Red Hat OpenStack Platform 13

Deploying Distributed Compute Nodes to Edge Sites

A guide to using Distributed Compute Nodes (DCN) at edge sites

Last Updated: 2021-03-06

Red Hat OpenStack Platform 13 Deploying Distributed Compute Nodes to Edge Sites

A guide to using Distributed Compute Nodes (DCN) at edge sites

OpenStack Team
rhos-docs@redhat.com

Legal Notice

Abstract

This guide explains how to design and manage a Distributed Compute Nodes deployment.

Table of Contents

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 .

CHAPTER 1. DEPLOYING DISTRIBUTED COMPUTE NODES TO EDGE SITES

1.1. OVERVIEW OF EDGE COMPUTING AND DCN

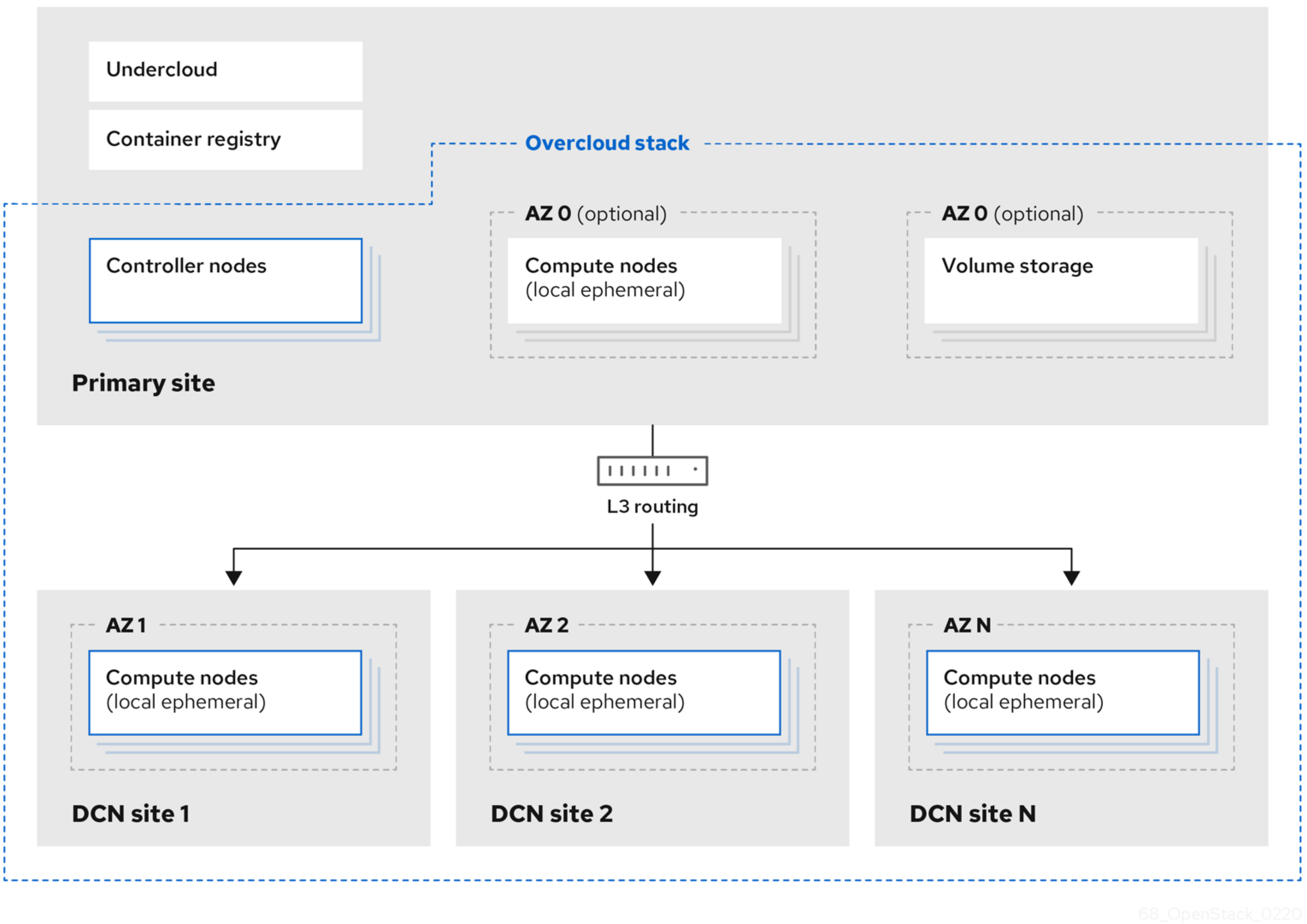
In Red Hat OpenStack Platform 13, *edge computing* refers to the practice of deploying Compute nodes to remote locations that might not be typical data centers. For example, a telco can deploy a Compute node to a remote location for data processing tasks.

You can use Red Hat OpenStack Platform 13 to implement edge computing using Distributed Compute Nodes (DCN). With this approach, you can deploy Compute nodes to the remote location, and the Controller services can be more tolerant of the network issues that might arise, such as connectivity and latency, among others.

1.2. DESIGNING YOUR DCN DEPLOYMENT

This chapter discusses some key architectural requirements and limitations to consider when planning your DCN edge deployment.

In this example design, the single overcloud heat stack spans multiple sites, with each site allocated to its own Availability Zone (AZ). The separate AZ approach allows you to target specific workloads to each site.



The following sections consider the reasons for these design decisions.

NOTE

It is possible to attach Block Storage (cinder) volumes at the primary site, using the same AZ name as the Compute nodes; you can then use **cross_az_attach=false** to prevent the volumes from being attached to the remote instance.

1.2.1. Availability zones

You must configure each edge site as a separate availability zone (AZ). When you deploy instances to this AZ, you can expect it to run on the remote Compute node. In addition, the central site must also be within a specific AZ (or multiple AZs), rather than the default AZ. Note that live migration between availability zones is not supported.

1.2.2. DCN network redundancy planning

You should consider redundancy when designing your DCN deployment. DCN allows you to host Compute nodes in different physical locations. To help tolerate a failure at one of these remote sites, consider using multiple redundant paths for your hosts. For your physical nodes, consider using redundant physical connections.

1.2.3. Limitations to consider

- **Only Compute** - In Red Hat OpenStack Platform 13, you can only deploy Compute nodes to edge sites. Other roles, such as storage, are not supported.

- **Network latency** - The edge Compute nodes are integrated with the wider overcloud deployment and must meet certain network latency requirements, with a round-trip time (RTT) that does not exceed 100ms. In some use cases, there is no L2 connectivity among edge sites and the central control plane; instead, there is a reasonably reliable WAN that operates with high network latency.

- **Network drop outs** - If the edge site temporarily loses its connection, then no OpenStack control plane API or CLI operations can be executed at the impacted edge site for the duration of the outage. For example, Compute nodes at the edge site are consequently unable to create a snapshot of an instance, issue an auth token, or delete an image. For more information on the impact of outages, see Section 1.2.7, "Planning for potential failures" below. General OpenStack control plane API and CLI operations remain functional during this outage, and can continue to serve any other edge sites that have a working connection.

- **Image sizing**:

  - **Overcloud node images** - Even though the edge site has its own provisioning network, overcloud node images are downloaded from the central undercloud node. These images are potentially large files that will be transferred across all necessary networks from the central site to the edge site during provisioning.

  - **Instance images** - Any base images stored in glance will traverse the WAN connection only during first use on that Compute node. For subsequent uses of the same image, it is locally cached on the Compute node. The images are distributed from the Controller node, and are cached on each Compute node after first use. There is no specific size limit for the images, as transfer times vary depending on the available bandwidth and network latency.

- **Provider networks** - This is the recommended networking approach for DCN deployments. If you use provider networks at remote sites, then you must consider that neutron does not place any limits or checks on where you can attach available networks. For example, if you use a provider network only in edge site A, you will need to make sure you do not try to attach to the provider network in edge site B. This is because there are no validation checks on the provider network when binding it to a Compute node.

- **Site-specific networks** - A limitation in DCN networking arises if you are using networks that are specific to a certain site: When deploying centralized neutron controllers with Compute nodes, there are no triggers in neutron to identify a certain Compute node as a remote node. Consequently, the Compute nodes receive a list of other Compute nodes and automatically form tunnels between each other; the tunnels are formed from edge to edge through the central site. If you are using VXLAN or Geneve, the result is that every Compute node at every site forms a tunnel with every other Compute node and Controller node whether or not they are actually local or remote. This is not an issue if you are using the same neutron networks everywhere. When using VLANs, neutron expects that all Compute nodes have the same bridge mappings, and that all VLANs are available at every site.

- **Additional sites** - If you need to expand from a central site to additional remote sites, you can use the **openstack** cli on the undercloud to add new network segments and subnets.

- **Autonomy** - There might be specific autonomy requirements for the edge site. This might vary depending on your requirements.

1.2.4. Planning your network

Routed L3 networks are recommended for DCN deployments:

1.2.4.1. Routed L3 networks

The remote location contains a physical node that serves as the Compute node. This node is attached to a local subnet that is designated as the provisioning network. A DHCP relay is necessary to pass provisioning traffic through to the undercloud node based in your main data center. You can then use director to deploy a Compute node onto the physical hardware based at the remote location.

For more information on L3 routing, see Spine Leaf Networking .

1.2.4.2. Improving resilience for North-South and East-West traffic

If spine-leaf does not suit your use case, and you decided to follow the traditional North-South/East-West topology, you can improve the reliability of the central control with the L3 HA networking configuration, which provides only North-South routing. You can improve the East-West routing effectiveness of edge networks with the DVR and local routing options, or highly available OVN.

IMPORTANT

OVN automatically uses DVR and does not require running Neutron DHCP/L3 agents. This approach can allow you to simplify your DCN deployment.

Alternatively, external provider networks allow you to use complex (and also efficient) routing topologies for North-South and East-West traffic.

1.2.4.3. Network recommendations

IMPORTANT

Neutron backends that perform tunnelling might not be optimal choices for edge DCN deployments, because of certain known issues.

For more information, see:

- Geneve - https://bugs.launchpad.net/tripleo/+bug/1808594
- VXLAN - https://bugs.launchpad.net/tripleo/+bug/1808062

Traditional or external provider networks with backbone routing at the edge could complement a custom distributed routing solution, such as the L3 Spine-Leaf topology. However, when there is a network failure that disconnects the edge from the central site, there is no SLA for the recovery time but only what the provider networks (or a particular SDN choice) can guarantee. For switched/routed/MPLS provider networks, that outage duration might span from 10s of milliseconds to a few seconds; outage thresholds are typically considered to be 15 seconds. These are subject to the various standards that apply here.

For dynamic IPv4 and stateful IPv6 IPAM deployments, you will also need DHCP on those provider networks in order to assign IPs to VM instances. External provider networks usually do not require Neutron DHCP agents, and can handle IPAM (and routing) on their own. For traditional or Neutron Routed Provider Networks, you should have a DHCP relay on every provider network when there is no L2 connectivity to edge over WAN, and must configure Neutron DHCP agents on controllers at the central site. Alternatively, DHCP agents can be moved to the edge; this approach does require highly reliable links between remote and central sites.

IMPORTANT

You might need a support exception when configuring DHCP relays or Neutron agents at the edge; this can also apply to Neutron-routed provider networks and external provider networks at edge sites.

Remote site availability could be improved by the ML2 and Open vSwitch plug-ins that allow the use of protocols (such as VRRP) to enable fast data plane failover allowed-address-pairs. Finally, classic network bonding and VF/PF DPDK bonds may strengthen overall tolerance for intermittent and short-lived hardware failures.

1.2.4.4. IPv6 Options

Another viable option is IPv6 for projects and infrastructure tunnels that interconnect the central site and the edge. However, IPv6 cannot be used for provisioning networks.

Some key benefits that IPv6 may provide for DCN:

- **Stateless Address Autoconfiguration (SLAAC)** - Requires no DHCP services placed on the provider networks.
- **Improved mobility** - Allows endpoints and NFV-like APIs to roam between different links and edge sites without losing connections and IP addresses.
- **End-to-end IPv6** - Has been shown to have better performance for large content networks. This is largely due to the performance impact that NAT has on most end-to-end IPv4 connections.

1.2.5. Configuration mechanisms

It is recommended that the edge locations use only provider networks. As a result, consider using **config-drive** or IPv6 SLAAC, or another configuration mechanism alongside **cloud-init**. **Config-drive** can serve as the metadata API server and uses virtual media capabilities of the BMC controller, so that no Neutron Metadata/DHCP agents, nor DHCP relays are required for VM instances to obtain IP addresses at edge sites. This also does require that the WAN between the remote site and central site is live during deployment of a VM, but after that the VM can run independently without a connection to the central site.

IMPORTANT

Certain instances might need to be configured with a static IP that matches the corresponding Neutron port. This can arise if they do not support **config-drive**, such as NFV-specific VNF appliances.

1.2.6. Storage recommendations

DCN is only available for Compute services using local ephemeral storage. Design your edge cloud applications with consideration for data availability, locality awareness, and replication mechanisms.

1.2.7. Planning for potential failures

This section describes the possible impact that certain types of outages can cause. It also discusses high availability recommendations to help mitigate these risks.

1.2.7.1. Plan your overcloud for potential failures

To properly manage a DCN failure scenario, start at the deployment planning process. For example, if you have certain SLA expectations, then you might consider choosing a particular SDN solution that uses provider networks.

For more information on overcloud planning, see https://access.redhat.com/documentation/en-us/red_hat_openstack_platform/13/html/director_installation_and_usage/chap-planning_your_overcloud.

1.2.7.2. Loss of control plane

A failure of the central control plane has no different impact than you might find in another product architecture, and there are no actions required that are specific to a DCN deployment. This section explains the impact this failure would have on a DCN site.

NOTE

A network outage to an edge site is more common than an outage to the central control plane, which is considered to be a catastrophic and extremely rare event.

This type of failure affects all DCN edge sites, as there are no autonomous control planes at the edge. As a result, no OpenStack control plane API or CLI operations can be executed. For example, you cannot create a snapshot of an instance, issue an auth token, or delete an image.

Aside from the impact to the API, this event normally causes no workload downtime for the edge sites, and should be handled by the typical High Availability (HA) deployments and procedures.

The Compute instances hosted at the edge site should continue to run unless the hosts are rebooted. If the instance stops, you must bring the control plane back online before you can recover the stopped or crashed workloads. If the Neutron DHCP agent is centralized, and your topology is forwarding DHCP requests to the central site, any VM instances that are trying to renew their IPs will eventually time out and lose connectivity.

IMPORTANT

Failure of a single Compute node normally affects only its edge site, and is not expected to cause downtime for the neighbor edge sites, or the central control plane.

When the control plane's uplink is restored, OpenStack's infrastructure services,such as Compute, automatically reconnect to the control services. However, you cannot resume any control plane operations that have timed out; you must retry them manually.

NOTE

You must configure each DCN edge site as a separate Availability Zone (AZ).

1.3. USING DIRECTOR TO DEPLOY NODES TO THE EDGE

This chapter explains the steps required to configure the undercloud and overcloud for DCN edge computing.

1.3.1. Configuring the undercloud

This section describes the steps required to configure the undercloud for DCN:

1.3.1.1. Using direct deploy instead of iSCSI

In a default undercloud configuration, ironic deploys nodes using the **iscsi** deploy interface. When using the **iscsi** deploy interface, the deploy ramdisk publishes the node's disk as an iSCSI target, and the **ironic-conductor** service then copies the image to this target.

For a DCN deployment, network latency is often a concern between the undercloud and the distributed compute nodes. Considering the potential for latency, the distributed compute nodes should be configured to use the **direct** deploy interface in the undercloud. For more information on configuring the **deploy** interface, see Section 1.3.1.3, "Configuring nodes to use the deploy interface" .

When using the **direct** deploy interface, the deploy ramdisk downloads the image over HTTP from the undercloud's Swift service, and copies it to the node's disk. HTTP is more resilient when dealing with network latency than iSCSI, so using the **direct** deploy interface provides a more stable node deployment experience for distributed compute nodes.

1.3.1.2. Configuring the Swift temporary URL key

Images are served by Swift and are made available to nodes using a HTTP URL, over the **direct** deploy interface. You must configure Swift with a temporary URL key so that it can create temporary URLs. Swift uses this key value for cryptographic signing and verification of the temporary URLs.

The following commands demonstrate how to configure the setting. In this example, **uuidgen** is used to randomly create a key value. Choose a unique key value that is difficult to guess. For example:

```
$ source ~/stackrc
$ openstack role add --user admin --project service ResellerAdmin
$ openstack --os-project-name service object store account set --property Temp-URL-Key=$(uuidgen
| sha1sum | awk '{print $1}')
```

1.3.1.3. Configuring nodes to use the deploy interface

This section describes how to configure the deploy interface for new and existing nodes:

1.3.1.3.1. Configuring the interface setting for new nodes

You can specify the deploy interface directly in the JSON structure for each node. For example, see the following **“deploy_interface”: “direct”** setting:

```
{
  "nodes":[
     {
        "mac":[
           "bb:bb:bb:bb:bb:bb"
        ],
        "name":"node01",
        "cpu":"4",
        "memory":"6144",
        "disk":"40",
        "arch":"x86_64",
        "pm_type":"ipmi",
        "pm_user":"admin",
        "pm_password":"p@55w0rd!",
        "pm_addr":"192.168.24.205",
        “deploy_interface”: “direct”
     },
     {
        "mac":[
           "cc:cc:cc:cc:cc:cc"
        ],
        "name":"node02",
        "cpu":"4",
        "memory":"6144",
        "disk":"40",
        "arch":"x86_64",
        "pm_type":"ipmi",
        "pm_user":"admin",
        "pm_password":"p@55w0rd!",
        "pm_addr":"192.168.24.206"
        “deploy_interface”: “direct”
     }
  ]
}
```

1.3.1.3.2. Configure the interface setting for existing nodes

You can update existing nodes to use the **direct** deploy interface. For example:

```
$ openstack baremetal node set --deploy-interface direct 4b64a750-afe3-4236-88d1-7bb88c962666
```

1.4. CONFIGURING THE OVERCLOUD FOR DCN

This section describes the steps required to configure the overcloud for DCN:

1.4.1. Creating deployment roles

The default Compute role that is defined at **/usr/share/openstack-tripleo-heat-templates/roles/Compute.yaml** is the only role that is supported for DCN deployment at edge sites. However, you can still copy the role and modify the names to allow for different software configurations at the edge sites.

The following example demonstrates how two different roles can use the same set of services for two separate AZs:

```
###############################################################################
# Role: ComputeAZ1
###############################################################################
- name: ComputeAZ1
 description: |
  Basic Compute Node role for AZ1
 CountDefault: 1
 networks:
  - InternalApi1
  - Tenant1
  - Storage1
 HostnameFormatDefault: '%stackname%-compute-az1-%index%'
 RoleParametersDefault:
  TunedProfileName: "virtual-host"
 disable_upgrade_deployment: True
 ServicesDefault:
  - OS::TripleO::Services::Aide
  - OS::TripleO::Services::AuditD
  - OS::TripleO::Services::CACerts
  - OS::TripleO::Services::CephClient
  - OS::TripleO::Services::CephExternal
  - OS::TripleO::Services::CertmongerUser
  - OS::TripleO::Services::Collectd
  - OS::TripleO::Services::ComputeCeilometerAgent
  - OS::TripleO::Services::ComputeNeutronCorePlugin
  - OS::TripleO::Services::ComputeNeutronL3Agent
  - OS::TripleO::Services::ComputeNeutronMetadataAgent
  - OS::TripleO::Services::ComputeNeutronOvsAgent
  - OS::TripleO::Services::Docker
  - OS::TripleO::Services::Fluentd
  - OS::TripleO::Services::Ipsec
  - OS::TripleO::Services::Iscsid
  - OS::TripleO::Services::Kernel
  - OS::TripleO::Services::LoginDefs
  - OS::TripleO::Services::MySQLClient
  - OS::TripleO::Services::NeutronBgpVpnBagpipe
  - OS::TripleO::Services::NeutronLinuxbridgeAgent
  - OS::TripleO::Services::NeutronVppAgent
  - OS::TripleO::Services::NovaCompute
  - OS::TripleO::Services::NovaLibvirt
  - OS::TripleO::Services::NovaMigrationTarget
  - OS::TripleO::Services::Ntp
  - OS::TripleO::Services::ContainersLogrotateCrond
  - OS::TripleO::Services::OpenDaylightOvs
  - OS::TripleO::Services::Rhsm
  - OS::TripleO::Services::RsyslogSidecar
  - OS::TripleO::Services::Securetty
  - OS::TripleO::Services::SensuClient
  - OS::TripleO::Services::SkydiveAgent
  - OS::TripleO::Services::Snmp
  - OS::TripleO::Services::Sshd
  - OS::TripleO::Services::Timezone
  - OS::TripleO::Services::TripleoFirewall
  - OS::TripleO::Services::TripleoPackages
  - OS::TripleO::Services::Tuned
  - OS::TripleO::Services::Vpp
  - OS::TripleO::Services::OVNController
  - OS::TripleO::Services::OVNMetadataAgent
  - OS::TripleO::Services::Ptp
###############################################################################
# Role: ComputeAZ2
###############################################################################
- name: ComputeAZ2
  description: |
    Basic Compute Node role for AZ2
  CountDefault: 1
  networks:
    - InternalApi2
    - Tenant2
    - Storage2
  HostnameFormatDefault: '%stackname%-compute-az2-%index%'
  RoleParametersDefault:
    TunedProfileName: "virtual-host"
  disable_upgrade_deployment: True
  ServicesDefault:
    - OS::TripleO::Services::Aide
    - OS::TripleO::Services::AuditD
    - OS::TripleO::Services::CACerts
    - OS::TripleO::Services::CephClient
    - OS::TripleO::Services::CephExternal
    - OS::TripleO::Services::CertmongerUser
    - OS::TripleO::Services::Collectd
    - OS::TripleO::Services::ComputeCeilometerAgent
    - OS::TripleO::Services::ComputeNeutronCorePlugin
    - OS::TripleO::Services::ComputeNeutronL3Agent
    - OS::TripleO::Services::ComputeNeutronMetadataAgent
    - OS::TripleO::Services::ComputeNeutronOvsAgent
    - OS::TripleO::Services::Docker
    - OS::TripleO::Services::Fluentd
    - OS::TripleO::Services::Ipsec
    - OS::TripleO::Services::Iscsid
    - OS::TripleO::Services::Kernel
    - OS::TripleO::Services::LoginDefs
    - OS::TripleO::Services::MySQLClient
    - OS::TripleO::Services::NeutronBgpVpnBagpipe
    - OS::TripleO::Services::NeutronLinuxbridgeAgent
    - OS::TripleO::Services::NeutronVppAgent
    - OS::TripleO::Services::NovaCompute
    - OS::TripleO::Services::NovaLibvirt
    - OS::TripleO::Services::NovaMigrationTarget
    - OS::TripleO::Services::Ntp
    - OS::TripleO::Services::ContainersLogrotateCrond
    - OS::TripleO::Services::OpenDaylightOvs
    - OS::TripleO::Services::Rhsm
    - OS::TripleO::Services::RsyslogSidecar
    - OS::TripleO::Services::Securetty
    - OS::TripleO::Services::SensuClient
    - OS::TripleO::Services::SkydiveAgent
    - OS::TripleO::Services::Snmp
    - OS::TripleO::Services::Sshd
    - OS::TripleO::Services::Timezone
    - OS::TripleO::Services::TripleoFirewall
    - OS::TripleO::Services::TripleoPackages
    - OS::TripleO::Services::Tuned
    - OS::TripleO::Services::Vpp
    - OS::TripleO::Services::OVNController
    - OS::TripleO::Services::OVNMetadataAgent
    - OS::TripleO::Services::Ptp
```

Note that the roles have the same set of services, even though they are differently named. You can use the **RoleParameters** parameter to configure individual service parameters. For example:

```
parameter_defaults:
 RoleParameters:
   ComputeAZ1:
     NovaReservedHostMemory: 4096
   ComputeAZ2:
     NovaReservedHostMemory: 2048
```

In the above example, nodes deployed with the **ComputeAZ1** role have **NovaReservedHostMemory** set to **4096**. Nodes deployed with the **ComputeAZ2** role have **NovaReservedHostMemory** set to **2048**.

To deploy the node to the AZ, follow your typical provisioning procedures of tagging a physical node to the new role and booting up in the provisioning network. For more information on director deployment, see https://access.redhat.com/documentation/en-us/red_hat_openstack_platform/13/html/director_installation_and_usage/.

1.4.2. Post-install configuration

1.4.2.1. Creating availability zones

After you have used director to provision your DCN at the edge, you must create an availability zone for the edge site. When you have created an AZ, you can launch an instance within it; the new instance runs on the DCN.

As part of this deployment, you created a new AZ for your central site. Add the applicable Compute nodes to the new central AZ. For more information on AZ configuration, see https://access.redhat.com/documentation/en-us/red_hat_openstack_platform/13/html-single/instances_and_images_guide/index#section-manage-host-aggregates.

IMPORTANT

Configure **default_schedule_zone** to match your central AZ setting, as this value is used as the default for new instances.

1.5. OPERATIONAL PROCEDURES

1.5.1. Monitoring network latency

The DCN deployment at an edge site is sensitive to high latency of 100ms or higher. As a result, you should perform network monitoring for these connections to generate alerts if a latency threshold is reached.

1.5.2. Managing DCN updates

For BAU operations, you can manage DCN nodes in the same way that you manage your local Compute nodes. For example, you can expect to see no difference between how you might apply updates to DCN or standard Compute nodes. You can control which nodes receive updates, which means that you can update the Compute nodes per site, instead of all at once.

For example, you can use the **--nodes** parameter of **openstack overcloud update run** to perform updates against either a list of computes, or per role. Since each remote site has its own role, you can perform per-site minor updates, or be even more precise with specific nodes. For more information, see https://access.redhat.com/documentation/en-us/red_hat_openstack_platform/13/html-single/keeping_red_hat_openstack_platform_updated/#updating_all_compute_nodes.

NOTE

When updating Controller nodes, recall that a DCN deployment uses non-standard Controller roles for spine-leaf configuration. For example, you might need to run the following command:

```
$ openstack overcloud update run --nodes Controller0
```

1.6. SCALING YOUR EDGE DEPLOYMENT

This chapter discusses how to scale out your undercloud and overcloud nodes.

1.6.1. Scaling the overcloud

The number of nodes that you can add to your overcloud is limited by what the undercloud can manage. As the amount of nodes in the overcloud deployment increases, so do the number of resources that need to be managed.

It is recommended that you do not exceed 300 nodes in any single deployment. This number heavily depends on the hardware sizing, overcloud configuration, and other factors. Previous testing with physical hardware has shown that a single undercloud can deploy an overcloud with 150 total nodes. Deployments with a higher number of nodes might be successful, but this has not been tested. Note that this recommendation might potentially increase as ongoing test results are reviewed.

The time it takes to complete a deployment is based on multiple factors, including undercloud resources, networking bandwidth, and node boot time. However, the single largest contributing factor to deployment time is the number of nodes in the deployment.

1.6.2. Using a single stack deployment

When deploying a DCN architecture in Red Hat OpenStack Platform 13, all nodes managed by the undercloud must be deployed in the same stack. This is because the undercloud can only deploy one overcloud stack. If additional overcloud stacks or deployments are needed, then you must use a separate undercloud for each deployment. When using multiple underclouds with multiple deployments, the deployments are assumed to be separate clouds, with no operational connectivity between them.

1.6.3. Scaling up incrementally

Since all nodes are deployed in a single stack, any failure during the deployment on any node causes the entire deployment to stop, fail, and report on the failure. As the number of nodes in the deployment increases, so does the likelihood that new issues occur during the deployment. You must resolve these issues before the deployment can continue. You can address these issues by either fixing them or by removing the affected node or nodes from the deployment altogether. When the issue is resolved, re-run the deployment.

To help mitigate the risk of an interrupted deployment, run the deployment incrementally. Start with the nodes for the control plane services, (Controller, among others,) and a small number of Compute nodes. After the successful initial deployment, you can add batches of Compute nodes in manageable sizes (10, 20, and so on) by scaling up the count of Compute nodes. The batch size that you add on each scale up attempt is dependent on environmental factors and your overall confidence in the absence of hardware and networking issues.

IMPORTANT

When scaling your deployment, you should only add Compute nodes in small manageable sizes; then review your deployment's health before continuing to add more.

1.7. TROUBLESHOOTING

1.7.1. Checking network latency

If your edge site DCN is behaving unexpectedly, check the network latency. High latency can manifest in unexpected ways in a DCN, depending on how high it is. For example, what might initially seem like network failure might actually be a consequence of high latency.

1.7.2. Node introspection failure

If your node deployment fails with the errors: **Connection timed out** and **No bootable device**, check that the node is configured to use **direct** deploy instead of **iscsi**. For more information, see Section 1.3.1.3, "Configuring nodes to use the deploy interface" .