Comparing System Configurations and Baselines in Red Hat Insights Inventory

Using drift service comparisons and baselines for system analysis
Using drift service comparisons and baselines for system analysis
Abstract

Compare system configurations of a system over time, or to other systems and baselines to identify discrepancies in your environment, and perform drift analysis.
# Table of Contents

- **MAKING OPEN SOURCE MORE INCLUSIVE** .................................................. 3
- **PROVIDING FEEDBACK ON RED HAT HYBRID CLOUD CONSOLE DOCUMENTATION** ......................... 4

## CHAPTER 1. USING COMPARISONS IN YOUR INSIGHTS FOR RHEL INVENTORY TO PERFORM DRIFT ANALYSIS ........................................ 5
  1.1. COMPARING SYSTEM CONFIGURATIONS IN YOUR INSIGHTS FOR RHEL INVENTORY .................. 5

## CHAPTER 2. VIEWING DRIFT DATA IN THE INSIGHTS FOR RED HAT ENTERPRISE LINUX APPLICATION ................................. 6
  2.1. ACCESSING THE DRIFT SERVICE ............................................. 6

## CHAPTER 3. CREATE AND APPLY COMPARISONS ........................................ 8
  3.1. USE COMPARISONS OF SYSTEMS IN YOUR INSIGHTS FOR RHEL INVENTORY ................. 8
    3.1.1. Creating and adding to comparisons ........................................... 8

## CHAPTER 4. REVIEWING AND MANAGING BASELINES .................................. 10
  4.1. THE ROLE OF FACTS IN BASELINES ........................................... 10
  4.2. NEW BASELINES CREATION ........................................... 10
    4.2.1. Creating baselines from scratch ........................................... 10
    4.2.2. Copying an existing baseline to create a new baseline .............. 11
    4.2.3. Copying an existing system to create a new baseline .......... 12
    4.2.4. Copying a historical system profile to create a new baseline .... 13
  4.3. EDITING OR DELETING BASELINES ........................................... 13
    4.3.1. Editing a baseline ........................................... 13
    4.3.2. Deleting a baseline ........................................... 14
  4.4. COMPARING SYSTEM CONFIGURATIONS TO BASELINES ...................... 14
  4.5. EXPORTING BASELINES ........................................... 15
    4.5.1. Exporting individual baselines, facts, and categories .......... 15
    4.5.2. Exporting all system baselines ........................................... 15

## CHAPTER 5. USING SYSTEM FACTS IN COMPARISONS .................................. 17
  5.1. THE ROLE OF SYSTEM FACTS IN COMPARISONS ................................ 17
  5.2. HOW FACTS ARE USED IN COMPARISONS ........................................ 17
  5.3. FILTERING SYSTEM FACTS USING THE USER INTERFACE (UI) ............ 17
  5.4. FILTERING SYSTEM FACTS USING UNIFORM RESOURCE LOCATOR (URL) PARAMETERS ........... 19
  5.5. SORTING SYSTEM FACTS ........................................... 20
  5.6. OBFUSCATED VALUE HIGHLIGHTS ........................................... 21
  5.7. UNDERSTANDING MULTI-VALUE FACTS ........................................ 22
  5.8. USING MULTI-FACT FILTERING ........................................... 22
  5.9. AVAILABLE FACTS AND THEIR FUNCTIONS .................................... 23

## CHAPTER 6. COMPARING SYSTEMS AND SYSTEM PROFILES ............................... 27
  6.1. COMPARING SYSTEM PROFILES ........................................... 27
  6.2. COMPARING SYSTEMS TO A REFERENCE POINT ................................ 27
    6.2.1. Comparing systems to a single point of reference ............... 27
  6.3. USING THE HISTORY OF SYSTEM PROFILES .................................... 27
  6.4. EXPORTING SYSTEM COMPARISON OUTPUT .................................... 28
    6.4.1. Exporting system comparison output to a CSV or JSON file ...... 28
MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see our CTO Chris Wright’s message.
PROVIDING FEEDBACK ON RED HAT HYBRID CLOUD CONSOLE DOCUMENTATION

We appreciate your input on our documentation. Please let us know how we could make it better. To do so, create a Bugzilla ticket:

1. Go to the Bugzilla website.
2. As the Component, use Documentation.
3. Fill in the Description field with your suggestion for improvement. Include a link to the relevant part(s) of documentation.
4. Click Submit Bug.
CHAPTER 1. USING COMPARISONS IN YOUR INSIGHTS FOR RHEL INVENTORY TO PERFORM DRIFT ANALYSIS

You can analyze and troubleshoot components of your Insights for RHEL inventory and perform drift analysis by examining and comparing your system configurations.

1.1. COMPARING SYSTEM CONFIGURATIONS IN YOUR INSIGHTS FOR RHEL INVENTORY

The drift service enables you to compare the system configuration of one system to other systems in your Insights for RHEL inventory. You can also compare the system configurations over time to track and analyze changes. In addition, you can set a reference point and compare all systems against that reference. If needed, you can use any of these comparisons to troubleshoot systems at various points in time.

When comparing across different systems, you can filter profile facts and system tags to highlight the ones that match, those that are different, and also see where information is missing or considered problematic.

You can also define and manage baselines. You can use defined baselines in your Insights for RHEL inventory to compare baselines and system configurations. In addition, you can generate a comma-separated values (CSV), or JavaScript Object Notation (JSON) output of the systems and baselines you are comparing.
CHAPTER 2. VIEWING DRIFT DATA IN THE INSIGHTS FOR RED HAT ENTERPRISE LINUX APPLICATION

2.1. ACCESSING THE DRIFT SERVICE

Prerequisites

1. For systems to be available to upload to the drift service, the Insights client must be installed and operational on the systems.

2. No additional User Access privileges are required.

The drift service is part of Insights for Red Hat Enterprise Linux. Access this service via Red Hat Hybrid Cloud Console.

NOTE

For steps on how to install the Insights client and register your systems to Insights for RHEL, see Insights for Red Hat Enterprise Linux, Get Started instructions.

Procedure

1. Navigate to the Red Hat Enterprise Linux > Drift > Comparison page. The Comparison screen opens.

2. Click Add to comparison. The Add to comparison screen opens, where you can add systems or baselines to compare.
On the Systems tab, you can add any systems you want to compare. This screen also lists any baselines that already exist in your Insights for RHEL inventory.

On the Baselines tab, you can create baselines. This screen also lists any baselines that already exist in your Insights for RHEL inventory.

3. Click Submit to add your systems or baselines for comparison.

**NOTE**

Baselines are defined as a standard configuration that a group of systems must maintain. Configurations are sets of name:value pairs, that can be created from scratch or copied from an existing system configuration.
CHAPTER 3. CREATE AND APPLY COMPARISONS

With Insights for RHEL, you can analyze and troubleshoot characteristics and performance of components in your Insights for RHEL inventory by viewing and making comparisons between them.

3.1. USE COMPARISONS OF SYSTEMS IN YOUR INSIGHTS FOR RHEL INVENTORY

Use system facts to create useful comparisons. Analyzing the facts about components in your Insights for RHEL inventory gives you a view of your system, or several of your systems, at a particular time. You can also create comparisons of your systems over time to learn about and analyze changes to systems.

Complete the following steps to create or add to comparisons in your Insights for RHEL inventory.

3.1.1. Creating and adding to comparisons

Complete the following steps to create or add comparisons in your Insights for RHEL inventory.

Prerequisites

1. For systems to be available to upload to the drift service, the Insights client must be installed and operational on the systems.

2. No additional User Access privileges are required.

The drift service is part of Insights for Red Hat Enterprise Linux. Access this service via Red Hat Hybrid Cloud Console.

NOTE

For steps on how to install the Insights client and register your systems to Insights for RHEL, see Insights for Red Hat Enterprise Linux, Get Started instructions.

Procedure

1. Navigate to the Red Hat Enterprise Linux > Drift > Comparison page. The Comparison screen opens.

![Comparison Screen]

Add systems or baselines to compare

You currently have no systems or baselines displayed. Add at least two systems or baselines to compare their facts.

Add to comparison
2. Click **Add to comparison**. The Add to comparison screen opens, where you can add systems or baselines to compare.

   **Add to comparison**

   ![Add to comparison screenshot]

   - **On the Systems tab**, you can add any systems you want to compare. This screen also lists any baselines that already exist in your Insights for RHEL inventory.

   - **On the Baselines tab**, you can create baselines. This screen also lists any baselines that already exist in your Insights for RHEL inventory.

3. Click **Submit** to add your systems or baselines for comparison.
CHAPTER 4. REVIEWING AND MANAGING BASELINES

There are many things you can do when reviewing and managing baselines, such as creating, copying, editing, deleting, exporting, or comparing them.

4.1. THE ROLE OF FACTS IN BASELINES

Baselines are defined as a standard configuration that a group of systems must maintain. Configurations are sets of name:value pairs, that can be created from scratch or copied from an existing system configuration.

You can manage system profile baseline definitions for your organization. That is, you can edit baselines by changing values on facts, or by deleting them. You can use the drift user interface to create, copy, edit, delete, and export baselines.

NOTE

You can also use the baselines API to create and manage baselines. Refer to the Insights for Red Hat Enterprise Linux API documentation for information about using the REST API to query and edit baselines.

4.2. NEW BASELINES CREATION

You can use a variety of system facts and choose the method you want to use to create a baseline. You can:

- Create a baseline from scratch
- Copy an existing baseline
- Copy an existing system
- Copy a historical system profile

4.2.1. Creating baselines from scratch

Baselines are defined as a standard configuration that a group of systems must maintain. Configurations are sets of name:value pairs, that can be created from scratch or copied from an existing system configuration. Complete the following steps to create new baselines:

Procedure


2. Click Create baseline. The Create baseline screen opens.
3. Select **Create baseline from scratch**

4. In the **Baseline name** field, enter a name for the baseline you want to create.

5. Click **Create baseline**. The Edit screen for the new baseline opens.

6. Click **Add fact or category**. The **Add fact** screen for the new baseline opens.

7. If this is a category (parent fact) under which you will add sub fact(s), select **This is a category** and only enter the **Category name**. Then click **Save**.

8. Otherwise, enter the **Fact name** and **Value**, then click **Save**.

9. To add sub facts under a category, click the options menu next to a category and select **Add sub fact**. You can also edit a category or delete a category, using the more options menu.

**NOTE**

Any changes performed are saved automatically. Therefore, once you have completed adding all facts, sub facts and their values, you can view and manage the baseline you have created by navigating to **Red Hat Enterprise Linux > Drift > Baselines**.

### 4.2.2. Copying an existing baseline to create a new baseline

You can copy an existing baseline to create a new baseline.

**Procedure**

1. Navigate to the **Red Hat Enterprise Linux > Drift > Baselines** page. The Baselines screen opens.

2. Click **Create baseline**. The Create baseline screen opens.

3. Select **Copy an existing baseline**.

4. In the **Baseline name** field, enter a name for the new baseline.
5. From the list of existing baselines, choose the baseline you want to copy from and click **Create baseline**. The screen that opens has the newly created baseline with all facts populated.

6. Edit or delete existing facts using the more options menu next to a fact.

7. You can also add a new fact or category by clicking **Add fact or category**. Similarly, you can add a value to a category.

Once you have completed adding or editing facts, and their values, you can view and manage the baseline you have created by navigating to the **Red Hat Enterprise Linux > Drift > Baselines** page.

**NOTE**

Based on typical expectations about fact behavior, the drift service alerts users to facts that may require attention. It does not flag facts that differ from each other if the differences are expected or are of no importance. It flags only important differences or similarities that may be problematic. You can then address these exceptions to the typical fact behavior.

### 4.2.3. Copying an existing system to create a new baseline

You can copy an existing system to create a new baseline.

**Procedure**

1. Navigate to the **Red Hat Enterprise Linux > Drift > Baselines** page. The Baselines screen opens.

2. Click **Create baseline**. The Create baseline screen opens.

3. Select **Copy an existing system**.

4. In the Baseline name field, enter a name for the new baseline.

5. From the existing list of systems, select the system you want to copy from and click **Create baseline**. The screen that opens has the newly created baseline with all facts populated.

6. To delete existing facts, use the **More options** menu to the right of the blue **Add fact or category** button.

7. To add a new fact or category, click **Add fact or category**. Similarly, you can add a sub fact to a category.

**NOTE**

Based on typical expectations about fact behavior, the drift service alerts users to facts that may require attention. It does not flag facts that differ from each other if the differences are expected or are of no importance. It flags only important differences or similarities that may be problematic. You can then address these exceptions to the typical fact behavior in a baseline.

Once you have completed adding or editing facts, sub facts and their values, you can view and manage the baseline you have created by navigating to **drift → Baselines**.
4.2.4. Copying a historical system profile to create a new baseline

You can copy a historical system profile to create a new baseline.

Procedure

2. Click Create baseline. The Create baseline screen opens.
3. Select Copy an existing system.
4. In the Baseline name field, enter a name for the new baseline.
5. From the existing list of systems, select the system you want to copy from. In the Historical profile column, click the Historical profile icon ( )
6. From the drop-down menu, select the profile you want to copy, based on the date and time stamp. After selection, drift displays the timestamp of the profile you selected under the name of the system in the Name column. If you change your mind and want to select a different historical profile, click the X next to the profile timestamp, and select a different historical profile.
7. When you are satisfied that you have the particular profile you want to copy, click Create baseline. A screen opens with the newly created baseline, with all facts populated.
8. To edit or delete existing facts, use the More Options menu next to a fact.
9. To add a new fact or category, click Add fact or category. Similarly, you can add a value to a fact.

Once you have completed adding or editing facts, and their values, you can view and manage the baseline you have created by navigating to Red Hat Enterprise Linux > Drift > Baselines.

NOTE

Based on typical expectations about fact behavior, the drift service alerts users to facts that may require attention. It does not flag facts that differ from each other if the differences are expected or are of no importance. It flags only important differences or similarities that may be problematic. You can then address these exceptions to the typical fact behavior in a baseline.

4.3. EDITING OR DELETING BASELINES

You can edit baselines by renaming the baseline, and changing, adding, or deleting facts, categories, and sub facts.

4.3.1. Editing a baseline

You can edit an existing baseline.

Procedure
1. To edit a baseline, click the baseline name. The Edit screen for the chosen baseline opens.

2. Click **Add fact or category** to add facts or categories to the baseline. The Add fact screen opens.

3. If this is a category (parent fact) under which you will add sub fact(s), select **This is a category** and only enter the **Category name**. Then click **Save**.

4. To edit facts, select the item you want to edit and click **Edit fact** on the more options on the right side of the line containing the fact. The Edit fact screen opens. Change the name or value and click **Save**.

5. To edit a baseline by deleting facts or categories, select the items you want to delete and click **Delete fact** on the More options on the right side of the line that contains the fact.

6. For fact categories, click the More options menu on the right side of the line that contains the fact category. Click to **Add sub fact**, **Edit category**, or **Delete category**, and the appropriate screen opens. Provide or edit the information requested and click **Save**.

### 4.3.2. Deleting a baseline

You can delete a baseline.

**Procedure**


2. Use the drop-down checkbox to select any or all baselines. When you make a selection, the field indicates the number of baselines you have selected.

3. To delete a baseline in the list, click **Delete** on the More options menu at the top of the screen. An alert opens with a reminder that deleting a baseline cannot be undone. If you still want to proceed, click **Delete baseline**.

### 4.4. COMPARING SYSTEM CONFIGURATIONS TO BASELINES

You can compare system configurations to one or more baselines to identify discrepancies in your environment and perform drift analysis.

**Procedure**

1. Navigate to the Red Hat Enterprise Linux > Drift > Comparison page. The Comparison screen opens.

2. Click **Add to comparison**.

3. On the **Systems** tab, select one or more systems from the list. Alternatively, you can search for a system by name, and then select the system.

4. On the **Baselines** tab, select one or more baselines from the list. Alternatively, you can search for a baseline by name, and then select the baseline.
5. Click **Submit**.

At any time, you can add more systems and baselines by clicking the **Add System** button on the right side of the systems already added for comparison.

Similarly, to remove a particular system or baseline under comparison, click the **X** sign in the upper right corner of the individual system or baseline name panel.

To remove all systems and baselines under comparison, click **Clear all comparisons** from the Options menu located at the top, and start again.

You can view the displayed comparison result and filter as necessary by fact name, comparison state, and category. To export the result, along with any current selections such as filters, to a comma-separated values (CSV) or JavaScript Object Notation (JSON) file for further analysis, click the **Export as CSV** or **Export as JSON** option in the menu.

### 4.5. EXPORTING BASELINES

You can export all system baselines or individual baseline facts or categories to a comma-separated values (CSV) or JavaScript Object Notation (JSON) file and analyze them externally. Note that any search filters applied are preserved in the downloaded CSV or JSON file.

#### 4.5.1. Exporting individual baselines, facts, and categories

You can export individual baselines, facts, and categories.

**Procedure**

1. Navigate to the **Red Hat Enterprise Linux > Drift > Baselines** page. The Baselines screen opens.

2. From the list of baselines, select a baseline from the list.

3. On the Edit screen for the baseline you selected, click the Export icon ( ) and choose **Export to comma-separated values (CSV)** or **Export to JavaScript Object Notation (JSON)**.

**NOTE**

Use this same procedure to export individual facts or categories to CSV or JSON.

#### 4.5.2. Exporting all system baselines

You can export all system baselines to a comma-separated values (CSV) or JavaScript Object Notation (JSON) file.

**Procedure**

1. Navigate to the **Red Hat Enterprise Linux > Drift > Baselines** page. The Baselines screen opens.

2. Click the Export icon ( ) next to the **Create baseline** button.
3. Choose Export to CSV or Export to JSON
CHAPTER 5. USING SYSTEM FACTS IN COMPARISONS

System facts are important components that help you understand your system comparisons. Examining them reveals detailed information about the performance and changes in your Insights for RHEL system inventory. System facts also alert you to system components whose state is unknown, as well as identifying parts of your systems that require attention.

5.1. THE ROLE OF SYSTEM FACTS IN COMPARISONS

Comparison states based on observed fact values provide guidance in managing your system. The application identifies facts whose behavior differs from expectations, facts whose state is unknown, and alerts users to facts that require attention.

The drift service displays the observed fact values in different colors.

- A red icon indicates an issue you should examine.

- A green icon denotes an expected state or value.

- A state shown in black, marked with a question mark icon, indicates the expected state is unknown.

Some facts are system specific and are considered unique. Their state is marked in red if values are equal for a given comparison. This is the case for fqdn and IP addresses. When marked in red, these facts require your attention.

The drift service expects other facts, such as last_boot_time, to be different for all compared systems. For such facts, it does not highlight the differences, and it marks the comparison state as unknown (no opinion).

5.2. HOW FACTS ARE USED IN COMPARISONS

Comparison states based on observed fact values provide guidance in managing your system. The drift service identifies facts whose behavior differs from expectations, facts whose state is unknown, and alerts users to facts that require attention.

5.3. FILTERING SYSTEM FACTS USING THE USER INTERFACE (UI)

The drift service enables multi-fact filtering, allowing you to create comparisons that can be tailored to your business. You can filter system facts in several ways: by the fact comparison state, by fact name, and/or by fact category.

Procedure

1. Navigate to the Red Hat Enterprise Linux > Drift > Comparison page. The Comparison screen opens.
2. On the Comparison screen, click Add to comparison.

3. On the Add to Comparison screen, select the Systems tab to see the systems that are checked into your Insights for RHEL inventory.

4. In the Name column, select the checkbox for two or more systems and click Submit. The Comparison screen opens, showing the state of facts in the systems.

Filtering by comparison state: Click the View drop-down list and select Same to show only the facts for which values are the same, Different to show only the facts that are different, or Incomplete Data to show only the facts where the information is incomplete. You can also select a combination of Same, Different, and Incomplete data states, and clear selections as necessary. When you first add systems for comparison, all three options are selected by default.

Filtering by fact name: Enter your first fact name in the search box at the top to filter by specific fact name. For example, entering kernel as a filter displays all facts containing kernel in their name. Or enter installed_packages in the search box to view all packages. Add additional fact names simply by pressing the enter key and then typing the next fact name. You can add as many fact names as you need for your filter.

Filtering by fact category: Enter your first fact category in the search box to compare systems by that category. Examples include installed_packages, installed_services, kernel_modules, network_interfaces, yum_repos, cpu_flags, and enabled_services. Add additional fact categories simply by pressing the enter key and then typing the next fact category. You can add as many fact categories as you need for your filter.

In the following example screen, you can see the system comparison data filtered by facts that show a difference across systems. Some facts, such as the fqdn, are expected to be different for each system, but the installed packages are expected to be the same. Over time, some packages have been upgraded on system 1, but not on system 2 and system 3, as shown by expanding the fact category installed_packages.
5.4. FILTERING SYSTEM FACTS USING UNIFORM RESOURCE LOCATOR (URL) PARAMETERS

The drift service enables multi-fact filtering, allowing you to create comparisons that can be tailored to your business. You can filter system facts in several ways: by the fact comparison state, by fact name, or by fact category.

Sorting by editing the URL

Expedite filtering by editing the URL parameters. Examine the following URL for an explanation of how to use this functionality:

```
insights/drift/?baseline_ids=<baseline-id>&system_ids=<system-id>&hsp_ids=<hsp-
id>&reference_id=<reference-
id>&filter[name]=bios,arch&filter[state]=same,different,incomplete_data&sort=-state,fact
```

**Parameters**

1. `insights/drift/`
2. `?`
3. `baseline_ids=<baseline-id>&system_ids=<system-id>&hsp_ids=<hsp-id>`
4. `&reference_id=<reference-id>`
5. `&filter[name]=bios,arch`
6. `&filter[state]=same,different,incomplete_data`
7. `&sort=-state,fact`

**How to use parameters**

1. **App service:** This reflects the Insights for Red Hat Enterprise Linux you are using. In this case, the drift service on Insights for Red Hat Enterprise Linux.
2. **Search parameter**: This is the character that tells drift you want to search on the parameters that follow.

3. **IDs of systems/baselines/historical profiles**: These are the IDs of the systems, baselines and historical profiles in your comparison. Each is preceded by the & symbol after the first ? symbol, and the respective parameter type (baseline_ids, system_ids or hsp_ids).

4. **ID of system/baseline/historical profile to use as reference**: This is the ID of the system, baseline or historical system profile that will be used as a reference to compare all other facts to. The reference-id must be specified in one of the parameters (baseline_ids, system_ids or hsp_ids). If not specified, no reference is set for the comparison.

5. **Fact name filters**: Begins with the & symbol and filter[name]. Each fact name filter applied is added after the = symbol and separated by a comma with no spaces.

6. **State filters**: Same as fact name filters, but preceded by filter[state]. Valid values for filter[state] are: same, different and incomplete_data. Multiple values can be specified by separating them with a comma and no space.

7. **Table sorting**: Preceded by the & symbol, state and/or fact are added after sort= and are comma separated. If state or fact is preceded by a - symbol, then it is sorted in descending order; otherwise, it sorts in ascending order. State sort has the ability to have no sort. In this case, state will not be added in the url parameter. Fact sorting, on the other hand, if left off, will default to ascending order.

These parameters can be entered and edited manually, but their primary function is to auto-populate as you make changes in the UI.

### 5.5. SORTING SYSTEM FACTS

You can sort system facts using the User Interface (UI) or by editing URL parameters.

**Sorting using the UI**

You can sort system facts in the UI alphabetically. Click the arrow next to **Fact** (Fact ↑) to switch sorting between ascending and descending order. Note that facts are shown in ascending order by default. You can also sort system facts by the comparison **State**. Click the arrow next to **State** (State ↓) to switch to sorting by state.

**NOTE**

Sorting works in combination with any applied filters. That is, if you have filtered for installed packages or viewing facts by comparison state, the filtered data can be sorted alphabetically or by comparison state.

**Sorting by editing the URL**

Sorting can be expedited by editing the URL parameters. Examine the following URL for an explanation of how to use this functionality:

```
insights/drift/?baseline_ids=<baseline-id>&system_ids=<system-id>&hsp_ids=<hsp-
id>&reference_id=<reference-
id>&filter[name]=bios,arch&filter[state]=same,different,incomplete_data&sort=-state,fact
```
Parameters

1. insights/drift/

2. ?

3. baseline_ids=<baseline-id>&system_ids=<system-id>&hsp_ids=<hsp-id>

4. &reference_id=<reference-id>

5. &filter[name]=bios,arch

6. &filter[state]=same,different,incomplete_data

7. &sort=-state,fact

How to use parameters

1. App service: This reflects the app service you are using. In this case, drift on Insights for Red Hat Enterprise Linux.

2. Search parameter: This is the character that tells drift you want to search on the parameters that follow.

3. IDs of systems/baselines/historical profiles: These are the IDs of the systems, baselines and historical profiles in your comparison. Each is preceded by the & symbol after the first ? symbol, and the respective parameter type (baseline_ids, system_ids or hsp_ids).

4. ID of system/baseline/historical profile to use as reference: This is the ID of the system, baseline or historical system profile that will be used as a reference to compare all other facts to. The reference-id must be specified in one of the parameters (baseline_ids, system_ids or hsp_ids). If not specified, no reference is set for the comparison.

5. Fact name filters: Begins with the & symbol and filter[name]. Each fact name filter applied is added after the = symbol and separated by a comma with no spaces.

6. State filters: Same as fact name filters, but preceded by filter[state]. Valid values for filter[state] are: same, different and incomplete_data. Multiple values can be specified by separating them with a comma and no space.

7. Table sorting: Preceded by the & symbol, state and/or fact are added after sort= and are comma separated. If state or fact is preceded by a - symbol, then it is sorted in descending order; otherwise, it sorts in ascending order. State sort has the ability to have no sort. In this case, state will not be added in the url parameter. Fact sorting, on the other hand, if left off, will default to ascending order.

These parameters can be entered manually, but their primary function is to auto-populate as you make changes in the UI.

5.6. OBfuscated VALUE HIGHLIGHTS

The Insights client provides IP address obfuscation and host name obfuscation. More information about setting up client data obfuscation can be found in the Client Configuration Guide.

In the instance that one of your fact values is redacted to protect information, drift will inform you that hidden data is obfuscated in the comparison. Indications of obfuscated data include:
• Graying out of the value cell.

• Including a lock icon in the grayed out cell with a tooltip telling you the value has been redacted.

• Including a tooltip on the "state" icon to describe why the "incomplete data" state is set for that row.

If one of the values in the comparison is redacted, the state of the comparison for that fact will be set to "incomplete data."

5.7. UNDERSTANDING MULTI-VALUE FACTS

Drift stores a list of all installed versions for a given package name. This means you can correctly evaluate and compare all versions installed when performing an analysis.

Inventory and drift APIs also provide facts with multiple values. More information can be found on the Insights for Red Hat Enterprise Linux API documentation. System Comparison API Documentation, https://console.redhat.com/insights/inventory[Red Hat Enterprise Linux > Inventory

NOTE

Previously, each installed_package fact had an associated value that was used to compare systems and baselines. This was a limitation for packages for which multiple values were present. For example, it is possible to install multiple kernel packages on your RHEL system, and select the version you want to run at startup. Another example is having multiple versions or architectures, or the same package installed on your RHEL system. This enables enhanced troubleshooting in drift.

5.8. USING MULTI-FACT FILTERING

Multi-fact filtering enables the creation of comparisons that are tailor made for your business. You can filter your comparison queries by specific groups of facts and tags. Multi-fact filtering enables the following functionality:
- Have multiple inputs in the fact name field.
- Avoid swapping back and forth between multiple filters.
- Exclude irrelevant facts.
- Compare facts related to a specific issue for improved troubleshooting.
- Share comparisons with other administrators or colleagues.

To lock in multiple filters on facts, press Enter after you have finished typing the fact name in the text box. This functionality is powered by an "OR" operator that allows you to filter out anything that does not match "this" or "that." The following example shows that the list of facts will only contain facts whose name matches "bios" or "arch."

### 5.9. AVAILABLE FACTS AND THEIR FUNCTIONS

The table below displays the system facts for use in system comparisons.

#### Table 5.1. System Facts

<table>
<thead>
<tr>
<th>Fact Name</th>
<th>Description</th>
<th>Example Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ansible</td>
<td>Category with a list of Ansible-related facts</td>
<td>controller_version with a value of 4.0.0</td>
</tr>
<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>Fact Name</td>
<td>Description</td>
<td>Example Value</td>
</tr>
<tr>
<td>----------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>--------------------------------------</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>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>mssql</td>
<td>Category with a list of MSSQL-related facts</td>
<td>mssql_version with a value of 15.0.4153.1</td>
</tr>
<tr>
<td>network_interfaces</td>
<td>List of facts related to network interfaces.</td>
<td></td>
</tr>
</tbody>
</table>
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.).

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`.

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.

<table>
<thead>
<tr>
<th>Fact Name</th>
<th>Description</th>
<th>Example Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>number_of_cpus</td>
<td>Total number of CPUs</td>
<td>1</td>
</tr>
<tr>
<td>number_of IBOutlets</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>cron, with a value of 1.</td>
</tr>
<tr>
<td>sap_instance_number</td>
<td>SAP instance number</td>
<td>42</td>
</tr>
<tr>
<td>sap_sids</td>
<td>SAP system ID (SID)</td>
<td>A42</td>
</tr>
<tr>
<td>sap_system</td>
<td>Boolean field that indicates if SAP is installed on the system</td>
<td>True</td>
</tr>
<tr>
<td>sap_version</td>
<td>SAP version number</td>
<td>2.00.052.00.1599 235305</td>
</tr>
<tr>
<td>satellite_managed</td>
<td>Boolean field that indicates is a system is registered to a Satellite server.</td>
<td>FALSE</td>
</tr>
<tr>
<td>selinux_current_mode</td>
<td>Current SELinux mode</td>
<td>enforcing</td>
</tr>
<tr>
<td>selinux_config_file</td>
<td>SELinux mode set in the config file</td>
<td>enforcing</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>system_memory</td>
<td>Total system memory in human-readable form</td>
<td>3.45 GiB</td>
</tr>
<tr>
<td>tuned_profile</td>
<td>Current profile resulting from the command <code>tuned-adm active</code></td>
<td>desktop</td>
</tr>
<tr>
<td>yum_repos</td>
<td>List of yum repositories. The repository name is added to the beginning of the fact. Each repository has the associated facts <code>base_url</code>, enabled, and gpgcheck.</td>
<td>Red Hat Enterprise Linux 7 Server (RPMs).base_url would have the value <code>https://cdn.redhat.com/content/dist/rhel/server/7/$releasever/$basearch/os</code></td>
</tr>
</tbody>
</table>
CHAPTER 6. COMPARING SYSTEMS AND SYSTEM PROFILES

You can compare systems to each other, to profiles, and to a reference point. When you have selected all the systems you want to compare, you can use the Comparison screen to compare the facts in the systems.

6.1. COMPARING SYSTEM PROFILES

After you have added to your inventory all the systems you want to compare, use the Comparison screen to compare the facts in the systems.

6.2. COMPARING SYSTEMS TO A REFERENCE POINT

At times, you may want to compare all systems to a single reference point rather than compare all systems with each other, or compare them as a group. For example, you may need to compare all systems to a baseline so that all systems are calculated against that baseline. You can also compare a system against time-stamped profiles to understand where and when changes have occurred.

You might also want to invert a comparison. For example, instead of comparing profiles with old time stamps against the latest profile, you may want to compare all profiles against the oldest known working version of a system. Such a comparison enables you to identify changes that deviate from the reference point.

6.2.1. Comparing systems to a single point of reference

You can compare multiple systems or all systems to a single reference point.

Procedure

1. On the Comparison screen, click the star icon on the system header.

The drift service compares each of your selected systems with the fact values in the reference system, displayed in the first position column, with its header highlighted in blue.

The fact values are displayed with every difference highlighted in red.

NOTE

A fact category with multiple values has no highlights until you expand the fact to view details. When you do so, drift shows the specific fact details highlighted in red.

6.3. USING THE HISTORY OF SYSTEM PROFILES

Every time you submit a system for comparison, the submission checks in the profile and marks it with a time stamp. By examining the different profile versions, you can see a view of the system over time.

NOTE

On the Choose systems screen, if a system is marked with a time stamp icon, you may be able to open that system directly. Otherwise, add the system you want to compare into drift before following Step 1.
6.4. EXPORTING SYSTEM COMPARISON OUTPUT

The drift service allows you to export system comparison output, along with any current selections such as filters, to a comma-separated values (CSV) file or a JavaScript Object Notation (JSON) file. You can then open the CSV or JSON file with the tool of your choice to compare exported facts and analyze discrepancies in your systems.

NOTE

The exported CSV or JSON report preserves all your current selections on the system comparison output, including any filters applied. That is, it follows the WYSIWYG (What You See Is What You Get) paradigm. Therefore, you will need to expand any nested fact categories (installed_packages, for example) to be exported in the report.

Procedure

1. On the Comparison output screen for two or more systems, click the Export icon, and then select either Export as CSV, or Export as JSON.

2. Open the CSV or JSON file with the tool of your choice so that you can easily compare exported facts and analyze discrepancies in systems.

6.4.1. Exporting system comparison output to a CSV or JSON file

You can export your system profile output, along with any filters you used, to a comma-separated values (CSV) or JavaScript Object Notation (JSON) file. The filters highlight the differences and incomplete data in your system profiles. You can use this information for researching characteristics about your systems, and troubleshooting any issues you discover.

Procedure

1. On the Comparison output screen for two or more systems, click the Export icon

2. , and then select either Export to CSV or Export to JSON.