Red Hat OpenStack Platform 16.0

High Availability for Compute Instances

Configure High Availability for Compute Instances

Last Updated: 2020-01-17

Red Hat OpenStack Platform 16.0 High Availability for Compute Instances

Configure High Availability for Compute Instances

OpenStack Team
rhos-docs@redhat.com

Legal Notice

Abstract

A guide for configuring High Availability for Compute Instances (Instance HA) in Red Hat OpenStack Platform. This document focuses on enabling Instance HA through Ansible.

Table of Contents

CHAPTER 1. OVERVIEW

This guide describes how to manage *Instance High Availability (Instance HA)* . Instance HA allows Red Hat OpenStack Platform to automatically evacuate and re-spawn instances on a different Compute node when their host Compute node fails.

The evacuation process that is triggered by Instance HA is similar to what users can do manually, as described in Evacuate Instances.

Instance HA works on shared storage or local storage environments, which means that evacuated instances maintain the same network configuration (static IP, floating IP, and so on) and the same characteristics inside the new host, even if they are spawned from scratch.

Instance HA is managed by 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.

CHAPTER 2. HOW INSTANCE HA WORKS

OpenStack uses Instance HA to automate the process of evacuating instances from a Compute node when that node fails. The following procedure describes the sequence of events that are triggered when a Compute node fails.

1. At the time of failure, the **IPMI** agent performs *first-layer fencing* and physically resets the node to ensure that it is powered off. Evacuating instances from online Compute nodes might result in data corruption or in multiple identical instances running on the overcloud. When the node is powered off, it is considered *fenced*.

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

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

   Where *FAILEDHOST* is the hostname of the failed Compute node.

3. The **nova-evacuate** agent continually runs in the background, periodically checking 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 using the process described in Evacuate Instances.

4. While the failed node boots up from the IPMI reset, the **nova-compute** process on that node starts automatically. Because the node was fenced earlier, it does not run any new instances until Pacemaker *unfences* it.

5. When Pacemaker sees that the Compute node is online again, it tries to start the **compute-unfence-trigger** resource on the node, reverting the force-down API call and setting the node as enabled again.

2.1. DESIGNATING SPECIFIC INSTANCES TO BE EVACUATED

By default, all instances are to be evacuated, but it is also possible to tag images or flavors for evacuation.

To tag an image:

```
$ openstack image set --tag evacuable ID-OF-THE-IMAGE
```

To tag a flavor:

```
$ nova flavor-key ID-OF-THE-FLAVOR set evacuable=true
```

CHAPTER 3. INSTALLING AND CONFIGURING INSTANCE HA

Instance HA is deployed and configured with the director. However, there are a few additional steps that you need to perform to prepare for the deployment.

This section includes all the steps needed to configure a new Instance HA deployment on a new overcloud with the goal of enabling Instance HA on a subset of Compute nodes with a custom role.

IMPORTANT

- If you want to enable instance HA in a different environment, such as an existing overcloud using either standard or custom roles, follow only the procedures that are relevant to your deployment and adapt your templates accordingly.
- 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? . **This workaround is provided as-is** You must verify the procedure in a test environment before implementing in production.

For general information on deploying the overcloud, see the Director Installation and Usage guide. For information on custom roles, see Composable Services and Custom Roles.

Configure the Instance HA role, flavor, and profile

1. Add the **ComputeInstanceHA** role to your roles data file and regenerate the file. For example:

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

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

2. Create the **compute-instance-ha** flavor to tag Compute nodes that you want to designate for Instance HA. For example:

```
$ 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 designate for Instance HA with the **compute-instance-ha** profile.

```
$ 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 content:

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

Enable fencing

1. Enable fencing on all Controller and Compute nodes in the overcloud by creating an environment file with fencing information. Make sure to create the environment file in an accessible location, such as ***~/templates***. For example:

```
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 the Fencing the Controller Nodes section of the *Advanced Overcloud Customization* guide.

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

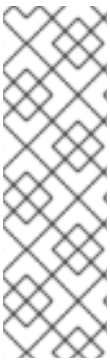
```
parameter_defaults:
  ExtraConfig:
    tripleo::instanceha::no_shared_storage: true
```

See Section 3.1, "Considerations for Shared Storage" for details on how to boot from an OpenStack Block Storage (cinder) volume rather than configuring shared storage for the disk image of instances.

Deploy the overcloud

Run the **openstack overcloud deploy** command with the **-e** option for each environment file that you created, as well as the ***compute-instanceha.yaml*** environment file. For example:

```
$ 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.
- Make sure to include the path to each environment file that you want to include in the overcloud deployment.
- You can configure Instance HA for your overcloud at any time after creating the undercloud. If you already deployed the overcloud, you need to rerun the **overcloud deploy** command with the new Instance HA files.

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

Upgrade from earlier versions

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

3.1. CONSIDERATIONS FOR SHARED STORAGE

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 boot 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 instances, and you can still 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 should boot from the same volume on another Compute node. As a result, the evacuated instances immediately restart their jobs because the OS image and the application data are stored on the OpenStack Block Storage volume..

CHAPTER 4. TESTING EVACUATION WITH INSTANCE HA

WARNING

The following procedure involves deliberately crashing a Compute node. Doing this forces the automated evacuation of instances through Instance HA.

1. Boot one or more instances on the overcloud before crashing the Compute node that hosts the instances to test.

```
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, using the **compute-n** format.

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

3. Crash the Compute node.

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

4. Wait a few minutes and then verify that these instances re-spawned on another Compute nodes.

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