INTRODUCTION OF KUBERNETES

TRANG NGUYEN PRESENTATION

Table of Contents

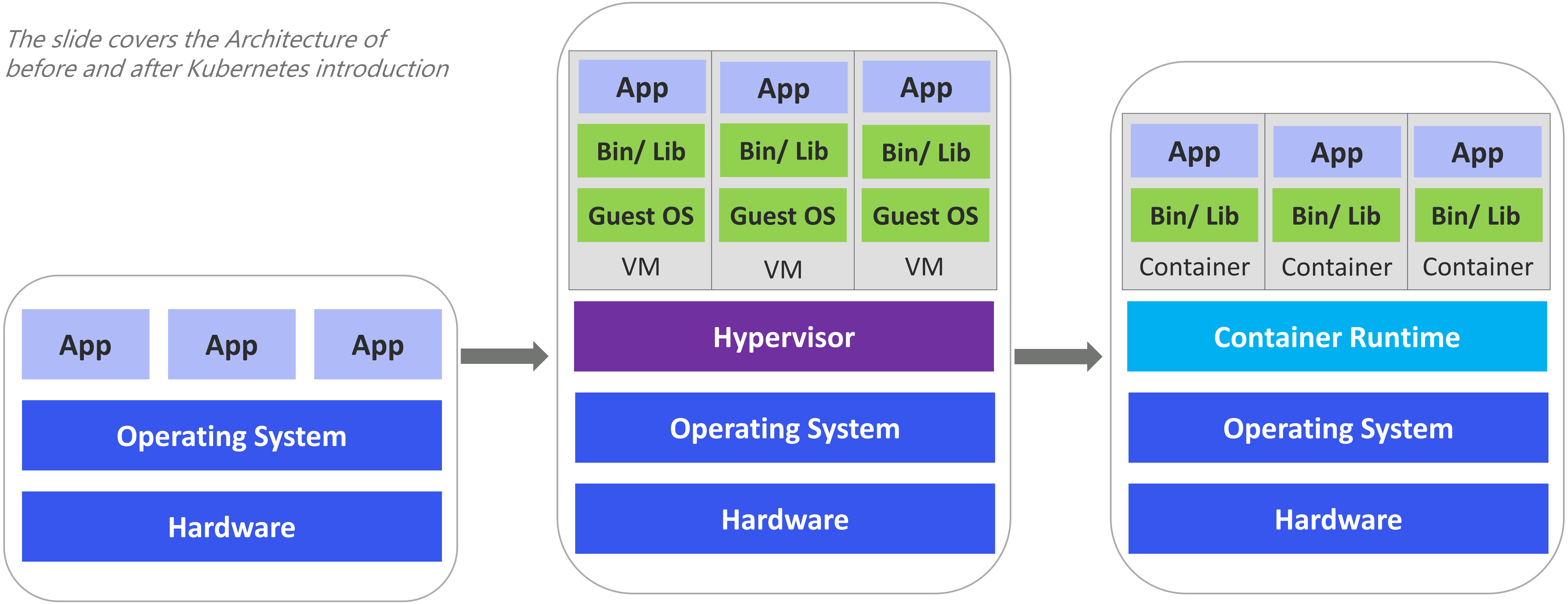
1

Introduction of Kubernetes

- Before Kubernetes
- Benefits of Containers
- Are containers the same thing as microservices?
- Microservices Architecture
- Microservices Architecture – Ecommerce
- Container Architecture
- Microservices use cases
- What is Kubernetes?
- Why organization should use Kubernetes?
- What Kubernetes can do?
- What Kubernetes can’t do?
- Roadmap to install Kubernetes
- Features of Kubernetes
- Kubernetes Architecture
- Kubernetes vs Docker Swarm
- Kubernetes 7 use cases
- Advantage of Kubernetes
- 30-60-90 day plan in Kubernetes

Before Kubernetes

The slide covers the Architecture of before and after Kubernetes introduction



App | App | App

Operating System

Hardware

App | App | App

Bin/ Lib | Bin/ Lib | Bin/ Lib

Guest OS | Guest OS | Guest OS

VM | VM | VM

Hypervisor

Operating System

Hardware

App | App | App

Bin/ Lib | Bin/ Lib | Bin/ Lib

Container | Container | Container

Container Runtime

Operating System

Hardware

Traditional Deployment

. Resource allocation issues
. If each application on a different physical server -> can’t
scale & expensive to maintain many physical servers.

Virtualized Deployment

. High cost of implementation
. Security risk
. Availability issue
. Limitations & Restrictions
. Time consuming

Container Deployment

Containers are similar to VMs, but share the OS among the applications -> portable across clouds and OS distributions.

Benefits of Containers

This slide highlights important factors why companies needs containers

300% Faster Time to Market

- ✓ Newer applications and services keep the competitive edge
- ✓ Agile application creation and deployment
- ✓ CI/CD pipeline
- ✓ Dev and Ops separation of concerns

Gain Freedom of Choice

- ✓ Cloud and OS distribution portability: Runs on Ubuntu, RHEL, CoreOS, on-premises, on major public clouds, and anywhere else

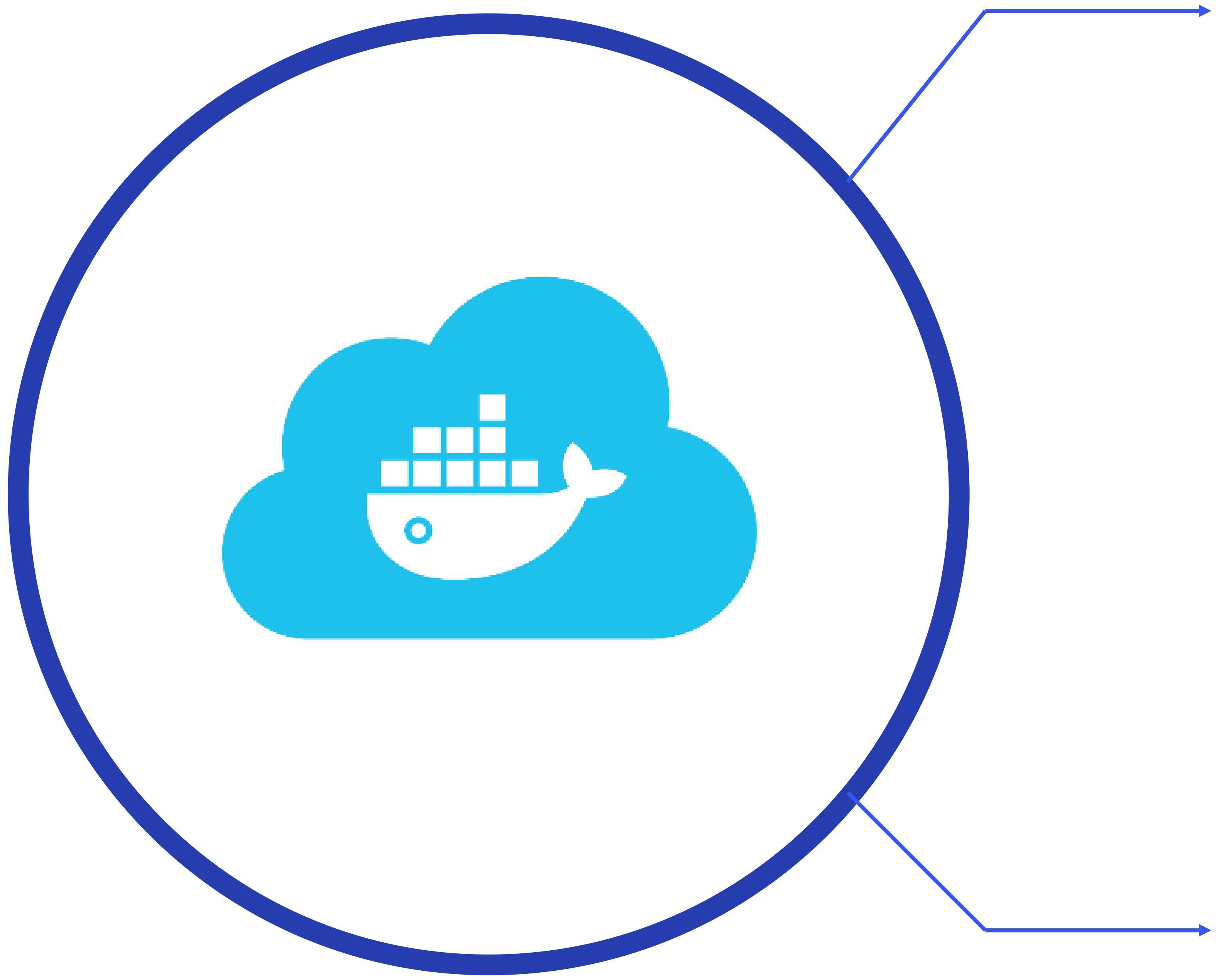
- 40% IT Infrastructure Reduction

- ✓ Increase application work density
- ✓ Recover utilization of the server density
- ✓ Reduce package licensing prices

40% IT Operational Efficiency

- ✓ Automate the management of various applications and infrastructure into one operational model

Are Containers the Same Thing as Microservices?



Microservices is an style of architecture.

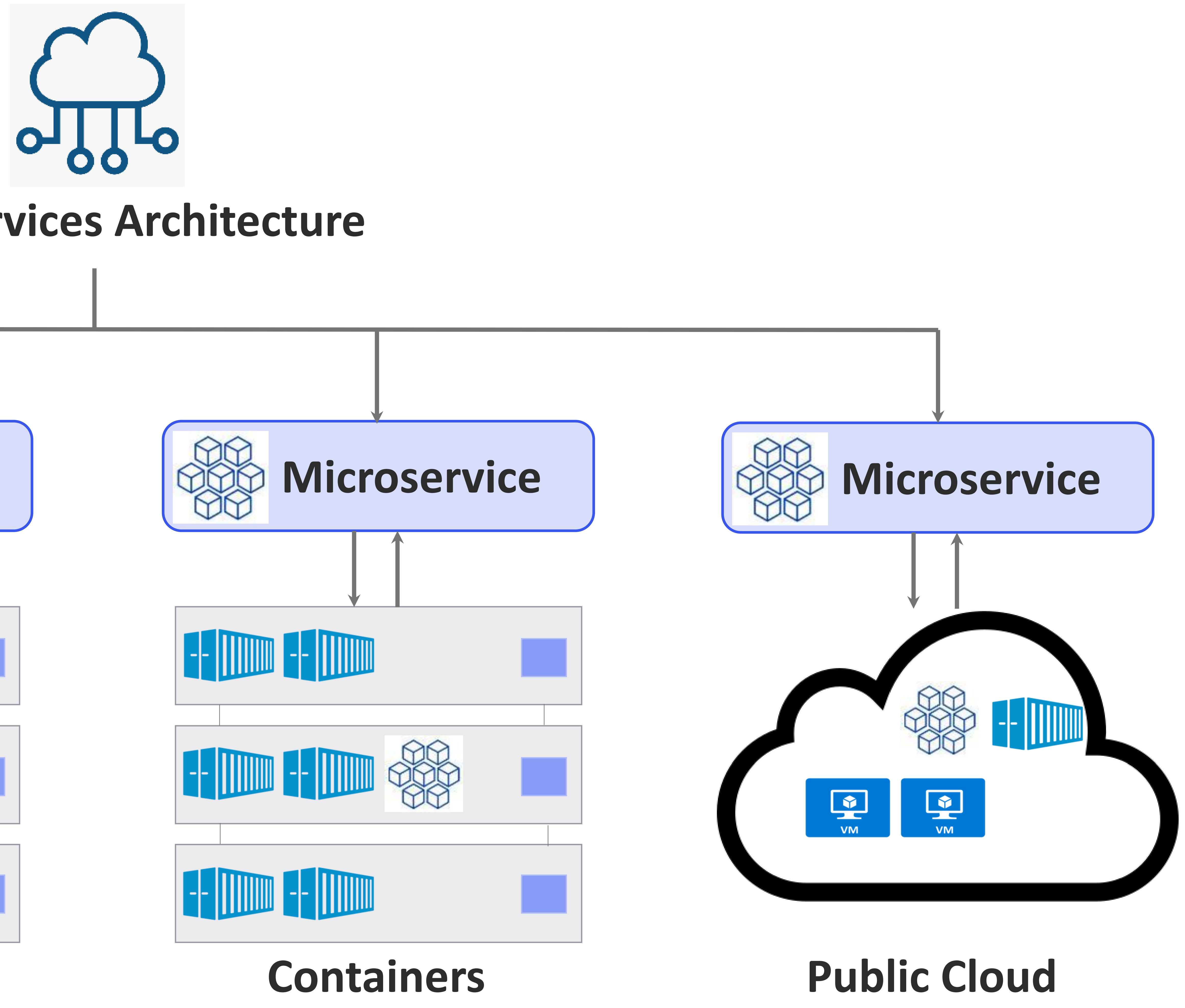
A microservices design structures an application as a set of loosely coupled, collaborating services that deliver specific business capabilities.

Containers facilitate build it happen.

Containers are a lightweight, efficient and standard way for applications to move between environments and run independently.

Microservices Architecture

This slide is to highlight the Microservices Architecture of Containers



Microservices Architecture

Microservice

Microservice

Microservice

Microservice

Bare Metal

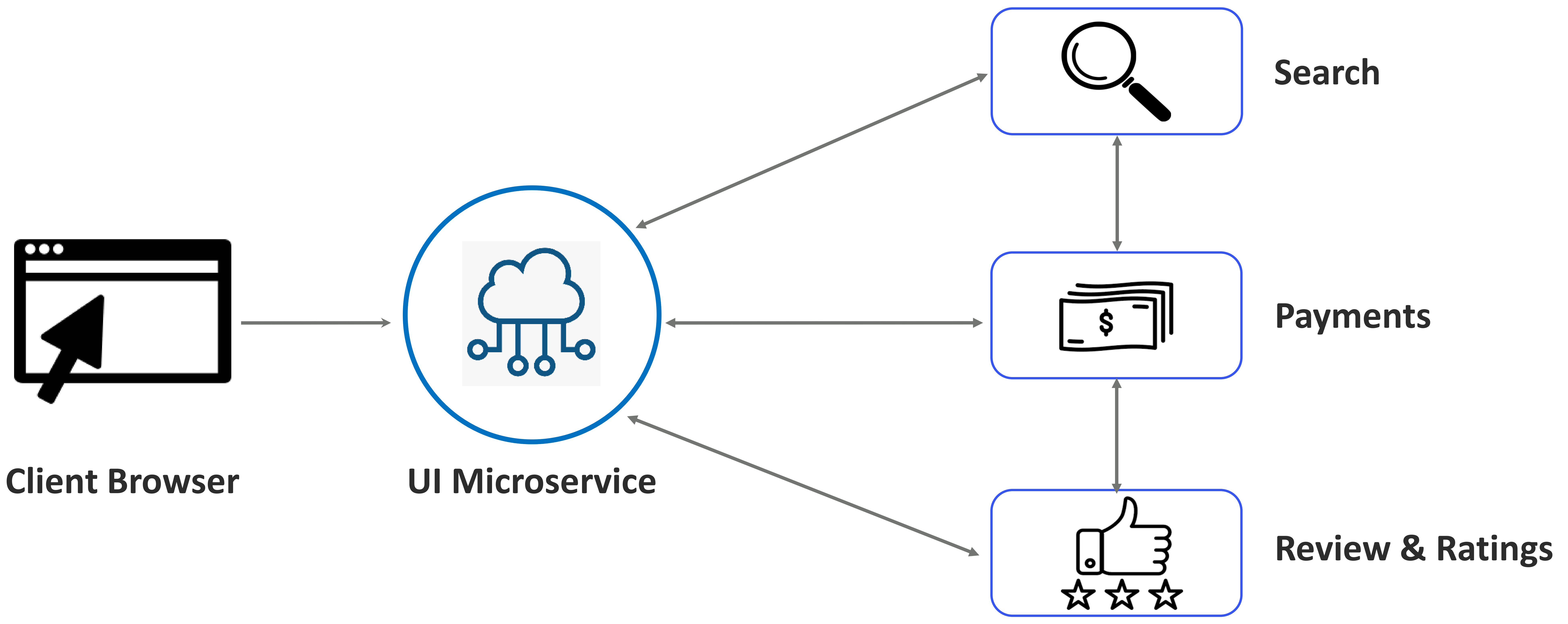
Virtualized

Containers

Public Cloud

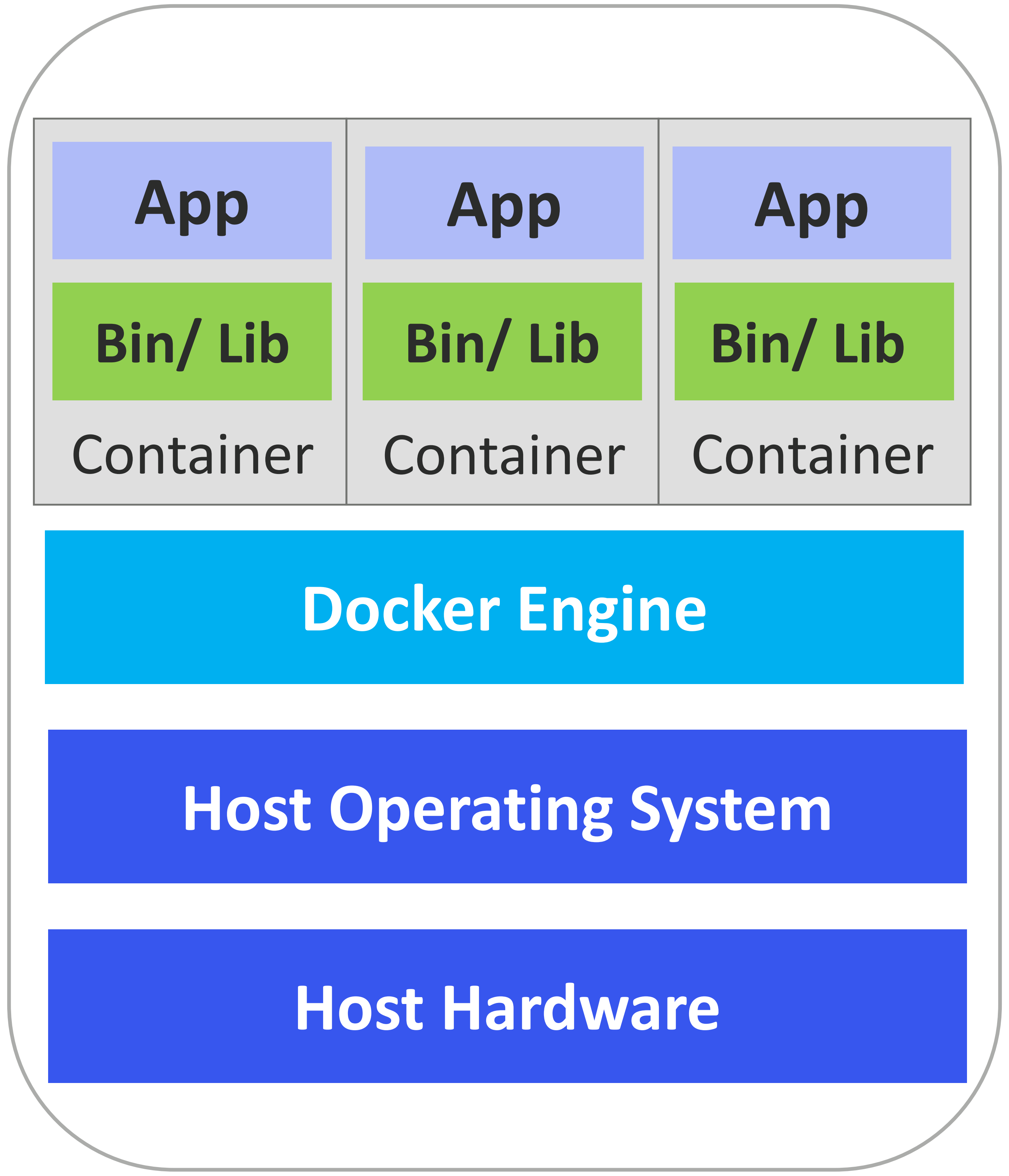
Applications

Microservices Architecture E-Commerce Application



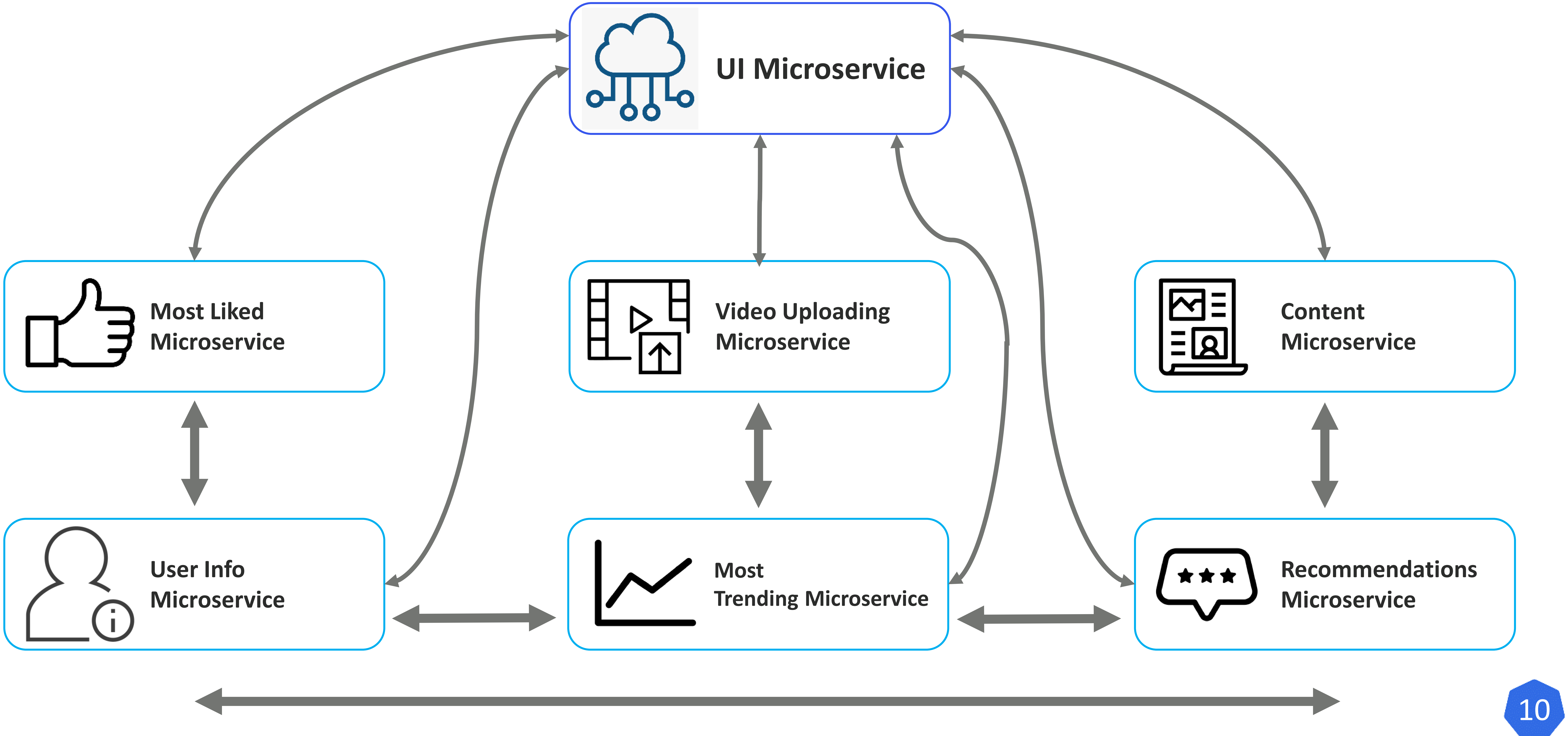
Container Architecture

This slide highlights the core Architecture of Containers and Applications hosted to the Docker Engine.



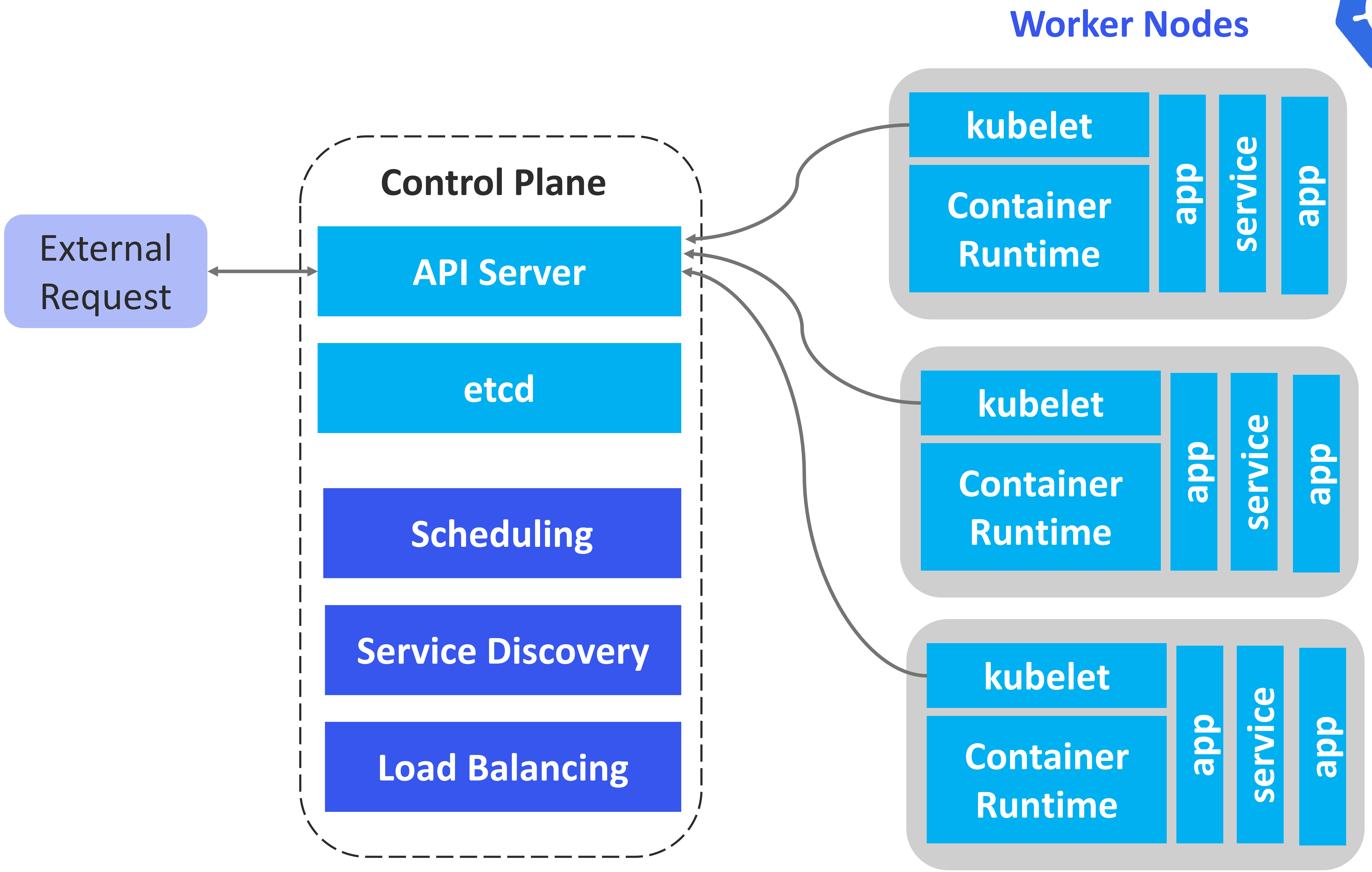
Microservices Use Cases

The slide covers the major microservices use cases such as user info microservices, most trending use cases and recommendations microservices.


UI Microservice
Most Liked Microservice
Video Uploading Microservice
Content Microservice
User Info Microservice
Most Trending Microservice
Recommendations Microservice

What is Kubernetes?

The slide shows the overview of Kubernetes.

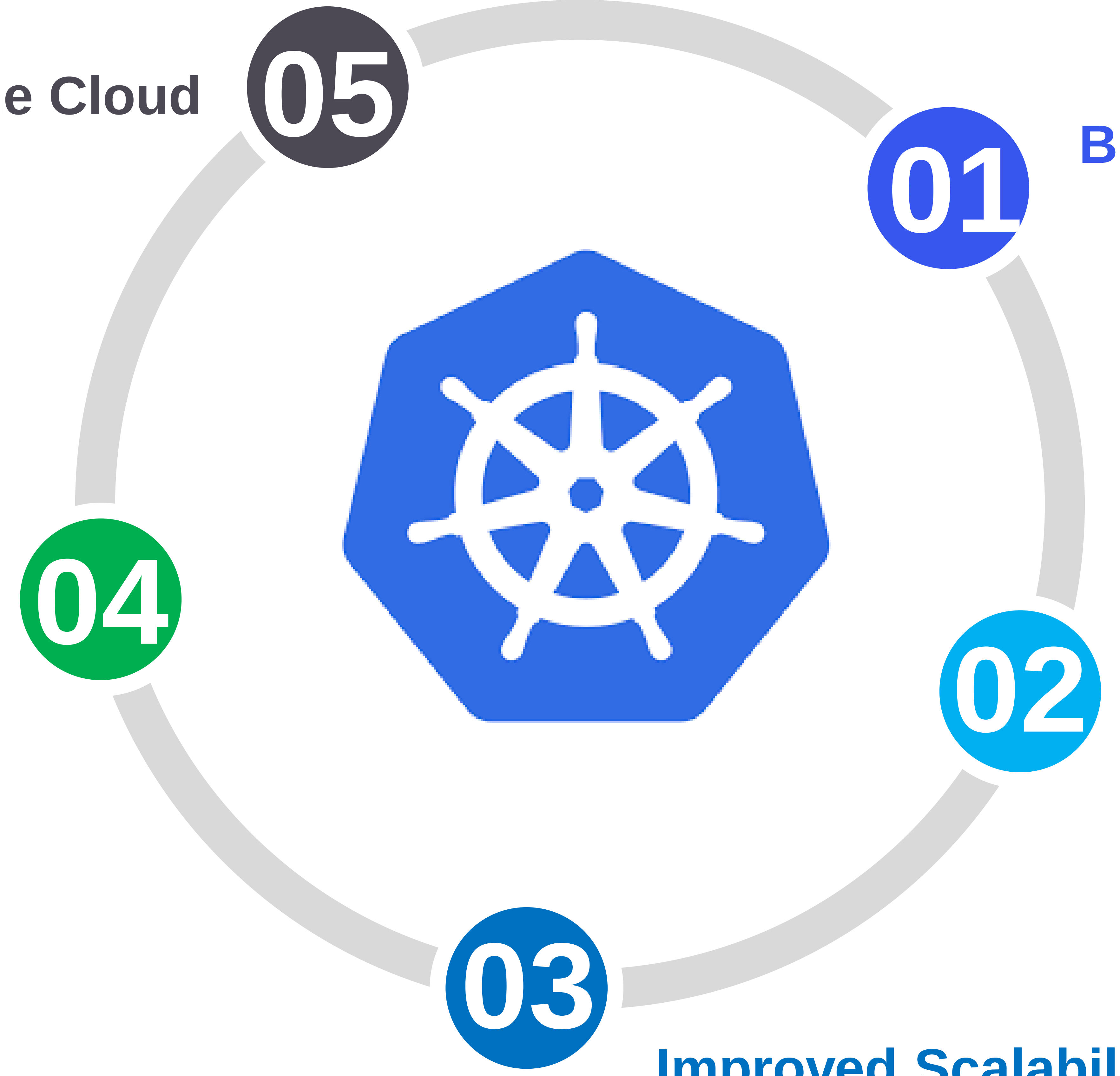


KUBERNETES

- ✓ designed by Google
- ✓ currently maintained by the Cloud Native Computing Foundation
- ✓ an open-source container orchestration system
- ✓ help to automate application deployment, scaling, and management

Why Organization Should use Kubernetes?

The slide highlights the core reasons why organization should opt Kubernetes.



01 Better Management through Modularity

02 IT Cost Optimization

03 Improved Scalability and Availability

04 Multi-cloud (and Hybrid Cloud) Flexibility

05 Effective Migration to the Cloud

What Kubernetes can do?

Service discovery and load balancing

- ✓ load balance and distribute the network traffic so that the deployment is stable.

Storage orchestration

- ✓ mount a storage system of your choice automatically, such as local storages, public cloud providers, and more.

Automated rollouts and rollbacks

- ✓ create new containers automatically.
- ✓ remove existing containers and adopt all their resources to the new container.

Automatic bin packing

- ✓ fit containers onto your nodes with CPU and memory (RAM) predefined to make the best use of your resources.

Self-healing

- ✓ restarts containers that fail, replaces containers, kills containers that don't respond to your user-defined health check

Secret and configuration management

- ✓ store and manage sensitive information
- ✓ update secrets and application configuration without rebuilding your container images

What Kubernetes can’t do?

Containers runtime

Kubernetes simply manages the containers once they are running. Kubernetes supports a range of container runtimes e.g. Docker

Containerizing applications

You have to package the application into containers, then give Kubernetes the container images you've built to deploy them.

Container image management

Kubernetes natively integrates with several 3rd-party registries to automate storage or management of the images.

Infrastructure provisioning

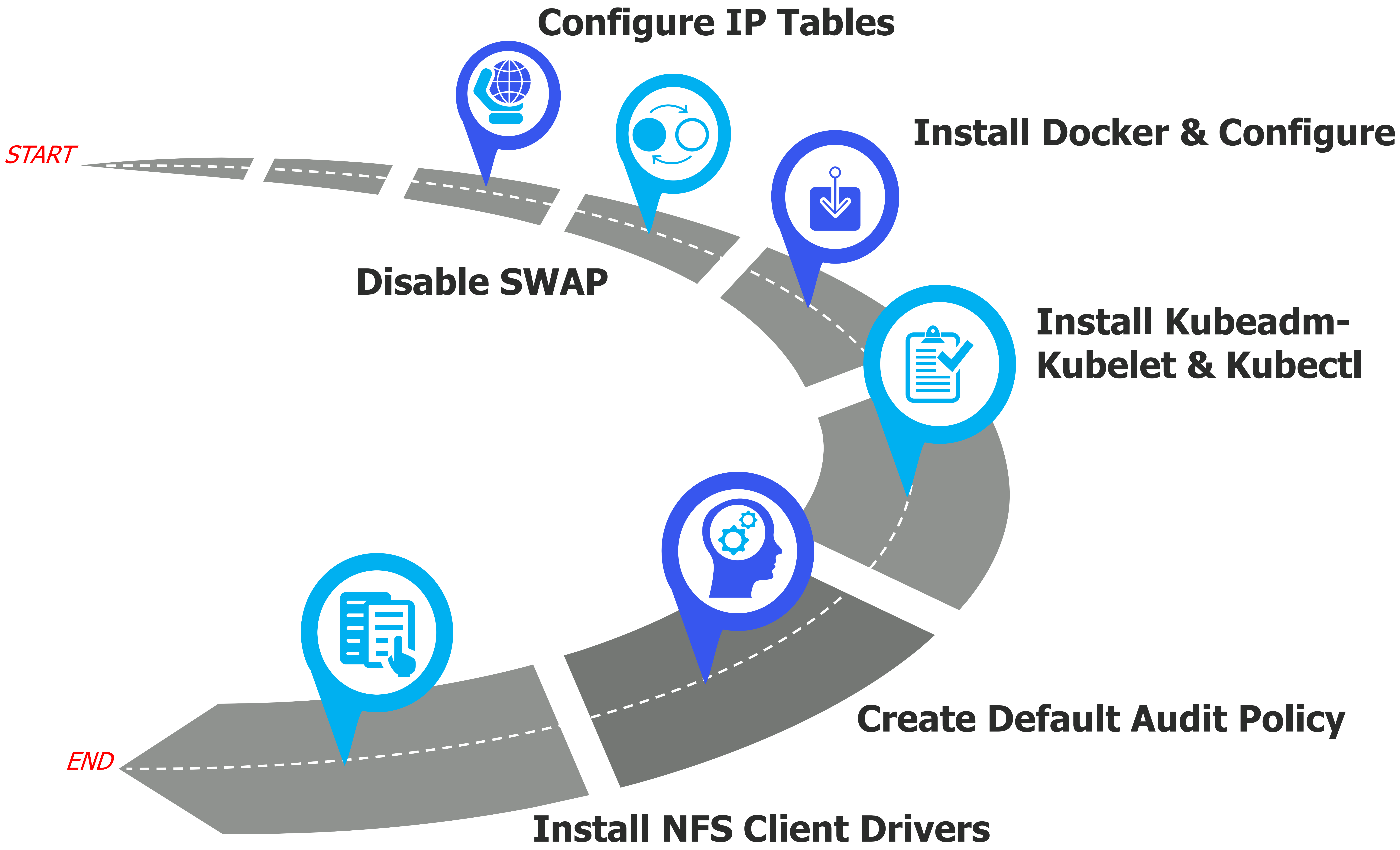
Kubernetes only manages the workloads that run on top of servers. Kubernetes can’t magically generate more server resources.

Security

Kubernetes can't detect malware inside container images or alert you to anomalous behavior. You need external tools to achieve these tasks.

Roadmap to install Kubernetes

The slide highlights the roadmap to install Kubernetes in the organization.



START
Configure IP Tables
Disable SWAP
Install Docker & Configure
Install Kubeadm-Kubelet & Kubectl
Create Default Audit Policy
Install NFS Client Drivers
END

Features of Kubernetes

01
Automatic Bin Packing

02
Service Discovery & Load Balancing

03
Storage Orchestration

04
Self Healing

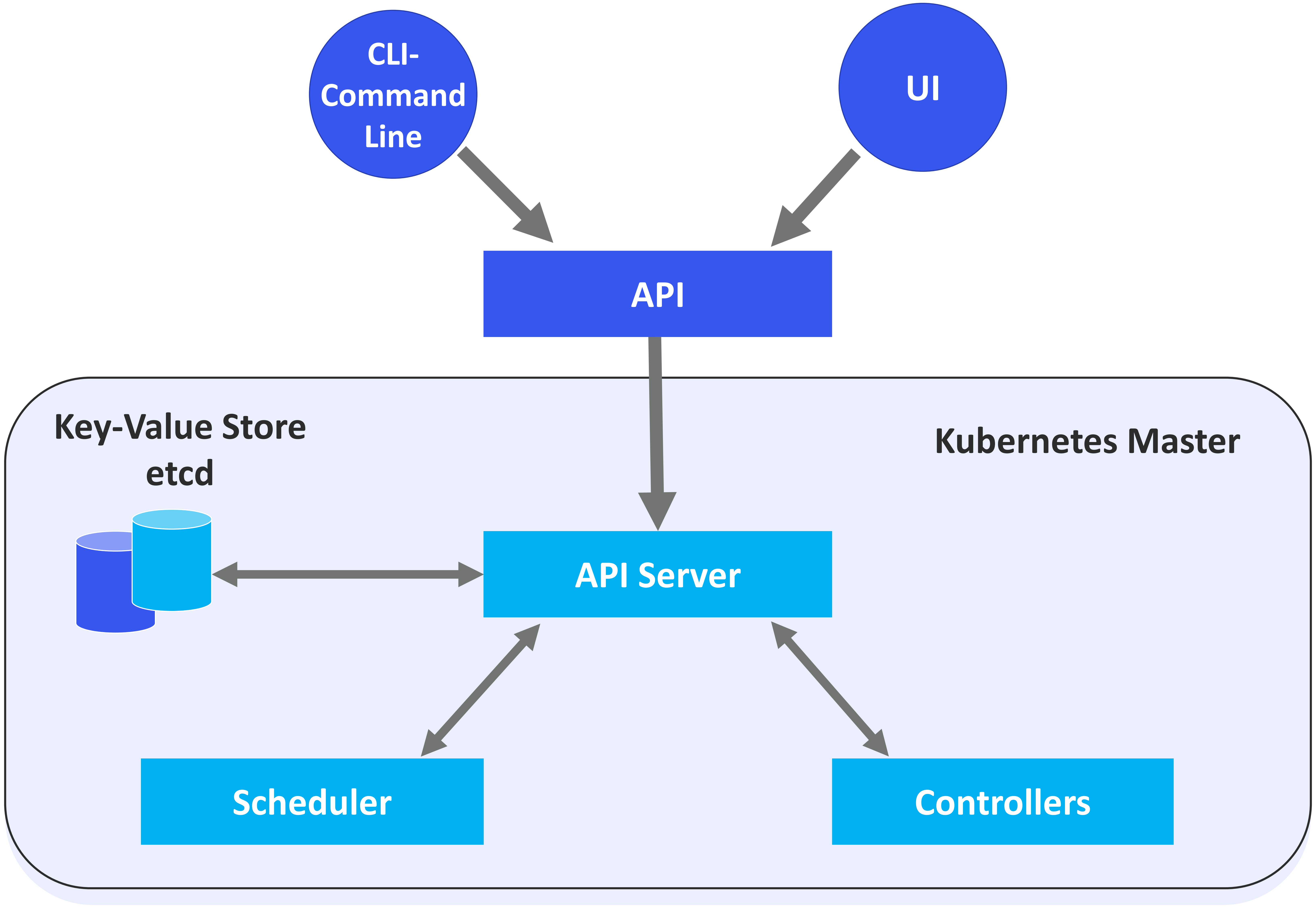
05
Secret & Configuration Management

06
Batch Execution

07
Horizontal Scaling

08
Automatic Rollbacks & Rollouts

Kubernetes Architecture



Kubernetes vs Docker Swarm

The slide provides the key difference between Kubernetes and Docker Swarm

Docker Swarm

01 No Auto Scaling

02 Good Community

03 Easy to Start a Cluster

04 Limited to the Docker API's Capabilities

05 Does not have as Much Experience with Production Deployments at Scale

Kubernetes

01 Auto Scaling

02 Great Active Community

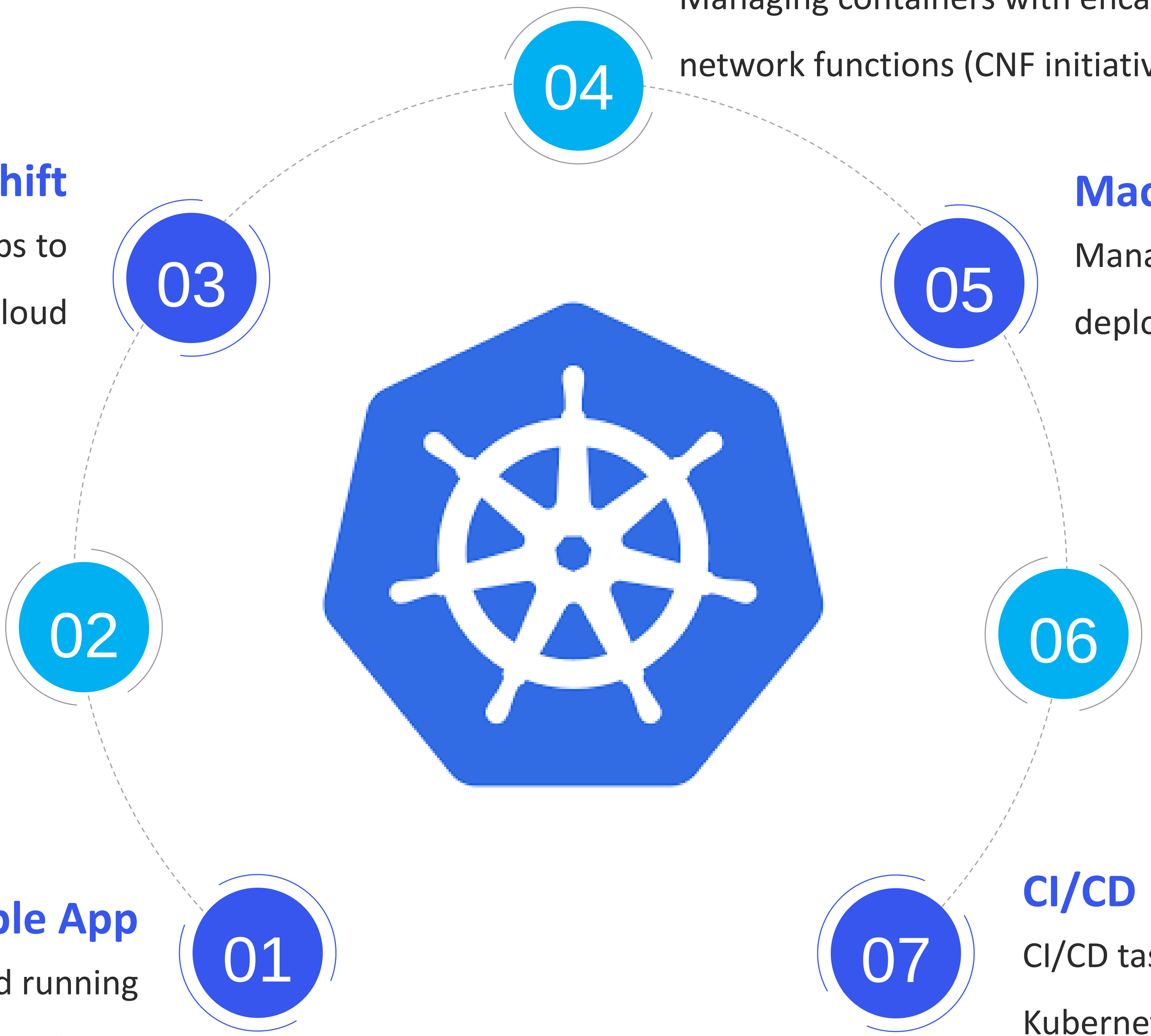
03 Difficult to Start a Cluster

04 Can Overcome Constraints of Docker and Docker API

05 Deployed at Scale more often among Organizations

Kubernetes 7 use Cases

The slide contains the important 7 use cases of Kubernetes



01 **Simple App**

Learning how to get your app up and running by deploying it on a Kubernetes cluster

02 **Microservices**

Orchestrating complicated apps based on a microservice architecture with many components

03 **Lift and Shift**

Facilitating the moving of on-prem apps to the cloud

04 **Cloud Native Network Functions**

Managing containers with encapsulated network functions (CNF initiatives)

05 **Machine Learning**

Managing machine learning workflows to deploy faster AI-based apps

06 **Heavy Computing**

Enabling computing-heavy tasks

07 **CI/CD**

CI/CD tasks are easier to perform with Kubernetes

Advantages of Kubernetes

The slide highlights the major advantages of Kubernetes

01 Using Kubernetes and its Brobdingnagian Scheme will improve your productivity

02 Kubernetes and a cloud-native technical school stack attracts talent

03 Kubernetes could be a future proof answer

04 Kubernetes helps to form your application run additional stable

05 Kubernetes is cheaper than its alternatives

06 Orchestrate containers on multiple hosts

30-60-90 Days Plan in Kubernetes

The slide highlights the 30-60-90 Days plan to implement Kubernetes

30 Days

60 Days

90 Days

- ✓ Starting out with containers and container orchestration tools
- ✓ Install Kubernetes
- ✓ Load balancing with Kubernetes
- ✓ Two-step load-balancer setup
- ✓ Configuring load balancing

- ✓ Implement Blue-green deployments in Kubernetes
- ✓ Making the deployments automatic
- ✓ Know your resource constraints

- ✓ Enable Logging
- ✓ Start Monitoring
- ✓ Fixing Issues

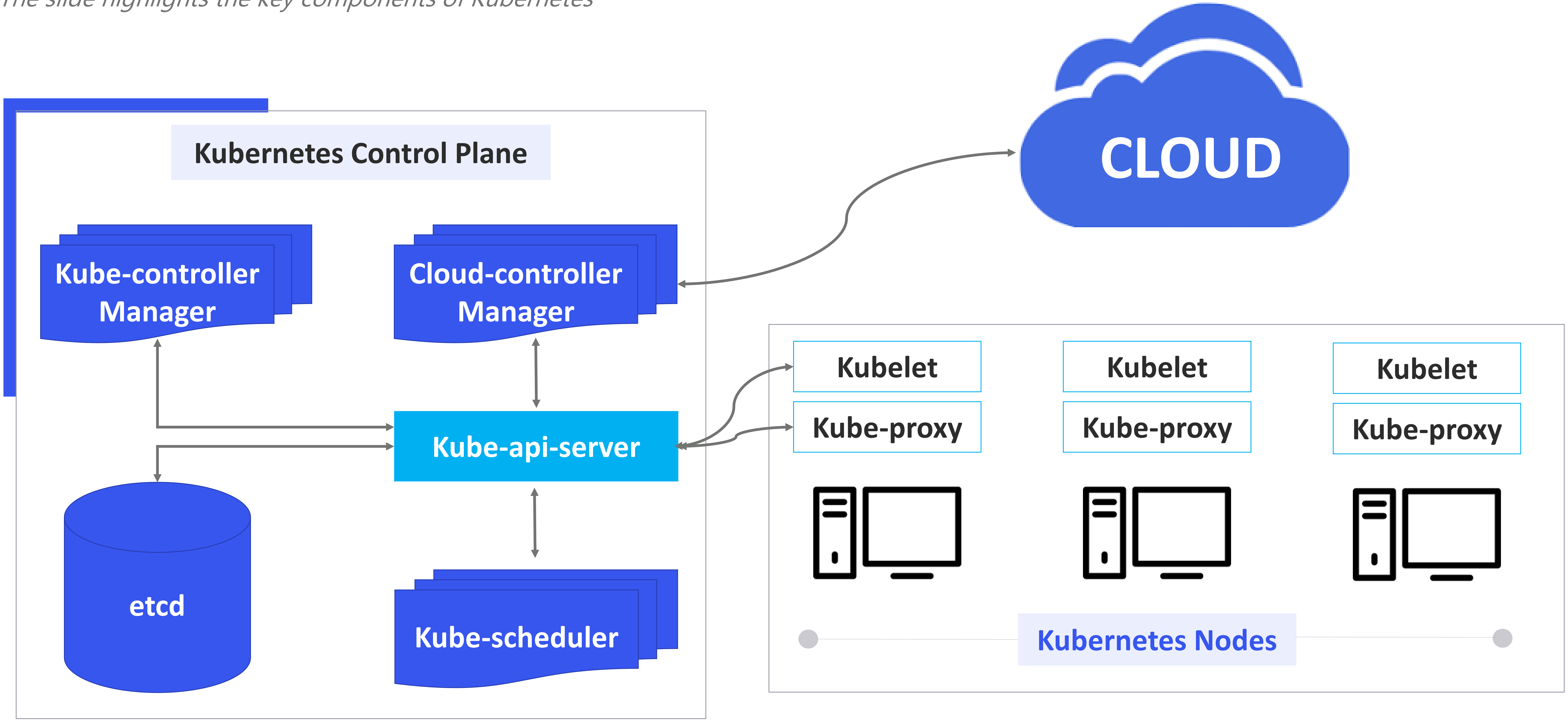
2

Kubernetes Component

- What is Kubelet?
- What is Kubectl?
- What is Kubeadm?

Kubernetes Components

The slide highlights the key components of Kubernetes



Kubernetes Control Plane

Kube-controller Manager

Cloud-controller Manager

Kube-api-server

etcd

Kube-scheduler

CLOUD

Kubelet

Kube-proxy

Kubelet

Kube-proxy

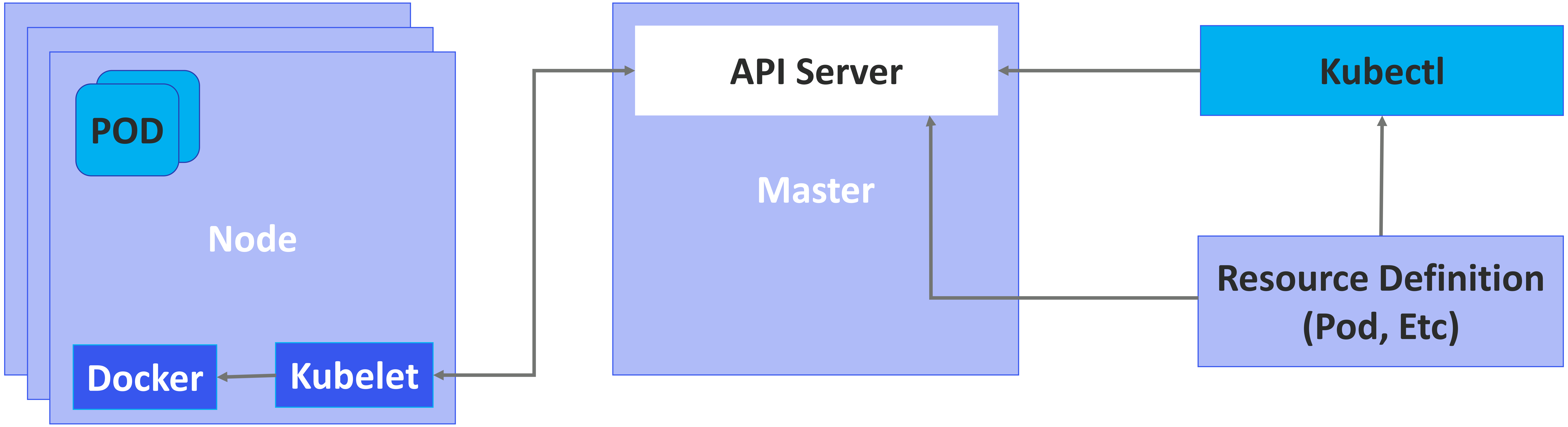
Kubelet

Kube-proxy

Kubernetes Nodes

What is Kubelet?

The slide shows the overview of Kubelet and explains its functionality

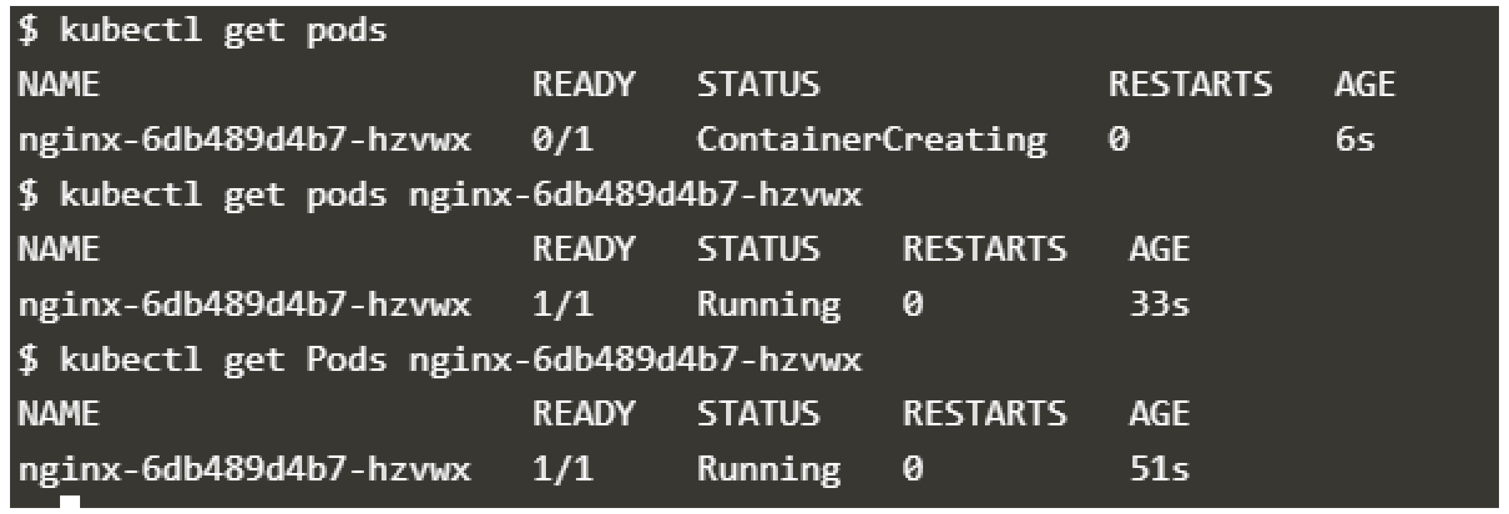


The kubelet is the primary "node agent" that runs on each node.

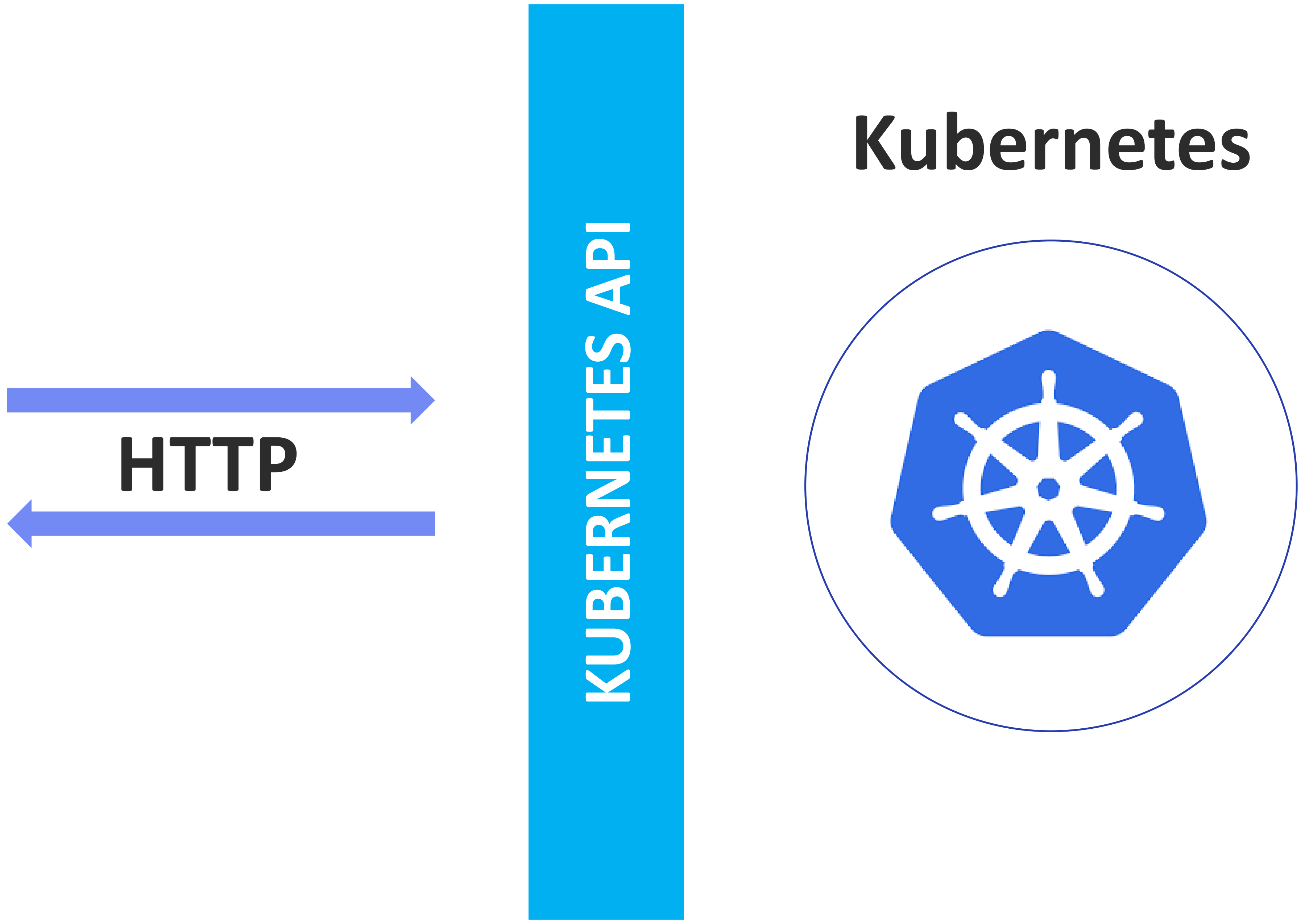
The kubelet works in terms of a PodSpec. A PodSpec is a YAML or JSON object that describes a pod. The kubelet takes a set of PodSpecs that are provided through various mechanisms (primarily through the apiserver) and ensures that the containers described in those PodSpecs are running and healthy. The kubelet doesn't manage containers which were not created by Kubernetes.

What is Kubectl?

KUBECTL



```
$ kubectl get pods
NAME                   READY   STATUS              RESTARTS   AGE
nginx-6db489d4b7-hzvwx   0/1     ContainerCreating   0          6s
$ kubectl get pods nginx-6db489d4b7-hzvwx
NAME                   READY   STATUS    RESTARTS   AGE
nginx-6db489d4b7-hzvwx   1/1     Running   0          33s
$ kubectl get Pods nginx-6db489d4b7-hzvwx
NAME                   READY   STATUS    RESTARTS   AGE
nginx-6db489d4b7-hzvwx   1/1     Running   0          51s
```

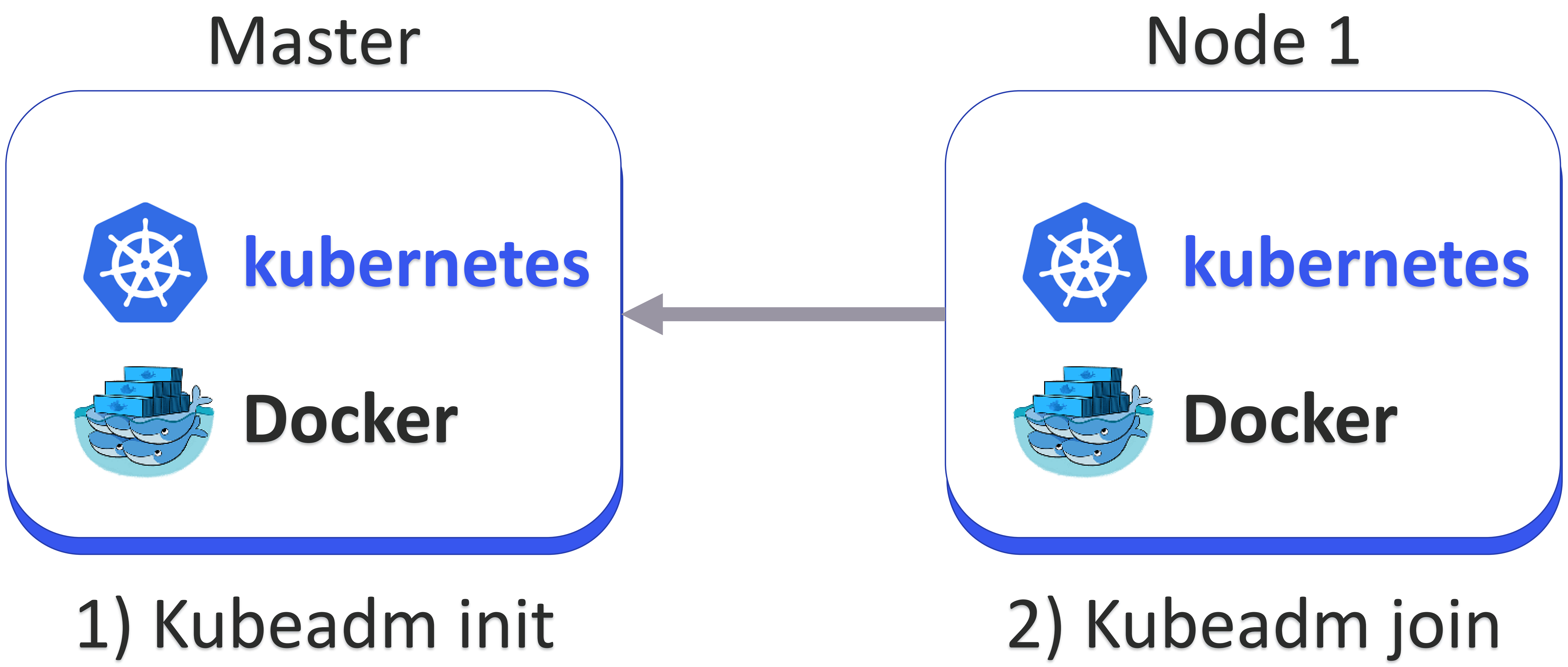


The kubectl command line tool lets you control Kubernetes clusters.

For configuration, kubectl looks for a file named config in the $HOME/.kube directory.

For details about each command, including all the supported flags and subcommands, see the kubectl reference documentation. For installation instructions see installing kubectl.

What is Kubeadm?



Using kubeadm, you can create a minimum viable Kubernetes cluster that conforms to best practices.

In fact, you can use kubeadm to set up a cluster that will pass the Kubernetes Conformance tests. kubeadm also supports other cluster lifecycle functions, such as bootstrap tokens and cluster upgrades.

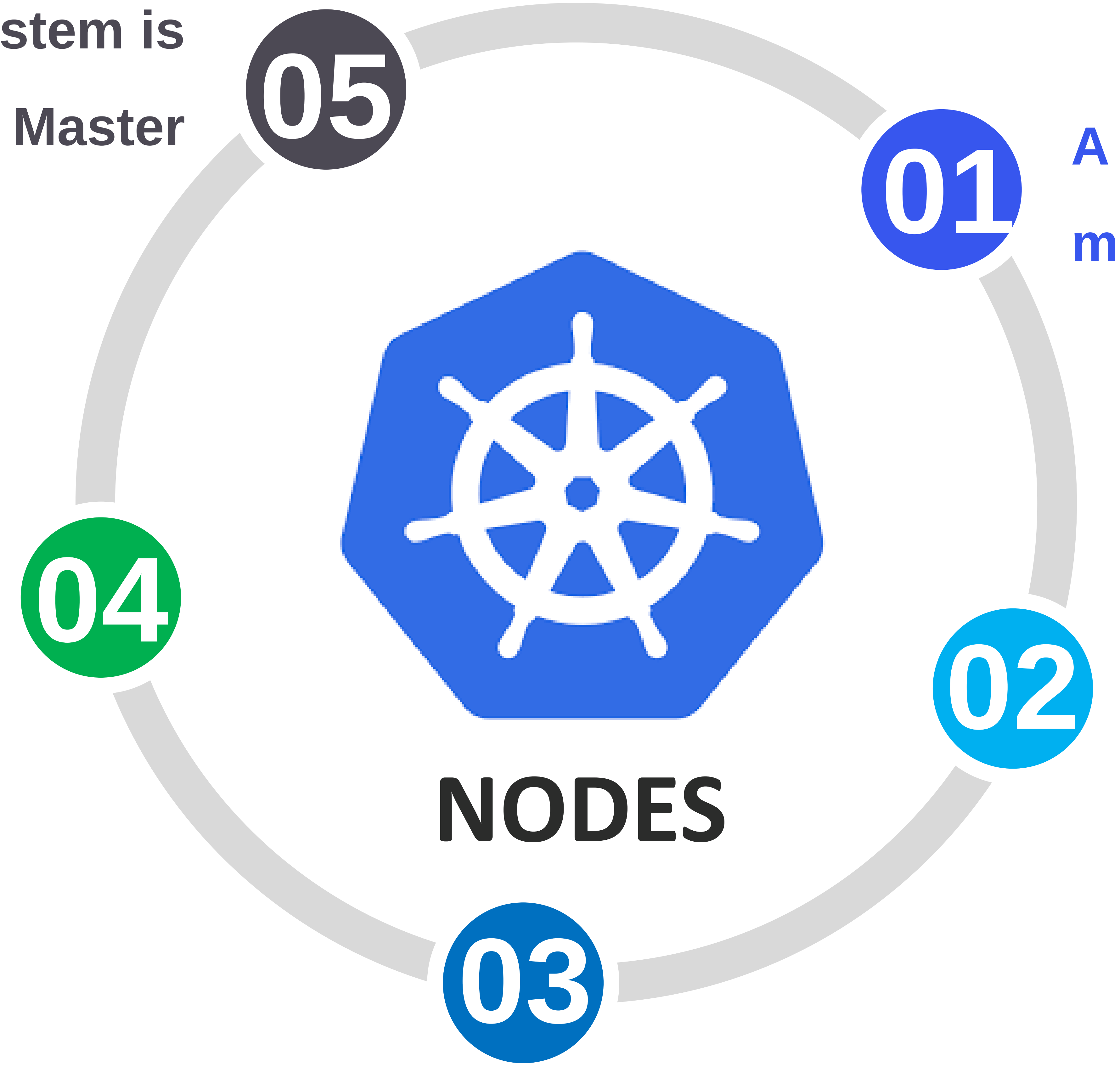
3

Nodes in Kubernetes

- What is a node in Kubernetes?
- Master node
- Worker node

What is a Node in Kubernetes?

The slide provides the overview of nodes in Kubernetes



01 A Node in Kubernetes Cluster is the main Worker Machine

02 They are also known as Minions

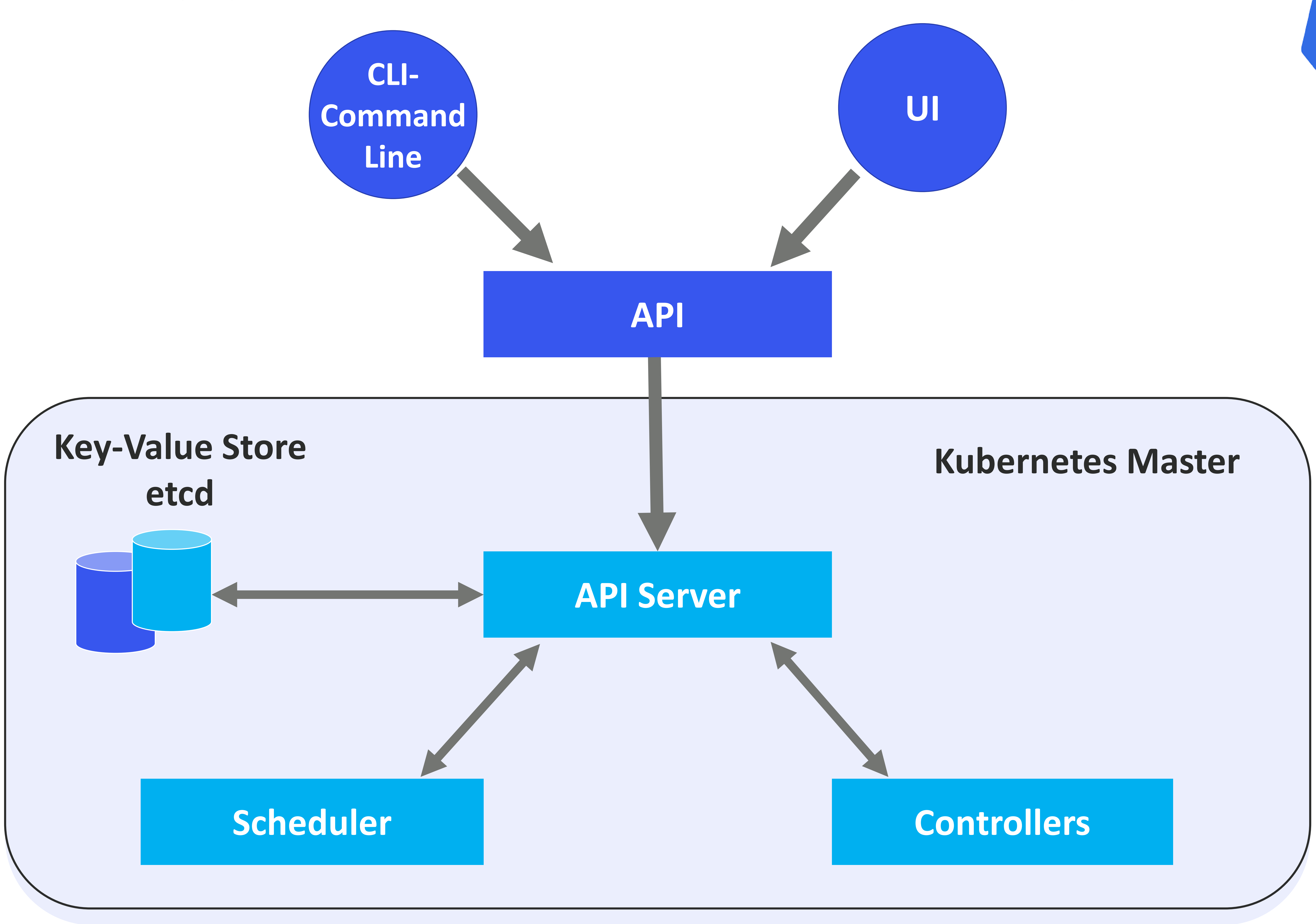
03 It Cloud Run on a Physical Machine or a VM

04 Node Provide All The Necessary Services to Run Pods

05 Node in Kubernetes System is Managed by the Master

Master Node in Kubernetes

The slide highlights overview of Master Node in Kubernetes and its functionality

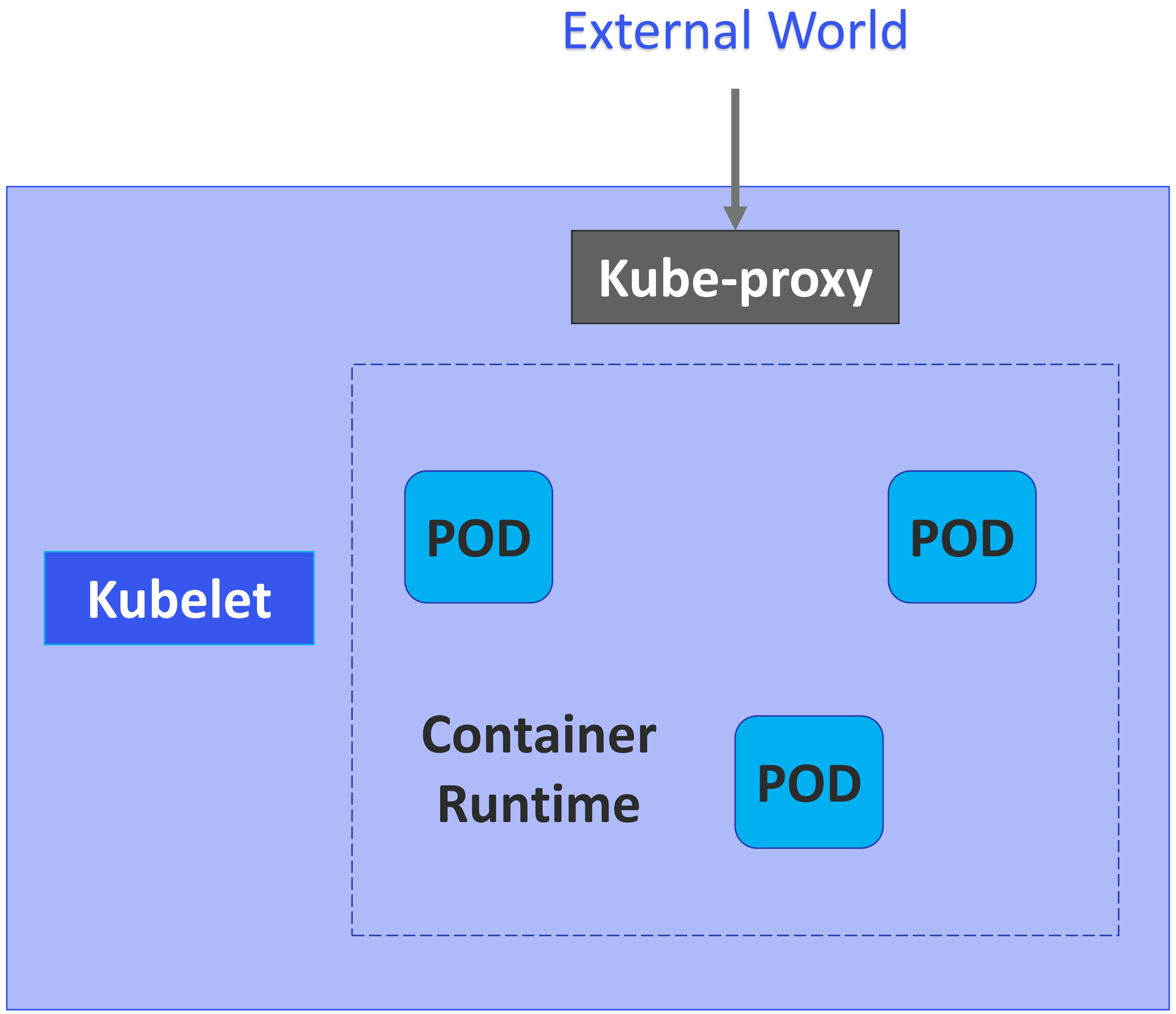


A Master Node

is a node which controls and manages a group of worker nodes (workloads runtime) and resembles a cluster in Kubernetes. All external communication to the cluster is via the API-Server, Kube-Controller-Manager, that runs a group of controllers for the running cluster.

Worker/Slave Node in Kubernetes

The slide highlights overview of Worker Node in Kubernetes and its functionality



A Worker Node

- ✓ It is a physical server otherwise you will say a VM that runs the applications victimization Pods (a pod programming unit) that is controlled by the Master Node.
- ✓ On a physical server (Worker/Slave Node), Pods area unit scheduled.
- ✓ For accessing the applications from the external world, we have a tendency to connect with nodes.

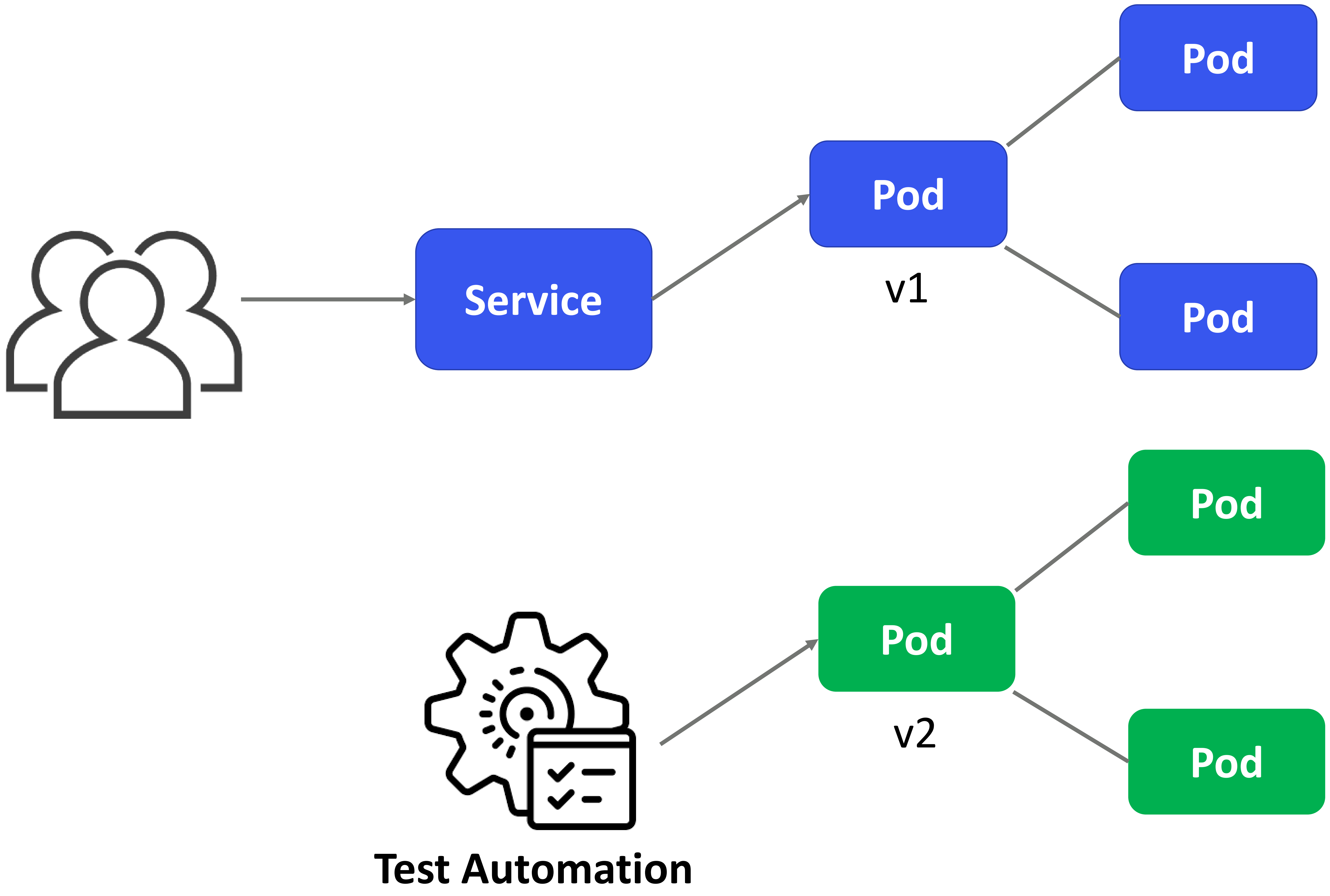
4

Kubernetes Development Process

- What is blue green deployment?
- How to automate the deployment?

What is Blue Green Deployment?

The slide explains what is blue green deployment and its functionality

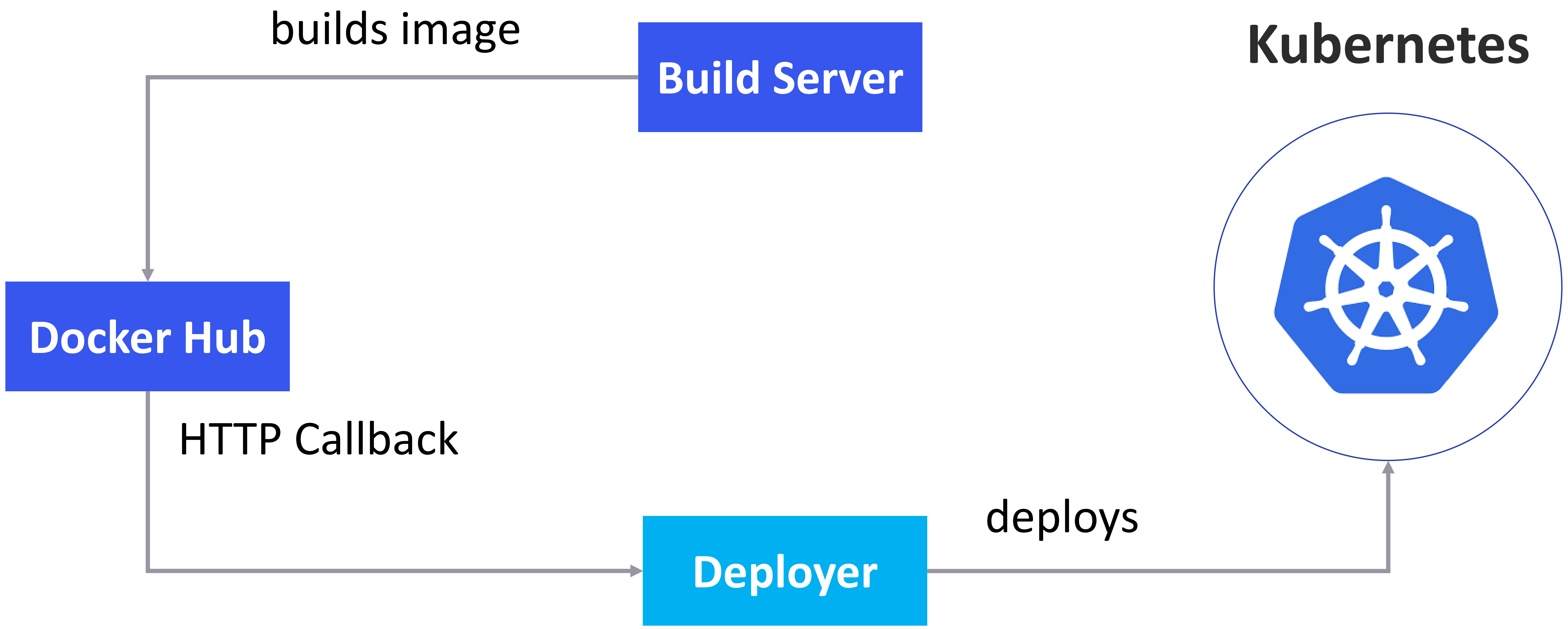


Blue Green Deployment

- ✓ Blue green deployment is a method that reduces period and risk by running 2 identical production environments called Blue and Green.
- ✓ At any time, just one of the environments is live, with the live atmosphere serving all production traffic.
- ✓ For this instance, Blue is presently live and Green is idle.

How to automate the deployment?

The slide highlights how to automate the deployment process



builds image

Build Server

Kubernetes

Docker Hub

HTTP Callback

deploys

Deployer

Container pipelines **automate each of the stages in the container deployment process, from building the initial image to deploying to production. Typically, the entire container pipeline consists of three stages:**

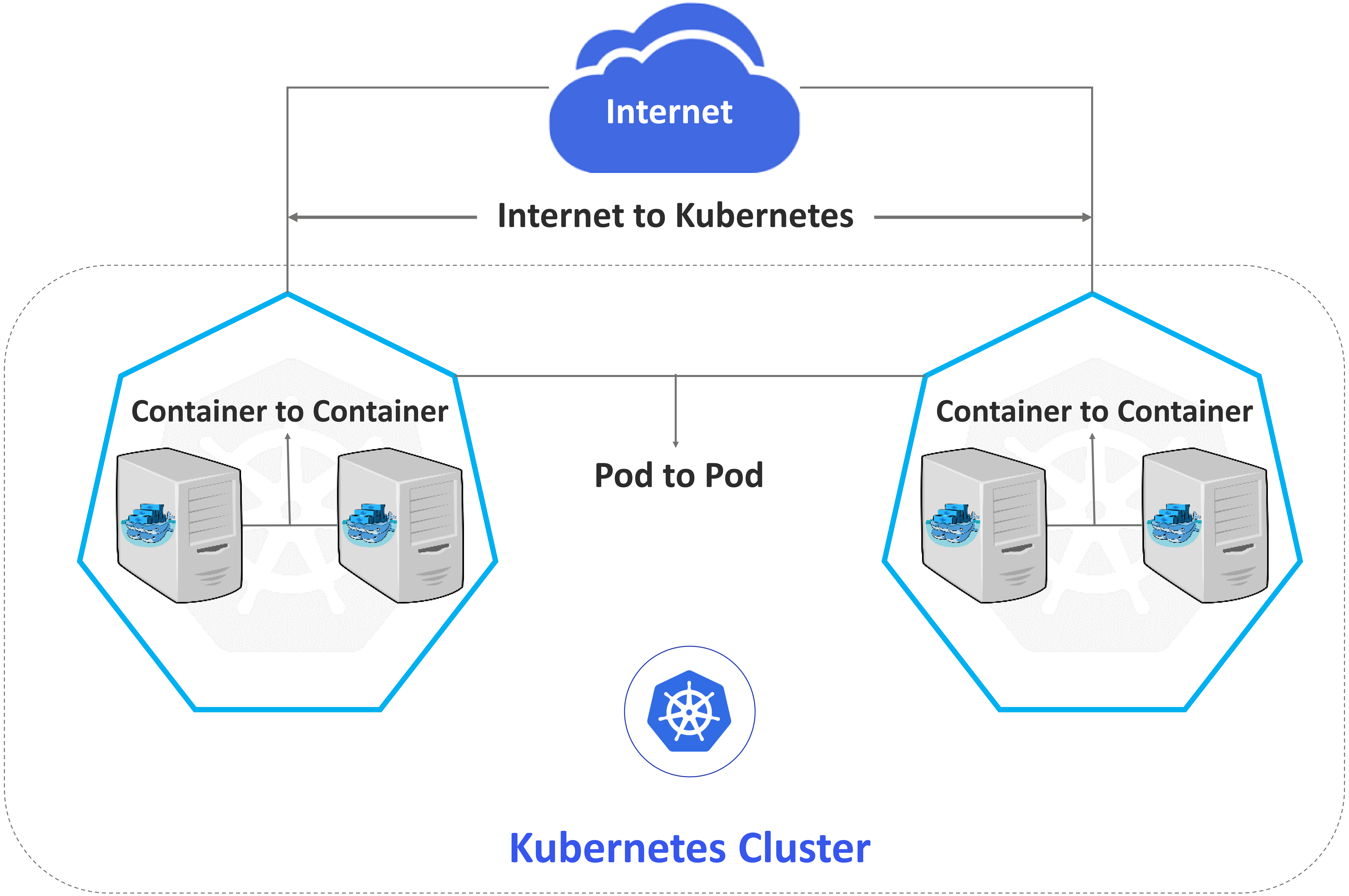
- ✓ **Integration: changes are checked into source control, triggering the build process and unit tests.**
- ✓ **Acceptance testing: the container is deployed to a test environment and verified for functionality.**
- ✓ **Deployment: the final, fully-tested image is deployed to production.**

5

Networking Kubernetes

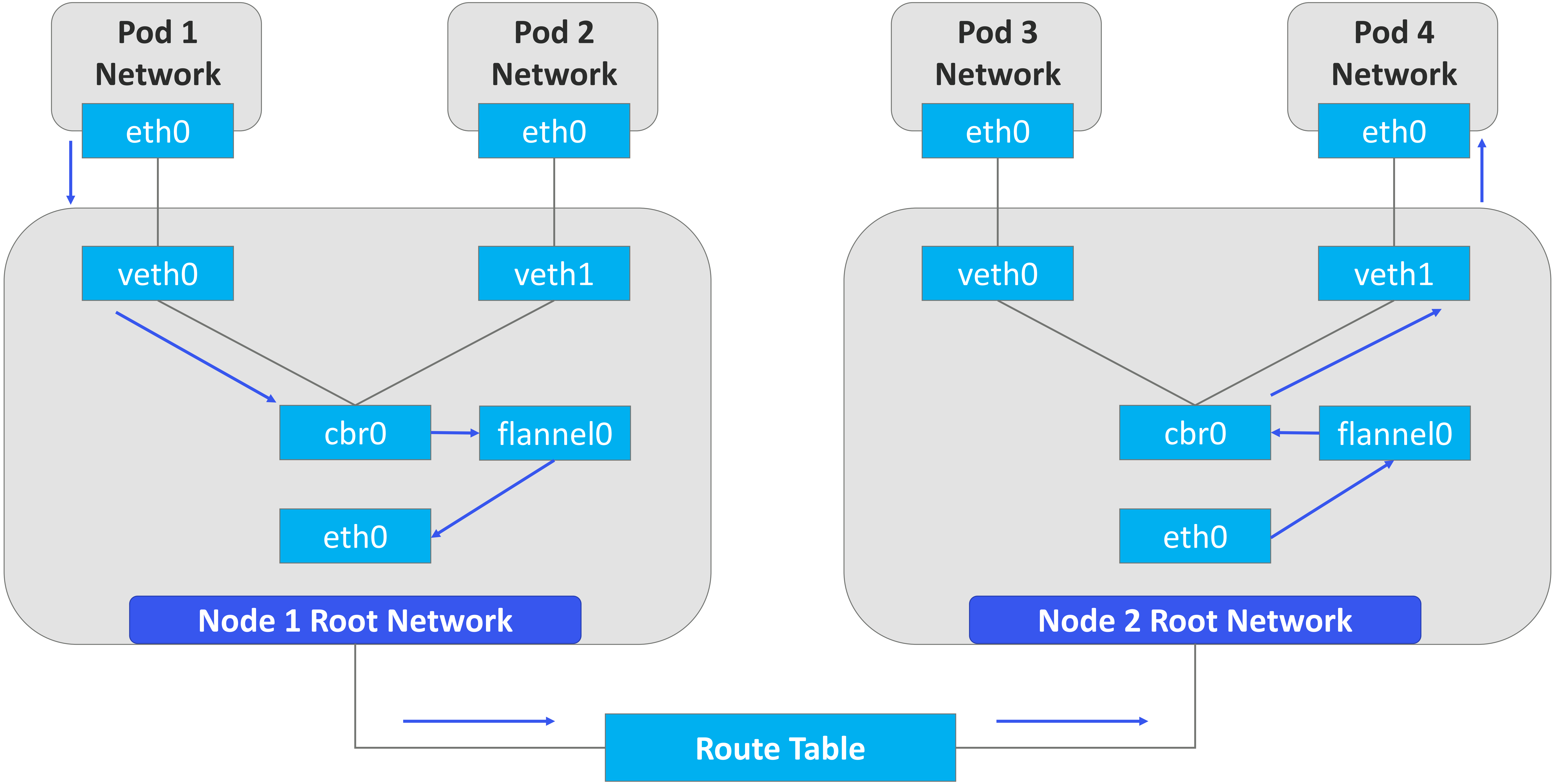
- Kubernetes networking model
- Ingress networking in Kubernetes

Kubernetes Networking Model



Ingress Networking in Kubernetes

The slide explains the Ingress Networking in Kubernetes and its working



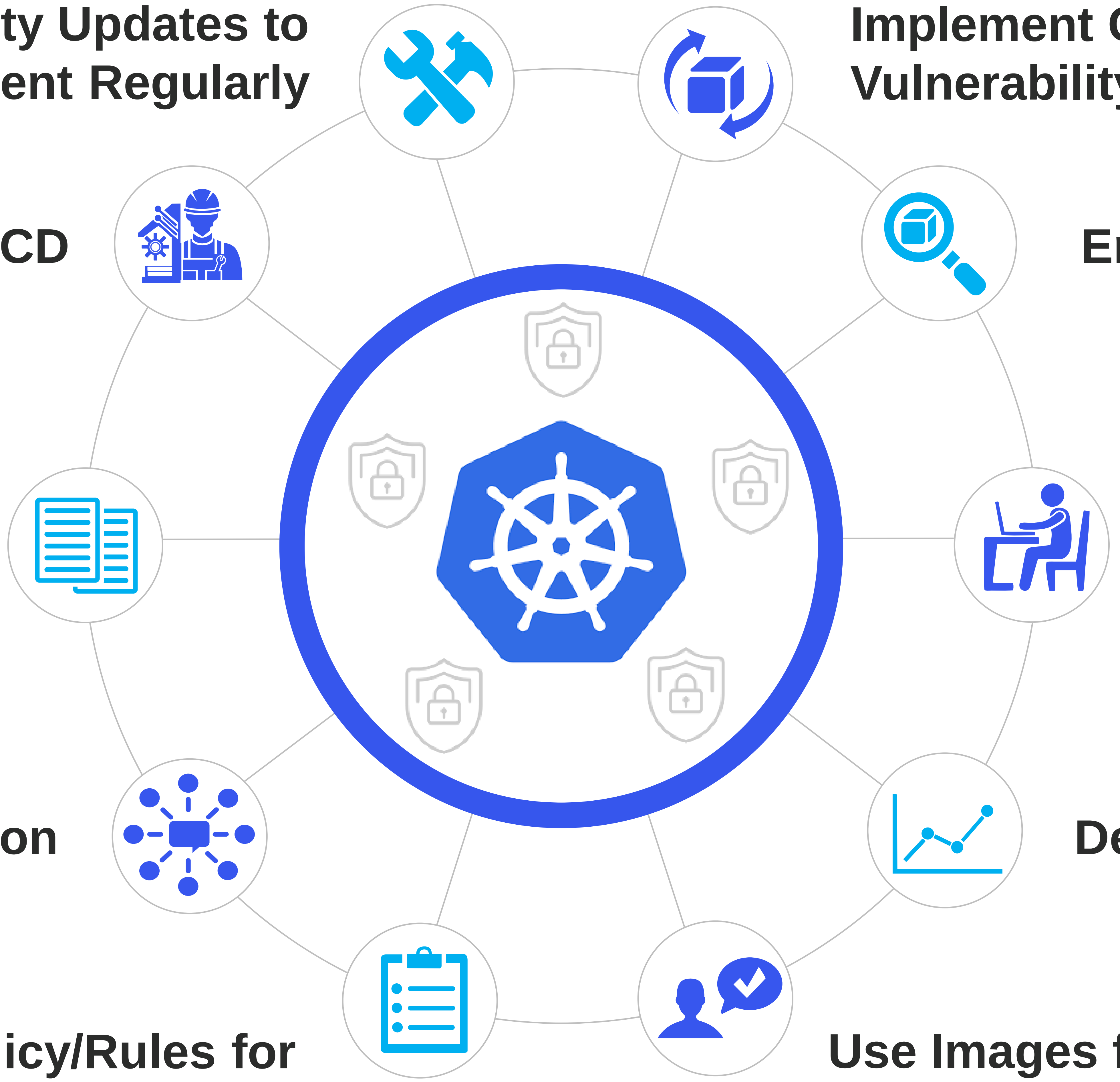
6

Security Measures in Kubernetes

- Best security measures in Kubernetes

Best Security Measures in Kubernetes

The slide consists of the best Security measures which needs to follow in Kubernetes



Apply Security Updates to Environment Regularly

Restrict Access to ETCD

Log everything on the Production Environment

Implement Network Segmentation

Decides Strict Policy/Rules for Resources

Implement Continuous Security Vulnerability Scanning

Enables Auditing

Providing limited direct access to Kubernetes Nodes

Defines Resources Quota

Use Images from Authorized Repository only

References

01 Learn about Kubernetes and its fundamental concepts
https://kubernetes.io/docs/home/

02 Introduction To Kubernetes | Simplilearn
https://www.youtube.com/watch?v=ISUmszlMQPU

03 Introduction to Kubernetes | Microsoft
https://docs.microsoft.com/en-us/learn/modules/intro-to-kubernetes/

04 Introduction to Kubernetes architecture
https://www.redhat.com/en/topics/containers/kubernetes-architecture

05 Understanding Kubernetes Architecture With Diagrams
https://phoenixnap.com/kb/understanding-kubernetes-architecture-diagrams

06 Kubernetes Tutorial - A Comprehensive Guide for Kubernete
https://www.edureka.co/blog/kubernetes-tutorial/

07 Kubernetes in Action: What It Does (and Doesn't) Do
https://www.itprotoday.com/containers/kubernetes-action-what-it-does-and-doesnt-do

08 Comparing Container Pipelines
https://dzone.com/articles/comparing-container-pipelines

09 Kubernetes slide templates | SlideTeam
https://www.slideteam.net/kubernetes-docker-container-implementation-ppt-powerpoint-presentation-slide-templates.html

Thanks

TRANG NGUYEN PRESENTATION