Red Hat Insights 2021

Monitoring and Reacting to Configuration Changes Using Policies

How to create policies to detect inventory configuration changes and send email notifications
Red Hat Insights 2021 Monitoring and Reacting to Configuration Changes Using Policies

How to create policies to detect inventory configuration changes and send email notifications

Red Hat Customer Content Services
Abstract

This document provides an overview of the Policies service and explains how to create a policy to detect system configuration changes and be notified by email. Providing Feedback: If you have a suggestion to improve this document or find an error, submit a Bugzilla report at http://bugzilla.redhat.com against Cloud Software Services (cloud.redhat.com) for the Policies component.
# Table of Contents

CHAPTER 1. RECEIVING AUTOMATIC NOTIFICATIONS FROM POLICIES ABOUT CHANGES IN YOUR INSIGHTS INVENTORY ................................................................. 3  
  1.1. POLICIES DETECTION AND NOTIFICATION OF INVENTORY CONFIGURATION CHANGES 3

CHAPTER 2. USER PREFERENCES .............................................................................. 4  
  2.1. SETTING USER PREFERENCES 4

CHAPTER 3. CREATING POLICIES ............................................................................. 5  
  3.1. CREATING A POLICY TO ENSURE PUBLIC CLOUD PROVIDERS ARE NOT OVER PROVISIONED 5  
  3.2. CREATING A POLICY TO DETECT IF SYSTEMS ARE RUNNING AN OUTDATED VERSION OF RHEL 6  
  3.3. CREATING A POLICY TO DETECT A VULNERABLE PACKAGE VERSION BASED ON RECENT CVE 7

CHAPTER 4. REVIEWING AND MANAGING POLICIES ............................................. 8

CHAPTER 5. APPENDIX ............................................................................................. 9  
  5.1. SYSTEM FACTS 9  
  5.2. OPERATORS 11
CHAPTER 1. RECEIVING AUTOMATIC NOTIFICATIONS FROM POLICIES ABOUT CHANGES IN YOUR INSIGHTS INVENTORY

Policies evaluate system configurations in your environment, and can send notifications about those changes. As soon as such changes, or potential security issues, occur, it uploads the data to cloud.redhat.com. When conditions you have set in your policies exist, the policies trigger actions you have defined, such as notifying you by email about the changes.

1.1. POLICIES DETECTION AND NOTIFICATION OF INVENTORY CONFIGURATION CHANGES

Policies are applicable to all systems registered in your Insights inventory. You can create and manage policies using the Insights user interface or via API.

Policies can assist you by managing tasks such as:

- Raising an alert when particular conditions occur in your system configuration.
- Emailing a team when security packages are out of date on a system.

Using policies to monitor configuration changes in your inventory and notifying by email requires:

- Setting user email preferences (if not already set).
- Creating a policy to detect configuration changes, and selecting email as the trigger action.

**NOTE**

- Use the Role Based Access Control (RBAC) capability in https://cloud.redhat.com (Settings > User access) to control user access for Policies.
- See Role Based Access Control for Red Hat Insights and cloud management services for Red Hat Enterprise Linux for more information about this feature and example use cases.
CHAPTER 2. USER PREFERENCES

Update your information and set email preferences for cloud.redhat.com services in your user preferences.

2.1. SETTING USER PREFERENCES

You can set or update your email preferences as follows.

Procedure

1. Click the user menu located on the upper-right side, then go to User preferences → Email preferences. The Email preferences screen opens.

   Alternatively, on the {PRODUCTNAME} dashboard, in the left-side navigation panel at the top, click “Red Hat Insights”, and then click “User Preferences”. The Email preferences screen opens.

2. Depending on your email notification preference, you can subscribe to instant notification emails for each system with triggered policies and/or Daily digest (summary) of all systems with triggered policies. On this page, you can also select your preference for other https://cloud.redhat.com emails you want to receive.

   NOTE

   Subscribing to instant notification can result in receiving many emails on large inventories, that is, one email per system checking in.

3. Click Submit.
CHAPTER 3. CREATING POLICIES

The following workflow examples explain how to create several types of policies that detect system configuration changes and send notification of the changes by email.

NOTE

When creating a policy, if you see a warning message that you have not opted in for email alerts, set your preferences to receive email from your policies. See Chapter 2, User preferences, for information.

3.1. CREATING A POLICY TO ENSURE PUBLIC CLOUD PROVIDERS ARE NOT OVER PROVISIONED

Procedure

1. In the cloud.redhat.com platform, under Red Hat Insights click Policies.

2. Click Create policy.

3. On the Create a policy page, click From scratch or As a copy of existing Policy as required. Note that the As a copy of existing Policy option will prompt you to select a policy from the list of existing policies to use as a starting point.

4. Click Next.

5. On the Conditions screen, enter a Name and Description for the policy.

6. Click Next.

7. Enter Condition. In this case, enter: facts.cloud_provider in ['alibaba', 'aws', 'azure', 'google'] and (facts.number_of_cpus >= 8 or facts.number_of_sockets >= 2). This condition will detect if an instance running on the specified public cloud providers are running with CPU hardware.
higher than the allowed limit.

NOTE

You can expand What condition can I define? and/or Review available system facts to view an explanation of conditions you can use, and see the available system facts, respectively. In this section are examples of syntax you can use.

8. Click Validate condition.

9. Once the condition is validated, click Next.

10. On the Trigger actions page, click Add trigger actions and select Email.

11. Click Next.

NOTE

On the Trigger actions page, you can also enable email alerts as well as open email preferences.

12. On the Review and enable page, click the toggle switch to activate the policy and review its details.

13. Click Finish.

Your new policy is created. When the policy is evaluated on a system check-in, if the condition in the policy is met, Policies automatically sends an email to all users on the account with access to Policies, depending on their email preferences.

3.2. CREATING A POLICY TO DETECT IF SYSTEMS ARE RUNNING AN OUTDATED VERSION OF RHEL

You can create a policy that detects if systems are running outdated versions of RHEL and notifies you by email about what it finds.

Procedure

1. In the cloud.redhat.com platform, click Policies under Red Hat Insights.

2. Click Create policy.

3. On the Create policy page, click From scratch or As a copy of existing Policy as required. Note that the As a copy of existing Policy option prompts you to select a policy from the list of existing policies to use as a starting point.

4. Click Next.

5. Enter a Name and Description for the policy.

6. Click Next.

7. Enter Condition. In this case, enter facts.os_release < 8.1 This condition will detect if systems still run an outdated version of our operating system based on RHEL 8.1.
8. Click Validate condition, then click Next.

9. On the Trigger actions page, click Add trigger actions and select Email.

10. Click Next.

11. On the Review and activate page, click the toggle switch to activate the policy and review its details.

12. Click Finish.

Your new policy is created. When the policy is evaluated on a system check-in, if the condition in the policy is met, Policies automatically sends an email to all users on the account with access to Policies, depending on their email preferences.

3.3. CREATING A POLICY TO DETECT A VULNERABLE PACKAGE VERSION BASED ON RECENT CVE

You can create a policy that detects vulnerable package versions based on recent CVE and notifies you by email about what it finds.

Procedure

1. In the cloud.redhat.com platform, under Red Hat Insights, click Policies.

2. Click Create policy.

3. On the Create Policy page, click From scratch or As a copy of existing Policy as required. Note that the As a copy of existing Policy option will prompt you to select a policy from the list of existing policies to use as a starting point.

4. Click Next.

5. Enter a Name and Description for the policy.

6. Click Next.

7. Enter Condition. In this case, enter facts.installed_packages contains ['openssh-4.5'] This condition will detect if systems still run a vulnerable version of an openssh package based on recent CVE.

8. Click Validate condition, then click Next.

9. On the Trigger actions page, click Add trigger actions and select Email.

10. Click Next.

11. On the Review and activate page, click the toggle switch to activate the policy and review its details.

12. Click Finish.

Your new policy is created. When the policy is evaluated on a system check-in, if the condition in the policy is met, Policies automatically sends an email to all users on the account with access to Policies, depending on their email preferences.
CHAPTER 4. REVIEWING AND MANAGING POLICIES

You can review and manage all created policies (enabled and disabled) by clicking Policies on the left-side menu in Red Hat Insights.

You can filter the list of policies by name and by active state. You can click the options menu next to a policy to perform the following operations:

- Enable and disable
- Edit
- Duplicate
- Delete

Additionally, you can perform the following operations in bulk by selecting multiple policies from the list of policies and clicking the options menu located next to the Create policy button at the top:

- Delete policies
- Enable policies
- Disable policies

**NOTE**

If you see a warning message about email alerts not opted in, set your preferences to receive email from your policies as described in Chapter 2, User preferences.
CHAPTER 5. APPENDIX

This appendix contains the following reference materials:

- System Facts
- Operators

5.1. SYSTEM FACTS

Table 5.1. System Facts and Their Functions

<table>
<thead>
<tr>
<th>Fact Name</th>
<th>Description</th>
<th>Example Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>arch</td>
<td>System architecture</td>
<td>x86_64</td>
</tr>
<tr>
<td>bios_release_date</td>
<td>BIOS release date; typically MM/DD/YYYY</td>
<td>01/01/2011</td>
</tr>
<tr>
<td>bios_vendor</td>
<td>BIOS vendor name</td>
<td>LENOVO</td>
</tr>
<tr>
<td>bios_version</td>
<td>BIOS version</td>
<td>1.17.0</td>
</tr>
<tr>
<td>cloud_provider</td>
<td>Cloud vendor. Values are google, azure, aws, alibaba, or empty</td>
<td>google</td>
</tr>
<tr>
<td>cores_per_socket</td>
<td>Number of CPU cores per socket</td>
<td>2</td>
</tr>
<tr>
<td>cpu_flags</td>
<td>Category with a list of CPU flags. Each name is the CPU flag (ex: vmx), and the value is always enabled.</td>
<td>vmx, with a value of enabled.</td>
</tr>
<tr>
<td>enabled_services</td>
<td>Category with a list of enabled services. Each name in the category is the service name (ex: crond), and the value is always enabled.</td>
<td>crond, with a value of enabled.</td>
</tr>
<tr>
<td>fqdn</td>
<td>System Fully Qualified Domain Name</td>
<td>system1.example.com</td>
</tr>
<tr>
<td>infrastructure_type</td>
<td>System infrastructure; common values are virtual or physical</td>
<td>virtual</td>
</tr>
<tr>
<td>infrastructure_vendor</td>
<td>Infrastructure vendor; common values are kvm, vmware, baremetal, etc.</td>
<td>kvm</td>
</tr>
<tr>
<td>installed_packages</td>
<td>List of installed RPM packages. This is a category.</td>
<td>bash, with a value of 4.2.46-33.el7.x86_64.</td>
</tr>
<tr>
<td>Fact Name</td>
<td>Description</td>
<td>Example Value</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>installed_services</td>
<td>Category with a list of installed services. Each name in the category is the service name (ex: crond), and the value is always installed.</td>
<td>crond, with a value of installed.</td>
</tr>
<tr>
<td>kernel_modules</td>
<td>List of kernel modules. Each name in the category is the kernel module (ex: nfs), and the value is enabled.</td>
<td>nfs, with a value of enabled.</td>
</tr>
<tr>
<td>last_boot_time</td>
<td>The boot time in YYYY-MM-DDTHH:MM:SS format. Informational only; we do not compare boot times across systems.</td>
<td>2019-09-18T16:54:56</td>
</tr>
<tr>
<td>network_interfaces</td>
<td>List of facts related to network interfaces.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>There are six facts for each interface: ipv6_addresses, ipv4_addresses, mac_address, mtu, state and type. The two address fields are comma-separated lists of IP addresses. The state field is either UP or DOWN. The type field is the interface type (ex: ether, loopback, bridge, etc.).</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Each interface (ex: lo, em1, etc) is prefixed to the fact name. For example, em1’s mac address would be the fact named em1.mac_address.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Most network interface facts are compared to ensure they are equal across systems. However, ipv4_addresses, ipv6_addresses, and mac_address are checked to ensure they are different across systems. A subexception for lo should always have the same IP and mac address on all systems.</td>
<td></td>
</tr>
<tr>
<td>number_of_cpus</td>
<td>Total number of CPUs</td>
<td>1</td>
</tr>
<tr>
<td>number_of_sockets</td>
<td>Total number of sockets</td>
<td>1</td>
</tr>
<tr>
<td>os_kernel_version</td>
<td>Kernel version</td>
<td>4.18.0</td>
</tr>
<tr>
<td>os_release</td>
<td>Kernel release</td>
<td>8.1</td>
</tr>
<tr>
<td>running_processes</td>
<td>List of running processes. The fact name is the name of the process, and the value is the instance count.</td>
<td>crond, with a value of 1.</td>
</tr>
<tr>
<td>sap_instance_number</td>
<td>SAP instance number</td>
<td>42</td>
</tr>
</tbody>
</table>
### Fact Name | Description | Example Value
--- | --- | ---
**sap_sids** | SAP system ID (SID) | A42
**sap_system** | Boolean field that indicates if SAP is installed on the system | True
**sap_version** | SAP version number | 2.00.052.00.1599235305
**satellite Managed** | Boolean field that indicates if a system is registered to a Satellite server | FALSE
**selinux_current_mode** | Current SELinux mode | enforcing
**selinux_config_file** | SELinux mode set in the config file | enforcing
**system_memory** | Total system memory in human-readable form | 3.45 GiB
**tuned_profile** | Current profile resulting from the command tuned-adm active | desktop
**yum_repos** | List of yum repositories. The repository name is added to the beginning of the fact. Each repository has the associated facts base_url, enabled, and gpgcheck | Red Hat Enterprise Linux 7 Server (RPMs).base_url I would have the value https://cdn.redhat.com/content/dist/rhel/server/7/$releasever/$basearch/os

### 5.2. OPERATORS

**Table 5.2. Available Operators in Conditions**

<table>
<thead>
<tr>
<th>Operators</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logical Operators</td>
<td>AND</td>
</tr>
<tr>
<td></td>
<td>OR</td>
</tr>
<tr>
<td>Boolean Operators</td>
<td>EQUAL</td>
</tr>
<tr>
<td></td>
<td>NOTEQUAL</td>
</tr>
<tr>
<td>Operators</td>
<td>Value</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>--------</td>
</tr>
<tr>
<td>Numeric Compare Operators</td>
<td>GT</td>
</tr>
<tr>
<td></td>
<td>GTE</td>
</tr>
<tr>
<td></td>
<td>LT</td>
</tr>
<tr>
<td></td>
<td>LTE</td>
</tr>
<tr>
<td>String Compare Operator</td>
<td>CONTAINS</td>
</tr>
<tr>
<td>Array Operators</td>
<td>IN</td>
</tr>
<tr>
<td></td>
<td>CONTAINS</td>
</tr>
<tr>
<td>Parser Operators</td>
<td>OR</td>
</tr>
<tr>
<td></td>
<td>AND</td>
</tr>
<tr>
<td></td>
<td>NOT</td>
</tr>
<tr>
<td></td>
<td>EQUAL</td>
</tr>
<tr>
<td></td>
<td>NOTEQUAL</td>
</tr>
<tr>
<td></td>
<td>CONTAINS</td>
</tr>
<tr>
<td></td>
<td>NEG</td>
</tr>
</tbody>
</table>