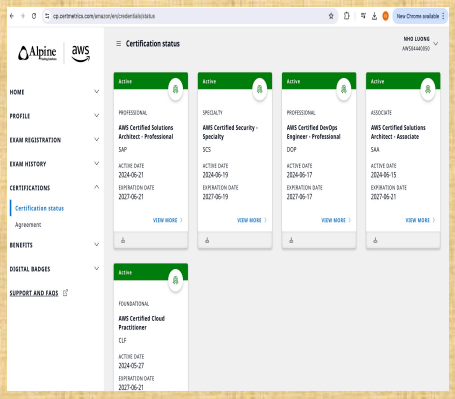
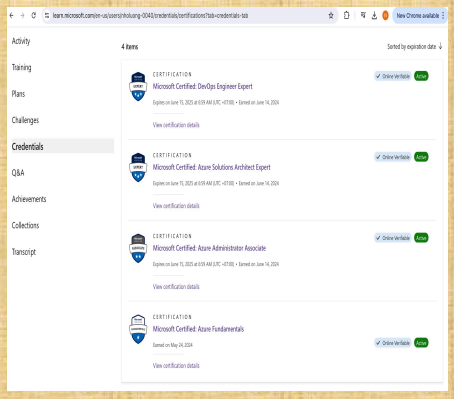
Kubernetes Ochestration

Author: Nho Luong

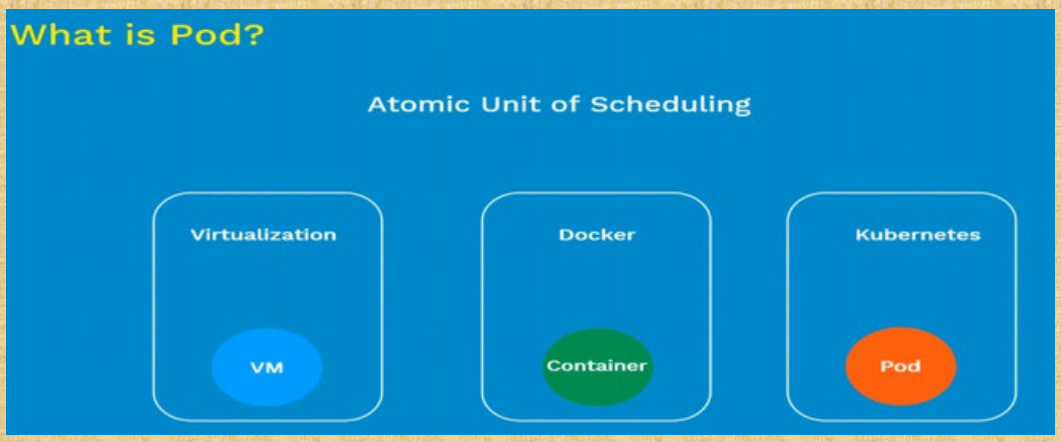
Skill: DevOps Engineer Lead



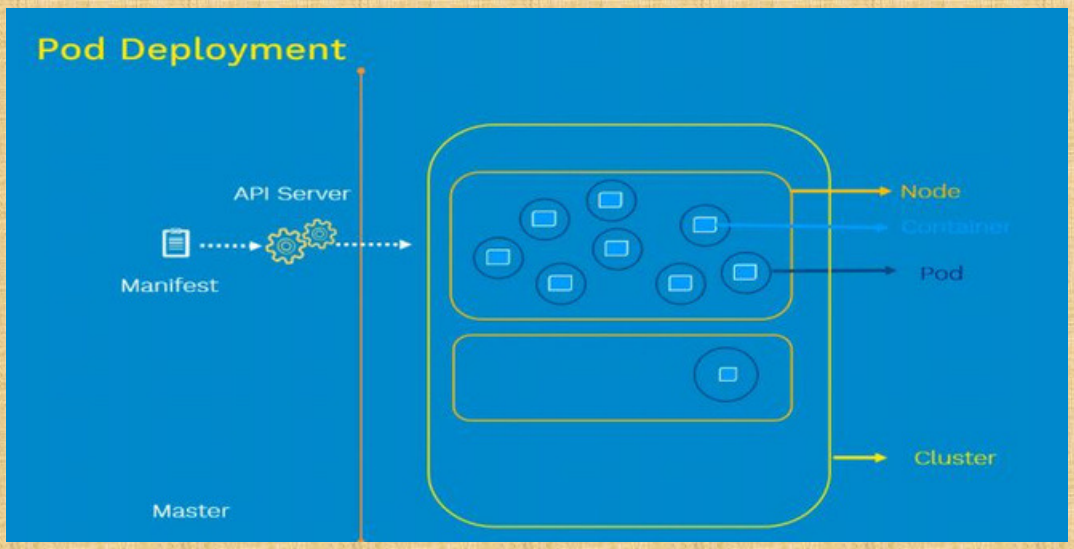


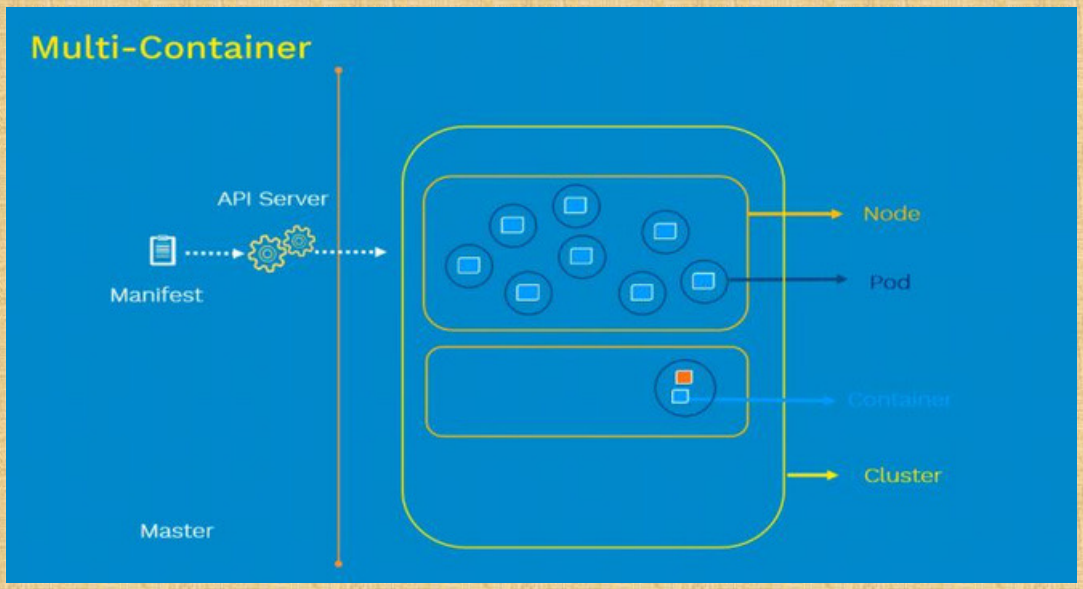
Kubernetes

Pods:-It is the Atomic unit of Scheduling.



AKubernetes podis a group of containers that are deployed together on the same host. If you frequently deploy single containers, you can generally replace the word "pod" with "container" and accurately understand the concept





Author: Nho Luong
Skill: DevOps Engineer Lead

OVERVIEW

What is Kubernetes?

COE

Manages containerized apps @ large scale

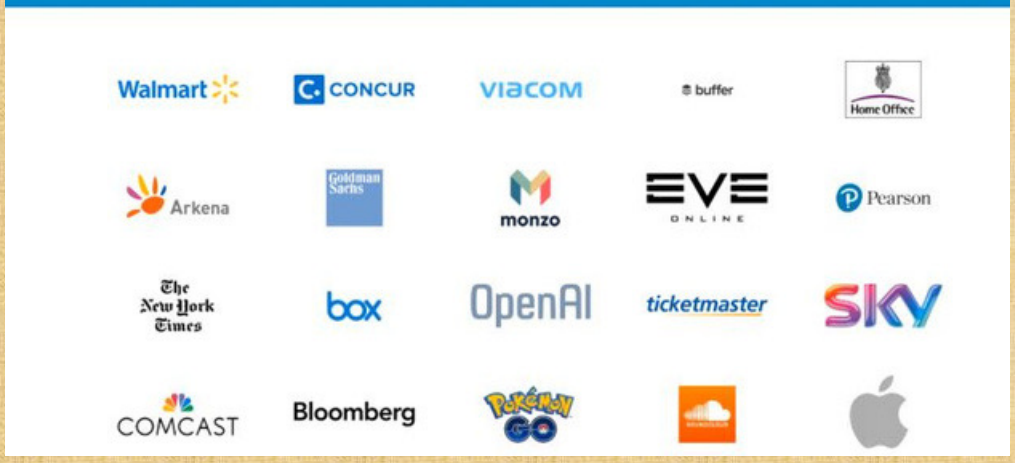
cluster management | scheduling | service discovery monitoring | secrets management & more

Google → CNCF

History @ Google Inc.

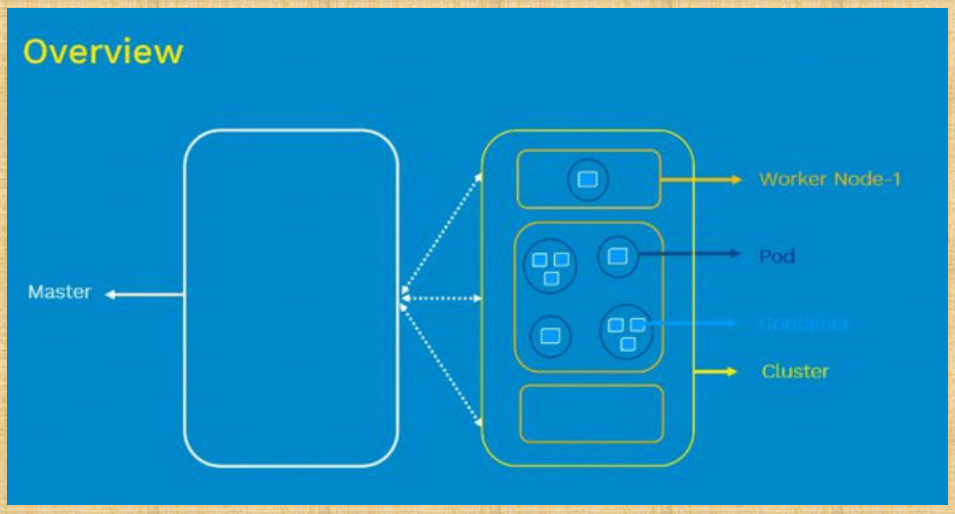
- More than Decade
- Runs billions of containers every week
- Gmail, YouTube, Search ... runs on containers
- Borg – Proprietary container manager of Google
- Container Orchestration Engine on Google Cloud

Who is using Kubernetes?

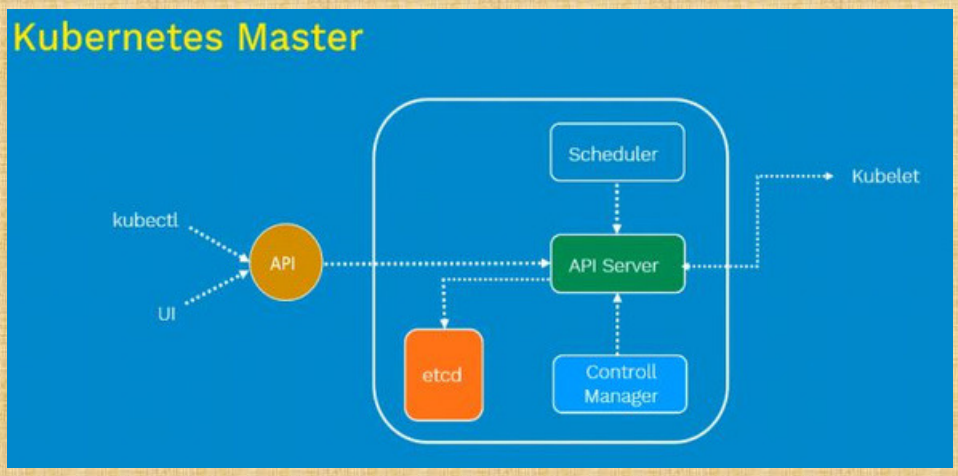


Author: Nho Luong
Skill: DevOps Engineer Lead

Architecture



It is consisted of at least one Master node It can have up to 5K worker nodes Pod is consisted of one or more container Container is where actual application is running and inside the container



Master Components

To manage all the worker node

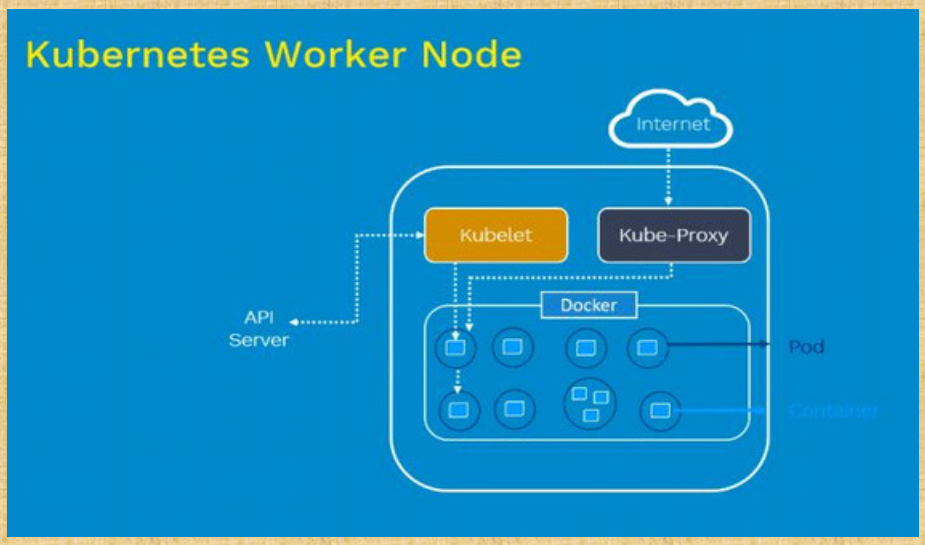
API Server The Kubernetes API server validates and configures data for the api objects which include pods, services, replicationcontrollers, and others.
The API Server services REST operations and provides the frontend to the cluster's shared state through which all other components interact.

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.

Control Manager The Kubernetes controller manager is a daemon that embeds the core control loops shipped with Kubernetes. In applications of robotics and automation, a control loop is a non-terminating loop that regulates the state of the system. In Kubernetes, a controller is a control loop that watches the shared state of the cluster through the apiserver and makes changes attempting to move the current state towards the desired state. Examples of controllers that ship with Kubernetes today are the replication controller, endpoints controller, namespace controller, and service accounts controller.

etcd etcd is a consistent and highly-available key value store used as Kubernetes' backing store for all cluster data.

Author: Nho Luong
Skill: DevOps Engineer Lead



Worker Node Components
Kubelet

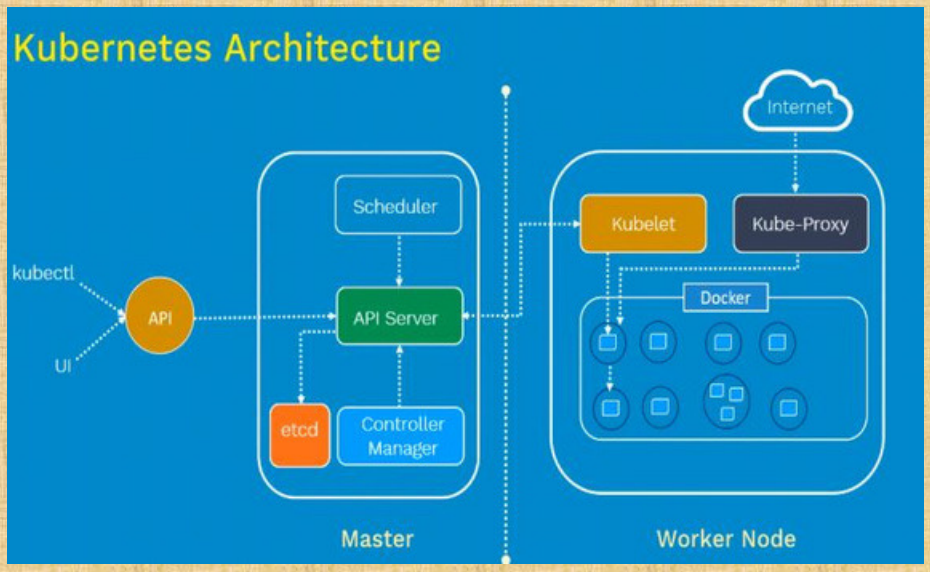
It takes a set of PodSpecs that are provided through various mechanisms (primarily through the apiserver) and ensures that the containers described in those PodSpecs are running and healthy.

The**kubelet**doesn't manage containers which were not created by Kubernetes.

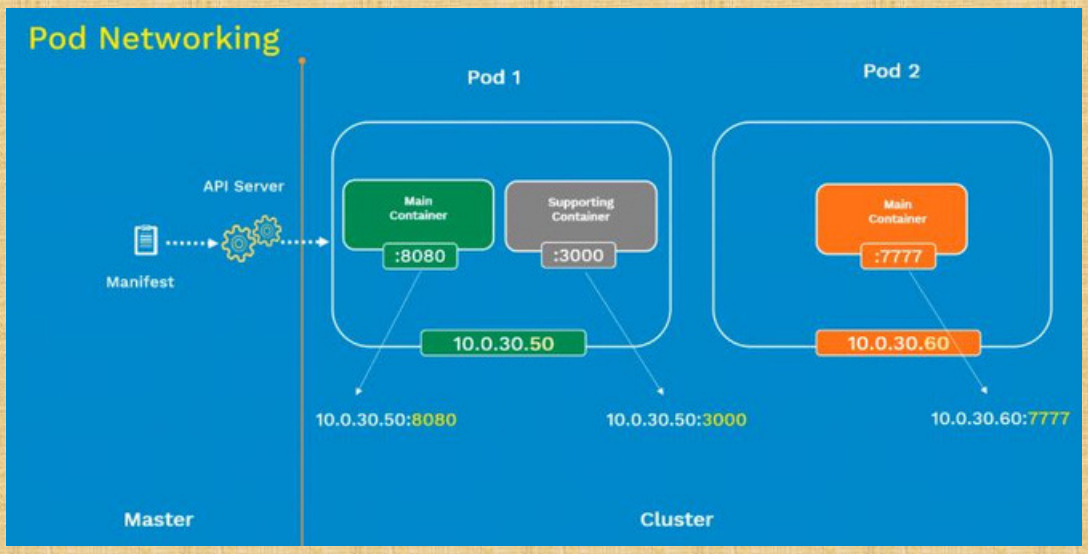
The Kubernetes network proxyruns on each node.

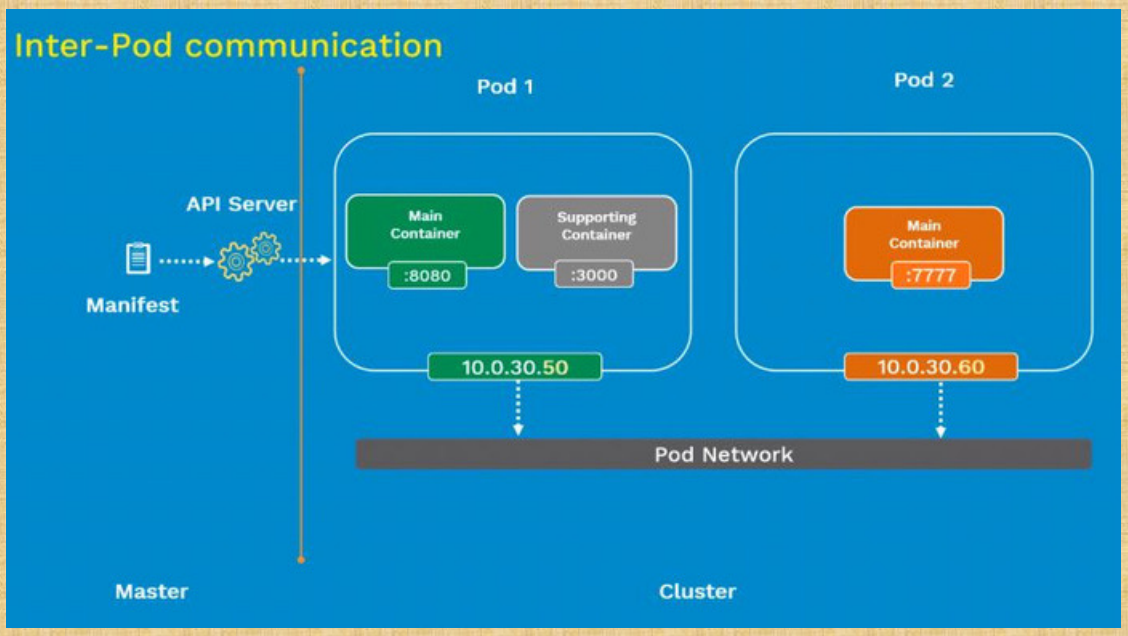
This reflects services as defined in the Kubernetes API on each node and can do simple TCP, UDP, and SCTP stream

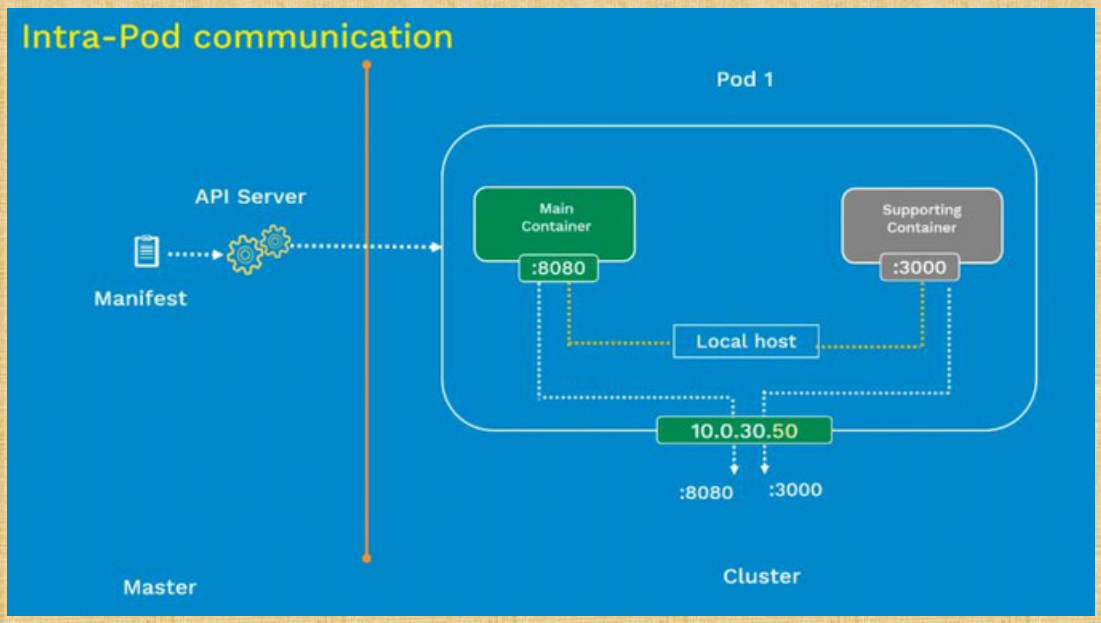
forwarding or round robin TCP, UDP, and SCTP forwarding across a set of backends



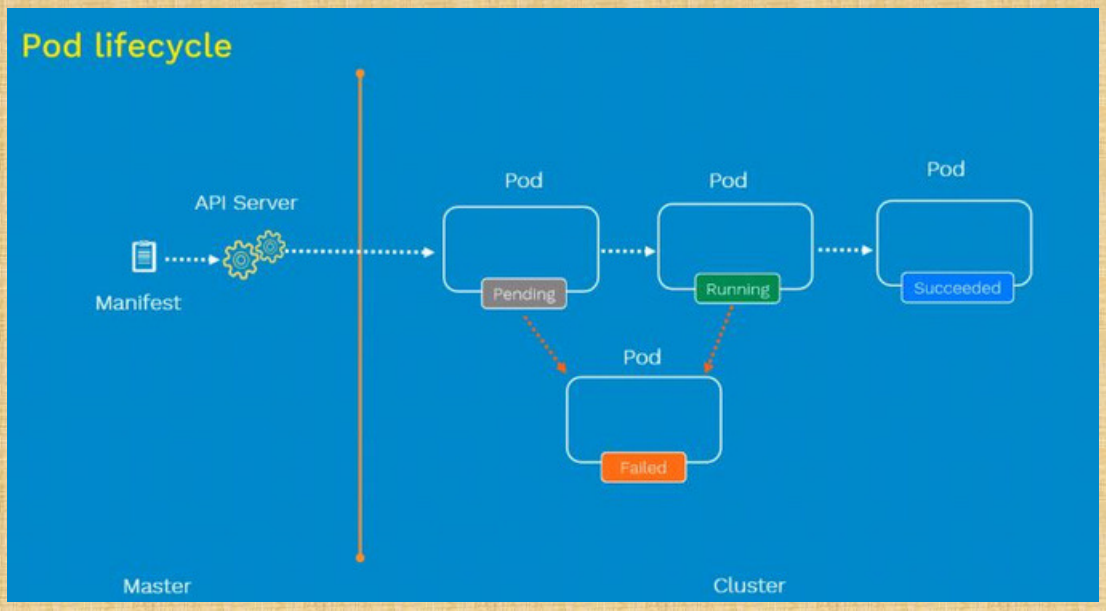
Author: Nho Luong
Skill: DevOps Engineer Lead







Author: Nho Luong
Skill: DevOps Engineer Lead



Pod - Config

```
# nginx-pod.yaml
apiVersion: v1
kind: Pod
metadata:
  name: nginx-pod
  labels:
    app: nginx
    tier: dev
spec:
  containers:
  - name: nginx-container
    image: nginx
```

Kind	apiVersion
Pod	v1
ReplicationController	V1
Service	v1
ReplicaSet	apps/v1
Deployment	apps/v1
DaemonSet	apps/v1
Job	batch/v1

DEMO:-Pod Creation

ConfigMaps

- Decouples configuration from pods and components
- Stores configuration data as Key-value pairs
  - Configuration files
  - Command line arguments
  - Environment variables
- Similar to Secrets but don't contain sensitive information

Author: Nho Luong
Skill: DevOps Engineer Lead

DEMO-Config Maps

Secrets

Kubernetes object to handle small amount of sensitive data

Overview

- Small amount of sensitive data
  - Passwords, Tokens, or Keys
- Reduces risk of exposing sensitive data
- Created outside of Pods
- Stored inside ETCD database on Kubernetes Master
- Not more than 1MB
- Used in two ways- Volumes or Env variables
- Sent only to the target nodes

Using Kubectl: Syntax

```
kubectl create secret [TYPE] [NAME] [DATA]
```

generic
- File
- Directory
- Literal Value

docker-registry

tls

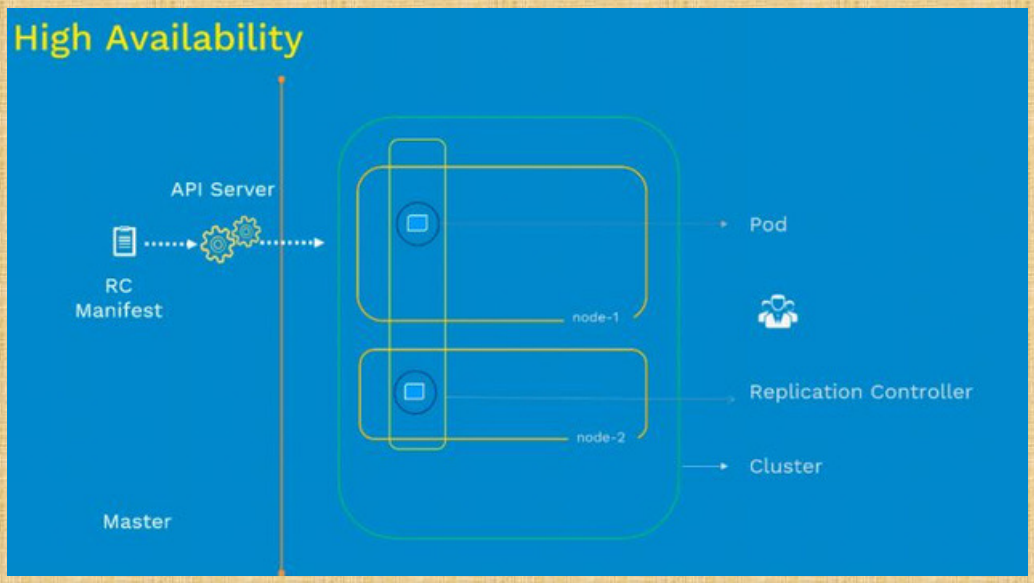
- Path to dir/file: --from-file
- Key-Value pair : --from-literal

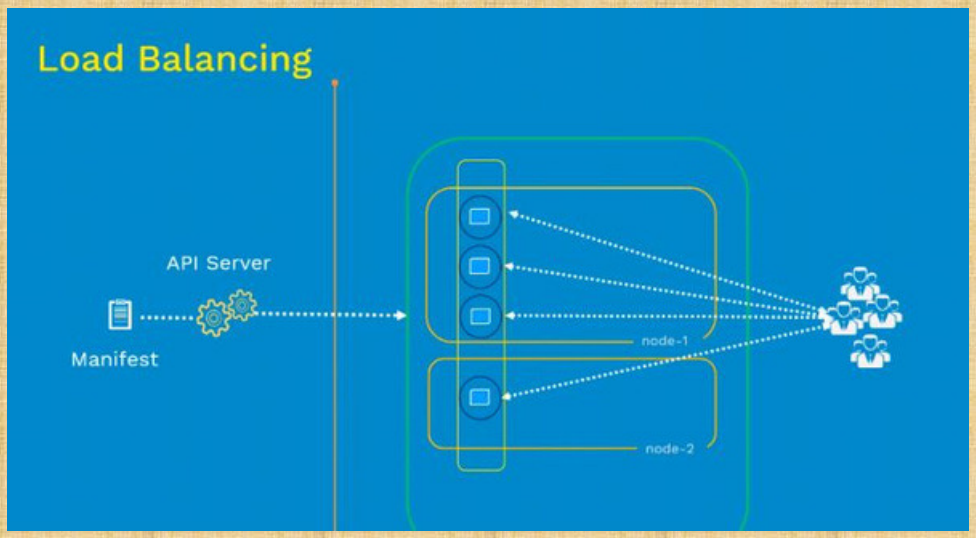
Secert Demo

Replication Controller

- Ensures that a specified number of pods are running at any time
  a. If there are excess Pods, they get killed and vice versa
  b. New Pods are launched when they get fail, get deleted or terminated
- Replication Controllers and Pods are associated with " labels "
- Creating a "rc" with count of 1 ensure that a pod is always available

Author: Nho Luong
Skill: DevOps Engineer Lead





Replication Controller Demo

ReplicaSet

- Ensures that a specified number of pods are running at any time
  - a. If there are excess Pods, they get killed and vice versa
  - b. New Pods are launched when they get fail, get deleted or terminated
- ReplicaSet and Pods are associated with " labels "

Author: Nho Luong
Skill: DevOps Engineer Lead

ReplicaSet vs. Replication Controller

ReplicaSet is Next-generation Replication Controller

ReplicaSet → Set-based Selectors

Replication Controller → Equality-based Selectors

Labels & Selectors

Pods

Labels

```
#Pod-Spec
apiVersion: v1
kind: pod
metadata:
  name: nginx-pod
  labels:
    app: guestbook
    tier: frontend
    env: dev
spec:
  replicas: 5
  . . . . .
```

Controllers & Services

Selectors

Equality-based

Operators:

`=` `==` `!=`

Examples:

```
environment = production
tier != frontend
```

Command line

```
$ kubectl get pods -l environment=production
```

In manifest:

```
...
selector:
  environment: production
  tier: frontend
...
```

Set-based

Operators:

in notin exists

Examples:

```
environment in (production, qa)
tier notin (frontend, backend)
```

Command line

```
$ kubectl get pods -l 'environment in (production)
```

In manifest:

```
...
selector:
  matchExpressions:
    - {key: environment, operator: In, values: [prod, qa]}
    - {key: tier, operator: NotIn, values: [frontend, backend]}
...
```

```
...
selector:
    app: nginx
    tier: frontend
...
```

=

```
...
selector:
  matchLabels:
    app: nginx
    tier: frontend
...
```

Supports on Older Resources such as:

- ReplicationControllers,
- Services

Supports on newer resources such as:

- ReplicaSets
- Deployments
- Jobs
- DaemonSet

Author: Nho Luong
Skill: DevOps Engineer Lead

Replica Set Demo

Imagine, you are upgrading application from v1 to v2

Upgrade with zero downtime?

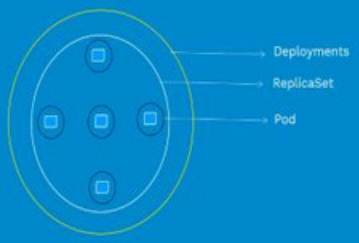
Upgrade sequentially, one after the other?

Pause and resume upgrade process?

Rollback upgrade to previous stable release

Deployments

Updates & Rollbacks



Features

- Multiple Replicas
- Upgrade
- Rollback
- Scale up or down
- Pause and Resume

Deployment Types

- Recreate
- RollingUpdate (Ramped or Incremental)
- Canary
- Blue / Green

Author: Nho Luong
Skill: DevOps Engineer Lead

Deployment Demo

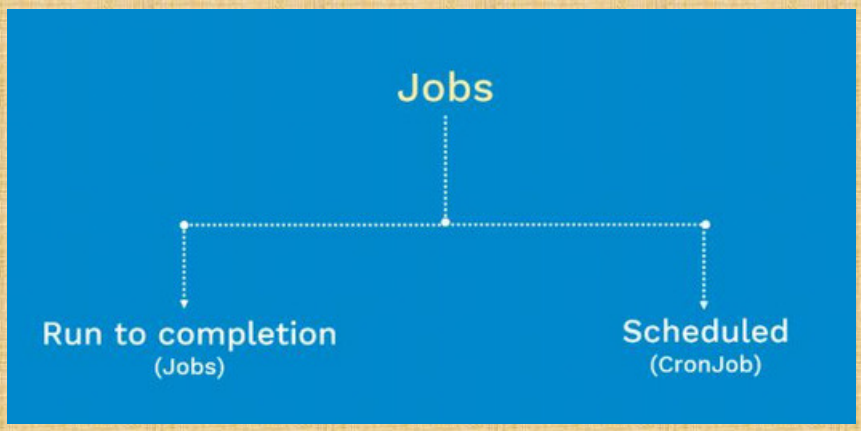
DaemonSet - Overview

- A DaemonSet ensures that all (or some) Nodes run a copy of a Pod.
- As nodes are added to the cluster, Pods are added
- As nodes are removed from the cluster, those Pods are garbage collected
- Deleting a DaemonSet will clean up the Pods it created

Use Cases:

- Node monitoring daemons: Ex: **collectd**
- Log collection daemons: Ex: **fluentd**
- Storage daemons: Ex: **ceph**

DaemonSet Demo



Run to completion

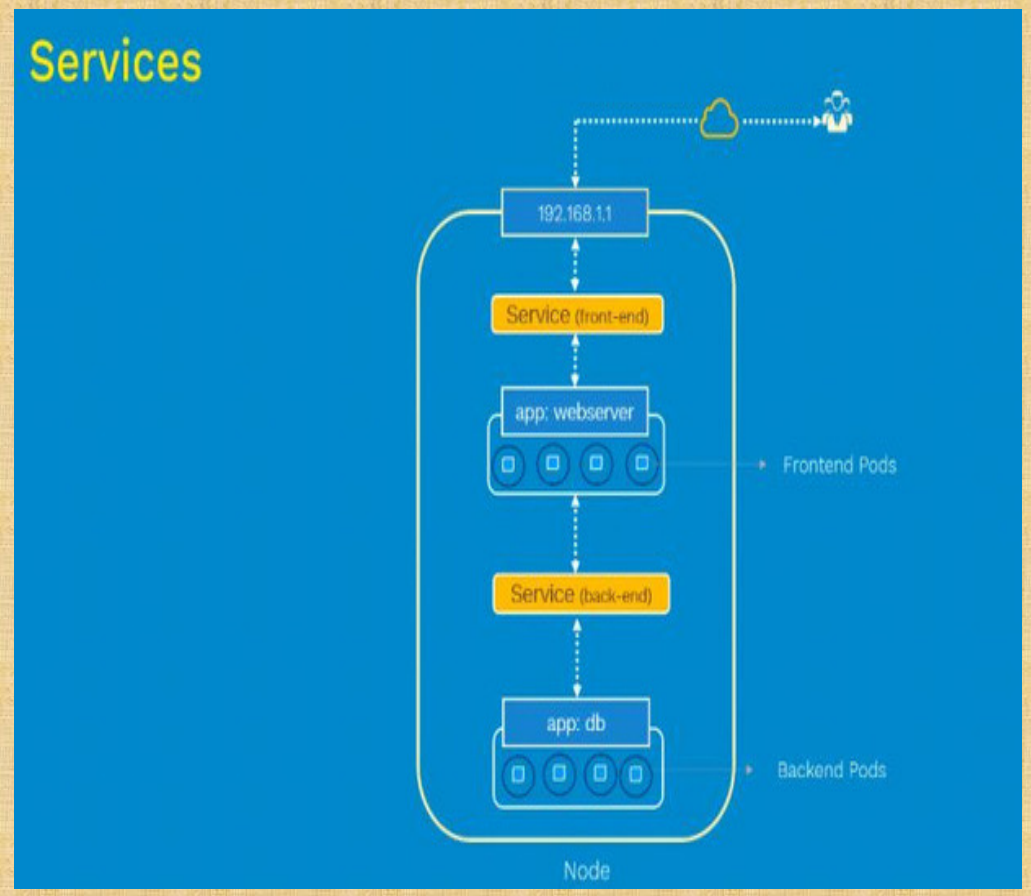
- Each job creates one or more Pods
- Ensures they are successfully terminated
- Job controller restarts or rescheduled if a pod or node fails during execution
- Can run multiple pods in parallel
- Can scale up using kubectl scale command

Use Cases:

- One time initialization of resources such as Databases
- Multiple workers to process messages in queue

Author: Nho Luong
Skill: DevOps Engineer Lead

JOB Demo



Author: Nho Luong
Skill: DevOps Engineer Lead

Thank You

Author: Nho Luong
Skill: DevOps Engineer Lead