ORACLE

OpenShift Cluster Setup with Assisted Installer on Private Cloud Appliance

Step-by-step to deploy OpenShift Clusters using Assisted Installer on Private Cloud Appliance

Version 11.0

Public

ORACLE

Purpose statement

This document provides step-by-step instructions to deploy Redhat OpenShift Clusters on Oracle's Private Cloud Appliance (PCA)

Disclaimer

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle software license and service agreement, which has been executed and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement, nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

This document is for informational purposes only and is intended solely to assist you in planning for the implementation and upgrade of the product features described. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, timing, and pricing of any features or functionality described in this document remains at the sole discretion of Oracle. Due to the nature of the product architecture, it may not be possible to safely include all features described in this document without risking significant destabilization of the code.

Introduction

In today's changing world of business IT, automation and native solutions play a crucial role in ensuring flexibility, scalability and effectiveness. As more companies embrace various Kubernetes platform as OpenShift for managing containers, the importance of deployment processes becomes clear. This guide helps to address the specific preference of using Redhat's OpenShift for Kubernetes by offering a step-by-step instructions of installing OpenShift on Oracle's Private Cloud Appliance.

OpenShift offers a variety of features, including automated installation, upgrades, and life cycle management. It also provides advanced security and monitoring, integrated storage, and CI/CD pipeline management. OpenShift is powered by Kubernetes, an open-source project that streamlines the entire application lifecycle, from development to delivery to management

This content is provided for informational purposes and self-supported guidance only. Consultancy or other assistance related to the content is not covered under the Oracle Support contract or associated service requests. If you have questions or additional needs, then please do reach out to your Oracle Sales contact directly.

ORACLE

Access and Considerations

Before getting into the steps for setting up OpenShift on Oracle Private Cloud Appliance (PCA), it's important to make sure you have all the necessary prerequisites ready. This section outlines the elements needed to carry out the installation with minimal disruptions.

The prerequisites covered in this document encompass operational aspects, including:

- **Access to Private Cloud Appliance:** Make sure you have an active account with the appropriate permissions to create and manage resources on Oracle Compute Cloud@Customer.
- **Access to Redhat Assisted Installer Clusters Portal:** During the step-by-step deployment process, you will be taking advantage of the Redhat Hybrid Cloud Console Portal: https://console.redhat.com/openshift/assisted-installer/clusters
- **Access Credentials:** You will need a bastion / jump host within your PCA target Compartment's Tenancy to connect with an external Terraform website to download and install Terraform. You will also use this Bastion host for OCI CLI, Running the Terraform Scripts and preparing the OpenShift Clusters.
- **Networking Considerations:** Ensure that your network setup allows for communication, which includes setting up VCN (Virtual Cloud Network) configuring subnets and managing security lists. If your PCA is disconnected, you will need to access external networks via a DRG.
- **Infrastructure Planning:** Develop a strategy outlining the resources needed capacity projections and design aspects customized to suit your organizations requirements.

By fulfilling these criteria, you will be ready to follow the instructions outlined in this paper guaranteeing a seamless and effective deployment of your Redhat OpenShift Clusters on Oracle's Private Cloud Appliance.

Bastion Configuration

A Bastion Host, also referred to as a jump server, provides a server created to enable access to a private network from an external network, like the internet. For the OpenShift deployment on Private Cloud Appliance, a bastion host is utilized to host Terraform scripts utilized to configure infrastructure via OCI CLI and build OpenShift Clusters. It will also be utilized for reaching resources and instances within a subnet in your Private Cloud Appliance environment.

Advantages of Using a Bastion Host are:

- Access Point: Acts as an entry point for administrators and users to connect to instances in the private network without exposing them directly to the internet.
- Offers a controlled gateway thereby reducing the vulnerability surface.
- Isolation of Critical Resources: Ensures that computing resources within the network are safeguarded and isolated from internet exposure. Helps in minimizing risks associated with attacks on systems.
- Monitoring: Centralizes access logging simplifying tracking and auditing access to resources. Can be integrated with security information and event management (SIEM) systems, for monitoring capabilities.

By implementing a bastion host, you can securely and efficiently manage access to your Oracle Compute Cloud@Customer or Private Cloud Appliance resources, ensuring that your private instances remain protected and accessible only through a secure, controlled entry point.

The installation of the OCI CLI will be revisited in the first section of this procedure:

Install OCI CLI in your bastion host. For the step-by-step and how to install OCI CLI and properly setup the OCI CLI configuration file in your instance, refer to the following links below: https://docs.oracle.com/en-us/iaas/Content/API/SDKDocs/cliinstall.htm and https://docs.oracle.com/en/engineered-systems/private-cloud-appliance/3.0-latest/user/user-usr-ce-cli.html#usr-cli-obtain-cabundle.

Overview

The installation process outlined below will involve building two hosts within the targeted compartment of the OpenShift Cluster installation.

The first instance to be installed will be a “bastion” host, within the designated compartment, and will be used to run two Terraform scripts. The first script will be used for building IAM Resources on your Private Cloud Appliance (PCA)The second Terraform Script will be for building the Infrastructure Resources on the PCA System to support the OpenShift Cluster (the OpenShift VCN, Public and Private Subnets, Load Balancers, Internet GW, NAT GW, and DNS Server). It will include all the resources needed to allow the Control Plane and Worker Nodes to function that form a cluster. The bastion will be installed in the designated OpenShift Compartment and must be configured to communicate through a designated PCA DRG Subnet or Internet GW Subnet within the PCA’s parent tenancy.

Finally, three Control Plane Nodes and three Worker Nodes will be provisioned, along with configuring the external and internal Load Balancers that form the Cluster.

Bastion Server - Prerequisites

In preparation to running Terraform Scripts, this section will be focused on the installation of Terraform and OCI CLI.

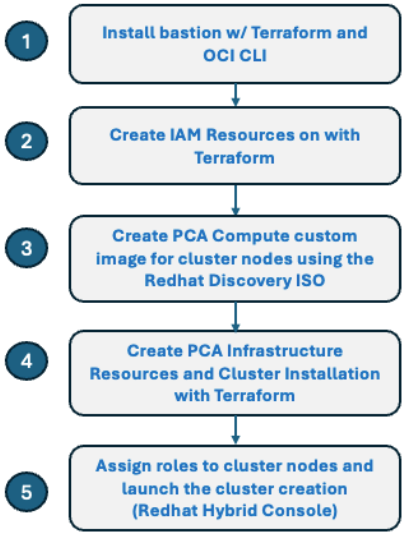
Currently PCA does not support OCI Resource Manager Stack (RMS). To run the customized PCA Terraform script, a Bastion Server will be installed on the within the targeted OpenShift Cluster deployment compartment. Terraform and OCI CLI will then be loaded on to the Bastion Server for execution.

Public IP Addresses: You will need a set of 3 Public IP Addresses. Use of CIDR notations will provide ample number of IP Addresses

The following is an overview of installations and directories that will be placed on to the bastion host for the execution of Terraform scripts. Before running the Terraform scripts, Terraform and oci-cli will be installed and configured.

The following figure provides an overview of the installation process.

Installation Overview – Assisted Installer

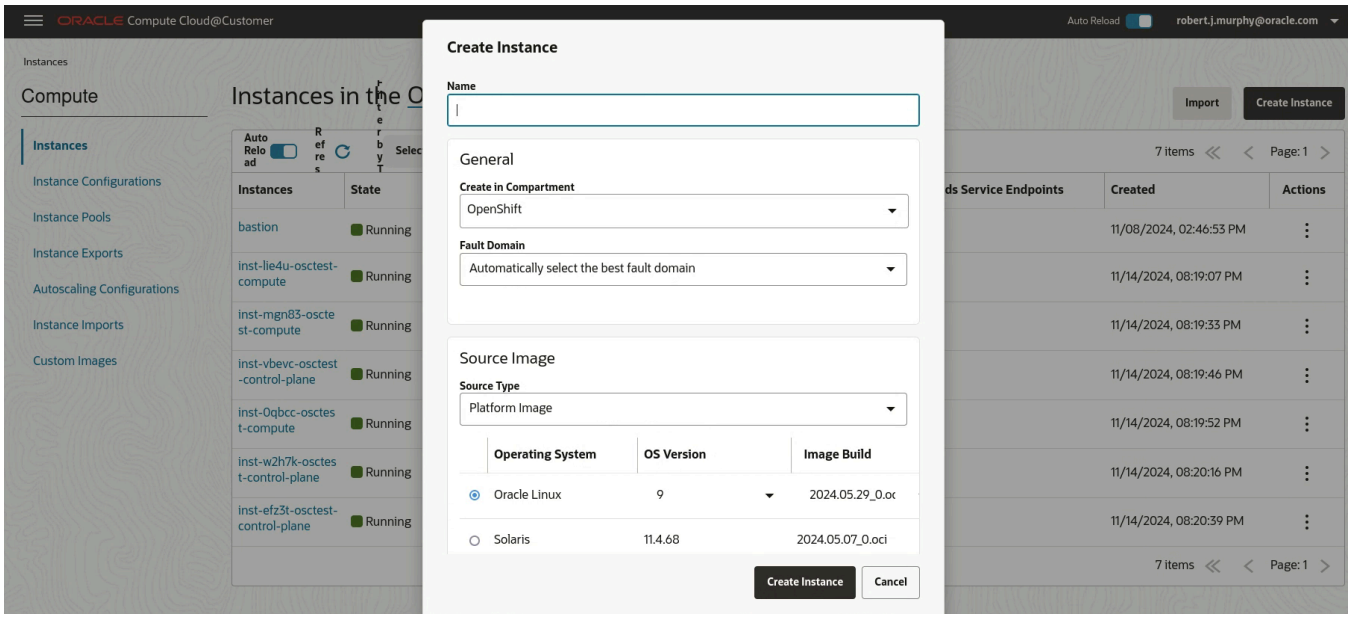


- **Currently C3/PCA does not support: Developer Services, Resource Manager**
- **The current procedure includes manual steps to set up running terraform scripts within target the compartment**
- **(1) Install an Instance bastion server** within Target Compartment & Install OCI CLI and Terraform
  - A preloaded custom image will be available w/ both
- **(2) Run Script-1 :** Create IAM Resources – Dynamic Groups and Policies to allow instances to manage resources
- **(3) Create Discovery ISO on Red Hat OpenShift Portal:** Download customize Discovery ISO to PCA for Automated Install in Script-2
  - Includes adding customized Manifests, CSI and CCM Files
- **(4) Run Script-2:** Create Infra Resources (VCN, LB's, IGW, NAT, Etc) and Launch typical 3 x 3 Cluster (3 CP Nodes / 3 Worker Nodes)
  - Topology is shown on next page
- **(5) Assign Roles / Cluster Creation:** Assign roles, upload manifests & create cluster via Red Hat Assisted Installer Portal
- **(6) Launch OpenShift Console & Advanced Cluster Manager**

Figure-1

Bastion Installation

1. Create a Compartment on your PCA that will be managing this deployment.
2. Login to the PCA System's Console
3. Click on "Compute" and switch to the Compartment made in Step-1
4. Create an Instance
   a. **Name:** Provide a name of your choosing (for the remaining of this procedure, we will refer to this as "bastion")
   b. **Create in Compartment:** Choose the Compartment used to build your Cluster
   c. **Fault Domain**: Automatically select …
   d. **Source Image:** Oracle Linux 9
   e. **Shape:** VM.PCAStandard1.Flex
   f. **OCPUs:** 2
   g. **Memory (GBs):** 20
   h. Subnet:
      1. For disconnected C3's, the VCN and Subnet chosen for VNIC attachment must have a DRG for external access
      2. Choose an appropriate **VCN** and **Subnet** for external access if you are using a DRG.
         1. Alternatively, choose a Public Subnet with IGW Access
   i. Public IP:
      1. Deselect for using DRG based install
      2. Select if using connected PCA and use of Public Subnet w/ IGW
   j. Private IP:
      1. Keep (Optional)
   k. **SSH Key:** Upload or Cut and Paste the Public Key into the dialogue box
   l. Create the Instance
   m. Record the Private IP – You will be using this to SSH into in the next step.

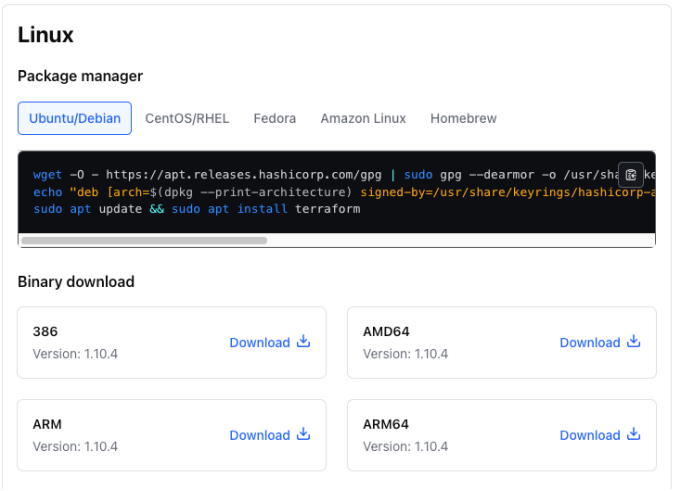


5. SSH into the **bastion** Instance using the Public or Private IP Address (if a DRG is utilized):
    a. [my_laptop ~]$ ssh opc@<bastion_IP_Address>

Terraform Installation

1. Download the Terraform binary AMD64 zip file onto your laptop from: https://developer.hashicorp.com/terraform/install

    Note: The Terraform binary version will continue to be updated. Choose the latest version.



2. SSH into your bastion instance
    a. [user@my_laptop ~]$ ssh opc@<bastion_private_IP_address>
3. On the bastion instance, make a directory “prerequisite”
    a. [bastion ~]$ mkdir /home/opc/prerequisite/
4. Transfer the Terraform zip file from your laptop to the bastion host via secure copy, scp:

a. [user@ my_laptop ~]$ scp terraform_1.8.3_linux_amd64.zip opc@<bastion_private_IP_address>:/home/opc/prerequisite/

5. Unzip the Terraform Binary and move it to the **bin** directory
   a. [opc@bastion ~ ./prerequisite]$ unzip terraform_1.8.3_linux_amd64.zip
   b. [opc@bastion ~ ./prerequisite]$ sudo mv terraform /usr/local/bin
   c. [opc@bastion ~ ./prerequisite]$ sudo chmod +x /usr/local/bin/terraform
   d. Verify the Terraform installation was a success.
      1. [opc@bastion ~ ./prerequisite]$ terraform version
      2. The processor type and version number will be returned

Installing and Configuring OCI CLI

1. **Install the OCI CLI**
   a. Goto: Offline Installation documentation
      1. **Choose**: Choose an "Oracle Linux 9 Offline" version
      2. **Example**: oci-cli-3.49.2-Oracle-Linux-9-Offline
   b. Download the zip file to your local machine
   c. Transfer from the file from your local machine to the targeted bastion server and store in the prerequisite folder:
      1. [user@my_laptop ~]$ scp oci-cli-3.49.2-Oracle-Linux-9-Offline.zip opc@<bastion_private_IP_address>:/home/opc/prerequisite/
   d. SSH into the bastion the bastion server
      1. [user@my_laptop ~]$ ssh opc@<bastion_private_IP_address>
   e. Unzip the oci cli file:
      1. [opc@bastion ~ ./prerequisite]$ unzip oci-cli-3.49.2-Oracle-Linux-9-Offline.zip
   f. Install oci cli
      1. [opc@bastion ~ ./prerequisite/]$ cd oci-cli-installation
      2. [opc@bastion ~ ./prerequisite/oci-cli-installation]$ bash install.sh --offline-install
         1. Answer **no** to any changes regarding paths and defaults presented **(leave blank)**

2. **Configure the OCI CLI**
   a. [opc@bastion ~]$ oci setup config
   b. [opc@bastion ~]$ **Enter a location for your config [/home/opc/.oci/config]:** Enter (keep path)
   c. [opc@bastion ~]$ **Enter a user OCID**: Enter your User OCID (Goto your PCA Console of your and navigate to your Identity and obtain your OCID)
   d. [opc@bastion ~]$ **Enter your Tenancy OCID**: Enter your Tenancy OCID (This can be found on the PCA Console in the upper right corner, Profile.)
   e. [opc@bastion ~]$ **Enter a Region by index or name**: Select the Region name's representative number and enter it
   f. [opc@bastion ~]$ **Do you want to generate a new API Signing RSA key pair?** Y
      1. You can opt for N, if you already have an API signing key on the PCA.
   g. [opc@bastion ~]$ **Enter a directory for your keys to be created [/home/opc/.oci]:** Enter (keep default)
   h. [opc@bastion ~]$ **Enter a name for your key [oci_api_key]: Enter** (keep default)
   i. [opc@bastion ~]$ **Enter a passphrase for your private key:** select N/A (or choose a passphrase if desired)

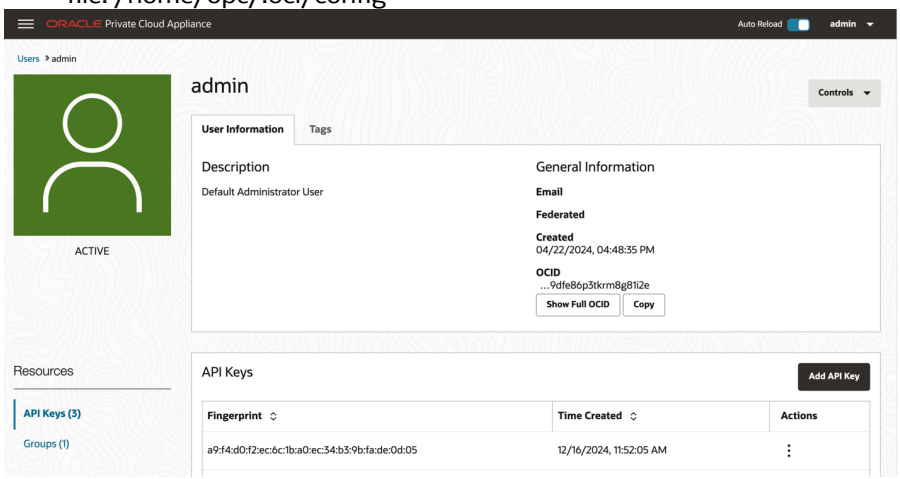
Note: If you declined a new api key pair, place your keys in this directory with the file names shown above.

This completes the initial configuration. A directory will have been made: /home/opc/.oci

It will contain the following files:

- config
- oci_api_key.pem
- oci_api_key_public.pem

3. **Make a Copy of your oci_api_key_public.pem that you will use in your PCA User profile's API Keys:**
   - [opc@bastion ~]$ cat oci_api_key_public.pem
   - Cut and then Paste this key in the next step
4. **Navigate to your PCA User Profile to update your User API Keys**
   - Identity
   - Select Users
   - Select your User identity
   - Scroll down on left nav to "API Keys" and select
   - Select "Add API"
   - Cut and Paste the oci_api_key_public.pem
   - Select "Add"
   - **Copy the Configuration profile View presented** (It will be used in the next step)
   - **Note the API key fingerprint.** This should be the same fingerprint found in the oci config file: /home/opc/.oci/config



3. Configure the OCI CLI Files

In this section, you will be making changes to the /.oci/config file. This file provides the necessary tenancy and security keys for accessing your PCA System.

a. [opc@bastion ~]$ vi ~/.oci/config
   1. Change the [DEFAULT] to [pca]
   2. Cut and paste the complete section and place beneath it. This will be the section for the PCA Region

Example:

[pca]
user=ocid1.user.oc1..aaaaaaaaqlnen3dpgyqn6xi6hwl7vuertngidgncd7rbgt2x2ga2rmyawvya
fingerprint=[redacted]
key_file=/home/opc/.oci/oci_api_key.pem
tenancy=ocid1.tenancy.oc1..aaaaaaaalmxrgmsqoou26rdybgthdwtwdwhn7km5oq3u25f6bdpoxihqef6q
region= scasg03.us.oracle.com

Note: For your region, simply look at the browser URL, where the console will appear as in this example: https://console.scasg02.us.oracle.com. Your region will be scasg02.us.oracle.com

4. For the next step, you will need a Certificate Authority Bundle. The instructions are provided here:

   Reference Doc: https://docs.oracle.com/en/engineered-systems/private-cloud-appliance/3.0-latest/user/user-usr-ce-cli.html#usr-cli-obtain-cabundle

   a. A summary of the above instructions is provided here
   b. Type into your browser: https://iaas.<your_PCA_domain>/cachain

5. Create the pca.pem file
   a. Create a file under the /home/opc/.oci directory and name it “pca.pem”
   b. [opc@bastion /home/opc/.oci~]$ vi pca.pem
      1. Cut and Paste the Cert File from the previous step and save.
6. Create the oci_cli_rc file
   a. [opc@bastion /home/opc/.oci~]$ vi oci_cli_rc
   b. Cut and paste and update with your tenancy id :

```
[pca]
tenancy=<your tenancy ocid>
cert-bundle=/home/opc/.oci/pca.pem
```

7. Check the CLI access on the PCA
   a. Next, modify the bastion host’s firewall for CLI access to PCA
      1. [bastion /home/opc/.oci]$ sudo systemctl stop firewalld
      2. [bastion /home/opc/.oci]$ sudo rm -f /etc/firewalld/direct.xml
      3. [bastion /home/opc/.oci]$ sudo systemctl start firewalld

8. Confirm CLI access on PCA
   a. [bastion /home/opc/.oci]$ oci os ns get –-profile pca

```
{
  "data": "axpu7uokzjmj"
}
```

9. Set up a proxy on your bastion host before running the terraform script for systems that do not have direct internet access
   a. [bastion /home/opc/.oci]$ sudo vi /etc/environment

```
http_proxy=http://www-proxy.<your_domain>.com:80
https_proxy=http://www-proxy.<your_domain>.com:80
```

```
no_proxy=localhost,127.0.0.1,1,2,3,4,5,6,7,8,9,0,.<your_domain>.com,.ie.oracle.com,.oraclecorp.com
```

Note: *You will need to contact your system administrator for proxy and port settings for your environment.*

Terraform Script Execution – Part 1 (Provisioning IAM Resources)

All the Identity related resources, including Dynamic Groups, Policies and Tags, will be created on the PCA in this initial step.

Two Terraform scripts will be run on the bastion server created in previous steps. The first script will be to create the needed Dynamic Group Identity resources on your PCA. The second script will be to create infrastructure resources on PCA.

1. SSH into the Bastion Server created in previous steps
    a. [user@my_laptop ~]$ ssh opc@<bastion_Private_IP_Address>

2. Make a directory, “openshift”
    a. [opc@bastion /home/opc]$ mkdir openshift

3. Make a directory, “createResourceOnHomeRegion” in the openshift directory
    a. [opc@bastion /home/opc/openshift]$ mkdir createResourceOnHomeRegion

4. Copy the two files from the following github repo as files into the directory, createResourceOnHomeRegion.
    a. https://github.com/oracle-quickstart/oci-openshift/tree/c3-la/edge/c3_pca into the directory, **createResourceOnHomeRegion**.
    b. Replace the variables in **terraform.tfvars**
    c. Replace the default values of “compartment_ocid”, “cluster_name” and “home_region_profile_name” (found during OCI CLI Configuration)

```
tenancy_ocid             = "<replace with C3 or PCA tenancy_ocid>"
compartment_ocid         = "<replace with the compartment_ocid>"
home_region_profile_name = "<replace with the oci config profile name of the home region or PCA region>"
cluster_name = "<replace with the OpenShift cluster name>"
```

Note-1: For home_region_profile_name, this is also used in your /home/opc/.oci/config file. In the example here it is “home” in the first section of config under [home].

5. Before launching the TF scripts, determine if you have tenancy privileges to create Dynamic Groups for specific compartment. **If you do not, please see Step-9 where you will need to ask your PCA System Admin to make these manual updates (alternatively, provide you permissions)**

6. Ensure the environment source for your proxy servers
    a. [opc@bastion createResourceOnHomeRegion~]$ source /etc/environment

7. Ensure your C3 Certificate is exported
    a. [opc@bastion createResourceOnHomeRegion~]$ export custom_cert_location=/home/opc/.oci/pca.pem

8. Apply the Terraform Script:
    a. [opc@bastion createResourceOnHomeRegion~]$ terraform init
    b. [opc@bastion createResourceOnHomeRegion~]$ terraform plan
    c. [opc@bastion createResourceOnHomeRegion~]$ terraform apply

Note-1: If **"terraform init**" provides an **error** "Failed to query available provider packages", exit the bastion host and ssh back into the bastion instance.

Note-2: You should witness the following response after terraform apply: Apply complete! Resources: 5 added, 0 changed, 0 destroyed.

9. If you do not have administrative rights to the tenancy in the PCA to create Dynamic Groups and Policies. This provisioning will be done manually and at **the tenancy** level:
    a. Create a Dynamic Group: "<your-cluster-name>_compute_nodes"
        1. On the Left Nav, navigate to Identity and Security and choose Domains and the choose Default
        2. On the Left Nav, choose Dynamic Groups and Create a Dynamic Group with the name above.
        3. In the Matching Rules section, copy the following.
        4. Replace the ${var.compartment_ocid} , in Rule-1, with the target compartment ocid
        5. Replace the ${var.cluster_name} ,in Rule-2, with the target compartment cluster name

        - Rule 1:
            - instance.compartment.id= '${var.compartment_ocid}'
        - Rule-2:
            - tag.openshift- ${var.cluster_name}.instance-role.value='compute'

    b. **Create another Dynamic Group**: **"<your-cluster-name>_control_plane_nodes>"**
        1. In the Matching Rules section, copy the following.
        2. Replace the `${var.compartment_ocid}` in Rule-1 with the target compartment ocid
        3. Replace the `${var.cluster_name}` in Rule-2 with the target compartment cluster name

        - Rule 1:
            - instance.compartment.id= '${var.compartment_ocid}'
        - Rule-2:
            - tag.openshift- ${var.cluster_name}.instance-role.value='compute'

    c. **Create a Policy** for the control plane nodes in your compartment and name it **: "<your-cluster-name>_control_plane_nodes>"**
        1. In the Policy Builder, using manual editor, copy the following section below and
            1. Replace the `${oci_identity_dynamic_group.openshift_control_plane_nodes.name}` with the **control plane DG** name created in the previous step.
            2. Replace the `${oci_identity_dynamic_group.openshift_control_plane_nodes.name}` with the target compartment's **cluster name**

- Allow dynamic-group ${oci_identity_dynamic_group.openshift_control_plane_nodes.name} to manage volume-family in compartment id c
- Allow dynamic-group ${oci_identity_dynamic_group.openshift_control_plane_nodes.name} to manage instance-family in compartment id ${var.compartment_ocid}
- Allow dynamic-group ${oci_identity_dynamic_group.openshift_control_plane_nodes.name} to manage security-lists in compartment id ${var.compartment_ocid}
- Allow dynamic-group ${oci_identity_dynamic_group.openshift_control_plane_nodes.name} to use virtual-network-family in compartment id ${var.compartment_ocid}
- Allow dynamic-group ${oci_identity_dynamic_group.openshift_control_plane_nodes.name} to manage load-balancers in compartment id ${var.compartment_ocid}

d. **Comment out lines 79-107 in the createIdentityResources.tf script**
   1. This will include:
      1. resource "oci_identity_dynamic_group" "openshift_control_plane_nodes"
      2. resource "oci_identity_policy" "openshift_control_plane_nodes"
      3. resource "oci_identity_dynamic_group" "openshift_compute_nodes"

OpenShift Image Preparation

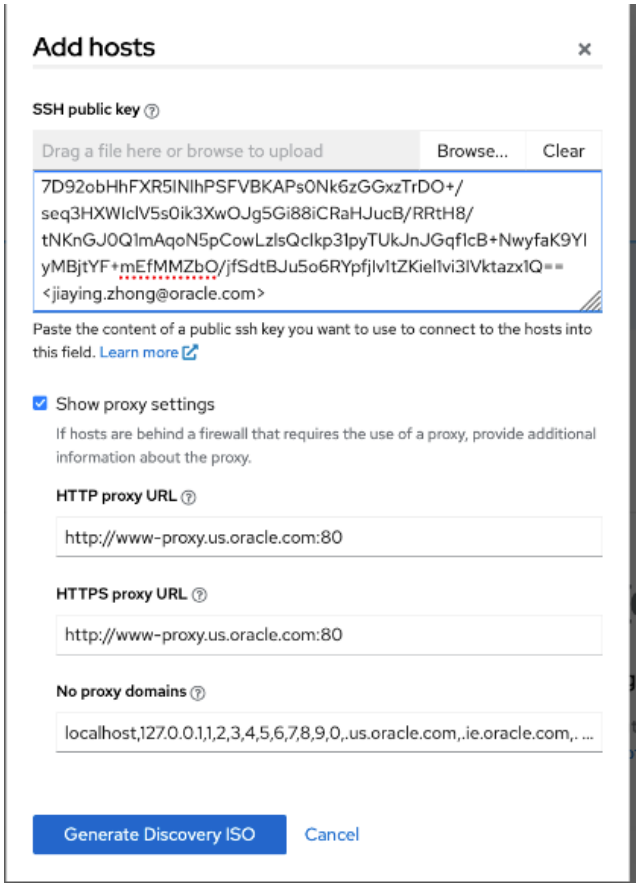
In this section we will generate the OpenShift ISO image on the Redhat Portal, convert it to an OCI compatible image and import it to you C3. This will be done ahead of the second terraform script being executed on your C3 platform.

This can be completed on your laptop and not on the bastion or within environments such as Oracle Solution Center as the resulting image will be uploaded to your C3 platform's custom images in the instructions that follow.

1. Navigate to the Redhat assisted-installer cluster portal: https://console.redhat.com/openshift/assisted-installer/clusters
2. **Select** Create New Cluster
3. **On the next page, enter the following:**
   a. **Cluster name:** The name of your OpenShift cluster. It should be the same as what was specified when creating the resource via Terraform scripts. The cluster_name value must be 1-54 characters. It can use lowercase alphanumeric characters or hyphen (-), but must start and end with a lowercase letter or a number.
   b. **Base domain**: This is the value used for the zone_dns variables in Terraform scripts which runs on C3. We will use it in the next section.
   c. **OpenShift version**: "Openshift 4.16.20"
      i. **You must use this compatible version.**
      ii. If you cannot find it directly, go to the bottom of the drop-down menu. Select "show all available versions" and type the version Openshift 4.16.20, then select it. Please use this specific version.
   d. **"Integrate with external partner platforms**": Select "Oracle Cloud Infrastructure"
   e. Leave all other values as their defaults. Select "Next"

4. **Under Operators**, do not select any check boxes. Select "Next"
5. **For Host Discovery**, select Add Hosts.
    a. Add your ssh public key in the window that opens.
    b. Click the "Show proxy settings" check box, and adding the proxy variables configured in previous steps and is **found in the bastion's /etc/environment** file configured earlier:

```
http_proxy=http://www-proxy.<your_domain>.com:80
https_proxy=http://www-proxy.<your_domain>.com:80
no_proxy=localhost,127.0.0.1,1,2,3,4,5,6,7,8,9,0,.<your_domain>.com
#(ie.oracle.com,.oraclecorp.com)
```



    c. Select Generate Discovery ISO
    d. Click "Download Discovery ISO". After you have the iso file, you can rename it as you want, for example "discovery_image_<your_cluster_name>.iso".

6. Convert ISO to OCI image, which we will import it to your PCA system to an OCI Home Region Object Store or equivalent to be able to import Custom Images from an Object Storage Bucket, or alternatively, to use the OCI CLI.

    Reference: https://www.oracle.com/docs/tech/oracle-private-cloud-appliance-x9-2-workload-import.pdf

a. Move the downloaded iso image from your /Downloads directory to a new directory to make the following changes
    a. [user@mylaptop ~]$ mkdir image_prep
    b. [user@mylaptop ~]$ mv /Downloads/discovery_image_<your_cluster_name>.iso ~./image_prep
b. Convert the iso to qcow2
    a. [user@mylaptop ~./imageprep]$ brew install qemu
    b. [user@mylaptop ~./imageprep]$ qemu-img convert -O qcow2 discovery_image_<your_cluster_name>.iso output.QCOW2
c. Create image_metadata.json as follows (setting FW as UEFI and launch mode as paravitualized)
    a. [user@mylaptop ~./imageprep]$ vi image_metadata.json
    b. Cut and paste the content below
    c. Save and quit

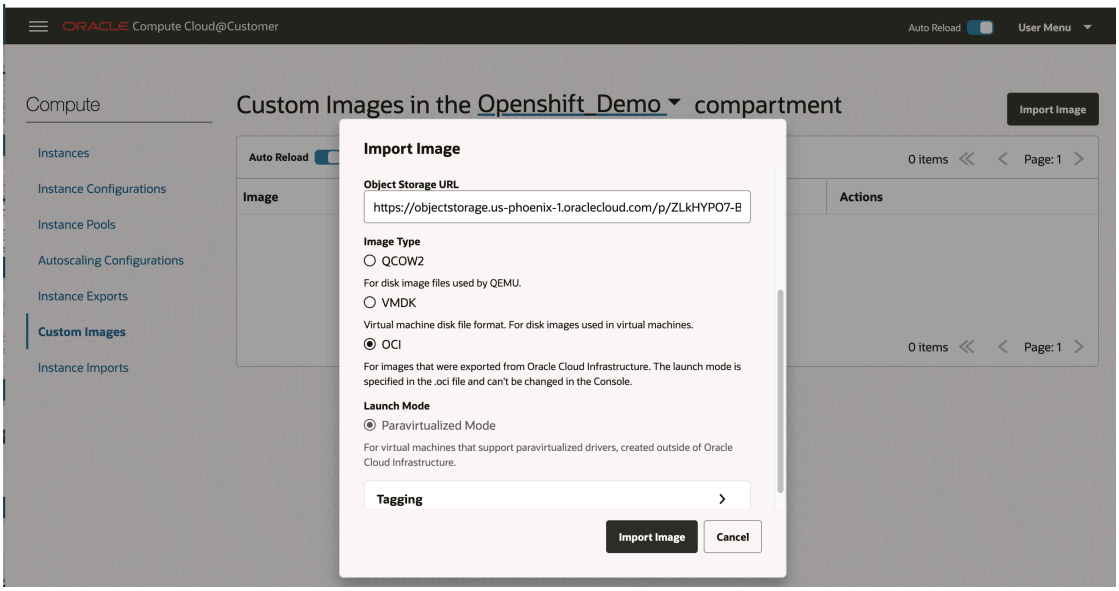
```
{
"version": 2,
"externalLaunchOptions": {
"firmware": "UEFI_64",
"networkType": "PARAVIRTUALIZED",
"bootVolumeType": "PARAVIRTUALIZED",
"remoteDataVolumeType": "PARAVIRTUALIZED",
"localDataVolumeType": "PARAVIRTUALIZED",
"launchOptionsSource": "PARAVIRTUALIZED",
"pvAttachmentVersion": 1,
"pvEncryptionInTransitEnabled": false,
"consistentVolumeNamingEnabled": false
},
"imageCapabilityData": null,
"imageCapsFormatVersion": null,
"operatingSystem": "Custom",
"operatingSystemVersion": "Custom"
}
```

d. Use the following tar command to combine the metadata image and qcow2 to an .oci file
e. [user@mylaptop ~./imageprep]$ tar cf discovery_image_<your_cluster_name>.oci image_metadata.json output.QCOW2

7. Upload the "discovery_image_<your_cluster_name>.oci" to an OCI bucket, and generate a Pre-Authenticated Request (PAR) URL
    a. Log into an OCI Region's Tenancy to access Object Storage (or use the document provided above to use OCI CLI.)
    b. Navigate to Object Storage & Archive Storage -> Buckets. Select your compartment. Create a bucket.
    c. Upload the "discovery_image_<your_cluster_name>.oci" to OCI bucket.

d. Select the hotdog (dot) menu next to it and **Create Pre-Authenticated Request** for the object, allowing object reads and extending the expiration date to your needs. Copy the PAR URL and keep it. We will use it in the next step.

8. Import the OCI image onto your PCA portal.
    a. Log into your to PCA's Console
    b. Click Compute → Click custom images (from the pull down) → Select your compartment
    c. Click import image
        1. → give it a name
        2. → Source Type choose "Import from an Object Storage URL"
        3. → Paste the PAR URL generated in previous step in the "Object Storage URL"
        4. → Image type select "OCI"
        5. → Launch Mode select "Paravirtualized Mode"
        6. → click import image
    d. After image states become available, **copy the ocid of the image and keep it**. We will use it as input variable **"image_id_manually_created_on_C3"** when running the tf scripts via C3 regions in the next step



Terraform Script Execution – Part 2 ("Creating PCA Resources & Cluster")

In this section, we will create all the infrastructure resources (LB's, VCN's, Instances, etc) on the PCA System.

1. SSH into the Bastion image from your laptop
    a. [my_laptop ~]$ ssh opc@<bastion_Private_IP_Address>
2. **Set your C3 Certificate location – For Authorizations**
    a. [opc@bastion ~] **export custom_cert_location=/home/opc/.oci/pca.pem**
    b. Check that the variable is set properly
    c. [opc@bastion ~] **`echo $custom_cert_location`**

Note: During the execution of the next TF script, and see an x509 Cert issue, use the command in 2a above to provide Certificate location

3. Make a directory **"createResourceOnC3"** and copy following two files into the directory https://github.com/oracle-quickstart/oci-openshift/tree/c3-la/edge/c3_pca
    a. Two files to be duplicated:
        1. createInfraResources.tf
        2. terraform.tfvars
    b. [Bastion /home/opc/openshift]$ mkdir createResourceOnC3
    c. Update all the variables in the C3 Region's **terraform.tfvars** file

```
c3_or_pca_region              = "<replace with C3 region or PCA region>"
compartment_ocid              = "<replace with the compartment_ocid>"
c3_or_pca_region_profile_name = "<replace with the oci config profile name of the C3
region or PCA region>"
cluster_name                    = "<replace with the OpenShift cluster name>"
zone_dns                        = "<replace with the name of cluster's DNS zone>"
image_id_manually_created_on_C3 = "<replace with the image ocid which manually created
on C3>"
control_plane_shape             = "<replace with the Compute shape of the control_plane
nodes>"
compute_shape                   = "<replace with the Compute shape of the compute
nodes>"
create_openshift_instance_pools = true
```

Note1: The C3 or PCA region is the FQDN of the console you logged into. Following are two examples:
(1) C3 Lab Console: console.scasg03.us.oracle.com. Region = scasg03.us.oracle.com
(2) PCA Lab Console: console.scasg02.us.oracle.com Region = scasg02.us.oracle.com

Note2: Compute Shapes
(1) For C3, insert and use (for both Control Plane & Compute): VM.PCAStandard.E5.Flex
(2) For PCA, insert and use (for both Control Plane & Compute): VM.PCAStandard1.Flex

4. Ensure the environment source for your proxy servers
    a. [opc@bastion createResourceOnHomeRegion~]$ source /etc/environment
5. Ensure your C3 Certificate is exported
    a. [opc@bastion createResourceOnHomeRegion~]$ export custom_cert_location=/home/opc/.oci/c3.pem
6. Apply the Terraform Script

```
[Bastion createResourceOnPCA]$ terraform init
[Bastion createResourceOnPCA]$ terraform plan
[Bastion createResourceOnPCA]$ terraform apply
```

5. **Record of the Terraform output**, as it will be used it in later steps. Below is a generic example:

```
useInstancePrincipals: true
compartment: <compartment-ocid>
vcn: <vcn-ocid>
```

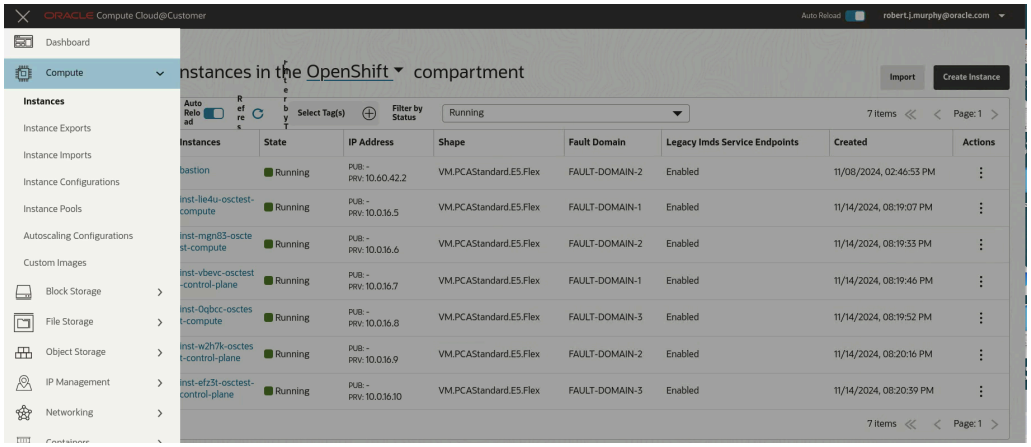
```
loadBalancer:
  subnet1: <subnet-ocid>
  securityListManagementMode: Frontend
  securityLists:
    <subnet-ocid>: <security-ocid>
rateLimiter:
  rateLimitQPSRead: 20.0
  rateLimitBucketRead: 5
  rateLimitQPSWrite: 20.0
  rateLimitBucketWrite: 5
```

6. **3 control plane (CP) nodes and 3 compute worker nodes** : Now running

   a. After you finished above steps, you will have 3 control plane (CP) nodes and 3 compute worker nodes that are up and running.

   b. You can validate this from the C3 console. Click "Compute" and switch to your compartment.

   c. You can distinguish the CP nodes from compute nodes by the suffix in the display name.



7. **Update the defined tags of all the CP and worker nodes:**

   a. Updating the instance tags will need to be done manually by updating the instance tags one by one using OCI CLI commands.

   b. In the bastion instance, run the following OCI CLI command to list all the running instances in your compartment:

      1. [opc@bastion ~]$ oci compute instance list --lifecycle-state RUNNING --compartment-id <your_compartment_id> --profile <your_pca_profile_name>

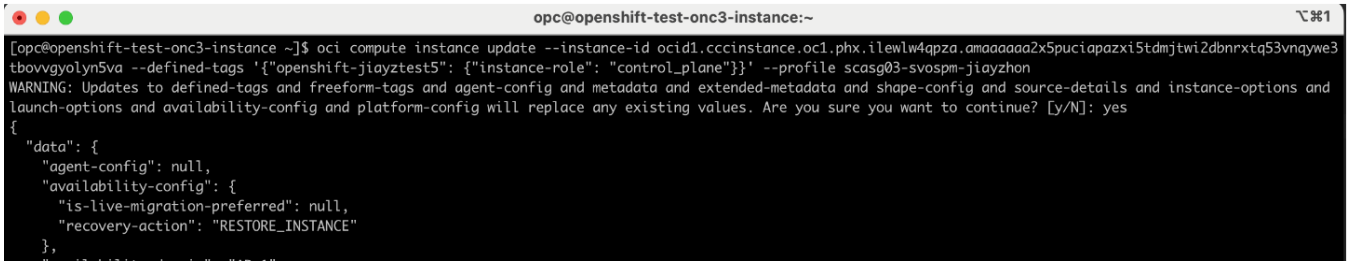
   Note: <your_C3_profile_name> is found in the config file on your bastion instance: [opc@bastion]$ /home/opc/.oci/config. In this example, it is “pca”.

   b. From the output, capture the ocids of the CP nodes and worker nodes.

   c. Alternatively, obtain each CP and Worker OCIDS from the PCA Console. With all the instances shown, as in Step 6 above, click on the respective instance hotdog menu under Actions and Copy the Instance OCID.

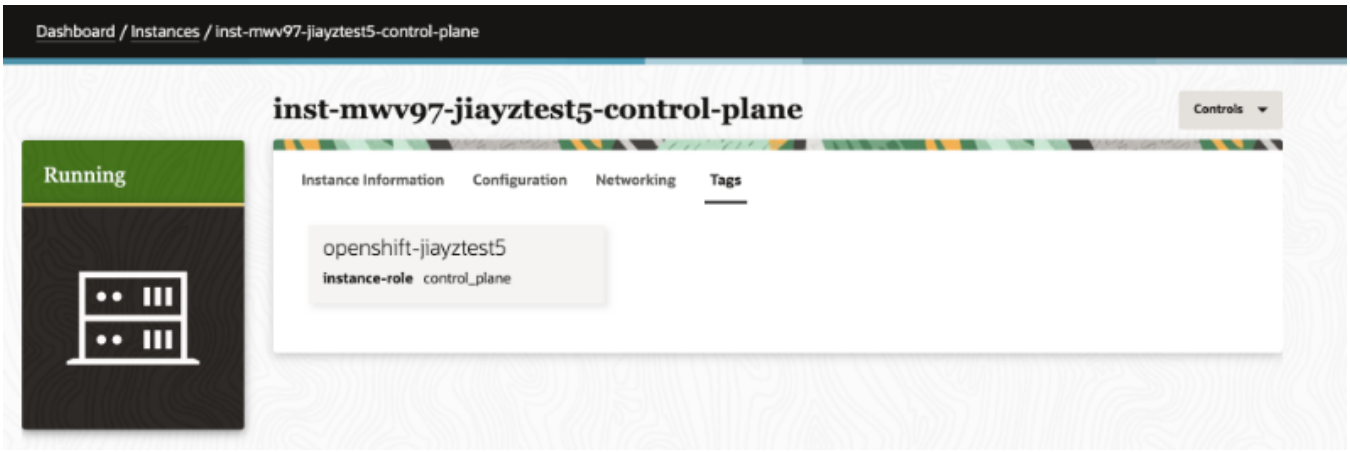
   d. Run following OCI CLI commands one by one to update the defined tags.

Note: The UPDATE tag operation will override the defined tag, (i.e., it will NOT Append the new tags at the end. If you don't want to lose the previous "defined tags", you need to manually capture them and append the Openshift tag at the end manually.



8. **Updating the Control Plane nodes -**
    a. [opc@bastion ~]$ oci compute instance update --instance-id <CP_instance_ocid_1> --defined-tags '{"openshift-<your_cluster_name>": {"instance-role": "control_plane"}}' --profile <your_pca_profile_name>
    b. [opc@bastion ~]$ oci compute instance update --instance-id <CP_instance_ocid_2> --defined-tags '{"openshift-<your_cluster_name>": {"instance-role": "control_plane"}}' --profile <your_pca_profile_name
    c. [opc@bastion ~]$ oci compute instance update --instance-id <CP_instance_ocid_3> --defined-tags '{"openshift-<your_cluster_name>": {"instance-role": "control_plane"}}' --profile <your_pca_profile_name>

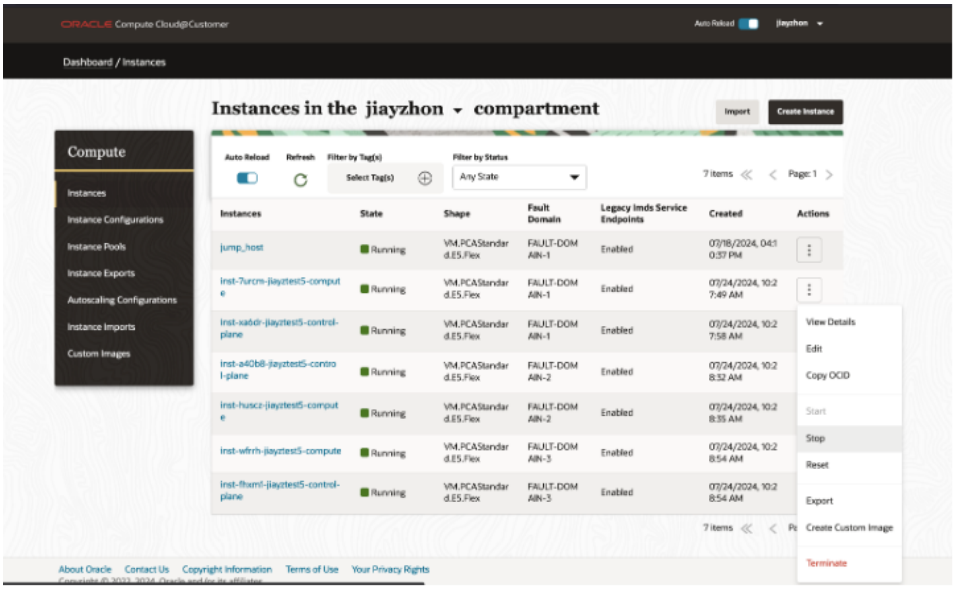
9. **Updating the Worker nodes -**
    a. [opc@bastion ~]$ oci compute instance update --instance-id <compute_worker_instance_ocid_1> --defined-tags '{"openshift-<your_cluster_name>": {"instance-role": "compute"}}' --profile <your_pca_profile_name>
    b. [opc@bastion ~]$ oci oci compute instance update --instance-id <compute_worker_instance_ocid_2> --defined-tags '{"openshift-<your_cluster_name>": {"instance-role": "compute"}}' --profile <your_pca_profile_name>
    c. [opc@bastion ~]$ oci compute instance update --instance-id <compute_worker_instance_ocid_3> --defined-tags '{"openshift-<your_cluster_name>": {"instance-role": "compute"}}' --profile <your_pca_profile_name>
    d. After updating the tags, you can verify your changes on the PCA console
        i. Navigate to one of the instances, and click the "Tags"
        ii. It should have the format of: openshift-<your_cluster_name>



e. Next, manually "stop" and "start" the instances one by one on the PCA portal. (Batch operations is not currently supported on PCA)

   i. Stop the CP and worker instance.

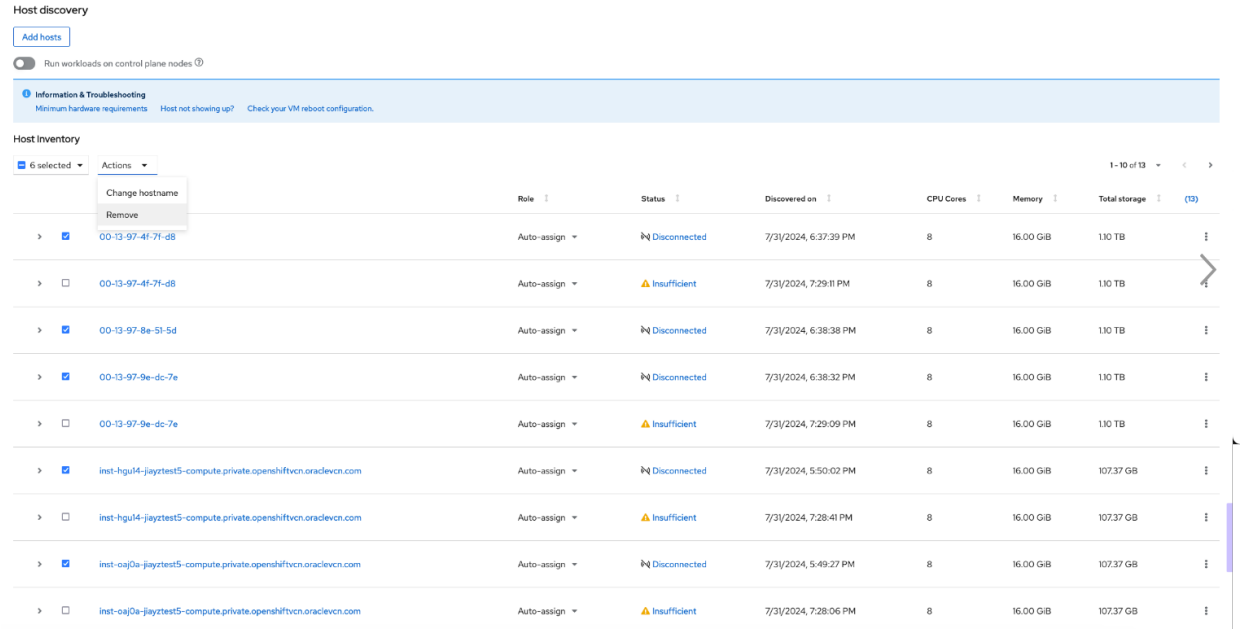
   ii. After each instance has "Stopped", click on Start.



Install the Cluster using the RH Assisted Installer UI

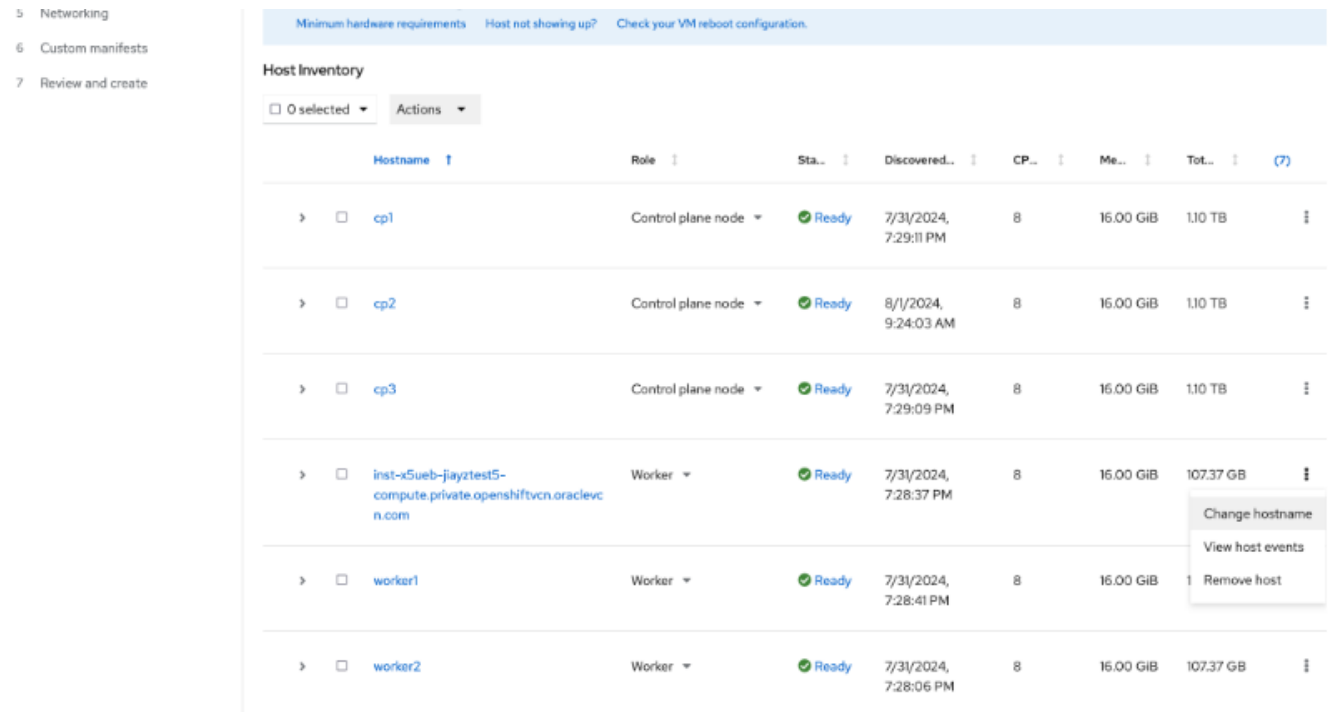
Installation of the OpenShift Cluster can now start, where all infrastructure has been configured, and instances are running and ready to be registered with Redhat.

1. Go back to the Redhat assisted-installer cluster portal: https://console.redhat.com/openshift/assisted-installer/clusters
2. Go to the "Host Discovery" section. If the Terraform scripts have been successful a list of hosts will appear on this page.
3. There should be 12 hosts where 6 of them are disconnected. The cause for this is that we had "restart" the instances on PCA in the above procedures. **These disconnected hosts can be removed**



4. Following the removal of "disconnected" nodes, all the other nodes indicating "insufficient" status will become "ready" in 2-3 minutes.
5. Assign the 3 nodes with a boot size of 1.1TB as the "Control Plane" Role, and 3 nodes with boot size 107GB as "Worker" Roles. Rename the hosts with a name shorter than 63 characters, otherwise the cluster installation will fail.

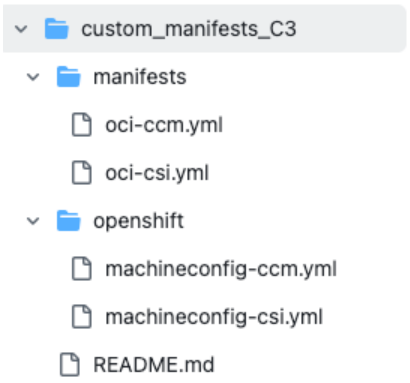
    1. **After completing this, select "Next".**



6. **Storage :** No updates will be required for Storage. Select "Next" to Continue
7. **Networking:** you will see "Some validations failed" for each of the hosts. Click the "**Some validations failed**" and click the "Add NTP sources", then **add the IP Address of "169.254.169.254"** for one of the nodes. Then **wait for 2-3 minutes and all the "Some validations failed" indicators shall disappear**. Then continue to **select "Next"**
8. **Custom Manifests:** refer to the manifests files in https://github.com/oracle-quickstart/oci-openshift/tree/c3-la/edge/c3_pca
9.

We will be uploading 4 manifests. Use the dropdown to select the appropriate matching title (openshift manifests) and **be sure that the file names match**. Two files from OCI should be manifest file type, and the other two from OpenShift should be marked as OpenShift file type.

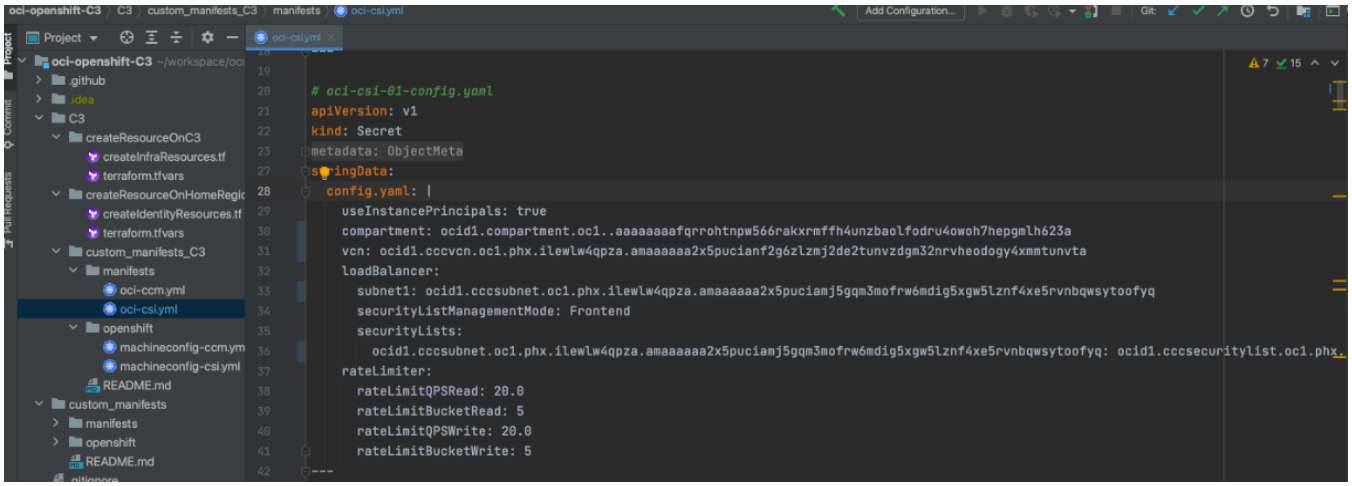
Create 4 files to in preparation to be uploaded. Copy the respective contents from the github link above. The **machineconfig files DO NOT need to be edited.**

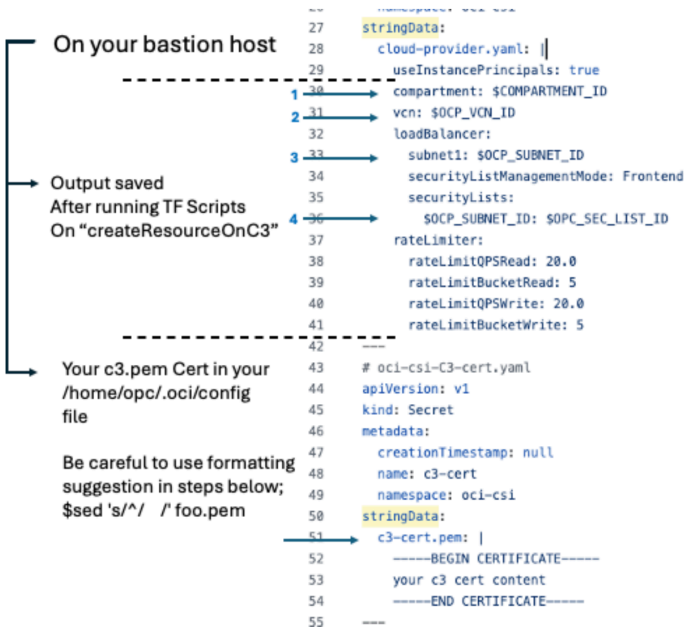


a. C3/custom_manifests_C3/openshift/
   i. machineconfig-ccm.yml
   ii. machineconfig-csi.yml

b. C3/custom_manifests_C3/manifests/
   i. oci-csi.yml
   ii. oci-ccm.yml

c. **In the oci-csi.yml file:** (found under section b above) , replace the content of **oci-csi-01-config.yaml** section (Line 29-41) with the saved output we obtained from a previous step of “Run Script via C3 region” section above.





d. **In the oci-csi.yml file:** (found under section b above), replace the content of **oci-csi-C3-cert.yaml** section (Lines 52-54) with the content of your PCA Certificate used in previous steps, referred to as " pca.pem " and can be found under [opc@bastion ~] /.oci/pca.pem

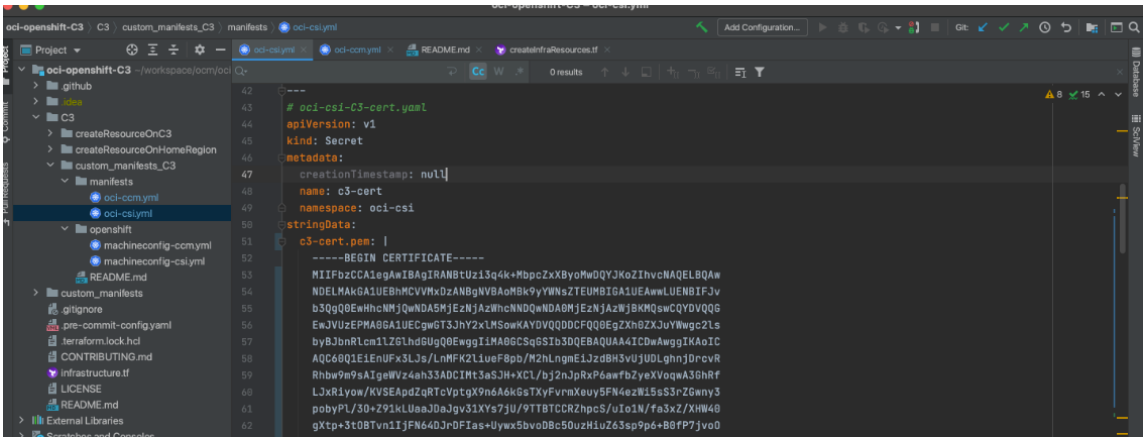
Important Note: The formatting of this pem within the two yml files are sensitive to justification.

- **To replicate the formatting required for the oci-ccm.yml & oci-csi.yml, use the following command on your pem file:**

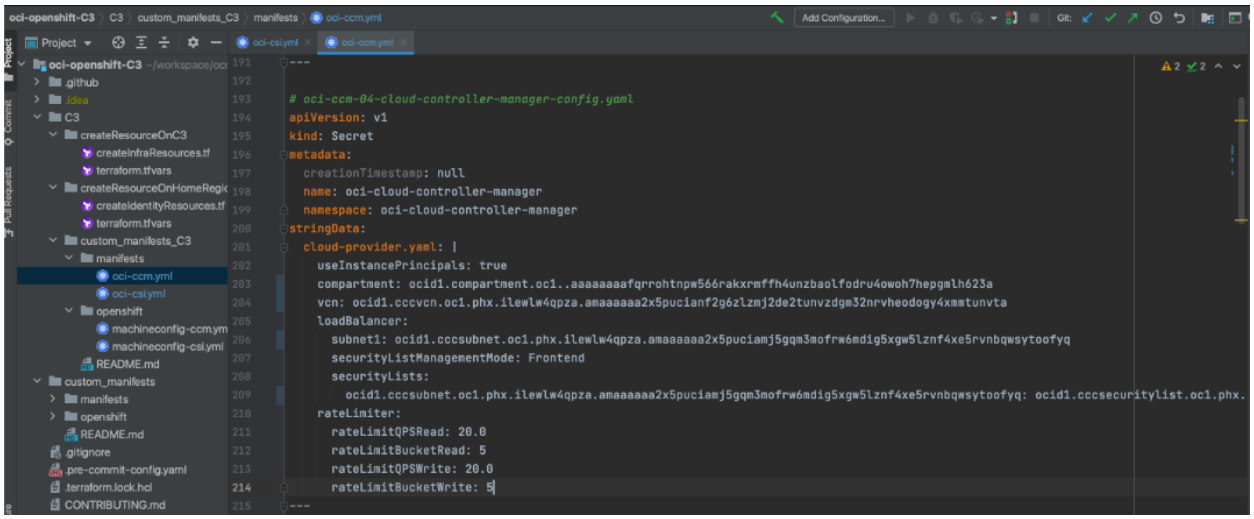
  [opc@test-ssh .oci]$sed 's/^/ /' pca.pem

  where “pca.pem” is the name of your pem file. the spaces between the two forward slashes is equal to 4 spaces. this is equal to the justification in the two files

The above command will create an output to cut and paste to place in both the oci-ccm.yml & oci-csi.yml files.



e. **In the oci-ccm.yml:** replace the content of **oci-ccm-04-cloud-controller-manager-config.yaml** section (Lines 202-214) with the **saved Output** we obtained in the previous step.



f. **In the oci-ccm.yml,** replace the contents of oci-ccm-C3-cert.yaml (Line 225-227) with the content of your PCA Certificate used in previous steps, referred to as " pca.pem ".

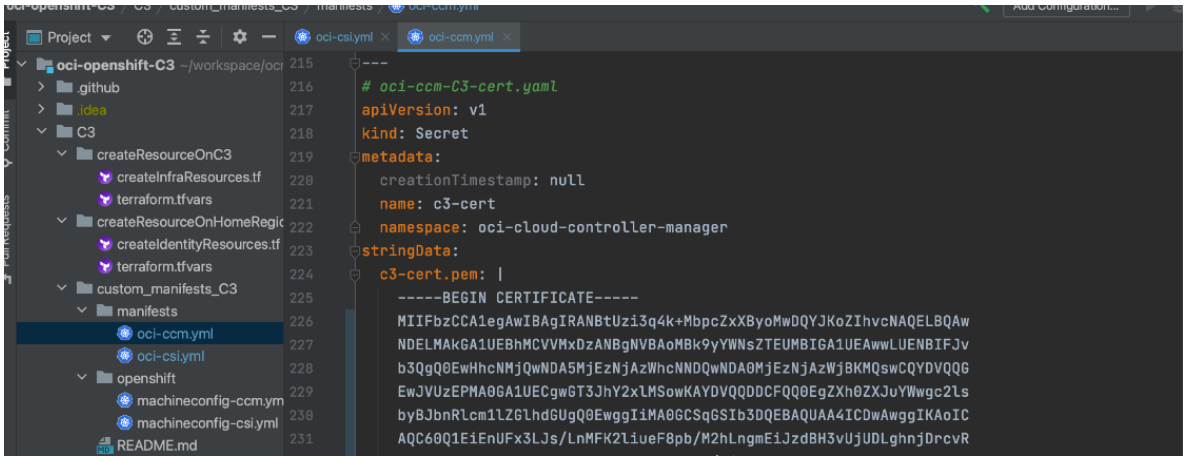
Important Note: The formatting of this pem within the two yml files are sensitive to justification.

- **To replicate the formatting required for the oci-ccm.yml & oci-csi.yml, use the following command on your pem file:**

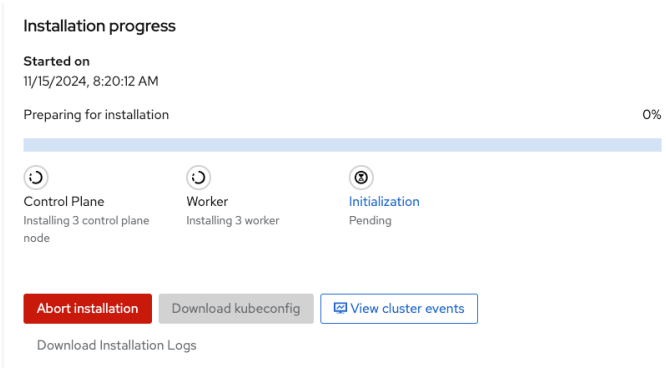
  [opc@test-ssh .oci]$sed 's/^/ /' pca.pem

  where “pca.pem” is the name of your pem file. the spaces between the two forward slashes is equal to 4 spaces. this is equal to the justification in the two files

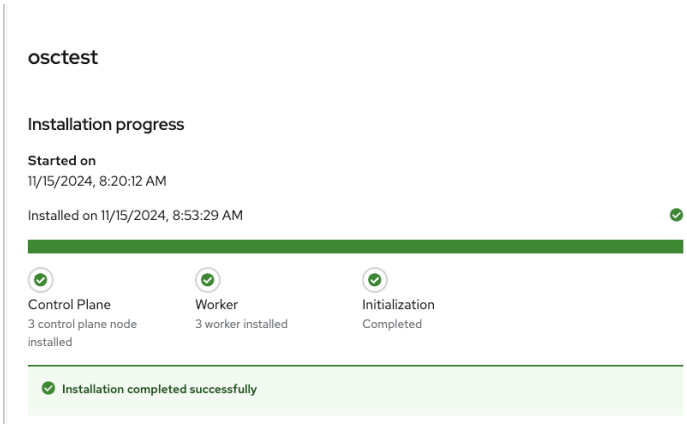
The above command will create an output to cut and paste to place in both the oci-ccm.yml & oci-csi.yml files.



9. Select "Next".
10. Page: Review and Create.
11. Select "Install cluster".
12. Installation Progress will begin
    1. It will take approximately 30m to complete



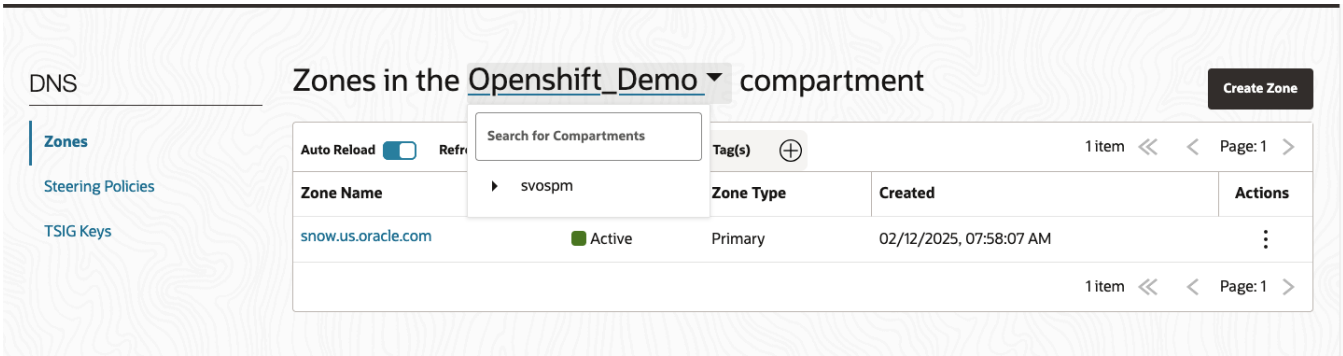
13. Installation Successful



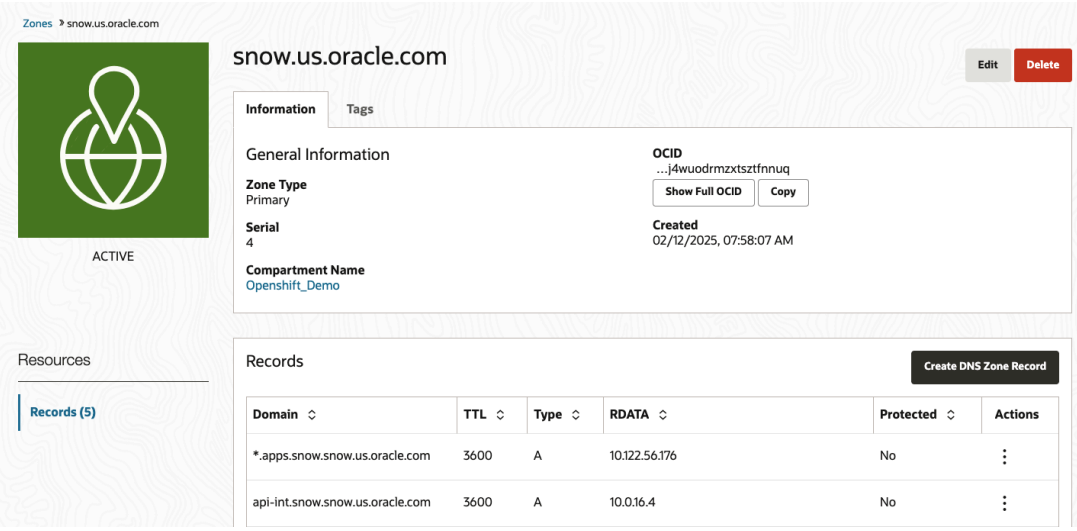
14. This completes the installation

15. To access the OpenShift Console

1. On the Left Hand Navigation, scroll down to DNS, and choose Zones
2. Make sure you're in the Compartment of your installed cluster



3. Click on the DNS Zone Name (in this example: snow.us.oracle.com)
4. Record the IP Address of the first entry, "*apps.snow.snow.us.oracle.com". This IP Address will be used to update your /etc/hosts file



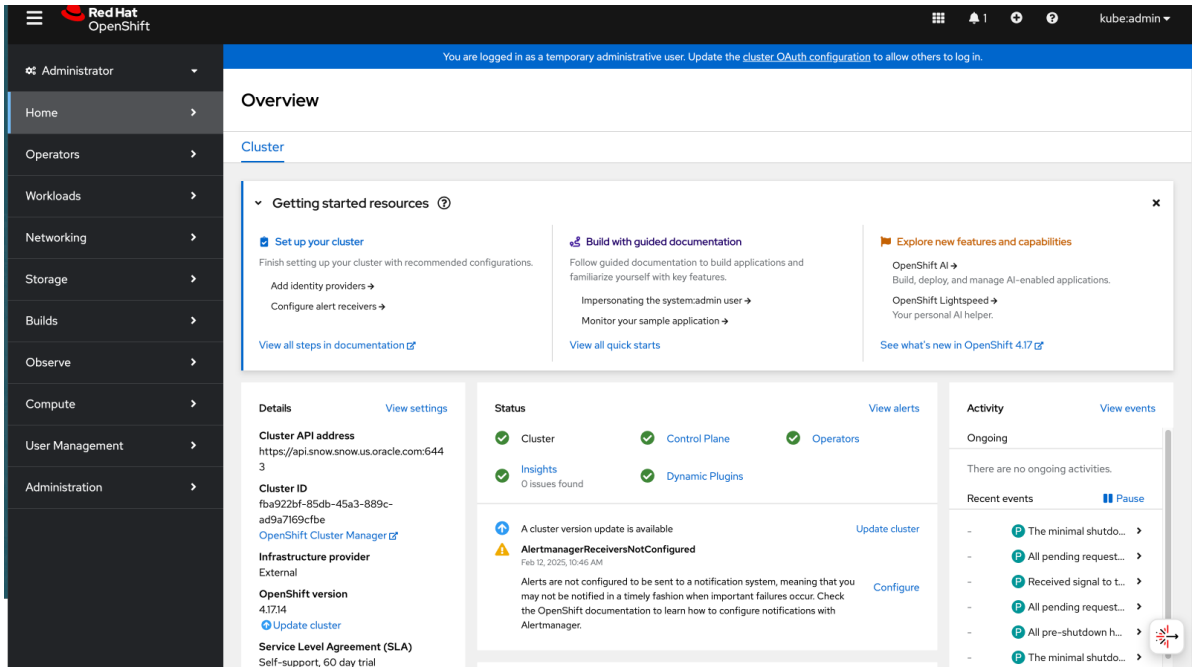
5. Go to your laptop's terminal, navigate and edit /etc/hosts
6. Insert a new line with the IP Address obtained in the previous step
7. After the IP Address, append your specific DNS on to the end
8. The new line should look like this:

10.122.56.176 console-openshift-console.<your dns entry here > oauth-openshift.<your dns entry here>

Using the above example, it would look like this:

10.122.56.176 console-openshift-console.apps.snow.snow.us.oracle.com oauth-openshift.apps.snow.snow.us.oracle.com

9. **You now have access to the OpenShift Portal:**



10. This completes the installation.

Connect with us

Call +**1.800.ORACLE1** or visit **oracle.com**. Outside North America, find your local office at: **oracle.com/contact**.

blogs.oracle.com facebook.com/oracle twitter.com/oracle

Author: Jiayang Zhong and Robert Murphy