**CUSTOMER**

**Red Hat Openshift Platform 4.14 &**

**Red Hat Ceph Storage**

**Low Level Solution Design Document**

## Purpose

This document has been produced to centrally capture the information which has been discussed during design workshop with the CUSTOMER team w.r.t. deployment of RHOCP & RHCS environments.

The document covers low level design details describing an RHOCP 4.14 & RHCS 6 platform design. Additionally, it provides guidelines for solution implementation and validation.

## Scope

The scope of this document is to describe the low level design that Red Hat can offer and support by leveraging the Features provided by Red Hat Openshift Container Platform (RHOCP) & Red Hat Ceph Storage (RHCS).

This LLD Solution Design document is the result of the design workshop and should cover all requirements that CUSTOMER has put forward during the workshop.

## Out Of Scope

This LLD document limits the discussions and recommendations only to Red Hat Components involved for RHOCP & RHCS Clusters. Anything not explicitly listed as in-scope is deemed to be out of scope of this design document.

# Design Workshop

An RHOCP & RHCS architecture & design workshop was conducted for CUSTOMER . This time was utilized to go over the RHOCP & RHCS high level architectural components, Its applications and all other needed details required to deploy the cluster including the open items, challenges & expectations from the Customer.

In the workshop session, it’s been decided to go ahead with OCP 4.14 deployment using UPI (User Provisioned Installation) in Connected mode. The workshop scope was confined to the OCP & Ceph design.

CUSTOMER application requirement is to use containerization technology of RHOCP to host their EDA Application.

# Technical Components & High Level Architecture

This section covers the key technical components that will be deployed for CUSTOMER at its data center in Bogura, Bangladesh.

As an introduction to OCP; OpenShift Container Platform is a platform for developing and running containerized applications. It is designed to allow applications and the data centers that support them to expand from just a few machines and applications to thousands of machines that serve millions of clients. With its foundation in Kubernetes, OpenShift Container Platform incorporates the same technology that serves as the engine for massive telecommunications, streaming video, gaming, banking and other applications. Its implementation in open Red Hat technologies lets you extend your containerized applications beyond a single cloud to on-premise and multi-cloud environments. Red Hat OpenShift Platform adds:

* Source code management, builds, and deployments for developers
* Managing and promoting images at scale as they flow through your system
* Application management at scale
* Team and user tracking for organizing a large developer organization
* Networking infrastructure that supports the cluster

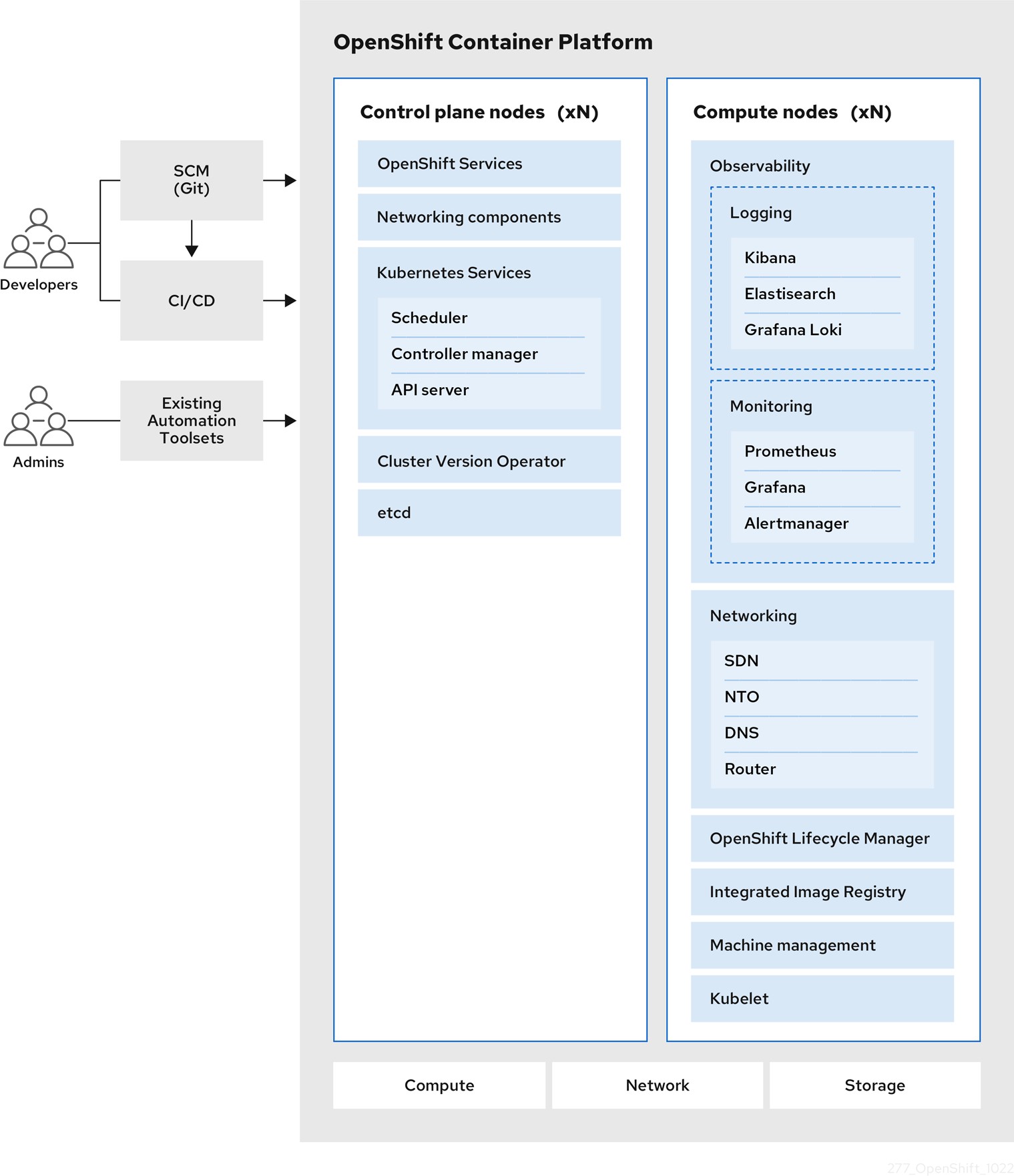


Figure - 3.a - High Level RHOCP Component Level Reference Architecture

RHOCP will be powered by Red Hat OpenShift Data Foundation which provides persistent software-defined storage based on Red Hat Ceph Storage and optimized for OpenShift. Here, we will be connecting to an existing Ceph storage cluster which was deployed for the CMS application. A new dedicated pool would be created in Ceph Storage and ODF will consume storage from ceph Storage. ODF will be deployed with external mode.

Below diagram depicts architecture concluded for OCP 4.14 at CUSTOMER .

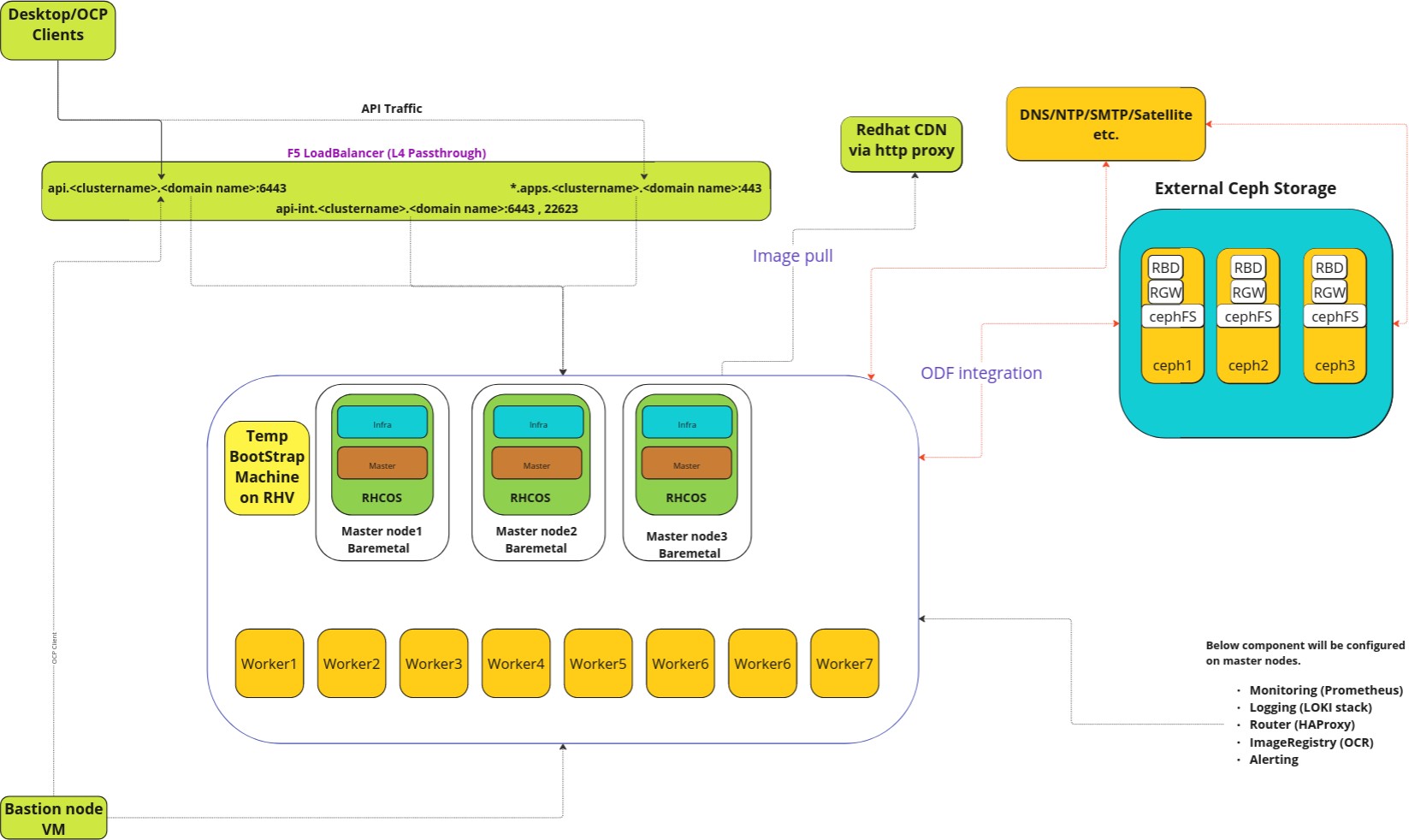


Figure - 3.b - Logical Diagram concluded at CUSTOMER

## Infrastructure Components

### 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.

Key Components of Openshift :-

* + - * 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.

### Design decisions

* + - * + Red Hat OCP 4.14 cluster to be deployed on BareMetal using the Connected User Provisioned Installation.
        + CUSTOMER will provide Internet using Internet Proxy required for deployment of OCP, Detail will be added in Pre-req Sheet.
        + CUSTOMER will use Hardware Load Balancer (F5 Load Balancer) required for LB VIPs in L4/Passthrough mode.
        + New OCP 4.14 cluster name will be pocpeda-nexp.CUSTOMER .com
        + There will be 1 bastion machine with the latest RHEL 9.x Installed. Bastion node will be a virtual machine deployed on RHV.
        + Infra components such as Router, Registry, Loki and Prometheus Monitoring will be enabled on master nodes only.
        + For Logging Loki will be enabled with 1x.medium sizing. Refer [article](https://access.redhat.com/documentation/en-us/openshift_container_platform/4.14/html-single/logging/index#loki-deployment-sizing_installing-log-storage)
        + 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 2 VIP’s1 will be provided by the CUSTOMER team from F5 Hardware Load Balancer for each cluster.

API VIP

API-INT VIP

\*.apps VIP

api and api-int DNS endpoints would point to the same VIP.

* + - * + The CUSTOMER 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.
        + CUSTOMER will provide 4 NTP servers.
        + CUSTOMER agreed to use self signed OCP Installer generated (\*.apps) wild card certificates to be used for Openshift Ingress Controller.
        + API will use self signed OCP generated certificates.
        + The master API and Web console will be available on port 6443. Please refer to section “5.3 Network Access Requirements” for the complete list of ports that need opening.
        + Other infrastructure requirements are covered in the 5. Implementation Prerequisites.

1 VIP:

[https://access.redhat.com/documentation/en-us/openshift\_container\_platform/4.14/html/installing/installing-on-bare-metal#ins](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) [tallation-load-balancing-user-infra\_installing-restricted-networks-bare-metal](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)

* + - * + Separate validation prerequisite spreadsheet has been shared with the CUSTOMER team. All details need to be provided before starting the deployments.
        + CUSTOMER has confirmed to use OVN Kubernetes as a default CNI.
        + CUSTOMER has plans to use multus in future for their workloads, Hence 2 network ports in the server will be left separately for future use of multus. As part of this deployment a second bond will not be configured. It will be configured in LACP bond during day 2 operations in case CUSTOMER has a requirement to add another interface to Pod using multus in future.
        + On all nodes(masters and workers) there will be one LACP bond and 2 nics will be configured under one LACP bond.
        + Openshift Will be integrated with External Ceph Storage using ODF. A dedicated pool would be created on the existing Ceph cluster for EDA application(Pool name would be eda-Customer x).
        + MTU size default 9000 will be used. Effective MTU at pod level will be 8900 as OVN-Kubernetes SDN uses 100 bytes as geneve overhead.
        + The CUSTOMER team will provide a sufficient pool for host networks considering the future growth requirements. This network will include IPs for each OCP node. CUSTOMER Also confirmed maximum Worker Nodes they are planning to add are approx 15.
        + CUSTOMER will provide an L3 network for OCP environment from the same subnet range for all OCP nodes including bastion, bootstrap, master and worker nodes.
        + Users authentication to be configured using local (HTPasswd based authentication provider) as well as LDAP Server. CUSTOMER has provided LDAP Server information in the Pre-Req Sheet.
        + RHOCP will be deployed with only the ipv4 network stack.
        + RHOCP will be integrated with Ceph for all (Block, File and Object) storage requirements.
        + Prometheus, Alert manager, Loki log retention period would be 15 days.
        + Temporary bootstrap VM will be configured on RHV.
        + Red Hat will Install ODF Operator on RHOCP & Integrate it with existing external Red Hat Ceph Storage.
        + A new storage pool would be created for EDA applications in the existing Ceph Storage cluster(eda-Customer ).
        + Storage required for Openshift Registry will be provided from Red Hat Ceph Storage. Rados Gateway will be configured on Ceph for the same. Existing RadosGW ”\*.rgw.pocpmp-nexp.CUSTOMER .com” created for the previous cluster would be used for EDA cluster integration.
        + Separate prerequisites sheet has been created and shared with the CUSTOMER team to fill all details pertaining to H/W, network, storage details etc.

*Please note: Redhat Does not provide any performance guarantee.*

### 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 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

### Design decisions

Registry:

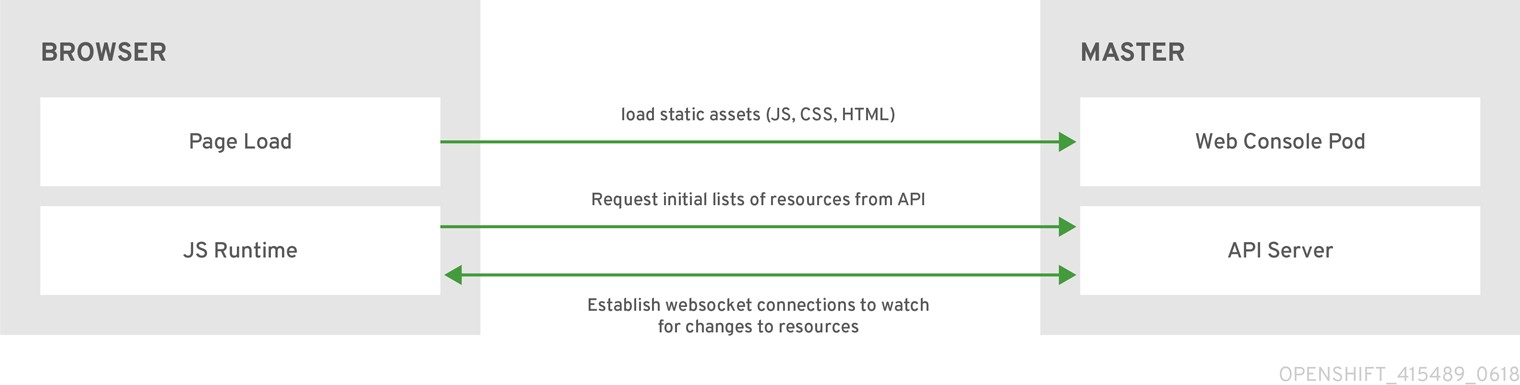
* + - * + Registry (OCR) will be implemented in clusters and will run as a container.
        + The registry stores container images and metadata. If you simply deploy the registry pod, it uses an ephemeral volume that is destroyed if the pod exits. Any images built or pushed into the registry would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

Hence, based on the available options at GP, OpenShift Container Platform registry storage will be provided from Red Hat Ceph Storage of size 500GB. CUSTOMER decided to provide us with the Ceph Rados Gateway for Storage. Refer Ceph Design decisions for more details on RGW & Its Load Balancer Requirement.

* + - * + Ceph Storage will be the backend for registry pods & It will be integrated using ODF.

### 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.



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.

### Design decisions

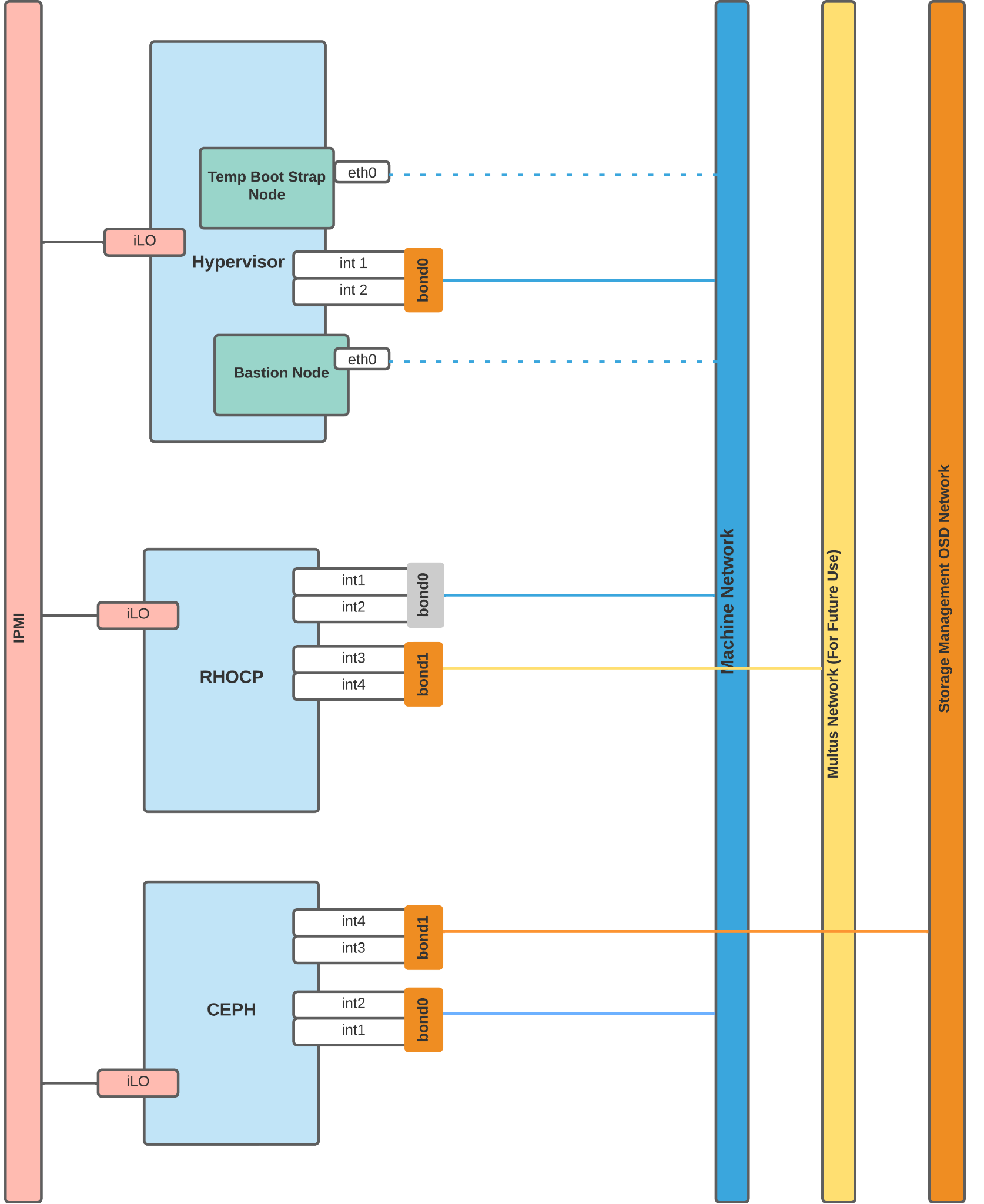
Below decisions are applicable for all clusters.

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

## Network Considerations

### 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.



### Design decisions

* + - * + CUSTOMER will use the default OVN-Kubernetes plugin.
        + RHOCP Cluster will be set up with only IPv4 schema.
        + The host network CIDR range for the OCP nodes will be provided by GP.
* **Cluster and Service subnet will be provided by the CUSTOMER team on a prerequisite sheet.**
* All OCP nodes will be reachable to each other. There is no microsegmentation at CUSTOMER Network.
* The cluster subnet and service subnet will be internal to the OCP cluster. However if applications hosted on this cluster need to connect with some external endpoint, It must not use IP from this cluster subnet and service subnet.
* 2 Interfaces in bond0 will be used for Machine Network. No requirement for additional bond configuration.

### 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.

### Design decisions & Recommendations

Below decisions are applicable for the cluster.

* + - * + 3 router pods will be deployed on 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.

### 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](https://sample-application.apps.devocp.example.local/)

### Load Balancer

The CUSTOMER team has confirmed the installation method would be UPI, below are the load balancer requirements.

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.

### API load balancer

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

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

Note:

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

### API-INT load balancer

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

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | built. |

Note:

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

### \*.APPS load balancer

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| API endpoint | Port | VIP | ssl mode | Backend server |
| \*.apps.<cluster\_name>.CUSTOMER .co m | 443  and 80 |  | passthrough | master1 master2 master3 |

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.

## 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.

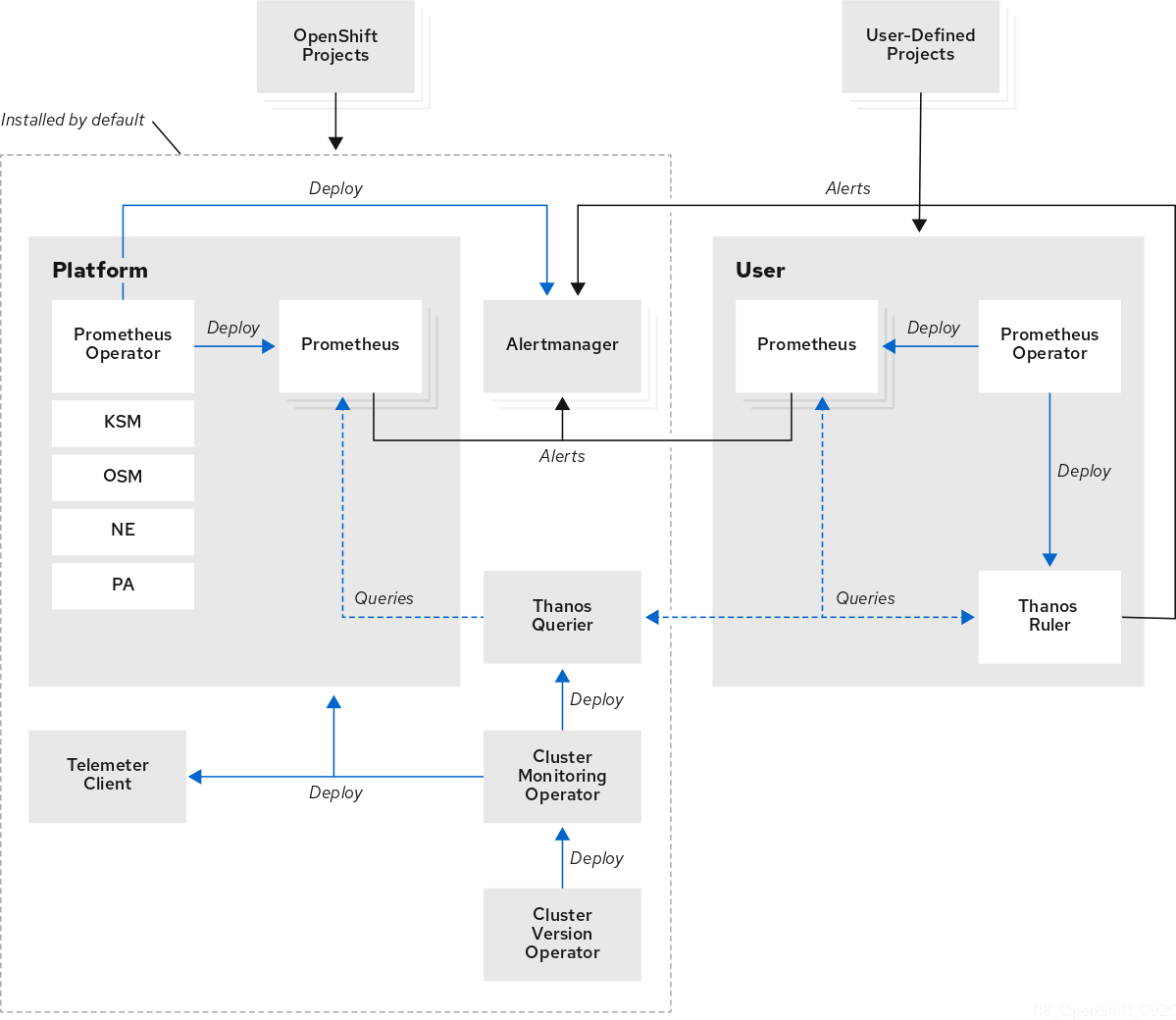
You also have the option to enable monitoring for user-defined projects.

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](https://docs.openshift.com/container-platform/4.11/monitoring/managing-metrics-targets.html#managing-metrics-targets) [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](https://docs.openshift.com/container-platform/4.11/monitoring/troubleshooting-monitoring-issues.html#troubleshooting-monitoring-issues) [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
* Elasticsearch (if Logging is installed)
* Etcd
* Fluentd (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)

### Key observation/challenges/requirements

1. The Monitoring(Prometheus+AlertManager) component will be deployed on master nodes, one pod on each node.
2. Grafana is deprecated in OCP 4.10 and removed from the 4.11 OCP version. Starting with

4.12 OCP Monitoring dashboards are an integrated part of the OCP web console itself.

1. As CUSTOMER does not want persistent internet connectivity but only provide during cluster deployment and during cluster upgrade, Insights Cluster Operator will be disabled.

**Note:** This may impact openshift nodes as the connectivity is required to download images. If a node panics or any of the operator pods are re-scheduled on another node, the pod would not be able to download images.

### Design Decisions

1. OCP Cluster monitoring will be installed2; 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 Ceph Storage. Depending upon usage, this size will be increased during day 2 ops.

2 Monitoring stack:

[https://access.redhat.com/documentation/en-us/openshift\_container\_platform/4.14/html-single/monitoring/index#](https://access.redhat.com/documentation/en-us/openshift_container_platform/4.14/html-single/monitoring/index#configuring-the-monitoring-stack) [conﬁguring-the-monitoring-stack](https://access.redhat.com/documentation/en-us/openshift_container_platform/4.14/html-single/monitoring/index#configuring-the-monitoring-stack)

1. The monitoring data will be stored for 15 days and ODF storage will be used to store the data.
2. Cluster/Namespace/Pod level monitoring dashboards are available by default in the OCP console.
3. Default alert rules will be installed.
4. Alert Manager provides various standard ways to send out notifications like email, chat platforms, webhooks etc. CUSTOMER will provide SMTP server details during OCP deployment, the same will be used to configure alert notifications.

**Note :-** RedHat OCP Monitoring solution only supports node exporters. Any customization is not supported.

### Storage Sizing

Based on the application workload details shared (covered in the Design Workshop section) 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 different nodes in production for HA.. Below is the calculation for prometheus storage , considering the monitoring solution for 7 days data :

Monitoring:

Per day Storage =~13 Gb

No of days to retain = 15 days

Total Storage required=~ 2 X 195 GB (for 2 Prometheus pods)

Alert manager runs with 2 pods by default. Alert manager storage = 10 Gb per pod Storage required = ~2 \* 10GB

\*\*\* 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 CUSTOMER team post go live if required.*

## 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:

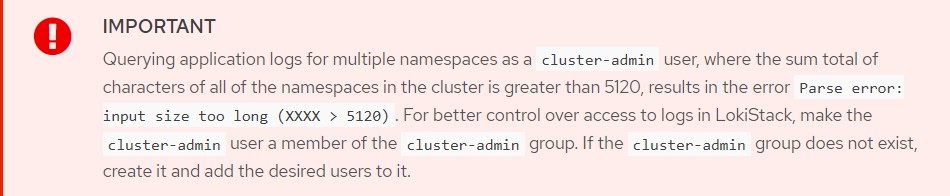
* A high scale log collection system
* Security Information and Event Monitoring (SIEM) compliant
* Historical or long term log retention or storage
* A guaranteed log sink
* Secure storage - audit logs are not stored by default

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.

* 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.

### 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 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 GP.
      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. CUSTOMER has to decide on the external storage options.



### Design Decisions

* + - 1. The Loki stack will be deployed on master nodes. For more detail please refer to the Red Hat documentation3.
      2. Loki stack needs to be configured for both applications and operations.
      3. OCP provides internal logging for applications hosted on OCP which is ideal for short-term logs, CUSTOMER team need to store the logs for a longer term than configured time period & Store Application & Audit logs in another Remote logging server. CUSTOMER Team will provide external logging solutions compatible with syslog or EFK where vector will forward the logs from OCP to the external log store.
      4. Vector will forward the logs to internal Loki storage and also forward the same to the external log store.
      5. Vector configuration for 1x.medium4 will be used and the operator will use the resource as per the specification.
      6. Persistent Volumes required for these Openshift Logging should be object storage.
      7. The threshold of 70% storage consumption has been considered for storage calculation as a best practice. Please refer to Red Hat documentation5 for more detail.



1. Any customization in the RedHat Logging solution is not supported.

3 Loki installation reference:

<https://docs.openshift.com/container-platform/4.14/logging/log_storage/cluster-logging-loki.html>

4

[https://access.redhat.com/documentation/en-us/openshift\_container\_platform/4.14/html/logging/log-storag](https://access.redhat.com/documentation/en-us/openshift_container_platform/4.14/html/logging/log-storage#installing-log-storage-loki) [e#installing-log-storage-loki](https://access.redhat.com/documentation/en-us/openshift_container_platform/4.14/html/logging/log-storage#installing-log-storage-loki)

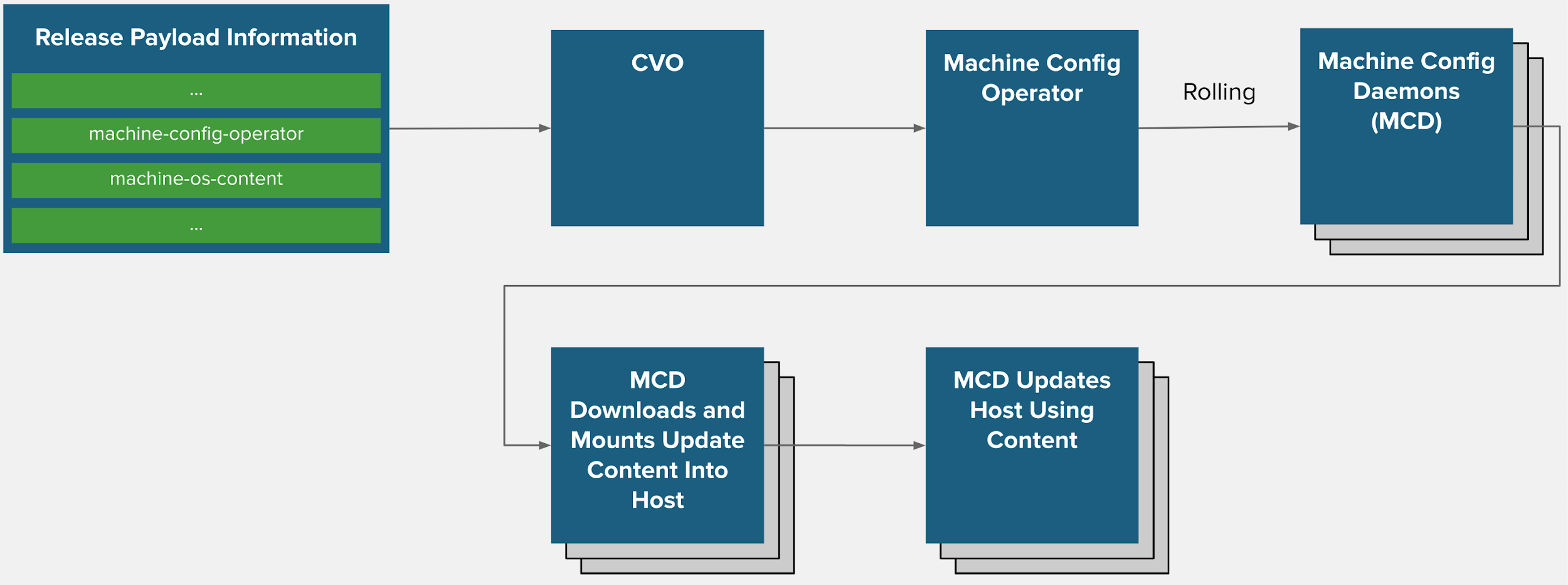
5 Storage threshold:

[https://access.redhat.com/documentation/en-us/openshift\_container\_platform/4.14/html-single/logging/index#cluster-logging-de](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) [ploy-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)

## Other considerations

### 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 documentation6.



### Design Decisions

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

Note - There is no upgrade rollback option for Openshift Cluster

6 Updating OCP 4.12 cluster:

<https://access.redhat.com/documentation/en-us/openshift_container_platform/4.14/html/updating_clusters/index>

## Sizing Considerations

OpenShift cluster limits are well documented at Red Hat customer portal7 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 factor8 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 510. 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.
* During the design discussion, CUSTOMER asked to increase the Max pods Per node to 500 instead of default 250. We will be changing this parameter during the cluster installation.
* To accommodate 500 pods per node, the subject assigned to each node for cluster network should also be considered with buffer IP’s. We agreed to assign /22 subnet for each node in openshift from the private cluster network.

## Authentication, Authorization & 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

7 Cluster limits:

[https://access.redhat.com/documentation/en-us/openshift\_container\_platform/4.14/html-single/scalability\_and\_per](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) [formance/index#planning-your-environment-according-to-object-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)

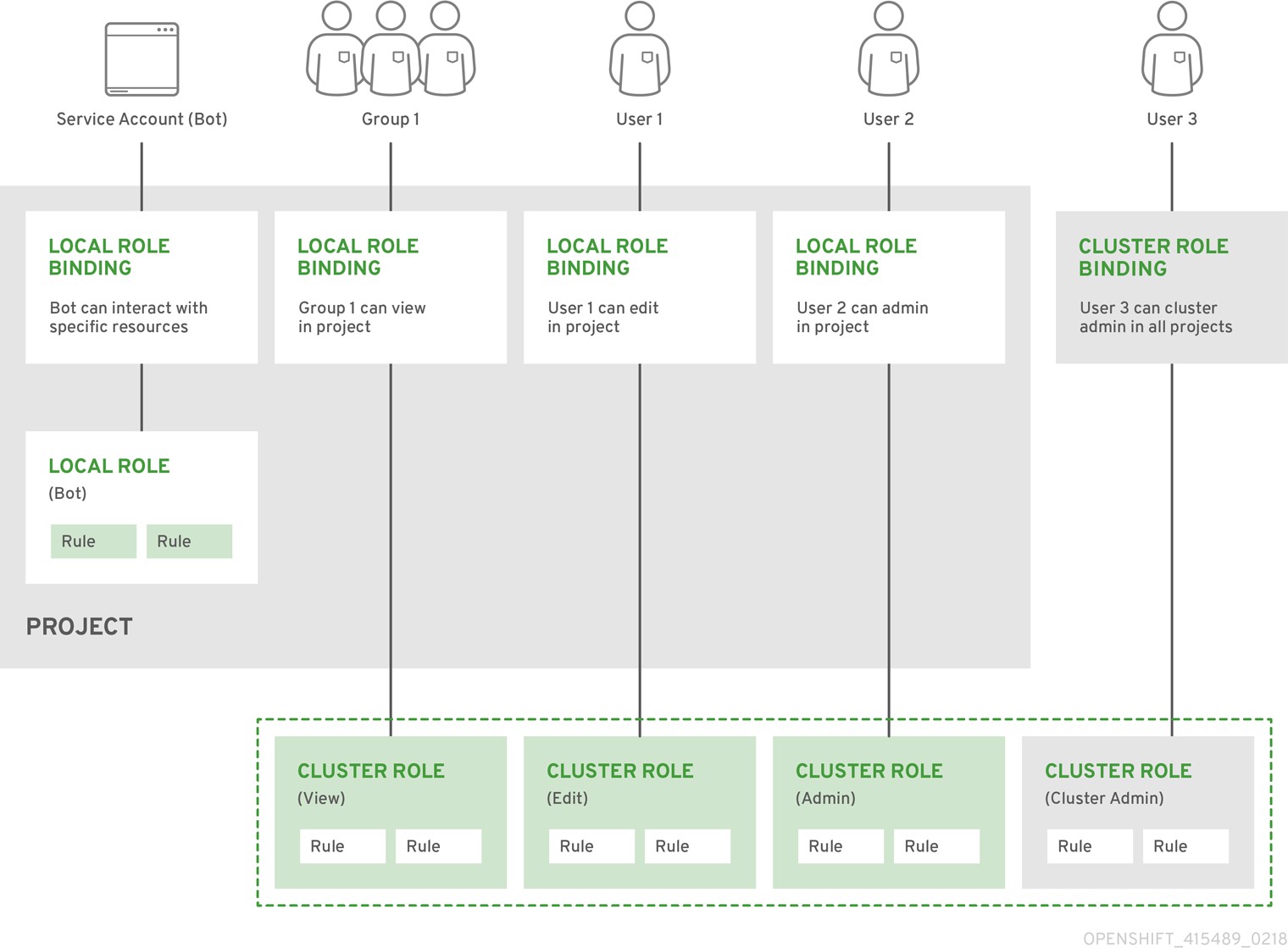
8 12 factor app principles: <https://12factor.net/>

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

**Design Decision -** CUSTOMER wants to Integrate RHOCP authentication with their Existing LDAP Server. Details of LDAP Server are added in the pre-req sheet.

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 documentation9.

### Key observation/challenges/requirements

1. The CUSTOMER team will authenticate the cluster via LDAP & HTpasswd both.

9 Authentication and authorisation:

[https://access.redhat.com/documentation/en-us/openshift\_container\_platform/4.14/html-single/authentication\_and](https://access.redhat.com/documentation/en-us/openshift_container_platform/4.14/html-single/authentication_and_authorization/index#understanding-authentication)

[\_authorization/index#understanding-authentication](https://access.redhat.com/documentation/en-us/openshift_container_platform/4.14/html-single/authentication_and_authorization/index#understanding-authentication)

### Design Decisions

1. Token, grant and session configuration options will be left to the defaults. This can be configured 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. Administrators can create new SCC based on the project's needs, post installation. For more detail refer to documentation10.

# 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 FortyADC Software load balancer. |

10 Volume security:

[https://access.redhat.com/documentation/en-us/openshift\_container\_platform/4.14/html-single/authentication\_and](https://access.redhat.com/documentation/en-us/openshift_container_platform/4.14/html-single/authentication_and_authorization/index#managing-pod-security-policies)

[\_authorization/index#managing-pod-security-policies](https://access.redhat.com/documentation/en-us/openshift_container_platform/4.14/html-single/authentication_and_authorization/index#managing-pod-security-policies)

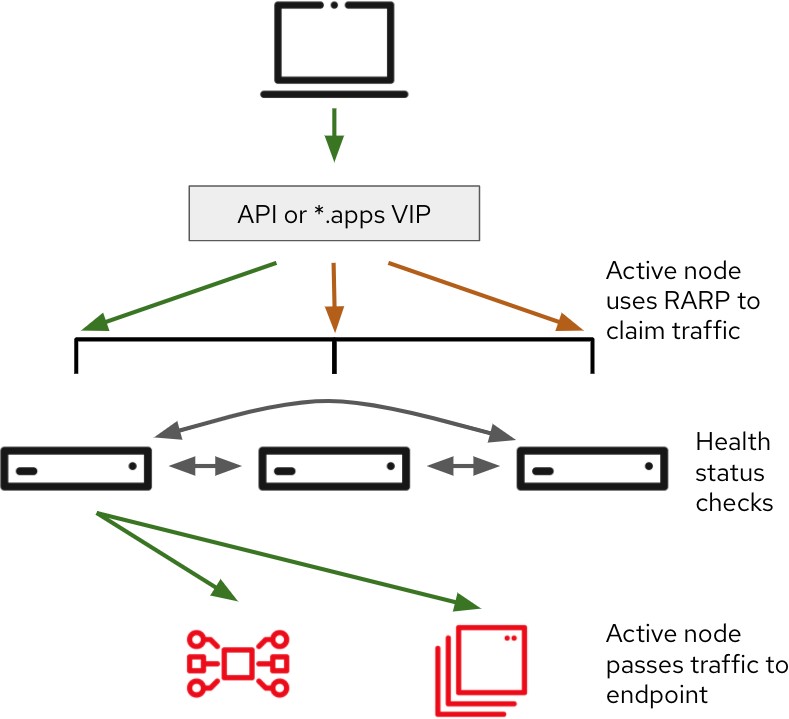
|  |  |  |
| --- | --- | --- |
| etcd | Active-Active | Fully redundant deployment with 3 nodes, 1 as leader and others nodes as followers. |
| Controller Manager | Active-Passive | Fully redundant deployment with 3 nodes, 1 as leader and others nodes as followers. |
| Scheduler | Active-Passive | Fully redundant deployment with 3 nodes, 1 as leader and others nodes as followers. |

## 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.

## 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.

## 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.

### Design Decisions

* + - * The CUSTOMER team has decided to keep MastersSchedulable disabled & run Application pod workload on worker nodes only.
      * OCP Infra & ODF Pods will be running on Master Nodes.

## Network connection redundancy

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

# Implementation Prerequisites

CUSTOMER 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.

## Red Hat subscriptions

CUSTOMER 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 Integration11.

## Hardware requirements

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Node Type | Node | OS | VCPU  per Node | RAM  per Node | Min Root disk per  Node (GB) | NIC Ports per Node |
| Master Node | 3 | RHCOS | 44 | 780G B | 300GB (Raid 1) | 4 |
| Workers | 10 | RHCOS | 88 | 1536  GB | 300GB (Raid 1) | 4 |
| Bootstrap(VM) | 1 | RHCOS | 8 | 16GB | 300GB | 1 required |
| Bastion(VM) | 1 | RHEL 9.x | 4 | 8GB | 300GB | 1 required |

Please note:

1. RHCOS installation for Bootstrap, Control plane & Worker nodes will be done by UPI method.
2. Worker node specification are different as there are 2 different hardware models being deployed.

11 Tested Integration - <https://access.redhat.com/articles/4128421>

## Network Access Requirements

A shared network must exist between the master, worker, ODF and other nodes (bastion, bootstrap, ceph grafana Dashboard)

Key activities are:

* + - Subnets Required:

1. Hosts Subnet

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

/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

/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

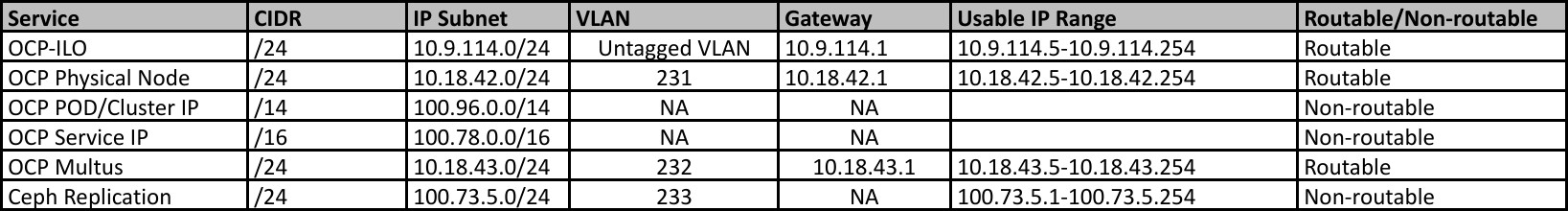
1. Storage Subnet

The subnet used on the secondary interfaces of the hosts for the ceph storage network.

These bonds will be created with the parameters described in the table below.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Interface | Bond Type | Slave Interface | Speed per NIC | Network Type | Network |
| bond0 | LACP | <<TBF>> | 10GB | L3 | Machine Network |
| bond1 | LACP | <<TBF>> | 10GB | <<TBD>> | Multus Network ( Reserved For Future use) |

Below are the network details:



* + - Ports to be opened:

|  |  |  |  |
| --- | --- | --- | --- |
| Machine | Protoco l | Port | Description |
|  |  | 2379-2380 | etcd server, peer, and metrics ports |
|  |  | 6443 | Kubernetes API |
|  |  |  | Host level services, including the |
|  |  | 9000-9999 | node exporter on ports 9100-9101  and the Cluster Version Operator on |
|  |  |  | port 9099 |
|  | TCP |  |  |
| 10249-10259 | The default ports that Kubernetes reserves |
|  |  | 1936 | Node Metrics Collection Port |
| All machines <--> |  |  |
|  | Kubernetes NodePort |
| all machines |  | 30000-32767 |
|  |  | 10256 | openshift-sdn |
|  |  | 4789 | VXLAN and GENEVE |
|  | UDP | 6081 | VXLAN and GENEVE |
|  | Host level services, including the |
|  |  | 9000-9999 | 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 LDAP server, SMTP, NTP, NFS, SSH,  Syslog, DNS | TCP | LDAP server,  SMTP, NTP,  NFS, SSH,  Syslog,DNS | Infrastructure services |
| OCP Machines to Ceph Machines | TCP | 3300, 6789,  6800-7300 | Ceph OSD & Ceph |

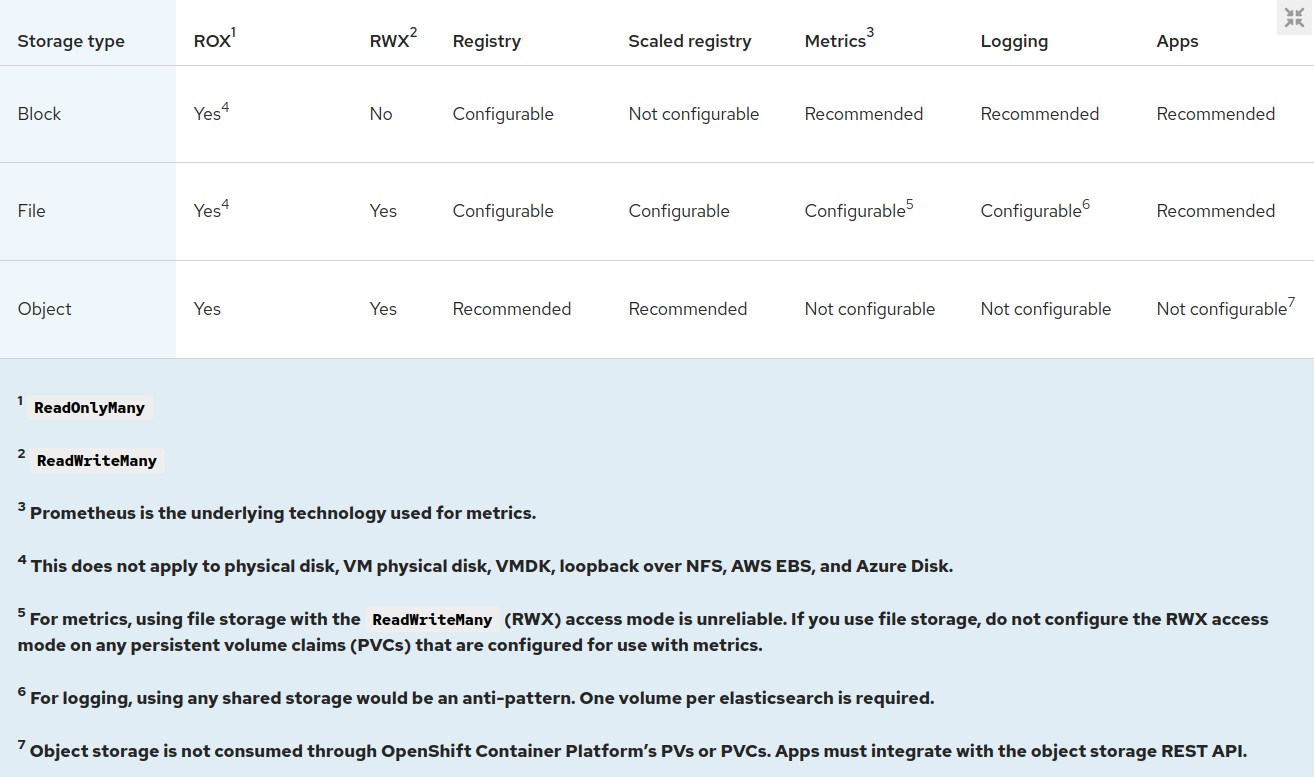
Detailed steps are covered in the RHOCP documentation42.

## Persistence Storage

Following OCP components require storage are as below:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Logging** | | | | | |
| OCP component | Disk  Type | Size (TB) | Access  Type | Remarks | **No. of Pods** |
| Loki-Backend | Object | 1TB | RWX | 15 days considering | 2 |
| storage-logging-loki-c  ompactor | Block | 10GB | RWO | 1x10GB | 1 |
| storage-logging-loki-in  dex-gateway | Block | 100GB | RWO | 2x50GB | 2 |
| storage-logging-loki-in  gester | Block | 30GB | RWO | 3x10GB | 3 |
| wal-logging-loki-ingest  er | Block | 450GB | RWO | 3x150GB | 3 |
|  |  |  |  |  |  |
| OCP component | Disk  Type | Size (TB) | Access  Type | Remarks | **No. of Pods** |
| Monitoring-  Prometheus | Block | 200GB | RWO | 200 GB (100x2) | 2 |
| Monitoring-Alert  Manager | Block | 20GB | RWO | 20 GB (10x2) | 2 |
| Internal Registry | Object | 1TB | RWX | 1024 GB | 1 |
| Compliance Operator | Block | 50GB | RWO | 50 GB (50 x 1) | 1 |
| ETCD Backup | NFS | 200GB | NFS | 200 GB | 1 |

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



## Internet Access

* + - Openshift Deployment will be a connected deployment.
    - HTTP Proxy must be whitelisted for below URLs for all nodes in Openshift including Bastion and Bootstrap:

|  |  |
| --- | --- |
| URL | Port |
| [registry.redhat.io](http://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](https://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, 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-](http://rhc4tp-prod-z8cxf-image-registry-us-east-1-evenkyleffocxqvofrk.s3.dualstack.us-east-1.amazonaws.com/)  [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 |
| [download.sonatype.com/nexus/3/](https://download.sonatype.com/nexus/3/) | 443 |

## 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

## Certificates

CUSTOMER confirmed that self signed certificates to be used.

## 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 documentation12. Note :- DNS requirements are part of a separate pre-requisite excel document

## Miscellaneous

### NTP synchronization

Red Hat recommends keeping all the hosts in sync via multiple NTP servers. CUSTOMER will provide 4x 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.

### 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. CUSTOMER Team has confirmed to take etcd backup and store in NFS Server. Details are added in the Pre-Req sheet.

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\_an](https://docs.openshift.com/container-platform/4.14/backup_and_restore/control_plane_backup_and_restore/backing-up-etcd.html) [d\_restore/backing-up-etcd.html](https://docs.openshift.com/container-platform/4.14/backup_and_restore/control_plane_backup_and_restore/backing-up-etcd.html)

12 DNS requirements(Section 23.1.5):

[https://access.redhat.com/documentation/en-us/openshift\_container\_platform/4.14/html/installing/installing-on-an](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) [y-platform#installation-user-provisioned-validating-dns\_installing-platform-agnostic](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)

## Software Versions

|  |  |
| --- | --- |
| Product | Version |
| OpenShift Container Platform | 4.14 |
| OpenShift Data Foundation | 4.14 |
| Red Hat Enterprise Linux | 9.x |

* 1. Hardware Supportability.

|  |  |  |  |
| --- | --- | --- | --- |
| **Server Role** | **Hardware** | **No of Nodes** | **Hardware Supportability Matrix** |
| Master | HPE ProLiant DL360  Gen10 | 3 | https://catalog.redhat.com/hardware/servers/deta  il/6589 |
| Worker | HPE ProLiant DL560 Gen10 | 10 | https://catalog.redhat.com/hardware/servers/deta il/6449 |

## Red Hat Advanced Cluster Security:

During the design discussion, CUSTOMER asked to install RHACS on the cluster which would self manage itself. CUSTOMER also requested to create a gatekeeper policy on RHACS to ensure Selinux is enforced for all RWX persistent Volumes.