## 3.1 Infrastructure Components

### 3.1.1 Kubernetes Infrastructure

Within the OpenShift Container Platform, Kubernetes manages containerized applications across a set of containers or hosts and provides mechanisms for deployment, maintenance, and application-scaling. The container runtime packages, instantiates, and runs containerized applications. The control plane, which is composed of master machines, manages the OpenShift Container Platform cluster. The control plane machines manage workloads running on the worker machines. The cluster itself manages all upgrades to the machines by the actions of the Cluster Version Operator, the Machine Config Operator, and a set of individual Operators.

* Kubernetes API Server: The Kubernetes API server validates and configures the data for Pods, Services, and replication controllers. It also provides a focal point for the cluster's shared state.
* Kube-Scheduler: A scheduler watches for newly created Pods that have no Node assigned. For every Pod that the scheduler discovers, the scheduler becomes responsible for finding the best Node for that Pod to run on.
* Etcd: etcd stores the persistent master state while other components watch etcd for changes to bring themselves into the specified state. Configmaps and Secrets inside the etcd database will be stored on-disk in an encrypted format.
* Controller Manager Server: The Controller Manager Server watches etcd for changes to objects such as replication, namespace, and service account controller objects, and then uses the API to enforce the specified state. Several such processes create a cluster with one active leader at a time.
* The CRI-O container engine (crio), which runs and manages the containers. OpenShift Container Platform 4.14 uses CRI-O instead of the Docker Container Engine.
* Operators: An Operator is a method of packaging, deploying and managing a Kubernetes-native application. A Kubernetes-native application is an application that is both deployed on Kubernetes and managed using the Kubernetes APIs and kubectl tooling.
* Operator Lifecycle Manager (OLM): The Operator Lifecycle Manager helps users install, update, and manage the lifecycle of all Operators and their associated services running across their clusters.

#### 

#### 3.1.1.1 Design decisions

* Red Hat OCP 4.14 cluster to be deployed on BareMetal using the disconnected User Provisioned Installation.
* As per discussion in the design workshop with XXX application and XXX, bare metal OCP deployment has been considered.
* Current OCP 3.11 is running on RHOSP VMs where XXXm application is running.
* XXX to provide F5 Load Balancer required for LB VIPs in L4/Passthrough mode for both API and application ingress traffic.
* Cluster will be deployed at Manesar datacenter.
* XXX to migrate ceph data to netapp before migration etc dedicated disk required baremetal.
* Existing Ceph will not be available in the new cluster. XXX Team to migrate the data from Ceph to Netapp before initiating the migration of workload from OCP3.11 to OCP 4.14.
* New OCP 4.14 cluster name will be provided by XXX to separate DNS URLs.
* There will be 1x bastion (provisioner/helper) machine with the latest RHEL 9.x Operating System. Bastion node will be a virtual machine deployed on RHOSP..
* There will be 1x Mirror Registry machine with the latest RHEL 9.x Operating System. Mirror Registry node will be a virtual machine deployed on a RHOSP. However, XXX showed interest in Red Hat Quay for Centralized Container Registry. XXX will work in parallel with Red Hat Account Team for it’s solution and design. This document doesn’t cover Red Hat Quay. If Quay comes in this environment, the design aspect will be considered separately based on its design workshop.
* Internet connectivity is required on the Registry mirror and will be provided via http proxy by XXX.
* All infra services i.e. monitoring, logging, internal registry, ingress controllers etc will be co-hosted on master nodes. Depending upon workload in future, XXX may need to move out some of these infra services from master nodes to dedicated infra nodes. Initially, as these master nodes have good compute capacity, this decision has been made to co-host infra services.
* RHCOS based 3 x Bare Metal machines will host the controlplane for RHOCP cluster and maintain the high availability of the master and etcd.
* Below mentioned 3 VIP’s[[1]](#footnote-1) will be provided by the XXX team from F5 Enterprise Load Balancer for each cluster.
  + API VIP
  + API-INT VIP
  + \*.apps VIP
* XXX team will provide DNS and NTP servers needed for the OCP environment. It is mandatory to configure forward and reverse lookup in DNS servers for all OCP node’s hostname with its IP address and Load Balancer VIPs aka api and \*.apps.
* XXX will provide 2 X NTP servers where 1 server will be configured as preferred time source. It was explained, 4 X NTP servers are recommended as stated at <https://access.redhat.com/solutions/778603>. XXX understands the risks for using 2 x NTP servers.
* XXX agreed to use self signed OCP Installer generated (\*.apps) wild card certificates to be used for Openshift Ingress Controller. If required, XXX may replace this self-signed wildcard certificate with XXX’s Internal CA signed certificate.
* API will use self signed OCP generated certificates.
* The master API will be available on port 6443. Please refer to section “[5.3 Network Access Requirements](#_heading=h.39kk8xu)” for the complete list of ports that need opening.
* Other infrastructure requirements are covered in the [5. Implementation Prerequisites](#_heading=h.3vac5uf).
* Separate validation prerequisite spreadsheet is being shared with the XXX team. All details need to be provided before starting the deployments.
* OCP comes with OVN-Kubernetes as a default CNI. XXX has no plan to use any thirdparty SDN. Hence default SDN i.e. OVN-Kuvernetes will be used in this cluster deployment.
* XXX wants to integrate Netapp storage to provide persistent storage for applications. During the design workshop discussion, Netapp confirmed, their CSI driver doesn’t support FC backed storage. Hence XXX will ensure, their Storage box gets connectivity with OCP using iSCSI protocol. As iSCSI connectivity is network based, XXX will ensure dedicated NICs to have dedicated storage connectivity using LACP bond.
* XXX to ensure iSCSI network connectivity with OCP nodes is at L2 to avoid any unwanted latency caused by gateways/firewalls etcs.
* Internal registry requires RWX storage. XXX will provide S3 object storage for storing registry images. XXX will provide required details for S3 object storage during implementation and ensure connectivity from OCP nodes to S3 object storage bucket.
* All OCP nodes will have MTU set to 9000, Hence effective MTU at pod level will be 8900 as OVN-Kubernetes SDN uses 1000 bytes as geneve overhead.
* XXX team will provide a sufficient pool for host networks considering the future growth requirements. This network will include IPs for each OCP node. Refer LLD spreadsheet for such details.
* XXX will provide L3 network for OCP environment from the same subnet range for all OCP nodes including bastion, bootstrap.
* Compliance Operator will be deployed.
* Users authentication to be configured using local (HTPasswd based authentication provider).
* RHOCP will be deployed using a dual stack deployment.
* RHOCP will be integrated with Netapp for Block, File and Object storage requirements.
* Netapp Storage integration with RHOCP including CSI Driver installation and respective configurations will be done at OCP and Netapp end by Netapp Team post skeleton OCP cluster deployment.
* Prometheus, Alert manager, Loki log retention period would be 30 days.
* For Etcd backup NFS storage to be provided by XXX.
* Etcd encryption will be configured.
* 4TB storage to be provided by XXX for mirror registry.
* Temporary bootstrap machine will be configured on worker bare metal nodes.

P.S.:- Values used in this LLD used as reference and highlighted, Kindly refer to a separate LLD spreadsheet for this cluster.

### 3.1.2 Container Registry

OpenShift Container Platform can utilize any server implementing the container image registry API as a source of images, including the Docker Hub, private registries run by third parties, and the integrated OpenShift Container Platform registry. Below are the options[[2]](#footnote-2) available:

* Integrated OpenShift Container Platform registry (OCR), OCP internal use, deployed on master nodes.
* Third Party Registries
* Red Hat Quay Registries
* Authentication enabled Red Hat registry

In a scaled/high-availability (HA) OpenShift Container Platform registry cluster deployment:

* The storage technology must support RWX access mode.
* The storage technology must ensure read-after-write consistency.
* The preferred storage technology is object storage.
* Red Hat OpenShift Data Foundation (ODF), Amazon Simple Storage Service (Amazon S3), Google Cloud Storage (GCS), Microsoft Azure Blob Storage, and OpenStack Swift are supported.
* Object storage should be S3 or Swift compliant.
* For non-cloud platforms, such as vSphere and bare metal installations, the only configurable technology is file storage.

#### 3.1.2.1 Key observation/challenges/requirements:

* NA

#### 

#### 3.1.2.2 Design decisions

Registry:

* Registry (OCR) will be implemented in clusters and will run as a container.
* The registry stores container images and metadata. If you simply deploy the registry pod, it uses an ephemeral volume that is destroyed if the pod exits. Any images built or pushed into the registry would disappear.

Hence, based on the available options at XXXs, OpenShift Container Platform registry storage will be provided from Netapp (S3 compliant Object Storage) of size 1024 GiB(initial size, depending on the size of container images stored on it, it may need to be increased during day 2 operations).

* Netapp Object storage will be the backend for registry pods.

### 3.1.3 Web Console

The OpenShift Container Platform web console is a user interface accessible from a web browser. Developers can use the web console to visualize, browse, and manage the contents of projects.

A screenshot of a computer

Description automatically generated

Web console will be deployed by default. System administrators may want to access the master apis via a command line tool called oc. Details are covered in the documentation[[3]](#footnote-3).

#### 3.1.3.1 Design decisions

Below decisions are applicable for all clusters.

1. No use-cases were identified. Default web console will be configured.

## 3.2 Network Considerations

### 3.2.1 OVN-Kubernetes & DNS

The OpenShift Container Platform cluster uses a virtualized network for pod and service networks. The OVN-Kubernetes Container Network Interface (CNI) plug-in is a network provider for the default cluster network. OVN-Kubernetes is based on Open Virtual Network (OVN) and provides an overlay-based networking implementation. A cluster that uses the OVN-Kubernetes network provider also runs Open vSwitch (OVS) on each node. OVN configures OVS on each node to implement the declared network configuration.

|  |
| --- |
|  |

#### 3.2.1.1 Design decisions

* XXX will use the default OVN-Kubernetes plugin.
* RHOCP Cluster will be based on dual stack (IPv4 & IPv6).
* The host network CIDR range for the OCP nodes will be provided by XXX.
* Cluster and Service subnet will be provided by XXX.
* XXX have confirmed that Egress IP may be used, depending on the application requirement.
* All OCP nodes will be reachable to each other. There is no microsegmentation at XXX Network.
* The cluster subnet and service subnet will be internal to OCP cluster however if applications hosted on this cluster need to connect with some external endpoint must not use IP from this cluster subnet and service subnet.

### 3.2.2 Ingress Controllers (Routers)

The Ingress Operator implements the ingress controller API and is the component responsible for enabling external access to OpenShift Container Platform cluster services. The Operator makes this possible by deploying and managing one or more HAProxy-based Ingress Controllers to handle routing. You can use the Ingress Operator to route traffic by specifying OpenShift Container Platform Route and Kubernetes Ingress resources.

OCP offers multiple ways in which HAProxy can be used via route objects. Please find the key features/configuration in documentation[[4]](#footnote-4).

#### 3.2.2.1 Design decisions & Recommendations

Below decisions are applicable for the cluster.

* 3 router pods will be deployed on the master nodes.
* RHOCP router is expected to work with HTTP/HTTPS protocol only.
* By default; router will expose the application over port 443 for https traffic and port 80 for http.
* The OCP router will be responsible to route the traffic to the pods as per the workflow.

#### 3.2.2.2 Application flow

1. VIP will be configured to pass the traffic to the router pods.
2. Applications can utilize the same VIP and expose the application on intranet. Like <https://sample-application.apps.devocp.example.local>
3. XXX application team may have different flow and that details will be documented under XXX approach document e.g. they might want to create their custom LB or flows to enable migration from OCP 3.11 to 4.14.

### 3.2.3 Load Balancer

XXX confirmed installation method would be UPI, below the load balancer requirement.

Note:

The load balancer must be configured to take a maximum of 30 seconds from the time the API server turns off the /readyz endpoint to the removal of the API server instance from the pool. Within the time frame after /readyz returns an error or becomes healthy, the endpoint must have been removed or added. Probing every 5 or 10 seconds, with two successful requests to become healthy and three to become unhealthy, are well-tested values.

#### 

#### 

#### 3.2.3.1 API load balancer

XXX would configure F5 load balancer for all external API access as below.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| API endpoint | Port | VIP | ssl mode | Backend server | Comments |
| api.<TBF> | 6443 |  | passthrough | bootstrap  master 1  master 2  master 3 | bootstrap is temporary and it has to be removed once the cluster is built. |

Note:

Session persistence is not required for the API load balancer to function properly.

#### 3.2.3.2 API-INT load balancer

XXX would configure F5 load balancer for all internal API access for the cluster below.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| API endpoint | Port | VIP | ssl mode | Backend server | Comments |
| api-int.<TBF> | 22623 and 6443 |  | passthrough | bootstrap  master 1  master 2  master 3 | bootstrap is temporary and it has to be removed once the cluster is built. |

Note:

Session persistence is not required for the API load balancer to function properly.

#### 3.2.3.3 \*.APPS load balancer

XXX would configure F5 load balancer for all application ingress access as below.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| API endpoint | Port | VIP | ssl mode | Backend server | Remarks |
| \*.apps.<TBF> | 443 and 80 |  | passthrough | master 1  master 2  master 3 | As Router pods will be co-hosted on masters. If customer decides to move router modes from master to dedicate nodes, these IPs need to be updated on respective LB |

Note:

* If the true IP address of the client can be seen by the application Ingress load balancer, enabling source IP-based session persistence can improve performance for applications that use end-to-end TLS encryption.

## 3.3 Prometheus Cluster Monitoring

OpenShift Container Platform ships with a pre-configured and self-updating monitoring stack that is based on the Prometheus open source project and its wider ecosystem. It provides monitoring of cluster components and ships with a set of alerts to immediately notify the cluster administrator about any occurring problems and a set of OCP monitoring dashboards.

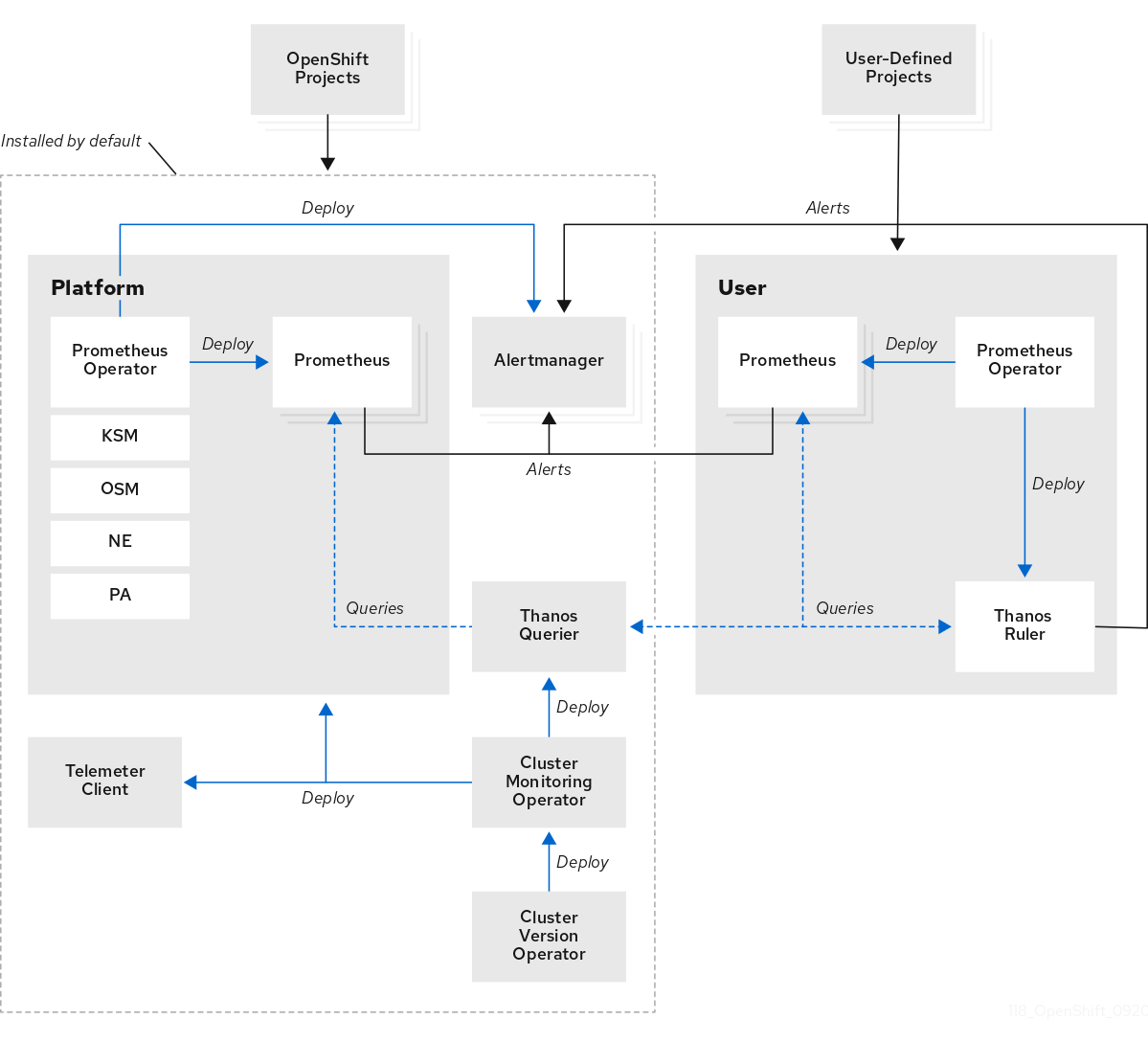
You also have the option to enable monitoring for user-defined projects.

A cluster administrator can configure the monitoring stack with the supported configurations. OpenShift Container Platform delivers monitoring best practices out of the box.

A set of alerts are included by default that immediately notify administrators about issues with a cluster. Default dashboards in the OpenShift Container Platform web console include visual representations of cluster metrics to help you to quickly understand the state of your cluster. With the OpenShift Container Platform web console, you can view and manage metrics, alerts, and review monitoring dashboards.

In the Observe section of OpenShift Container Platform web console, you can access and manage monitoring features such as [metrics](https://docs.openshift.com/container-platform/4.11/monitoring/managing-metrics.html#managing-metrics), [alerts](https://docs.openshift.com/container-platform/4.11/monitoring/managing-alerts.html#managing-alerts), [monitoring dashboards](https://docs.openshift.com/container-platform/4.11/monitoring/reviewing-monitoring-dashboards.html#reviewing-monitoring-dashboards), and [metrics targets](https://docs.openshift.com/container-platform/4.11/monitoring/managing-metrics-targets.html#managing-metrics-targets).

After installing OpenShift Container Platform, cluster administrators can optionally enable monitoring for user-defined projects. By using this feature, cluster administrators, developers, and other users can specify how services and pods are monitored in their own projects. As a cluster administrator, you can find answers to common problems such as user metrics unavailability and high consumption of disk space by Prometheus in [Troubleshooting monitoring issues](https://docs.openshift.com/container-platform/4.11/monitoring/troubleshooting-monitoring-issues.html#troubleshooting-monitoring-issues).



In addition to the components of the stack itself, the monitoring stack monitors:

* CoreDNS
* LokiStack (if Logging is installed)
* Etcd
* Vector (if Logging is installed)
* HAProxy
* Image registry
* Kubelets
* Kubernetes apiserver
* Kubernetes controller manager
* Kubernetes scheduler
* Metering (if Metering is installed)
* OpenShift apiserver
* OpenShift controller manager
* Operator Lifecycle Manager (OLM)

### 3.3.1 Key observation/challenges/requirements

1. The Monitoring(Prometheus+AlertManager) component will be deployed on infra nodes, one pod on each node.
2. Grafana is deprecated in OCP 4.10 and removed from the 4.11 OCP version. As part of 4.14 OCP Monitoring dashboards are an integrated part of the OCP web console itself.

### 3.3.2 Design Decisions

1. OCP Cluster monitoring will be installed[[5]](#footnote-5); Monitoring stack will be colocated with master nodes only.
2. Persistence block storage for both prometheus and alert manager will be enabled and be given from Netapp. Depending upon usage, this size will be increased during day 2 ops.
3. The monitoring data will be stored for 30 days and block storage will be used to store the data.
4. Cluster/Namespace/Pod level monitoring dashboards are available by default in the OCP console.
5. Default alert rules will be installed.
6. Alert Manager provides various standard ways to send out notifications like email, chat platforms, webhooks etc.

### 

### 3.3.3 Storage Sizing

Based on the application workload details shared (covered in the [Design Workshop section](#_heading=h.qsh70q)) and the number of pods and nodes are calculated; below are the storage sizing:

Prometheus runs with 2 pods by default hence 2 PVs required. Red Hat recommends to run with 2 prometheus spread across 2 VMs in production for HA. Below is the calculation for prometheus storage , considering the monitoring solution for 30 days data :

|  |
| --- |
| Monitoring:  Per day Storage =~16 Gb  No of days to retain = 30 days  Total Storage required=~ 2 X 500 GB (for 2 Prometheus pods)  Alert manager runs with 2 pods by default.  Alert manager storage = 20 Gb per pod  Storage required = ~2 \* 20GB  \*\*\* assumption: prometheus pods are well distributed across nodes. This is just an assumption calculation, Red Hat recommends to calculate the exact values during the performance testing. |

*Please note; if the replica set is increased then the storage too needs to be adjusted and provisioned.*

*Note: This section is completed based on Assumption. Subject to change by the XXX team post go live if required.*

## 

## 3.4 Log Aggregation Solution

========================================

The logging subsystem for Red Hat OpenShift is an opinionated collector and normalizer of application, infrastructure, and audit logs. It is intended to be used for forwarding logs to various supported systems.

The logging subsystem for Red Hat OpenShift is not:

* A high scale log collection system
* Security Information and Event Monitoring (SIEM) compliant
* Historical or long term log retention or storage
* A guaranteed log sink
* Secure storage - audit logs are not stored by default

You can use an internal Loki or Elasticsearch log store on your cluster for storing logs, or you can use a ClusterLogForwarder custom resource (CR) to forward logs to an external store. XXX-XXX application team wants to use their own logging solution for application logs

* Loki is a horizontally scalable, highly available, multi-tenant log aggregation system offered as an alternative to Elasticsearch as a log store for the logging.
* Elasticsearch indexes incoming log records completely during ingestion. Loki only indexes a few fixed labels during ingestion and defers more complex parsing until after the logs have been stored. This means Loki can collect logs more quickly.

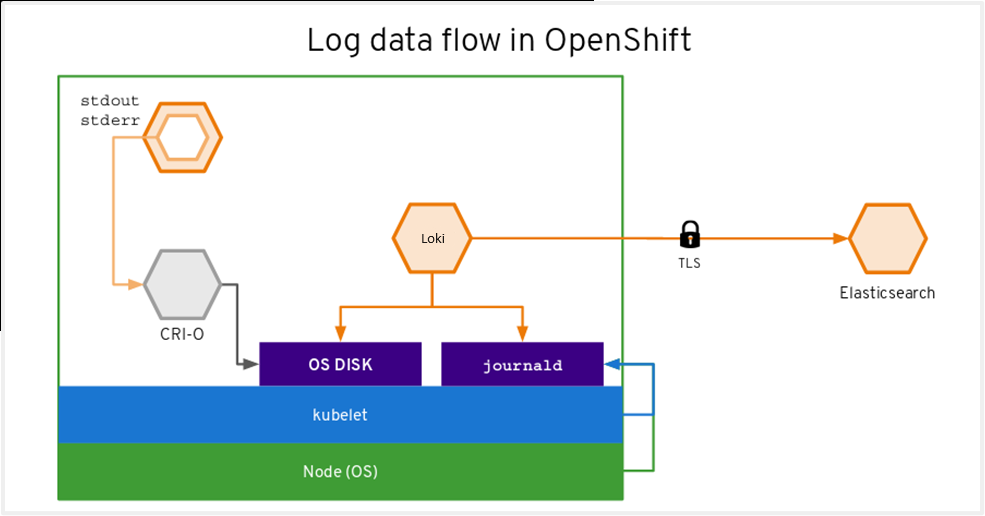
### 3.4.1. Key observation/challenges/requirements:

1. The OpenShift Elasticsearch Operator is deprecated and is planned to be removed in a future release. Red Hat provides bug fixes and support for this feature during the current release lifecycle, but this feature no longer receives enhancements. As an alternative to using the OpenShift Elasticsearch Operator to manage the default log storage, Loki Operator can be used. Hence the customer has decided to go for a Loki Operator based Logging solution.
2. Fluentd is deprecated and is planned to be removed in a future release. Red Hat provides bug fixes and support for this feature during the current release lifecycle, but this feature no longer receives enhancements. As an alternative to Fluentd, Vector will be used in XXX-XXX OCP Platform.
3. The logging subsystem Loki instance is optimized and tested for short-term storage. If you want to retain your logs over a longer term, it is recommended you move the data to a third-party storage system. Red Hat recommends a retention of 7 days. XXX-XXX team has decided to use logging operators only for infra and audit logs with a retention period of 10 days

### 3.4.2. Design Decisions

1. The Loki stack that comes out of the box from OCP will be deployed on master nodes. For more detail please refer to the Red Hat documentation[[6]](#footnote-6).
2. Loki stack needs to be configured only for infra and audit logs.
3. Vector will forward the logs to internal Loki storage
4. Vector configuration for 1x.large[[7]](#footnote-7) will be used and the operator will use the resource as per the specification.
5. Persistent Volumes required for these Openshift Logging should be Netapp object storage with 1TB allocation
6. The threshold of 70% storage consumption has been considered for storage calculation as a best practice. Please refer to Red Hat documentation[[8]](#footnote-8) for more detail.

========================================



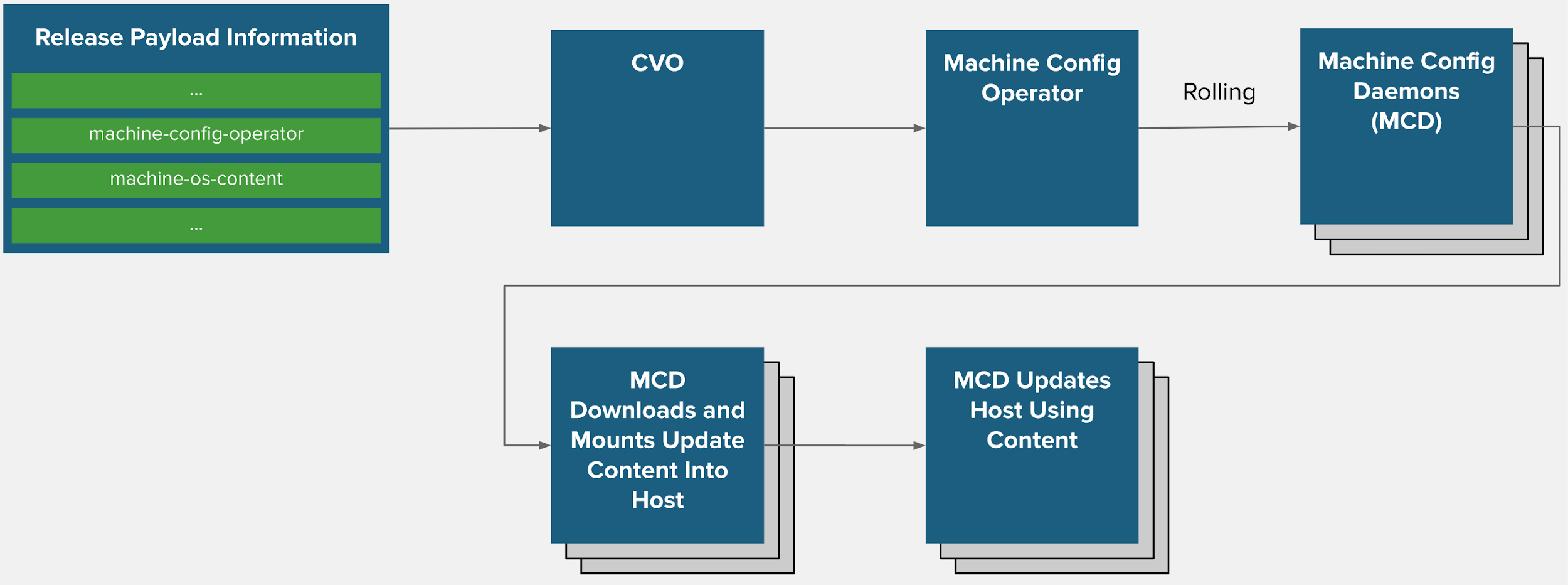
## 3.5 List of Supported Components and versions

OCP 4.14 architecture and design for XXX has been carried out keeping the support matrix in mind. The detailed support matrix is available at Red Hat Customer Portal[[9]](#footnote-9).

## 3.6 Other considerations

### 3.6.1 Upgrading OCP Cluster

The OpenShift Container Platform update service is a hosted service that provides over-the-air updates to both OpenShift Container Platform and Red Hat Enterprise Linux CoreOS (RHCOS). The Cluster Version Operator (CVO) in your cluster checks with the OpenShift Container Platform update service to see the valid updates and update paths based on current component versions and information in the graph. As an example; below is a representation of RHCOS based master machines. For more details refer to Red Hat documentation[[10]](#footnote-10).



#### 

#### 3.6.1.1 Design Decisions

1. Clients can use both CLI and Web console to update the cluster.

## 3.7 Sizing Considerations

OpenShift cluster limits are well documented at Red Hat customer portal[[11]](#footnote-11) and can be used as reference. Based on the application details received, below is the calculation for the environment.

To plan for a smooth application execution; a healthy environment for all the pods needs to be planned. In the cloud native application development, nature of application may also decide high availability of the application e.g. whether app is 12 factor[[12]](#footnote-12) complaint or not. To discover the number of nodes required; distribute pods evenly across clusters using the below formula.

|  |
| --- |
| Maximum Pods per Cluster / Expected Pods per Node = Total Number of Nodes |

Please note that

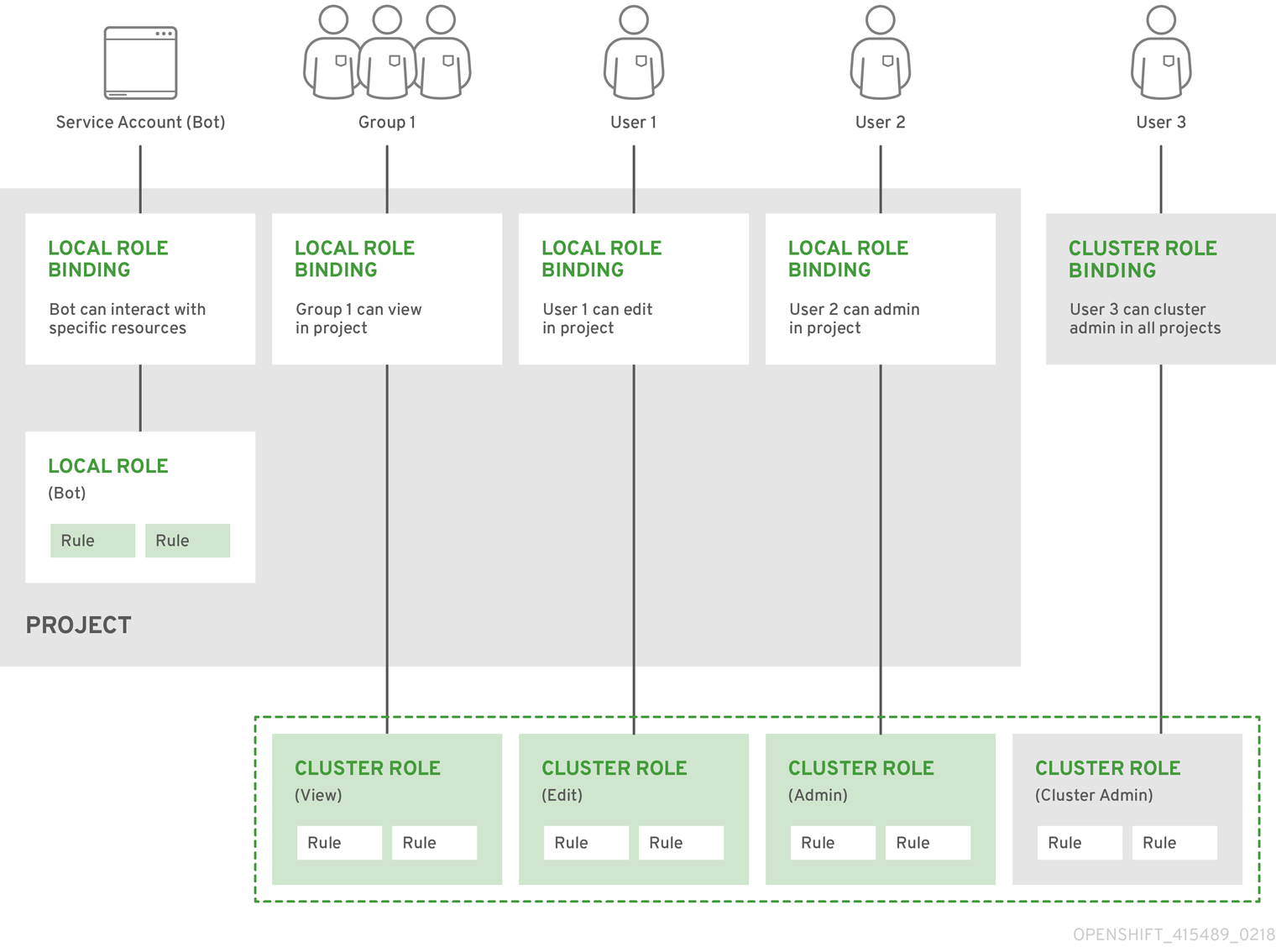
* Number of pods accommodates headroom for the growing number of pods. None-the-less the values may change based on real-life metrics.
* “Maximum Pods per Cluster” needs to consider all the transient pods like build pods and deploy pods etc. Multiple of all those can co-exist at the same time. These numbers are evolving and should be adjusted over time based on the OCP PaaS platform.
* “Expected pods per node” is too configurable which can go up to 250. There are many factors to consider to conclude application pods per node like memory footprint of the container/pod, amount of CPU demand of application etc.

## 3.8 Authentication, Authorisation & Security

Authentication: The OpenShift Container Platform master includes a built-in OAuth server. Developers and administrators obtain OAuth access tokens to authenticate themselves to the API. OAuth using the master configuration, can be configured, file to specify an identity provider. It is a best practice to configure your identity provider during cluster installation, but this can be configured after installation too. There are multiple identity providers like:

* AllowAllPasswordIdentityProvider
* DenyAllPasswordIdentityProvider
* HTPasswdPasswordIdentityProvider
* LDAPPasswordIdentityProvider
* GitHub and GitHub Enterprise
* GoogleIdentityProvider
* OpenIDIdentityProvider etc

Authorization: Role-based Access Control (RBAC) objects determine whether a user is allowed to perform a given action within a project. It allows developers to use local roles and bindings to control who has access to their projects. Authorisation is managed using Rules, Roles, Bindings. The relationships between cluster roles, local roles, cluster role bindings, local role bindings, users, groups and service accounts are illustrated below.



In addition to the RBAC resources that control what a user can do, OpenShift Container Platform provides security context constraints (SCC) that control the actions that a pod can perform and what it has the ability to access. Administrators can manage SCCs using the CLI. SCCs are also very useful for managing access to persistent storage.

For more details, please refer to the documentation[[13]](#footnote-13).

### 3.8.1 Key observation/challenges/requirements

1. The XXX team will authenticate the cluster via HTPasswd.

### 3.8.2 Design Decisions

1. Token, grant and session configuration options will be left to the defaults. This can be reconfigured post installation as per need.
2. User creation for the OCP cluster will be carried out based on the application project needs, post installation.
3. Default SCC will not be modified as changes could lead to issues when upgrading OCP. Administrators can create new SCC based on the project's needs, post installation. For more detail refer to documentation[[14]](#footnote-14).
4. OpenShift Container Platform creates a cluster administrator, kubeadmin, after the installation process completes. After an identity provider is defined and a new cluster-admin user is created , the kubeadmin user will be removed to improve cluster security.
5. Red Hat Compliance Operator will be installed to enforce OCP CIS profile. The Compliance Operator uses OpenSCAP, a NIST-certified tool, to scan and enforce security policies provided by the content.
6. No custom bundle profile required.
7. Block Storage of 50 Gb is required. Which will be provided from block storage.

# 

# 4. OpenShift High Availability

Master nodes implement control-plane infrastructure management. Three control-plane nodes establish a unified control plane for the operation of an OpenShift cluster. The control plane operates outside the application container workloads and is responsible for ensuring the overall continued viability, health, availability, and integrity of the container ecosystem.

Each master node will have the following control components:

* API Server
* ETCD
* Controller Manager
* Scheduler

The table and figure below depict the HA role matrix of OpenShift control components:

|  |  |  |
| --- | --- | --- |
| Role | Style | Notes |
| API Server | Active-Active | Balanced by a virtual IP mechanism using an external F5 load balancer. |
| etcd | Active-Active | Fully redundant deployment with 3 nodes, 1 as leader and others nodes as followers. |
| Controller Manager | Active-Passive | Fully redundant deployment with 3 nodes, 1 as leader and others nodes as followers. |
| Scheduler | Active-Passive | Fully redundant deployment with 3 nodes, 1 as leader and others nodes as followers. |

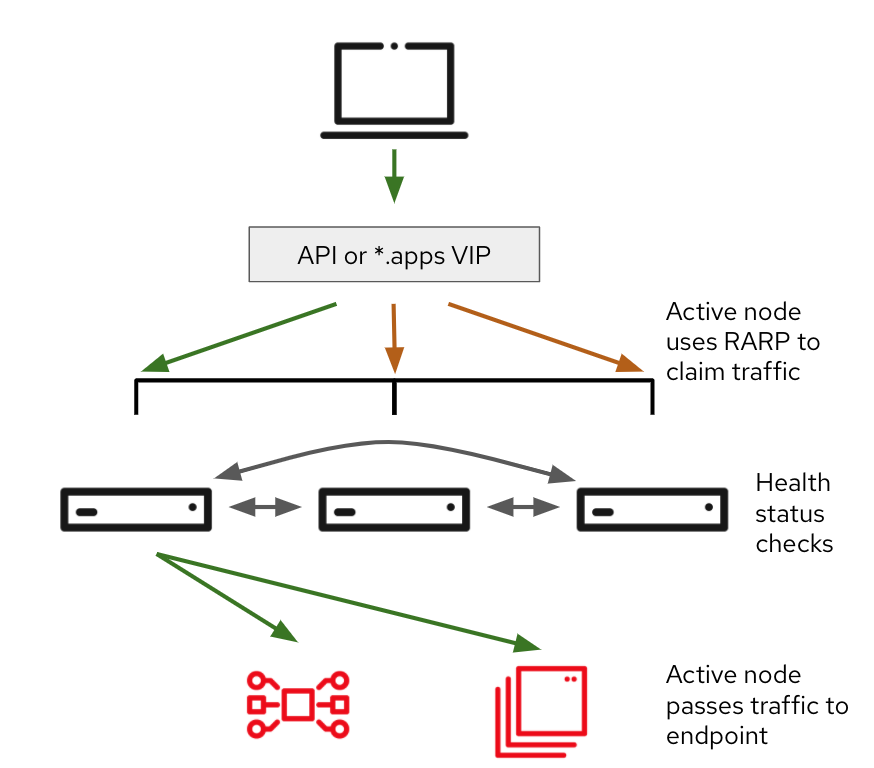
## 

## 4.1 ETCD HA Design

Etcd is the distributed data store where all the API objects will be kept, to make it highly available the architecture includes three instances of etcd, one on each master node. Each etcd instance will have the IP address and port number of other instances, which means that every etcd instance will have information of other two instances. Etcd will replicate data across all instances. In case, If one instance failed or disconnected other two nodes would maintain the majority and can accept the cluster state changes and would be able to communicate with the clients (api server). Once the failed node comes back it will be updated with the latest information by the other instances.

## 4.2 API HA Design

The API server would remain stateless, although it does cache etcd. The architecture proposes to run 3 instances of API server, one on each master node to make it accessible through API VIP using keepalived.



* The VRRP protocol is used by Keepalived to determine the node health and elect an IP owner
* The node health are checked every second for each service (separated checks, one for API and another for INGRESS)
* The ARP is used to associate the VIP with the owner node’s interface (masters)
* Active Node uses RARP (Reverse ARP) to claim traffic
* Active node passes traffic to endpoint
* All controlplane nodes will be hosted on different racks to obtain rack level redundancy.

## 4.3 Scheduler and controller HA design

As Scheduler and controller actively watch the cluster state and act quickly when it changes, running multiple active instances of scheduler and controller could cause race conditions and affect the performance. For this reason, when running multiple instances of these components, only one instance may be active at any given time.

The controller and scheduler take care of this by following the leader-election procedure. The instance will only be active when it’s the elected leader. Only the leader performs actual work, whereas all other instances are standing by and waiting for the current leader to fail. When it does, the remaining instances elect a new leader, which then takes over the work.

This mechanism ensures that two components are never operating at the same time and doing the same work. The controller and scheduler would run collocated with API and etcd.

## 

## 4.4 Network connection redundancy

* Network decisions will be updated once deployment approach is finalized.

## 

# 5. Implementation Prerequisites

XXX needs to provision the system prerequisites for all the OCP 4.14 cluster deployment. The design is based on the joint workshop conducted. Please refer to the design section for rationale around the system & environment requirements.

## 5.1 Red Hat subscriptions

XXX must have an active Red Hat subscription on the Red Hat account. Subscription details can be seen in the management section of Red Hat customer portal13.

Also, supporting infrastructure for the cluster should be a part of the Tested Integration[[15]](#footnote-15).

## 5.2 Hardware requirements

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Node Type | Node | OS | CPU per  Node | RAM per  Node | Root disk per  Node (GB) | NIC Ports per Node |
| Master Node | 3 | RHCOS | 64 | 1.5 TB | 960 | 4 |
| Workers | 5 | RHCOS | 64 | 1.5 TB | 960 | 4 |
| Bootstrap(VM) | 1 | RHCOS | 64 | 1.5 TB | 960 | 4 |
| Bastion(VM) | 1 | RHEL 9.x | 4 | 8 GB | 300 | 1 |
| Mirror registry (VM)\* | 1 | RHEL 9.x | 4 | 16 GB | 300 | 1 |

* Additional 4 TB disk to be attached to the mirror registry for hosting container images

Please note:

1. RHCOS installation for Bootstrap, Control plane, Infra & Worker nodes will be done by UPI method.

## 5.3 Network Access Requirements

A shared network must exist between the master and other nodes.

Key activities are:

* Subnets Required:
  1. Hosts Subnet (10.240.68.0/23 and X:X:X:X::X:X:X:X/119)

The subnet used on the primary interfaces of the hosts that OpenShift will be installed on.

WARNING: must not conflict with:

* Pod IP Subnet
* Services IP Subnet
  1. Pod Subnet (10.170.0.0/14 and fd02::/48)

/14 CIDR IP Range used for SDN. This should be big enough to cover number of pods you may want to deploy

WARNING: must not conflict with:

* Services IP Subnet
* Network Bridge IP Subnet
* Hosts IP Subnet
* Any other CIDR in customer's datacenters
  1. Services Subnet (172.32.0.0/16 and fd03::/112)

/16 CIDR IP Range used for SDN

WARNING: must not conflict with

* Pod IP Subnet
* Network Bridge IP Subnet
* Hosts IP Subnet
* Any other CIDR in customer's datacenters

d. Storage Subnet (192.168.90.0/23 and fc00:0:0:7::0/119)

The subnet used on the secondary interfaces of the hosts for the storage network. Subnet should be the same as the host subnet.

* Ports to be opened:

| Machine | Protocol | Port | Description |
| --- | --- | --- | --- |
| All machines <-->  all machines | TCP | 2379-2380 | etcd server, peer, and metrics ports |
| 6443 | Kubernetes API |
| 9000-9999 | Host level services, including the node exporter on ports 9100-9101 and the Cluster Version Operator on port 9099 |
| 10249-10259 | The default ports that Kubernetes reserves |
| 1936 | Node Metrics Collection Port |
| 30000-32767 | Kubernetes NodePort |
| 10256 | openshift-sdn |
| UDP | 4789 | VXLAN and GENEVE |
| 6081 | VXLAN and GENEVE |
| 9000-9999 | Host level services, including the node exporter on ports 9100-9101. |
| 30000-32767 | Kubernetes NodePort |
| Load Balancer (Internal API VIP) <-->  Control, Bootstrap | TCP | 6443 | Kubernetes API |
| 22623 | Machine config |
| Load Balancer (API VIP) <-->  Control, Bootstrap | TCP | 6443 | Kubernetes API |
| Application Load Balancer <-->  Router Nodes | TCP | 443, 80, 1936 | Ingress |
| all-machine to control plane communications | TCP | 6443 | Kubernetes API |
| all-machine to AD server, SMTP, NTP, NFS, Vmware API, SSH, Syslog,DNS | TCP | AD server, SMTP, NTP, NFS, Vmware API, SSH, Syslog,DNS | Infrastructure services |
| HTTP Server | TCP | 80 | HTTP Server |

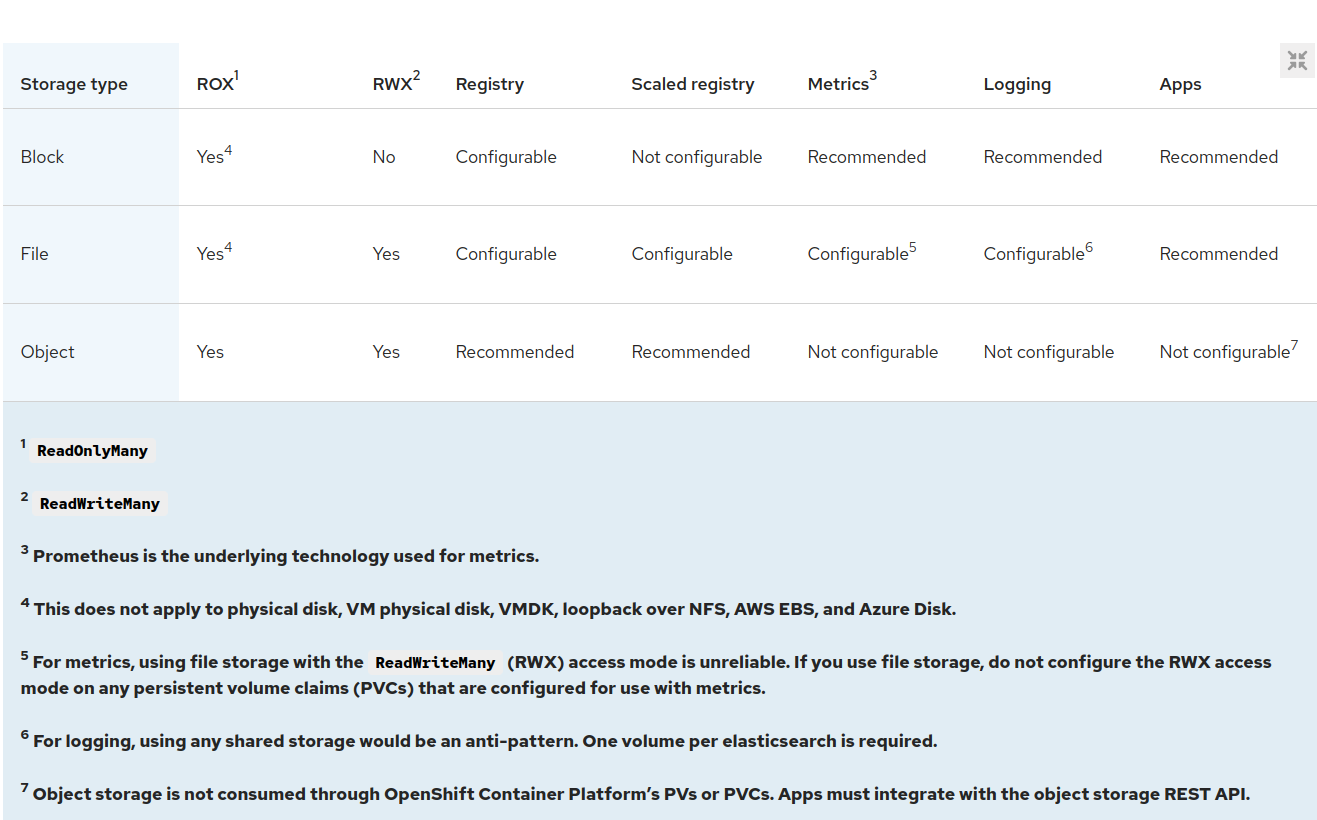
Detailed steps are covered in the RHOCP documentation42.

## 5.4 Persistence Storage

Following OCP components require storage are as below:

|  |  |  |
| --- | --- | --- |
| OCP Component | Access Type | Storage type |
| Internal Registry | RWX | Object storage |
| Monitoring | RWO | Block storage |
| Alert Manager | RWO | Block storage |
| Logging | RWX | Object storage |
| Compliance Operator | RWO | Block storage |
| ETCD Backup | RWX | NFS |
| Mirror Registry | RWO | Block |

Note:- XXX will increase the storage size in future based on utilization of respective components if required.



## 

## 5.5 Internet Access

* Deployment will be a disconnected deployment.
* HTTP Proxy must be whitelisted for below URLs:

|  |  |
| --- | --- |
| **URL** | **Port** |
| registry.redhat.io | 443, 80 |
| [quay.io](http://quay.io/) | 443, 80 |
| [cdn.quay.io](http://cdn.quay.io/) | 443, 80 |
| [cdn01.quay.io](http://cdn01.quay.io/) | 443, 80 |
| [cdn02.quay.io](http://cdn02.quay.io/) | 443, 80 |
| [cdn03.quay.io](http://cdn03.quay.io/) | 443, 80 |
| [sso.redhat.com](http://sso.redhat.com/) | 443, 80 |
| [cert-api.access.redhat.com](http://cert-api.access.redhat.com/) | 443, 80 |
| [api.access.redhat.com](http://api.access.redhat.com/) | 443, 80 |
| [infogw.api.openshift.com](http://infogw.api.openshift.com/) | 443, 80 |
| [console.redhat.com/api/ingress](http://console.redhat.com/api/ingress), [cloud.redhat.com/api/ingress](http://cloud.redhat.com/api/ingress) | 443, 80 |
| [mirror.openshift.com](http://mirror.openshift.com/) | 443, 80 |
| [storage.googleapis.com/openshift-release](http://storage.googleapis.com/openshift-release) | 443, 80 |
| [quayio-production-s3.s3.amazonaws.com](http://quayio-production-s3.s3.amazonaws.com/) | 443, 80 |
| [api.openshift.com](http://api.openshift.com/) | 443, 80 |
| [rhcos.mirror.openshift.com](http://rhcos.mirror.openshift.com/) | 443, 80 |
| [console.redhat.com/openshift](http://console.redhat.com/openshift) | 443, 80 |
| [registry.access.redhat.com](http://registry.access.redhat.com/) | 443, 80 |
| [sso.redhat.com](http://sso.redhat.com/) | 443, 80 |
| [registry.connect.redhat.com](http://registry.connect.redhat.com/) | 443, 80 |
| [rhc4tp-prod-z8cxf-image-registry-us-east-1-evenkyleffocxqvofrk.s3.dualstack.us-east-1.amazonaws.com](http://rhc4tp-prod-z8cxf-image-registry-us-east-1-evenkyleffocxqvofrk.s3.dualstack.us-east-1.amazonaws.com/) | 443, 80 |
| [oso-rhc4tp-docker-registry.s3-us-west-2.amazonaws.com](http://oso-rhc4tp-docker-registry.s3-us-west-2.amazonaws.com/) | 443, 80 |

## 

## 5.6 Machine for Application deployment (Bastion Node)

A RHEL machine in the same zone is required to access the OCP for deployment. Tools to be installed on the RHEL machine.

* 1. OC client
  2. openshift-install
  3. Podman
  4. Skopeo
  5. Butane

## 5.7 Certificates

NA, XXX confirmed that self signed certificates to be used.

## 5.8 DNS

OpenShift Container Platform requires a fully functional DNS server in the environment. This is ideally a separate host running DNS software and can provide name resolution to hosts and containers running on the platform. Key supporting activities to be carried out are:

1. Configuring hosts to use forward and backward hostname resolution.
2. Configuring a DNS wildcard for the router to use.
3. Configuring node hostnames to remain resolvable from all other node

Detailed steps along with verification steps are covered in the documentation[[16]](#footnote-16).

|  |  |
| --- | --- |
|  | DNS requirements are part of a separate pre-requisite excel document |

## 5.9 Miscellaneous

### 5.9.1 NTP synchronization

Red Hat recommends keeping all the hosts in sync via multiple NTP servers. XXX will provide 2x internal NTP servers from physical devices i.e network switches and the sync configuration will be done using machine config, for more details please refer to Red Hat documentation.

### 5.9.2 ETD Backup and Restore

Redhat Recommends to backup ETCD before OCP upgrade. etcd is the key-value store for OpenShift Container Platform, which persists the state of all resource objects.

Recommendations:

1. Back up your cluster’s etcd data by performing a single invocation of the backup script on a control plane host.
2. Only save a backup from a single control plane host. Do not take a backup from each control plane host in the cluster.

Decisions:

1. XXX Team has confirmed to take etcd backup and store in NFS storage provided by Netapp and NFS storage must be reachable from bastion as well as master nodes.
2. Application backup will be taken care of by the respective application team with their own solution.

Note: At This point of Time, complete rollback up of the Openshift cluster is not supported.

For more information please refer to: <https://docs.openshift.com/container-platform/4.14/backup_and_restore/control_plane_backup_and_restore/backing-up-etcd.html#backup-etcd>

### 5.10 Firewall Rules

|  |  |  |
| --- | --- | --- |
| Ports used for all-machine to all-machine communications | | |
| **Protocol** | **Port** | **Description** |
| ICMP | N/A | Network reachability tests |
| TCP | 1936 | Metrics |
| 9000-9999 | Host level services, including the node exporter on ports 9100-9101 and the Cluster Version Operator on port 9099. |
| 10250-10259 | The default ports that Kubernetes reserves |
| 10256 | openshift-sdn |
| UDP | 4789 | VXLAN |
| 6081 | Geneve |
| 9000-9999 | Host level services, including the node exporter on ports 9100-9101. |
| 500 | IPsec IKE packets |
| 4500 | IPsec NAT-T packets |
| TCP/UDP | 30000-32767 | Kubernetes node port |

|  |  |  |
| --- | --- | --- |
| Ports used for all-machine to control plane communications | | |
| **Protocol** | **Port** | **Description** |
| TCP | 6443 | Kubernetes API |

|  |  |  |
| --- | --- | --- |
| Ports used for control plane machine to control plane machine communications | | |
| **Protocol** | **Port** | **Description** |
| TCP | 2379-2380 | etcd server and peer ports |

|  |  |  |
| --- | --- | --- |
| Ports used for bastion node and all nodes | | |
| **Protocol** | **Port** | **Description** |
| TCP | 80 | ignition files access from bastion node |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Firewall ACL Rules** | | | | | | | | | |
| **No.** | **Zone** | **Source** | | | **Destination** | | | | **Remarks** |
| **Source Application Name** | **IP Address** | **Direction** | **Destination Application Name** | **IP Address** | **Protocol (TCP/UDP)** | **Destination Port** |
| **1** |  | OCP-All\_Nodes |  | SRC to Dest | OCP-API LB |  | TCP | 6443 | Passthrough  (layer 4 routing) |
| **2** |  | OCP-All\_Nodes |  | SRC to Dest | OCP-APP LB |  | TCP | 443, 1936 | Passthrough  (layer 4 routing) |
| **3** |  | OCP-All\_Nodes |  | SRC to Dest | DNS |  | TCP,UDP | 53 | DNS |
| **4** |  | OCP-All\_Nodes |  | SRC to Dest | NTP |  | TCP,UDP | 123 | NTP |
| **5** |  | OCP Master Nodes |  | SRC to Dest | OCP-SMTP |  | TCP | 587 | SMTP |
| **6** |  | OCP - Mirror Registory Node |  | SRC to Dest | HTTP Proxy |  | TCP | proxy port | HTTP Proxy for connectivity from OCP nodes |
| **7** |  | OCP-ALL\_Nodes |  | SRC to Dest | Siem /syslog |  | TCP,UDP |  | to forward logs to central log server |
| **8** |  | OCP-ALL\_Nodes |  | SRC to Dest | Netapp Mgmt Port |  | TCP | Netapp Mgmt Port | ocp to netapp mgmt communication |
| **9** |  | OCP-ALL\_Nodes |  | SRC to Dest | Netapp Object Storage |  | TCP |  | Object Storage for OCR |
| **10** |  | Bastion Node, NOC network, OCP-All\_Nodes |  | SRC to Dest | iLO/IPMI Port |  |  |  | iDRAC |
| **11** |  | NOC network, XXX Prometheus |  | SRC to Dest | OCP LB |  | TCP | 6443 | API Access |
| **12** |  | NOC network, XXX Prometheus |  | SRC to Dest | OCP LB |  | TCP | 443, 1936 | \*.apps |
| **13** |  | NOC network |  | SRC to Dest | Bastion Node, Registry Mirror, KVM |  | TCP | 22 | SSH Port |
|  |  | Vulenerability Scan Servers |  |  | OCP- All\_nodes |  |  |  |  |

### 

Appendix A

|  |  |  |
| --- | --- | --- |
| ID | Prerequisite | Result |
| 1 | Overall Cluster Size/Growth | Maximum cluster size is expected to be less than 200 Nodes in the future. |
| 2 | LoadBalancer Model | F5, SSL passthrough. |
| 3 | Retention periods | 1. OS logs, Application Logs, Audit logs - 10 Days 2. Monitoring/Metrics - 30 days |
| 4 | Storage for Prometheus, Alert Manager & Logging Components | Block storage from Netapp for Monitoring and  Logging components |
| 5 | Storage for OCR Registry | Netapp Object Storage |
| 6 | Storage for Compliance Operator | Netapp Block Storage |
| 7 | SMTP / Mail Server details for Monitoring/Alerts | YES AVAILABLE: Remark: At the time of installation XXX will provide  SMTP server and credentials (like smtp\_from, smtp\_smarthost, smtp\_auth\_username, smtp\_auth\_password etc) will be required to configure alertmanager notifications. |
| 8 | HTpasswd Authentication | Yes. |
| 9 | LDAP/AD | NA |

## TESTING

## Test Scenario 1: Openshift Web Console

Summary

Test case to verify that Web Console is working properly.

Description

|  |  |
| --- | --- |
| Tasks | Description |
| Environment | System Server Components: OpenShift Master: <https://console-openshift-console.apps>.<cluster\_name>.<base\_domain> |
| Prerequisite | FQDN should be resolvable and reachable.  All Red Hat OpenShift components are working properly. |
| Chronological Test | 1. Navigate to openshift-web-console project. 2. Check that all pods are running. 3. Make sure the console is accessible with a browser. |
| Conditions of Success | Browse URL <https://console-openshift-console.apps>.<cluster\_name>.<base\_domain> is working properly. |
| Unforeseen circumstances | Troubleshooting will be done where time permits. |
| Result |  |
| Authorized Name/Signatory |  |

## Test Scenario 2: Administrative Portal with kubeadmin

Summary

Test to verify serviceability of Red Hat OpenShift via login of administration portal using Web Browser.

Description

|  |  |
| --- | --- |
| Tasks | Description |
| Environment | System Server Components: OpenShift Master: <https://console-openshift-console.apps>.<cluster\_name>.<base\_domain> |
| Prerequisite | FQDN should be resolvable and reachable |
| Chronological Test | 1. Access Red Hat OpenShift Admin URL from browser: <https://console-openshift-console.apps>.<cluster\_name>.<base\_domain> 2. Enter kubeadmin login & password. |
| Conditions of Success | Successful login into OCP console. |
| Unforeseen circumstances | Troubleshooting will be done where time permits. |
| Result |  |
| Authorized Name/Signatory |  |

## Test Scenario 3: Administrative Portal with HTPASSWD

Summary

Test to verify serviceability of Red Hat OpenShift via htpasswd login of administration portal using Web Browser.

Description

|  |  |
| --- | --- |
| Tasks | Description |
| Environment | System Server Components: OpenShift Master: <https://console-openshift-console.apps>.<cluster\_name>.<base\_domain> |
| Prerequisite | FQDN should be resolvable and reachable  Red Hat OpenShift services are working properly. |
| Chronological Test | 1. Access Red Hat OpenShift Admin URL from browser: <https://console-openshift-console.apps>.<cluster\_name>.<base\_domain> 2. Enter HTPASSWD user name login & password. |
| Conditions of Success | Successful login into OCP console. |
| Unforeseen circumstances | Troubleshooting will be done where time permits. |
| Result |  |
| Authorized Name/Signatory |  |

## Test Scenario 4: OpenShift Masters and Workers Status

Summary

Test to verify all OpenShift nodes are in operating state.

Description

|  |  |
| --- | --- |
| Tasks | Description |
| Environment | System Server Components: OpenShift Master: <https://console-openshift-console.apps>.<cluster\_name>.<base\_domain> |
| Prerequisite | FQDN should be resolvable and reachable.  Red Hat OpenShift services are working properly. |
| Chronological Test | 1. $ oc login -u kubeadmin https://api.<cluster\_name>.<base\_domain>:6443 (or other cluster-admin) 2. $ oc get nodes. |
| Conditions of Success | The command output should present “STATUS” as “Ready” on Masters and Workers. |
| Unforeseen circumstances | Troubleshooting will be done where time permits. |
| Result |  |
| Authorized Name/Signatory |  |

## Test Scenario 5: Dynamic Provisioning

Summary

Test that all dynamic provisioning is working properly.

Description

|  |  |
| --- | --- |
| Tasks | Description |
| Environment | System Server Components: OpenShift Master: <https://console-openshift-console.apps>.<cluster\_name>.<base\_domain> |
| Prerequisite | FQDN should be resolvable and reachable.  Red Hat OpenShift services are working properly. |
| Chronological Test | 1. $ oc get storageclass should return all storage class:    1. <isilon-file>    2. <isilon-block> 2. Create pvc with the storageclass. |
| Conditions of Success | PV should be created and bound to pvc. Users should be able to write to PV for [Test Scenario 8](#_heading=h.2wwbldi). |
| Unforeseen circumstances | Troubleshooting will be done where time permits. |
| Result |  |
| Authorized Name/Signatory |  |

# 

## Test Scenario 6: Deploying Sample App (Stateless)

Summary

Deploying Sample App (Stateless)

Description

|  |  |
| --- | --- |
| Tasks | Description |
| Environment | System Server Components: OpenShift Master: <https://console-openshift-console.apps>.<cluster\_name>.<base\_domain> |
| Prerequisite | FQDN should be resolvable and reachable.  Red Hat OpenShift services are working properly.  Test Scenario 6 should be passed |
| Chronological Test | 1. $ oc new-app <image> 2. Expose services 3. Navigate to route with browser. |
| Conditions of Success | The hello-openshift application page should be displayed at the configured URL. |
| Unforeseen circumstances | Troubleshooting will be done where time permits. |
| Result |  |
| Authorized Name/Signatory |  |

## Test Scenario 7: Deploying Sample App (Stateful)

Summary

Deploying Sample App (Stateless)

Description

|  |  |
| --- | --- |
| Tasks | Description |
| Environment | System Server Components: OpenShift Master: <https://console-openshift-console.apps>.<cluster\_name>.<base\_domain> |
| Prerequisite | FQDN should be resolvable and reachable.  Red Hat OpenShift services are working properly. |
| Chronological Test | 1. Create a new project. 2. Create pvc with storage class 3. $ oc new-app <image> 4. Mount PVC to deployment 5. $ oc rsh into pod and create file with some data. 6. $ oc delete <pod> to redeploy pod 7. $ oc rsh into pod to check if the file with the same data is still there |
| Conditions of Success | The mariadb application should be deployed.  Redeployed pod should still show the added file. |
| Unforeseen circumstances | Troubleshooting will be done where time permits. |
| Result |  |
| Authorized Name/Signatory |  |

## Test Scenario 8: Add Resource Quota to Namespace

Summary

Check that the resource quota applied to namespaces is applied and working.

Description

|  |  |
| --- | --- |
| Tasks | Description |
| Environment | System Server Components: OpenShift Master: <https://console-openshift-console.apps>.<cluster\_name>.<base\_domain> |
| Prerequisite | FQDN should be resolvable and reachable.  Red Hat OpenShift services are working properly. |
| Chronological Test | 1. Create resource quotas on namespace created from [Test Scenario 7](#_heading=h.1c1lvlb) 2. Check that existing deployments have pod request and limit applied 3. Scale up pods. |
| Conditions of Success | The number of pods available should be within limits/resourcequota set |
| Unforeseen circumstances | Troubleshooting will be done where time permits. |
| Result |  |
| Authorized Name/Signatory |  |

## Test Scenario 9: Add Network Policy to Namespace

Summary

Check that networkpolicy applied to namespaces is applied and working.

Description

|  |  |
| --- | --- |
| Tasks | Description |
| Environment | System Server Components: OpenShift Master: <https://console-openshift-console.apps>.<cluster\_name>.<base\_domain> |
| Prerequisite | FQDN should be resolvable and reachable.  Red Hat OpenShift services are working properly. |
| Chronological Test | 1. Check for pod IP for pods created from [Test Scenario 6](#_heading=h.1c1lvlb) 2. $ oc rsh into another pod and try to curl into pod IP from step 1 3. Pods should be accessible. 4. Create deny ingress to namespace from [Test Scenario 6](#_heading=h.1c1lvlb) 5. Repeat step 2, pod would now be inaccessible. |
| Conditions of Success | Pods should comply with network policy applied |
| Unforeseen circumstances | Troubleshooting will be done where time permits. |
| Result |  |
| Authorized Name/Signatory |  |

# 

## Test Scenario 10: Openshift Monitoring

Summary

Test that Prometheus, OpenShift Monitoring is working properly.

Description

|  |  |
| --- | --- |
| Tasks | Description |
| Environment | System Server Components: OpenShift Master: <https://console-openshift-console.apps>.<cluster\_name>.<base\_domain> |
| Prerequisite | FQDN should be resolvable and reachable.  Red Hat OpenShift services are working properly. |
| Chronological Test | 1. Navigate to the openshift-monitoring project. 2. Check that all services provisioned by the operator should be working and running properly. 3. Check that PVC is mounted to the pods properly. |
| Conditions of Success | Verify that all prometheus PODs are running properly.   * Browse URL https://grafana-openshift-monitoring.apps.<cluster\_name>.<base\_domain> is working properly. * Browse URL https://prometheus-k8s-openshift-monitoring.apps.<cluster\_name>.<base\_domain> is working properly. With all TARGETS status showing ‘UP’. * Browse URL https://alertmanager-main-openshift-monitoring.apps.<cluster\_name>.<base\_domain> is working properly. |
| Unforeseen circumstances | Troubleshooting will be done where time permits. |
| Result |  |
| Authorized Name/Signatory |  |

## Test Scenario 11: Openshift Logging

Summary

Test that OpenShift Logging is working properly.

Description

|  |  |
| --- | --- |
| Tasks | Description |
| Environment | System Server Components: OpenShift Master: <https://console-openshift-console.apps>.<cluster\_name>.<base\_domain> |
| Prerequisite | FQDN should be resolvable and reachable.  Red Hat OpenShift services are working properly. |
| Chronological Test | 1. Navigate to the openshift-logging project. 2. Check that all pods are running. 3. Check that OBC is mounted properly. |
| Conditions of Success | Browse URL https://kibana.apps.<cluster\_name>.<base\_domain> is working properly.  Dashboard should be showing the latest logs. |
| Unforeseen circumstances | Troubleshooting will be done where time permits. |
| Result |  |
| Authorized Name/Signatory |  |

## Test Scenario 12: Metrics Server

Summary

Test that the Openshift metrics server is working properly.

Description

|  |  |
| --- | --- |
| Tasks | Description |
| Environment | System Server Components: OpenShift Master: <https://console-openshift-console.apps>.<cluster\_name>.<base\_domain> |
| Prerequisite | FQDN should be resolvable and reachable.  Red Hat OpenShift services are working properly. |
| Chronological Test | 1. Run $ oc adm top nodes |
| Conditions of Success | There should be output showing usage for each node. |
| Unforeseen circumstances | Troubleshooting will be done where time permits. |
| Result |  |
| Authorized Name/Signatory |  |

## Test Scenario 13: Master High Availability

Summary

Test that Master High Availability is working.

Description

|  |  |
| --- | --- |
| Tasks | Description |
| Environment | System Server Components: OpenShift Master: <https://console-openshift-console.apps>.<cluster\_name>.<base\_domain> |
| Prerequisite | FQDN should be resolvable and reachable.  Red Hat OpenShift services are working properly. |
| Chronological ZTest | 1. Shutdown one of the master nodes and make sure $ oc get nodes shows status ‘NOT READY’ for the master node that was shutdown. 2. Create sample projects to test that the control plane of the cluster is still functional. 3. Start master node after test. |
| Conditions of Success | OpenShift Administrative Web Console should still be functional. |
| Unforeseen circumstances | Troubleshooting will be done where time permits. |
| Result |  |
| Authorized Name/Signatory |  |

## Test Scenario 14: Router High Availability

Summary

Test that Router High Availability is working.

Description

|  |  |
| --- | --- |
| Tasks | Description |
| Environment | System Server Components: OpenShift Master: <https://console-openshift-console.apps>.<cluster\_name>.<base\_domain> |
| Prerequisite | FQDN should be resolvable and reachable.  Red Hat OpenShift services are working properly. |
| Chronological Test | 1. Use pods and routes from [Test Scenario 6](#_heading=h.1c1lvlb). 2. Navigate to the published application URL. 3. Shutdown 1 master node which has an ingress pod is running and make sure $ oc get nodes shows status ‘NOT READY’ for the master node that was shutdown. 4. Start the master node after the test. |
| Conditions of Success | The application should still be served correctly from the same URL. |
| Unforeseen circumstances | Troubleshooting will be done where time permits. |
| Result |  |
| Authorized Name/Signatory |  |

# 

## Test Scenario 15: Resilience of Workers

Summary

Test that worker resilience is working.

Description

|  |  |
| --- | --- |
| Tasks | Description |
| Environment | System Server Components: OpenShift Master: <https://console-openshift-console.apps>.<cluster\_name>.<base\_domain> |
| Prerequisite | FQDN should be resolvable and reachable.  Red Hat OpenShift services are working properly. |
| Chronological Test | 1. Use pods and routes from [Test Scenario 6](#_heading=h.1c1lvlb). 2. Add liveness probe to the deployment. 3. $ oc get pods -o wide to check which node is the pod deployed. 4. Shutdown the worker node where the pod resides. 5. Start the worker node after the test. |
| Conditions of Success | The pod should be redeployed to a healthy node. |
| Unforeseen circumstances | Troubleshooting will be done where time permits. |
| Result |  |
| Authorized Name/Signatory |  |

## Test Scenario 16: Resilience of Loki

Summary

Test that Loki resilience is working.

Description

|  |  |
| --- | --- |
| Tasks | Description |
| Environment | System Server Components: OpenShift Master: <https://console-openshift-console.apps>.<cluster\_name>.<base\_domain> |
| Prerequisite | FQDN should be resolvable and reachable.  Red Hat OpenShift services are working properly. |
| Chronological Test | 1. Shutdown one of the infra logging nodes. 2. Browse logs on console. 3. Restart the infra logging nodes 4. Console should still be receiving logs from the servers |
| Conditions of Success | OpenShift Console should be showing the latest logs from the cluster. |
| Unforeseen circumstances | Troubleshooting will be done where time permits. |
| Result |  |
| Authorized Name/Signatory |  |

## Test Scenario 17: Resilience of Monitoring nodes

Summary

Test that openshift-monitoring resilience.

Description

|  |  |
| --- | --- |
| Tasks | Description |
| Environment | System Server Components: OpenShift Master: <https://console-openshift-console.apps>.<cluster\_name>.<base\_domain> |
| Prerequisite | FQDN should be resolvable and reachable.  Red Hat OpenShift services are working properly. |
| Chronological Test | 1. Shutdown one of the infra-monitoring node where prometheus-k8s is running 2. $ oc get pods -o wide -n openshift-monitoring 3. Browse route served by prometheus-k8s and grafana. 4. Monitoring services should still be served, some targets in prometheus-k8s are showing down, however metrics should still be collected by the active prometheus. |
| Conditions of Success | Metrics shown should be the latest and the targets should be active |
| Unforeseen circumstances | Troubleshooting will be done where time permits. |
| Result |  |
| Authorized Name/Signatory |  |

## Test Scenario 18: Taints and Tolerations

Summary

Ensure that no application pods are deployed on infra nodes

Description

|  |  |
| --- | --- |
| Tasks | Description |
| Environment | System Server Components: OpenShift Master: <https://console-openshift-console.apps>.<cluster\_name>.<base\_domain> |
| Prerequisite | FQDN should be resolvable and reachable.  Red Hat OpenShift services are working properly. |
| Chronological Test | 1. Run $ oc get pods --all-namespaces -o wide |
| Conditions of Success | All application pods should be scheduled on worker nodes. |
| Unforeseen circumstances | Troubleshooting will be done where time permits. |
| Result |  |
| Authorized Name/Signatory |  |

# 

## Test Scenario 19: [Scale Out Worker Nodes](https://access.redhat.com/solutions/4799921)

Summary

Demonstrate new compute nodes can be added.

Description

|  |  |
| --- | --- |
| Tasks | Description |
| Environment | System Server Components: OpenShift Master: <https://console-openshift-console.apps>.<cluster\_name>.<base\_domain> |
| Prerequisite | FQDN should be resolvable and reachable.  Red Hat OpenShift services are working properly.  New compute node’s DNS and DHCP entry is present. |
| Chronological Test | 1. Power off the bare metal node. 2. Create 2 secrets in openshift-machine-api namespace , one with the network interface details and one with the username and password of the baseboard management controller of the bare metal node. 3. Create BMH (BareMetalHost) custom resource in the cluster. 4. Verify all the resources are created. 5. Power on the bare metal host and scale the machienset. |
| Conditions of Success | $ oc get nodes should show the new node. |
| Unforeseen circumstances | Troubleshooting will be done where time permits. |
| Result |  |
| Authorized Name/Signatory |  |

1. VIP: [https://access.redhat.com/documentation/en-us/openshift\_container\_platform/4.14/html/installing/installing-on-bare-metal#installation-load-balancing-user-infra\_installing-restricted-networks-bare-metal](https://access.redhat.com/documentation/en-us/openshift_container_platform/4.12/html/installing/installing-on-bare-metal#installation-load-balancing-user-infra_installing-restricted-networks-bare-metal) [↑](#footnote-ref-1)
2. OCP Registries: <https://access.redhat.com/documentation/en-us/openshift_container_platform/4.14/html-single/registry/index#registry-options> [↑](#footnote-ref-2)
3. Web console: <https://access.redhat.com/documentation/en-us/openshift_container_platform/4.14/html-single/web_console/index> [↑](#footnote-ref-3)
4. Router and routes: <https://access.redhat.com/documentation/en-us/openshift_container_platform/4.14/html-single/networking/index#route-configuration> [↑](#footnote-ref-4)
5. Monitoring stack:

   <https://access.redhat.com/documentation/en-us/openshift_container_platform/4.14/html-single/monitoring/index#configuring-the-monitoring-stack> [↑](#footnote-ref-5)
6. Loki installation reference: <https://docs.openshift.com/container-platform/4.14/logging/log_storage/cluster-logging-loki.html> [↑](#footnote-ref-6)
7. <https://access.redhat.com/documentation/en-us/openshift_container_platform/4.14/html/logging/log-storage#installing-log-storage-loki> [↑](#footnote-ref-7)
8. Storage threshold: [https://access.redhat.com/documentation/en-us/openshift\_container\_platform/4.14/html-single/logging/index#cluster-logging-deploy-storage-considerations\_cluster-logging-deploying-about](https://access.redhat.com/documentation/en-us/openshift_container_platform/4.14/html-single/logging/index%23cluster-logging-deploy-storage-considerations_cluster-logging-deploying-about) [↑](#footnote-ref-8)
9. OpenShift tested configuration: [https://access.redhat.com/articles/4763741](https://access.redhat.com/articles/4128421) [↑](#footnote-ref-9)
10. Updating OCP 4.14 cluster: <https://access.redhat.com/documentation/en-us/openshift_container_platform/4.14/html/updating_clusters/index> [↑](#footnote-ref-10)
11. Cluster limits: <https://access.redhat.com/documentation/en-us/openshift_container_platform/4.14/html-single/scalability_and_performance/index#planning-your-environment-according-to-object-limits> [↑](#footnote-ref-11)
12. 12 factor app principles: <https://12factor.net/> [↑](#footnote-ref-12)
13. Authentication and authorisation: <https://access.redhat.com/documentation/en-us/openshift_container_platform/4.14/html-single/authentication_and_authorization/index#understanding-authentication> [↑](#footnote-ref-13)
14. Volume security:

    <https://access.redhat.com/documentation/en-us/openshift_container_platform/4.14/html-single/authentication_and_authorization/index#managing-pod-security-policies> [↑](#footnote-ref-14)
15. Tested Integration - <https://access.redhat.com/articles/4128421> [↑](#footnote-ref-15)
16. DNS requirements(Section 24.1.5): <https://access.redhat.com/documentation/en-us/openshift_container_platform/4.14/html/installing/installing-on-any-platform#installation-user-provisioned-validating-dns_installing-platform-agnostic> [↑](#footnote-ref-16)