Red Hat OpenStack Platform 16.1

High Availability for Compute Instances

Configure High Availability for Compute Instances

Last Updated: 2020-09-16

Red Hat OpenStack Platform 16.1 High Availability for Compute Instances

Configure High Availability for Compute Instances

OpenStack Team
rhos-docs@redhat.com

Legal Notice

Abstract

This guide describes how to manage Instance High Availability (Instance HA). With Instance HA, Red Hat OpenStack Platform (RHOSP) can automatically evacuate and re-create instances on a different Compute node when a Compute node fails.

Table of Contents

CHAPTER 1. HOW INSTANCE HA WORKS 3

CHAPTER 2. PLANNING YOUR INSTANCE HA DEPLOYMENT 4

CHAPTER 3. INSTALLING AND CONFIGURING INSTANCE HA 5
3.1. CONFIGURING THE INSTANCE HA ROLE, FLAVOR, AND PROFILE 5
3.2. ENABLING FENCING 6
3.3. DEPLOYING THE OVERCLOUD 7
3.4. TESTING INSTANCE HA EVACUATION 7

CHAPTER 4. DESIGNATING SPECIFIC INSTANCES TO EVACUATE 9

CHAPTER 1. HOW INSTANCE HA WORKS

When a Compute node fails, the Instance High Availability (HA) tool evacuates and re-creates the instances on a different Compute node.

Instance HA uses the following resource agents:

Agent name	Name inside cluster	Role
fence_compute	**fence-nova**	Marks a Compute node for evacuation when the node becomes unavailable.
NovaEvacuate	**nova-evacuate**	Evacuates instances from failed nodes. This agent runs on one of the Controller nodes.
Dummy	**compute-unfence-trigger**	Releases a fenced node and enables the node to run instances again.

The following events occur when a Compute node fails and triggers Instance HA:

1. At the time of failure, the **IPMI** agent performs first-layer fencing, which includes physically resetting the node to ensure that it shuts down and preventing data corruption or multiple identical instances on the overcloud. When the node is offline, it is considered fenced.

2. After the physical IPMI fencing, the **fence-nova** agent automatically performs second-layer fencing and marks the fenced node with the **"evacuate=yes"** cluster per-node attribute by running the following command:

   ```
   $ attrd_updater -n evacuate -A name="evacuate" host="FAILEDHOST" value="yes"
   ```

   FAILEDHOST is the name of the failed Compute node.

3. The **nova-evacuate** agent continually runs in the background and periodically checks the cluster for nodes with the **"evacuate=yes"** attribute. When **nova-evacuate** detects that the fenced node contains this attribute, the agent starts evacuating the node. The evacuation process is similar to the manual instance evacuation process that you can perform at any time. For more information about instance evacuation, see: Evacuate Instances.

4. When the failed node restarts after the IPMI reset, the **nova-compute** process on that node also starts automatically. Because the node was previously fenced, it does not run any new instances until Pacemaker un-fences the node.

5. When Pacemaker detects that the Compute node is online, it starts the **compute-unfence-trigger** resource agent on the node, which releases the node and so that it can run instances again.

Instance HA works with shared storage or local storage environments, which means that evacuated instances maintain the same network configuration, such as static IP and floating IP. The re-created instances also maintain the same characteristics inside the new Compue node.

CHAPTER 2. PLANNING YOUR INSTANCE HA DEPLOYMENT

Before you deploy Instance HA, review the following considerations:

- Compute node host names and Pacemaker remote resource names must comply with the W3C naming conventions. For more information, see Declaring Namespaces and Names and Tokens in the W3C documentation.

- Typically, Instance HA requires that you configure shared storage for disk images of instances. Therefore, if you attempt to use the **no-shared-storage** option, you might receive an **InvalidSharedStorage** error during evacuation, and the instances will not start on another Compute node.
  However, if all your instances are configured to boot from an OpenStack Block Storage (**cinder**) volume, you do not need to configure shared storage for the disk image of the instances, and you can evacuate all instances using the **no-shared-storage** option.

  During evacuation, if your instances are configured to boot from a Block Storage volume, any evacuated instances boot from the same volume on another Compute node. Therefore, the evacuated instances immediately restart their jobs because the OS image and the application data are stored on the OpenStack Block Storage volume.

- If you deploy Instance HA in a Spine-Leaf environment, you must define a single **internal_api** network for the Controller and Compute nodes. You can then define a subnet for each leaf. For more information about configuring Spine-Leaf networks, see Creating a roles data file in the *Spine Leaf Networking* guide.

- From Red Hat OpenStack Platform 13 and later, you use director to upgrade Instance HA as a part of the overcloud upgrade. For more information about upgrading the overcloud, see Keeping Red Hat OpenStack Platform Updated guide.

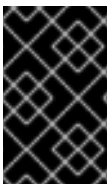
- Disabling Instance HA with the director after installation is not supported. For a workaround to manually remove Instance HA components from your deployment, see the article How can I remove Instance HA components from the controller nodes? .

IMPORTANT

This workaround is not verified for production environments. You must verify the procedure in a test environment before you implement it in a production environment.

CHAPTER 3. INSTALLING AND CONFIGURING INSTANCE HA

Red Hat OpenStack Platform (RHOSP) director deploys Instance High Availability (HA). However, you must perform additional steps to configure a new Instance HA deployment on a new overcloud. After you complete the steps, Instance HA will run on a subset of Compute nodes with a custom role.



IMPORTANT

To enable instance HA in a different environment, such as an existing overcloud that uses standard or custom roles, perform only the procedures that are relevant to your deployment and adapt your templates accordingly.

For general information about deploying the overcloud, see Director Installation and Usage guide. For more information on custom roles, see Composable Services and Custom Roles in the *Advanced Overcloud Customization* guide.

3.1. CONFIGURING THE INSTANCE HA ROLE, FLAVOR, AND PROFILE

Procedure

You can modify the example file and role names in the following procedure according to your environment.

1. Add the **ComputeInstanceHA** role to your **roles-data.yaml** file and regenerate the file.

```
$ openstack overcloud roles generate -o ~/my_roles_data.yaml Controller Compute
ComputeInstanceHA
```

   The **ComputeInstanceHA** role includes all the services in the default **Compute** role, the **ComputeInstanceHA** services, and the **PacemakerRemote** services. For general information about custom roles and about the **roles-data.yaml** file, see Roles in the *Advanced Overcloud Customization* guide.

2. Create the **compute-instance-ha** flavor to tag the Compute nodes to manage with Instance HA.

```
$ source ~/stackrc
$ openstack flavor create --id auto --ram 6144 --disk 40 --vcpus 4 compute-instance-ha
$ openstack flavor set --property "cpu_arch"="x86_64" --property
"capabilities:boot_option"="local" --property "capabilities:profile"="compute-instance-ha"
compute-instance-ha
```

3. Tag each Compute node that you want to manage with Instance HA with the **compute-instance-ha** profile, and replace **<NODE UUID>** with the actual UUID:

```
$ openstack baremetal node set --property capabilities='profile:compute-instance-
ha,boot_option:local' <NODE UUID>
```

4. Map the **ComputeInstanceHA** role to the **compute-instance-ha** flavor by creating an environment file with the following parameter:

```
parameter_defaults:
  OvercloudComputeInstanceHAFlavor: compute-instance-ha
```

3.2. ENABLING FENCING

Enable fencing on all Controller and Compute nodes in the overcloud by creating an environment file with fencing information.

Procedure

1. Create the environment file in an accessible location, such as ***~/templates***, and include the following content:

```
parameter_defaults:
  EnableFencing: true
  FencingConfig:
    devices:
    - agent: fence_ipmilan
      host_mac: 00:ec:ad:cb:3c:c7
      params:
        login: admin
        ipaddr: 192.168.24.1
        ipport: 6230
        passwd: password
        lanplus: 1
    - agent: fence_ipmilan
      host_mac: 00:ec:ad:cb:3c:cb
      params:
        login: admin
        ipaddr: 192.168.24.1
        ipport: 6231
        passwd: password
        lanplus: 1
    - agent: fence_ipmilan
      host_mac: 00:ec:ad:cb:3c:cf
      params:
        login: admin
        ipaddr: 192.168.24.1
        ipport: 6232
        passwd: password
        lanplus: 1
    - agent: fence_ipmilan
      host_mac: 00:ec:ad:cb:3c:d3
      params:
        login: admin
        ipaddr: 192.168.24.1
        ipport: 6233
        passwd: password
        lanplus: 1
    - agent: fence_ipmilan
      host_mac: 00:ec:ad:cb:3c:d7
      params:
        login: admin
        ipaddr: 192.168.24.1
        ipport: 6234
        passwd: password
        lanplus: 1
```

For more information about fencing and STONITH configuration, see Fencing Controller Nodes with STONITH in the *High Availability Deployment and Usage* guide.

2. Instance HA uses shared storage by default. If shared storage is not configured for your Compute instance, add the following parameter to the environment file that you created:

```
parameter_defaults:
  ExtraConfig:
    tripleo::instanceha::no_shared_storage: true
```

For information about how to boot from an OpenStack Block Storage (cinder) volume instead of configuring shared storage for the disk image of an instance, see Chapter 2, *Planning your Instance HA deployment*.

3.3. DEPLOYING THE OVERCLOUD

You can configure Instance HA for your overcloud at any time after you create the undercloud. If you already deployed the overcloud, you must rerun the **openstack overcloud deploy** command with the additional Instance HA files you created.

Procedure

- Use the **openstack overcloud deploy** command with the **-e** option for each environment file that you created and with the ***compute-instanceha.yaml*** environment file. Replace **<FLAVOR_ENV_FILE>** and **<FENCING_ENV_FILE>** with the appropriate file names in your environment:

```
$ openstack overcloud deploy --templates \
  -e <FLAVOR_ENV_FILE> \
  -e <FENCING_ENV_FILE> \
  -e /usr/share/openstack-tripleo-heat-templates/environments/compute-instanceha.yaml
```

NOTE

- Do not modify the **compute-instanceha.yaml** environment file.
- Include the full path to each environment file that you want to include in the overcloud deployment.

Result

After the deployment is complete, each Compute node includes a **STONITH** device and a **GuestNode** service.

3.4. TESTING INSTANCE HA EVACUATION

WARNING

The following procedure involves deliberately crashing a Compute node, which triggers the automated evacuation of instances with Instance HA.

Procedure

1. Start one or more instances on the overcloud.

```
stack@director $ . overcloudrc
stack@director $ nova boot --image cirros --flavor 2 test-failover
stack@director $ nova list --fields name,status,host
```

2. Log in to the Compute node that hosts the instances. Replace **compute-n** with the name of the Compute node:

```
stack@director $ . stackrc
stack@director $ ssh -l heat-admin compute-n
```

3. Crash the Compute node.

```
heat-admin@compute-n $ echo c > /proc/sysrq-trigger
```

4. Wait a few minutes for the node to restart, and then verify that the instances from the Compute node that you crash are re-created on another Compute node:

```
stack@director $ nova list --fields name,status,host
stack@director $ nova service-list
```

CHAPTER 4. DESIGNATING SPECIFIC INSTANCES TO EVACUATE

By default, Instance HA evacuates all instances from a failed node. You can also configure Instance HA to only evacuate instances with specific images or flavors.

Prerequisites

1. Log in to the undercloud as the **stack** user.
2. Source the **overcloudrc** file:

```
$ source ~/overcloudrc
```

Tagging an image to evacuate

- Tag an image and replace **IMAGE_ID** with the ID of the image that you want to evacuate:

```
(overcloud) $ openstack image set --tag evacuable IMAGE_ID
```

Tagging a flavor to evacuate

- Tag a flavor and replace **FLAVOR_ID** with the ID of the flavor that you want to evacuate:

```
(overcloud) $ nova flavor-key FLAVOR_ID set evacuable=true
```