



Red Hat OpenStack Platform 14

Deploy Fernet on the Overcloud

Deploy Fernet on the Red Hat OpenStack Platform director overcloud

Red Hat OpenStack Platform 14 Deploy Fernet on the Overcloud

Deploy Fernet on the Red Hat OpenStack Platform director overcloud

OpenStack Team
rhos-docs@redhat.com

Legal Notice

Copyright © 2019 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

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

Deploy Fernet on the Red Hat OpenStack Platform director overcloud.

Table of Contents

CHAPTER 1. USING FERNET TOKENS IN THE OVERCLOUD	3
1.1. REVIEW THE FERNET DEPLOYMENT	3
1.2. ROTATE THE FERNET KEYS	4
1.2.1. Rotate the Fernet Keys Using Mistral	4

CHAPTER 1. USING FERNET TOKENS IN THE OVERCLOUD

Fernet is now the default token provider, replacing **uuid**. This guide describes how to review your Fernet deployment, and how to rotate the Fernet keys.

1.1. REVIEW THE FERNET DEPLOYMENT

This procedure reviews your configuration to confirm that Fernet tokens are working correctly.

1. Retrieve the IP address of the controller node.

```
[stack@director ~]$ source ~/stackrc
[stack@director ~]$ openstack server list
+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+
| ID                                     | Name
| Status | Networks          |
+-----+-----+-----+-----+
| 756fbd73-e47b-46e6-959c-e24d7fb71328 | overcloud-controller-0 |
ACTIVE | ctlplane=192.0.2.16 |
| 62b869df-1203-4d58-8e45-fac6cd4cfbee | overcloud-novacompute-0 |
ACTIVE | ctlplane=192.0.2.8  |
+-----+-----+-----+-----+
+-----+-----+-----+-----+

```

2. SSH to the controller.

```
[heat-admin@overcloud-controller-0 ~]$ ssh heat-admin@192.0.2.16
```

3. Retrieve the values of the token driver and provider settings.

```
[heat-admin@overcloud-controller-0 ~]$ sudo crudini --get
/var/lib/config-data/puppet-generated/keystone/etc/keystone/keystone.conf token driver
sql
[heat-admin@overcloud-controller-0 ~]$ sudo crudini --get
/var/lib/config-data/puppet-generated/keystone/etc/keystone/keystone.conf token provider
fernet
```

4. Test the Fernet provider.

```
[heat-admin@overcloud-controller-0 ~]$ exit
[stack@director ~]$ source ~/overcloudrc
[stack@director ~]$ openstack token issue
+-----+-----+-----+-----+-----+-----+
-----+
| Field | Value |
+-----+-----+-----+-----+-----+
-----+
-----+
-----+

```

```

-----+
| expires | 2016-09-20 05:26:17+00:00 |
| id |
gAAAAABX4LppE8vaiFZ992eah2i3edp01aDFx1KZq6a_RJzxUx56QVKORrmW0-oZK3-
Xuu2wcnpYq_eeK2SGLz250eLpZ0zxKBR0GsoMfxJU8mEFF8NzfLNcbuS-iz7SV-
N1re3XEywSDG90JcgvjQfXW-8jtCm-n3LL5IaZexAYIw059T_-cd8 |
| project_id | 26156621d0d54fc39bf3adb98e63b63d |
| user_id | 397daf32cadd490a8f3ac23a626ac06c |
+-----+-----+-----+-----+-----+-----+-----+-----+
-----+
-----+

```

The result should include the long Fernet token.

1.2. ROTATE THE FERNET KEYS

It is recommended that you rotate your Fernet keys regularly, as a compromised keystone key can allow an attacker to generate their own tokens and subsequently grant themselves access to a project.

Fernet uses three types of keys, which are stored in `/var/lib/config-data/puppet-generated/keystone/etc/keystone/fernet-keys`. The highest-numbered directory contains the primary key, which is used to generate new tokens and decrypt existing ones.

During the key rotation process, the primary key is relegated to secondary key status, and a new primary key is issued, thereby reducing the value of a compromised primary key. Secondary keys can only be used to decrypt tokens that were created with previous primary keys, and cannot issue new ones.

1.2.1. Rotate the Fernet Keys Using Mistral

By default, director is configured to manage the overcloud's Fernet keys; this setting is managed in the environment file using `ManageKeystoneFernetKeys`. As a result, the Fernet keys are stored in Mistral (under `KeystoneFernetKeys`). This approach means that you can rotate the Fernet keys with Mistral, and they will still persist after stack updates.

1. Review the existing Fernet keys.
 - a. Identify the Fernet key location.

```

# SSH back to the controller
[heat-admin@overcloud-controller-0 ~]$ sudo crudini --get
/var/lib/config-data/puppet-
generated/keystone/etc/keystone/keystone.conf fernet_tokens
key_repository
/etc/keystone/fernet-keys

```



NOTE

The `/etc/keystone/` directory refers to the container file system path.

- b. Review the current Fernet key directories.


```
[heat-admin@overcloud-controller-0 ~]$ sudo ls /var/lib/config-
data/puppet-generated/keystone/etc/keystone/fernet-keys
0 1 2
```

- **0** - Contains the *staged* key, (which becomes the next primary key) and will always be numbered **0**.
- **1** - Contains the *secondary* key.
- **2** - Contains the *primary* key. This number will increment each time the keys are rotated, with the highest number always serving as the primary key.



NOTE

- The maximum number of keys is determined by the `max_active_keys` property, by default 5 keys.
- The keys are propagated across all controllers.

2. Rotate the Fernet keys using the Mistral workflow.

```
[stack@director ~]$ source ~/stackrc
[stack@director ~]$ openstack workflow execution create
tripleo.fernet_keys.v1.rotate_fernet_keys '{"container":
"overcloud"}'
```

Field	Value
ID	58c9c664-b966-4f82-b368-af5ed8de5b47
Workflow ID	78f0990a-3d34-4bf2-a127-10c149bb275c
Workflow name	tripleo.fernet_keys.v1.rotate_fernet_keys
Description	
Task Execution ID	<none>
State	RUNNING
State info	None
Created at	2017-12-20 11:13:50
Updated at	2017-12-20 11:13:50

3. Get the ID and ensure that the workflow was executed successfully.

```
[stack@director ~]$ openstack workflow execution show 58c9c664-b966-
4f82-b368-af5ed8de5b47
```

Field	Value
ID	58c9c664-b966-4f82-b368-af5ed8de5b47
Workflow ID	78f0990a-3d34-4bf2-a127-10c149bb275c
Workflow name	tripleo.fernet_keys.v1.rotate_fernet_keys
Description	
Task Execution ID	<none>
State	SUCCESS
State info	None

```

| Created at          | 2017-12-20 11:13:50          |
| Updated at         | 2017-12-20 11:15:00          |
+-----+-----+-----+

```

4. On the controller, review the number of Fernet keys, and compare with the previous result.

```

[heat-admin@overcloud-controller-0 ~]$ sudo ls /var/lib/config-
data/puppet-generated/keystone/etc/keystone/fernet-keys
0 1 2 3

```

- **0** - Contains the *staged* key, and will always be numbered **0**. This key will be promoted to a primary key during the next rotation.
- **1 & 2** - Contain the *secondary* keys.
- **3** - Contains the *primary* key. This number will increment each time the keys are rotated, with the highest number always serving as the primary key.



NOTE

- The maximum number of keys is determined by the `max_active_keys` property, by default 5 keys.
- The keys are propagated across all controllers.