***Kubernetes dashboard***

# **Introduction:**

* Kubernetes Dashboard is a web-based graphical user interface (GUI) for managing and monitoring Kubernetes clusters. It provides a visual representation of the Kubernetes resources running in the cluster and allows users to interact with them through a browser.
* The Kubernetes Dashboard allows users to view the overall health and status of the cluster, manage deployments, view logs, and configure various settings. It also provides a way to manage and monitor applications running on the cluster, including the ability to scale them up or down and perform rolling updates.
* The Dashboard is built on top of Kubernetes APIs and is integrated with the RBAC (Role-Based Access Control) system, which allows users to log in with their Kubernetes credentials and only access resources that they have permission to view or modify.
* The Dashboard can be installed in a Kubernetes cluster using the **kubectl** command line tool, and can also be accessed from outside the cluster using a proxy or by exposing it through a load balancer.
* Overall, the Kubernetes Dashboard provides an easy-to-use, visual interface for managing and monitoring Kubernetes clusters, making it a valuable tool for both developers and system administrators.

# **Project Summary:**

|  |  |
| --- | --- |
| Website | <https://github.com/kubernetes/dashboard>. <https://kubernetes.io/>. |
| Organization/Foundation Name | Cloud Native Computing Foundation (CNCF). |
| License | Kubernetes is released under the Apache License, Version 2.0. |

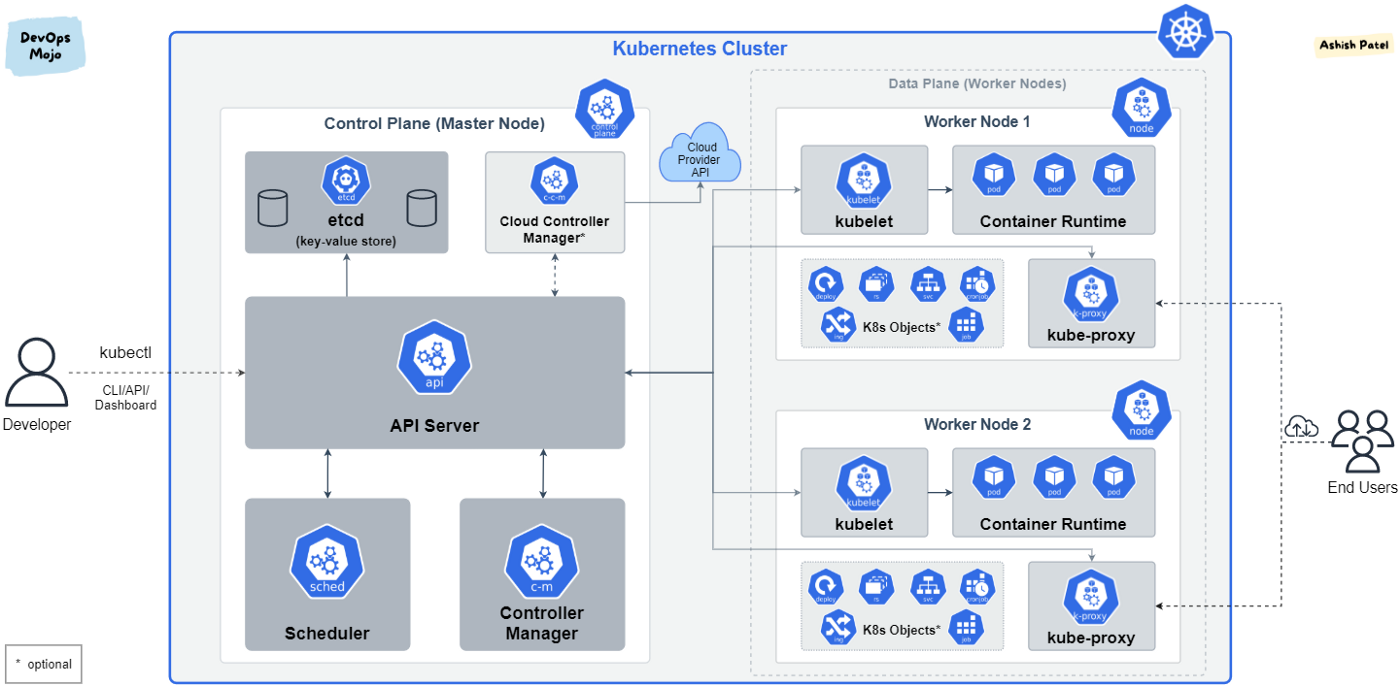
|  |  |
| --- | --- |
| Open/Proprietary | Kubernetes is an open-source project. |
| Source Path(if open source) | The source code for Kubernetes can be found on GitHub at <https://github.com/kubernetes/kubernetes> |
| Brief Description | * Kubernetes is an open-source container orchestration platform that automates the deployment, scaling, and management of containerized applications. * It provides a platform-independent API for managing containerized applications, enabling applications to be deployed and scaled across multiple environments with ease. * Kubernetes has gained significant popularity due to its flexibility, scalability, and community support, making it a popular choice for managing containerized workloads in production environments. * The source code for Kubernetes is available on GitHub, and the project is managed by the Cloud Native Computing Foundation (CNCF). |

# **Project Details :**

# **Key Features :**

1. Container orchestration: Kubernetes automates the deployment, scaling, and management of containerized applications, providing a platform-independent API for managing containers.
2. Automatic load balancing: Kubernetes automatically distributes incoming network traffic to healthy containers in a service or deployment, ensuring optimal resource utilization and minimal downtime.
3. Scalability: Kubernetes allows for horizontal scaling of containerized applications, automatically adding or removing containers based on resource utilization and demand.
4. Self-healing: Kubernetes automatically detects and replaces failed containers, ensuring that applications remain highly available and resilient to failures.
5. Configurability: Kubernetes provides a rich set of configuration options, allowing users to customize their deployment and management strategies to suit their specific needs.
6. Rolling updates: Kubernetes enables rolling updates of applications, allowing for seamless updates and minimal downtime.
7. Resource management: Kubernetes provides detailed resource utilization metrics and allows for resource limits and requests to be set for containers, ensuring optimal resource allocation and utilization.

* **Architecture:**



[Kubernetes (K8s)](https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/) is an open-source system for automating deployment, scaling, and management of containerized applications. Kubernetes abstracts away complex container management and

provides us with declarative configuration to orchestrate containers in different compute environments.

# Kubernetes Architecture

Kubernetes deployment is called a cluster. A Kubernetes cluster consists of at least one main (control) plane, and one or more worker machines, called [nodes](https://kubernetes.io/docs/concepts/architecture/nodes/). Both the control planes and node instances can be physical devices, virtual machines, or instances in the cloud.

## Control Plane

* Also known as master node or head node.
* The control plane manages the worker nodes and the Pods in the cluster.
* In production environments, the control plane usually runs across multiple computers and a cluster usually runs multiple nodes, providing fault-tolerance and high availability.
* The control plane receives input from a CLI or UI via an API.
* It is not recommended to run user workloads on master mode.

## Node(s)

* Also known as worker node or compute node.
* A virtual or physical machine that contains the services necessary to run containerized applications.
* A Kubernetes cluster needs at least one worker node, but normally have many.
* The worker node(s) host the Pods that are the components of the application workload.
* Pods are scheduled and orchestrated to run on nodes.
* You can scale up and scale down cluster by adding and removing nodes.
* **Current Usage :**

Kubernetes is widely used in production environments across a range of industries, including technology, finance, healthcare, and government. According to a recent survey by the Cloud Native Computing Foundation (CNCF), over 90% of respondents reported using Kubernetes in production, with adoption rates continuing to grow year over year.

Some common use cases for Kubernetes include:

1. Microservices: Kubernetes provides a platform for deploying and scaling microservices-based applications, enabling organizations to break down monolithic applications into smaller, more manageable services.
2. Hybrid cloud: Kubernetes provides a consistent platform for deploying and managing applications across hybrid cloud environments, enabling organizations to run workloads in on-premises data centers, public clouds, and edge locations.
3. Machine learning and AI: Kubernetes provides a platform for deploying and scaling machine learning and AI workloads, enabling organizations to build and deploy machine learning models at scale.
4. High-performance computing: Kubernetes can be used to orchestrate high-performance computing workloads, such as scientific simulations and data analysis, providing a highly scalable and flexible platform for compute-intensive applications.

* **Technical Details :**
* Community: Kubernetes has a large and active community of developers and contributors from around the world. The project is hosted by the Cloud Native Computing Foundation (CNCF), a non-profit organization that also hosts other cloud-native projects like Prometheus and Envoy.
* Certification: The CNCF offers a Kubernetes certification program for developers and administrators. The certification provides a way for individuals to demonstrate their knowledge and skills with Kubernetes, and can be a valuable asset for career advancement.
* Ecosystem: Kubernetes has a large and growing ecosystem of tools and services that extend and enhance the platform. This includes things like monitoring and logging tools, continuous integration and deployment (CI/CD) platforms, and container registry services.
* Cloud providers: All major cloud providers, including AWS, Azure, and Google Cloud Platform, offer Kubernetes as a managed service. This can make it easier to deploy and manage Kubernetes clusters in the cloud, but it also means that organizations need to carefully consider vendor lock-in and compatibility when choosing a cloud provider.
* Complexity: Kubernetes can be a complex platform to set up and manage, particularly for smaller organizations or teams. While Kubernetes provides a flexible and scalable platform for container orchestration, it can require a significant investment of time and resources to learn and implement effectively.
* **Project Comparison:**

There are several other container orchestration platforms available in the market that can be compared to Kubernetes. Here are some of the notable ones:

* Docker Swarm: Docker Swarm is a container orchestration platform that is built into Docker. It provides similar functionality to Kubernetes, but with a simpler and more streamlined architecture
* Amazon ECS: Amazon ECS (Elastic Container Service) is a managed container orchestration platform provided by AWS. ECS is a simpler and more streamlined platform than Kubernetes, but it may not be as flexible or customizable as

Kubernetes.

* Apache Mesos: Apache Mesos is a distributed systems kernel that can be used for container orchestration, among other things. Mesos provides a highly scalable and fault-tolerant platform for running containerized workloadsbernetes.
* OpenShift: OpenShift is a container application platform that is built on top of Kubernetes. OpenShift provides additional features and capabilities on top of Kubernetes, such as integrated CI/CD pipelines and developer tools

* **Additional info :**

1. Community: Kubernetes has a large and active community of developers and contributors from around the world. The project is hosted by the Cloud Native Computing Foundation (CNCF), a non-profit organization that also hosts other cloud-native projects like Prometheus and Envoy.
2. Certification: The CNCF offers a Kubernetes certification program for developers and administrators. The certification provides a way for individuals to demonstrate their knowledge and skills with Kubernetes, and can be a valuable asset for career advancement.
3. Ecosystem: Kubernetes has a large and growing ecosystem of tools and services that extend and enhance the platform. This includes things like monitoring and logging tools, continuous integration and deployment (CI/CD) platforms, and container registry services.
4. Cloud providers: All major cloud providers, including AWS, Azure, and Google Cloud Platform, offer Kubernetes as a managed service. This can make it easier to deploy and manage Kubernetes clusters in the cloud, but it also means that organizations need to carefully consider vendor lock-in and compatibility when choosing a cloud provider.
5. Complexity: Kubernetes can be a complex platform to set up and manage, particularly for smaller organizations or teams. While Kubernetes provides a flexible and scalable platform for container orchestration, it can require a significant investment of time and resources to learn and implement effectively.

* **Reference :**
* <https://kubernetes.io/> :official site of kubernetes project.
* <https://github.com/kubernetes/dashboard> :

community in github.