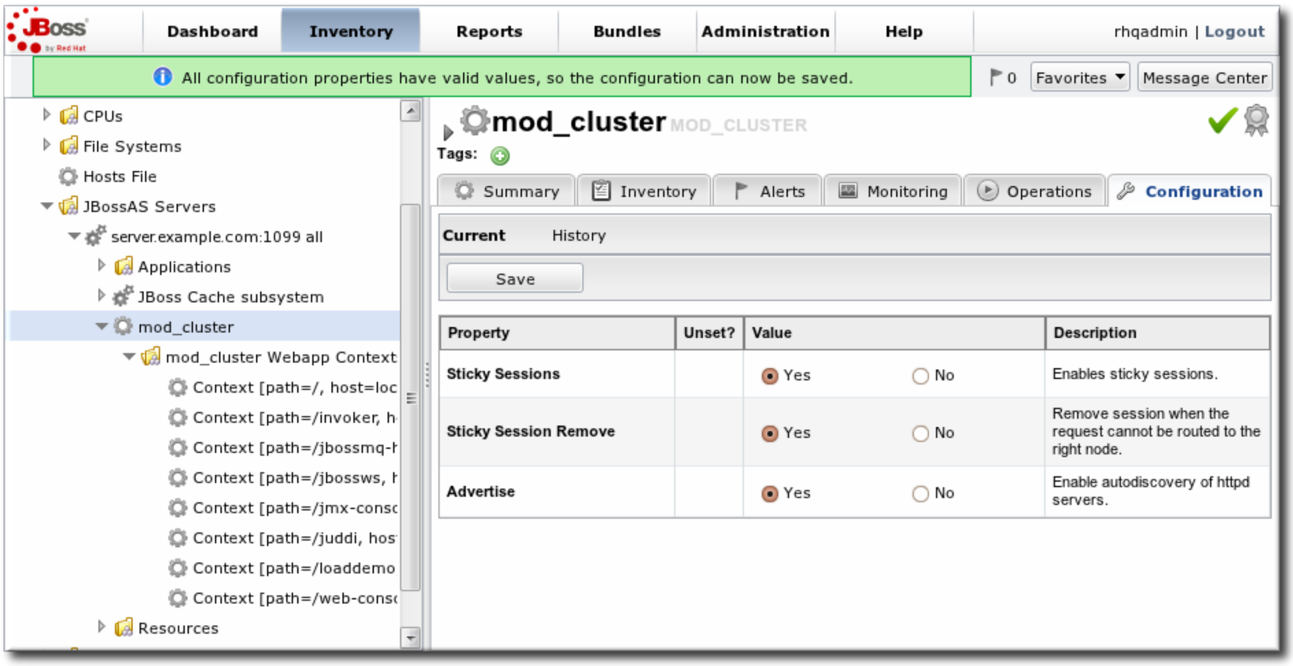
There are a number of resources available for installing and configuring `mod_cluster`:

- http://docs.jboss.org/mod_cluster/1.1.0/html/
- http://docs.redhat.com/docs/en-US/JBoss_Enterprise_Application_Platform/5/html/HTTP_Connectors_Load_Bal

32.5.2. Managing mod_cluster

The `mod_cluster` properties provide direct management over how the `mod_cluster` domain operates. Almost any part of the `mod_cluster` configuration can be managed through JBoss ON, but two elements are critical for domain behavior:

- How the `mod_cluster` domain handles sticky sessions. Sticky sessions are enabled in `mod_cluster` by default, but this can be disabled or its behavior can be changed through the configuration properties.
- Enabling advertising (multicast). `mod_cluster` can send the JBoss information to any VirtualHost configured in the HTTP server. This allows workers to find the cluster and register themselves with the JBoss server dynamically.



All configuration properties have valid values, so the configuration can now be saved.

Property	Unset?	Value	Description
Sticky Sessions		[✓ Yes] No	Enables sticky sessions.
Sticky Session Remove		[✓ Yes] No	Remove session when the request cannot be routed to the right node.
Advertise		[✓ Yes] No	Enable autodiscovery of httpd servers.

Figure 32.5. Setting Server-Level Properties

The server-level operations apply to all web application contexts configured within the `mod_cluster` domain. The obvious ones that impact the web application contexts are enabling and disabling all contexts. The other options are used to reset the `mod_cluster` domain after an error (reset the node) or reload the cluster configuration after making changes to the cluster properties.

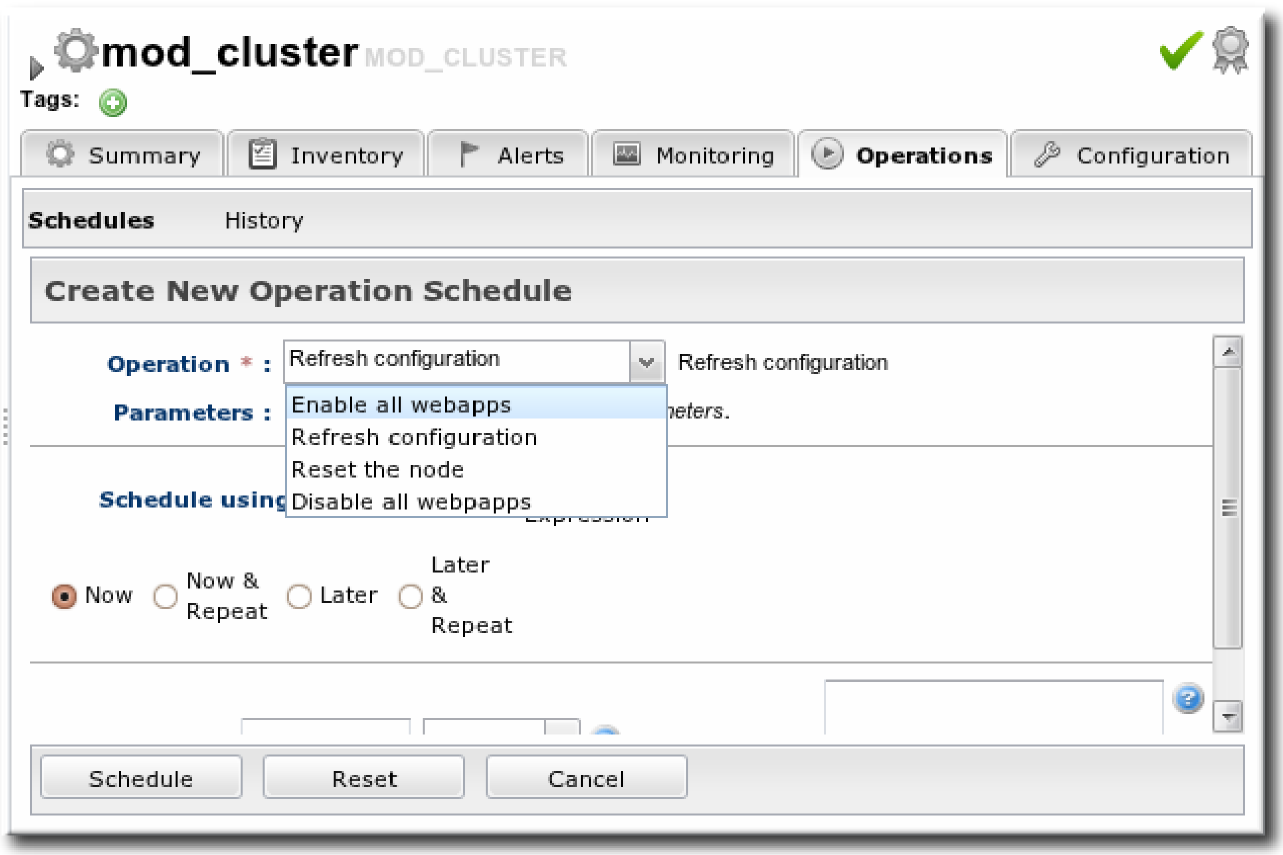


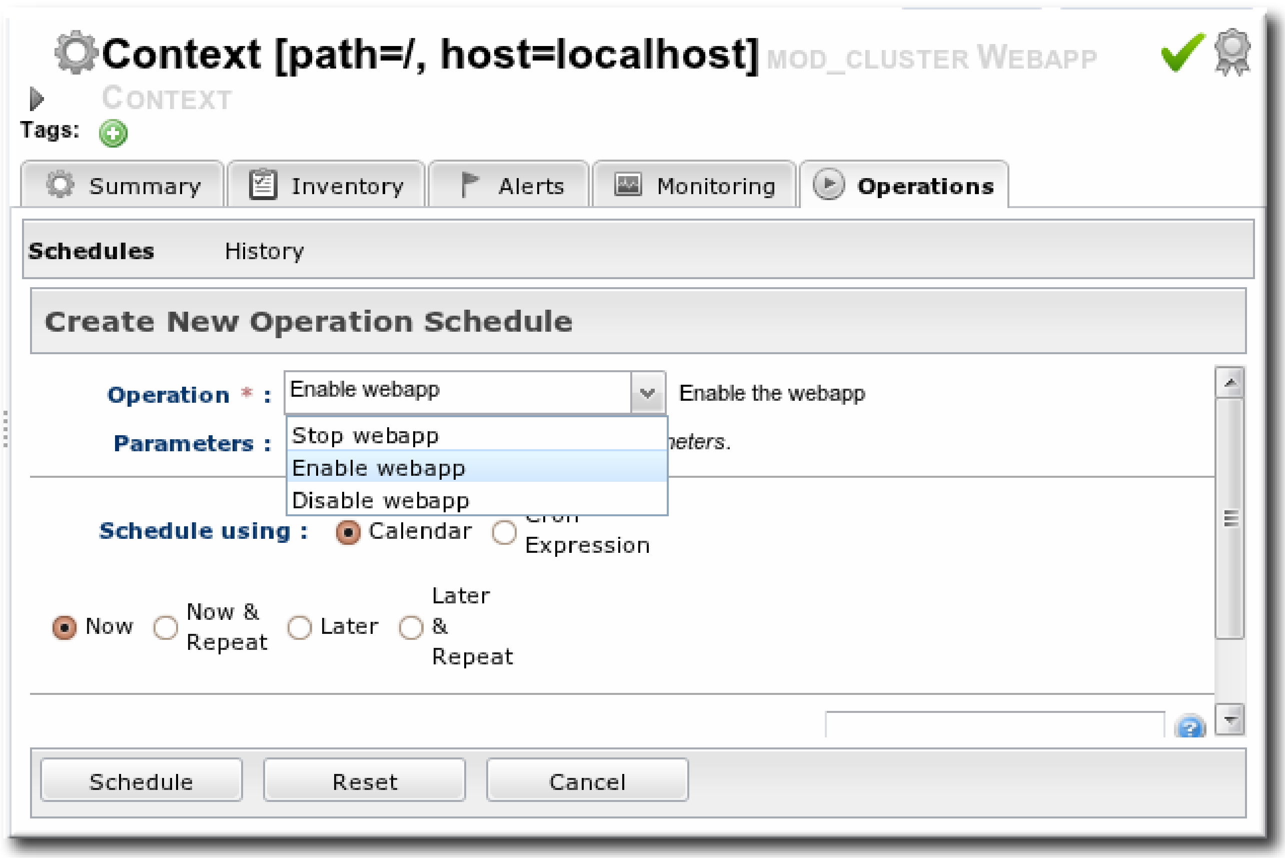
Figure 32.6. Running Server-Level Operations

32.5.3. Managing Web Applications Contexts

JBoss ON manages the full lifecycle of web application contexts within the **`mod_cluster`** load-balancing cluster.

Web application contexts can be stopped or disabled. Stopping or disabling a webapp context removes it from load-balancing balancing so that the Apache server cannot forward requests to the webapp, but it leaves the application running and available directly from the JBoss server address. (Stop allows the webapp context to drain before removing it from the load-balancing, so this essentially shuts down the webapp gracefully. It can take several minutes or even hours for the webapp context to stop. Disabling a webapp context immediately removes it from load balancing.)

JBoss ON has operations that allow JBoss administrators to manage the state of their web contexts within the **`mod_cluster`** domain.


Context [path=/, host=localhost] MOD_CLUSTER WEBAPP CONTEXT
Tags:
Summary
Inventory
Alerts
Monitoring
Operations
Schedules
History
Create New Operation Schedule
Schedule
Reset
Cancel

Operation * : Enable webapp — Enable the webapp

Parameters : …eters.

Stop webapp
Enable webapp
Disable webapp

Schedule using : [✓ Calendar] Cron Expression

[✓ Now] Now & Repeat Later Later & Repeat

Figure 32.7. Running Web Application Context Operations

Web context resource operations only apply to the specific selected context.

---

[7] The JBoss EAP/AS 5 server's Java home setting overrides the agent Java setting, if that start script value is set.

APPENDIX A. FREQUENTLY ASKED QUESTIONS

A.1. General

Q: What is the difference between JBoss Operations Network and RHQ?

A: RHQ is the upstream, open source version of JBoss Operations Network. JBoss Operations Network is a commercial product which is built on RHQ, with extensive testing and official customer support through Red Hat.

Q: Is there a publicly available issue tracker system to search for bugs and submit enhancement requests?

A: To search for a bug, report a bug, or submit an enhancement request, use Bugzilla. Select the `Other` distribution and then the `RHQ Project` component.

Q: What databases are supported?

A: PostgreSQL 8.2.4 and higher 8.2.x versions; all releases of PostgreSQL 8.3, 8.4, and 9.0; and Oracle 11 are supported databases.

The full list of supported databases, server and agent platforms, and Java versions is available at http://www.redhat.com/jboss_on/requirements.

Q: Why can't I start JBoss ON with Java 5?

A: Java 5 is no longer supported. Upgrade to Java 6.

The full list of supported databases, server and agent platforms, and Java versions is available at http://www.redhat.com/jboss_on/requirements.

Q: How can I find what my user preferences are?

A: In either the database client or from the `http://`*server.hostname*`:7080/admin/test/sql.jsp` page, run this SQL command:

```
select id, name, string_value
from rhq_config_property
where configuration_id = (select configuration_id
from rhq_subject
where name = 'your-user-name')
```

Q: What is the syntax for regular expressions used within JBoss ON?

A: JBoss ON uses regular expressions in several places in both the UI and in configuration files. All of the regular expressions follow the standard Java format, as covered in the Sun documentation for regex syntax and date/time format syntax.

Q: How often does JBoss ON check the availability of resources?

A: Agents check for resource available every minute for servers and every 10 minutes for services. The agent itself runs an availability check every 30 seconds, on a subset of its total inventory; this keeps a steady load on the agent and prevents memory-intensive usage spikes. The agent

The frequency that a specific resource is checked for availability is scheduled, much like scheduling the frequency for a metric. This availability check interval can be changed for a resource on its **`Monitoring > Schedules`** tab.

The frequency that the agent itself runs an availability check is defined in the ***`rhq.agent.plugins.availability-scan.period-secs`*** setting. The default is 30 seconds. **For performance reasons, it should never be lower than 30 seconds.** It is possible to extend the scan interval by setting a new interval as one of the **`ADDITIONAL_JAVA_OPTIONS`** values. For example:

```
RHQ_AGENT_ADDITIONAL_JAVA_OPTS="-Drhq.agent.plugins.availability-
scan.period-secs=45"
```

Q: **Why is the JBoss ON agent waiting at startup?**

A: Sometimes when you start the JBoss ON agent from the command line, it will not proceed to the **sending>** prompt, but sit there and wait. There are several possible reasons for this.

The server has rejected the agent registration request.

If the agent returns this message at start up, it means that the agent is known to the server under one name but is sending a different name when it starts:

```
Cause:
[org.rhq.core.clientapi.server.core.AgentRegistrationException:The
agent asking for registration is trying to register the same
address/port
[172.31.7.7:16163] that is already registered under a different name
[example];
if this new agent is actually the same as the original, then re-
register with
the same name]
```

To solve this, start the agent with option **`--clean`** and give the correct name.

The agent cannot reach the server.

This is an agent state where the server cannot be reached because the server is down or because a firewall has blocked the traffic. Make sure port 7080 on the server machine is reachable from the agent's machine. You can check this through a web browser.

The server cannot connect to the agent.

An error saying that the server cannot ping the agent's endpoint means that the agent can communicate with the server, but the server cannot communicate with the agent. This may mean that the agent port is blocked by a firewall.

```
The server has rejected the agent registration request. Cause:
[org.rhq.core.clientapi.server.core.AgentRegistrationException:Server
cannot
ping the agent's endpoint. The agent's endpoint is probably invalid or
there
```

```
is a firewall preventing the server from connecting to the agent.
Endpoint:
socket://172.31.7.3:12345/....
```

The agent does not have plug-ins - it will now wait for them to be downloaded.

This usually means that the server has a different security token than the one the agent was sending. This could have resulted from the java preferences entry being mangled, for example, by testing with different agent versions or VMs.

You will see this message only on initial agent startup when it does not have any plug-ins. If plug-ins were downloaded in a previous run, you will probably run in the situation shown below.

If you see this on the agent, you should also see messages like this on the server:

```
11:40:48,454 WARN [CommandProcessor] {CommandProcessor.failed-
authentication}Command failed to be authenticated! This command will
be
ignored and not processed: Command: type=[remotepojo]; cmd-in-
response=
[false]; config=[{rhq.security-token=1217855913569-109582636-
403140853869881172, rhq.send-throttle=true}];
params=
[{targetInterfaceName=org.rhq.core.clientapi.server.core.CoreServerSer
vice,
invocation=NameBasedInvocation[getLatestPlugins]}]
```

To solve this, start the agent interactively with the **`--clean`** option.

Agent startup is OK, but ping command fails.

Here, the agent successfully starts, but there may be other agent communication problems, like no monitoring data are sent. Trying to ping the agent command line can return an error like the following:

```
sending> ping


Pinging...


Failed to execute prompt command [ping]. Cause:
org.rhq.enterprise.communications.command.server.AuthenticationExcepti
on:Command
failed to be authenticated! This command will be ignored and not
processed:
Command: type=[remotepojo]; cmd-in-response=[false]; config=
[{rhq.security-
token=1214208960346-102975580-7334156733284942657, rhq.send-
throttle=true}];
params=[{targetInterfaceName=org.rhq.enterprise.communications.Ping,
invocation=NameBasedInvocation[ping]}]
```

The server log will probably contain several CommandProcessor.failed-authentication messages. This means that the agent's port is probably blocked by a firewall so that the server cannot communicate with it.

The forward and backward mappings of the IP address for high availability don't match.

Make sure the IP address of your computer can be reverse-mapped to the computer name, and that this name maps back to the same IP address. This needs to be true for all your hosts.

For example, if you have an IP address of 172.31.7.7, then the results of name resolution will look like the following:

```
$ dig -x 172.31.7.7
[...]
;; ANSWER SECTION:
7.7.31.172.in-addr.arpa. 86400 IN     PTR     example
$
$ dig example
[...]
;; ANSWER SECTION:
example       74030   IN      A       172.31.7.7
```

If your agent-server communication was working and it stopped suddenly, go to the JBoss ON server UI. Then go to **`Administration`** > **`High Availability`** > **`Server`** and check if the name displayed matches what you expect. Check the same for the agents.

If this all works, and the agent is still hanging, there may be other possibilities for misconfiguration.

Q: How do I install a supported version of PostgreSQL on Red Hat Enterprise Linux?

A: The default PostgreSQL version on Red Hat Enterprise Linux 5.5 is an older, unsupported version of PostgreSQL, so it has to be updated.

To install Postgres from Red Hat Customer Portal:

1. Log into http://rhn.redhat.com with your RHN/JBoss credentials.
2. Add the Red Hat Application Stack v2 channel.
3. Update the system:

   ```
   sudo yum update
   ```

4. Then update PostgreSQL specifically:

   ```
   sudo yum install postgresql-server
   ```

   The **`data`** directory is installed in **`/var/lib/pgsql/data`**. JBoss ON supports PostgreSQL 8.2.4 and later 8.2.x versions and all releases of PostgreSQL 8.3, 8.4, and 9.0.

5. Install and configure the JBoss ON server as normal.

Q: How can I run SQL commands against the JBoss ON database from the JBoss ON console?

A: Go to the SQL page:

```
http://server.example.com:7080/admin/test/sql.jsp
```

Q: **Is JBoss ON supported on VMWare?**

A: VMWare ESX is supported by Red Hat Enterprise Linux as equivalent to bare metal. In the case of a hardware issue with a VMWare product, work with VMWare support to resolve the issue.

JBoss EAP is fully supported on all current versions of Red Hat Enterprise Linux. Support levels and SLA will vary depending on the entitlements you purchased.

Q: **To help debug Out Of Memory conditions, how do I get the agent or server to dump heap when it runs out of memory or on demand?**

A: Pass these JVM arguments to the server or agent (setting the **RHQ_AGENT_ADDITIONAL_JAVA_OPTS** or **RHQ_SERVER_ADDITIONAL_JAVA_OPTS** variables).

```
-XX:+HeapDumpOnOutOfMemoryError -XX:+HeapDumpOnCtrlBreak
```

To drop the heap dump file in a particular location, add a path:

```
-XX:HeapDumpPath=location
```

See the SUN JVM Debugging Options for more info.

A.2. Installation and Upgrade Issues

Q: **I'm seeing error messages when I install (or upgrade) my server. What do they mean?**

A: During the upgrade, you may see error messages in the console similar to the following:

```
ERROR [ClientCommandSenderTask] {ClientCommandSenderTask.send-
failed}Failed to send
command [Command: type=[remotepojo]; cmd-in-response=[false]; config=
[{rhq.timeout=1000,
rhq.send-throttle=true}]; params=
[{targetInterfaceName=org.rhq.enterprise.communications.Ping,
invocation=NameBasedInvocation[ping]}]]. Cause:
org.jboss.remoting.CannotConnectException:[.....]
```

These can be ignored.

Q: **I've installed my server, but I can't connect to it. What's wrong?**

A: If the installer was not bound to 0.0.0.0 when it was run, then the necessary connection properties are not set for the server in the **rhq-server.properties** file. The ***java.rmi.server.hostname*** parameter must be set manually to the real IP address of the server, which matches the value of the ***jboss.bind.address*** parameter. Restart the server after editing the **rhq-server.properties** file to load the new settings.

Q: **The installer fails on PostgreSQL with "Relation RHQ_Principal does not exist."**

A: First, ensure that the JBoss ON server installer has the correct permissions to connect to PostgreSQL. Open the PostgreSQL configuration file `pg_hba.conf` and check that the permissions have been enabled. The *Installation Guide* has more information on setting up PostgreSQL for installation.

Q: **I upgraded my server, but when I try to connect to the installer page to configure it, it keeps trying to redirect me to the (old) `coregui/` module. How do I get to the installer?**

A: Some browsers cache the previous `coregui/` module and attempt to redirect you there automatically after upgrading, even though the upgraded `coregui/` module has not yet been loaded.

Simply navigate directly to the installer page:

```
http:/server.example.com:7080/installer/start.jsf
```

Q: **The JBoss ON install fails on Oracle with the ORA-01843.**

A: This issue occurs when Oracle runs in a locale where the abbreviation for April is not APR, as in English and German. There are currently two workarounds:

Put Oracle in a different locale.

Edit one of the server distribution files before running the installer:

1. Remove the old server directory and unzip the install package again.
2. Open the ***serverRoot*/jon-server-3.2.GA1/jbossas/server/default/rhq-installer.war/WEB-INF/classes** directory.
3. Edit `db-data-combined.xml`. Update a few dates in the form *01-APR-08* to be in the current locale.
4. Save the file.
5. Re-run the installer and choose to overwrite the database.

A.3. User Interface

Q: **How can I ignore an auto-discovered resource?**

A: If your agent discovers a new platform and finds a few resources that you do not want to take into inventory, you have to tell the server to ignore those resources.

First, you can select the resources to import in the auto-discovery portlet and deselect the unwanted resources. As long as they are displayed in the portlet, they are not imported.

The other option is to select the resource you do not want to import and click on **Ignore**, so it no longer appears in the portlet. However, if you try this on a resource on a freshly discovered platform, it will fail. The reason for this is that the inventory is organized in a tree-like manner with the platform as a tree-root. When a server or service is taken into the system, regardless of whether it is imported or ignored, it is attached below that root. When the platform is not yet imported into the inventory, there is no root that the ignored resource can be attached to.

You can ignore a server on a platform by performing the following steps:

1. Import the platform and leave that server unchecked.
2. When the platform is successfully imported, select the server and click **`Ignore`**.

It is not possible to just ignore a platform. If you want to ignore a platform, do not run an agent on it.

Q: I selected a search suggestion from the resource search box, but I didn't get any results. Why?

A: The suggestions in the search drop-down are not filtered by category — but the search results *are*. For example, if you are in the **`Server`** tab, the dynamic search suggestions will prompt for **`type==CPU`**, even though CPU is in the service category. If you select **`type==CPU`**, then nothing is returned in the search, because the search results *are* filtered by the category, and the search is implicitly set to the server category.

Q: Errors and stack traces in the GWT Message Center are sometimes not helpful. How can I find out what the real problem is?

A: If there are errors in the UI, an error ID is displayed, enclosed in square brackets. That can be used to track down the error and stack trace in the JBoss ON server's log file. For example:

```
java.lang.RuntimeException:[1312480384219] ...
```

The server-side log information is more useful because these exceptions are occur on the server-side and are forwarded to the GWT client.

Q: Why are the graphs and charts on the `MONITOR` tab in the GUI not displayed?

A: Errors like the following can appear in the JBoss ON server log:

```
java.lang.NoClassDefFoundError: Could not initialize class
org..enterprise.gui.image.chart.ColumnChart
```

To generate the text in graphs and charts, Java requires specific system fonts to be installed. Error messages will appear if the required font package is not installed. On Red Hat Enterprise Linux, make sure the urw-fonts package is installed:

```
yum install urw-fonts
```

A.4. Server

Q: When I start the server, I see servlet errors in my logs. What's wrong?

A: As the server starts and if agents are already running, there can be errors related to the **`Servlet.service()`** class recorded in the logs:

```
22:55:35,319 ERROR [[ServerInvokerServlet]] Servlet.service() for
servlet
ServerInvokerServlet threw exception
java.lang.reflect.UndeclaredThrowableException
        at $Proxy421.processRequest(Unknown Source)
        at
org.jboss.remoting.transport.servlet.web.ServerInvokerServlet.processR
equest(ServerInvokerServlet.java:128)
        at
org.jboss.remoting.transport.servlet.web.ServerInvokerServlet.doPost(S
erverInvokerServlet.java:157)
        at
javax.servlet.http.HttpServlet.service(HttpServlet.java:710)
        at
javax.servlet.http.HttpServlet.service(HttpServlet.java:803)
... more ...
```

This error is normal and is related to the sequence that the server loads its classes when it starts. The remoting classes are loaded early in the startup sequence, which means the server begins attempting to contact agents before it is fully started, and this can cause the errors recorded in the logs. These errors should go away once the server is completely started.

The errors can be safely ignored.

Q: **How do I get debug messages from the JBoss ON server?**

A: You can edit the ***serverRoot*/jon-server-3.2.GA1/jbossas/server/default/conf/jboss-log4j.xml** configuration file to enable debug messages by uncommenting the **org.rhq** category. This will set its priority to DEBUG. Debug messages will now be emitted for all JBoss ON subsystems to the log file. If you want debug messages to be emitted only for a smaller subset of the JBoss ON server internals, you can specify which categories you want by uncommenting them, or alternatively, you can add your own categories.

There are several commented-out categories in **log4j.xml** with comments that briefly explain what types of debug messages can be expected from a particular category. You can also emit debug messages for third-party subsystems like JBoss/Remoting and Hibernate. Some of these are already commented out in **log4j.xml**.

After you make your changes to the **log4j.xml** file, save the file and then restart the JBoss ON server.

```
serverRoot/jon-server-3.2.GA1/bin/rhq-server.sh|bat stop
serverRoot/jon-server-3.2.GA1/bin/rhq-server.sh|bat start
```

Debug messages are in the log file, ***serverRoot*/jon-server-3.2.GA1/logs/rhq-server-log4j.log**.

NOTE

By default, the console window will not display the debug messages. This is because the log4j CONSOLE appender has a threshold at **INFO**. If you want your debug messages to also appear on the console, you must change the CONSOLE appender's threshold setting to **DEBUG**.

In some cases, you will want debug messages from the JBoss ON server launcher scripts. To do this, you need to set the environment variable **RHQ_SERVER_DEBUG** to any value. After setting this variable when you start the launcher, scripts will output debug messages.

Q: **How can I specify command-line options for the server JVM?**

A: On Red Hat Enterprise Linux, override the default max heap and permgen sizes, set them via the **RHQ_SERVER_JAVA_OPTS** environment variable. For example:

```
RHQ_SERVER_JAVA_OPTS="-Dapp.name=rhq-server -Xms256M -Xmx1024M
-XX:PermSize=128M -XX:MaxPermSize=256M
-Djava.net.preferIPv4Stack= true"
 export RHQ_SERVER_JAVA_OPTS
```

Set all other JVM options via the **RHQ_SERVER_ADDITIONAL_JAVA_OPTS** environment variable. For example:

```
RHQ_SERVER_ADDITIONAL_JAVA_OPTS= "-Dfoo= true"
export RHQ_SERVER_ADDITIONAL_JAVA_OPTS
```

On Windows, for all other JVM options, add **wrapper.java.additional.n** lines to ***<server-install-dir>*\bin\wrapper\rhq-server-wrapper.inc** (you may need to create the file). For example:

wrapper.java.additional.12=-verbosegc:file=gc-log.txt

wrapper.java.additional.13=-XX:+HeapDumpOnOutOfMemoryError

wrapper.java.additional.14=-XX:HeapDumpPath=heap-dump.txt

Q: **How do I purge my schema of all data?**

A: There are instances where it's necessary to completely purge the database schema of all data. This is helpful when writing custom plug-ins and a lot of the resource hierarchy information and metadata needs to be replaced. To delete all the data from the database but keep the schema intact, simply re-install the server:

1. Save the current JBoss ON server directory.

```
mv jon-server-3.2.GA1/ jon-server-3.2.GA1.bak/
```

2. Unzip the latest JBoss ON binaries.

   ```
   unzip jon-server-3.2.GA1.zip
   ```

3. Start the new server process.

   ```
   serverRoot/jon-server-3.2.GA1/bin/rhqctl.sh start
   ```

4. Open the JBoss ON GUI and go through the installation setup. When given the choice, select the option to **`Overwrite existing data`**. This removes all of the data for the previous installation of the server.

Q: **How can I debug JDBC access and trace SQL?**

A: You can debug JDBC and access and trace SQL using log4jdbc.

Q: **How can I confirm my server's email/SMTP settings are correct?**

A: To check that the server can send emails successfully, log into the GUI as the **`rhqadmin`** user and open the email test page:

```
http://server.example.com/admin/test/email.jsp
```

Q: **My server machine does not have a writable directory called /var/run. How can I get my rhq-server.sh script to successfully write out its pid file?**

A: Set the environment variable **`RHQ_SERVER_PIDFILE_DIR`** to the full path of the directory where you want the pid file to be stored. When you run the script, that variable's value will override the default location. If you have a script that is 2.1 or older, directly edit **`rhq-server.sh`** and change **`/var/run`** to the desired directory.

Q: **When I try to start the server, I get an exception with the cause "Exception creating identity" and the server fails to start. How can I fix this?**

A: The message you are referring to probably looks similar to this:

```
Caused by: java.lang.RuntimeException: Exception creating identity:
my.host.name.com: my.host.name.com
|         at org.jboss.remoting.ident.Identity.get(Identity.java:211)
```

This is not JBoss ON-specific. It is caused by a failure with JBoss/Remoting communications. The core issue is typically because your hostname is not resolvable. The issue is normally hidden from you because JBoss/Remoting isn't producing the real error message. This error normally indicates that a machine's hostname is not externally resolvable. In order for JBoss ON to work correctly, all servers and agents must be able to resolve each other's hostnames. Best practice is to maintain a mapping of all servers and agents by using host files (e.g. **`/etc/hosts`**). This will ensure that JBoss ON will continue to work correctly even if DNS fails. However, using host files may not be practical for your environment. If this is the case, please take some time before you begin your JBoss ON installation to verify that each host you plan to run JBoss ON on can correctly resolve every other hostname in your planned environment using a tool such as nslookup.

NOTE

This applies even if you use IP addresses exclusively for configured values, as the server and agents perform host lookups for certain functions.

Q: My server logs are showing the message "Have not heard from agent ... Will be backfilled since we suspect it is down." What does that mean?

A:

```
[org.rhq.enterprise.server.core.AgentManagerBean] Have not heard from
agent [agent_name]
since [timestamp]. Will be backfilled since we suspect it is down
```

This means that the agent did not send its availability report in the required amount of time. The default is 2 minutes, but you can configure this on the **Administration** > **System Configuration** > **Settings** page. When the availability report is not sent in the required amount of time, the server assumes the agent is down. At this time it back-fills the availability of all resources managed by that agent to DOWN and the resource availabilities turn red.

This can happen for a number of reasons:

1. The agent actually shut down or crashed.
2. The machine the agent is running on shut down or crashed.
3. The network between the agent and server went down, prohibiting the agent from connecting to the server and sending the availability report.
4. The machine the agent is running on is bogged down, thus slowing up the agent and prohibiting the agent from being able to send up reports fast enough.

Q: What ports do I have to be concerned about when setting up a firewall between servers and agents?

A:

NOTE

These are the default values. Different values can be configured for JBoss ON servers or agents when they are installed.

The default server ports are 7080 (standard) and 7443 (secure SSL).

The default agent port is 16163 for both standard and secure connections.

The server also has to communicate with its database. The default port depends on the type of database.

Q: I installed the server as a Windows service, but it is failing to start with no error messages. How can I start the server as a Windows service?

A: You probably installed the server to run as the local system account and that account probably doesn't have the proper permissions to run the server or machine has been locked down due to security concerns and that local system account cannot access the network or run Java.

To solve this, create a user on your Windows box that can run the server properly. To test the user permissions, log in as the user and execute **`rhq-server.bat console`** to see if it can be run by that user. Then, install the server as a Windows Service with the **`RHQ_SERVER_RUN_AS_ME`** environment variable set to **`true`**:

```
rhq-server.bat remove
set RHQ_SERVER_RUN_AS_ME=true
rhq-server.bat install
```

Q: How do I fix an *ORA-12519, TNS:no appropriate service handler found*error when using Oracle XE?

A: Although Oracle XE is not supported for production environments, some places use it for test or development environments. To stop the ORA-12519 error, set this setting:

```
ALTER SYSTEM SET PROCESSES=150 SCOPE=SPFILE;
```

Then restart the Oracle XE database.

Q: I am seeing this error in my server logs or stack trace: *WARN [QueryTranslatorImpl] firstResult/maxResults specified with collection fetch; applying in memory*. What does that mean and what is causing it?

A: This error is issued by the Hibernate service and can be triggered for a number of different reasons. This error can be ignored.

Q: How do I stop the server from periodically logging messages that say a plug-in is "the same logical plug-in" but has "different content" and "will be considered obsolete"?

A: This is a known issue, in Bugzilla 676073. To work around it, shutdown the server, remove the plug-in jars from the server's filesystem, and restart the server.

Q: What is the difference between LDAP user authentication and LDAP group authorization in JBoss ON?

A: *Authentication* is the process that is used to verify that an entity attempting to access a resource is the identity that it is claiming to be. *Authorization* is the process of determining what rights an entity has to access a resource after its identity has been established. Let's say that a user named **`jsmith`** is attempting to log into JBoss ON. Authentication is the process of checking that the **`jsmith`** trying to log in is the same as the **`jsmith`** user that JBoss ON has in its database; this can be validated by verifying the password. Once **`jsmith`** logs in, then JBoss ON determines what resources **`jsmith`** can view and whether he can edit those resources' configuration, provision new applications, change server settings, and perform other tasks in JBoss ON.

Typically, JBoss ON uses its own user database to identify and authenticate users. It is possible to enable LDAP authentication by letting JBoss ON check an LDAP server for user information first, and using that base of users for valid JBoss ON users. This is *LDAP authentication*. This is essentially pass-through authentication. A user attempts to log into JBoss ON. JBoss ON first sends the credentials to the LDAP server to see if the LDAP server has stored that user; if authentication fails at the LDAP server, then JBoss ON checks its own database.

All authorization in JBoss ON is based on roles. Both users and resource groups are added to roles, and then permissions are assigned to those roles. These roles are created and managed in JBoss ON, but it is possible to use group membership in an LDAP group to supply the users in JBoss ON role. Essentially, this takes an existing list of users in LDAP and just says, "use this list for the role members." The LDAP group is added to a JBoss ON role, and then every member in that LDAP group automatically has whatever rights the role has. That is *LDAP authorization*.

NOTE

LDAP authentication is recommended for LDAP authorization, but it is not required.

This covered in more detail in "Integrating LDAP Services for Authentication and Authorization" .

Q: **How do I set up LDAP group authorization?**

A: LDAP authorization is set up in the **`Administration`** tab, under **`System Settings`**.

First set up JBoss ON to allow LDAP users to authenticate using LDAP user accounts ("Configuring LDAP User Authentication"). (LDAP authentication isn't required, but it is recommended.) Then, configure JBoss ON to check for LDAP groups on the LDAP server ("Associating LDAP User Groups to Roles in JBoss ON").

There are five elements in the LDAP server configuration that you need to know to configure LDAP group authorization:

The information to connect to the LDAP server, in the form of an LDAP URL. For example, *ldap://server.example.com:1389*.

The username and password to use to connect to the server. This account should have read access to the subtrees being searched.

The search base. This is the point in the directory tree to begin looking for entries. This should be high enough to include all entries that you want to include and low enough to improve performance and prevent unwanted access. For example, if you have **`ou=Web Team,dc=example,dc=com`** and **`ou=Engineering,dc=example,dc=com`** and you want to include groups in both subtrees in JBoss ON, then set the search base high up the tree, to **`dc=example,dc=com`**. If you only want the engineering groups to be used by JBoss ON, then set the search base to **`ou=Engineering,dc=example,dc=com`**.

The group filter. This creates the search filter to use to search for group entries. This can use the group object class, which is particularly useful if there is a custom attribute for JBoss ON-related entries. This can also point to other elements — like the group name, a locality, or a string in the entry description — that are useful or meaningful to identify JBoss ON-related groups.

The member attribute. There are different types of group object classes, and most use different attributes to identify group members. For example, the **`groupOfUniqueNames`** object classes lists its members with the ***uniqueMember*** attribute.

After LDAP authorization is enabled, then you can associate the roles in JBoss ON to the appropriate groups in the LDAP directory.

A.5. Agent

Q: **I have a physical machine hosting multiple virtual machines with shared disk resources. How can I run an agent on each virtual instance?**

A: You can run multiple agents on the same box, but the agents must run out of two different agent installation directories.

Q: **How do I get debug messages from the JBoss ON agent?**

A: The easiest and quickest way to get your agent to start logging debug messages is, before starting your JBoss ON agent, to set the environment variable **RHQ_AGENT_DEBUG** to any value. When you start the agent, both the launcher scripts and the agent itself will output debug messages. When you use this environment variable, the agent will use an internal **log4j** configuration file.

For more fine-grained control over what **log4j** categories have **DEBUG** priority, edit the **conf/log4j.xml** file and restart the agent to load the changes. Do not set **RHQ_AGENT_DEBUG** if you want the agent to use the **log4j.xml** file; setting that environment variable causes the agent to override this **log4j.xml** with an internally configured **log4j** configuration.

The log messages can be found in the log files located in the ***agentRoot*/rhq-agent/logs** directory.

If you are launching the JBoss ON agent on Microsoft Windows using the service wrapper, you must set **RHQ_AGENT_DEBUG** and then install the service:

```
rhq-agent-wrapper.bat install
```

Q: **How do I restrict which agents are allowed to connect to the server?**

A: Change the server's JBoss/Remoting servlet configuration to restrict the IPs agent requests can come from. If the agents are on a specific subnet, then connections can be restricted to only that subnet.

1. Create a file for the restriction rule, with this name and location:

   ```
   vim
   serverInstallDir/jbossas/server/default/deploy/rhq.ear/jboss-
   remoting-servlet-invoker-2x.r3040.jon.war/WEB-INF/context.xml
   ```

2. Add this content to the file:

   ```
   <?xml version="1.0" encoding="UTF-8"?>
   <Context>
   <Valve className="org.apache.catalina.valves.RemoteAddrValve"
   ```

```
allow="192.168.*,142.104.128.*,10.224.27.182"/>
</Context>
```

The **allow=** attribute lists the IPs that are allowed to connect to the server. All other IPs are blocked.

Q: Do I have to run the agent as root?

A: You do not have to run the agent as root.

However, that said, some resource types have very strict limits on what users can access their configuration files and processes, and running the agent as root may be the only way to manage or monitor those resources.

For example, a PostgreSQL plug-in lets the agent probe the PostgreSQL configuration file, **postgres.conf**. Running the agent as a non-root user without PostgreSQL privileges means that the agent cannot read and manage the file. (And there will be log messages in the agent log saying so.) There are other resources that have similarly privileged files, like iptables and even some JBoss servers.

Running the agent as root gives the agent privileges to manage all those things. Without that privilege level, the agent has restricted views of managed resources.

Q: How do I clean start the JBoss ON agent, as if newly installed?

A: There are three points of configuration for the agent: the agent's (local) persisted configuration, the agent inventory (and associated resource data), and the platform entry in the server inventory.

To clean the agent configuration and restart the agent as if it were new, then there are two steps to take:

1. Remove the platform entry from the JBoss ON server inventory. Since the platform entry is representative of the agent entry, this effectively removes the agent from the JBoss ON topology.

2. Stop the agent and then restart it using the **--fullcleanconfig (-L)** command-line option.

   ```
   agentRoot/rhq-agent/bin/rhq-agent.sh --fullcleanconfig
   ```

   The **--fullcleanconfig** option removes all of the local inventory for the agent, reloads its configuration fresh from the **agent-configuration.xml** file, and re-registers the agent with the server.

   Optionally, pass the **--config** argument to have it start up with a user-specified configuration file. Otherwise, the default **conf/agent-configuration.xml** file is used. If no directory is given, then the command looks for the configuration file in the agent's **conf/** directory.

   ```
   agentRoot/rhq-agent/bin/rhq-agent.sh --fullcleanconfig -c my-
   agent-configuration.xml
   ```

Q: **How can I do a "clean config" for an agent running as a background Windows service?**

A: The JBoss ON agent Windows service (like all Windows services) runs as a specific user. Go to that user's Windows registry and delete the agent's configuration node. The RHQ Agent uses the standard Java Preferences API, so the agent's configuration is stored as a node under the normal Java Preferences location in the Windows registry, such as HKEY_CURRENT_USER\Software\JavaSoft\Prefs\rhq-agent\default. ("default" is the name of the preferences node.)

Deleting the node only removes the previous configuration; the agent has to be reconfigured before it can be started again. The simplest path is to start the agent in the foreground and go through the interactive agent configuration, then stop that session and start the agent as a service.

Alternatively, the agent can be forced to read its configuration anew from the **`agent-configuration.xml`** file. To force the agent to re-read its configuration from file, you won't be able to start it in the foreground, which makes re-configuring it a little bit more difficult.

If the agent has been added as a JBoss ON resource, you can invoke the "Execute Command Prompt" operation and run the **`config --import agent-configuration.xml`** command.

Alternatively, you can edit the **`rhq-agent-wrapper.conf`** file and add a line for a third parameter:

```
wrapper.app.parameter.3=--cleanconfig
```

This forces the agent to re-read its configuration from the **`agent-configuration.xml`** every time it is started as a service. In this case, the **`agent-configuration.xml`** must be preconfigured with all of the required (and optional) settings for the agent, so that it restarts with the correct configuration.

Q: **How can I update the plug-ins on all my agents?**

A: When you add a new plug-in to your system or upgrade an existing plug-in, you can instruct your agents to update their existing plug-ins with the new plug-in versions.

You can do this individually by executing the prompt command **`plugins update`** at any agent prompt or through the UI with the **`Update All Plugins`** task in the agent's **`OPERATION`** tab.

To update all of agents with the latest plug-ins, launch a group operation on the JBoss ON agents autogroup. First, create an autogroup by navigating to the **`Browse Resources`** page. Then click **`Group Definitions`** > **`New Definition`** button. The autogroup definition should have the expression:

```
resource.resourceType.pluginName = RHQAgent
resource.resourceType.typeName = RHQ Agent
```

This creates a compatible group that dynamically adds all JBoss ON agents as members to that group.

NOTE

If you already have a compatible group with your agents as members, you can skip this group creation step.

Next, navigate to the agent group. In the **OPERATIONS** tab, invoke the **Update All Plugins** operation. This tells all of your agents in that group to update their plug-ins. Once that group operation is completed, all of your agents will have up-to-date versions of all plug-ins.

Q: **How can I change the agent name after it has already been registered?**

A: When you start the agent for the first time, you are asked in the setup screen for an agent name. This name must be unique across all agents in your environment. Once it is registered you cannot change this name. If you attempt to re-register this agent, you must re-register it with the same name that it was registered with before.

The agent name is not the same as the JBoss ON agent resource name in the UI. If you import an JBoss ON agent resource into inventory, that resource's name is *agent_name JON agent*. This JBoss ON agent resource name can be changed by editing its value within the **INVENTORY** tab. Changing this name does not change the name that the agent is registered under. Your agent is still registered under its original agent name.

Q: **I want to run agents on all my machines, but only one starts OK. The rest fail due to binding to a wrong address.**

A: There are a couple of things that can cause this error.

```
FATAL [main] (org.jboss.on.agent.AgentMain)-
{AgentMain.startup-error}The agent encountered an error during startup
and must abort java.net.BindException: Cannot assign requested address
```

First, did you change the `agent-configuration.xml` manually (to change IP addresses, for example) *after* setting up the agent? The agent's configuration XML file is *not* referenced after the agent is setup because its configuration is persisted using Java Preferences. Persisting the configuration allows the agent to be updated or re-installed without losing its configuration. To change the agent's configuration file and have those changes picked up, restart the agent and pass the `--config` command line option (or `-c` which is shorthand for `--config`). This tells the agent to re-read the configuration file and override any old configuration it persisted before.

If your home directory is stored on NFS, then you are probably picking up the same Java preferences across all machines (`$HOME/.java` on Red Hat Enterprise Linux). This is usually not an issue on Windows since the Java preferences are stored in the registry. If you are running the agents as the same user and your user's home directory is shared, then have the agents use different Java preferences names. Edit your agents' `agent-configuration.xml` files and change their Java preferences node names from `default` to something that makes them unique across all agents. For example:

```
<node name="NewName">
```

Since you overrode the default location, every time you start your agent you need to tell the agent where it can find its preferences. You tell the agent the new preference name using the **`--pref`** option. Since you changed the configuration file, restart the agent with the **`-c`** to specify the configuration file.

```
agentRoot/rhq-agent/bin/rhq-agent.sh --pref NewName -c agent-
configuration.xml
```

Subsequently, always restart the agent with the **`--pref`** option to pass in the node name.

Alternatively, define the system property **`java.util.prefs.userRoot`** to point to another, unique, location for preference. When the agent starts, Java will use the value of that system property as the location where it will store its Java Preferences. You can set this system property on the agent by setting the **`RHQ_AGENT_ADDITIONAL_JAVA_OPTS`** environment variable. When you set that environment variable, **`rhq-agent.sh`** will add its value to the default set of Java options when passing in options to the agent's Java VM:

```
set RHQ_AGENT_ADDITIONAL_JAVA_OPTS="-
Djava.util.prefs.userRoot=/etc/rhq-agent-prefs"
agentRoot/rhq-agent/bin/rhq-agent.sh
```

Q: **When starting the agent via a Windows service, the agent fails to start, and I see the error "java.lang.IllegalStateException: The name of this agent is not defined - you cannot start the agent until you give it a valid name" in the agent wrapper log file. What does this mean?**

A: The agent cannot ask for its initial setup configuration when installing as a Windows service because there is no console for the user to see and answer the prompts. This means that you need to either pass the information through a preconfigured file or run the agent in standard, non-service, mode once as the user that should run the service to configure it before installing it as a service.

Q: **My agent setup is correct but my agent is getting the following message: "Cause: org.jboss.remoting.CannotConnectException: Can not connect http client invoker."**

A: This error is typically seen when the server's endpoint address is not set to something that can be resolved by the agent. The public endpoint address set for each server *must* be resolvable by every JBoss ON agent because of the high availability cloud configuration of JBoss ON servers.

Check your server endpoint information in the high availability pages in the GUI and change the settings if necessary. After the update, restart the agent.

Q: **My agent machine does not have a writable directory called `/var/run`. How can I get my rhq-agent-wrapper.sh script to successfully write out its pid file?**

A: Set the environment variable **`RHQ_AGENT_PIDFILE_DIR`** to the full path of the directory where you want the pid file to be stored. When you run the script, that variable's value will override the default location. If you have an older script (2.1 or older), directly edit **`rhq-agent-wrapper.sh`** and change **`/var/run`** to the desired directory.

Q: How often does the agent scan for resources?

A: When an agent is installed, it *scans* the platform, and all applications on it, for any servers, services, or other items which can be included into the inventory. The process of finding potential

There are different scans for each type of resource: platform, server, and service. High level scans for servers and platforms are initiated by the agent every 15 minutes. A service scan detects lower-level services that are running in servers that have already been imported into the inventory. These scans run by default every 24 hours. Both of these intervals are configurable.

NOTE

A server must be imported into the inventory before any of its child processes, servers, or services can be detected by the discovery scan.

Q: How can I view the agent's persisted configuration?

A: The agent's configuration is initially read from **`agent-configuration.xml`** and overlaid with the value given at the agent setup. After the agent is initially configured, it persists that configuration and never refers back to **`agent-configuration.xml`**, unless you clear the configuration.

There are several ways to view the agent's persisted configuration:

1. If the agent is in the JBoss ON inventory, simply go to your agent's **`Configuration`** tab to view its live configuration. This is the same configuration that is persisted.

2. If the agent is currently running in non-daemon mode (i.e. you have the agent prompt on your console), you can use the **`getconfig`** or **`config`** prompt commands to view the live configuration. Type **`help getconfig`** or **`help config`** for more information.

3. If the agent is in the JBoss ON inventory, run the **`Execute Prompt Command`** operation and invoke the **`getconfig`** prompt command.

4. Because the agent configuration is stored in the standard Java Preferences API backing store, you can use any tool that can examine Java preferences, such as Google's Java Preferences Tool. This is a GUI tool that can give you a file system-like view into your Java preferences. The agent preferences are stored in the **`User`** preferences node under the node name **`rhq-agent`**. Depending on the **`-p`** option that is passed to the agent for its node name when it is started, the actual configuration settings are found under a sub-node under **`rhq-agent`**. The default preferences node is called **`default`**, so typically your agent's persisted configuration is found in the user preferences under **`rhq-agent/default`**.

WARNING

Do not attempt to change the values of the preferences using third-party tools without knowing what you are doing. This could disable the agent if you change the wrong preference to the wrong value. Use this mechanism only to view your agent's configuration.

Q: **How can I find out what environment variables and Java system properties are set in my agent JVM process?**

A: The **`version`** agent prompt command shows a list of the agent process' environment variables and system properties. **`version --sysprops`** provides a list of all the system properties, and **`version --env`** provides a list of all the environment variables. (At the agent prompt, run **`help version`** for the syntax of that command.)

Q: **How can I get a dump of inventory information from an agent running on another machine?**

A: If the agent inventory becomes corrupted, dumping the agent's inventory can help debug the problem.

To get this information, get the agent's **`data/inventory.dat`** file. Copy that file to the local machine. Then, run an agent on the local machine, with the same plug-ins as the other agent. The agent doesn't necessarily have to be connected to a server, but the plug-in container must be started, so the agent has to have been registered. Then, export the information from the imported DAT file.:

```
inventory --xml --export=/bad-inventory.xml /the/bad/inventory.dat
```

If you do not specify the **`--export`** option, the XML will simply be dumped to the stdout console window.

Now you have an XML file that describes what the customer's agent thinks is its inventory.

Q: **I need to change the IP address of my agent machine. How do I keep my server and agent up to date with that change?**

A: The agent has a configuration preference named ***`rhq.communications.connector.bind-address`*** which sets the value of the IP address the agent binds to when it starts its server socket (the thing it listens to for incoming messages from the server).

If you change the agent's IP address (and invalidate the old agent IP address), you have to do a couple things:

1. Change the agent's configuration so that preference value is the same as the new IP address. Issue a setconfig prompt command on the agent prompt:

   ```
   setconfig rhq.communications.connector.bind-address=IP_address
   ```

Do not change **`agent-configuration.xml`**; the changes will not take effect.

If the agent is running in the background as a daemon process, shut it down with the script (**`rhq-agent-wrapper.sh|bat`**) and restart it.

2. Restart the agent after editing its configuration.

Once the agent is restarted, it will use that new IP address.

Q: **How can I stop my agent from thinking the server keeps going up and down when the server has remained running the whole time?**

A: There can be errors like the following appears in the agent logs:

```
INFO (org.rhq.enterprise.agent.AgentAutoDiscoveryListener)-
{AgentAutoDiscoveryListener.server-offline}
The Agent has auto-detected the Server going offline [InvokerLocator
[servlet://server:7080/jboss-remoting-servlet-invoker
/ServerInvokerServlet?rhq.communications.connector.rhqtype=server]] -
the agent will stop sending new messages
...
INFO (org.rhq.enterprise.agent.AgentAutoDiscoveryListener)-
{AgentAutoDiscoveryListener.server-online}
The Agent has auto-detected the Server coming online [InvokerLocator
[servlet://server:7080/jboss-remoting-servlet-invoker
/ServerInvokerServlet?rhq.communications.connector.rhqtype=server]] -
the agent will be able to start sending messages now
```

This means the agent has auto-detected, through the multicast detector, the server going down and then back up. This is different from the detection-via-polling, which is the second way the agent attempts to detect the server's status.

If the agent is erroneously detecting the server going up or down, it is possible that either the network does not support multicast traffic or the multicast network is acting abnormally. In either case, disable the agent multicast detector and have the agent instead rely on polling to detect changes in the server status.

To turn off the multicast detection, set the following agent preferences to false:

rhq.agent.server-auto-detection

rhq.communications.multicast-detector.enabled

Since you are disabling multicast detection, ensure that the polling detection feature is enabled, meaning the ***`rhq.agent.client.server-polling-interval-msecs`*** value is larger than 0, typically 60000. Otherwise, the agent will never be able to know when the server goes down.

Once you reconfigure the agent, restart the agent so the communications subsystem can detect the changes.

A.6. Log Messages

Q: **What are "Command failed to be authenticated" messages?**

A: Agents are assigned security tokens when they first register with the server. The token is one way an agent identifies itself with the server. If an agent does not identify itself with any token, or if it identifies itself with a wrong token, the server will deny access to that agent. The server will

```
02:31:33,095 WARN  [CommandProcessor] {CommandProcessor.failed-
authentication}
Command failed to be authenticated! This command will be ignored and
not processed:
Command: type=[identify]; cmd-in-response=[false]; config=[{}];
params=[null]
```

Failure to authenticate errors usually mean the agent has been misconfigured, or it is an unknown agent attempting to identify itself as another agent. Restart your agent with the **`--cleanconfig`** command line option to clean out its configuration and re-register.

NOTE

Do not rely on the security token mechanism as a way of protecting your JBoss ON environment from intrusion. Configure SSL for agent-server communications.

Q: What are "fail-safe cleanup" messages?

A: These messages can be ignored. These relate to the Hibernate services used by JBoss ON and how it automatically cleans up after itself to prevent memory leaks.

```
13:43:10,781 WARN [LoadContexts] fail-safe cleanup (collections) :
org.hibernate.engine.loading.CollectionLoadContext@103583b
<rs=org.postgresql.jdbc3.Jdbc3ResultSet@d16f5b>
```

A.7. Server and Agent Plug-ins

Q: How can I extend JBoss ON?

A: New plug-ins can be written to support other external projects and custom applications.

Q: How can I write a plug-in for JBoss ON?

A: Writing both agent and server plug-ins is covered in the *Plug-in Writing Guide* in the JBoss ON documentation set.

Q: What is the skeleton plug-in module?

A: To start writing custom plug-ins, begin with the custom-plug-in maven module skeleton, a template available with the JBoss ON source code. This includes a maven pom with defined dependencies, a starter set of resource components and a minimal **`rhq-plug-in.xml`** plug-in descriptor.

A.8. General Resource Questions

Q: I deleted a Platform from inventory. How can I rediscover it, so I can re-import it?

A: You can force an agent discovery by issuing the following command at the agent command prompt:

```
> discovery -f
```

Q: On a Red Hat Enterprise Linux platform, interface "sit0" is discovered, but it is always red. How can I remove this interface from inventory?

A: Because network adapters are child services, there is currently no way to tell JBoss ON to ignore them or not inventory them.

However, there is an easy way to disable this interface on the machine itself. If you have root or sudo access to the box, disable all IPv6 support.

Change the **NETWORKING_IPV6** value in **/etc/sysconfig/network**:

```
NETWORKING_IPV6=no
```

WARNING

Disabling this interface will disable IPv6 support.

Then, edit **/etc/modprobe.conf** to include the following lines:

```
alias net-pf-10 off
alias ipv6 off
```

Stop the ipv6tables service:

```
service ip6tables stop
```

Disable the ipv6tables service:

```
chkconfig ip6tables off
```

Q: How can I collect syslog messages as JBoss ON events?

A: The Linux platform plug-in can monitor syslog messages by emitting them as events. Syslog messages can be collected by the plug-in by either reading syslog message files or by receiving them over a socket listener.

The syslog must be configured to format the messages in a way that JBoss ON can parse. You can either tell JBoss ON (in the platform's plug-in configuration or connection properties) what regular expressions can parse syslog messages, or format the messages in the syslog config file

(**/etc/rsyslog.conf**) so that JBoss ON understands. For example:

```
$template RHQfmt,"%timegenerated:::date-rfc3339%,%syslogpriority-
text%,%syslogfacility-text%:%msg%\n"
```

If you then use **RHQfmt** in the syslog configuration so it writes messages out in that format, JBoss ON understands the log messages fully. For example:

```
$template RHQfmt,"%timegenerated:::date-rfc3339%,%syslogpriority-
text%,%syslogfacility-text%:%msg%\n"
*.* /var/log/messages-for-rhq;RHQfmt
*.* @@127.0.0.1:5514;RHQfmt
```

That both writes syslog messages to **/var/log/messages-for-rhq** and sends the messages over TCP to a listener on port 5514, as configured in the platform's connection properties.

Q: **Executing a script resource fails on Red Hat Enterprise Linux.**

A: If invoking the **Execute** operation on a script resource, it immediately fails with an error message saying that the script cannot be executed, then ensure that the script itself is executable. Set the script to execute:

```
chmod a+x scriptname
```

A.9. JBoss Resources

Q: **Why does only one JBoss AS server show green availability and all the rest show red, even though I made sure all of my JNP credentials are configured properly in my resources' connection properties?**

A: There is a problem in the way the JBoss AS JNP client works. If you are managing multiple JBoss AS servers on a single box, all of your security credentials for those servers must be the same (i.e. the JNP username and password must be the same).

Q: **When I import a server like JBoss EAP 5 or Tomcat, I see its child JVM resource in inventory, but it is red (DOWN). Why?**

A: If a server is started with JMX remoting enabled *and secured*, the agent cannot connect to the JMX server because it cannot detect the proper credentials.

For example, if the JMX server has these system properties:

```
-Dcom.sun.management.jmxremote
-Dcom.sun.management.jmxremote.port=5222
-Dcom.sun.management.jmxremote.ssl=false
-Dcom.sun.management.jmxremote.authenticate=true
-Dcom.sun.management.jmxremote.password.file=/jmxremote.password
-Dcom.sun.management.jmxremote.access.file=/jmxremote.access
```

The agent's JMX plug-in examines the command line for the JMX server's process, sees that the JMX server is remoted and secured, and tries to set up its secure, remote JMX connector. Because the agent does not have the appropriate credentials, it cannot connect to the remote

JMX MBean server and assumes it is in a DOWN state.

Edit the resource's **`Connections Settings`**, under the resource's **`Inventory`** tab, to enter the valid username and password set in the JMX remote access files. This enables JBoss ON to go through the remote JMX endpoint.

Alternatively, JBoss ON can connect to the parent resource, and then use that to connect to the JMX server. In that case, in the **`Connection Settings`** subtab, unset all of the connection properties *except* for the **`Type`** property, which should be set to **`Parent`**. The parent of the JVM, the JBoss EAP resource, can provide the information to connect to the JVM.

Q: When trying to monitor a JBoss EAP instance, I get the error "Connection failure Failed to authenticate principal=null, securityDomain=jmx-console."

A: As explained in the JBoss EAP documentation and the JBoss EAP 4.3 documentation, the jmx-console is secured by default. Define the username and password as instructed in the EAP documentation. Then add this username and password under the **`Inventory`** > **`Configuration Properties`** page of the JBoss EAP instance.

NOTE

Starting a JBoss EAP instance without specifying a configuration parameter (**`-c`**) starts the instance in production configuration.

Q: When monitoring a JBoss AS instance, I'm not seeing any JVM resources beneath it.

A: For JBoss ON to discover JVM resources for a JBoss AS resource, the corresponding JBoss AS instance needs to be running Java 5 or later. It also needs to have been started with the **`jboss.platform.mbeanserver`** system property set. For example, in Red Hat Enterprise Linux, the **`${JBOSS_HOME}\bin\run.conf`** file should have this setting:

```
JAVA_OPTS="$JAVA_OPTS -Djboss.platform.mbeanserver"
```

Q: Can I monitor JBoss AS 5.1?

A: No. There are problems in the JBoss AS 5.1 profile service which prevent the agent from discovering it.

You can monitor JBoss EAP 5.0 and later versions and JBoss AS 6.0.

Q: My agent can detect my JBoss server and gets its connection properties, but the JNP connection fails. Why?

A: This primarily happens on Windows if the agent is installed in a directory with spaces in the pathname, such as **`C:\Program Files`**.

If the agent can detect the JBoss server and can find all of its connection properties, then check the logs for failures to connect to the profile service:

```
2012-01-12 15:03:38,982 DEBUG [ResourceContainer.invoker.daemon-1]
(org.rhq.plugins.jbossas5.ApplicationServerComponent)- Failed to
```

```
connect to Profile Service.
java.lang.RuntimeException: Failed to lookup JNDI name
'ProfileService' from InitialContext.
 at
org.rhq.plugins.jbossas5.connection.AbstractProfileServiceConnectionPr
ovider.lookup(AbstractProfileServiceConnectionProvider.java:84)

[snip]
```

Move the agent to a directory without spaces in the pathname and then re-discover the JBoss resources.

A.10. Postgres Resources

Q: Why is the agent showing an error in my PostgreSQL discovery about authentication failed for user "postgres"?

A: The Postgres plug-in attempts to log into the database server using the username and password of postgres. In many installations, this is a default superuser and will work. However, it is also possible that this login could fail for several different reason:

The postgres user has been deleted.

The password for the postgres user has been changed.

On Linux, the administrative login has been set to **`ident sameuser`**.

To resolve this:

Inventory the discovered Postgres resource. Its availability will show as down and it will not find any child resources.

Navigate to the **`INVENTORY`** tab for the Postgres resource.

Under **`Connection Properties`**, click the **`Edit`** button.

Change the role name and password fields to reflect a valid super user account on the Postgres instance.

Additionally, Postgres may need to be changed on Red Hat Enterprise Linux systems to allow password based logins by changing settings in the **`pg_hba.conf`** file.

Q: Why are most of the metrics for my Postgres resource showing up as NaN?

A: In many installations, Postgres will not start its statistics collector by default. To enable statistics collection, add or change the following line in the **`postgres.conf`** file:

```
stats_start_collector = on
```

Q: How many database connections are necessary to monitor a Postgres database?

A: Each Postgres database inventoried in JBoss ON requires one connection.

Q: **Why can't I drop my database that is inventoried in JBoss ON?**

A: With the frequency of availability and statistics monitoring, the Postgres plug-in keeps an open connection to the database. When attempting to drop a database currently inventoried in JBoss ON, an error is thrown about the database being in use. To drop the database, the JBoss ON agent monitoring the database must be shut down or the database resource should be removed from JBoss ON. This will close the postgres plug-in's connection to the Postgres server and allow you to drop the database.

A.11. Apache Resources

Q: **Where can I get the Apache connectors?**

A: The Apache plug-in monitors an Apache Web server through custom modules like the SNMP connector. The connectors can be downloaded from the JBoss ON server on the **`Downloads`** page in the GUI.

Q: **I have instrumented Apache with the Response Time module, but no RT metrics are being shown for my VirtualHosts.**

A: If there are no RT metrics being displayed for your VirtualHosts there are two things you should check:

1. Can the agent's system user read the **`_rt log`** files? For instance, under RHEL/Apache, the default permissions of **`/var/log/httpd`** are 700, root:

   ```
   ls -arltd /var/log/httpd/
   drwx------ 2 root root 4096 Jul 28 11:36 /var/log/httpd/
   ```

   A workaround is to specify an alternate log directory for the httpd logs or, alternatively, to change the permissions of **`/var/log/httpd`**. Both of these have specific security implications. You could also run the agent as root; while this is the least preferable option, there are cases where this is necessary in order to not compromise system security by modifying file permissions. For example, JBoss ON cannot monitor the postgres daemon without root permissions, due to 700, postgres permissions on its data directory. Since those permissions shouldn't be altered, the only remaining option is to run the agent as root.

2. Have you enabled the Response Time Metric for the Apache Vhost Template and enabled Response Time? It is disabled by default.

To do this, go to:

1. **`Administration`** > **`System Configuration`** > **`Templates | Apache HTTP Server`** > **`Apache Virtual Host`**, and click **`Edit Metric Template`**.

2. Select the checkbox next to **`HTTP Response Time`**.

3. At the bottom of the page, select **`Update schedules for existing resources of marked type`**.

4. Set the collection interval.

5. Click the Go button (**`[>]`**).

NOTE

The RT metrics will now work. It may take approximately 10 minutes for the metrics to appear.

Q: Some of my Apache metrics show values of zero. Why?

A: Three metrics show values of zero when you are monitoring with the SNMP module:

Bytes Received for GET Requests per Minute

Bytes Received for POST Requests per Minute

Total Number of Bytes Received per Minute

This is because of how SNMP interprets information from the request body. First, SNMP provides various length values for the request body and a GET request does not have a body, so GET responses are not calculated and, therefore, have a value of zero. Second, Apache does not calculate a request body size if there is request chunking.

Q: What is the Augeas plug-in?

A: The Augeas plug-in is an abstract plug-in that exists solely as an extension point for other plug-ins. The Augeas plug-in provides the Java JNI classes necessary for other dependent plug-ins to use to access the Augeas native library. For example, the OpenSSHD plug-in depends on the Augeas plug-in because it uses the Augeas library to access the OpenSSH daemon configuration. Several other JBoss ON plug-ins use this Augeas plug-in:

hosts

grub

apt

Q: Why does my agent log have the error "java.lang.UnsatisfiedLinkError: Unable to load library 'augeas': libaugeas.so: cannot open shared object file: No such file or directory"?

A: This means that an Augeas-based plug-in has been deployed, but the Red Hat Enterprise Linux box does not have the Augeas native library installed. See http://augeas.net for more information on installing Augeas libraries.

Q: Why does my Apache SNMP module fail to start with an error?

A: There are a couple of common errors and causes.

Error	Cause
Syntax error on line 1376 of /etc/httpd/conf/httpd.conf: Unable to write to SNMPvar directory" (on stderr)	Ensure the directory specified via the "SNMPVar" directive exists and is writable by the user that owns the Apache process.

Error	Cause
init_master_agent: Invalid local port (Permission denied)" (in the error_log file)	See if your Apache error_log contains a log message similar to the following: *[notice] SELinux policy enabled; httpd running as context user_u:system_r:httpd_t:s0* This means the SELinux (Security-Enhanced Linux) policy is preventing the httpd process from binding to the SNMP agent port, 1610 by default. To resolve the problem, change SELinux to permissive mode by running the command **`/usr/bin/setenforce 0`** and then restarting Apache. You should then see a message similar to the following in your error_log: *[notice] SELinux policy enabled; httpd running as context user_u:system_r:unconfined_t"* This message has the term unconfined_t. This indicates SELinux is no longer restricting the process.

A.12. Tomcat Resources

Q: I get the error "This resource's configuration has not yet been initialized" when I go to the Configuration tab for a Tomcat resource. Why?

A: Specifically, the error occurs when going to the **Tomcat** → **JVM** → **Logging** resource and attempting to open the **`Configuration`** tab.

This means that configuration management has not been enabled for the Tomcat resource. This can be done by going to the Tomcat server's **Inventory** tab, opening the **Connections** subtab, and enabling configuration management explicitly.

A.13. Provisioning and Content

Q: When I try to create a bundle by uploading a Ant recipe XML directly, the XML content seems to get corrupted and tags are placed out of order.

A: If you file upload a ANT script as the recipe, you can't use XML notation like **`<property name="a" />`**. It has to have an explicitly defined closing tag, like **`<property name="a"></property>`**. If you don't want to revise your Ant script XML file, copy and paste the recipe directly into the text field instead of uploading the file.

Q: What does the JBoss Patch Content feature of JBoss ON actually do? Is it completely automated?

A: The patch process updates existing jar/class files with upgraded jar/class files that are contained in the patch zip. Some changes may need to be completed manually, such as any *Not Performed* steps. If your configuration does not include one of the jars to be patched, then that step is skipped.

The patching process does not explicitly care about what the server configuration profile is called or which base configuration it is derived from.

Patching JBoss servers, and other methods of streaming content to JBoss ON resources, is covered in the *Basic Admin Guide*.

A.14. Alerts

Q: I just created an alert definition, and I know that my agent reported data that should have fired an alert immediately. But I don't see an alert. Why not?

A: After an alert definition is created, it takes a few seconds to be inserted into the JBoss ON server alert caches and then propagated throughout the JBoss ON server cloud. An alert won't be fired until that alert definition is in the server alert cache.

When the alert definition is inserted into the cache, a message is recorded in the JBoss ON server logs:

```
INFO  [CacheConsistencyManagerBean] localhost took [51]ms to reload
global cache
INFO  [CacheConsistencyManagerBean] localhost took [49]ms to reload
cache for 1 agents
```

It generally takes around 30 seconds for an alert definition to be added to the cache. Wait at least a minute after creating a definition before checking if it fires an alert.

Q: Why do I see alerts triggered on different metric values on different alert definition conditions when they are using the same metric?

A: This can occur due to the nature of how alert conditions are processed when measurement data comes in from the agent. This happens if a single alert definition has multiple conditions that use the same metric and that alert definition uses the "ALL" conjunction. For example, if an alert definition has one condition for "alert if metric X is greater than 5" *and* a separate condition for "alert if metric X less than 10."

The alert condition range works around this by doing range checking. For more information, see Bugzilla 735262.

A.15. Monitoring

Q: Why does the Events tab not capture the events of one of the "Log Event Sources"?

A: When you define an event source you need to ensure three things:

It is enabled

The path to the log is valid; and

The selected date format matches what you have in the logs

If you select the default date format when creating the event source, and that format does not match what you have in the logs, nothing is captured for the log events.

For example, if you select the default date format 2013-07-14 15:36:25,075 and the logs have 2013.08.27-09:46:34, then no log event is captured.

Specify a different date format according to `java.text.SimpleDateFormat`.

Q: When do baselines auto-calculate?

A: Go to the **`Administration`** page of the JBoss ON GUI and click the **`Server Configuration`** link. You will see settings for **`Automatic Baseline Configuration Properties`**. **`Baseline Frequency`** determines how often the baselines are calculated. The default is 3 days. This means that every 3 days a new set of baselines are calculated, except for those that were manually set by the user. These remain pinned to the baselines set by the user.

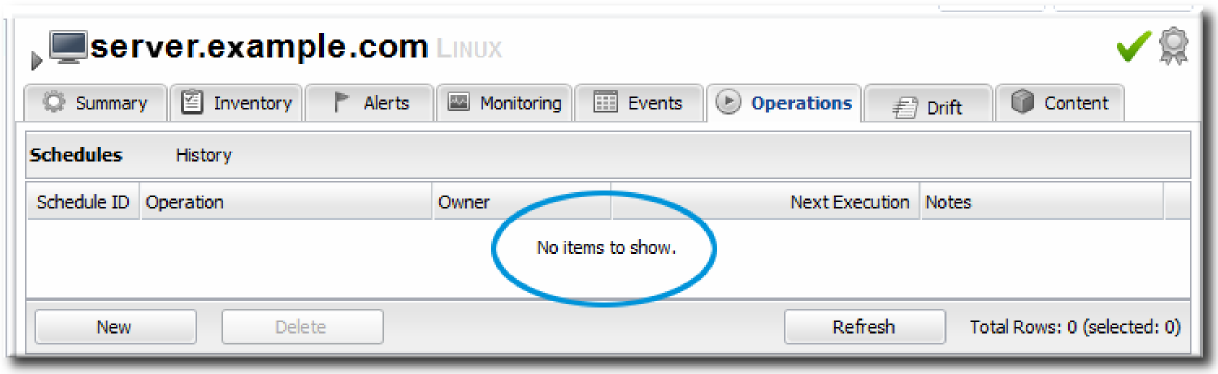
`Baseline Dataset` determines the minimum set of data that must have been collected for a measurement before a baseline for that measurement is calculated. The default is 7 days. For example, when it is determined that baselines should be calculated, every third day by default, only those measurements that have data that is 7 days old or older will have a baseline calculated. Any measurements that do not have data from 7 days ago are skipped. This ensures that when a measurement's baseline is calculated, you have a good representative set of data to include in the calculation (e.g. by default, you will have 7 days worth of data that are included in the baseline calculation).

A.16. Operations

Q: I clicked the Operations tab, but I don't see any available operations, and it says "No items to show." How do I schedule an operation?

A: The **`Schedules`** tab shows a list of scheduled operations, meaning operations which are configured but have not yet been run. If there are no scheduled operations, then the tab has a description that reads *No items to show.*

That does not mean that there are no operations available for the resource; it only means that no operations have been scheduled.



Schedule ID	Operation	Owner	Next Execution	Notes

No items to show.

Total Rows: 0 (selected: 0)

To schedule an operation, click the **New** in the lower left corner of the **`Operations`** tab, and fill in the operation information.

INDEX