**Deployment Document**

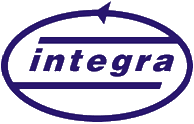
**Project**

**OCP 4.16 NON-PROD Cluster**

**v1.1**

**Client: Tata Consultancy Services**





**Integra Micro Systems Pvt. Ltd.**

No. 4, 1st Floor, Bellary Road,

12th KM, Jakkur,

Bangalore 560 064, INDIA

Tel: 080-46632400

[**www.integramicro.com**](http://www.integramicro.com/)

**Disclaimer**

Copyright © 2013:  Integra Micro Systems Private Limited, Registered Office #4, Bellary Rd, 12, KM, Yashoda Nagar, Jakkur, Bangalore, Karnataka India.

All rights about this document are reserved and  shall not be , in whole or in part, copied, photocopied, reproduced, translated, or reduced to any manner including but not limited to electronic, mechanical, machine readable ,photographic, optic recording or otherwise without prior consent, in writing, of Integra Micro Systems Private Limited (the Company).

The information in this document is subject to changes without notice. This describes only the product defined in the introduction of this documentation. This document is intended for the use of prospective customers of the Company Products Solutions and or Services for the sole purpose of the transaction for which the document is submitted. No part of it may be reproduced or transmitted in any form or manner whatsoever without the prior written permission of the company. The Customer, who/which assumes full responsibility for using the document appropriately. The Company welcomes customer comments as part of the process of continuous development and improvement.

The Company, has made all reasonable efforts to ensure that the information contained in the document are adequate, sufficient and free of material errors and omissions. The Company will, if necessary, explain issues, which may not be covered by the document. However, the Company does not assume any liability of whatsoever nature, for any errors in the document except the responsibility to provide correct information when any such error is brought to company’s knowledge. The Company will not be responsible, in any event, for errors in this document or for any damages, incidental or consequential, including monetary losses that might arise from the use of this document or of the information contained in it.

This document and the Products, Solutions and Services it describes are intellectual property of the Company and/or of the respective owners thereof, whether such IPR is registered, pending for registration, applied for registration or not.

The only warranties for the Company Products, Solutions and Services are set forth in the express warranty statements accompanying its products and services. Nothing here in should be construed as constituting an additional warranty. The Company shall not be liable for technical or editorial errors or omissions contained herein.

The Company logo is a trademark of the Company. Other products, names, logos mentioned in this document, if any, may be trademarks of their respective owners.

Copyright © 2013:  Integra Micro Systems Private Limited. All rights reserved.

**Release History**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1. Release Version | 1. Release Date | 1. Description of the release | 1. Created By | 1. Approved By |
| 0.1 | 1. 09-02-2025 | 1. First Draft | 1. Sai Akhil/Kiran/Sanket | 1. Murali.K.Muddada |
| 0.2 | 13-02-2025 | Updated ELK details | 1. Sai Akhil/Kiran/Sanket | 1. Murali.K.Muddada |
| 0.3 | 17-02-2025 | Updated prometheous and Graphana details | 1. Sai Akhil/Kiran/Sanket | 1. Murali.K.Muddada |
| 0.4 | 18-02-202 | Added SMTP details | 1. Sai Akhil/Kiran/Sanket | 1. Murali.K.Muddada |

**Confidentiality Statement**

Copyright © 2011-2012, Integra Micro Systems Private Limited. All rights reserved. This product or document may not, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine readable form, by any means electronic, mechanical, photographic, optic recording or otherwise without prior consent, in writing, of the copyright owner. Statutory declaration under section 52A of the Copyright Act 1957.

**Table of Contents**

[1 About the Guide 6](#_Toc190343938)

[1.1 Target Audience 6](#_Toc190343939)

[1.2 Acronyms and Abbreviations 6](#_Toc190343940)

[2 Introduction 8](#_Toc190343941)

[2.1 Topics Covered 8](#_Toc190343942)

[2.2 Scope 8](#_Toc190343943)

[2.3 Out of Scope 8](#_Toc190343944)

[3 Implementation Details 9](#_Toc190343945)

[3.1 Environment Setup 9](#_Toc190343946)

[3.1.1 Network Information 9](#_Toc190343947)

[3.1.2 Network Services 9](#_Toc190343948)

[3.1.3 Node Information 9](#_Toc190343949)

[3.1.4 Storage Information 9](#_Toc190343950)

[3.1.5 Certificates 9](#_Toc190343951)

[4 Technical Implementation 10](#_Toc190343952)

[4.1 Check Bastion Node for all the pre-Installation requirements 10](#_Toc190343953)

[4.1.1 Operating System Details 10](#_Toc190343954)

[4.1.2 Block Device (Hard Disk) Details 10](#_Toc190343955)

[4.1.3 Network Configuration 10](#_Toc190343956)

[4.1.4 Route and Default Gateway details 10](#_Toc190343957)

[4.2 Installing a Connected Cluster 10](#_Toc190343958)

[4.2.1 Configure Environmental Services in Bastion. 10](#_Toc190343959)

[4.2.2 Generating an SSH private key and adding it to the agent 11](#_Toc190343960)

[4.2.3 Create install-config.yaml file 11](#_Toc190343961)

[4.2.4 Creating the Openshift/Kubernetes manifest 12](#_Toc190343962)

[4.2.5 Create the Ignition config files 12](#_Toc190343963)

[4.3 Deploying Openshift 13](#_Toc190343964)

[4.3.1 Monitoring the Bootstrap Process 13](#_Toc190343965)

[4.3.2 Remove the Bootstrap Node 13](#_Toc190343966)

[4.3.3 Join Nodes 14](#_Toc190343967)

[4.4 Accessing the Cluster 14](#_Toc190343968)

[4.4.1 Accessing the Openshift Console 14](#_Toc190343969)

[4.4.2 Accessing the Cluster Nodes. 15](#_Toc190343970)

[4.5 Trident CSI driver Installation and configuration 16](#_Toc190343971)

[4.5.1 Trident operator Installation 16](#_Toc190343972)

[4.5.2 Trident RHCOS changes 19](#_Toc190343973)

[4.5.3 Trident Backend Config 19](#_Toc190343974)

[4.5.4 Trident Storage class Config 21](#_Toc190343975)

[4.5.5 Identify backend for storage class 22](#_Toc190343976)

[4.6 Deploying Openshift Internal Registry with Persistent Storage 23](#_Toc190343977)

[4.7 Configure System Clock Sync with NTP Server 24](#_Toc190343978)

[4.7.1 Configure master nodes system clock sync with NTP server 24](#_Toc190343979)

[4.8 Configuring an HTPasswd identity provider 25](#_Toc190343980)

[4.9 Registering the Cluster 26](#_Toc190343981)

[4.10 Encrypting the ETCD data 27](#_Toc190343982)

[4.11 Backing Up ETCD data 28](#_Toc190343983)

[4.12 Deploying and Configuring Openshift Logging 28](#_Toc190343984)

[4.12.1 Install Cluster Logging Operators 28](#_Toc190343985)

[4.12.2 Create Cluster Logging Instance 29](#_Toc190343986)

[4.12.3 Viewing cluster logs from the Kibana dashboard 30](#_Toc190343987)

[4.13 Applying a custom Alertmanager configuration with the SMTP details 31](#_Toc190343988)

[4.14 Configuring Openshift monitoring components 33](#_Toc190343989)

[4.15 Openshift Virtualization 34](#_Toc190343990)

[4.15.1 Installing the OpenShift Virtualization Operator by using the web console 34](#_Toc190343991)

[4.15.2 Adding a Bootable Volume 36](#_Toc190343992)

[4.15.3 Clone a VirtualMachine Template 37](#_Toc190343993)

[4.15.4 Create a VirtualMachine Using the Custom Template 39](#_Toc190343994)

# About the Guide

This design document currently includes a Prod environment which will be hosted on Virtual Machines.

## Target Audience

This document is intended for Client technical staff responsible for the environment.

## Acronyms and Abbreviations

The table below provides a glossary of the terms and acronyms used within this document.

|  |  |
| --- | --- |
| **Acronym** | **Description** |
| RH | Red Hat, Inc |
| OKE | Openshift Kubernetes Engine |
| OCP | Openshift Container Platform |
| AD | Active Directory |
| API | Application Programming Interface |
| AZ | Availability Zone |
| CA | Certificate Authority |
| CR | Custom Resource |
| DC | Data Centre |
| DNS | Domain Name System |
| DHCP | Dynamic Host Configuration Protocol |
| DVR | Distributed Virtual Routers, a networking configuration for routing using Compute and Network nodes for separate networking functions |
| FQDN | Fully Qualified Domain Name |
| Guest | Also see “VM”. This is virtual machine running on a Host. |
| HA | High-Availability or Highly-Available |
| Host | The physical hardware or the logical OS which runs virtualisation technology allowing one or more Guest OS’s to run on the hardware owned by the Host |
| KVM | Kernel-based Virtual Machine |
| L2 | Layer 2, part of the TCP/IP Network Stack |
| L3 | Layer 3, part of the TCP/IP Network Stack |
| LBaaS | Load Balancing as a Service |
| NAT | Network Address Translation |
| NIC | Network Interface Card. References a virtual or a physical port allowing network access and interface to a Host or Guest VM. |
| NTP | Network Time Protocol |
| OCP | OpenShift Container Platform |
| OS | Operating System |
| OVS | Open vSwitch, a Linux-based virtual switch appliance used in OpenStack |
| PXE | Preboot Execution Environment |
| QA | Quality Assurance |
| SAML/SAML2 | A protocol allowing authentication and federation of multiple domains |
| SAN | Storage Area Network |
| SSL | Secure Sockets Layer |
| TLS | Transport Layer Security |
| VIP | Virtual IP address |
| VLAN | Virtual LAN is a networking virtualisation technology |
| VM | Virtual machine, in OSP terms, synonymous with “Workload” or “Guest” |
| VXLAN | Virtual Extensible LAN (VXLAN) is a network virtualisation technology |
| Workload | Synonym for “Guest” or “VM” |
| K8s | Kubernetes |
| RHCOS | Red Hat CoreOS |
| DIY | Do It Yourself |
| VM | Virtual Machine |
| OCR | OpenShift Container Registry (Integrated Registry) |

# Introduction

## Topics Covered

Openshift topology - Discussion about OCP high level architecture, subsystems as well as high-level overview of process, technology and operations.

Design guidelines and architecture - Interactive discussion covering minimal HA architectures for development environment and constraints.

## Scope

The scope of this document is to describe the architecture that Red Hat can offer for Openshift. This document is the result of the design workshop and should cover all requirements that TCS put forward during the workshop. As a part of this project engagement with TCS, Red Hat team will:

● Design and validate Openshift Kubernetes Engine (OKE) 4.16 in prod Datacenters.

## Out of Scope

This document limits the discussions and recommendations only to Red Hat Components involved for TCS.

# Implementation Details

## Environment Setup

Steps to prepare or create the environment for the proposed architecture

### Network Information

*Table 1. Network Data*

|  |  |  |
| --- | --- | --- |
| **Name** | **IP Address Pool** | **Comments** |
| Cluster Network | 10.128.0.0/14 | Non Routable |
| Service Network | 172.30.0.0/16 | Non Routable |

### Network Services

*Table 2. Network Service Data*

|  |  |  |
| --- | --- | --- |
| **Services** | **Host** | **Comments** |
| DNS Server | 10.101.141.119 |  |
| DNS Server | 10.101.141.118 |  |

### Node Information

*Table 3. Nodes Data*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Server FQDN** | **IP** | **Role** | **Subs Used** | **OS** |
| valsno.valpocp.vedantaconnect.com | 10.101.143.23 | Master/Control-Plane/worker | N/A | RHCOS |

### Storage Information

*Table 4. Storage Data*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Application** | **Storage Type** | **CSI Driver** | **Storage Class** | **Size** |
| Registry | File | Trident CSI | ontap-nas-sc | 400GB |
| ELK Application Deployment | block | Trident CSI | ontap-san-storsge-class | 3\*400GB |
| Prometheus and Grafana | block | Trident CSI | ontap-san-storsge-class | 2\*200GB, 2\*7.5GB |

### Certificates

TCS has planned to use self-signed certificates that come with the cluster for the Wild Card (\*.apps) Domain Ingress Controller.

# Technical Implementation

## Check Bastion Node for all the pre-Installation requirements

### Operating System Details

• Bastion Node

|  |
| --- |
| [nonprod@bastion ~]$ cat /etc/redhat-release  Red Hat Enterprise Linux release 9.4 (Plow) |

### Block Device (Hard Disk) Details

• Bastion Node

|  |
| --- |
| [nonprod@bastion ~]$ df -hT  Filesystem Type Size Used Avail Use% Mounted on  devtmpfs devtmpfs 4.0M 0 4.0M 0% /dev  tmpfs tmpfs 1.8G 0 1.8G 0% /dev/shm  tmpfs tmpfs 732M 8.7M 723M 2% /run  /dev/mapper/rhel-root xfs 145G 6.2G 139G 5% /  /dev/sda1 xfs 960M 266M 695M 28% /boot  tmpfs tmpfs 366M 4.0K 366M 1% /run/user/0 |

### Network Configuration

* Bastion Node

|  |
| --- |
| [nonprod@bastion ~]$ cat /etc/resolv.conf  # Generated by NetworkManager  search valpocp.vedantaconnect.com  nameserver 101.101.141.119  nameserver 10.101.141.118 |

### Route and Default Gateway details

* Bastion Node

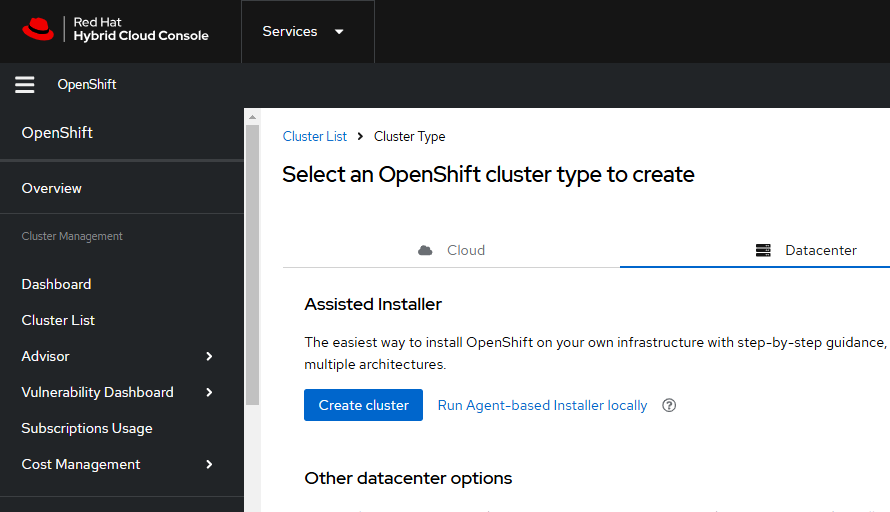
|  |
| --- |
| [nonprod@bastion ~]$ route -n  Kernel IP routing table  Destination Gateway Genmask Flags Metric Ref Use Iface  0.0.0.0 10.101.143.1 0.0.0.0 UG 100 0 0 eth0  10.101.143.0 0.0.0.0 255.255.255.0 U 100 0 0 eth0 |

## Installing OpenShift on a single node.

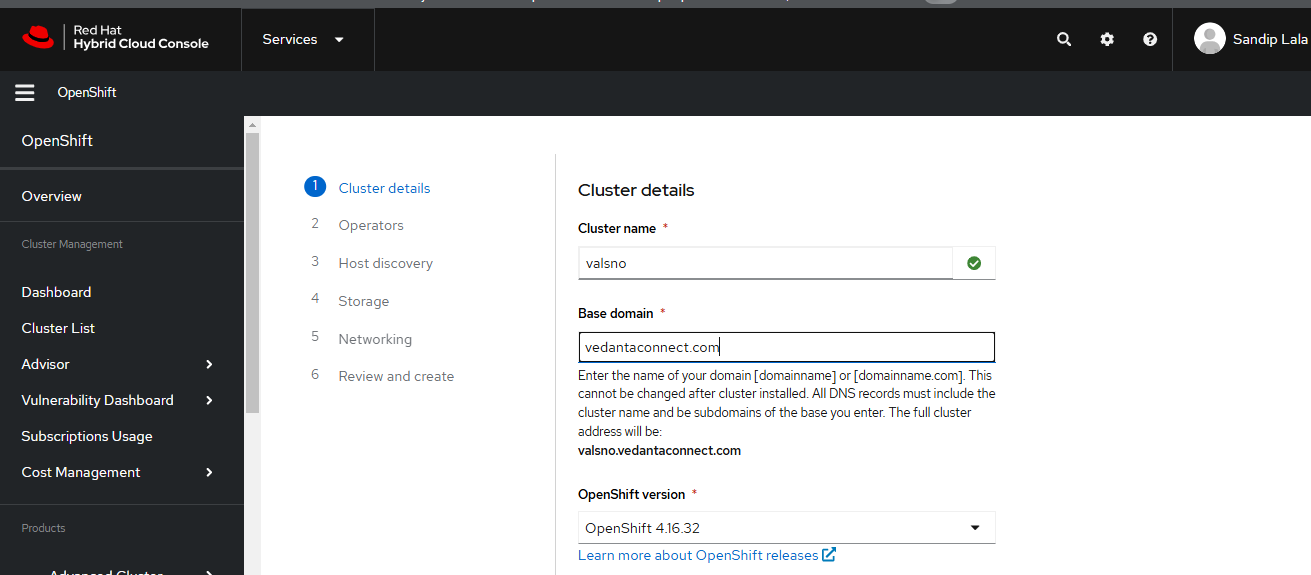
### Generating the discovery ISO with the Assisted Installer.

Installing OpenShift Container Platform on a single node requires a discovery ISO, which the Assisted Installer can generate

1. On the administration host, open a browser and navigate to Red Hat OpenShift Cluster Manager.
2. Click Create New Cluster to create a new cluster.



1. In the Cluster name field, enter a name for the cluster.
2. In the Base domain field, enter a base domain.



1. Select Install single node OpenShift (SNO) and complete the rest of the wizard steps. Download the discovery ISO.

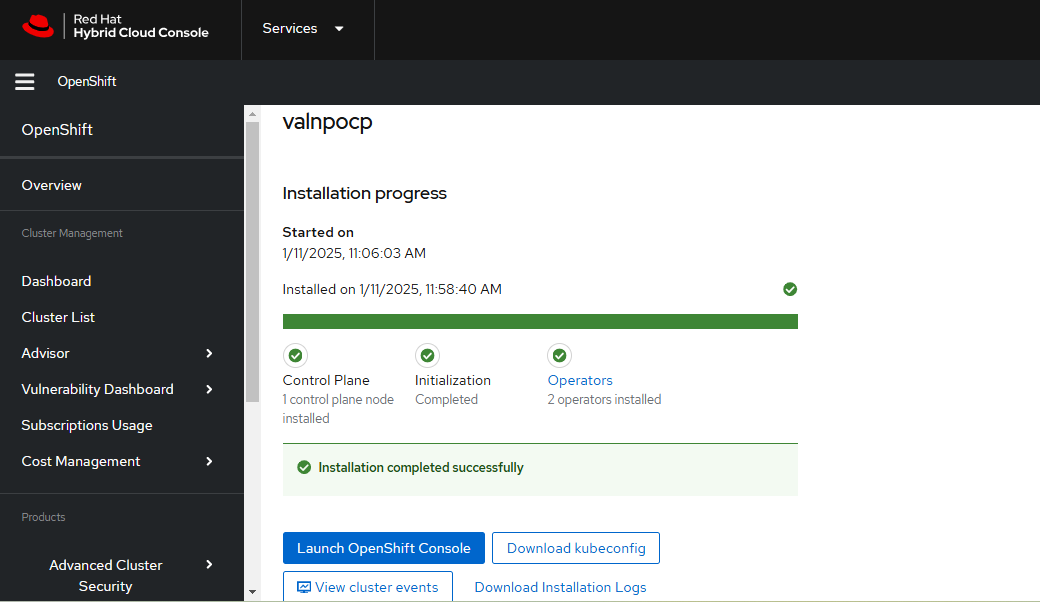
### Install the Single Node using the discovery ISO

**Prerequisites**:

* Ensure that the boot drive order in the server BIOS settings defaults to booting the server from the target installation disk.

Procedure

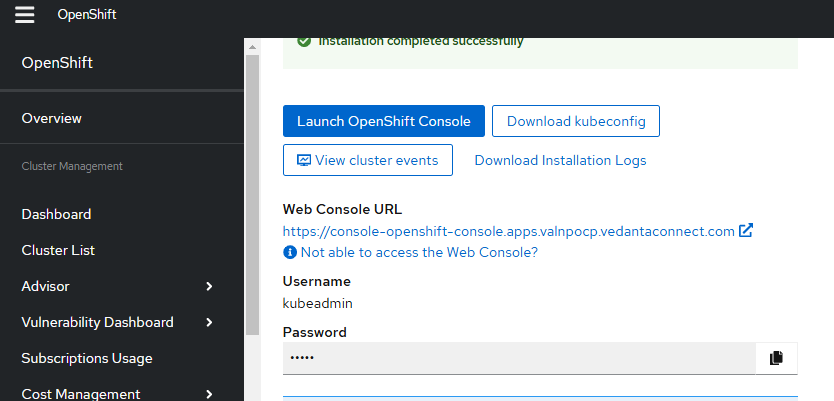
1. Attach the discovery ISO image to the target host.
2. Boot the server from the discovery ISO image. The discovery ISO image writes the system configuration to the target installation disk and automatically triggers a server restart.
3. On the administration host, return to the browser. Wait for the host to appear in the list of discovered hosts. If necessary, reload the Assisted Clusters page and select the cluster name.
4. Complete the install wizard steps. Add networking details, including a subnet from the available subnets. Add the SSH public key if necessary.
5. Monitor the installation’s progress. Watch the cluster events. After the installation process finishes writing the operating system image to the server’s hard disk, the server restarts.
6. Optional: Remove the discovery ISO image.
7. The server restarts several times automatically, deploying the control plane.



## Accessing the Cluster

### Accessing the Openshift Console

1. Navigate to the [OpenShift Console URL](https://console-openshift-console.apps.valnpocp.vedantaconnect.com/) and log in as the 'kubeadmin' user.



## Trident CSI driver Installation and configuration

### Trident operator Installation

* Trident Binaries download and installation
* Download trident installer from

https://github.com/NetApp/trident/releases/download/v24.10.0/trident-installer-24.10.0.tar.gz

[nonprod@bastion trident-new-release]$ ll

drwxr-xr-x. 6 nonprod nonprod 84 Jan 24 16:55 trident-installer

-rw-r--r--. 1 nonprod nonprod 91725414 Jan 24 16:55 trident-installer-24.10.0.tar.gz

[nonprod@bastion trident-installer]$ ll

total 120364

drwxr-xr-x. 3 nonprod nonprod 4096 Jan 24 16:55 deploy

drwxr-xr-x. 4 nonprod nonprod 30 Jan 24 16:55 extras

drwxr-xr-x. 2 nonprod nonprod 45 Jan 24 16:55 helm

drwxr-xr-x. 7 nonprod nonprod 132 Jan 24 16:55 sample-input

-rwxr-xr-x. 1 nonprod nonprod 123244696 Jan 24 16:55 tridentctl

[nonprod@bastion trident-installer]$ ./tridentctl images

+--------------------+---------------------------------------------------------------+

| KUBERNETES VERSION | CONTAINER IMAGE |

+--------------------+---------------------------------------------------------------+

| v1.25.0 | netapp/trident:24.10.0 |

| | docker.io/netapp/trident-autosupport:24.10 |

| | registry.k8s.io/sig-storage/csi-provisioner:v5.1.0 |

| | registry.k8s.io/sig-storage/csi-attacher:v4.7.0 |

| | registry.k8s.io/sig-storage/csi-resizer:v1.12.0 |

| | registry.k8s.io/sig-storage/csi-snapshotter:v8.1.0 |

| | registry.k8s.io/sig-storage/csi-node-driver-registrar:v2.12.0 |

| | netapp/trident-operator:24.10.0 (optional) |

+--------------------+---------------------------------------------------------------+

| v1.26.0 | netapp/trident:24.10.0 |

| | docker.io/netapp/trident-autosupport:24.10 |

| | registry.k8s.io/sig-storage/csi-provisioner:v5.1.0 |

| | registry.k8s.io/sig-storage/csi-attacher:v4.7.0 |

| | registry.k8s.io/sig-storage/csi-resizer:v1.12.0 |

| | registry.k8s.io/sig-storage/csi-snapshotter:v8.1.0 |

| | registry.k8s.io/sig-storage/csi-node-driver-registrar:v2.12.0 |

| | netapp/trident-operator:24.10.0 (optional) |

+--------------------+---------------------------------------------------------------+

| v1.27.0 | netapp/trident:24.10.0 |

| | docker.io/netapp/trident-autosupport:24.10 |

| | registry.k8s.io/sig-storage/csi-provisioner:v5.1.0 |

| | registry.k8s.io/sig-storage/csi-attacher:v4.7.0 |

| | registry.k8s.io/sig-storage/csi-resizer:v1.12.0 |

| | registry.k8s.io/sig-storage/csi-snapshotter:v8.1.0 |

| | registry.k8s.io/sig-storage/csi-node-driver-registrar:v2.12.0 |

| | netapp/trident-operator:24.10.0 (optional) |

+--------------------+---------------------------------------------------------------+

| v1.28.0 | netapp/trident:24.10.0 |

| | docker.io/netapp/trident-autosupport:24.10 |

| | registry.k8s.io/sig-storage/csi-provisioner:v5.1.0 |

| | registry.k8s.io/sig-storage/csi-attacher:v4.7.0 |

| | registry.k8s.io/sig-storage/csi-resizer:v1.12.0 |

| | registry.k8s.io/sig-storage/csi-snapshotter:v8.1.0 |

| | registry.k8s.io/sig-storage/csi-node-driver-registrar:v2.12.0 |

| | netapp/trident-operator:24.10.0 (optional) |

+--------------------+---------------------------------------------------------------+

| v1.29.0 | netapp/trident:24.10.0 |

| | docker.io/netapp/trident-autosupport:24.10 |

| | registry.k8s.io/sig-storage/csi-provisioner:v5.1.0 |

| | registry.k8s.io/sig-storage/csi-attacher:v4.7.0 |

| | registry.k8s.io/sig-storage/csi-resizer:v1.12.0 |

| | registry.k8s.io/sig-storage/csi-snapshotter:v8.1.0 |

| | registry.k8s.io/sig-storage/csi-node-driver-registrar:v2.12.0 |

| | netapp/trident-operator:24.10.0 (optional) |

+--------------------+---------------------------------------------------------------+

| v1.30.0 | netapp/trident:24.10.0 |

| | docker.io/netapp/trident-autosupport:24.10 |

| | registry.k8s.io/sig-storage/csi-provisioner:v5.1.0 |

| | registry.k8s.io/sig-storage/csi-attacher:v4.7.0 |

| | registry.k8s.io/sig-storage/csi-resizer:v1.12.0 |

| | registry.k8s.io/sig-storage/csi-snapshotter:v8.1.0 |

| | registry.k8s.io/sig-storage/csi-node-driver-registrar:v2.12.0 |

| | netapp/trident-operator:24.10.0 (optional) |

+--------------------+---------------------------------------------------------------+

| v1.31.0 | netapp/trident:24.10.0 |

| | docker.io/netapp/trident-autosupport:24.10 |

| | registry.k8s.io/sig-storage/csi-provisioner:v5.1.0 |

| | registry.k8s.io/sig-storage/csi-attacher:v4.7.0 |

| | registry.k8s.io/sig-storage/csi-resizer:v1.12.0 |

| | registry.k8s.io/sig-storage/csi-snapshotter:v8.1.0 |

| | registry.k8s.io/sig-storage/csi-node-driver-registrar:v2.12.0 |

| | netapp/trident-operator:24.10.0 (optional) |

+--------------------+---------------------------------------------------------------+

...

Create trident project

…

[nonprod@bastion trident-installer]$ **oc new-project trident**

[nonprod@bastion trident-installer]$ cd deploy/crds/

[nonprod@bastion crds]$ ll

total 36

-rw-r--r--. 1 nonprod nonprod 1485 Jan 24 16:55 trident.netapp.io\_tridentconfigurators\_crd.yaml

-rw-r--r--. 1 nonprod nonprod 585 Jan 24 16:55 trident.netapp.io\_tridentorchestrators\_crd\_post1.16.yaml

-rw-r--r--. 1 nonprod nonprod 585 Jan 24 16:55 trident.netapp.io\_tridentorchestrators\_crd.yaml

-rw-r--r--. 1 nonprod nonprod 168 Jan 24 16:55 tridentorchestrator\_cr\_audit\_log.yaml

-rw-r--r--. 1 nonprod nonprod 261 Jan 24 16:55 tridentorchestrator\_cr\_autosupport.yaml

-rw-r--r--. 1 nonprod nonprod 178 Jan 24 16:55 tridentorchestrator\_cr\_customimage.yaml

-rw-r--r--. 1 nonprod nonprod 179 Jan 24 16:55 tridentorchestrator\_cr\_default.yaml

-rw-r--r--. 1 nonprod nonprod 203 Jan 24 16:55 tridentorchestrator\_cr\_imagepullsecrets.yaml

-rw-r--r--. 1 nonprod nonprod 195 Jan 24 16:55 tridentorchestrator\_cr.yaml

[nonprod@bastion crds]$ pwd

/home/nonprod/post\_config\_files/trident/trident-new-release/trident-installer/deploy/crds

[nonprod@bastion crds]$

[nonprod@bastion crds]$ **oc create -f trident.netapp.io\_tridentorchestrators\_crd.yaml**

[nonprod@bastion crds]$ cd ..

[nonprod@bastion deploy]$ **oc create -f bundle\_post\_1\_25.yaml**

[nonprod@bastion deploy]$ pwd

/home/nonprod/post\_config\_files/trident/trident-new-release/trident-installer/deploy

[nonprod@bastion deploy]$

[nonprod@bastion deploy]$ cd crds/

[nonprod@bastion crds]$ **oc create -f tridentorchestrator\_cr.yaml**

[nonprod@bastion crds]$ **oc get tridentorchestrators.trident.netapp.io trident -n trident**

NAME AGE

trident 13d

* Verify trident operator, controller and daemonset pods are running

[nonprod@bastion ~]$ oc get pods -n trident -owide

NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES

trident-controller-66b97d5f8d-cw6pt 6/6 Running 24 27d 10.128.0.120 valsno.valnpocp.vedantaconnect.com <none> <none>

trident-node-linux-mm8zw 2/2 Running 17 (3d22h ago) 27d 10.101.143.23 valsno.valnpocp.vedantaconnect.com <none> <none>

trident-operator-568cfdfc6b-qdxjn 1/1 Running 4 27d 10.128.0.118 valsno.valnpocp.vedantaconnect.com <none> <none>

[nonprod@bastion trident-installer]$ **./tridentctl version -n trident**

+----------------+----------------+

| SERVER VERSION | CLIENT VERSION |

+----------------+----------------+

| 24.10.0 | 24.10.0 |

+----------------+----------------+

### Trident RHCOS changes

[nonprod@bastion netapp-csi]$ cat mcp\_iscsi.yaml

apiVersion: machineconfiguration.openshift.io/v1

kind: MachineConfig

metadata:

name: 99-worker-ontap-iscsi

labels:

machineconfiguration.openshift.io/role: master

spec:

config:

ignition:

version: 3.2.0

storage:

files:

- contents:

source:data:text/plain;charset=utf-8;base64,ZGVmYXVsdHMgewogICAgICAgIHVzZXJfZnJpZW5kbHlfbmFtZXMgbm8KICAgICAgICBmaW5kX211bHRpcGF0aHMgbm8KfQoKYmxhY2tsaXN0X2V4Y2VwdGlvbnMgewogICAgICAgIHByb3BlcnR5ICIoU0NTSV9JREVOVF98SURfV1dOKSIKfQoKYmxhY2tsaXN0IHsKfQoK

verification: {}

filesystem: root

mode: 420

path: /etc/multipath.conf

systemd:

units:

- name: iscsid.service

enabled: true

state: started

- name: multipathd.service

enabled: true

state: started

[nonprod@bastion trident-installer]$ cd ~/post\_config\_files/netapp-csi/

[nonprod@bastion netapp-csi]$ oc apply -f mcp\_iscsi.yaml

machineconfig.machineconfiguration.openshift.io/99-worker-ontap-iscsi created

### Trident Backend Config

* Create TBC(trident backend config) for ontap SAN and NAS

[nonprod@bastion netapp-csi]$ pwd

/home/nonprod/post\_config\_files/netapp-csi

[nonprod@bastion netapp-csi]$ ll

total 24

-rw-r--r--. 1 nonprod nonprod 477 Jan 28 18:36 backend-tbc-ontap-nas.yaml

-rw-r--r--. 1 nonprod nonprod 473 Jan 29 19:05 backend-tbc-ontap-san.yaml

-rw-r--r--. 1 nonprod nonprod 820 Jan 29 18:45 mcp\_iscsi.yaml

-rw-r--r--. 1 nonprod nonprod 261 Jan 24 16:55 storage-class-ontapnas.yaml

-rw-r--r--. 1 nonprod nonprod 283 Jan 31 12:22 storage-class-ontap-san-ext4.yaml

-rw-r--r--. 1 nonprod nonprod 260 Jan 29 19:14 storage-class-ontap-san.yaml

[nonprod@bastion netapp-csi]$ cat backend-tbc-ontap-san.yaml

apiVersion: v1

kind: Secret

metadata:

name: backend-tbc-ontap-san-secret

type: Opaque

stringData:

username: vsadmin

password: xxxxxxxxx

---

apiVersion: trident.netapp.io/v1

kind: TridentBackendConfig

metadata:

name: backend-tbc-ontap-san

spec:

version: 1

storageDriverName: ontap-san

managementLIF: 10.101.143.30

backendName: tbc-ontap-san

sanType: fcp

svm: svm\_san

credentials:

name: backend-tbc-ontap-san-secret

[nonprod@bastion netapp-csi]$ oc apply -f backend-tbc-ontap-san.yaml -n trident

secret/backend-tbc-ontap-san-secret created

tridentbackendconfig.trident.netapp.io/backend-tbc-ontap-san created

[nonprod@bastion netapp-csi]$ cat backend-tbc-ontap-nas.yaml

apiVersion: v1

kind: Secret

metadata:

name: backend-tbc-ontap-nas-secret

type: Opaque

stringData:

username: admin

password: xxxxxxxxx

---

apiVersion: trident.netapp.io/v1

kind: TridentBackendConfig

metadata:

name: backend-tbc-ontap-nas

spec:

version: 1

storageDriverName: ontap-nas

managementLIF: 10.101.143.33

dataLIF: 10.101.143.34

backendName: tbc-ontap-nas

svm: svm\_nas

credentials:

name: backend-tbc-ontap-nas-secret

[nonprod@bastion netapp-csi]$ oc apply -f backend-tbc-ontap-nas.yaml -n trident

secret/backend-tbc-ontap-nas-secret created

tridentbackendconfig.trident.netapp.io/backend-tbc-ontap-nas created

[nonprod@bastion ~]$ oc get tbc -n trident

NAME BACKEND NAME BACKEND UUID PHASE STATUS

backend-tbc-ontap-nas tbc-ontap-nas 4e8aa902-7592-43f2-bc2a-63c8f61cbcd5 Bound Success

backend-tbc-ontap-san backend-tbc-ontap-san bc6ca4c3-b2d3-4897-bfb9-8727ad1d1e1a Bound Success

### Trident Storage class Config

* Create storage class to bound with backend config for SAN/NAS.

[nonprod@bastion netapp-csi]$ cat storage-class-ontapnas.yaml

apiVersion: storage.k8s.io/v1

kind: StorageClass

metadata:

name: ontap-nas-sc

provisioner: csi.trident.netapp.io

parameters:

backendType: "ontap-nas"

provisioningType: "thin"

volumeBindingMode: Immediate

reclaimPolicy: Retain

allowVolumeExpansion: true

[nonprod@bastion netapp-csi]$ oc apply -f storage-class-ontapnas.yaml

storageclass.storage.k8s.io/ontap-nas-sc created

[nonprod@bastion netapp-csi]$ cat storage-class-ontap-san.yaml

apiVersion: storage.k8s.io/v1

kind: StorageClass

metadata:

name: ontap-san-storage-class

provisioner: csi.trident.netapp.io

parameters:

backendType: "ontap-san"

provisioningType: "thin"

allowVolumeExpansion: True

volumeBindingMode: Immediate

reclaimPolicy: Delete

[nonprod@bastion netapp-csi]$ oc apply -f storage-class-ontap-san.yaml

storageclass.storage.k8s.io/ ontap-san-storage-class created

[nonprod@bastion ~]$ oc get sc

NAME PROVISIONER RECLAIMPOLICY VOLUMEBINDINGMODE ALLOWVOLUMEEXPANSION AGE

ontap-nas-sc csi.trident.netapp.io Retain Immediate true 28d

ontap-san-storage-class (default) csi.trident.netapp.io Retain Immediate true 27d [nonprod@bastion netapp-csi]$ cd ../trident/trident-new-release/trident-installer/

[nonprod@bastion trident-installer]$ ./tridentctl get storageclass -n trident

+-------------------+

| NAME |

+-------------------+

| ontap-nas-sc |

| ontap-san-storage-class |

+-------------------+

### Identify backend for storage class

* Valiidate class to backend mapping.

[nonprod@bastion trident-installer]$ ./tridentctl get storageclass -n trident -o json | jq '[.items[] | {storageClass: .Config.name, backends: [.storage]|unique}]'

[

{

"storageClass": "ontap-nas-sc",

"backends": [

{

"tbc-ontap-nas": [

"data\_aggr1"

]

}

]

},

{

"storageClass": " ontap-san-storage-class",

"backends": [

{

"tbc-ontap-san": [

"data\_aggr1"

]

}

]

}

]

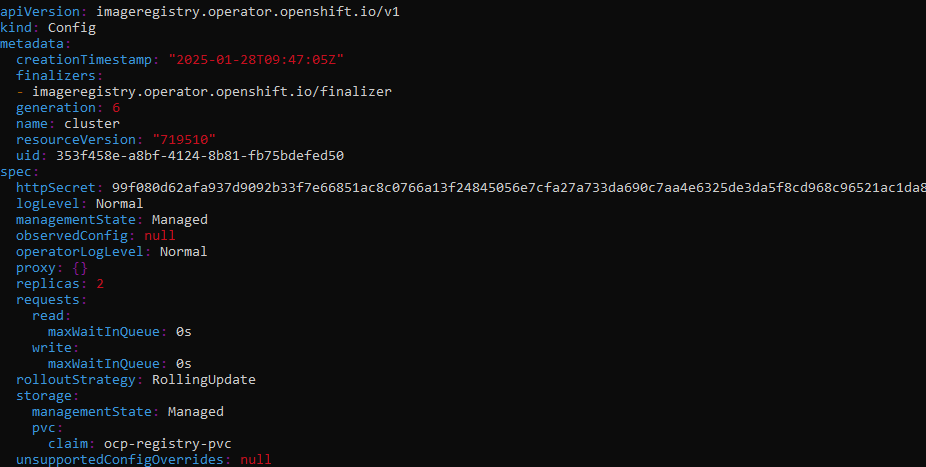
}

]

## Deploying Openshift Internal Registry with Persistent Storage

* By default, there will be no registry pod running as the registry operator’s "managementState" will be set as "Removed". Configure image registry to "Managed" state and move the registry pods to the dedicated infra nodes as shown below. Right now an object storage volume is being used inside the registry pods.

|  |
| --- |
| oc edit configs.imageregistry.operator.openshift.io/cluster |



* Create a PVC “ocp-registry-pvc” and Verify if the registry pod is running.

|  |
| --- |
| [dbpuser@bastion ~]$  [nonprod@bastion image-registry]$ cat pvc-provisioning.yaml  apiVersion: v1  kind: PersistentVolumeClaim  metadata:  name: ocp-registry-pvc  spec:  # SAME NAME AS THE STORAGECLASS  storageClassName: ontap-nas-sc  accessModes:  - ReadWriteMany  # Must be the same as PersistentVolume  resources:  requests:  storage: 400Gi  [nonprod@bastion image-registry]$ #oc apply -f pvc-provisioning.yaml  [nonprod@bastion image-registry]$ oc get pods  NAME READY STATUS RESTARTS AGE  cluster-image-registry-operator-5d89674fbd-fg6rn 1/1 Running 0 13d  image-registry-6985c79cbc-97xgx 1/1 Running 0 13d |

## Configure System Clock Sync with NTP Server

### Configure master nodes system clock sync with NTP server

* Create a Butane config including the contents of the chrony.conf file, to configure chrony on worker nodes, create a 99-worker-chrony.bu file and use Butane to generate a MachineConfig object file, 99- worker-chrony.yaml, containing the configuration to be delivered to the nodes. Repeat the same steps for other machine-config-pool like master. .

|  |
| --- |
| [nonprod@bastion ntp\_config]$ cat 99-worker-chrony.bu  variant: openshift  version: 4.16.0  metadata:  name: 99-worker-chrony  labels:  machineconfiguration.openshift.io/role: worker  storage:  files:  - path: /etc/chrony.conf  mode: 0644  overwrite: true  contents:  inline: |  server 10.101.141.3 iburst  driftfile /var/lib/chrony/drift  makestep 1.0 3  rtcsync  logdir /var/log/chrony |

|  |
| --- |
| [nonprod@bastion ntp\_config]$./butane 99-worker-chrony.bu -o 99-worker-chrony.yaml |

* Apply the MachineConfig YAML file

|  |
| --- |
| [nonprod@bastion ntp\_config]$ oc create -f ./99-worker-chrony.yaml  machineconfig.machineconfiguration.openshift.io/99-worker-chrony created  [nonprod@bastion ntp\_config]$ pwd  /home/nonprod/post\_config\_files/ntp\_config  [nonprod@bastion ntp\_config]$ ll  total 7900  -rw-r--r--. 1 nonprod nonprod 387 Jan 24 16:55 99-master-chrony.bu  -rw-r--r--. 1 nonprod nonprod 572 Jan 24 16:55 99-master-chrony.yaml  -rw-r--r--. 1 nonprod nonprod 387 Jan 24 17:12 99-worker-chrony.bu  -rw-r--r--. 1 nonprod nonprod 572 Jan 24 17:13 99-worker-chrony.yaml  -rwxr-xr-x. 1 nonprod nonprod 8070568 Jan 24 16:55 butane |

* Once NTP config are applied, the single node will get rebooted and come back online.

|  |
| --- |
| [nonprod@bastion ntp\_config]$ ssh core@10.101.143.23  [core@valsno ~]$ chronyc sources  MS Name/IP address Stratum Poll Reach LastRx Last sample  ===============================================================================  ^\* 10.101.141.3 1 10 377 896 +72us[ +98us] +/- 1015us |

|  |
| --- |
| [core@valsno ~]$ chronyc tracking  Reference ID : 0A658D03 (10.101.141.3)  Stratum : 2  Ref time (UTC) : Wed Feb 05 12:23:08 2025  System time : 0.000058596 seconds fast of NTP time  Last offset : +0.000025681 seconds  RMS offset : 0.000192539 seconds  Frequency : 16.776 ppm fast  Residual freq : +0.001 ppm  Skew : 0.121 ppm  Root delay : 0.001474854 seconds  Root dispersion : 0.001345289 seconds  Update interval : 1031.3 seconds  Leap status : Normal |

Repeat the above commands on rest of master nodes to verify the system clock with NTP and make sure "System clock synchronized: yes"

## Configuring an HTPasswd identity provider

* Create a flat file with a user name and hashed password

|  |
| --- |
| [nonprod@bastion htpasswd]$ pwd  /home/nonprod/post\_config\_files/auth\_providers/htpasswd  [nonprod@bastion htpasswd]$ cd ..  [nonprod@bastion auth\_providers]$ pwd  /home/nonprod/post\_config\_files/auth\_providers  [nonprod@bastion auth\_providers]$ mkdir htpasswd  [nonprod@bastion auth\_providers]$ cd htpasswd/  [nonprod@bastion auth\_providers]$ htpasswd -cB users.htpasswd ocpadmin  New password:  Re-type new password:  Adding password for user ocpadmin |

The password used for the ocpadmin account hasn’t been disclosed here but the same has been shared with the customer.

* Create an OpenShift Container Platform Secret object that contains the HTPasswd users file

[

[nonprod@bastion auth\_providers]$ oc create secret generic htpass-secret --from-file=htpasswd=users.htpasswd -n openshift-config

secret/htpass-secret created

The secret key containing the users file for the --from-file argument must be named

**htpasswd**

, as shown in the above command.

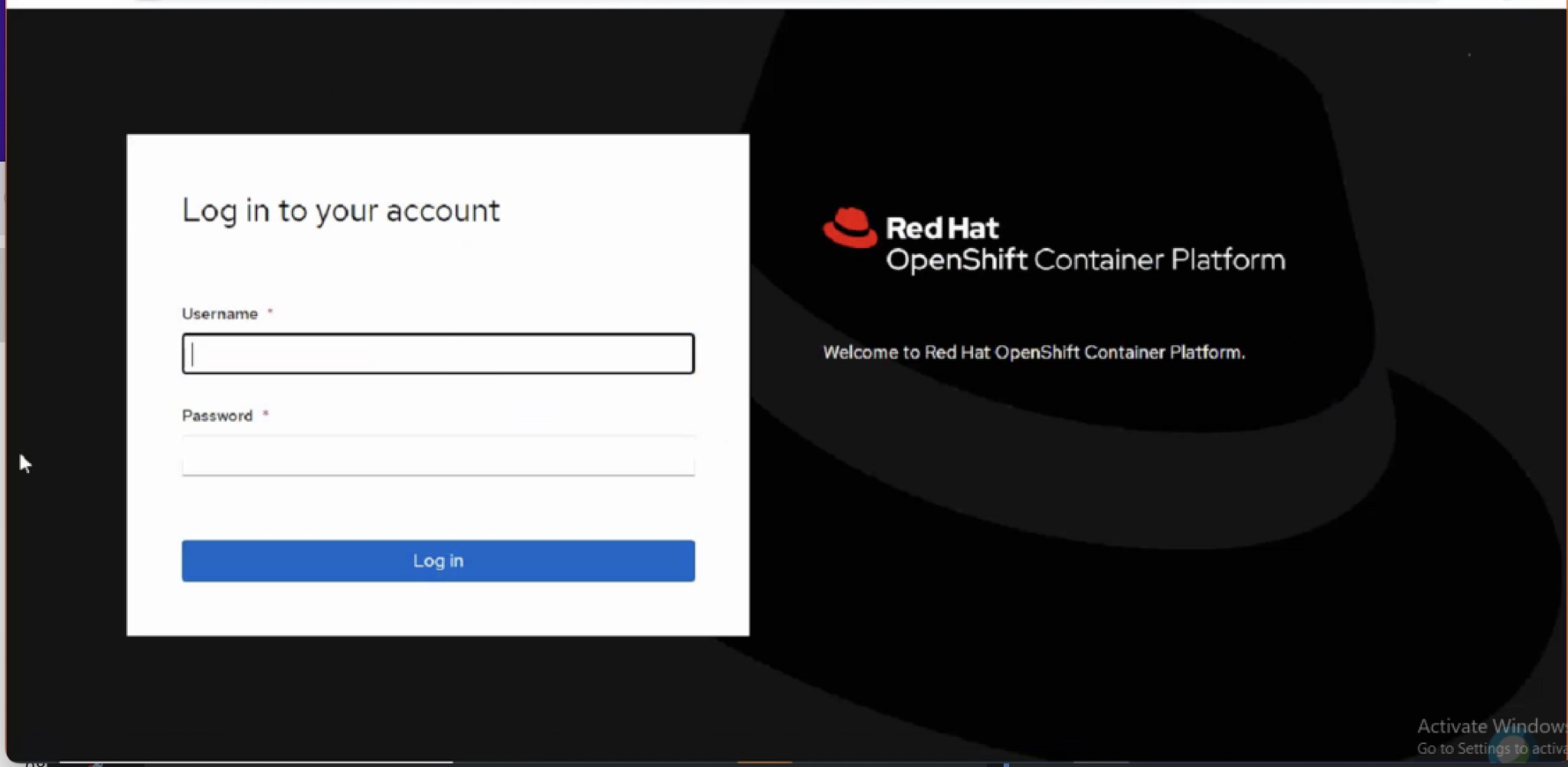
* Create HTPasswd identity provider YAML manifests as below

|  |
| --- |
| [nonprod@bastion htpasswd]$ cat oauth.yaml  apiVersion: config.openshift.io/v1  kind: OAuth  metadata:  name: cluster  spec:  identityProviders:  - name: VAL\_ITMS\_htpasswd  mappingMethod: claim  type: HTPasswd  htpasswd:  fileData:  name: htpasswd-secret |

* Adding HTPasswd identity provider to the cluster

|  |
| --- |
| [nonprod@bastion htpasswd]$ oc apply -f oauth.yaml  Warning: oc apply should be used on resource created by either oc create --save-config or oc apply oauth.config.openshift.io/cluster configured |

* Access the OCP Web Console and click on the HTPasswd identity provider & login using ocpadmin as the user with the respective password set earlier while creating the user account



* Grant the **ocpadmin** user account with **cluster-admin** access as shown below.

|  |
| --- |
| [nonprod@bastion htpasswd]$ oc adm policy add-cluster-role-to-user cluster-admin ocpadmin clusterrole.rbac.authorization.k8s.io/cluster-admin added: "ocpadmin"  [core@rlonvsacmdrcm1 htpasswd]$ cd .. |

Please store the 'ocpadmin' password in a secured place as it has been granted with cluster-admin privilege to manage the entire OCP cluster.

## Registering the Cluster

* Subscription and support

The cluster will be automatically registered with a 60 day evaluation subscription which does not include support. In order to receive support for your cluster, you will need to register OpenShift Container Platform 4 cluster on cloud.redhat.com.

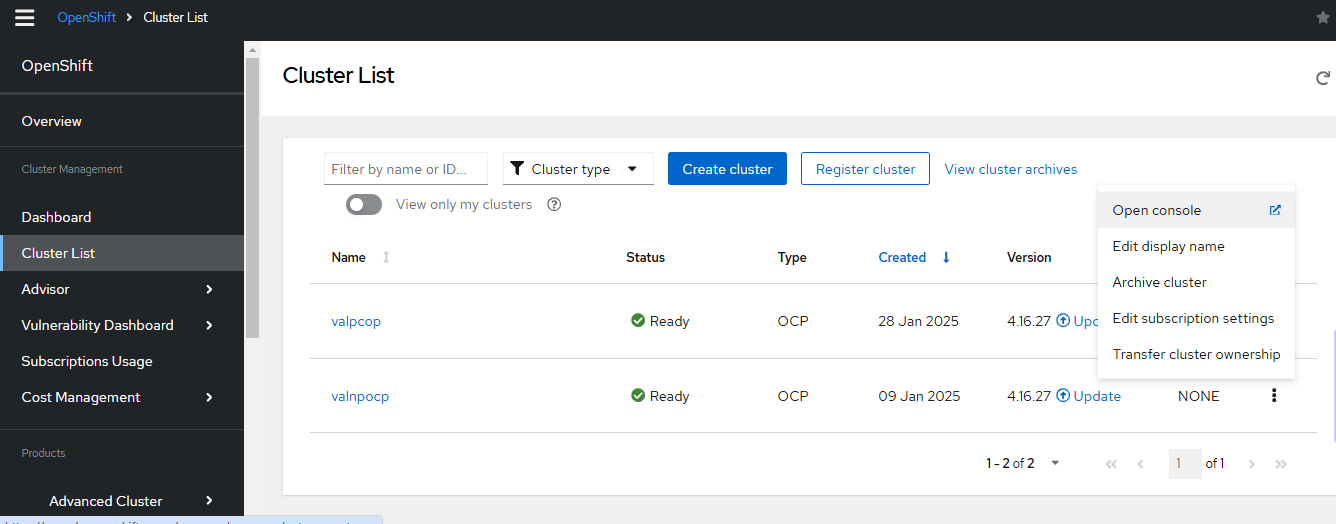
* To register a disconnected OCP 4 cluster on cloud.redhat.com the cluster profile needs to be created manually as outlined below:

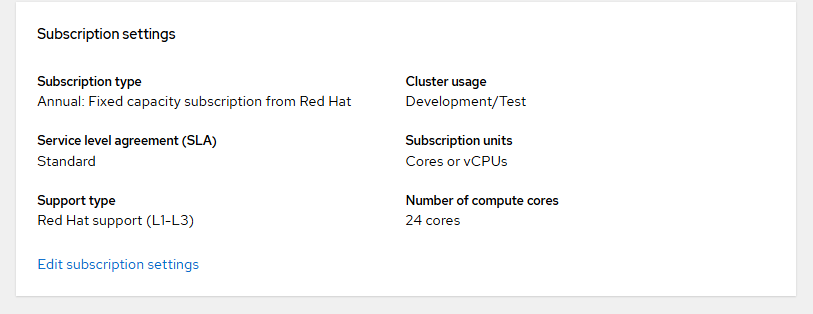
◦ Go to cloud.redhat.com and login with Vedanta\_jharsuguda login credentials.

◦ Go to the Clusters tab and select the three dots near Create cluster and click on the Register cluster button. Alternatively, use [**https://cloud.redhat.com/openshift/register**](https://cloud.redhat.com/openshift/register) ◦ ◦ Enter all the required details like cluster-id, number of sockets, memory, etc.

◦ After filling all the required details click on the Register cluster. This will create a cluster profile in disconnected mode.

◦ After that, click on the respective cluster name in the 'Clusters' tab. Edit the Subscription Settings and attach the subscription according to the cluster configuration and environment by scrolling down to 'Subscription settings'.





To get more details, Please visit the below URL:

<https://access.redhat.com/solutions/4930131>

## Encrypting the ETCD data

* Edit the apiserver object & make an entry under spec as shown below

|  |
| --- |
| [nonprod@bastion]$ oc edit apiserver  …  …  spec:  encryption:  type: aescbc  audit:  profile: Default |

* Verify if all the resource under **openshiftapiserver** and **kubeapiserver** are encrypted as shown below.

|  |
| --- |
| [nonprod@bastion ~]$oc get openshiftapiserver -o=jsonpath='{range .items[0].status.conditions[?(@.type=="Encrypted")]}{.reason}{"\n"}{.message}{"\n"}'  EncryptionCompleted  All resources encrypted: routes.route.openshift.io  [nonprod@bastion ~]$ oc get kubeapiserver -o=jsonpath='{range .items[0].status.conditions[?(@.type=="Encrypted")]}{.reason}{"\n"}{.message}{"\n"}'  EncryptionCompleted  All resources encrypted: secrets, configmaps |

## Backing Up ETCD data

* Login to one of the master node and perform the backup of ETCD data as shown below

|  |
| --- |
| [nonprod@bastion monitoring]$ ssh core@10.101.143.14  [core@valsno ~]$ sudo -i  [root@valsno ~]# cd /usr/local/bin/  [root@valsno bin]# mkdir -p /home/core/assets/backup  [root@valsno bin]# ll  total 84  -rwxr-xr-x. 1 root root 4791 Jan 28 09:57 cluster-backup.sh  -rwxr-xr-x. 1 root root 5990 Jan 28 09:57 cluster-restore.sh  -rwxr-xr-x. 1 root root 41347 Jan 29 13:03 configure-ovs.sh  -rwxr-xr-x. 1 root root 100 Jan 29 13:03 kubenswrapper  -rwxr-xr-x. 1 root root 2431 Jan 29 13:03 mco-hostname  -rwxr-xr-x. 1 root root 5733 Jan 29 13:03 nm-clean-initrd-state.sh  -rwxr-xr-x. 1 root root 1218 Jan 29 13:03 recover-kubeconfig.sh  -rwxr-xr-x. 1 root root 708 Jan 29 13:03 wait-for-primary-ip.sh  [root@valsno bin]# ./cluster-backup.sh /home/core/assets/backup |

* Create a directory to store the ETCD Backup in Bastion Node

[prod@bastion ~]$ mkdir -p etcd-backups/18-jan-2025

* Copy the backed up files from the master node to the Bastion node as shown below

|  |
| --- |
| [nonprod@bastion ~]$ scp core@10.101.143.14:/home/core/assets/backup/static\_kuberesources\_2025-01-30\_082946.tar.gz ./etcd-backups/18-jan-2025 /  static\_kuberesources\_2025-01-18\_082116.tar.gz 100% 80KB 12.5MB/s 00:00 |

## Deploying and Configuring Openshift Logging

### Install Cluster Logging Operators

* Install the Elasticsearch Operator
* In the OpenShift Container Platform web console, click Operators → OperatorHub.
* Choose Elasticsearch Operator from the list of available Operators, and click Install.
* Ensure that the all namespaces on the cluster is selected under Installation Mode.
* Ensure that openshift-operators-redhat is selected under Installed Namespace.
* Select Enable operator recommended cluster monitoring on this namespace.
* Select stable as the Update Channel.
* Select an Approval Strategy to Automatic
* Click Install.
* Verify that the Elasticsearch Operator installed by switching to the Operators → Installed Operators page.
* Ensure that Elasticsearch Operator is listed in all projects with a Status of Succeeded.
* Install the Cluster Logging Operator.
* In the OpenShift Container Platform web console, click Operators → OperatorHub.
* Choose Cluster Logging from the list of available Operators, and click Install.
* Ensure that the A specific namespace on the cluster is selected under Installation Mode.
* Ensure that Operator recommended namespace is openshift-logging under Installed Namespace.
* Select Enable operator recommended cluster monitoring on this namespace.
* Select stable as the Update Channel.
* Select an Approval Strategy to Automatic. Click Install.
* Verify that the Cluster Logging Operator installed by switching to the Operators → Installed Operators page.
* Ensure that Cluster Logging is listed in the openshift-logging project with a Status of Succeeded.

### Create Cluster Logging Instance

* Create a cluster logging instance YAML manifest
* Create a Cluster Logging instance

|  |
| --- |
| [nonprod@bastion logging]$ cat cluster-logging.yaml  kind: ClusterLogging  apiVersion: logging.openshift.io/v1  metadata:  name: instance  namespace: openshift-logging  spec:  collection:  type: vector  logStore:  elasticsearch:  nodeCount: 3  redundancyPolicy: SingleRedundancy  resources:  limits:  cpu: 2000m  memory: 4Gi  requests:  cpu: 1500m  memory: 4Gi  storage:  size: 400Gi  storageClassName: ontap-san-storage-class  retentionPolicy:  application:  maxAge: 5d  audit:  maxAge: 5d  infra:  maxAge: 5d  type: elasticsearch  managementState: Managed  visualization:  kibana:  replicas: 1  resources:  limits:  memory: 1Gi  requests:  cpu: 500m  memory: 1Gi  type: kibana |

* Create the ClusterLogging instance object as shown below

|  |
| --- |
| [nonprod@bastion logging]$ oc apply -f cluster-logging.yaml  clusterlogging.logging.openshift.io/instance created |

* Verify the cluster logging pods deployed in the openshift-logging project

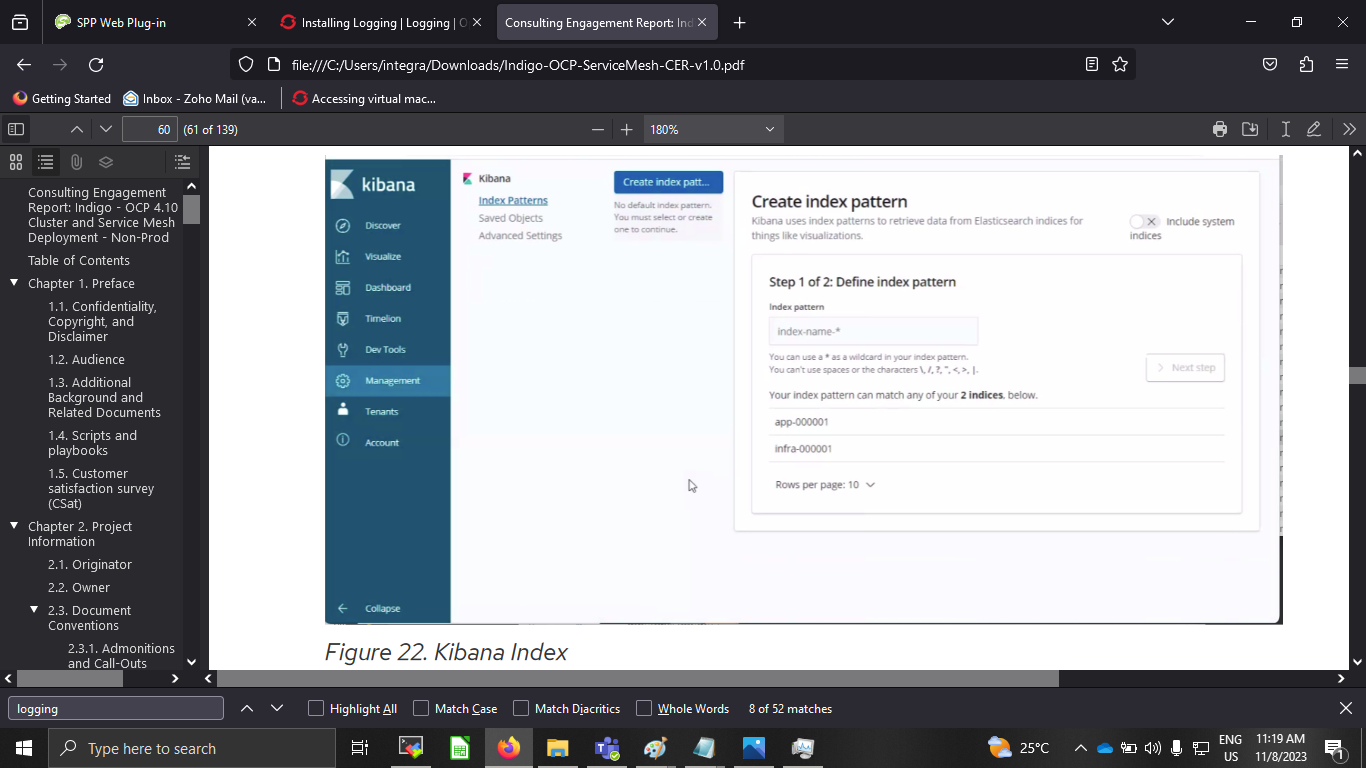
|  |
| --- |
| [nonprod@bastion logging]$ oc get po -o wide -n openshift-logging  NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES  cluster-logging-operator-648bbb699c-rbpjr 1/1 Running 0 37h 10.128.1.172 valsno.valnpocp.vedantaconnect.com <none> <none>  collector-pwg6p 1/1 Running 0 37h 10.128.1.173 valsno.valnpocp.vedantaconnect.com <none> <none>  elasticsearch-cdm-08fdseho-1-866466f64f-ppcqg 2/2 Running 0 40h 10.128.1.64 valsno.valnpocp.vedantaconnect.com <none> <none>  elasticsearch-cdm-08fdseho-2-7d9dc5b78-rmsp5 2/2 Running 0 40h 10.128.1.62 valsno.valnpocp.vedantaconnect.com <none> <none>  elasticsearch-cdm-08fdseho-3-8669db996c-47twq 2/2 Running 0 40h 10.128.1.63 valsno.valnpocp.vedantaconnect.com <none> <none>  elasticsearch-im-app-28991865-d54kt 0/1 Completed 0 12m 10.128.0.251 valsno.valnpocp.vedantaconnect.com <none> <none>  elasticsearch-im-audit-28991865-t2ffw 0/1 Completed 0 12m 10.128.0.250 valsno.valnpocp.vedantaconnect.com <none> <none>  elasticsearch-im-infra-28991865-wrjpz 0/1 Completed 0 12m 10.128.0.249 valsno.valnpocp.vedantaconnect.com <none> <none>  kibana-54cfd6fdbb-xxdgf 2/2 Running 0 40h 10.128.1.60 valsno.valnpocp.vedantaconnect.com <none> <none> |

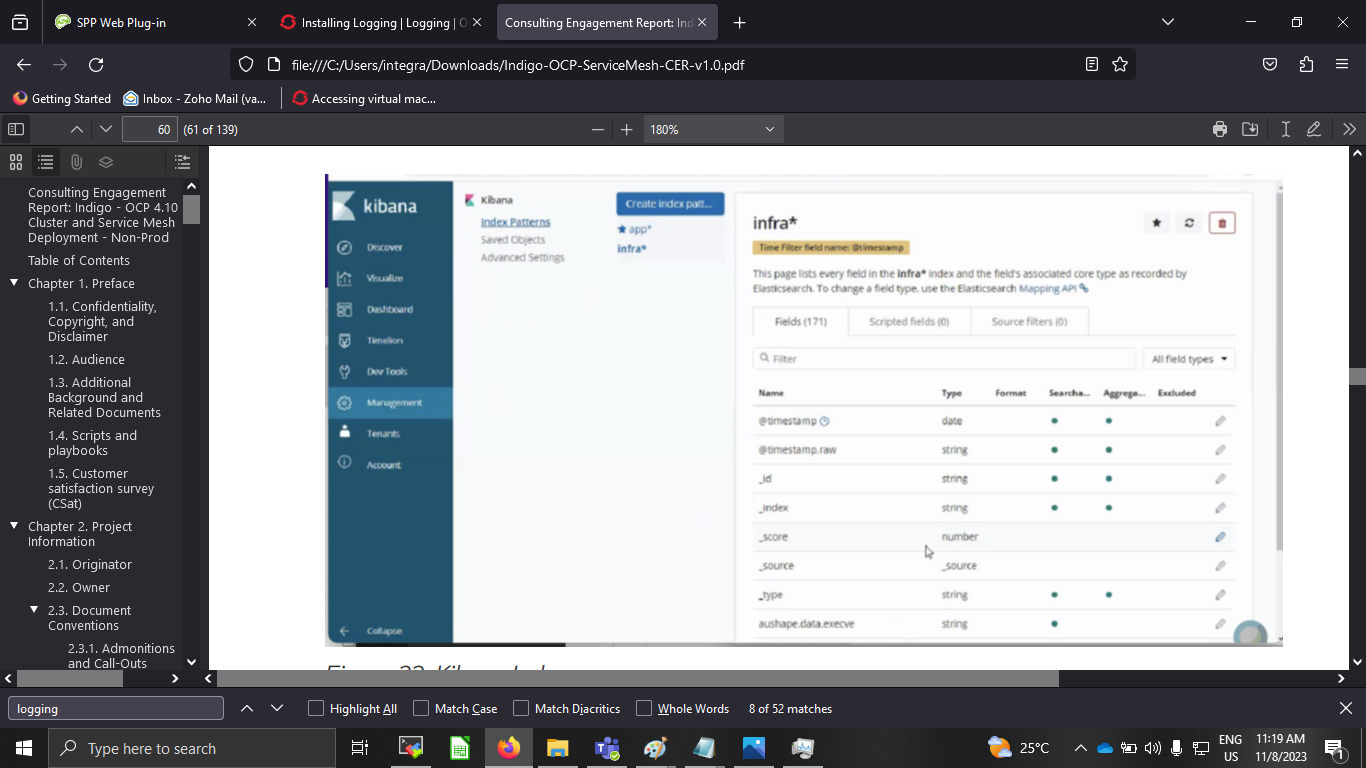
* Verify PVCs created for Elasticsearch components

|  |
| --- |
| [nonprod@bastion logging]$ oc get pvc -n openshift-logging  NAME STATUS VOLUME CAPACITY ACCESS MODES STORAGECLASS VOLUMEATTRIBUTESCLASS AGE  elasticsearch-elasticsearch-cdm-08fdseho-1 Bound pvc-197fb7ec-894a-47fd-9069-c198ad4a94d0 400Gi RWO ontap-san-storage-class <unset> 40h  elasticsearch-elasticsearch-cdm-08fdseho-2 Bound pvc-d3c876d8-09ce-4f67-84d6-6571a932bf9c 400Gi RWO ontap-san-storage-class <unset> 40h  elasticsearch-elasticsearch-cdm-08fdseho-3 Bound pvc-d9235406-9a85-4dbf-b34a-d752ccfd2477 400Gi RWO ontap-san-storage-class <unset> 40h |

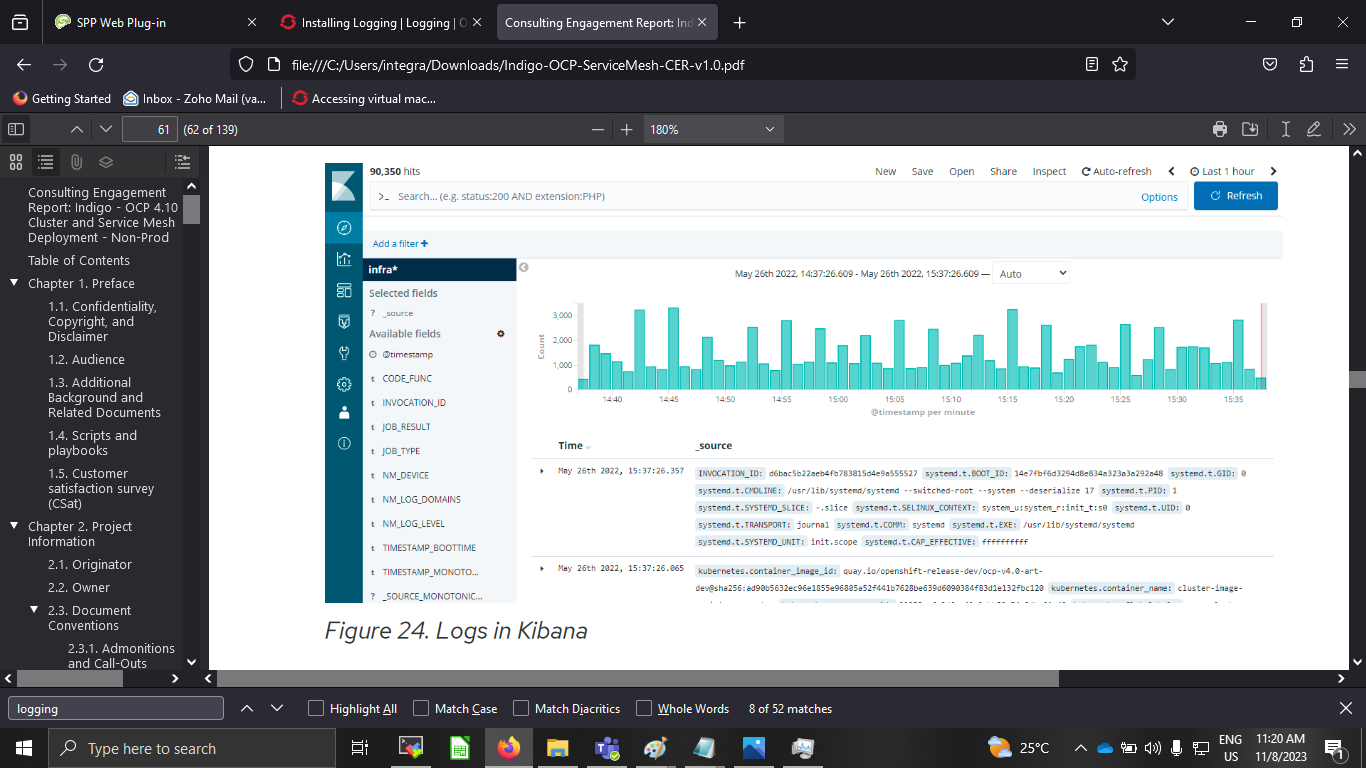
### Viewing cluster logs from the Kibana dashboard

* To define index patterns and create visualizations in Kibana
* In the OpenShift Container Platform console, click the Application Launcher and select Logging.
* Create your Kibana index patterns by clicking Management → Index Patterns → Create index pattern.
  + Users must manually create index patterns to see logs for their projects. Users should create a new index pattern named app and use the @timestamp time field to view their container logs.
  + Admin users must create index patterns for the app, infra, and audit indices using the @timestamp time field.
* Create Kibana Visualizations from the new index patterns.





* To view logs in Kibana
* In the OpenShift Container Platform console, click the Application Launcher and select  
  Logging.
* Log in using the same credentials you use to log in to the OpenShift Container Platform console.
* In Kibana, click Discover.
* Select the index pattern you created from the drop-down menu in the top-left corner: app, audit, or infra. The log data displays as time-stamped documents.
* Expand one of the time-stamped documents.



## Applying a custom Alertmanager configuration with the SMTP details

• Print the currently active Alertmanager configuration into a file **alertmanager.yaml**

|  |
| --- |
| [nonprod@bastion post\_config\_files]$ mkdir smtp  [nonprod@bastion post\_config\_files]$ cd smtp/  [nonprod@bastion smtp]$ oc -n openshift-monitoring get secret alertmanager-main --template='{{ index .data "alertmanager.yaml" }}' | base64 -d > alertmanager.yaml  [nonprod@bastion smtp]$ ll  total 4  -rw-r--r--. 1 nonprod nonprod 657 Feb 12 14:21 alertmanager.yaml |

• Edit the configuration in the exported alertmanager.yaml file

|  |
| --- |
| "global":  "resolve\_timeout": "5m"  "smtp\_from": "VLJ.ITMS@vedanta.co.in"  "smtp\_smarthost": "smtp.office365.com:587"  "smtp\_auth\_username": "VLJ.ITMS@vedanta.co.in"  "smtp\_auth\_password": "H!ghPr0t3ct!0n"  "smtp\_require\_tls": false  "inhibit\_rules":  - "equal":  - "namespace"  - "alertname"  "source\_match":  "severity": "critical"  "target\_match\_re":  "severity": "warning|info"  - "equal":  - "namespace"  - "alertname"  "source\_match":  "severity": "warning"  "target\_match\_re":  "severity": "info"  "receivers":  - "name": "Default"  "email\_configs":  - "to": "samiran.chatterjee@tcs.com"  "headers":  "subject": 'Openshift-NONPROD-ITMS Cluster Alert {{ template "email.default.subject" . }}'  - "to": "suhel.ahemed@tcs.com"  "headers":  "subject": 'Openshift-NONPROD-ITMS Cluster Alert {{ template "email.default.subject" . }}'  - "to": "ahmed.m1@tcs.com"  "headers":  "subject": 'Openshift-NONPROD-ITMS Cluster Alert {{ template "email.default.subject" . }}'  - "to": "praveen.babeley@tcs.com"  "headers":  "subject": 'Openshift-NONPROD-ITMS Cluster Alert {{ template "email.default.subject" . }}'  - "to": "arnab.ghosh4@tcs.com"  "headers":  "subject": 'Openshift-NONPROD-ITMS Cluster Alert {{ template "email.default.subject" . }}'  "route":  "group\_by":  - "namespace"  "group\_interval": "5m"  "group\_wait": "30s"  "receiver": "Default"  "repeat\_interval": "12h"  "routes":  - "match":  "alertname": "Watchdog"  "receiver": "Default"  - "match":  "severity": "critical"  "receiver": "Default" |

Apply the new edited configuration as shown below.

|  |
| --- |
| $ oc -n openshift-monitoring create secret generic alertmanager-main --from-file=alertmanager.yaml --dry-run -o=yaml | oc -n openshift-monitoring  replace secret --filename=-  secret/alertmanager-main replaced |

## Configuring Openshift monitoring components

Create a configmap as shown below

|  |
| --- |
| [nonprod@bastion monitoring]$ cat cluster-monitor-config.yaml  apiVersion: v1  kind: ConfigMap  metadata:  name: cluster-monitoring-config  namespace: openshift-monitoring  data:  config.yaml: |+  prometheusK8s:  volumeClaimTemplate:  metadata:  name: prometheus  spec:  storageClassName: ontap-san-sc-ext4  resources:  requests:  storage: 200Gi  retention: 14d  alertmanagerMain:  volumeClaimTemplate:  metadata:  name: alertmanager  spec:  storageClassName: ontap-san-sc  resources:  requests:  storage: 7.5Gi |

|  |
| --- |
| [nonprod@bastion monitoring]$ oc create -f cluster-monitoring-config.yaml |

|  |
| --- |
| [nonprod@bastion monitoring]$ oc get po -o wide -n openshift-monitoring  NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES  alertmanager-main-0 6/6 Running 18 23d 10.128.0.129 valsno.valnpocp.vedantaconnect.com <none> <none>  cluster-monitoring-operator-78dd644bb4-rg44m 1/1 Running 7 34d 10.128.0.20 valsno.valnpocp.vedantaconnect.com <none> <none>  kube-state-metrics-84774f484d-lk5xl 3/3 Running 21 33d 10.128.0.92 valsno.valnpocp.vedantaconnect.com <none> <none>  metrics-server-744dbcddc-q7qxw 1/1 Running 0 41h 10.128.1.33 valsno.valnpocp.vedantaconnect.com <none> <none>  monitoring-plugin-55dd5f67d4-zgx7r 1/1 Running 7 33d 10.128.0.95 valsno.valnpocp.vedantaconnect.com <none> <none>  node-exporter-5qtpb 2/2 Running 14 33d 10.101.143.23 valsno.valnpocp.vedantaconnect.com <none> <none>  openshift-state-metrics-b9456474c-xmw7j 3/3 Running 21 33d 10.128.0.91 valsno.valnpocp.vedantaconnect.com <none> <none>  prometheus-k8s-0 6/6 Running 18 23d 10.128.0.70 valsno.valnpocp.vedantaconnect.com <none> <none>  prometheus-operator-8474d68cb5-4mb8m 2/2 Running 15 33d 10.128.0.84 valsno.valnpocp.vedantaconnect.com <none> <none>  prometheus-operator-admission-webhook-7cc8ddb54f-8n575 1/1 Running 7 33d 10.128.0.58 valsno.valnpocp.vedantaconnect.com <none> <none>  telemeter-client-5659657475-z8f6n 3/3 Running 22 33d 10.128.0.94 valsno.valnpocp.vedantaconnect.com <none> <none>  thanos-querier-f67f6b7c5-w4t9q 6/6 Running 43 33d 10.128.0.117 valsno.valnpocp.vedantaconnect.com <none> <none> |

## Openshift Virtualization

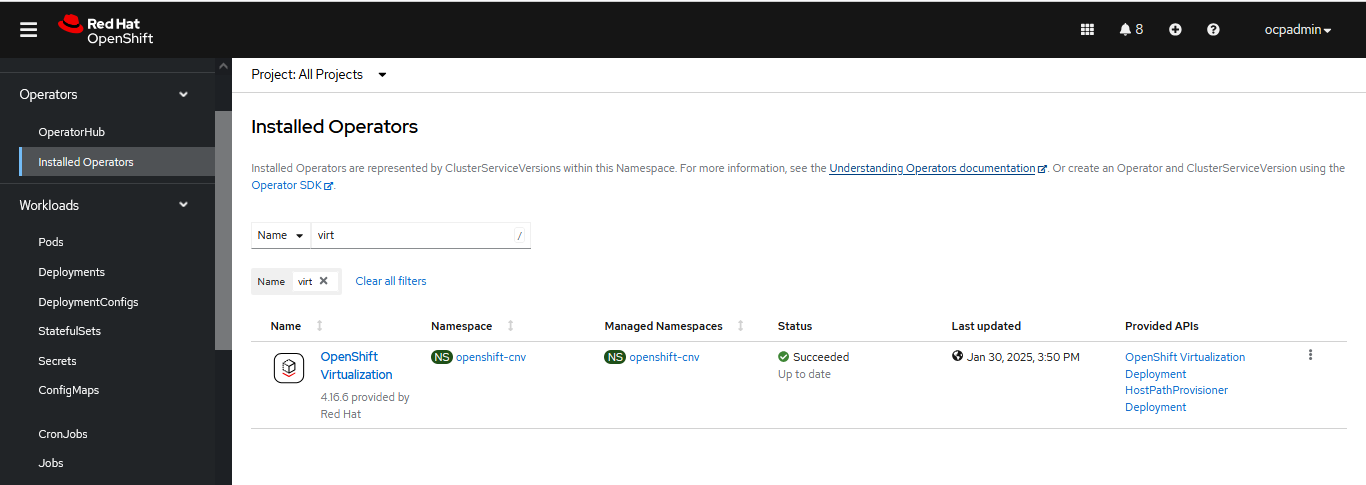
### Installing the OpenShift Virtualization Operator by using the web console

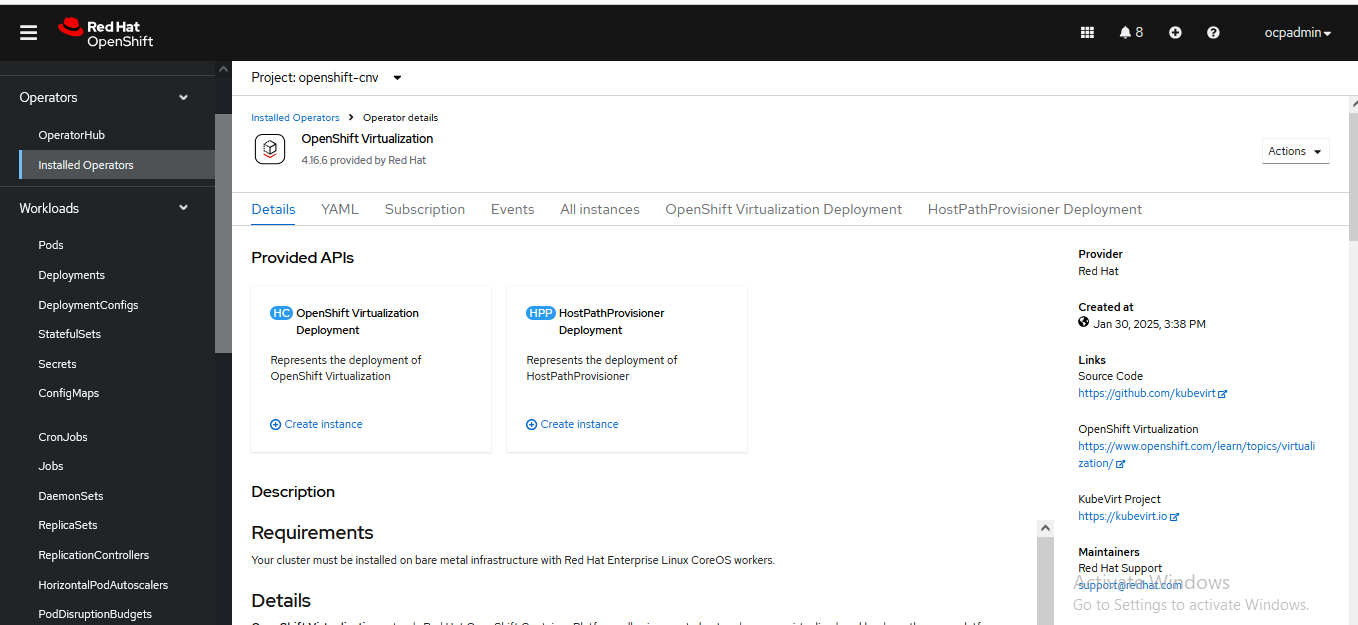
From the Administrator perspective, click Operators → OperatorHub.

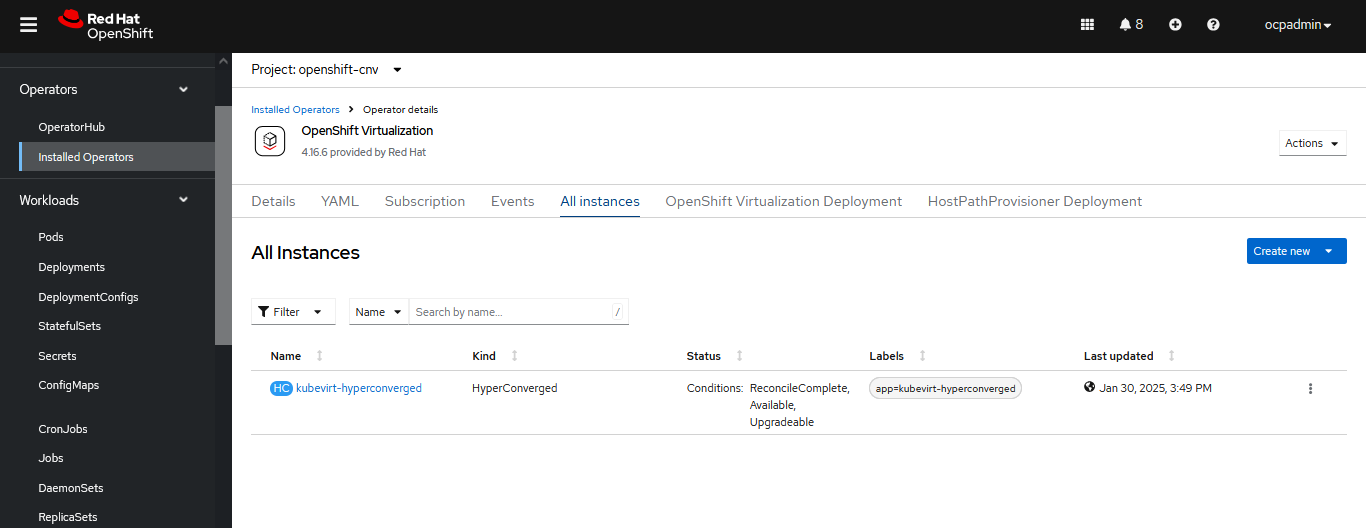
* In the Filter by keyword field, type Virtualization
* Select the OpenShift Virtualization Operator tile with the Red Hat source label
* Read the information about the Operator and click Install.
* On the Install Operator page:
  + Select stable from the list of available Update Channel options. This ensures that you install the version of OpenShift Virtualization that is compatible with your OpenShift Container Platform version.
  + For Installed Namespace, ensure that the Operator recommended namespace option is selected. This installs the Operator in the mandatory openshift-cnv namespace, which is automatically created if it does not exist.
  + For **Approval Strategy**, it is highly recommended that you select **Automatic**, which is the default value, so that OpenShift Virtualization automatically updates when a new version is available in the **stable** update channel.

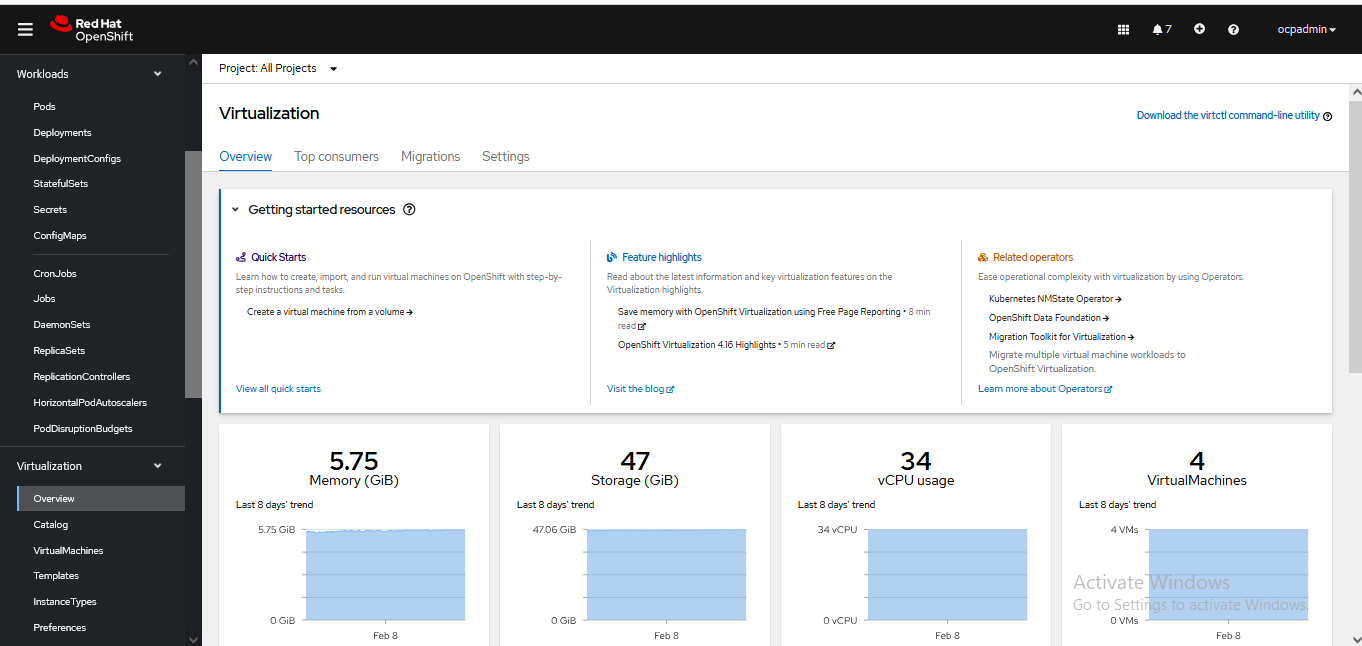
While it is possible to select the **Manual** approval strategy, this is inadvisable because of the high risk that it presents to the supportability and functionality of your cluster. Only select **Manual** if you fully understand these risks and cannot use **Automatic**.

* Click **Install** to make the Operator available to the openshift-cnv namespace.
* When the Operator installs successfully.
* Optional: Configure **Infra** and **Workloads** node placement options for OpenShift Virtualization components.
* Click **Create** to launch OpenShift Virtualization.





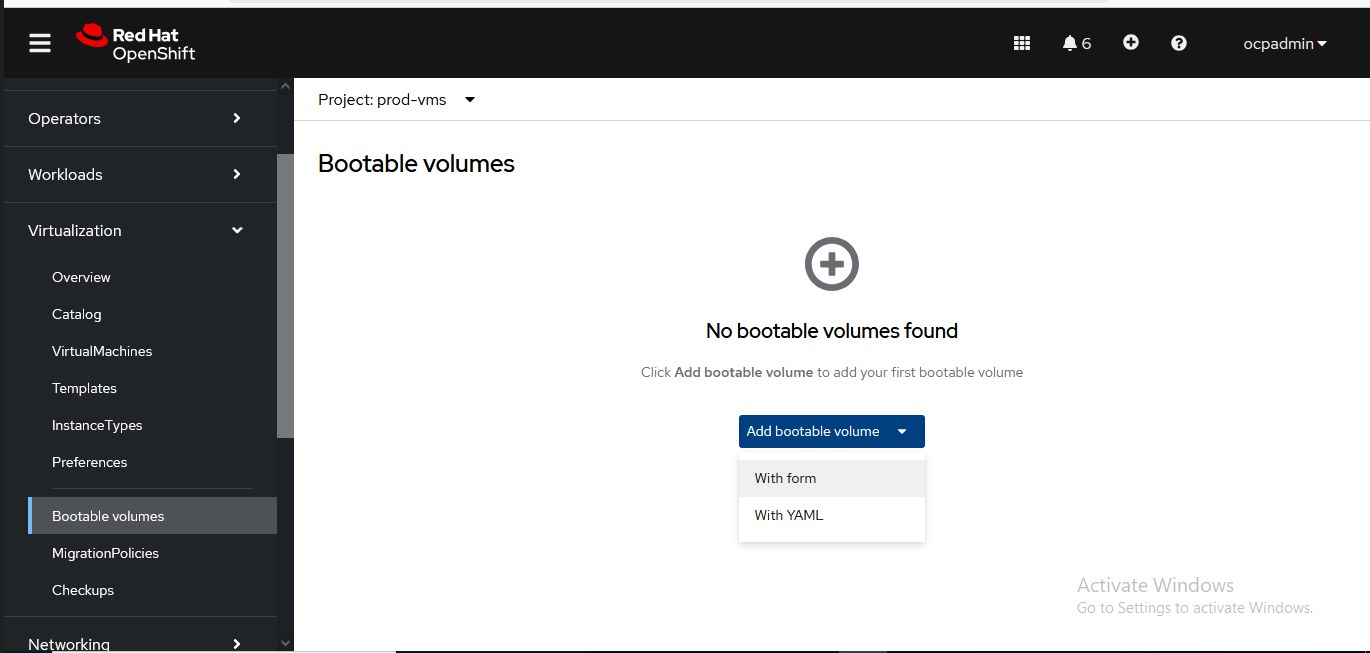


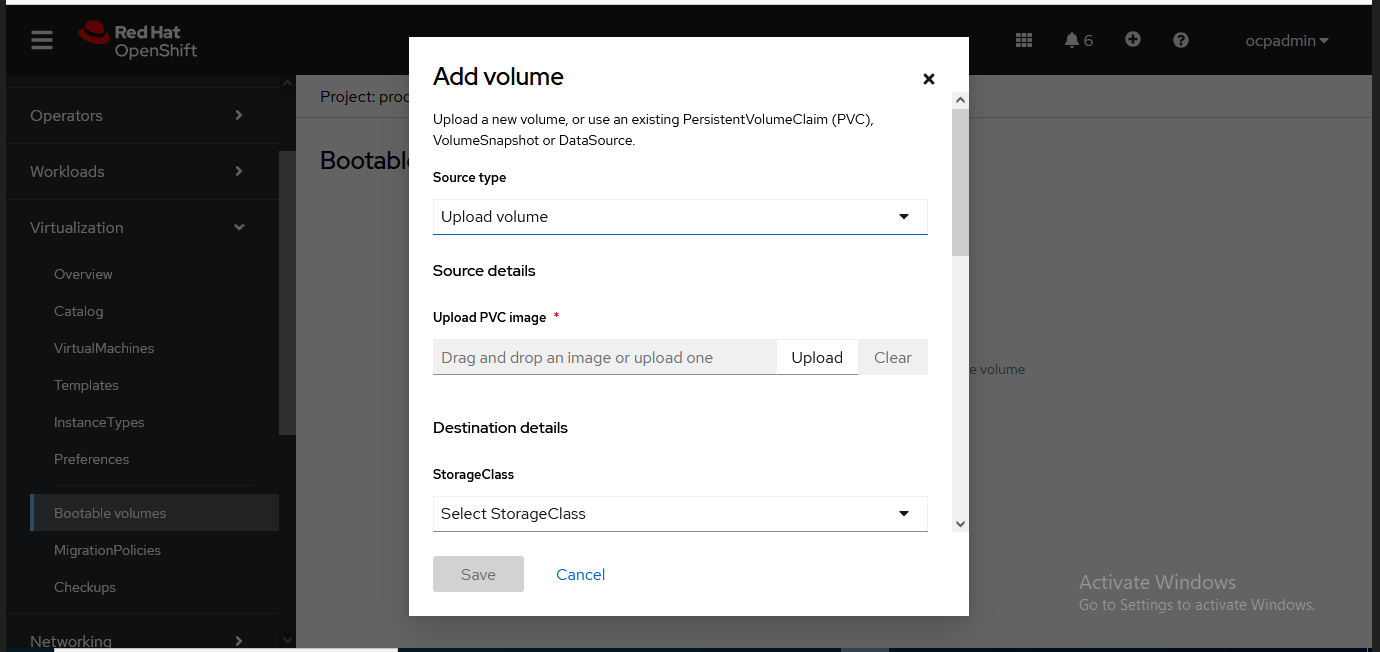


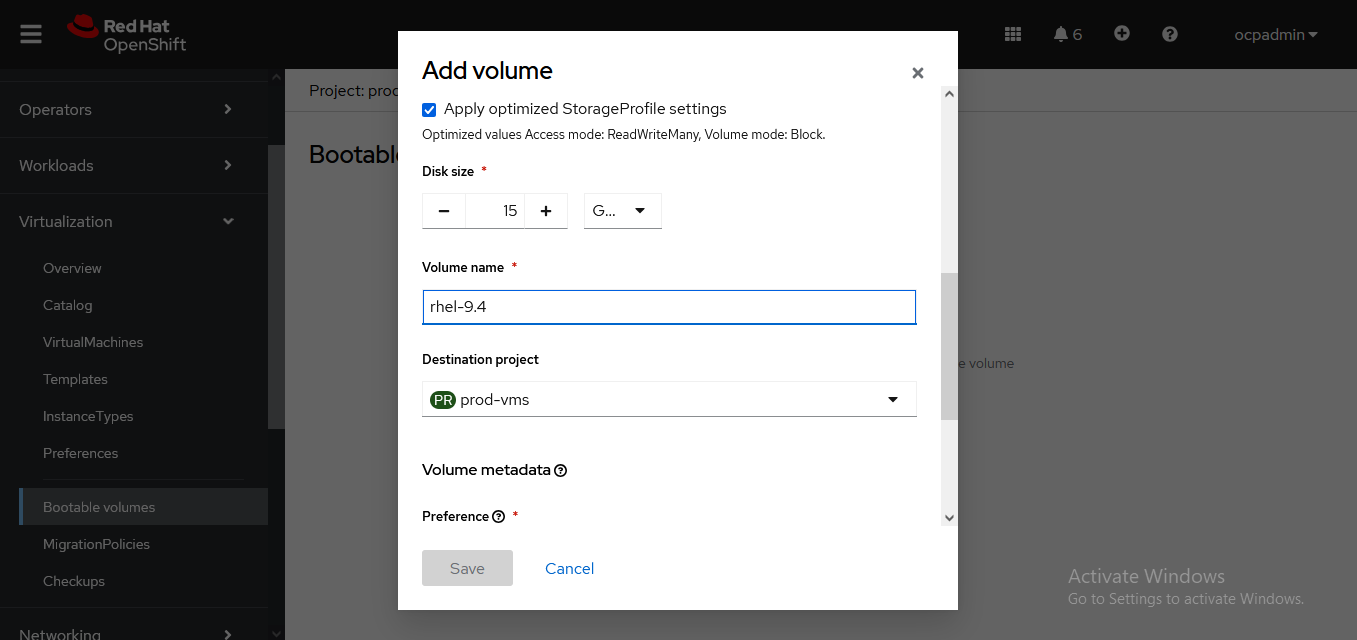
### Adding a Bootable Volume

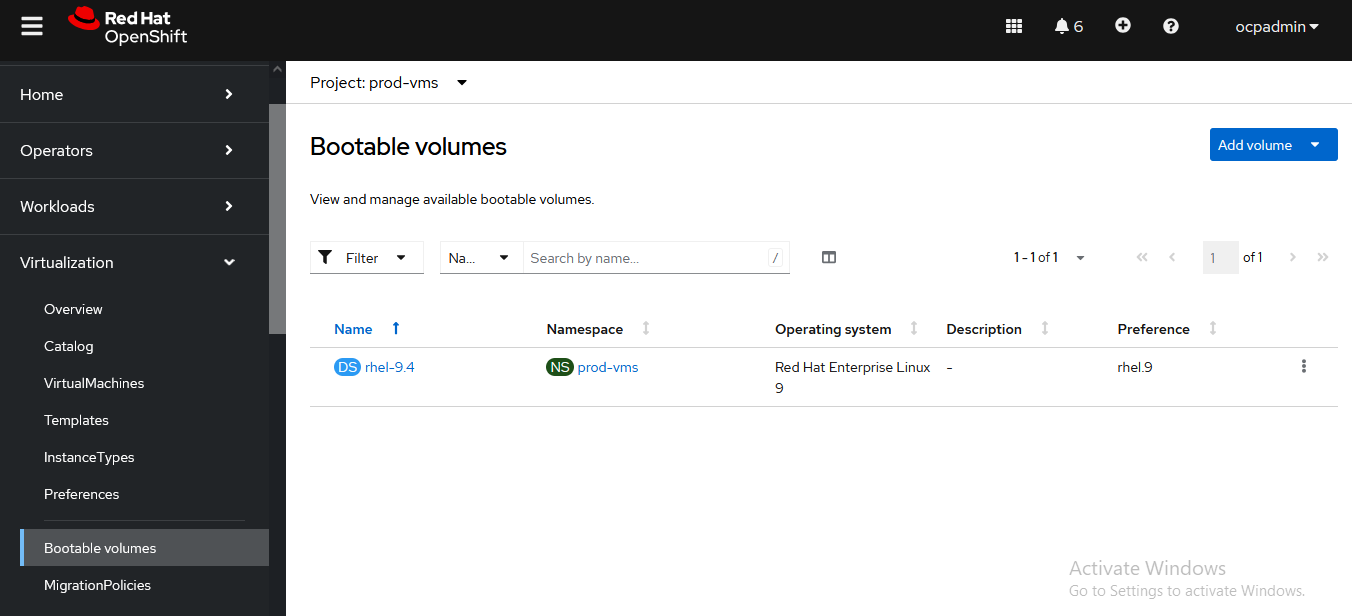
Navigate to the **OpenShift Web Console**.

* Go to **Virtualization → Bootable Volumes**.
* Click **Add Bootable Volume**.
* Provide a **name**, **source type**, select **Storage Class**, and specify the **size**.
* Click **Create** to add the bootable volume.



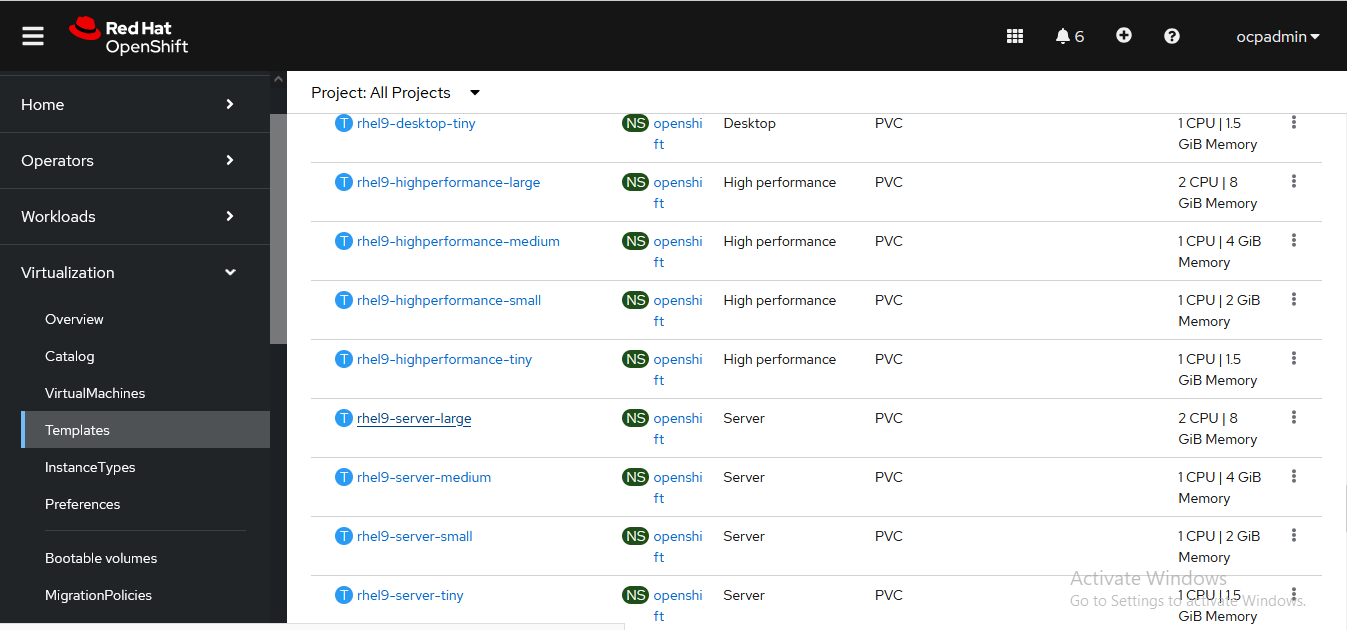


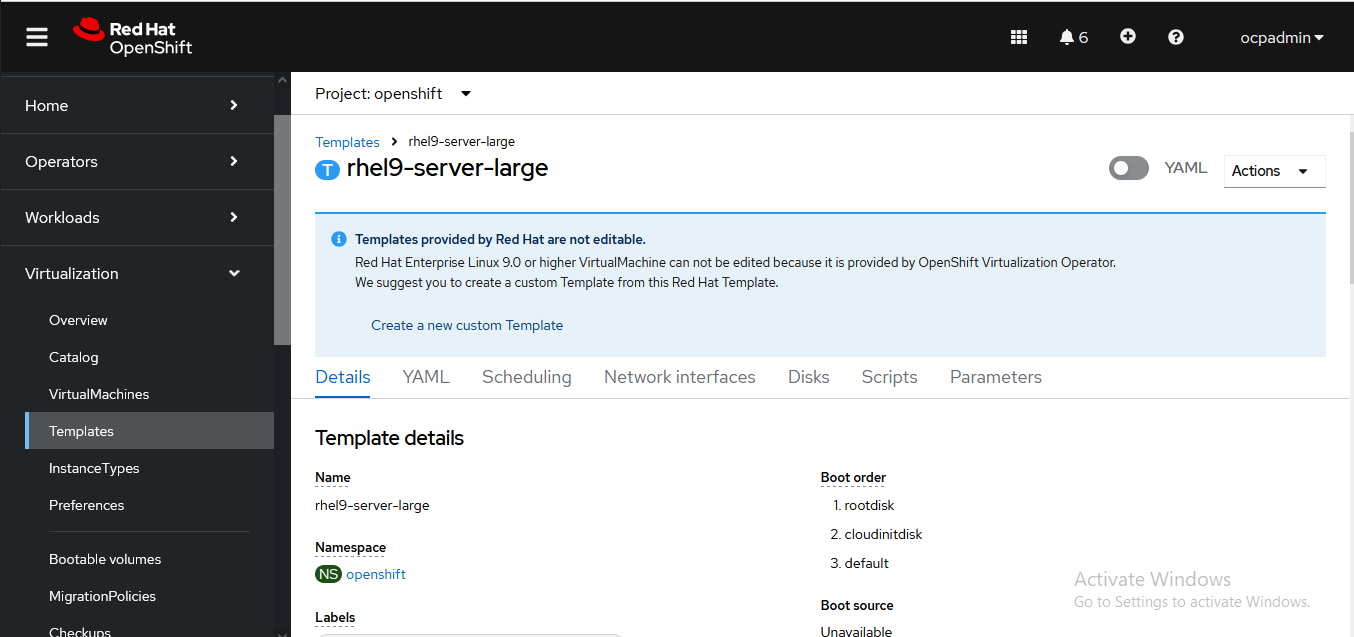


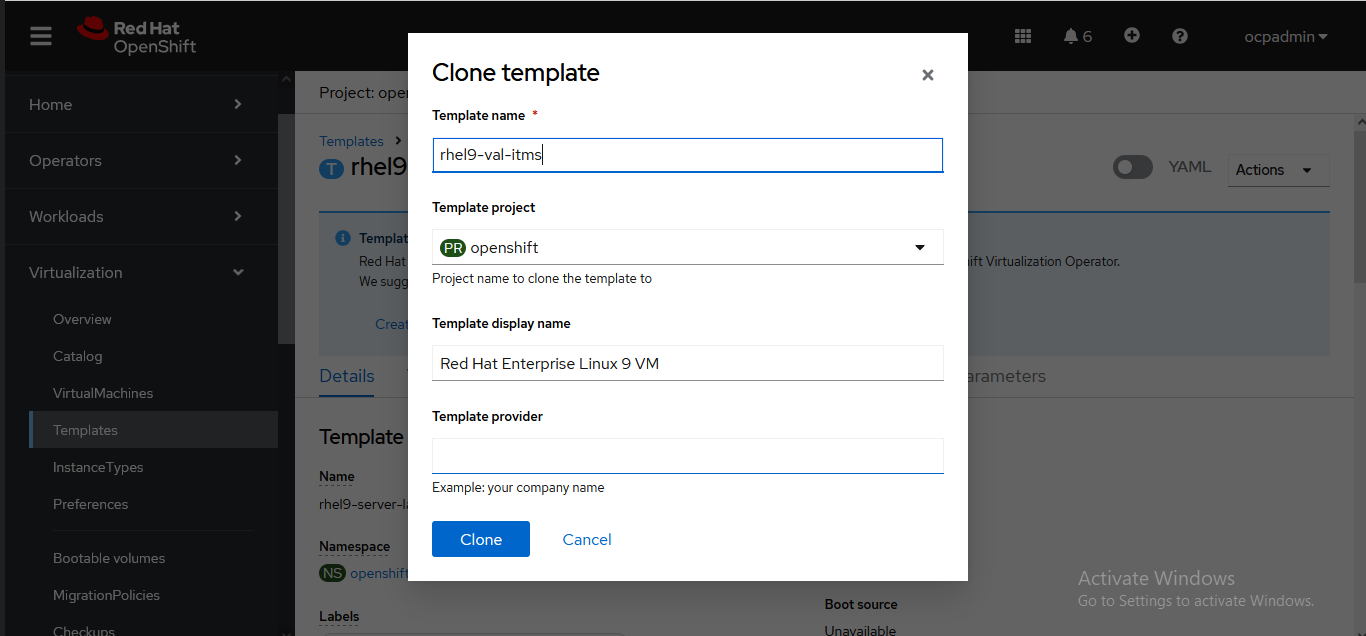


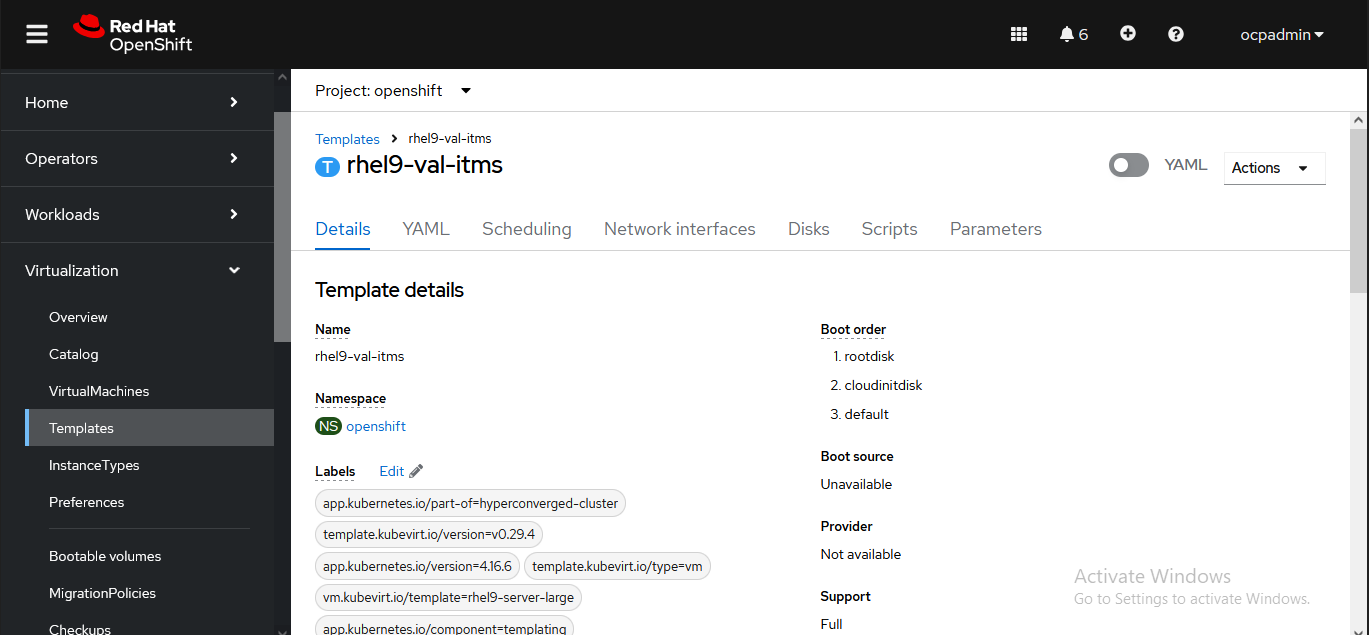
### Clone a ****VirtualMachine**** Template

* Click **Virtualization** → **Templates** from the left-hand menu.
* Find the template you want to clone.
* Click the **⋮ (three dots) menu** next to the template name.
* Select **Clone Template**



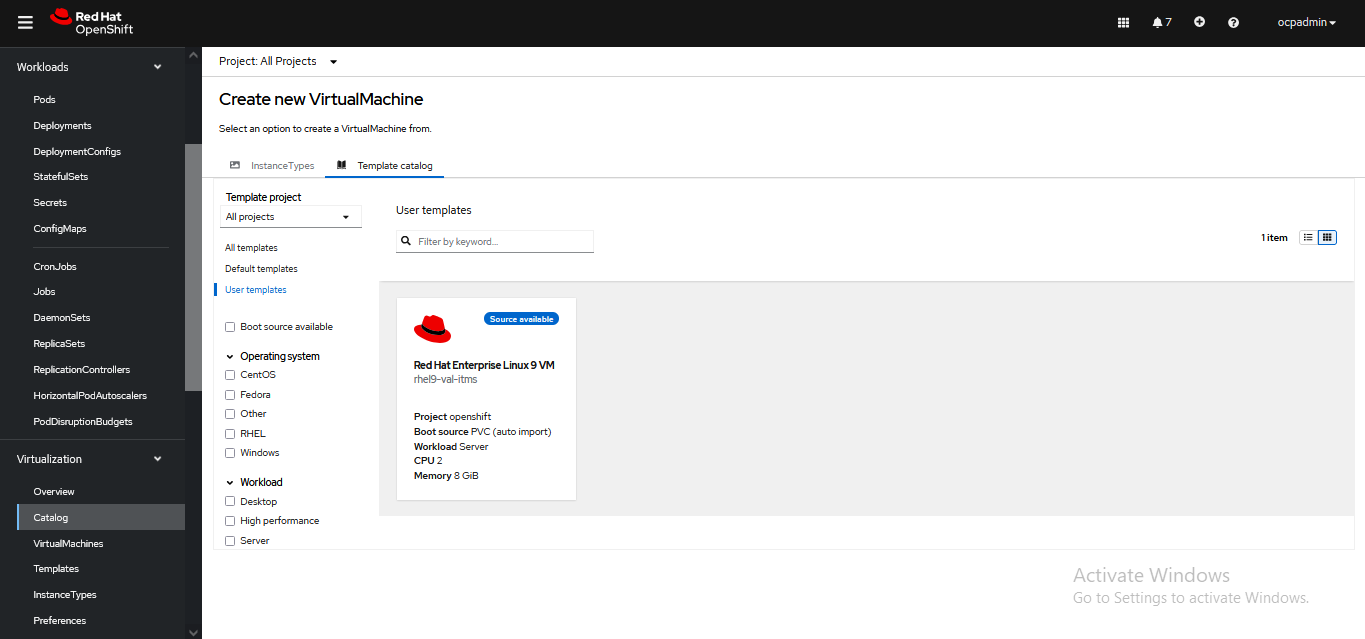


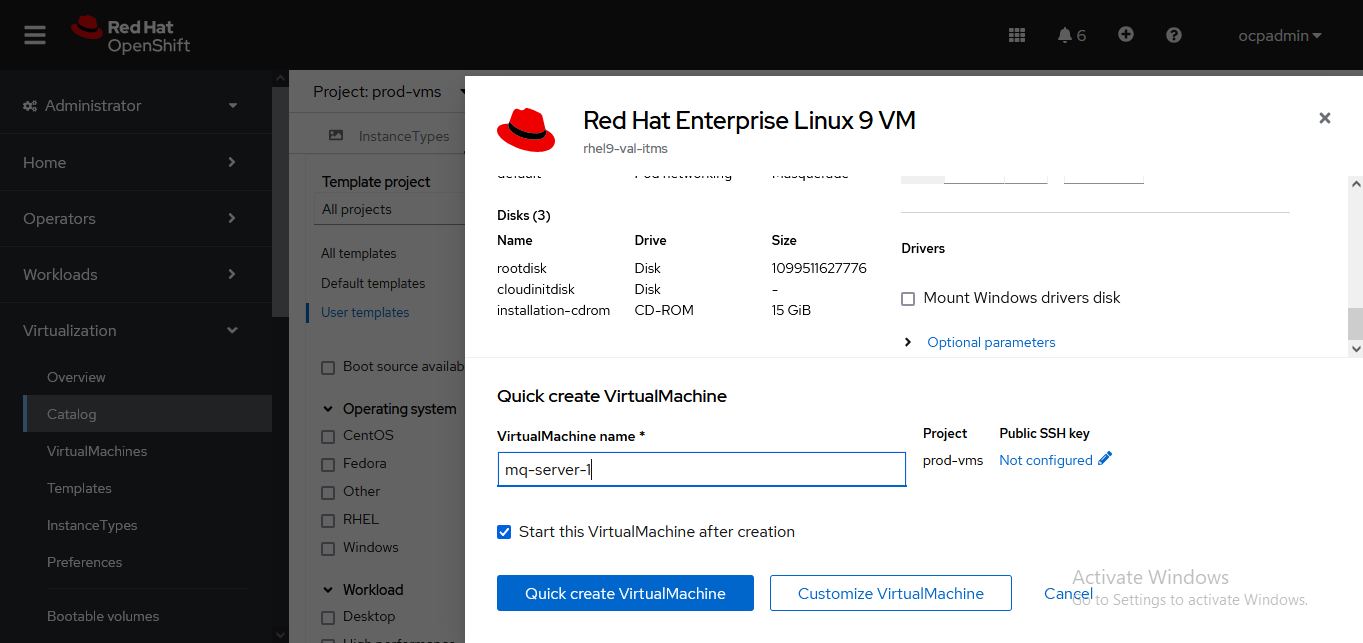


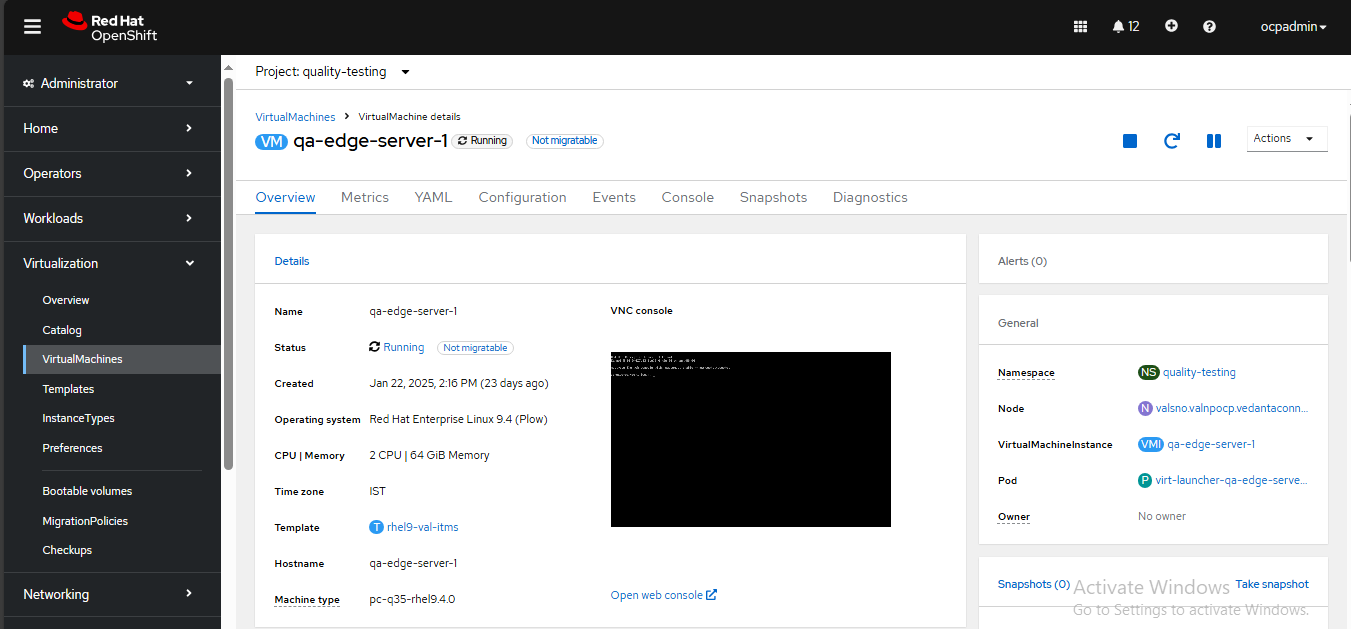


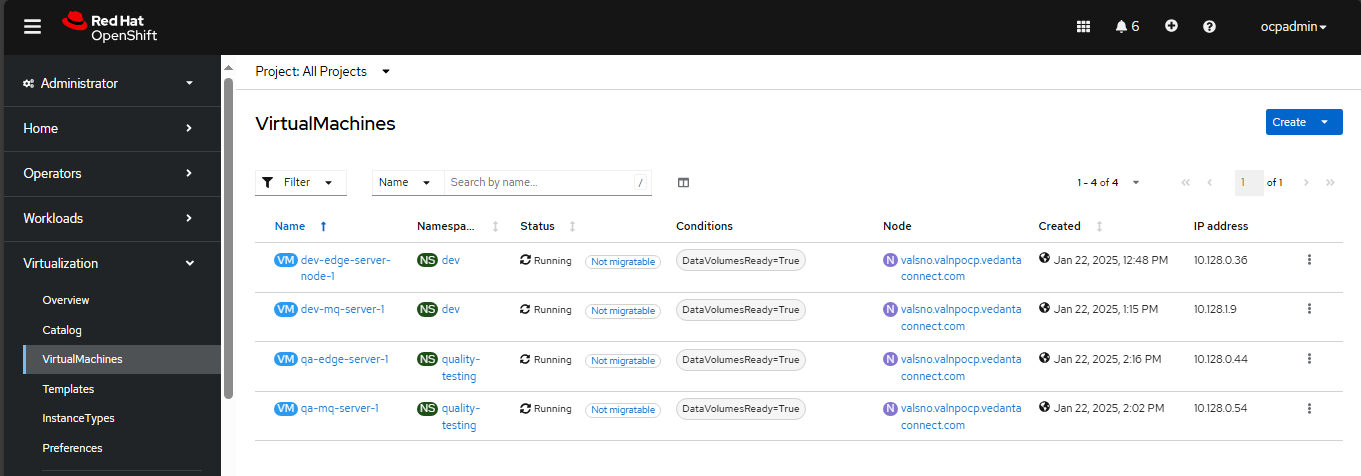
### Create a ****VirtualMachine**** Using the Custom Template

* Navigate to **Virtualization** → **Catalog**.
* Select the **User-Defined Template** from the list.
* Add the **CD ROM,** select the PVC of Bootable Volume **RHEL 9.4**
* Click **Customize** **VirtualMachine**.
* Review and modify the **VM configuration** as needed.
* Click **Create Virtual Machine**.









Note: At the time this document was prepared, Virtual Machines were not accessible from the external network because additional network ports on the server were unavailable, and as a result, external IPs were not assigned to the VMs.

**Thank You**

1. Visit us at www.integramicro.com