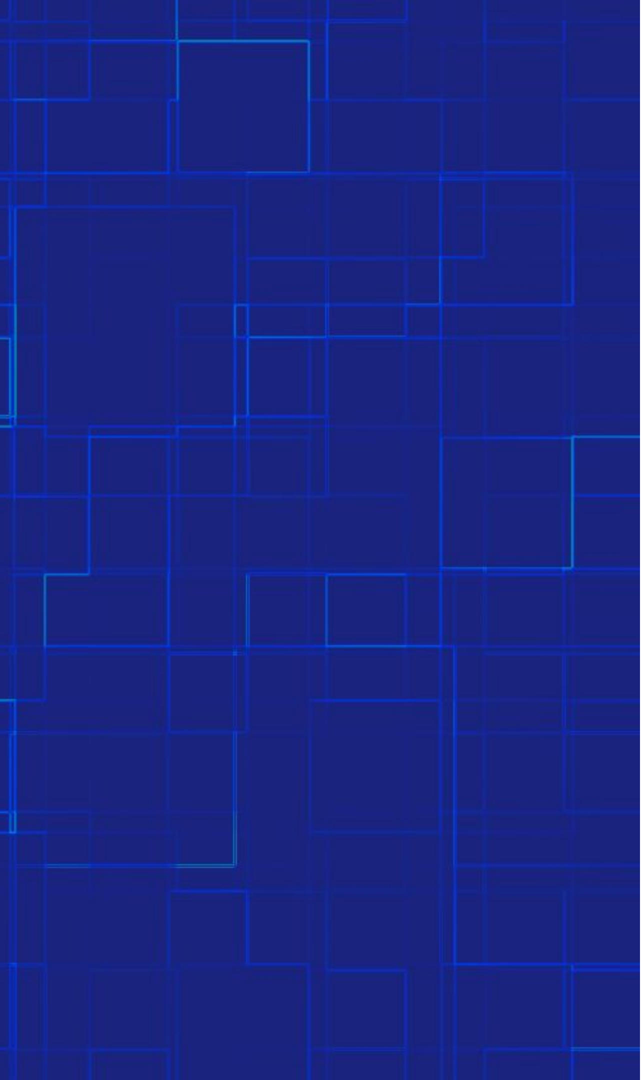
Welcome

I am **Md Nasir Uddin** here as your Certified Kubernetes Administrator (CKA) Course Instructure.

Md Nasir uddin
DevOps Engineer (Manager)
Anwar Enterprise System Ltd
2x Kubernetes, 2x AWS, 2x Linux, NSE4, CCNA

nasirnjs@gmail.com



What is Kubernetes?

Kubernetes is an open source container orchestration platform that automates many of the manual processes involved in deploying, managing, and scaling containerized applications.

Who contributes to Kubernetes?

Kubernetes was originally developed and designed by engineers at Google.

- 2014 Google Introduces Kubernetes
- mid-2014: Google introduced Kubernetes as an open source version of Borg
- June 7 Initial release – first github commit for Kubernetes
- July 10 Microsoft, RedHat, IBM, Docker joins the Kubernetes community.

Google donated the Kubernetes project to the newly formed Cloud Native Computing Foundation (CNCF) in 2015.

What is Kubernetes?

Kubernetes is an open-source container orchestration system for automating software deployment, scaling, and management.

To understand Kubernetes, we must first understand two things:

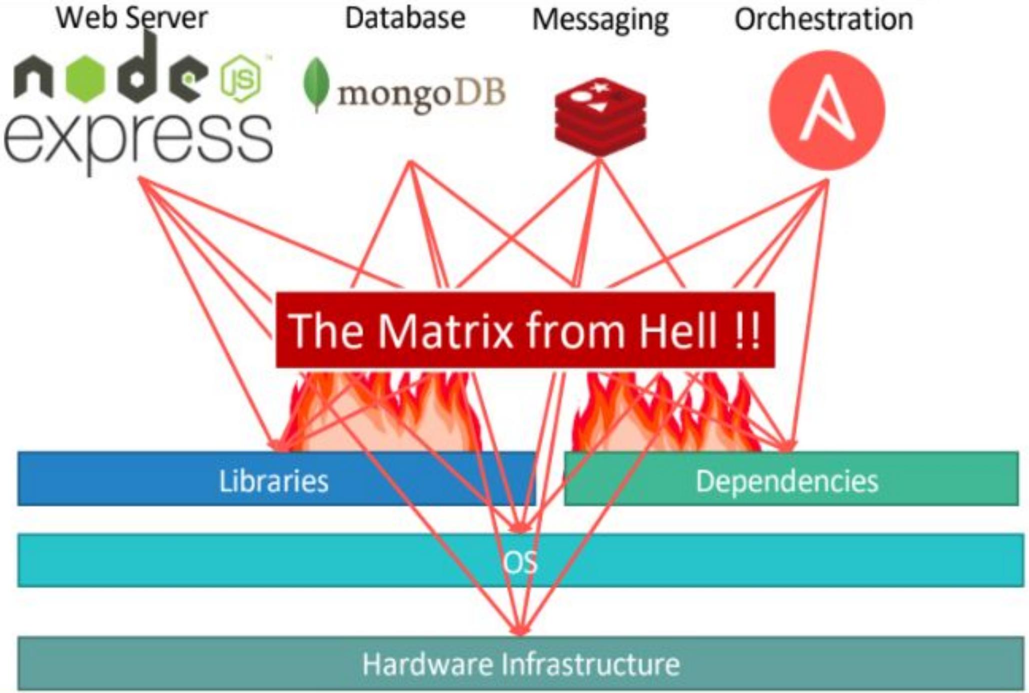
1. **Container**
2. **Orchestration**

Why do you need containers?

Compatibility/Dependency

Long setup time

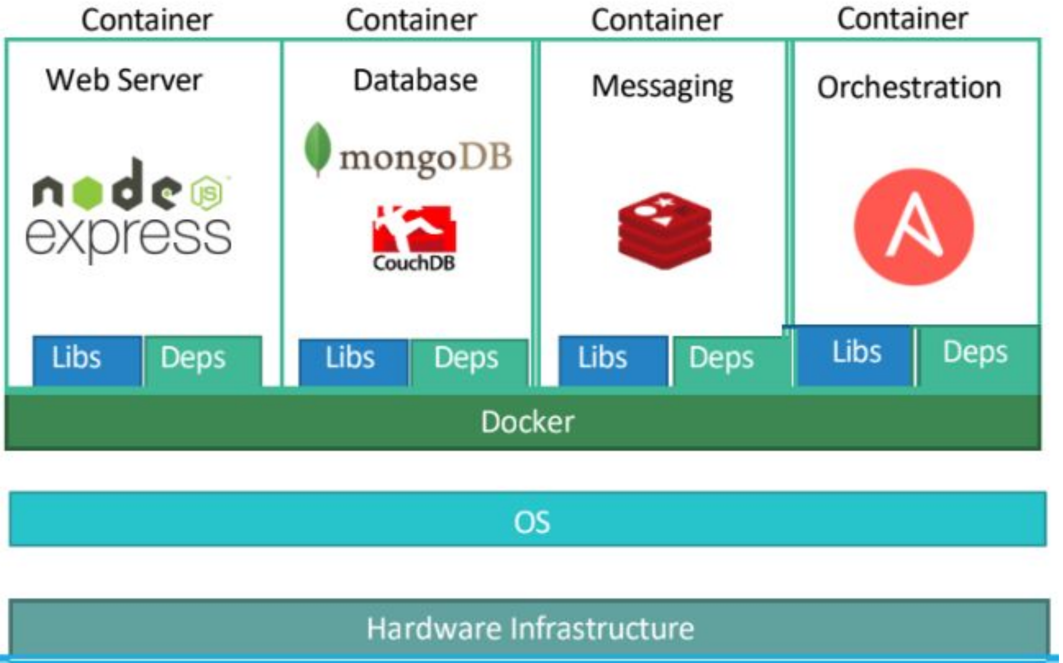
Different Dev/Test/Prod environments



What can it Do?

Containerize Applications

Run each service with its own dependencies in separate containers



Container
Web Server
Libs
Deps

Container
Database
Libs
Deps

Container
Messaging
Libs
Deps

Container
Orchestration
Libs
Deps

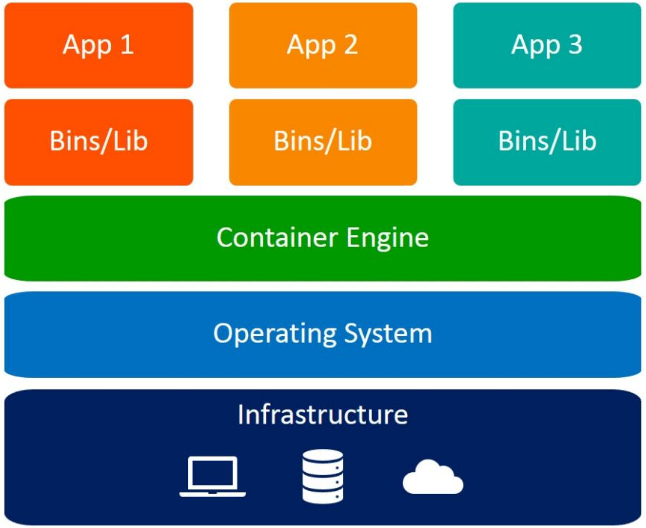
Docker

OS

Hardware Infrastructure

So what are containers?

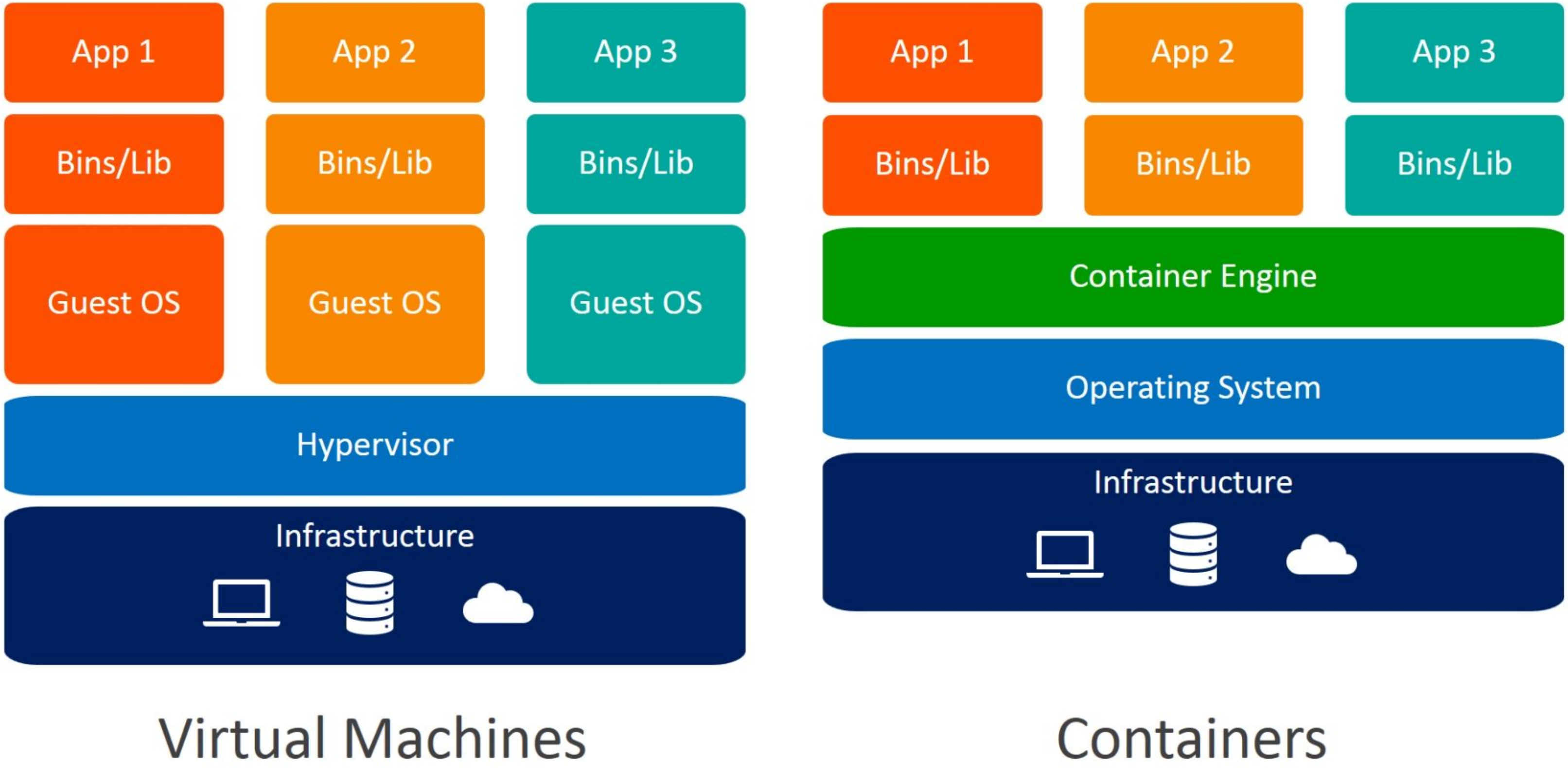
- Containers are completely isolated environments, as in they can have their own processes or services, their own network interfaces, their own mounts.
- Just like Virtual machines, except that they all share the same OS kernel.



nasirnjs@gmail.com

kubernetes

Container vs Virtual Machines



nasirnjs@gmail.com

kubernetes

Container Orchestration!!

Orchestration a platform to orchestrate the connectivity between the containers.

Automatically scale up or down based on the load, automatically deploying and managing containers is known as Container Orchestration.

Top Container Orchestration:

1. Kubernetes
2. Red Hat OpenShift
3. Apache Mesos
4. Mirantis Kubernetes Engine
5. Helios
6. Amazon Elastic Container Service (Amazon ECS)
7. Google Kubernetes Engine (GKE)
8. Azure Service Fabric
9. Amazon Elastic Kubernetes Service (EKS)
10. Docker Swarm

What are Kubernetes clusters?

Kubernetes cluster is a set of interconnected physical or virtual machines (nodes) that work together to run containerized applications.

Kubernetes clusters can span hosts across on-premise, public, private, or hybrid clouds.

Here are some key components of a Kubernetes cluster:

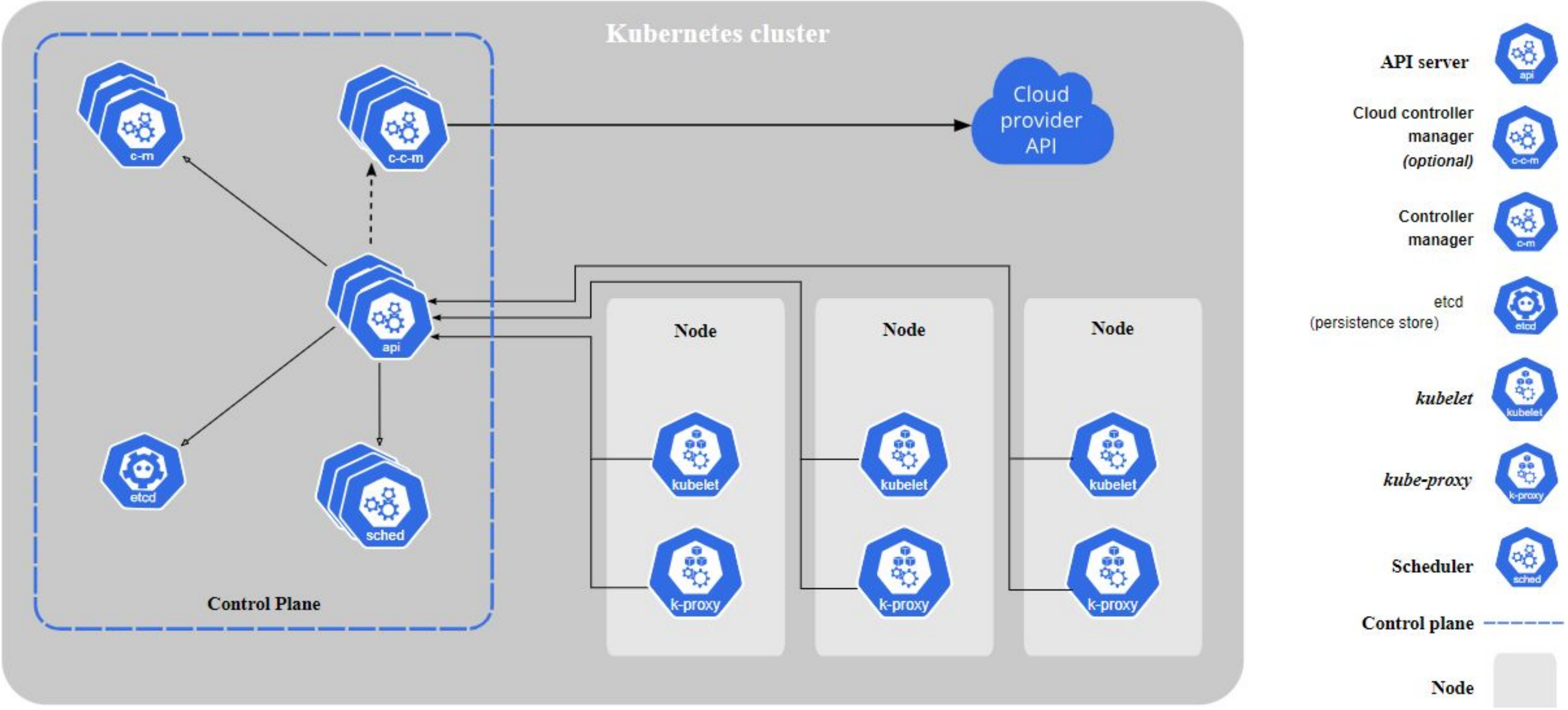
Master Node, Worker Node, etcd, Kubelet, Kube Proxy, Pods, Controller Manager, Scheduler

Kubernetes Components and Architecture!

Kubernetes Architecture?

- Kubernetes follows a client server architecture.
- It’s possible to have a multi-master setup (for high availability), but by default there is a single master server which acts as a controlling node and point of contact.

Kubernetes Components and Architecture!



nasirnjs@gmail.com

kubernetes

key components of a Kubernetes cluster!!

Master Node: The Master Node in a Kubernetes cluster is responsible for managing and controlling the overall cluster's state and operations.

Master node several key components:

Kubernetes API Server: Kubernetes API, which allows users, administrators, and other components to interact with the cluster. All requests and commands, including creating, updating, and deleting resources, are processed by the API server.

etcd: etcd is a distributed key-value store used for storing the configuration data and the state of the cluster.

key components of a Kubernetes cluster!!

Controller Manager: Responsible for maintaining the desired state of different Kubernetes resources. Example, Replication Controller, ReplicaSet Controller, Deployment Controller, and others.

Scheduler: The Scheduler is responsible for making decisions about where to place newly created pods on worker nodes.

Container Runtime: Responsible for running containers

The master node components work together to maintain the desired state of the cluster, monitor the health of worker nodes, schedule & distribute workloads, & handle API requests and resource management.

key components of a Kubernetes cluster!!

Worker Node: Worker nodes are responsible for running the actual containerized applications and services.

Kubelet: The Kubelet is an agent that runs on each Worker Node.

It communicates with the Kubernetes Master Node and ensures that containers are running in Pods as specified by the desired state defined in Kubernetes manifests.

Container Runtime: Container Runtime responsible for running containers. Docker most popular choices.

Other runtimes like containerd and CRI-O are also supported.

key components of a Kubernetes cluster!!

Kube Proxy: Kube Proxy is a network proxy service that runs on each Worker Node. It is responsible for network routing and load balancing.

Kube Proxy maintains network rules to enable communication between Pods within the cluster and from external sources.

CNI (Container Network Interface) Plugin: The CNI plugin is responsible for configuring network connectivity for Pods.

Popular CNI plugins include Calico, Flannel, and Weave, among others.