Red Hat Insights 2020-04

Monitoring and Reacting to Configuration Changes Using Policies

How to create policies to detect system configuration changes and get notified by email

Last Updated: 2020-08-18

Red Hat Insights 2020-04 Monitoring and Reacting to Configuration Changes Using Policies

How to create policies to detect system configuration changes and get notified by email

Red Hat Customer Content Services

Legal Notice

Abstract

This document provides an overview of the Policies service and demonstrates how to create a policy to detect system configuration changes and get 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. OVERVIEW

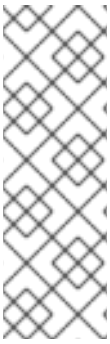
Policies evaluate system configurations in your environment and are processed on reception of uploading an insights-client payload to cloud.redhat.com. If condition(s) in your policy are met, defined action(s) are triggered. Policies are applied to all systems registered in the Insights inventory. Users can create and manage policies using the user interface or via API.

Use policies to assist operational management with simple tasks such as:

- Raising an alert when some conditions are met on system configuration.
- Emailing a team when security packages are out of date on a system.

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

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



NOTE

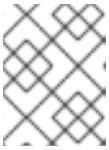
- Use the Role Based Access Control (RBAC) capability in https://cloud.redhat.com (Settings  > User access) to control user access for Policies.
- See Role Based Access Control for Red Hat Insights and cloud management services for Red Hat Enterprise Linux for more information about this feature and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

CHAPTER 2. USER PREFERENCES

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

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

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



NOTE

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

3. Click **Submit**.

CHAPTER 3. CREATING A POLICY TO DETECT CONFIGURATION CHANGES AND GET NOTIFIED BY EMAIL

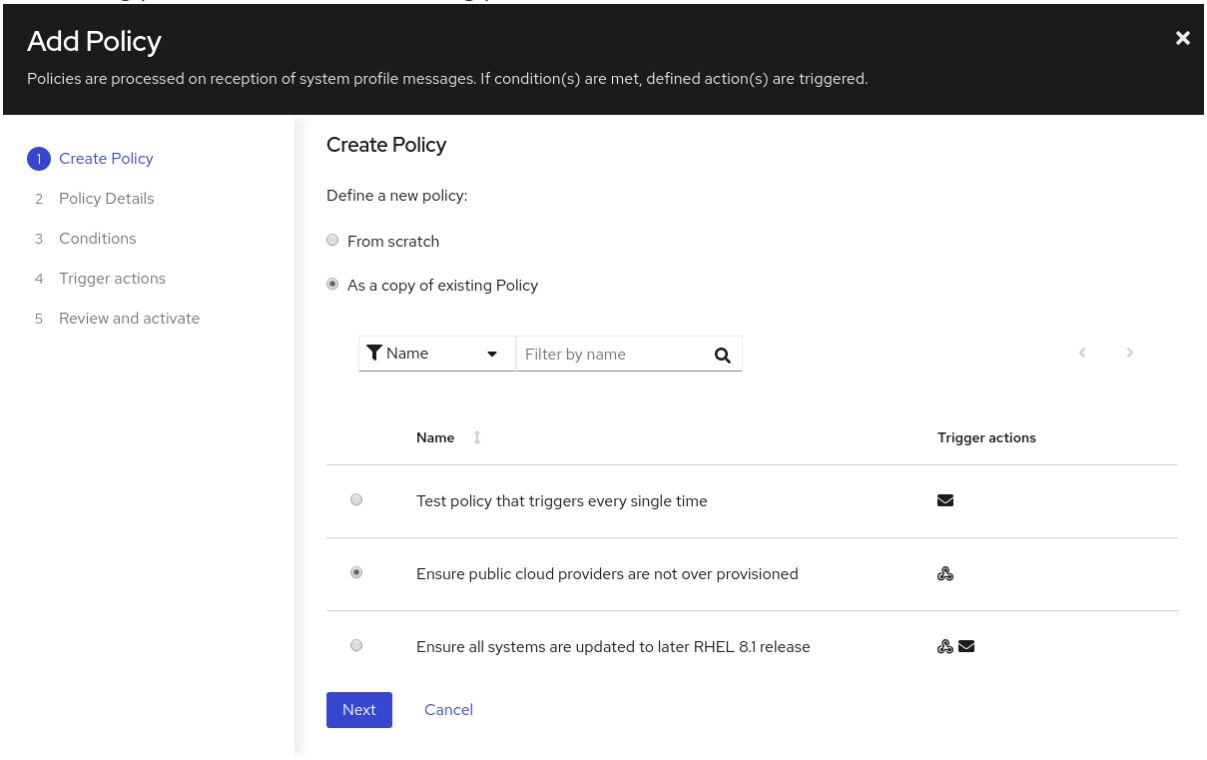
The following workflow examples demonstrate how to create a policy to detect system configuration changes and get notified by email.

NOTE

If you see a warning message about email alerts not opted in when creating your policy, set your preferences to receive email from your policies as described in the Chapter 2, *User preferences* section.

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

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 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.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 said public cloud providers are running with CPU hardware higher than the allowed limit.

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, and if the condition in the policy is met, an email will be sent to all users on the account with access to Policies according to their email preferences.

3.2. CREATING A POLICY TO DETECT IF SYSTEMS ARE RUNNING AN OUTDATED VERSION OF RHEL (OLDER THAN RHEL 8.1) AND GET NOTIFIED BY EMAIL

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 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.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, and if the condition in the policy is met, an email will be sent to all users on the account with access to Policies according to their email preferences.

3.3. CREATING A POLICY TO DETECT A VULNERABLE PACKAGE VERSION BASED ON RECENT CVE AND GET NOTIFIED BY EMAIL

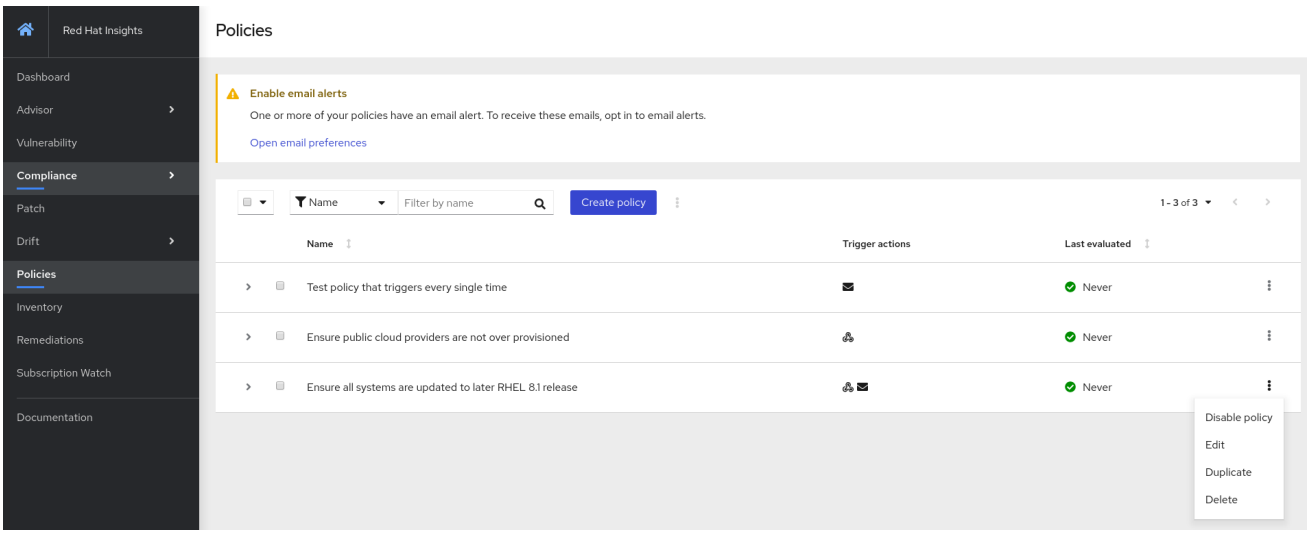
1. In the cloud.redhat.com platform, click Policies under Red Hat Insights.

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 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, and if the condition in the policy is met, an email will be sent to all users on the account with access to Policies according to 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 the Chapter 2, *User preferences* section.

CHAPTER 5. APPENDIX

5.1. AVAILABLE FACTS AND THEIR FUNCTIONS

Table 5.1. System Facts and Their Functions

Fact Name	Description	Example Value
arch	System architecture	**x86_64**
bios_release_date	BIOS release date; typically **MM/DD/YYYY**	01/01/2011
bios_vendor	BIOS vendor name	LENOVO
bios_version	BIOS version	1.17.0
cloud_provider	Cloud vendor. Values are **google**, **azure**, **aws**, **alibaba**, or empty	**google**
cores_per_socket	Number of CPU cores per socket	2
cpu_flags	Category with a list of CPU flags. Each name is the CPU flag (ex: **vmx**), and the value is always **enabled**.	**vmx**, with a value of **enabled**.
enabled_services	Category with a list of enabled services. Each name in the category is the service name (ex: **crond**), and the value is always **enabled** .	**crond**, with a value of **enabled**.
fqdn	System Fully Qualified Domain Name	*system1.example.com*
infrastructure_type	System infrastructure; common values are **virtual** or **physical**	**virtual**
infrastructure_vendor	Infrastructure vendor; common values are **kvm**, **vmware**, **baremetal**, etc.	**kvm**
installed_packages	List of installed RPM packages. This is a category.	**bash**, with a value of **4.2.46-33.el7.x86_64**.
installed_services	Category with a list of installed services. Each name in the category is the service name (ex: **crond**), and the value is always **installed**.	**crond**, with a value of **installed**.
kernel_modules	List of kernel modules. Each name in the category is the kernel module (ex: **nfs**), and the value is **enabled**.	**nfs**, with a value of **enabled**.

Fact Name	Description	Example Value
last_boot_time	The boot time in **YYYY-MM-DDTHH:MM:SS** format. Informational only; we do not compare boot times across systems.	**2019-09-18T16:54:56**
network_interfaces	List of facts related to network interfaces.	
	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.	
number_of_cpus	Total number of CPUs	**1**
number_of_sockets	Total number of sockets	**1**
os_kernel_version	Kernel version	**4.18.0**
os_release	Kernel release	**8.1**
running_processes	List of running processes. The fact name is the name of the process, and the value is the instance count.	**crond**, with a value of **1**.
satellite_managed	Boolean field that indicates is a system is registered to a Satellite server.	**FALSE**
system_memory	Total system memory in human-readable form	**3.45 GiB**

5.2. AVAILABLE OPERATORS IN CONDITIONS

Available Operators	Value
Logical Operators	AND
	OR
Boolean Operators	EQUAL
	NOTEQUAL
Numeric Compare Operators	GT
	GTE
	LT
	LTE
String Compare Operator	CONTAINS
Array Operators	IN
	CONTAINS
Parser Operators	OR
	AND
	NOT
	EQUAL
	NOTEQUAL
	CONTAINS
	NEG