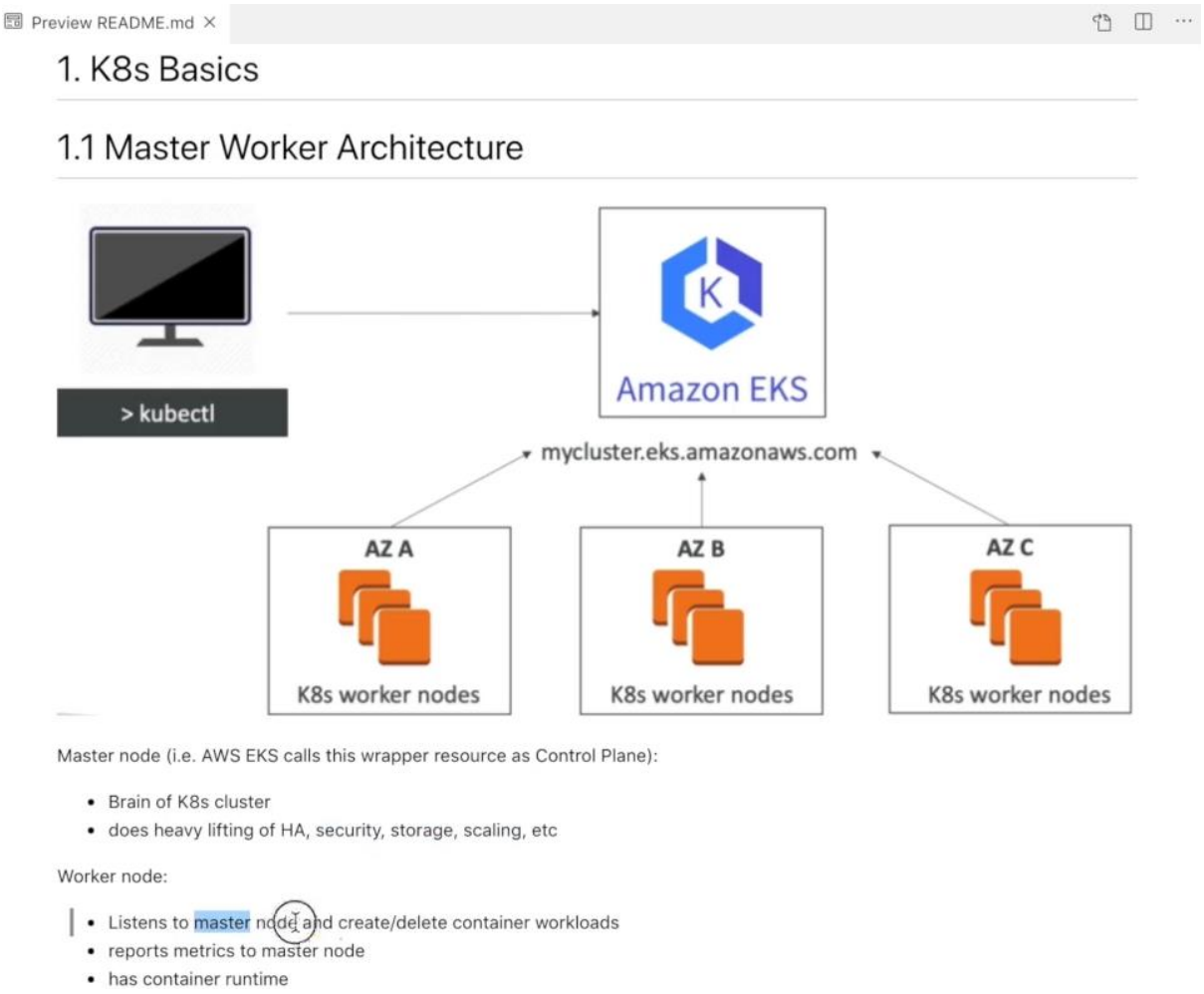
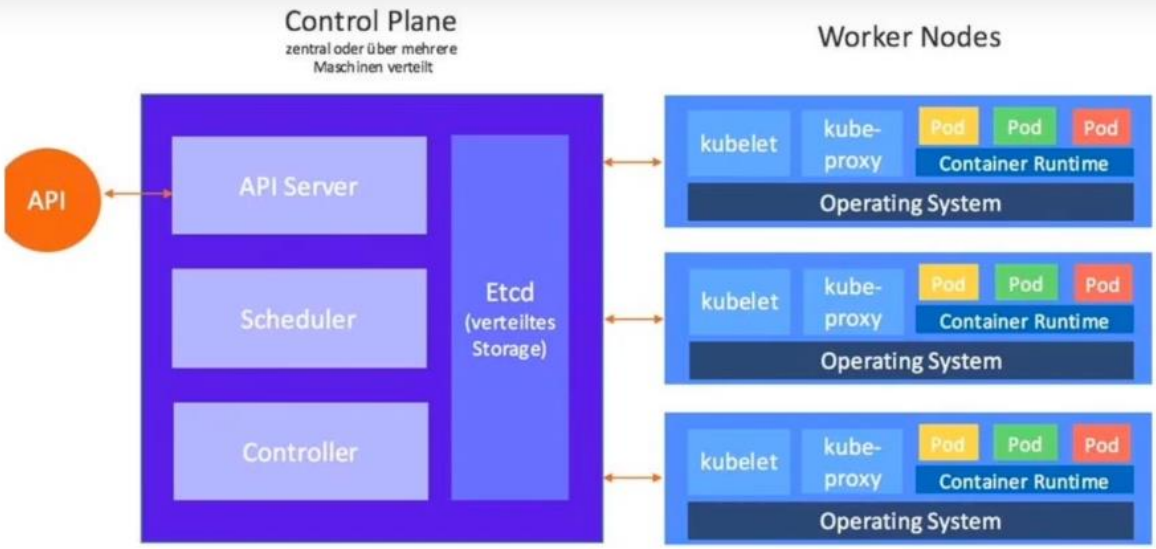
AWS EKS Best Practices:

EKS is one the services that AWS offers that manages the control plane in K8S and configuring the worker nodes can be self-managed, aws-managed and serverless. We need to follow few best practices while creating the EKS cluster for Enterprise level for security and other reasons as listed below.



1.2 Master Node (Control Plane)



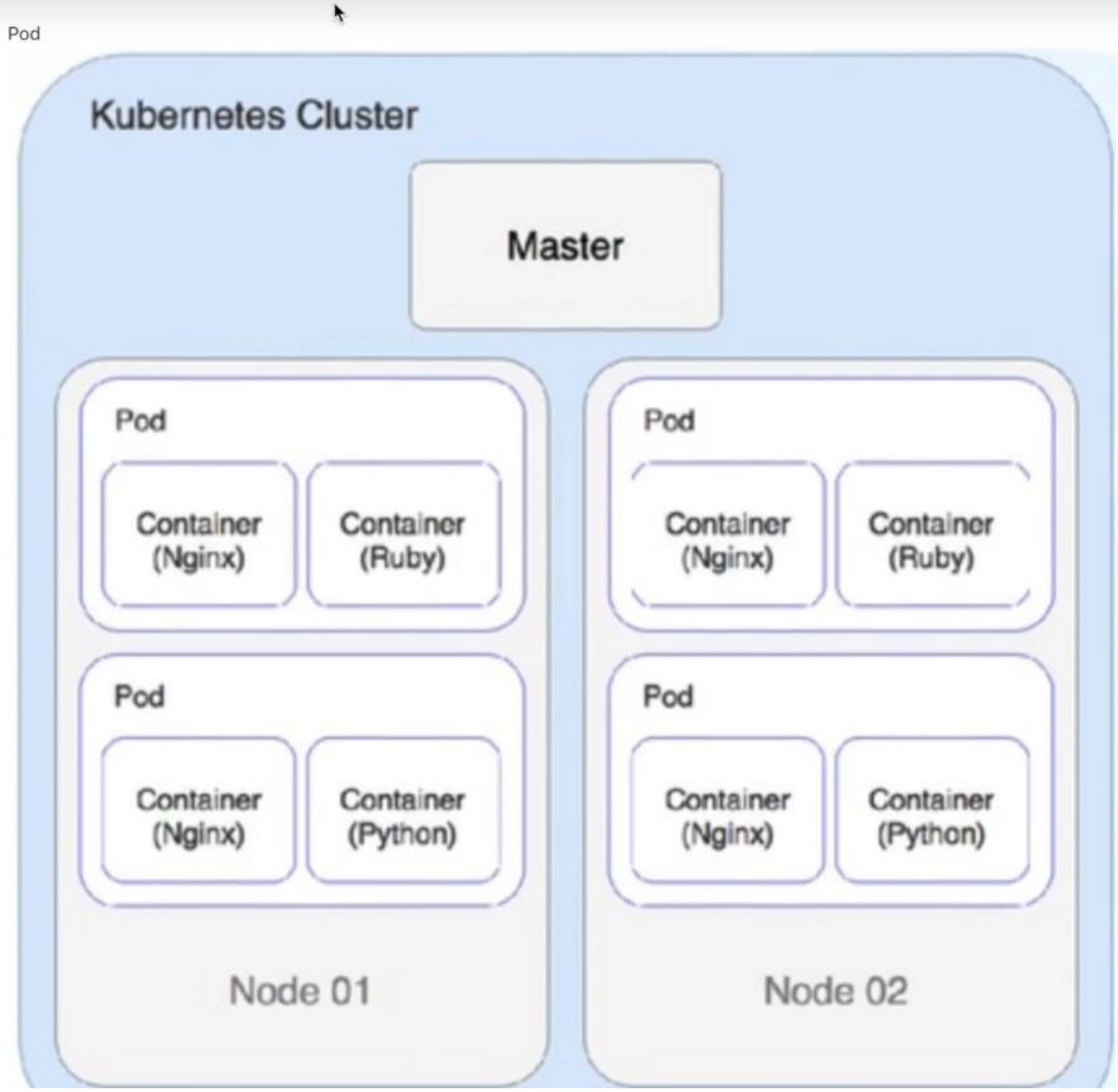
- API server: interacts with kubectl CLI
- Etcd: key-value store, implements locks
- Controller: health check, makes sure pods are running
- Scheduler: creates new pods and assign them to nodes

1.3 Worker Nodes (Data Plane)

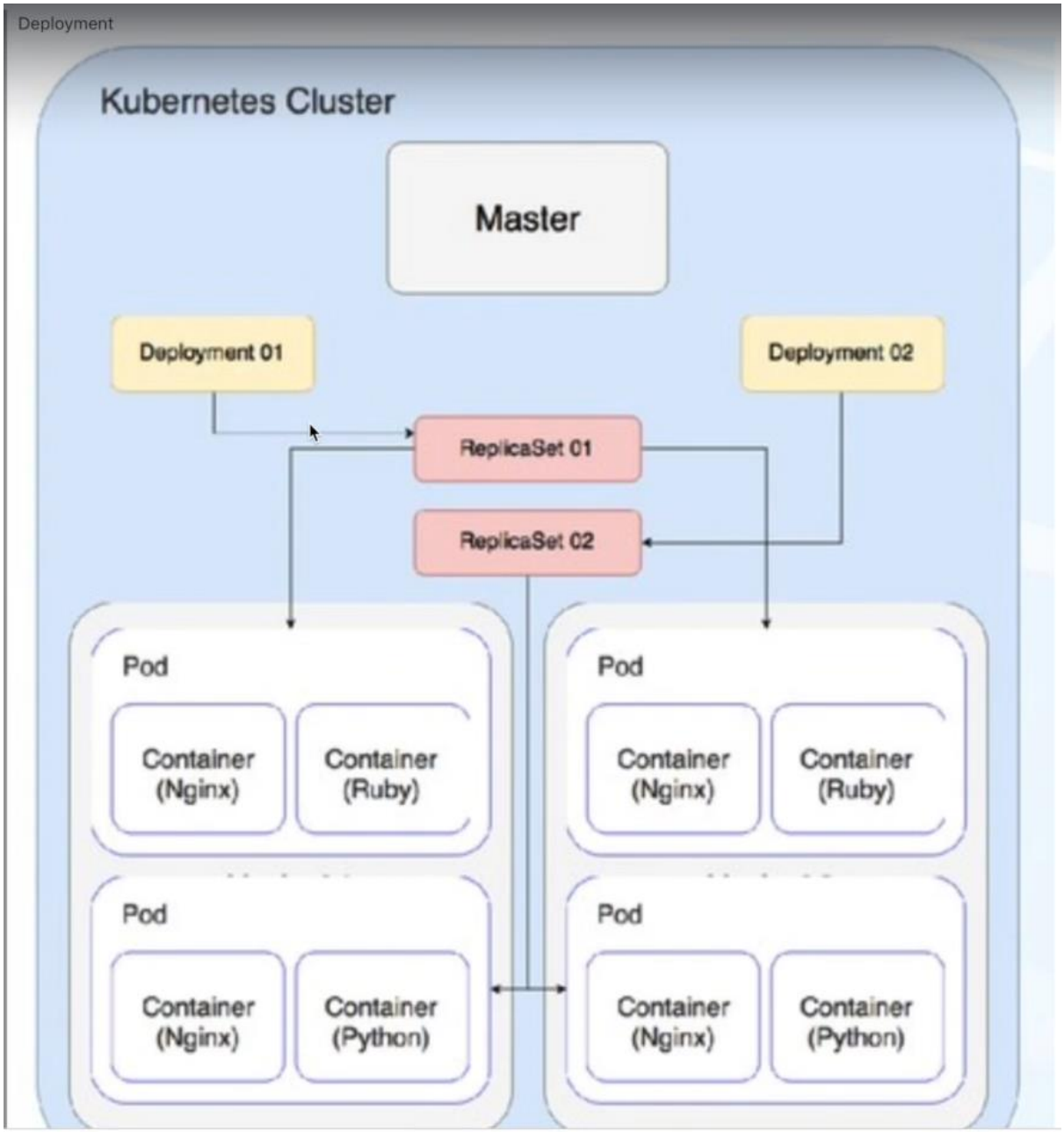
- Kubelet: agent running on cluster nodes
- Container runtime: such as Docker runtime
- Kubectl: CLI to manage/deploy apps on cluster

PODS:

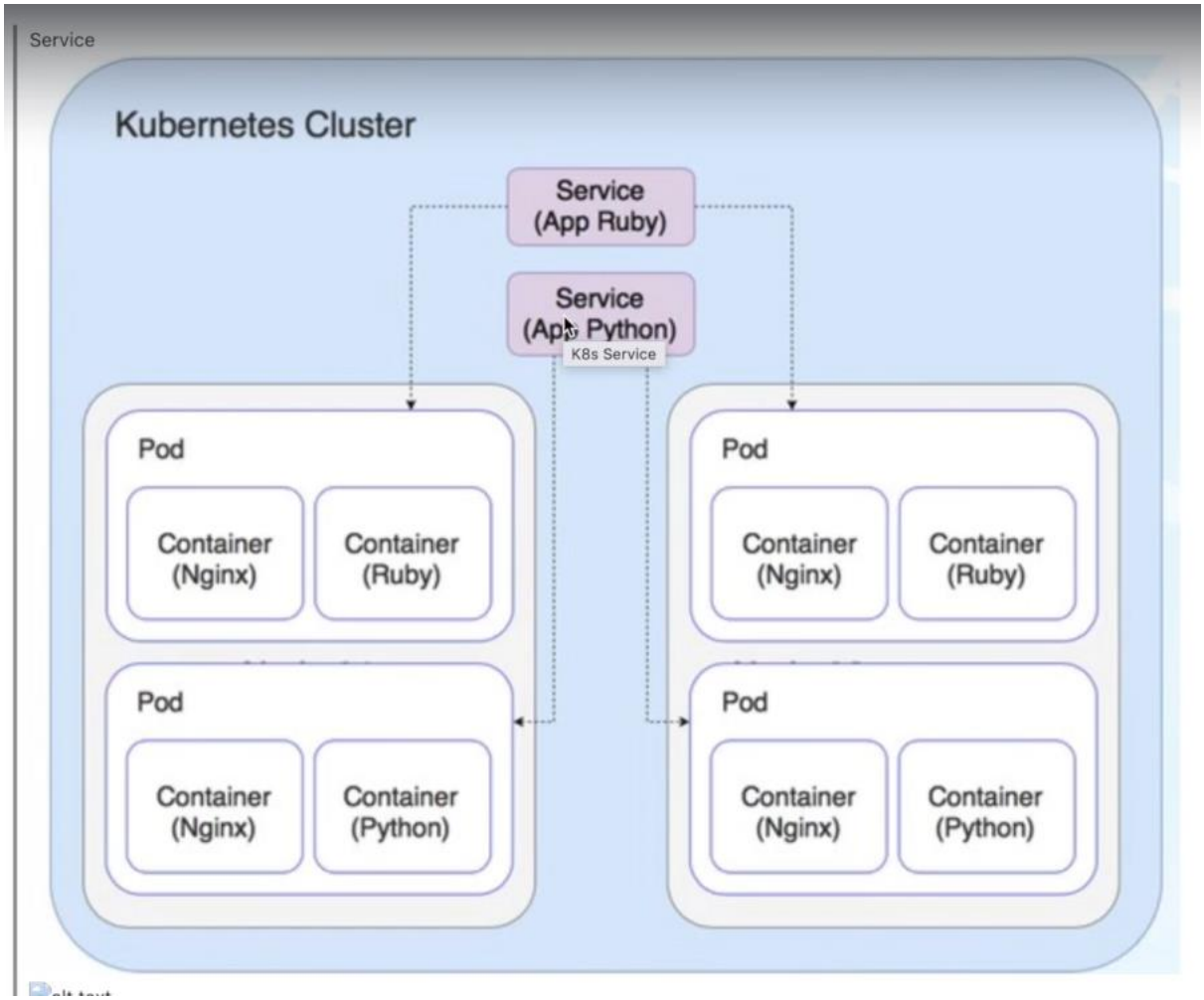
1.4 K8s Objects - pod, deployment, service, configmap, serviceaccount, ingress, etc



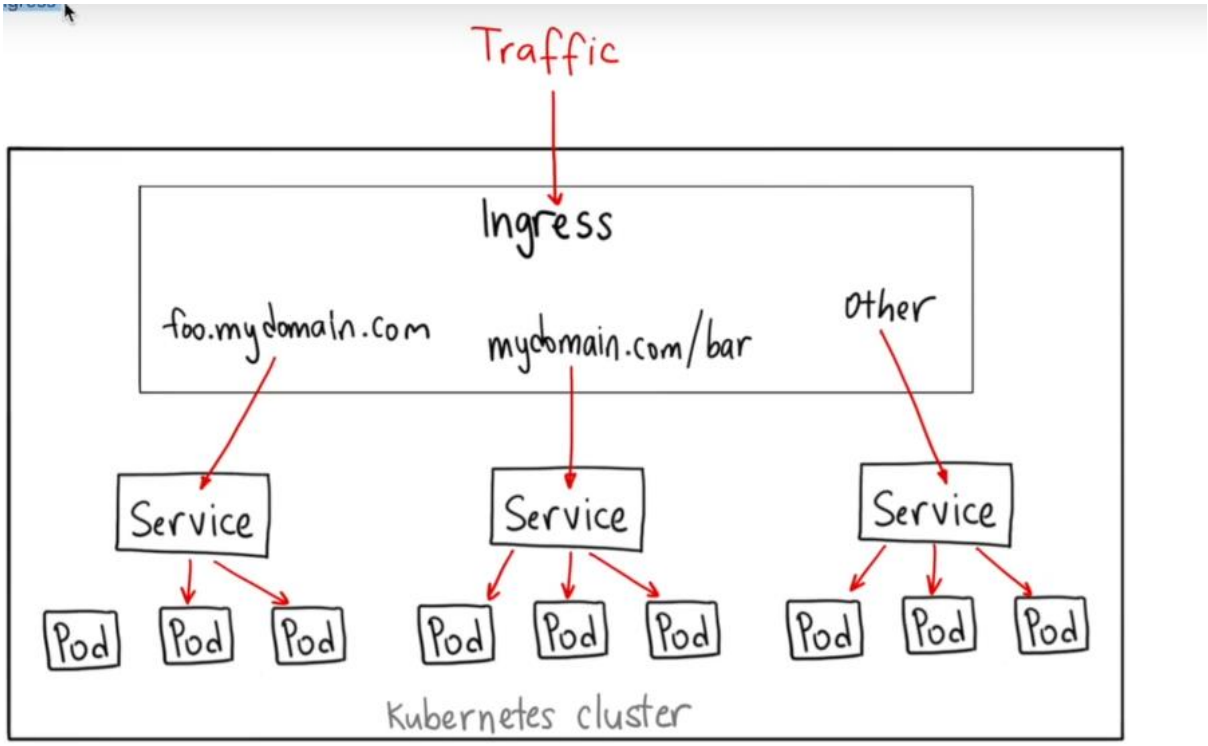
Deployments:


Deployment
Kubernetes Cluster
Master
Deployment 01
Deployment 02
ReplicaSet 01
ReplicaSet 02
Pod
Container (Nginx)
Container (Ruby)
Pod
Container (Nginx)
Container (Ruby)
Pod
Container (Nginx)
Container (Python)
Pod
Container (Nginx)
Container (Python)

Service:

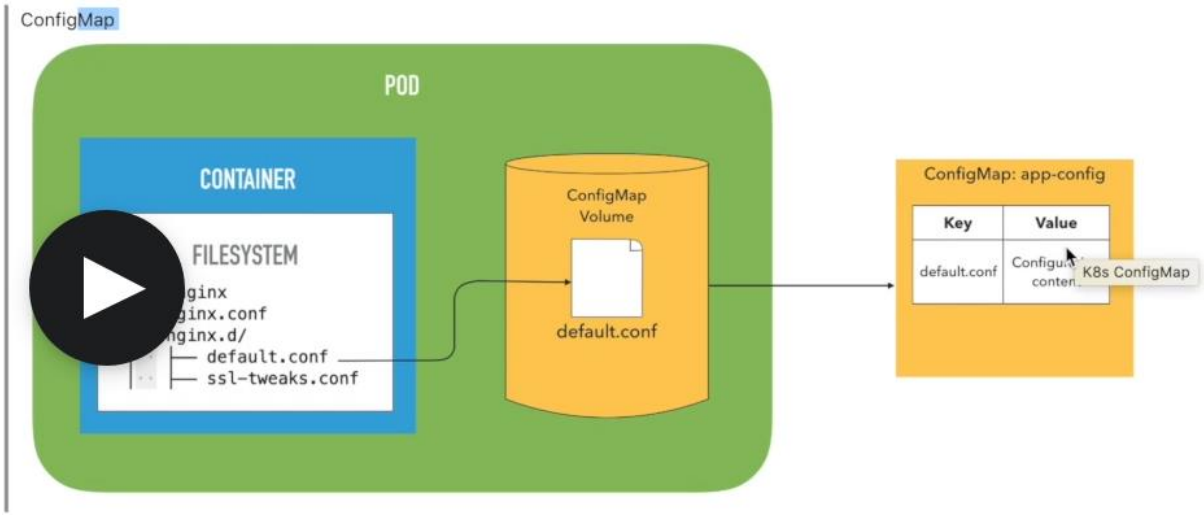

Service
Kubernetes Cluster
Service
(App Ruby)
Service
(App Python)
K8s Service
Pod
Container
(Nginx)
Container
(Ruby)
Pod
Container
(Nginx)
Container
(Python)
Pod
Container
(Nginx)
Container
(Ruby)
Pod
Container
(Nginx)
Container
(Python)

Ingress:



ConfigMap:

It can be as service or volume mounted on to default.conf to know more check CKA_Crash_course.ipynb.



In this course you see the Best practices for PROD EKS, However the infrastructure should be spun using IaC (Terraform). Refer the other course stuff for that. Here the info is related to only the EKS.