Red Hat OpenStack Platform 17.1

Deploying Red Hat OpenStack Platform at scale

Hardware requirements and recommendations for large deployments

Last Updated: 2024-03-01

Red Hat OpenStack Platform 17.1 Deploying Red Hat OpenStack Platform at scale

Hardware requirements and recommendations for large deployments

OpenStack Team
rhos-docs@redhat.com

Legal Notice

Abstract

This guide contains several recommendations for deploying Red Hat OpenStack Platform at scale. These recommendations include hardware recommendations, undercloud tuning, and overcloud configuration.

Table of Contents

MAKING OPEN SOURCE MORE INCLUSIVE 3

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION 4

CHAPTER 1. RECOMMENDATIONS FOR LARGE DEPLOYMENTS 5

CHAPTER 2. RECOMMENDED SPECIFICATIONS FOR YOUR LARGE RED HAT OPENSTACK DEPLOYMENT . 6

2.1. UNDERCLOUD SYSTEM REQUIREMENTS 6
2.2. OVERCLOUD CONTROLLER NODES SYSTEM REQUIREMENTS 6
2.3. OVERCLOUD COMPUTE NODES SYSTEM REQUIREMENTS 9
2.4. RED HAT CEPH STORAGE NODES SYSTEM REQUIREMENTS 10

CHAPTER 3. RED HAT OPENSTACK DEPLOYMENT BEST PRACTICES 11

3.1. RED HAT OPENSTACK DEPLOYMENT PREPARATION 11
3.2. RED HAT OPENSTACK DEPLOYMENT CONFIGURATION 11
3.3. TUNING THE UNDERCLOUD 15

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 DOCUMENTATION

We appreciate your input on our documentation. Tell us how we can make it better.

Providing documentation feedback in Jira

Use the Create Issue form to provide feedback on the documentation. The Jira issue will be created in the Red Hat OpenStack Platform Jira project, where you can track the progress of your feedback.

1. Ensure that you are logged in to Jira. If you do not have a Jira account, create an account to submit feedback.
2. Click the following link to open a the **Create Issue** page: Create Issue
3. Complete the **Summary** and **Description** fields. In the **Description** field, include the documentation URL, chapter or section number, and a detailed description of the issue. Do not modify any other fields in the form.
4. Click **Create**.

CHAPTER 1. RECOMMENDATIONS FOR LARGE DEPLOYMENTS

Use the following undercloud and overcloud recommendations, specifications, and configuration for deploying a large Red Hat OpenStack Platform (RHOSP) environment. RHOSP 17.1 deployments that include more than 100 overcloud nodes are considered large environments. Red Hat has tested and validated optimum performance on a large scale environment of 750 overcloud nodes using RHOSP 17.1 without using minion.

For DCN-based deployments, the number of nodes from the central and edge sites can be very high. For recommendations about DCN deployments, contact Red Hat Global Support Services.

CHAPTER 2. RECOMMENDED SPECIFICATIONS FOR YOUR LARGE RED HAT OPENSTACK DEPLOYMENT

You can use the provided recommendations to scale your large cluster deployment.

The values in the following procedures are based on testing that the Red Hat OpenStack Platform Performance & Scale Team performed and can vary according to individual environments.

2.1. UNDERCLOUD SYSTEM REQUIREMENTS

For best performance, install the undercloud node on a physical server. However, if you use a virtualized undercloud node, ensure that the virtual machine has enough resources similar to a physical machine described in the following table.

Table 2.1. Recommended specifications for the undercloud node

System requirement	Description
Counts	1
CPUs	32 cores, 64 threads
Disk	500 GB root disk (1x SSD or 2x hard drives with 7200RPM; RAID 1)
Memory	256 GB
Network	25 Gbps network interfaces or 10 Gbps network interfaces

2.2. OVERCLOUD CONTROLLER NODES SYSTEM REQUIREMENTS

All control plane services must run on exactly 3 nodes. Typically, all control plane services are deployed across 3 Controller nodes.

Scaling controller services

To increase the resources available for controller services, you can scale these services to additional nodes. For example, you can deploy the **db** or **messaging** controller services on dedicated nodes to reduce the load on the Controller nodes.

To scale controller services, use composable roles to define the set of services that you want to scale. When you use composable roles, each service must run on exactly 3 additional dedicated nodes and the total number of nodes in the control plane must be odd to maintain Pacemaker quorum.

The control plane in this example consists of the following 9 nodes:

- 3 Controller nodes
- 3 Database nodes
- 3 Messaging nodes

For more information, see Composable services and custom roles in *Customizing your Red Hat OpenStack Platform deployment*.

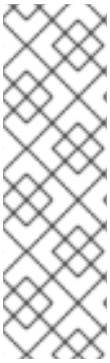
For questions about scaling controller services with composable roles, contact Red Hat Global Consulting.

Storage considerations

Include sufficient storage when you plan Controller nodes in your overcloud deployment.

If your deployment does not include Ceph storage, use a dedicated disk or node for Object Storage (swift) that overcloud workloads or Image (glance) services can use. If you use Object Storage on Controller nodes, use an NVMe device separate from the root disk to reduce disk use during object data storage.

The Block Storage service (cinder) requires extensive concurrent operations to upload volumes to the Image Storage service (glance). Images put a considerable I/O load on the Controller disk. This is not a recommended workflow for bulk operations, but if it is necessary, use SSD disks on the Controller node to provide a higher IOPS for such operations.



NOTE

- Older Telemetry services based on Ceilometer, gnocchi and the Alarming service (aodh) are disabled by default and are not recommended because of a negative affect on performance impact. If you enable these Telemetry services, gnocchi is I/O intensive and sends metrics to Object Storage nodes when Ceph is not enabled.
- All large scale testing is done on environments with a Director-deployed Ceph cluster.

CPU considerations

The number of API calls, AMQP messages, and database queries that the Controller nodes receive influences the CPU memory consumption on the Controller nodes. The ability of each Red Hat OpenStack Platform (RHOSP) component to concurrently process and perform tasks is also limited by the number of worker threads that are configured for each of the individual RHOSP components. To avoid a degradation of performance, the maximum number of worker threads for components with a large number of tasks on a Controller node is limited by the CPU count.

The number of worker threads for components that RHOSP director configures on a Controller is limited by the CPU count.

The following specifications are recommended for large scale environments with more than 700 nodes when you use Ceph Storage nodes in your deployment:

Table 2.2. Recommended specifications for Controller nodes when you use Ceph Storage nodes

System requirement	Description

System requirement	Description
Counts	3 Controller nodes with controller services contained within the Controller role. Optionally, to scale controller services on dedicated nodes, use composable services. For more information, see Composable services and customer roles in the *Customizing your Red Hat OpenStack Platform deployment* guide.
CPUs	2 sockets each with 32 cores, 64 threads
Disk	500 GB root disk (1x SSD or 2x hard drives with 7200RPM; RAID 1) 500GB dedicated disk for Swift (1x SSD or 1x NVMe)
Memory	384 GB
Network	25 Gbps network interfaces or 10 Gbps network interfaces. If you use 10 Gbps network interfaces, use network bonding to create two bonds: • Provisioning (bond0 - mode4); Internal API (bond0 - mode4); Project (bond0 - mode4) • Storage (bond1 - mode4); Storage management (bond1 - mode4)

The following specifications are recommended for large scale environments with more than 700 nodes when you do not use Ceph Storage nodes in your deployment:

Table 2.3. Recommended specifications for Controller nodes when you do not use Ceph Storage nodes

System requirement	Description
Counts	3 Controller nodes with controller services contained within the Controller role. Optionally, to scale controller services on dedicated nodes, use composable services. For more information, see Composable services and customer roles in the *Customizing your Red Hat OpenStack Platform deployment* guide.
CPUs	2 sockets each with 32 cores, 64 threads

System requirement	Description
Disk	500GB root disk (1x SSD) 500GB dedicated disk for Swift (1x SSD or 1x NVMe)
Memory	384 GB
Network	25 Gbps network interfaces or 10 Gbps network interfaces. If you use 10 Gbps network interfaces, use network bonding to create two bonds: ● Provisioning (bond0 - mode4); Internal API (bond0 - mode4); Project (bond0 - mode4) ● Storage (bond1 - mode4); Storage management (bond1 - mode4)

2.3. OVERCLOUD COMPUTE NODES SYSTEM REQUIREMENTS

When you plan your overcloud deployment, review the recommended system requirements for Compute nodes.

Table 2.4. Recommended specifications for Compute nodes

System requirement	Description
Counts	Red Hat has tested a scale of 750 nodes with various composable compute roles.
CPUs	2 sockets each with 12 cores, 24 threads
Disk	500 GB root disk
Memory	128 GB (64 GB per NUMA node); 2 GB is reserved for the host out by default. With Distributed Virtual Routing, increase the reserved RAM to 5 GB.
Network	25 Gbps network interfaces or 10 Gbps network interfaces. If you use 10 Gbps network interfaces, use network bonding to create two bonds: ● Provisioning (bond0 - mode4); Internal API (bond0 - mode4); Project (bond0 - mode4) ● Storage (bond1 - mode4)

2.4. RED HAT CEPH STORAGE NODES SYSTEM REQUIREMENTS

For Ceph Storage nodes system requirements, see the following resources:

- For more information about hardware prerequisites for Ceph nodes, see General principles for selecting hardware in the Red Hat Storage 4 *Hardware Guide*.
- For more information about deployment configuration for Ceph nodes, see Deploying Red Hat Ceph Storage and Red Hat OpenStack Platform together with director.
- For more information about changing the storage replication number, see Pools, placement groups, and CRUSH Configuration reference in the *Red Hat Ceph Storage Configuration Guide* .

CHAPTER 3. RED HAT OPENSTACK DEPLOYMENT BEST PRACTICES

Review the following best practices when you plan and prepare to deploy OpenStack. You can apply one or more of these practices in your environment.

3.1. RED HAT OPENSTACK DEPLOYMENT PREPARATION

Before you deploy Red Hat OpenStack Platform (RHOSP), review the following list of deployment preparation tasks. You can apply one or more of the deployment preparation tasks in your environment:

Set a subnet range for introspection to accommodate the maximum overcloud nodes for which you want to perform introspection at a time

When you use director to deploy and configure RHOSP, use CIDR notations for the control plane network to accommodate all overcloud nodes that you add now or in the future.

Enable Jumbo Frames for preferred networks

When a high-use network uses jumbo frames or a higher MTU, the network can send larger datagrams or TCP payloads and reduce the CPU overhead for higher bandwidth. Enable jumbo frames only for networks that have network switch support for higher MTU. Standard networks that are known to give better performance with higher MTU are the Tenant network, Storage network and the Storage Management network. For more information, see Configuring jumbo frames in *Installing and managing Red Hat OpenStack Platform with director* .

Set the World Wide Name (WWN) as the root disk hint for each node to prevent nodes from using the wrong disk during deployment and booting

When nodes contain multiple disks, use the introspection data to set the WWN as the root disk hint for each node. This prevents the node from using the wrong disk during deployment and booting. For more information, see Defining the Root Disk for multi-disk Ceph clusters in the *Installing and managing Red Hat OpenStack Platform with director* guide.

Enable the Bare Metal service (ironic) automated cleaning on nodes that have more than one disk

Use the Bare Metal service automated cleaning to erase metadata on nodes that have more than one disk and are likely to have multiple boot loaders. Nodes might become inconsistent with the boot disk due to the presence of multiple bootloaders on disks, which leads to node deployment failure when you attempt to pull the metadata that uses the wrong URL.
To enable the Bare Metal service automated cleaning, on the undercloud node, edit the **undercloud.conf** file and add the following line:

```
clean_nodes = true
```

Limit the number of nodes for Bare Metal (ironic) introspection

If you perform introspection on all nodes at the same time, failures might occur due to network constraints. Perform introspection on up to 50 nodes at a time.
Ensure that the **dhcp_start** and **dhcp_end** range in the **undercloud.conf** file is large enough for the number of nodes that you expect to have in the environment.

If there are insufficient available IPs, do not issue more than the size of the range. This limits the number of simultaneous introspection operations. To allow the introspection DHCP leases to expire, do not issue more IP addresses for a few minutes after the introspection completes.

3.2. RED HAT OPENSTACK DEPLOYMENT CONFIGURATION

Review the following list of recommendations for your Red Hat OpenStack Platform(RHOSP) deployment configuration:

Validate the heat templates with a small scale deployment

Deploy a small environment that consists of at least three Controllers, one Compute note, and three Ceph Storage nodes. You can use this configuration to ensure that all of your heat templates are correct.

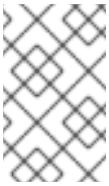
Improve instance distribution across Compute

During the creation of a large number of instances, the Compute scheduler does not know the resources of a Compute node until the resource allocation of previous instances is confirmed for the Compute node. To avoid the uneven spawning of Compute nodes, you can perform one of the following actions:

- Set the value of the **NovaSchedulerShuffleBestSameWeighedHosts** parameter to **true**:

```
parameter_defaults:
  NovaSchedulerShuffleBestSameWeighedHosts: `True`
```

- To ensure that a Compute node is not overloaded with instances, set **max_instances_per_host** to the maximum number of instances that any Compute node can spawn and ensure that the **NumInstancesFilter** parameter is enabled. When this instance count is reached by a Compute node, then the scheduler will no longer select it for further instance spawn scheduling.



NOTE

The **NumInstancesFilter** parameter is enabled by default. But if you modify the **NovaSchedulerEnabledFilters** parameter in the environment files, ensure that you enable the **NumInstancesFilter** parameter.

```
parameter_defaults:
  ControllerExtraConfig
    nova::scheduler::filter::max_instances_per_host: <maximum_number_of_instances>
    NovaSchedulerEnabledFilters:
    - AvailabilityZoneFilter
    - ComputeFilter
    - ComputeCapabilitiesFilter
    - ImagePropertiesFilter
    - ServerGroupAntiAffinityFilter
    - ServerGroupAffinityFilter
    - NumInstancesFilter
```

  - Replace **<maximum_number_of_instances>** with the maximum number of instances that any Compute node can spawn.

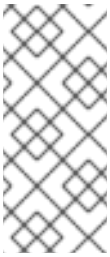
Scale configurations for the Networking service (neutron)

The settings in Table 3.1. were tested and validated to improve performance and scale stability on a large-scale openstack environment.

The server-side probe intervals control the timeout for probes sent by **ovsdb-server** to the clients: **neutron**, **ovn-controller**, and **ovn-metadata-agent**. If they do not get a reply from the client before the timeout elapses, they will disconnect from the client, forcing it to reconnect. The most likely scenario for a client to timeout is upon the initial connection to the **ovsdb-server**, when the client loads a copy of the database into memory. When the timeout is too low, the **ovsdb-server**

disconnects the client while it is downloading the database, causing the client to reconnect and try again and this cycle repeats forever. Therefore, if the maximum timeout interval does not work then set the probe interval value to zero to disable the probe.

If the client-side probe intervals are disabled, they use TCP keepalive messages to monitor their connections to the **ovsdb-server**.



NOTE

Always use tripleo heat template (THT) parameters, if available, to configure the required settings. Because manually configured settings will be overwritten by config download runs, when default values are defined in either THT or Puppet. Furthermore, you can only manually configure settings for existing environments, therefore the modified settings will not be applied to any new or replaced nodes.

Table 3.1. Recommended scale configurations for the Networking service

Setting	Description	Manual configuration	THT parameter
OVS server-side inactivity probe on Compute nodes	Increase this probe interval from 5 seconds to 30 seconds.	`ovs-vsctl set Manager . inactivity_probe=30000`	
OVN Northbound server-side inactivity probe on Controller nodes	Increase this probe interval to 180000 ms or set it to 0 to disable it.	`podman exec -u root ovn_controller ovn-nbctl --no-leader-only set Connection . inactivity_probe=180000`	
OVN Southbound server-side inactivity probe on Controller nodes	Increase this probe interval to 180000 ms or set it to 0 to disable it.	`podman exec -u root ovn_controller ovn-sbctl --no-leader-only set Connection . inactivity_probe=180000`	
OVN controller remote probe interval on Compute nodes	Increase this probe interval to 180000 ms or set it to 0 to disable it.	`podman exec -u root ovn_controller ovs-vsctl --no-leader-only set Open_vSwitch . external_ids:ovn-remote-probe-interval=180000`	OVNRemoteProbeInterval: 180000

Setting	Description	Manual configuration	THT parameter
Networking service client-side probe interval on Controller nodes	Increase this probe interval to 180000 ms or set it to 0 to disable it.	`crudini --set /var/lib/config-data/puppet-generated/neutron/etc/neutron/plugins/ml2/ml2_conf.ini ovn ovsdb_probe_interval 180000`	OVNOvsdbProbeInterval: 180000
Networking service **api_workers** on Controller nodes	Increase the default number of separate API worker processes from 12 to 16 or more, based on the load on the **neutron-server**.	`crudini --set /var/lib/config-data/puppet-generated/neutron/etc/neutron/neutron.conf DEFAULT api_workers 16`	NeutronWorkers: 16
Networking service **agent_down_time** on Controller nodes	Set **agent_down_time** to the maximum permissible number for very large clusters.	`crudini --set /var/lib/config-data/puppet-generated/neutron/etc/neutron/neutron.conf DEFAULT agent_down_time 2147483`	NeutronAgentDownTime: 2147483
OVN metadata **report_agent** on Compute nodes	Disable the **report_agent** on large installations.	`crudini --set /var/lib/config-data/puppet-generated/neutron/etc/neutron/neutron_ovn_metadata_agent.ini agent report_agent false`	
OVN **metadata_workers** on Compute nodes	Reduce the **metadata_workers** to the minimum on all Compute nodes to reduce the connections to the OVN Southbound database.	`crudini --set /var/lib/config-data/puppet-generated/neutron/etc/neutron/neutron_ovn_metadata_agent.ini DEFAULT metadata_workers 1`	NeutronMetadataWorkers: 1
OVN metadata **rpc_workers** on Compute nodes	Reduce the **rpc_workers** to the minimum on all Compute nodes.	`crudini --set /var/lib/config-data/puppet-generated/neutron/etc/neutron/neutron_ovn_metadata_agent.ini DEFAULT rpc_workers 0`	NeutronRpcWorkers: 0

Setting	Description	Manual configuration	THT parameter
OVN metadata client-side probe interval on Compute nodes	Increase this probe interval to 180000 ms or set it to 0 to disable it.	`crudini --set /var/lib/config-data/puppet-generated/neutron/etc/neutron/neutron_ovn_metadata_agent.ini ovn ovsdb_probe_interval 180000`	OVNOvsdbProbeInterval: 180000

Limit the number of nodes that are provisioned at the same time

Fifty is the typical amount of servers that can fit within an average enterprise-level rack unit, therefore, you can deploy an average of one rack of nodes at one time.
To minimize the debugging necessary to diagnose issues with the deployment, deploy a maximum of 50 nodes at one time. If you want to deploy a higher number of nodes, Red Hat has successfully tested up to 100 nodes simultaneously.

To scale Compute nodes in batches, use the **openstack overcloud deploy** command with the **--limit** option. This can result in saved time and lower resource consumption on the undercloud.

Disable unused NICs

If the overcloud has any unused NICs during the deployment, you must define the unused interfaces in the NIC configuration templates and set the interfaces to **use_dhcp: false** and **defroute: false**. If you do not define unused interfaces, there might be routing issues and IP allocation problems during introspection and scaling operations. By default, the NICs set **BOOTPROTO=dhcp**, which means the unused overcloud NICs consume IP addresses that are needed for the PXE provisioning. This can reduce the pool of available IP addresses for your nodes.

Power off unused Bare Metal Provisioning (ironic) nodes

Ensure that you power off any unused Bare Metal Provisioning (ironic) nodes in maintenance mode. Bare Metal Provisioning does not track the power state of nodes in maintenance mode and incorrectly reports the power state of nodes from previous deployments left in maintenance mode in a powered on state as off. This can cause problems with ongoing deployments if the unused node has an operating system with stale configurations, for example, IP addresses from overcloud networks. When you redeploy after a failed deployment, ensure that you power off all unused nodes.

3.3. TUNING THE UNDERCLOUD

Review this section when you plan to scale your Red Hat OpenStack Platform (RHOSP) deployment to configure your default undercloud settings.

Ensure that you increase the open file limit of your undercloud to 4096, by editing the following parameters in the **/etc/security/limits.conf** file:

```
* soft  nofile  4096
* hard  nofile  4096
```

Separate the provisioning and configuration processes

- To create only the stack and associated RHOSP resources, you can run the deployment

command with the **--stack-only** option.

- Red Hat recommends separating the stack and **config-download** steps when deploying more than 100 nodes.

Include any environment files that are required for your overcloud:

```
$ openstack overcloud deploy \
  --templates \
  -e <environment-file1.yaml> \
  -e <environment-file2.yaml> \
  ...
  --stack-only
```

- After you have provisioned the stack, you can enable SSH access for the **tripleo-admin** user from the undercloud to the overcloud. The **config-download** process uses the **tripleo-admin** user to perform the Ansible based configuration:

```
$ openstack overcloud admin authorize
```

- To disable the overcloud stack creation and to only apply the **config-download** workflow to the software configuration, you can run the deployment command with the **--config-download-only** option. Include any environment files that are required for your overcloud:

```
$ openstack overcloud deploy \
 --templates \
 -e <environment-file1.yaml> \
 -e <environment-file2.yaml> \
  ...
 --config-download-only
```

- To limit the **config-download** playbook execution to a specific node or set of nodes, you can use the **--limit** option.

- For scale-up operations, to only apply software configuration on the new nodes, you can use the **--limit** option with the **--config-download-only** option.

```
$ openstack overcloud deploy \
--templates \
-e <environment-file1.yaml> \
-e <environment-file2.yaml> \
...
--config-download-only --config-download-timeout --limit <Undercloud>,<Controller>,
<Compute-1>,<Compute-2>
```

If you use the **--limit** option always include **<Controller>** and **<Undercloud>** in the list. Tasks that use the **external_deploy_steps** interface, for example all Ceph configurations, are executed when **<Undercloud>** is included in the options list. All **external_deploy_steps** tasks run on the undercloud.

For example, if you run a scale-up task to add a Compute node that requires a connection to Ceph and you do not include **<Undercloud>** in the list, then this task fails because the Ceph configuration and **cephx** key files are not provided.

Do not use the **--skip-tags external_deploy_steps** option or the task fails.