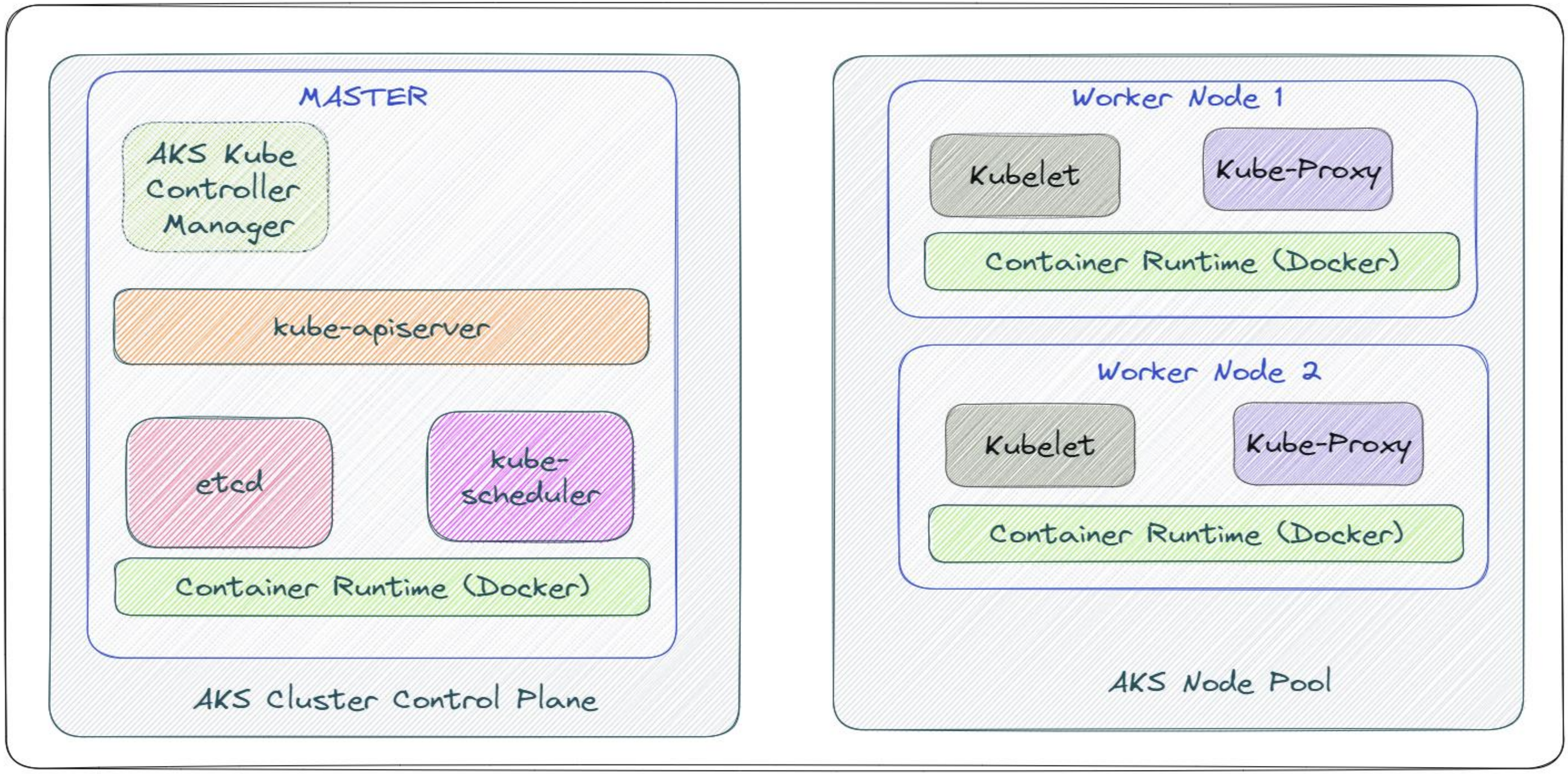
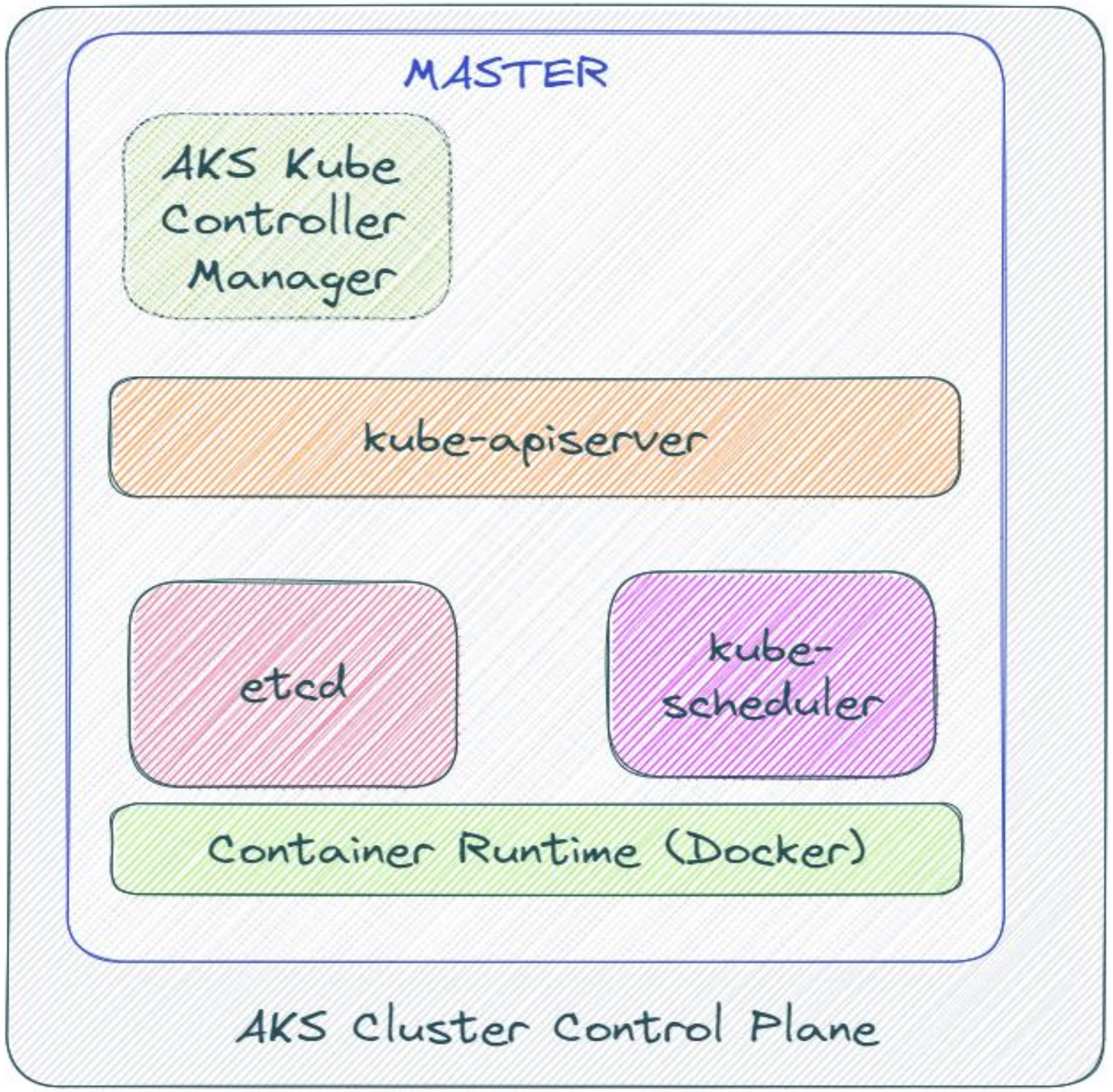
AZURE AKS
Create Cluster
Introduction

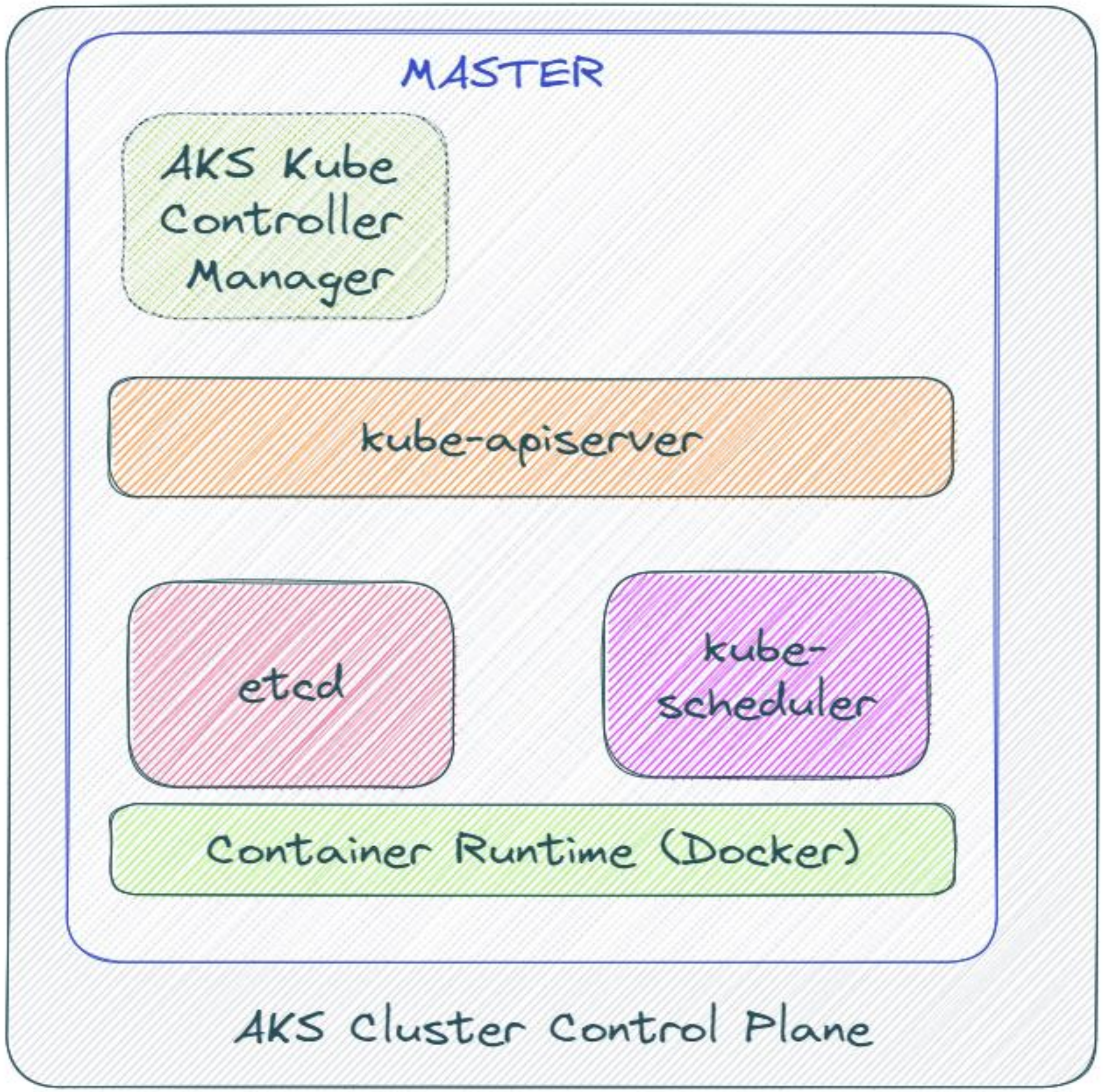
AKS - Introduction

- AKS – Azure Kubernetes Service
- AKS is highly available, secure and fully managed Service
- When compared to other cloud providers, AKS is the one which is available in highest number of regions
- Will be able to run any type of workloads
    - Windows based applications like .Net Apps
    - Linux supported applications like Java
    - IOT device deployment and management on demand
    - Machine Learning Model training with AKS
- Able to run in Hybrid Platforms
    - Azure Stack HCI
    - Windows Servers with Linux Distros
    - Planing for Vmware Platform


MASTER
AKS Kube Controller Manager
kube-apiserver
etcd
kube-scheduler
Container Runtime (Docker)
AKS Cluster Control Plane
Worker Node 1
Kubelet
Kube-Proxy
Container Runtime (Docker)
Worker Node 2
Kubelet
Kube-Proxy
Container Runtime (Docker)
AKS Node Pool


MASTER
AKS Kube Controller Manager
kube-apiserver
etcd
kube-scheduler
Container Runtime (Docker)
AKS Cluster Control Plane

- kube-apiserver
  - It acts as front end for the Kubernetes control plane. It exposes the Kubernetes API
  - Command line tools (like kubectl), Users and even Master components (scheduler, controller manager, etcd) and Worker node components like (Kubelet) everything talk with API Server.
- etcd
  - Consistent and highly-available key value store used as Kubernetes' backing store for all cluster data.
  - It stores all the masters and worker node information.
- kube-scheduler
  - Scheduler is responsible for distributing containers across multiple nodes.
  - It watches for newly created Pods with no assigned node, and selects a node for them to run on.



MASTER

AKS Kube Controller Manager

kube-apiserver

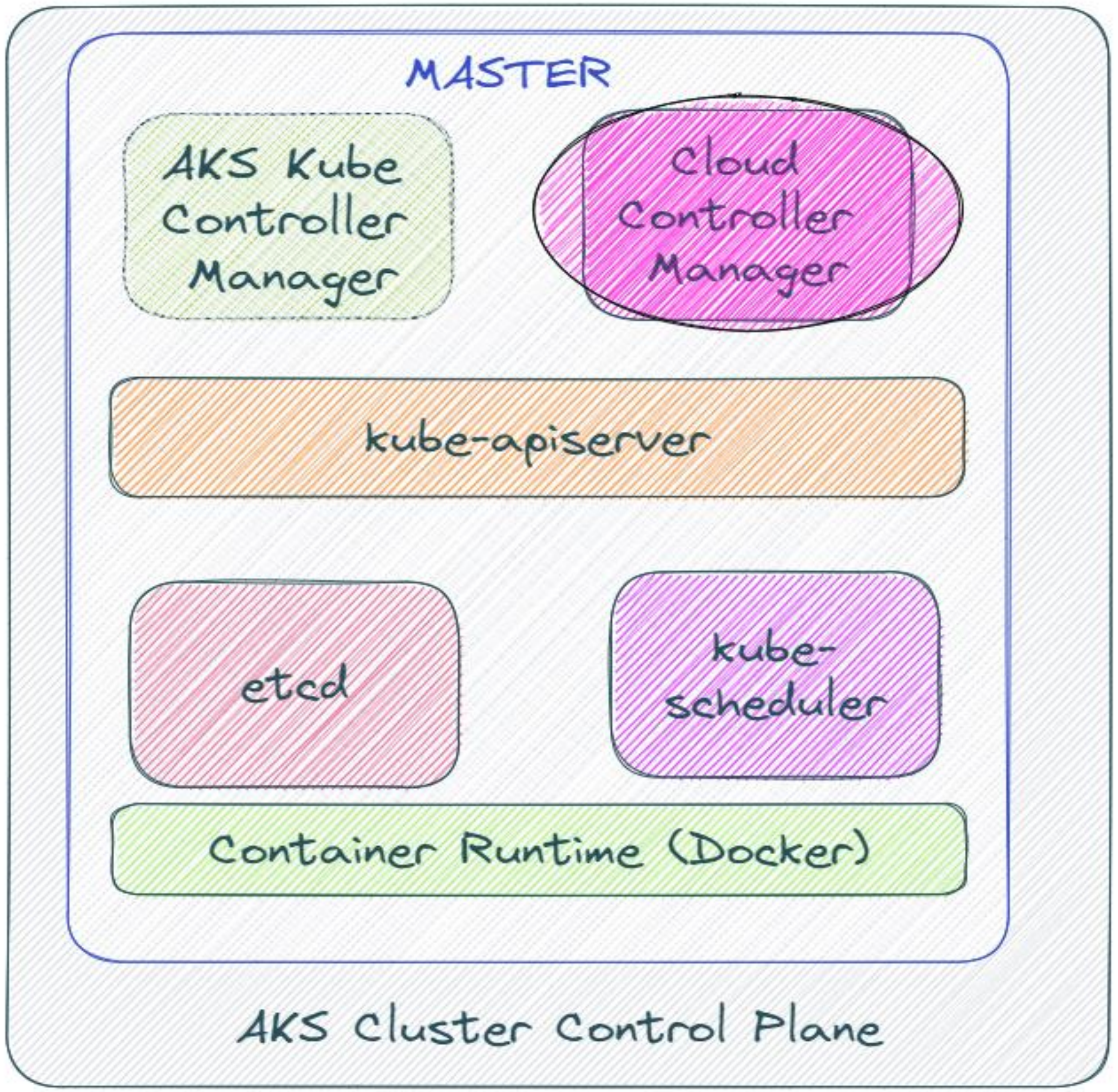
etcd

kube-scheduler

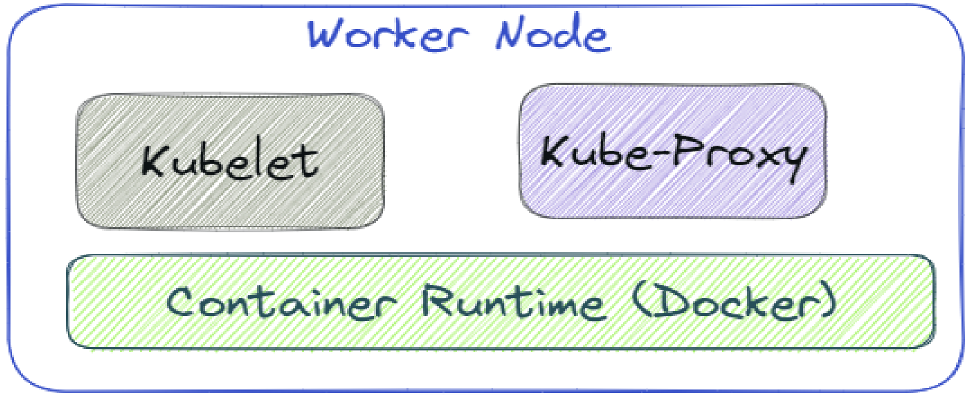
Container Runtime (Docker)

AKS Cluster Control Plane

- kube-controller-manager
  - Controllers are responsible for noticing and responding when nodes, containers or endpoints go down. They make decisions to bring up new containers in such cases.
  - Node Controller: Responsible for noticing and responding when nodes go down.
  - Replication Controller: Responsible for maintaining the correct number of pods for every replication controller object in the system.
  - Endpoints Controller: Populates the Endpoints object (that is, joins Services & Pods)
  - Service Account & Token Controller: Creates default accounts and API Access for new namespaces.


MASTER
AKS Kube Controller Manager
Cloud Controller Manager
kube-apiserver
etcd
kube-scheduler
Container Runtime (Docker)
AKS Cluster Control Plane

- cloud-controller-manager
  - A Kubernetes control plane component that embeds cloud-specific control logic.
  - It only runs controllers that are specific to your cloud provider.
  - On-Premise Kubernetes clusters will not have this component.
  - Node controller: For checking the cloud provider to determine if a node has been deleted in the cloud after it stops responding.
  - Route controller: For setting up routes in the underlying cloud infrastructure.
  - Service controller: For creating, updating and deleting cloud provider load balancer.



○ Container Runtime

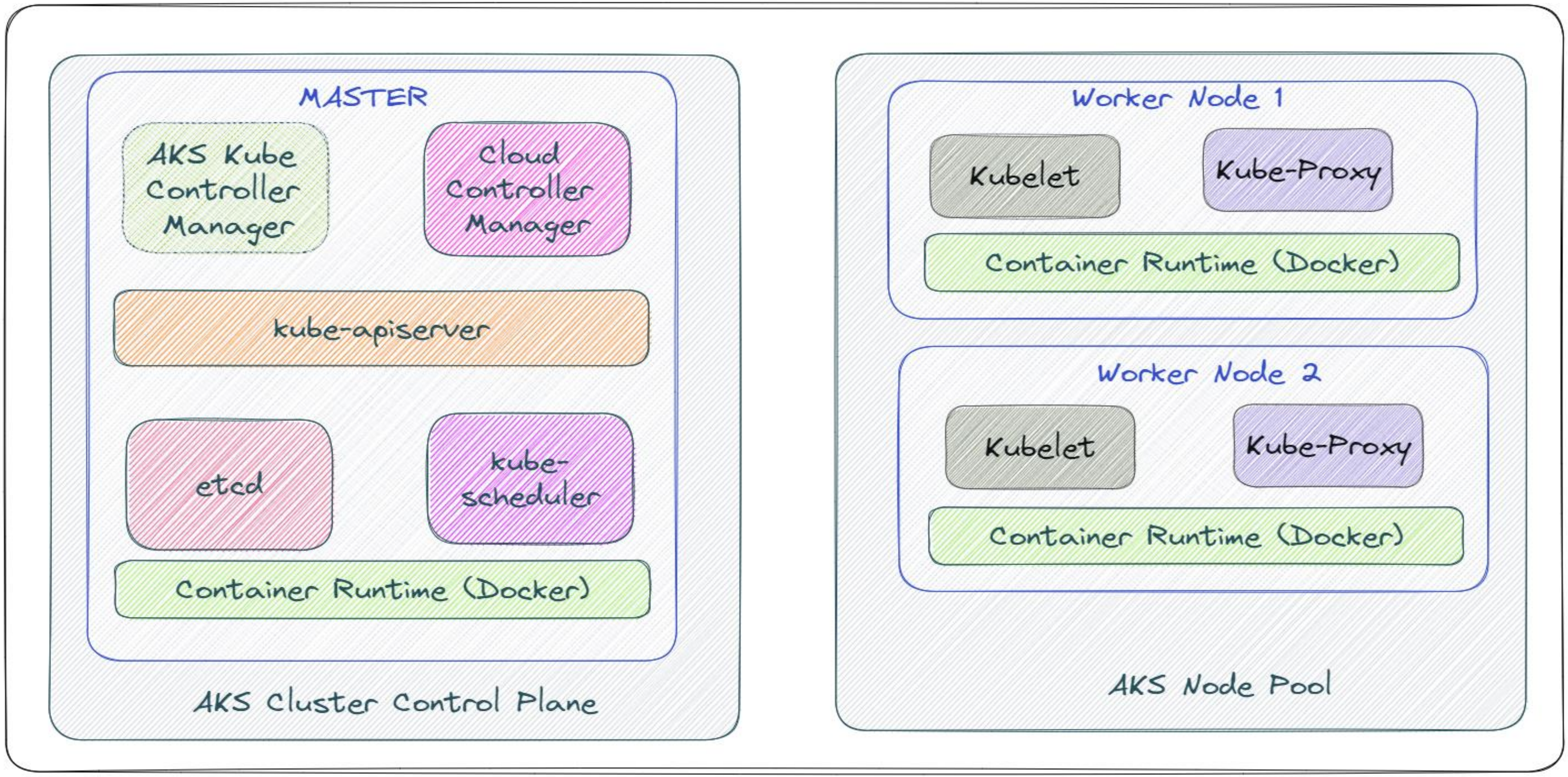
- Container Runtime is the underlying software where we run all these
- We are using Docker, but we have other runtime options like rkt, container-d etc.

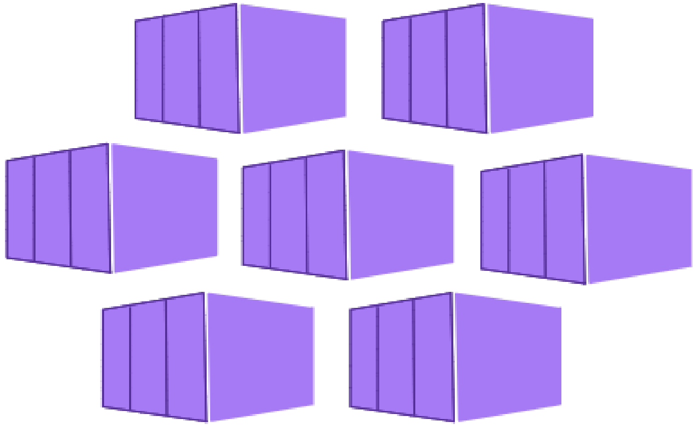
○ Kubelet

- Kubelet is the agent that runs on every node in the cluster
- This agent is responsible for making sure that containers are running in a Pod on a node.

○ Kube-Proxy

- It is a network proxy that runs on each node in your cluster.
- It maintains network rules on nodes
- In short, these network rules allow network communication to your Pods from network sessions inside or outside of your cluster.


MASTER
AKS Kube Controller Manager
Cloud Controller Manager
kube-apiserver
etcd
kube-scheduler
Container Runtime (Docker)
AKS Cluster Control Plane
Worker Node 1
Kubelet
Kube-Proxy
Container Runtime (Docker)
Worker Node 2
Kubelet
Kube-Proxy
Container Runtime (Docker)
AKS Node Pool



Kubernetes Fundamentals

Pod,ReplicaSet, Deployment and Service.