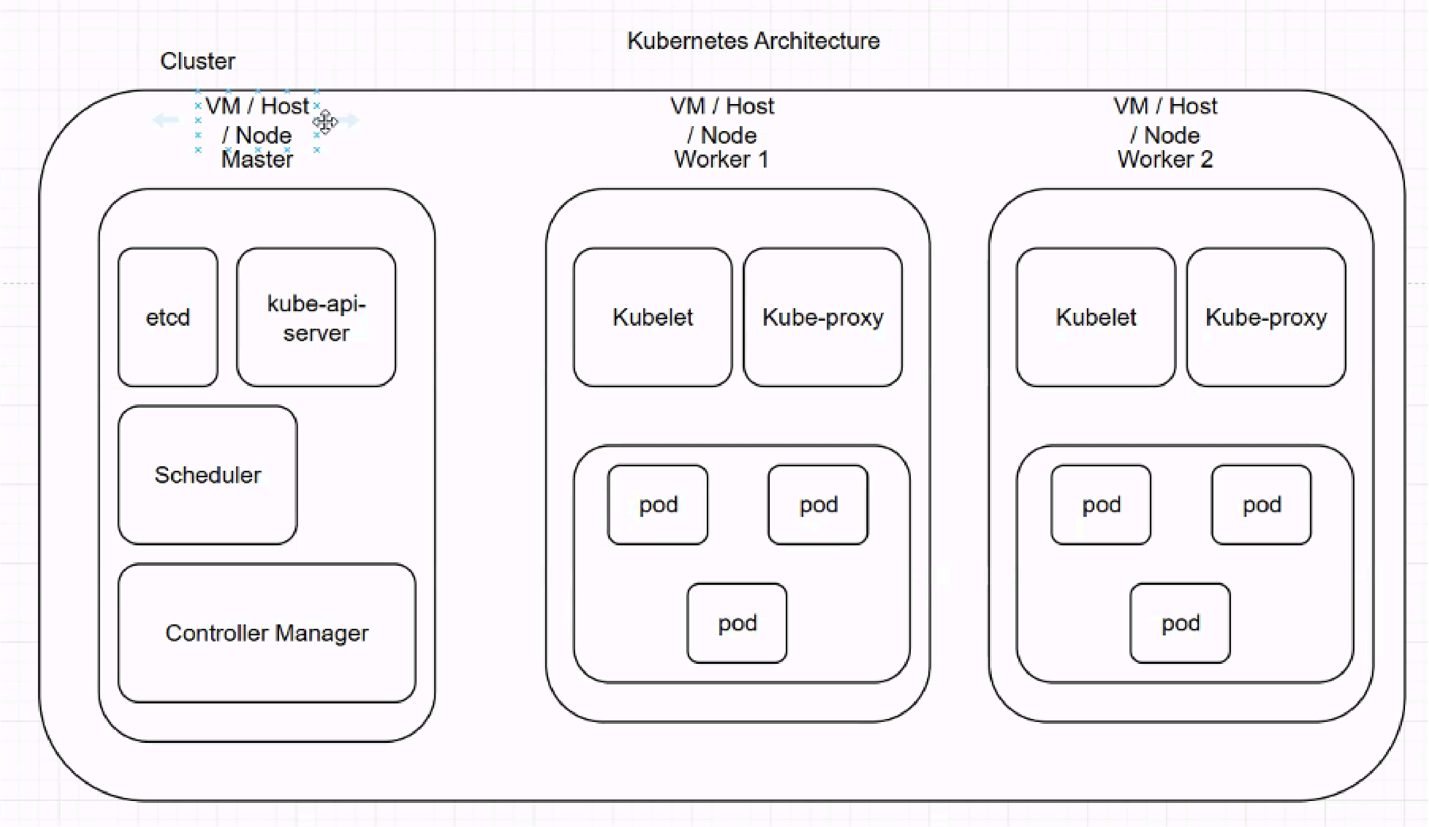
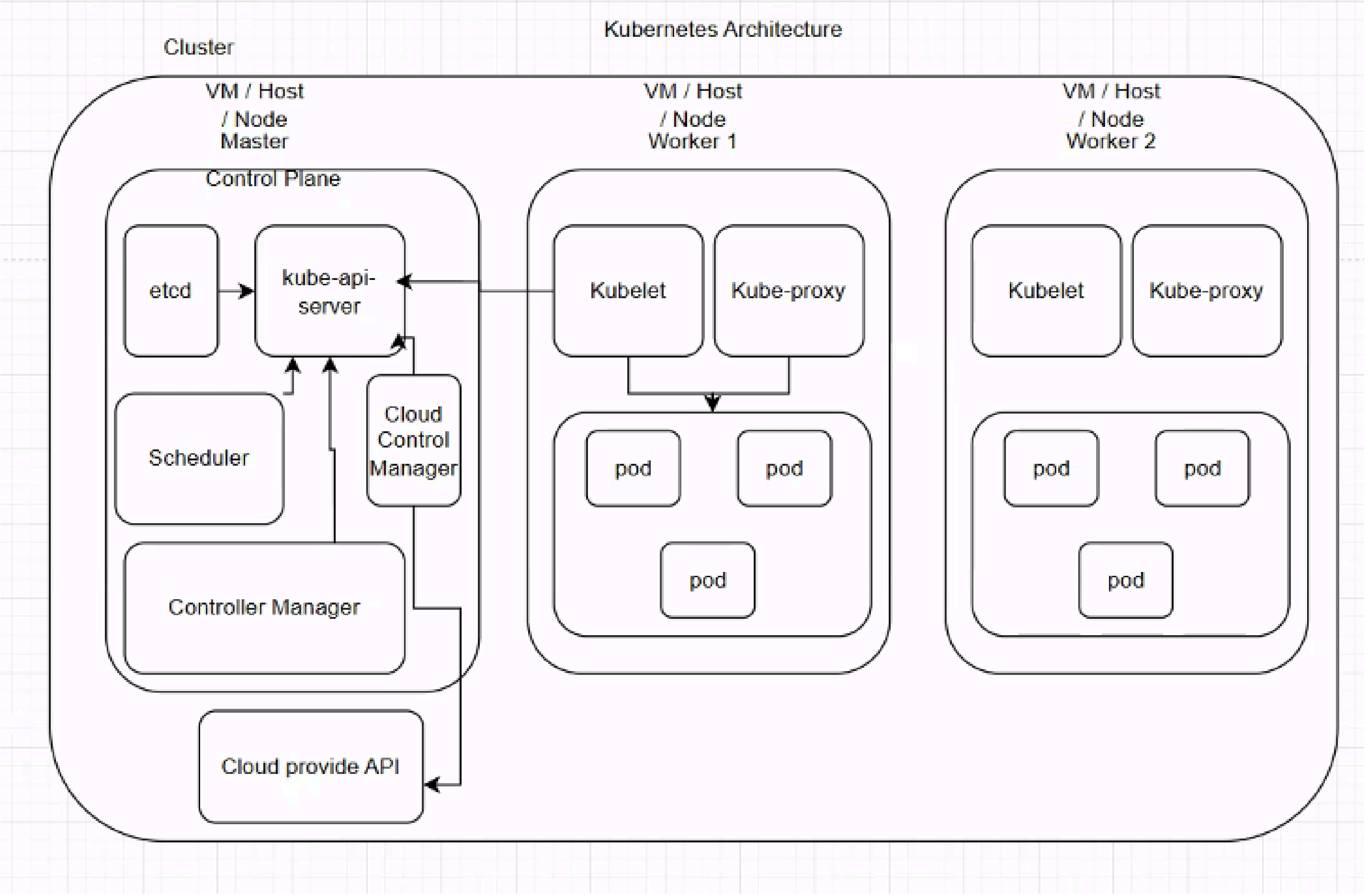
Kubernetes Architecture:-



Kubernetes Architecture with Flow:-



Kubernetes Core Concepts :-

Node- A node is a physical or virtual machine that runs containers. It is a worker machine in the Kubernetes cluster. Nodes are responsible for executing the containers and providing the runtime environment.

Pods- A pod is the smallest deployable unit in Kubernetes. It represents a single instance of a running process in the cluster. Pods can host one or more containers that share the same network namespace and storage volumes.

ReplicaSet- A ReplicaSet is responsible for maintaining a specified number of replica pods. It ensures that the desired number of identical pod replicas are running at all times, helping with availability and scaling.

Deployment- A Deployment is a higher-level abstraction that manages ReplicaSets. It allows you to declaratively define the desired state of your application and handles updates and rollbacks of your application's pods.

Some more key concepts in Kubernetes are-

Service- A Service is an abstraction that defines a logical set of pods and a policy to access them. It provides load balancing and a stable network IP and DNS name for accessing the pods, even as they come and go.

Namespace- A Namespace is a virtual cluster within a Kubernetes cluster. It provides a way to create isolated scopes for resources, helping to manage different environments, projects, or teams within the same cluster.

Ingress- An Ingress is an API object that manages external access to services within a cluster. It provides features like routing based on hostnames, SSL termination, and more.

ConfigMap and Secret- ConfigMap store configuration data as key-value pairs, while Secrets store sensitive information like passwords or API keys. Both allow you to separate configuration from code.

Deployments :-

What are Deployments?

> A higher-level abstraction that manages Pods and ReplicaSets, providing declarative updates to applications.

> Purpose: to manage the deployment and scaling of a set of Pods and ensure a specified number of Pods are running at all times.

Key Points-

Declarative Updates: Allows you to describe an application's life cycle, including updates and rollbacks.

Rolling Updates and Rollbacks: Ensures updates happen progressively to avoid downtime.

Rolling Updates and Rollbacks: Easily scale the number of Pods replicas.

Key Features of Kubernetes :-

1. Automated rollouts and rollbacks.

2. Services discovery and load balancing.

3. Storage Orchestration.

4. Self-healing.

5. Secret and Configuration management.

Why Use Kubernetes ?

1. Scalability:- Easily scale applications up and down.

2. Portability:- Run applications consistently across environments.

3. Efficiency:- Optimize resources utilization.

4. Resilience:- Ensure applications are highly available.

We'll furthur discuss over each core concepts of Kubernetes in details.