OPENSHIFT

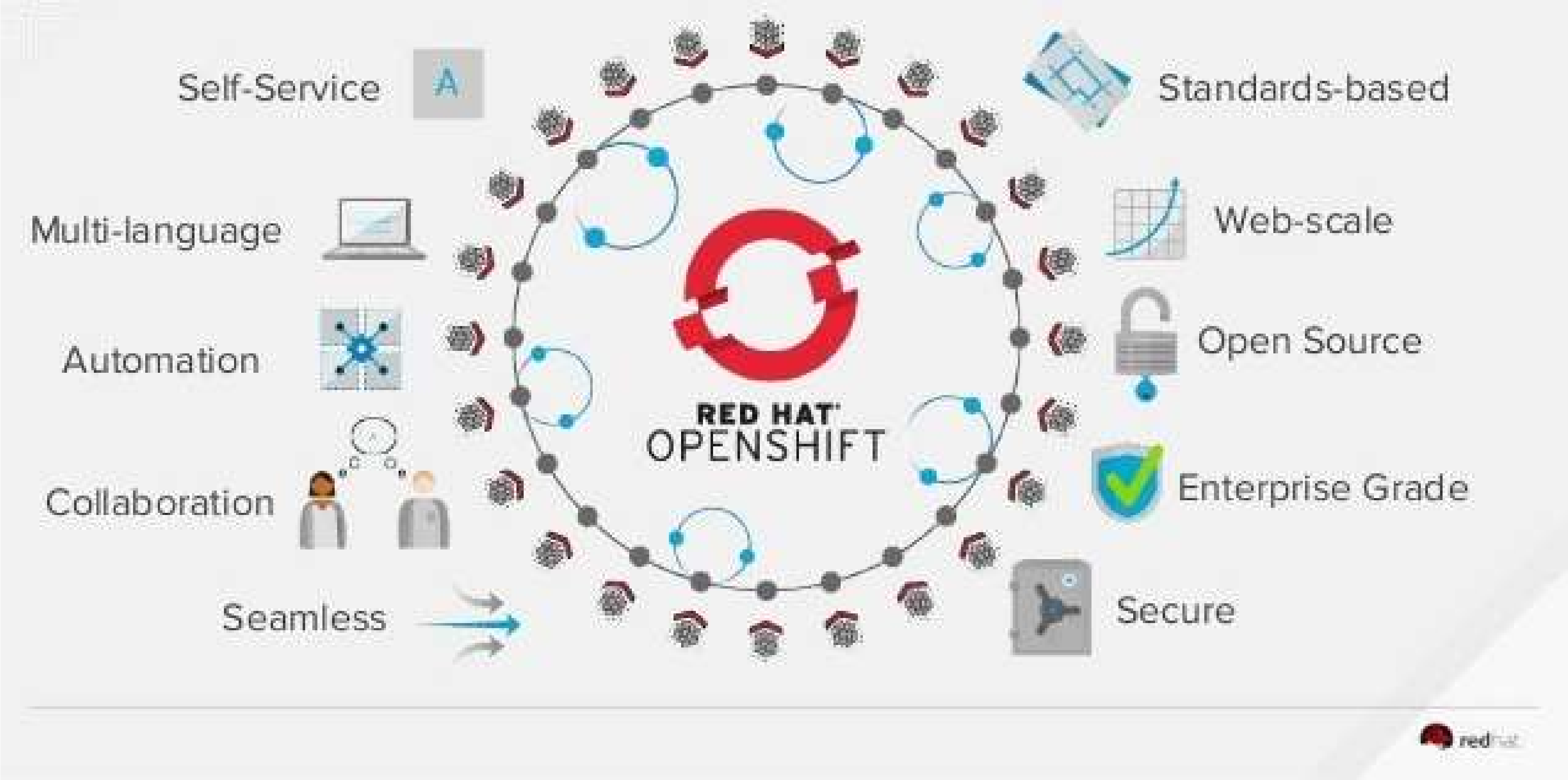
What is Red Hat OpenShift?

Red Hat OpenShift is an enterprise-ready Kubernetes container platform

Provide a developer-friendly environment with integrated tools for

- Building, deploying, and scaling containerized applications.

Features


Self-Service
Multi-language
Automation
Collaboration
Seamless
RED HAT OPENSHIFT
Standards-based
Web-scale
Open Source
Enterprise Grade
Secure
redhat

Benefits of OpenShift

Simplified Management

- Automates deployment, scaling, and operations of application containers

Flexibility

- Supports hybrid and multi-cloud environments

Enhanced Security

- Built-in security features and compliance controls

Core Components of OpenShift

Kubernetes

- Container orchestration engine

OpenShift Container Platform

- A fully supported Kubernetes platform with additional features like developer-friendly tooling and enterprise-grade security.

OpenShift Origin (OKD)

- The open-source upstream project for Red Hat OpenShift, providing a community-driven platform for container orchestration.

OpenShift CLI (oc)

- Command-line interface for managing applications

Developer Tools

- Web console and integrated development environment (IDE) support.

OpenShift Service Mesh

- Integrated service mesh based on Istio for managing microservices architecture.

Architecture Overview

Master Nodes:

- Responsible for managing the Kubernetes API, scheduling, and cluster management.

Worker Nodes:

- Run the containerized applications, managed by the master nodes.

ETCD:

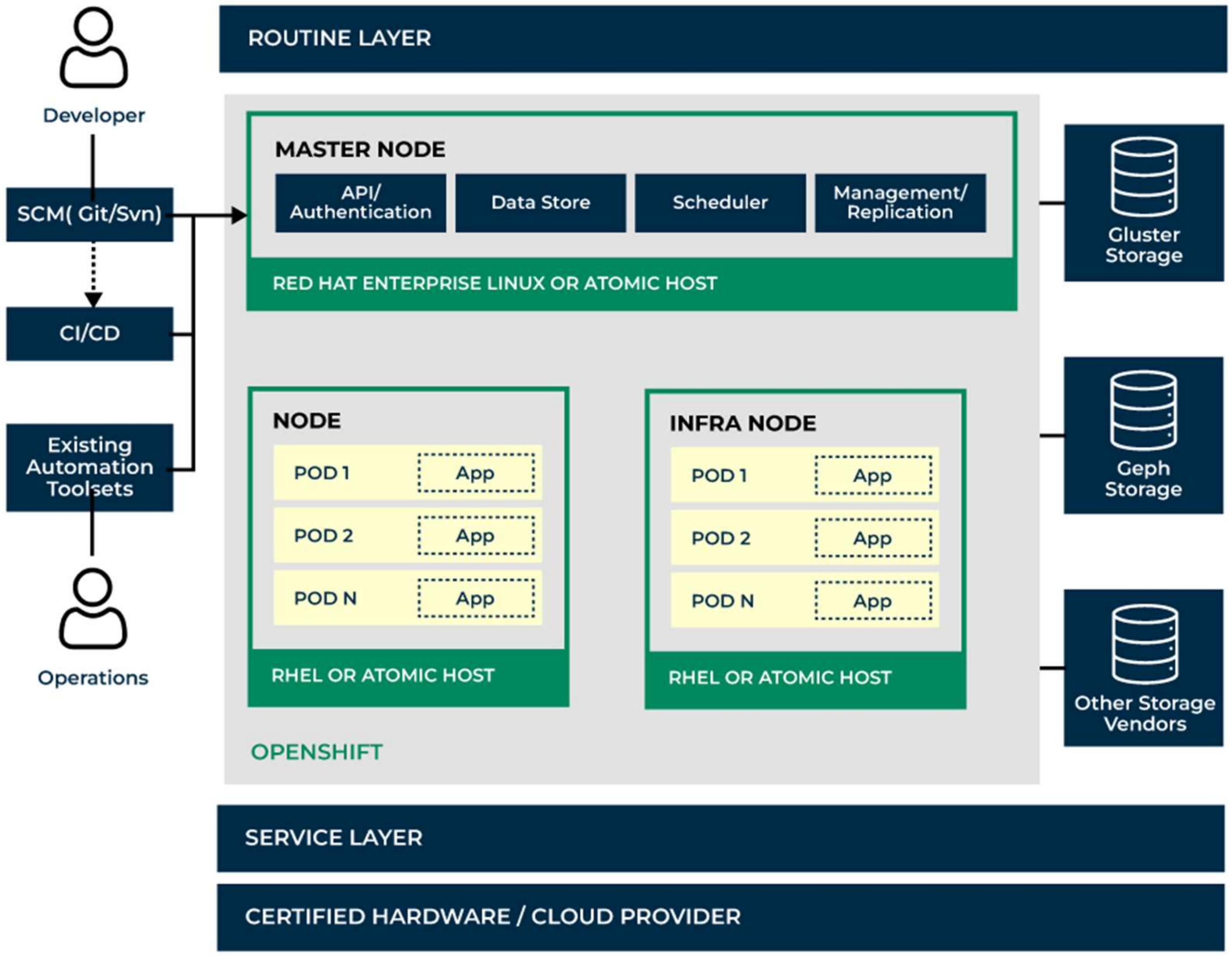
- Distributed key-value store for backing up all cluster data.

Networking and Routing:

- OpenShift provides an integrated SDN (Software Defined Network) and supports external networking plugins.

Persistent Storage:

- Allows storage options like NFS, GlusterFS, or cloud-based block storage for stateful applications.


Developer
SCM(Git/Svn)
CI/CD
Existing Automation Toolsets
Operations
ROUTINE LAYER
MASTER NODE
API/ Authentication
Data Store
Scheduler
Management/ Replication
RED HAT ENTERPRISE LINUX OR ATOMIC HOST
NODE
POD 1
App
POD 2
App
POD N
App
RHEL OR ATOMIC HOST
INFRA NODE
POD 1
App
POD 2
App
POD N
App
RHEL OR ATOMIC HOST
OPENSHIFT
Gluster Storage
Geph Storage
Other Storage Vendors
SERVICE LAYER
CERTIFIED HARDWARE / CLOUD PROVIDER

Installing OpenShift – Hardware Requirements

Refer:

- https://docs.RedHat.com/en/documentation/red_hat_OpenShift_local/2.41/html/getting_started_guide/index
- https://docs.RedHat.com/en/documentation/red_hat_OpenShift_local/2.41/html/getting_started_guide/installing#minimum_system_requirements

4 physical CPU cores

10.5 GB of free memory

35 GB of storage space

Installing OpenShift – Software Requirements

- Windows 10 or Microsoft Windows 11
- macOS 13 Ventura or later
- Red Hat Enterprise Linux
- CentOS 8 and 9
- Ubuntu 18.04 LTS or later and Debian 10 or later are not supported

Our Setup

- Standard D8s v3 (8 vcpus, 32 GiB memory)
- Rocky Linux OS 9.2
- AlmaLinux 9

Installing OpenShift - Download

- sudo yum update -y
- sudo dnf update -y
- sudo dnf install NetworkManager –y
- sudo yum install httpd-tools -y

- wget https://developers.RedHat.com/content-gateway/rest/mirror/pub/OpenShift-v4/clients/crc/latest/crc-linux-amd64.tar.xz

Installing OpenShift - Install

- tar -xvf crc-linux-amd64.tar.xz
- rm crc-linux-amd64.tar.xz
- mkdir -p ~/bin
- cp ~/crc-linux-*-amd64/crc ~/bin
- export PATH=$PATH:$HOME/bin
- echo 'export PATH=$PATH:$HOME/bin' >> ~/.bashrc
- mkdir ~/crc_backup
- cp -R ~/.crc ~/crc_backup

Installing OpenShift - Install

- crc config set network-mode user
- crc cleanup
- crc setup
- ls ~/.crc/cache
- mkdir -p ~/crc_backup
- cp ~/.crc ~/crc_backup

Installing OpenShift - Start

- crc config set pull-secret-file pull-secret.txt
- crc start --bundle ~/.crc/cache/crc_libvirt_4.16.7_amd64.crcbundle
- crc console --url
- crc console --credentials

Install oc cli

- sudo mkdir /ocp-tools
- sudo chmod 777 /ocp-tools
- cd /ocp-tools
- wget https://mirror.OpenShift.com/pub/OpenShift-v4/clients/ocp/stable-4.6/OpenShift-client-linux.tar.gz -P /ocp-tools
- ls -la /ocp-tools
- tar xvf OpenShift-client-linux.tar.gz oc kubectl
- sudo cp oc kubectl /usr/local/bin
- oc version
- kubectl version

Architecture

A Kubernetes cluster consists of two main components:

- Master (Control Plane)
- Worker Nodes.

Master has following components. These components are responsible for maintaining the state of the cluster:

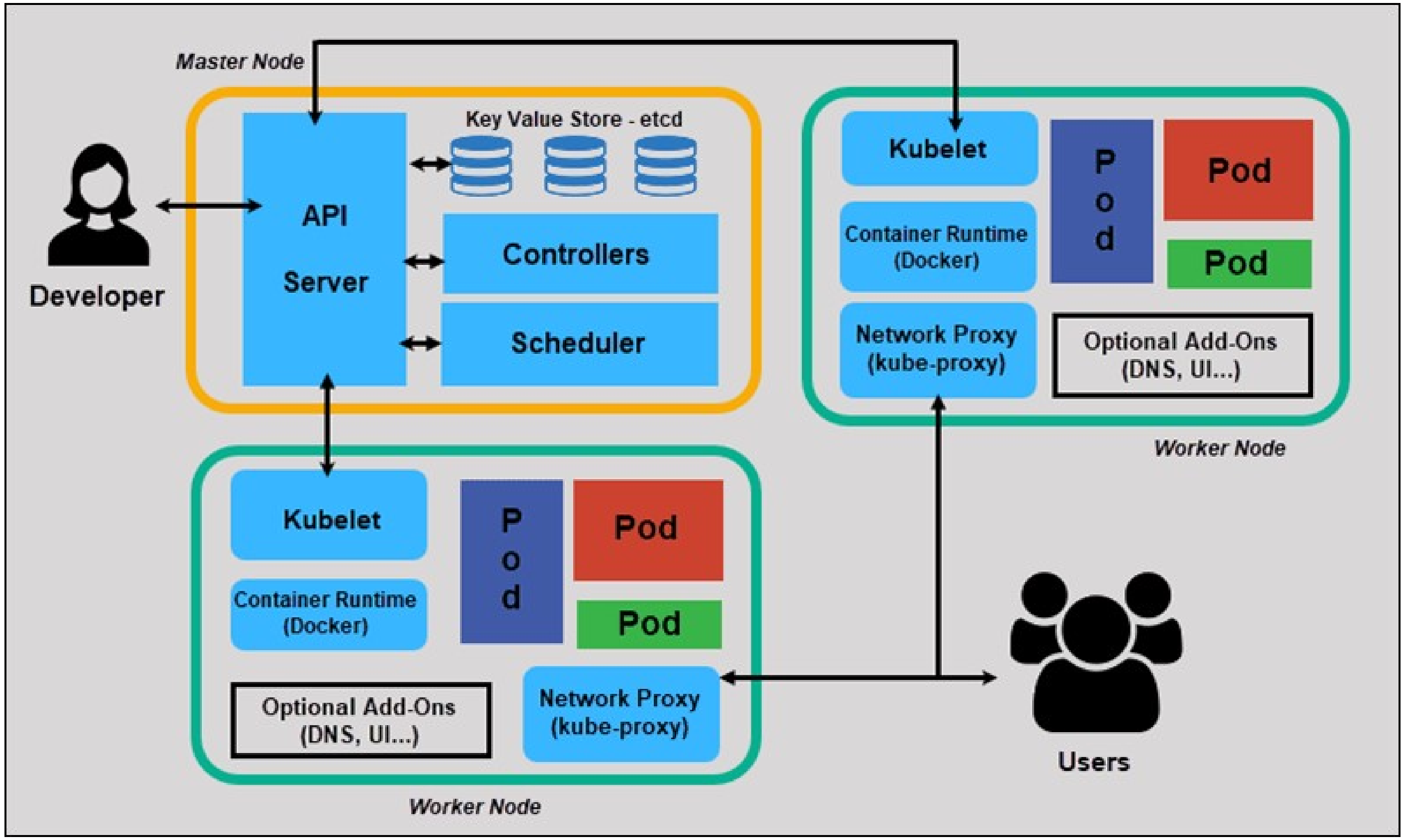
- etcd distributed key value store.
- API Server.
- Controller Manager
- Scheduler

Every worker node consists of the following components.

These components are responsible for deploying and running the application containers.

- Kubelet
- Container Runtime (Docker)

Kubernetes Architecture



Configuring OpenShift Cluster

- After installing OpenShift, several configurations are needed to ensure the platform operates optimally based on the environment.
- Key configuration areas:
  - User access and authentication
  - Networking
  - Storage
  - Resource limits and quotas
  - Monitoring and logging

User Access and Authentication

- OpenShift integrates with several identity providers (IDPs) for user authentication and access control.
- **Steps to Configure Authentication**:
  - Edit the OAuth configuration
    - Navigate to OpenShift Web Console → Administration → OAuth.
  - Select an IDP
    - Supported IDPs include **GitHub**, **Google**, **LDAP**, and **OpenID Connect**.
  - **Configure Role-Based Access Control (RBAC)**
    - Use OpenShift's RBAC to assign roles to users, controlling access to projects, nodes, and resources.
  - Example Command
    - oc adm policy add-cluster-role-to-user cluster-admin <username>

Cluster Management & Basic Commands

- Start the OpenShift Cluster: After installing OpenShift Local, you can start the cluster with:
    - crc start
    - crc stop
    - crc status
- Get OpenShift Web Console URL
    - When the cluster starts, it provides the web console URL. However, you can retrieve it later using:
        - crc console

Credentials Management

- Get Default User Credentials:
  - After starting the cluster with crc start, the credentials for the default administrator are typically displayed:
    - Kubeadmin user: kubeadmin
    - Kubeadmin password: <generated_password>
  - To retrieve the password later, you can run
    - crc console --credentials
  - Login to OpenShift using the oc command-line tool: Use the OpenShift client (oc) to interact with the cluster from the terminal.
    - oc login -u kubeadmin -p <password> https://api.crc.testing:6443

User Management

```
sudo yum install httpd-tools
htpasswd -c -B -b users.htpasswd amit amit
oc create user amit
oc adm policy add-cluster-role-to-user admin amit
oc create secret generic htpass-secret --from-file=htpasswd=users.htpasswd -n openshift-config
oc create identity allow_all:amit
oc create useridentitymapping allow_all:amit amit
oc get identity
cat users.htpasswd
oc get secret htpass-secret -n openshift-config
```

Common commands

- crc console --url
- crc console
- crc status
- oc new-project my-first-project
- oc new-app nginx --name=my-nginx-app
- oc expose svc/my-nginx-app
- oc get route my-nginx-app
- oc get pods -w
- oc status

Common commands

- oc get nodes
- oc get pods --all-namespaces
- oc logs <pod-name>
- oc get events
- oc exec <pod-name> -- <command>
  - oc exec my-nginx-app-xyz123 -- /bin/bash
- oc describe pod <pod-name>
- oc scale --replicas=3 deployment/my-nginx-app

Common commands

- oc get routes
- oc debug pod/<pod-name>
- oc get svc

Thanks