**OpenShift Container Platform Archiceture:-**

Red Hat OpenShift Container Platform (RHEL OCP) is an enterprise Kubernetes platform built on Red Hat Enterprise Linux (RHEL) and Kubernetes, designed to manage containerized applications at scale. Below is a brief overview of its core components:

1. **Master Components (Control Plane)**

These manage the OpenShift cluster:

* API Server: Frontend for the Kubernetes control plane; all cluster operations go through this.
* Controller Manager: Maintains desired state (e.g., ensuring a pod is running if declared).
* Scheduler: Assigns pods to nodes based on resource availability.
* etcd: Key-value store holding cluster state and configuration (highly available and backed up).

1. **Worker Nodes (Compute Nodes)**

These run your application workloads:

* **Kubelet:** Agent on each node to communicate with the control plane and manage containers.
* **CRI-O or containerd:** Lightweight container runtime used instead of Docker.
* **OpenShift SDN / OVN-Kubernetes:** Provides pod-to-pod network communication and network isolation (network plugins).

**3. OpenShift-Specific Components**

* **Machine API:** Manages infrastructure provisioning (e.g., auto-scaling worker nodes).
* **Cluster Version Operator (CVO):** Automates upgrades and updates of the cluster components.
* **Operators:** Extend Kubernetes functionality used for lifecycle management (e.g., etcd, monitoring, storage).
* **OAuth Server:** Handles authentication using identity providers (LDAP, GitHub, etc.).
* **Ingress Controller (Router):** Manages external access to services using routes (based on HAProxy).

**4. Developer-Focused Tools**

* **Developer Console**: Web UI for deploying apps, monitoring, and CI/CD integration.
* **Source-to-Image (S2I):** Builds container images directly from source code.
* **BuildConfig & ImageStreams:** Used to define and automate application builds and deployments.

**5. Monitoring and Logging**

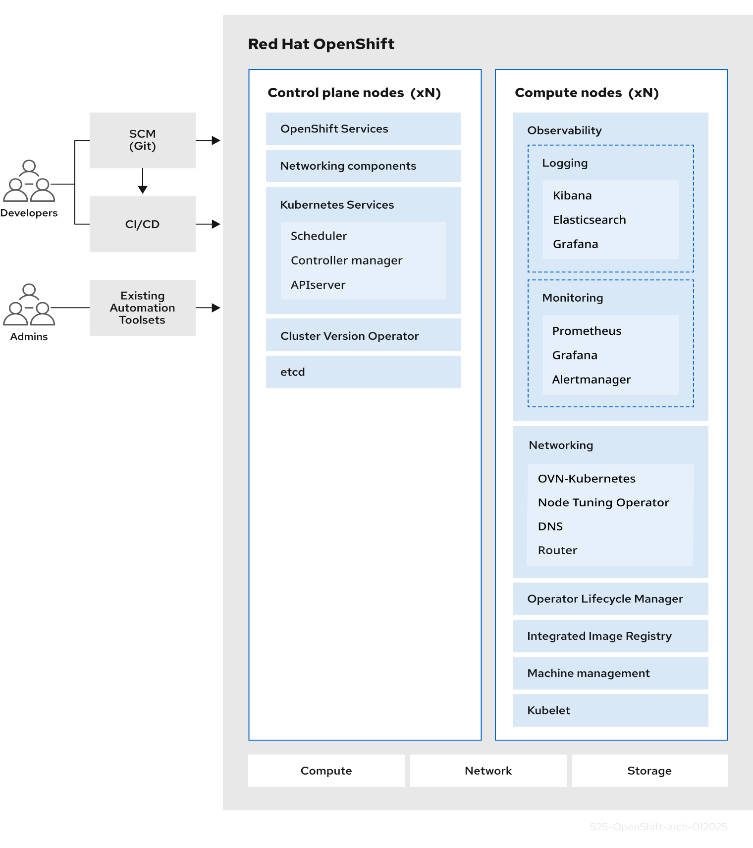
 **Prometheus & Grafana:** Cluster metrics and performance monitoring.

 **Alertmanager:** Sends alerts based on Prometheus rules.

 **Elasticsearch & Fluentd:** Log aggregation and search.

**6. Storage & Persistent Volumes**

* **Persistent Volume (PV) / Persistent Volume Claim (PVC)**: Kubernetes standard for managing persistent storage.
* **OpenShift Container Storage (OCS)**: Integrated storage solution using Ceph.

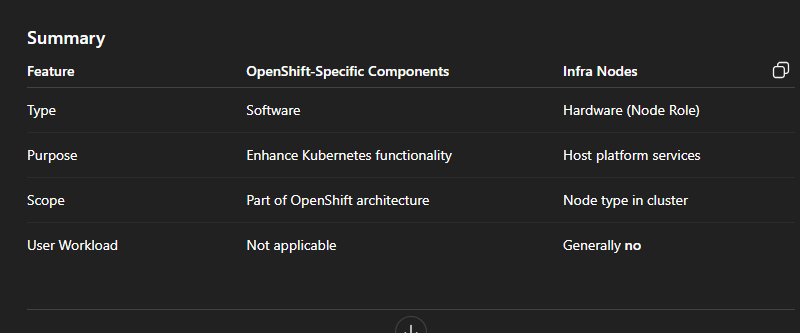


**Infrastructure Nodes (Infra Nodes)**

These are dedicated nodes in the OpenShift cluster that host infrastructure workloads only, such as:

* Ingress (Router)
* Logging (Fluentd, Elasticsearch)
* Monitoring (Prometheus, Alertmanager)
* Internal container registry
* OAuth server

Infra nodes do not run user applications (Pods) unless explicitly allowed.



**Routes:-**

In OpenShift, a Route is a way to expose a Service (usually a web application running inside the cluster) to the external world using HTTP, HTTPS, or TLS.

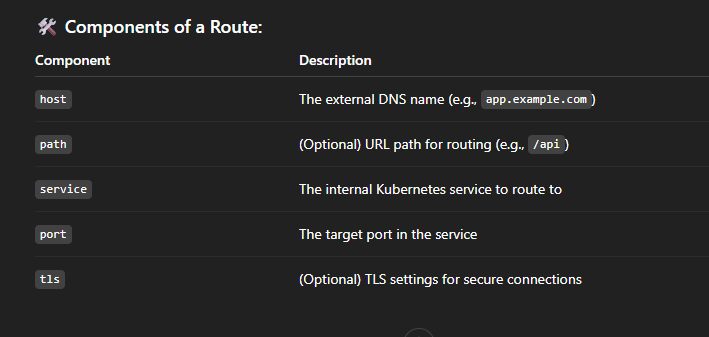
It acts like a reverse proxy and ingress controller, directing external traffic to the appropriate pods inside the cluster.

**Key Concepts:**

* A **Route** maps an external **hostname (URL)** to a **Service** in OpenShift.
* Built on top of the **HAProxy** router by default.
* Supports **path-based routing**, **TLS termination**, **wildcard routes**, and **custom certificates**.

**Route Architecture:**

User --> OpenShift Router (HAProxy) --> Service --> Pod

****

**Use Cases:**

* Exposing web apps to the internet (e.g., frontend UI)
* Hosting multiple apps with different paths or domains
* Terminating SSL/TLS at the edge
* Creating custom domain names for internal/external access

**Workloads**:- pods, Deployments, Deploymentconfigs, Statefulsets, Daemonsets, jobs, cronjobs, Replicasets, ReplicationControllers, Secrets and ConfigMaps.

**Operators:-**

In **OpenShift**, an **Operator** is a method of **packaging, deploying, and managing Kubernetes applications**. Operators extend Kubernetes capabilities by automating **Day-1 and Day-2 operations** — like installation, upgrades, backup, failover, and more.

Operators are built using **Kubernetes Custom Resource Definitions (CRDs)** and **controllers** to manage complex workloads **as code**.

**Why Operators?**

Traditional YAML manifests work for deployment, but **don’t handle ongoing lifecycle tasks** like:

* Auto-healing
* Configuration drift
* Seamless upgrades
* Custom metrics checks
* Dependency management

Operators fill this gap by **encapsulating operational knowledge into code**.

**How Operators Work:**

1. Define a **Custom Resource Definition (CRD)**: Defines a new Kubernetes object (e.g., MyDatabase).
2. Operator watches that resource.
3. Based on changes or states, it takes **automated actions** (create pods, upgrade app, reconfigure settings, etc.).

**Operator Lifecycle Manager (OLM)**

**OLM** is a built-in component in OpenShift that:

* Manages installation and upgrades of Operators.
* Provides a **catalog of Operators** via OperatorHub.
* Handles **dependencies** between Operators.
* Supports multi-tenancy (cluster-scoped vs. namespace-scoped Operators).

You can access OLM via:

* **Web Console**: OperatorHub → Install → Configure
* **CLI**: oc apply with Subscription, OperatorGroup, etc.

**Ignition**

A utility that RHCOS uses to manipulate disks during initial configuration. It completes common disk tasks, including partitioning disks, formatting partitions, writing files, and configuring users.

**Machine Config Daemon (MCD)**

A daemon that regularly checks the nodes for configuration drift.

**Machine Config Operator (MCO)**

An Operator that applies the new configuration to your cluster machines.

**Machine Config Pools (MCP)**

A group of machines, such as control plane components or user workloads, that are based on the resources that they handle.

**mirror registry**

A registry that holds the mirror of OpenShift Container Platform images.

**OpenShift CLI (oc)**

A command-line tool to run OpenShift Container Platform commands on the terminal.

**OpenShift image registry**

A registry provided by OpenShift Container Platform to manage images.

**OperatorHub**

A platform that contains various OpenShift Container Platform Operators to install.

**Replication controllers**

An asset that indicates how many pod replicas are required to run at a time.

**Role-based access control (RBAC)**

A key security control to ensure that cluster users and workloads have only access to resources required to execute their roles.

**Route**

Routes expose a service to allow for network access to pods from users and applications outside the OpenShift Container Platform instance.

**Telemetry**

A component to collect information such as size, health, and status of OpenShift Container Platform.

**Template**

A template describes a set of objects that can be parameterized and processed to produce a list of objects for creation by OpenShift Container Platform.