edureka!

Microsoft Azure DevOps Solution Certification (AZ-400)

COURSE OUTLINE

Azure AZ-400

edureka!

Build Containers with Azure DevOps

Topics

Following are the topics covered in this module:

- Introduction to Container
- Introduction to Orchestration
- Introduction to Kubernetes
- Azure Kubernetes Service (AKS)

Objectives

After completing this module, you should be able to:

- Implement a container build strategy
- Build & run container with Docker
- Manage container lifecycle with Container Orchestration
- Manage Kubernetes Cluster
- Monitor AKS Cluster from Azure Monitor
- Monitor AKS Cluster from AKS
- Deploy a multi-container application to AKS
- Modernize an ASP.NET application to Azure

Flipkart’s Huge Sale

Flipkart’s Upcoming Plan

- Flipkart is an e-commerce giant. It is planning for the Big Billion Days Sale in the next month
- Usually, during every sale, most of the customers' login and shop at the same time. Due to which there is heavy traffic to the website

The Issue

Due to the heavy load/traffic on the website, there will be a probability of scaling the instance rapidly as and when required. To address this issue, the company has hired Mr. Jeff as a Senior DevOps Engineer

Mr. Jeff
Senior DevOps Engineer

The Research

To address this issue, I have decided to use Azure and Docker to do the following:

- Deploy a multi-container application to AKS
- Modernize the ASP.NET application to Azure
- And then perform Docker-based web app development for scaling the instances whenever required

The Solution

Using Azure Kubernetes Service (AKS)

- AKS makes it quick and easy to deploy and manage containerized applications
- With AKS, one can create resources and infrastructure inside Azure Kubernetes Cluster through Deployments and Services manifest files instead of creating resources in the cloud

Modernizing ASP.NET Application to Azure

- Choosing to move an ASP.NET Application to the cloud increases the agility and speed of their applications
- Thousands of servers (VMs) can be set up in the cloud in minutes, compared to the weeks it typically takes to set up on-premises servers

Introduction to Container

edureka!

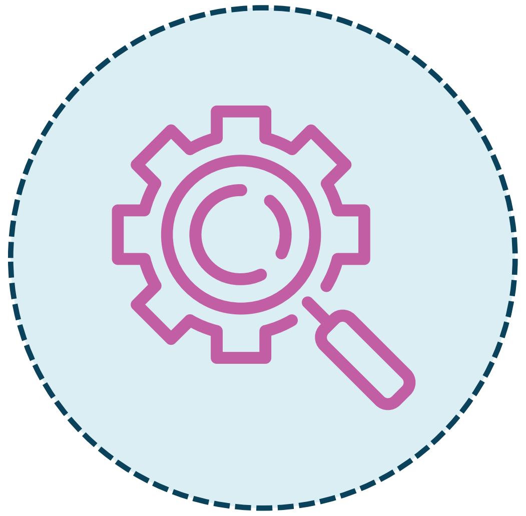
Why Containers?

Containers are required for application portability.

The portability issue can happen due to many reasons.

Missing library

Missing prerequisite software



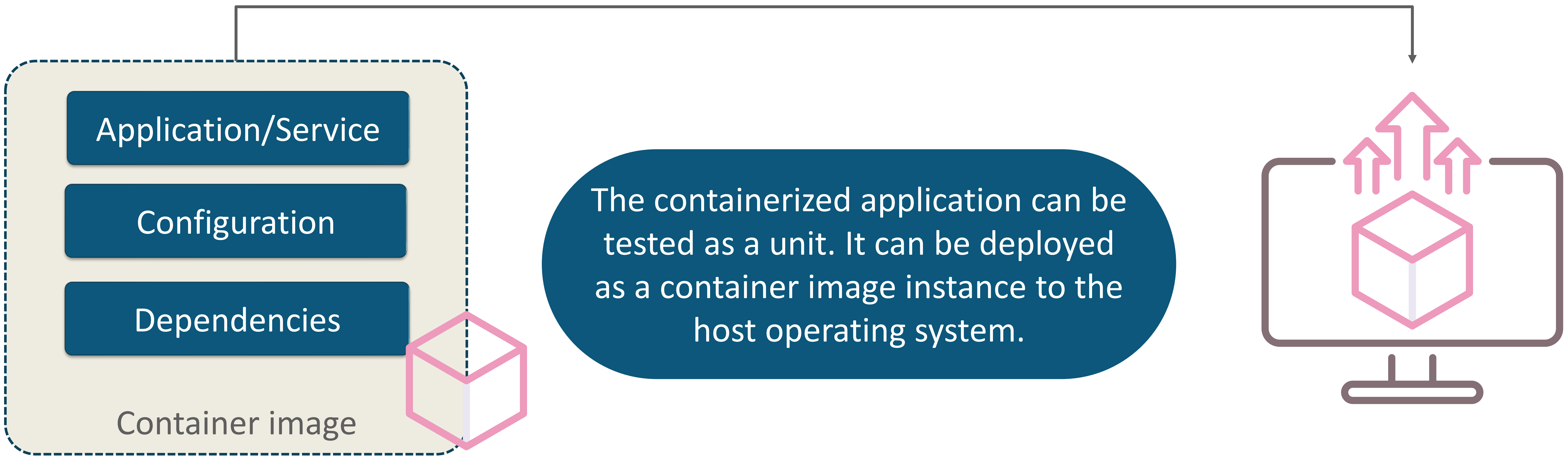
Missing configurations

A lot of time is spent on fixing these issues

edureka!

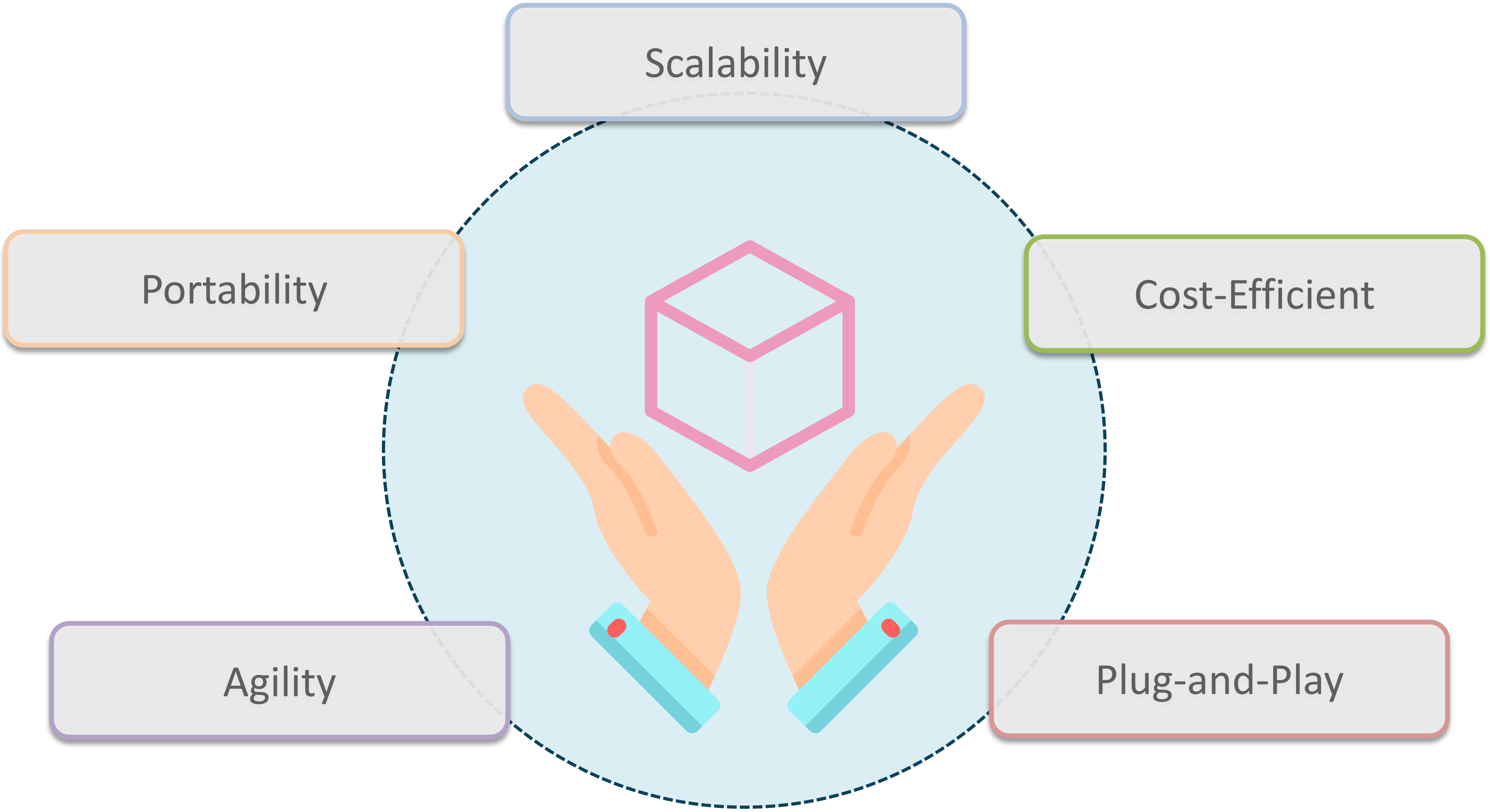
What are Containers?

Containers address compatibility issues by providing lightweight, immutable infrastructure for application packaging and deployment.



The containerized application can be tested as a unit. It can be deployed as a container image instance to the host operating system.

Advantages of Containers



Container Registry

Container Registry

- Allows to store, build and manage container images
- Helps with the fast, scalable deployment of container deployment

edureka!

Azure Container Registry

Container Registry

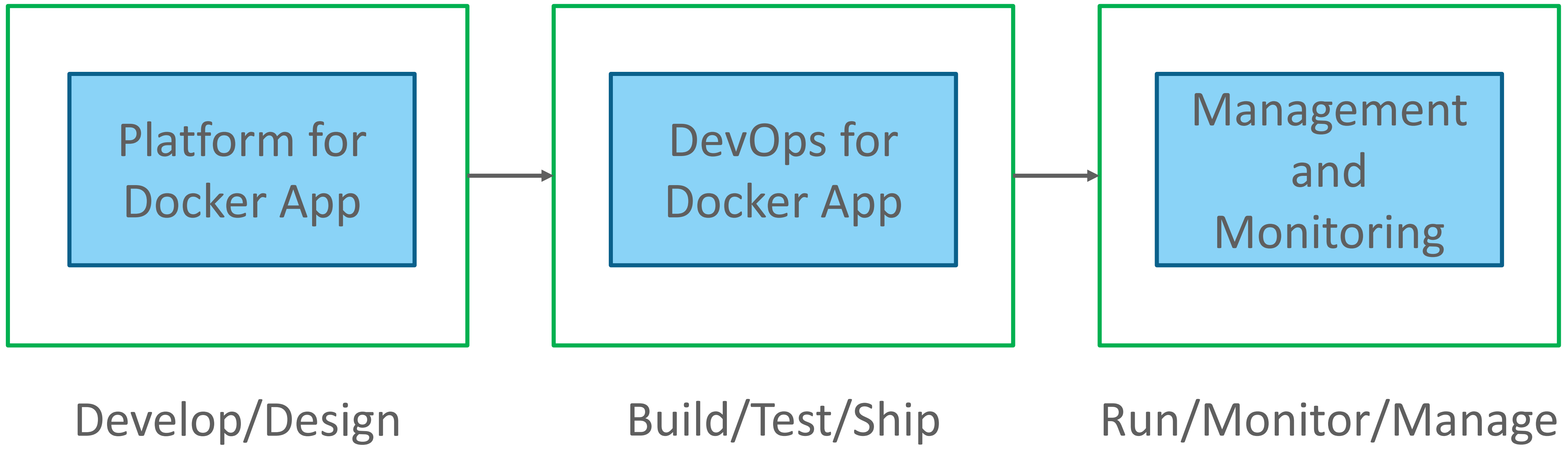
For deployment, Azure provides a service name and Container Registry to register the container images.

The container virtualizes the operating system within it.

- Any application deployed in a container perceives that it has all the required infrastructure like **OS, RAM, storage, and network connection**.
- Containers can be deployed to any OS like Windows, Linux, etc. Due to the convenience of its portability and other benefits, containers are used a lot in application development these days.

edureka!

Docker Containers



Docker Containers

In Azure, Docker containers are used. Docker containers can be used directly from Azure Container Registry, Docker Hub, or Private Registry.

Applications are hosted in container along with all its dependencies.

Azure Container Registry

Azure Container Instance

edureka!

Containers Working

Images can be stored in Azure Container Registry. Registry allows easy access to these images for container creation.

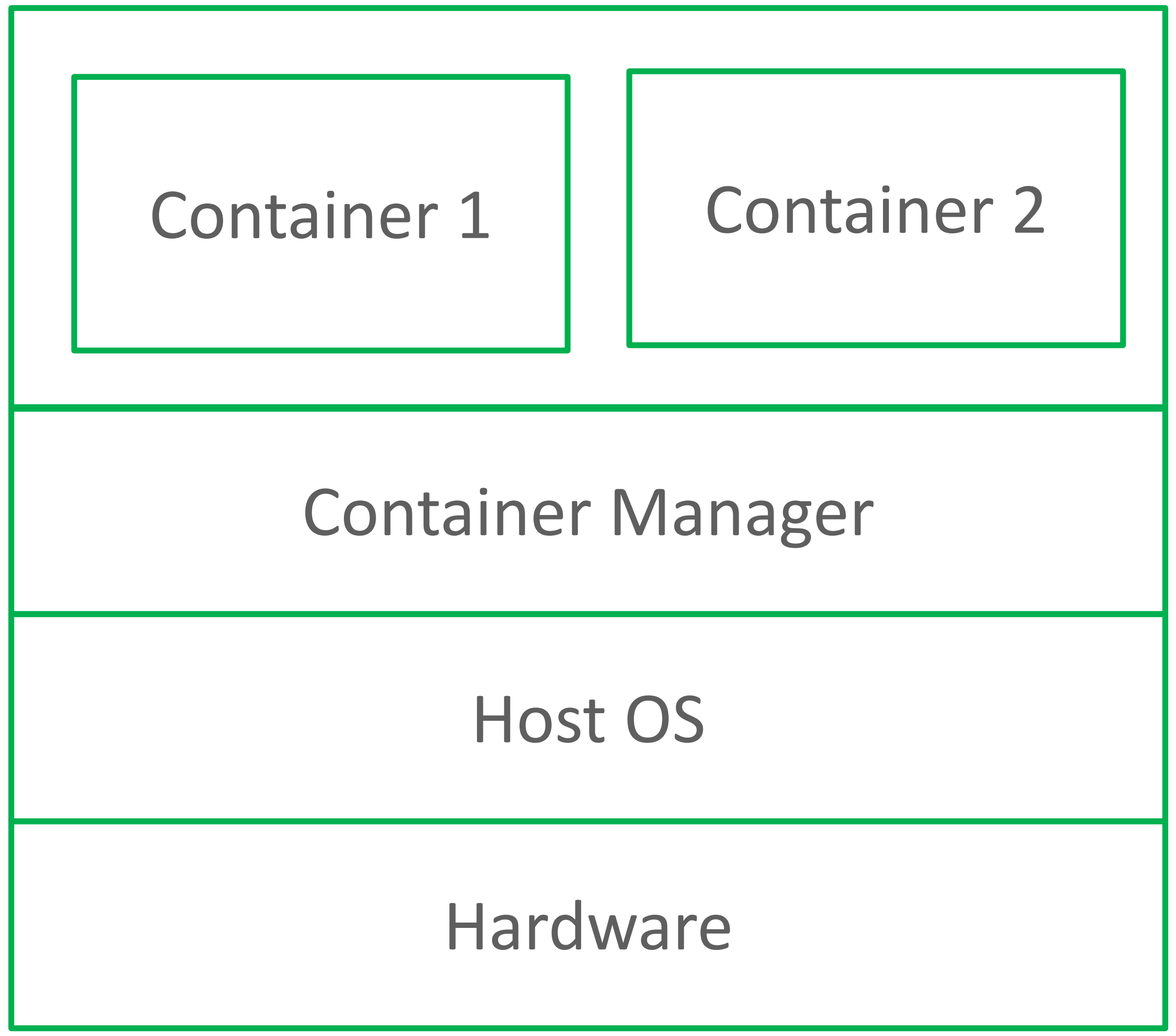
This is helpful for distributed application or microservice-based application development.

Image is immutable

The advantage of this approach is that the same image is used for production and testing. It makes sure that the testing is done on the application, which is the same as production.

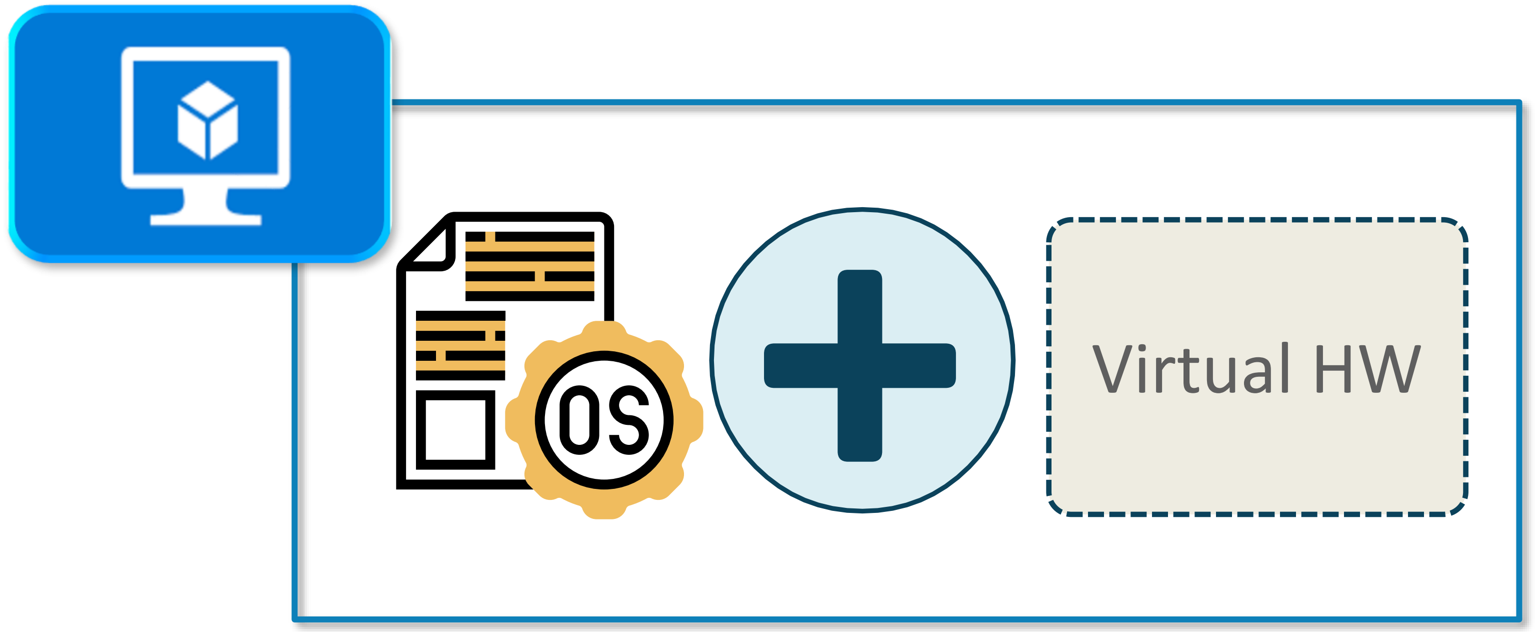
Container Architecture

- A container does not create a virtualized environment like VM, rather it uses the host OS for its computing infrastructure
- Containers sits on top of the host OS or physical server and operate from here
- Container utilizes the host OS for supporting the program, libraries, and system resources to run the program contained in it



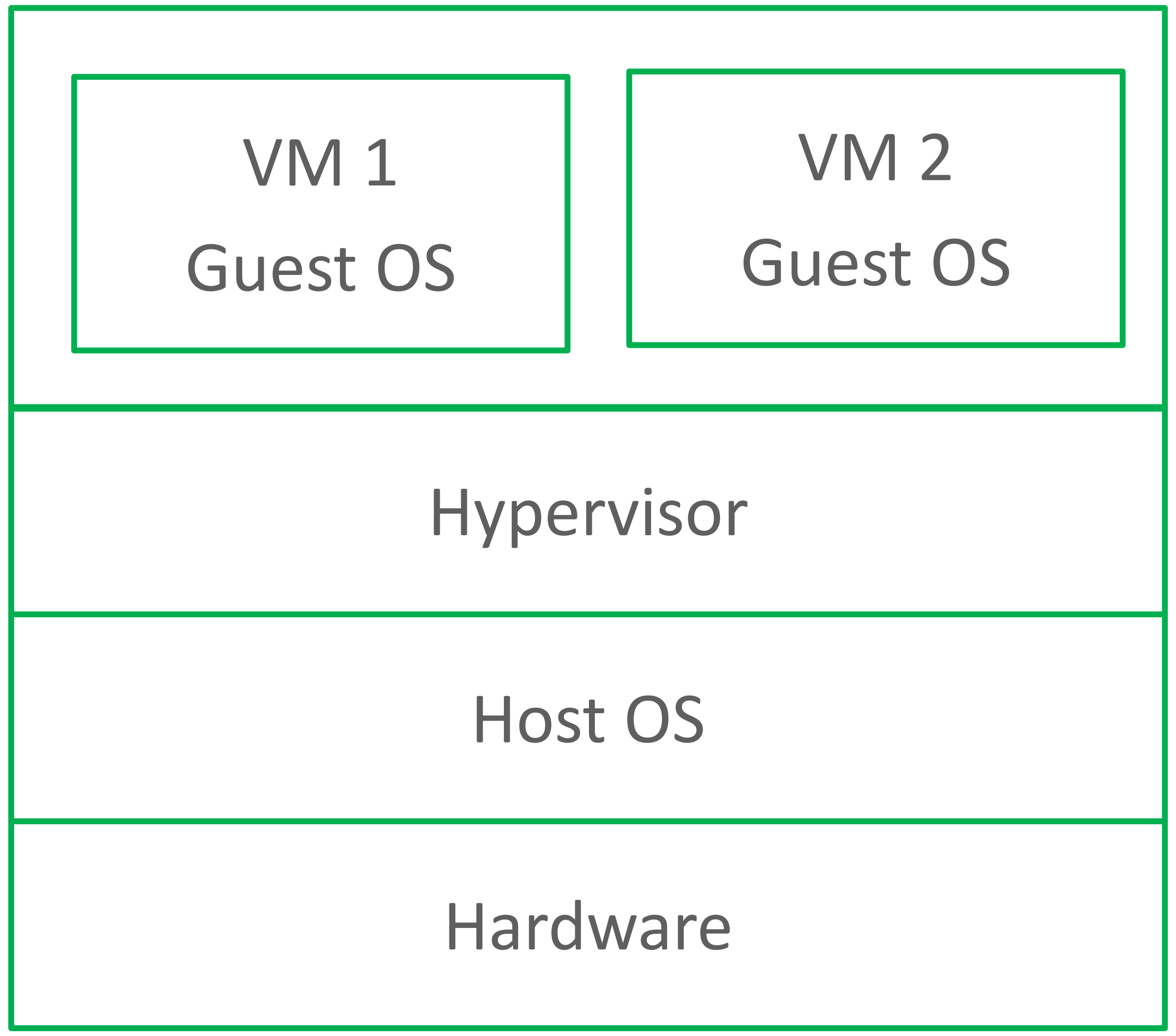
Container Architecture

Virtual Machine



Virtual Machine

- The virtual machine utilizes the host machines hardware
- The server provides the hardware
- Between the VM and hardware, there is a layer known as Hypervisor



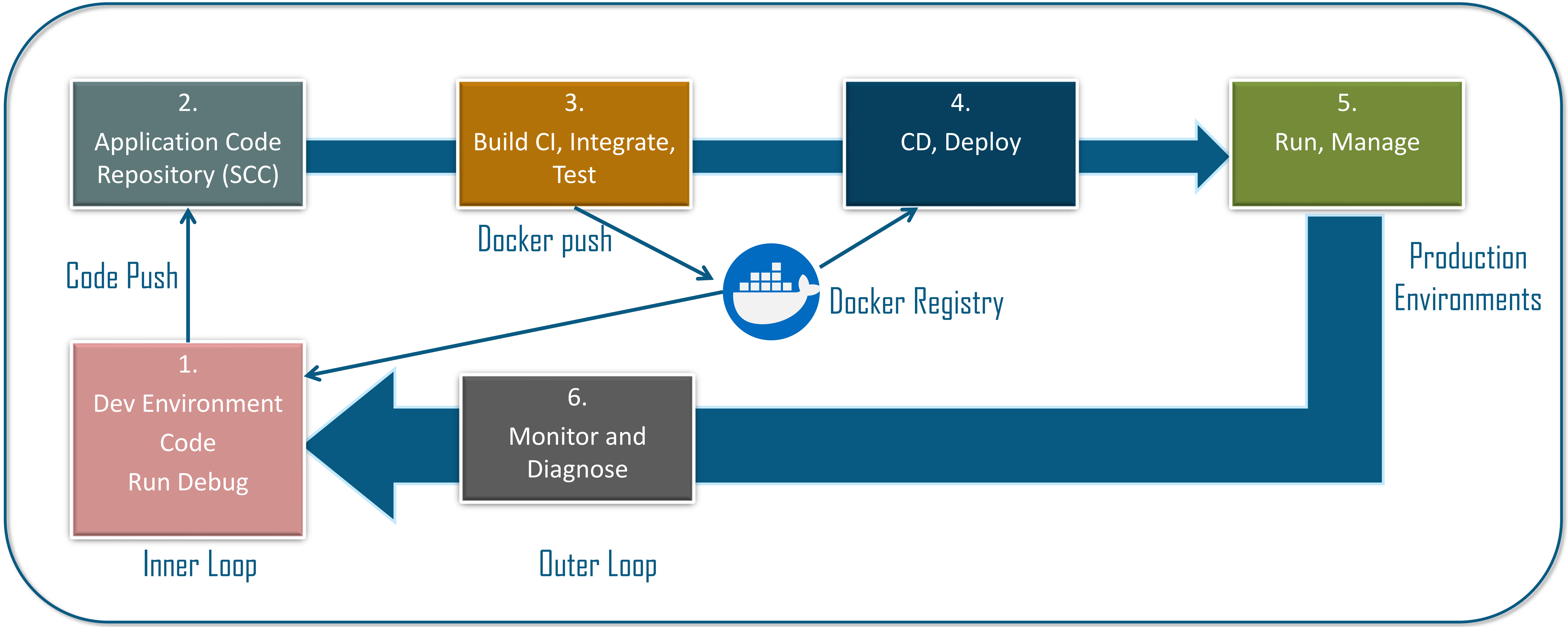
Architecture of Virtual Machine

Containers vs. Virtual Machines

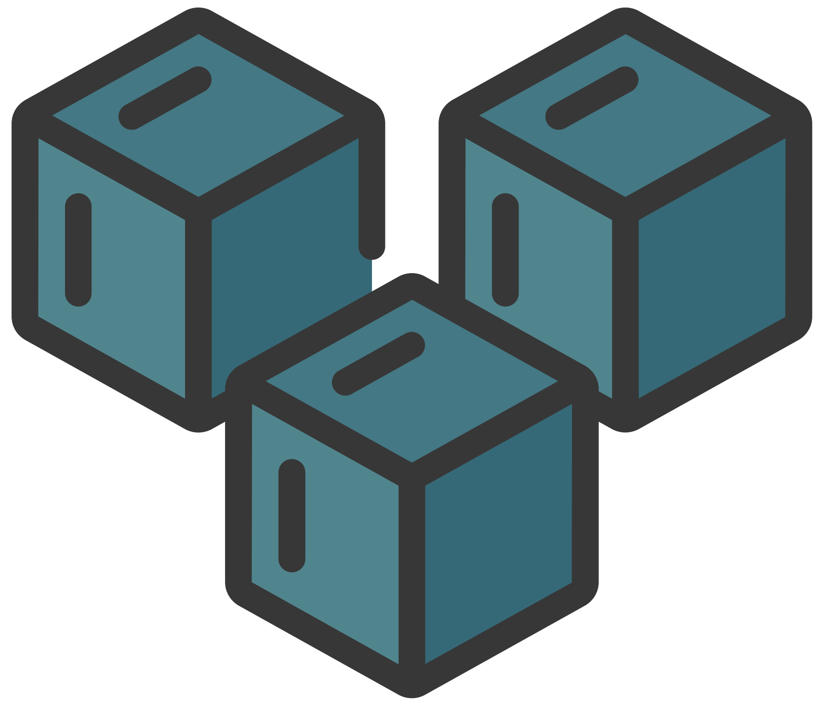
	Containers	Virtual Machines
Number of Apps on Server	Can run a limited number of applications, depending on VM Configuration	Can run multiple instances of different applications together
Resource Overhead	Light-weight with little to no overhead	Highly resource-intensive with performance overhead of managing multiple OSs
Speed of Deployment	Very fast deployment. Container images can deploy new instances in seconds	Very slow deployment. Requires setting up OS, app dependencies, etc.
Security	Users usually require root level permissions to execute container related tasks	Provides better security
Portability	Highly portable with emphasis on consistency across environments	Very limited portability

edureka!

Implementing a Container Build Strategy



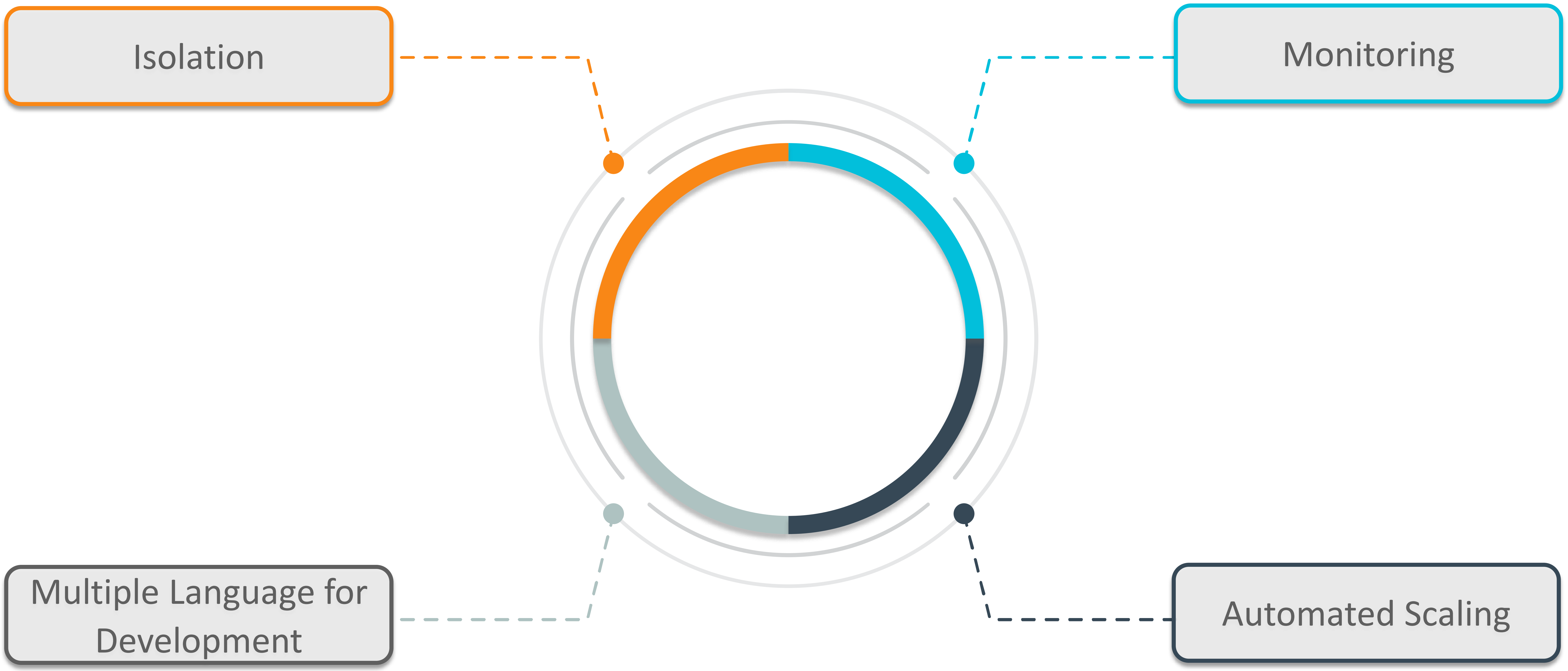
Container Usage in Microservices



Microservices is an architectural style that structures an application as a collection of loosely coupled and independent services.

- Mostly used for large application development
- The large-scale application is divided into smaller microservices
- The communication among the microservices follow SOA(Service oriented Architecture) principle
- Microservices can be developed in any language
- Containers are used to deploy applications with microservice architecture

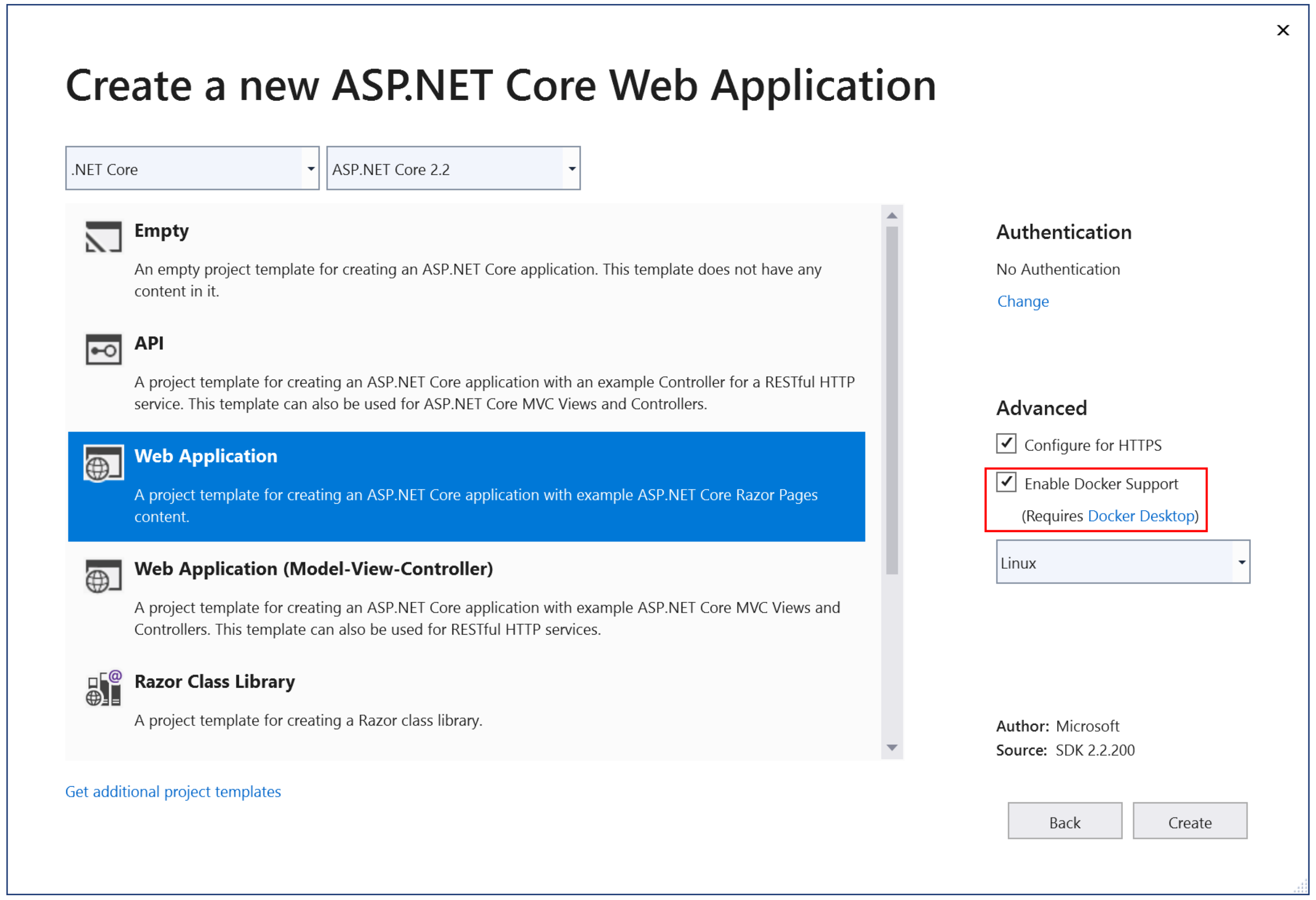
Container Usage in Microservices



Visual Studio Support for Docker Containers

Any application can be created as a containerized application using Docker.

- The Visual Studio 2017 and up comes with Docker support
- Create an ASP.NET Core Web application
- Check 'Enable Docker Support'


Create a new ASP.NET Core Web Application

.NET Core | ASP.NET Core 2.2

Empty
An empty project template for creating an ASP.NET Core application. This template does not have any content in it.

API
A project template for creating an ASP.NET Core application with an example Controller for a RESTful HTTP service. This template can also be used for ASP.NET Core MVC Views and Controllers.

Web Application
A project template for creating an ASP.NET Core application with example ASP.NET Core Razor Pages content.

Web Application (Model-View-Controller)
A project template for creating an ASP.NET Core application with example ASP.NET Core MVC Views and Controllers. This template can also be used for RESTful HTTP services.

Razor Class Library
A project template for creating a Razor class library.

Get additional project templates

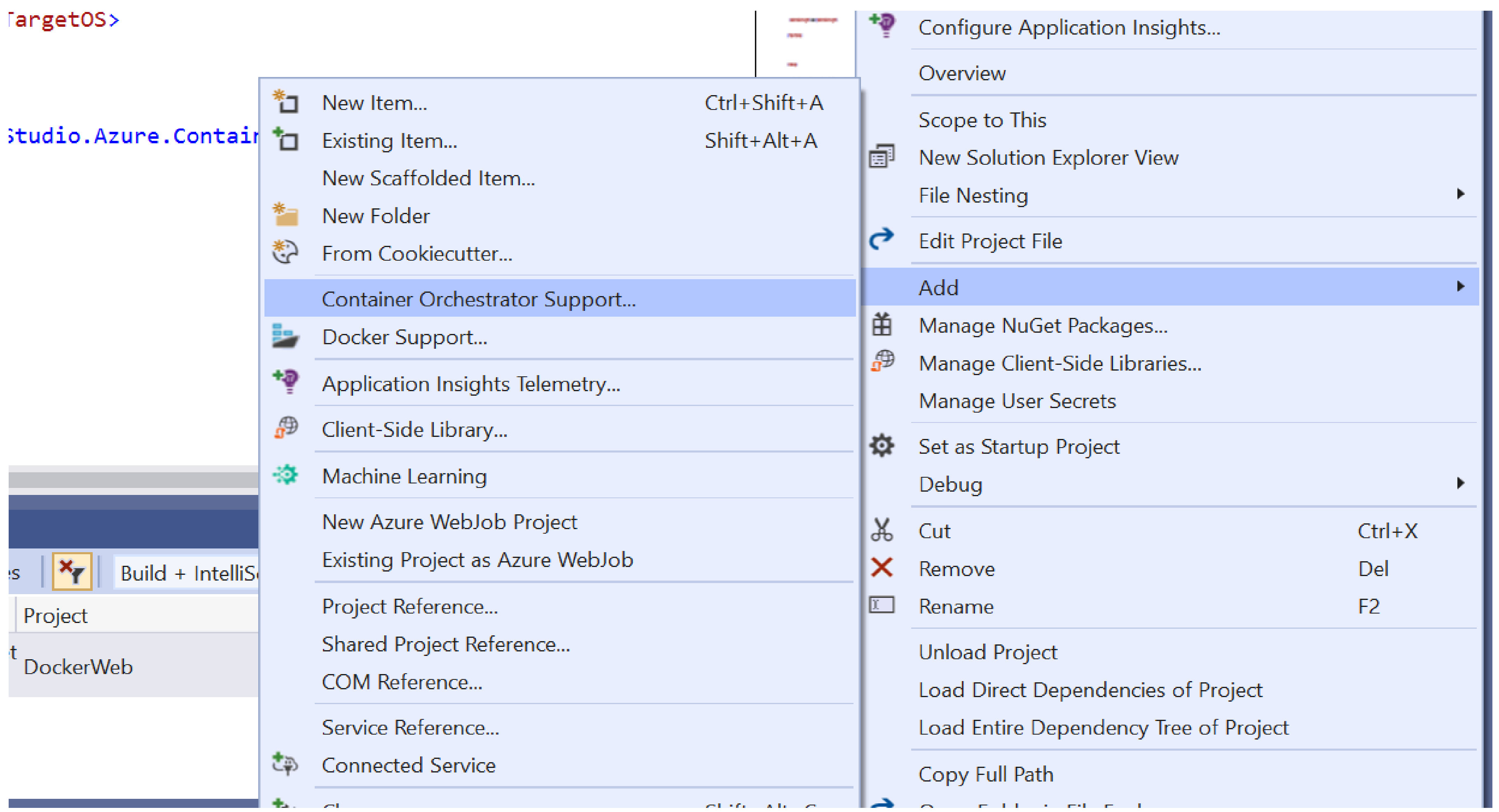
Authentication
No Authentication
Change

Advanced
✓ Configure for HTTPS
✓ Enable Docker Support
(Requires Docker Desktop)
Linux

Author: Microsoft
Source: SDK 2.2.200

Manage Container with Docker (Cont.)

- Install Docker
- The solution will be created using Docker
- To enable Orchestration, right-click on the project, and do as per the given snapshot



This will add the required file for Orchestration, which can be used for deploying to Azure Kubernetes service.

edureka!

Introduction to Orchestration

edureka!

What is Orchestration?

Orchestration is the end-to-end management of containers.

Service discovery

Health check

Load balancing

Scalability

Control

edureka!

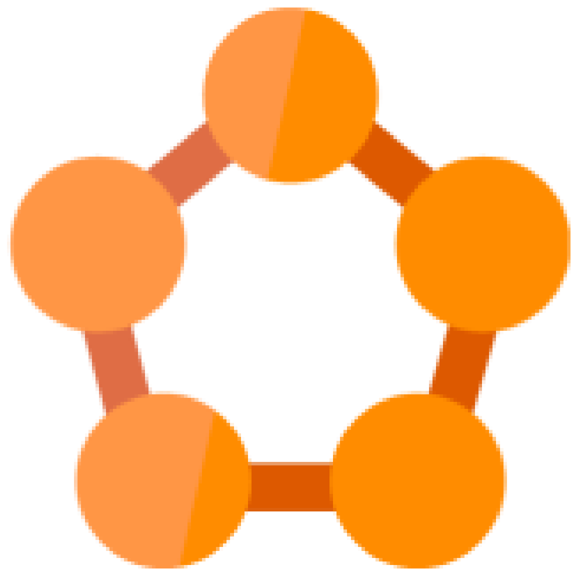
Orchestration Using Kubernetes

- Kubernetes is an open-source product that provides functionality that ranges from cluster infrastructure provisioning and container scheduling to orchestrating capabilities
- It allows users to automate deployment, scaling, and operations on applications containers across clusters of hosts

Orchestration Types: Azure Kubernetes Service (AKS)

- Azure Kubernetes Service (AKS) is defined as a managed Kubernetes container orchestration service
- It simplifies the Kubernetes Cluster's management, deployment, and operations

Orchestration Types: Azure Service Fabric



- Azure Service Fabric is a Microsoft microservices platform used for building applications
- It is an orchestrator of services that creates clusters of machines
- It can deploy services as containers or as plain processes

Orchestration Types: Azure Service Fabric Mesh

- Azure Service Fabric Mesh provides the same reliability, mission-critical performance, and scale as Service Fabric
- It also offers a fully managed and serverless platform

Introduction to Kubernetes

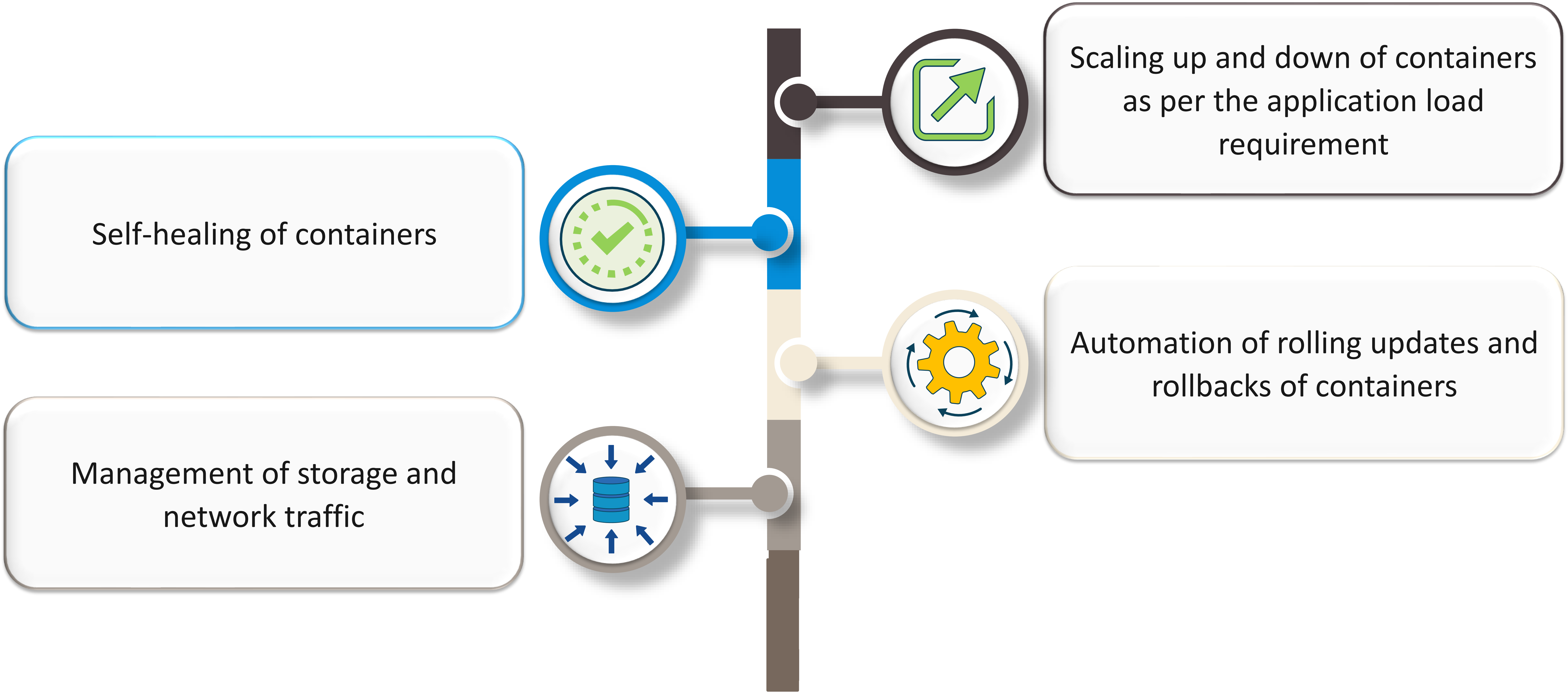
edureka!

What is Kubernetes?

Kubernetes is a portable, extensible open-source platform for managing and orchestrating containerized workloads.

- It abstracts away complex container management tasks and provides declarative configuration to orchestrate containers in different computing environments
- This orchestration platform gives the user the same ease of use and flexibility as a Platform-as-a-Service (PaaS) or Infrastructure-as-a-Service (IaaS)

Kubernetes Benefits



Scaling up and down of containers as per the application load requirement

Self-healing of containers

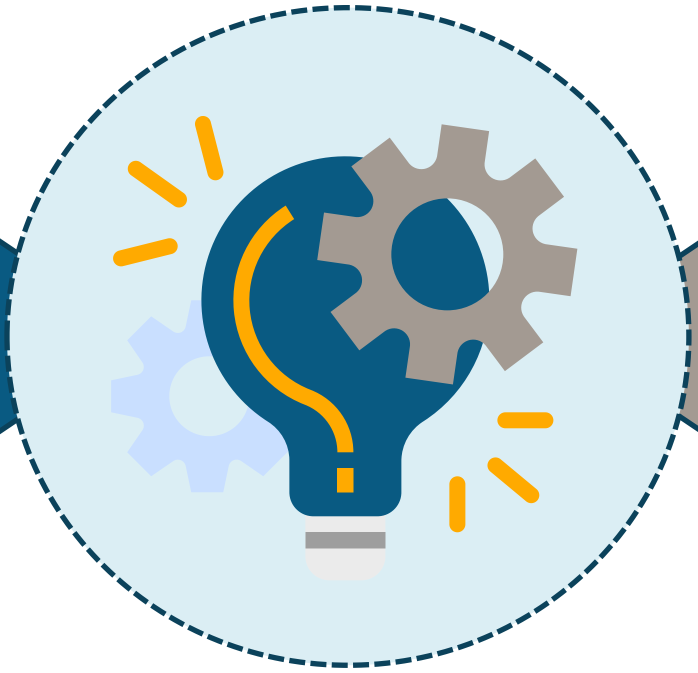
Automation of rolling updates and rollbacks of containers

Management of storage and network traffic

edureka!

Kubernetes Core Concepts

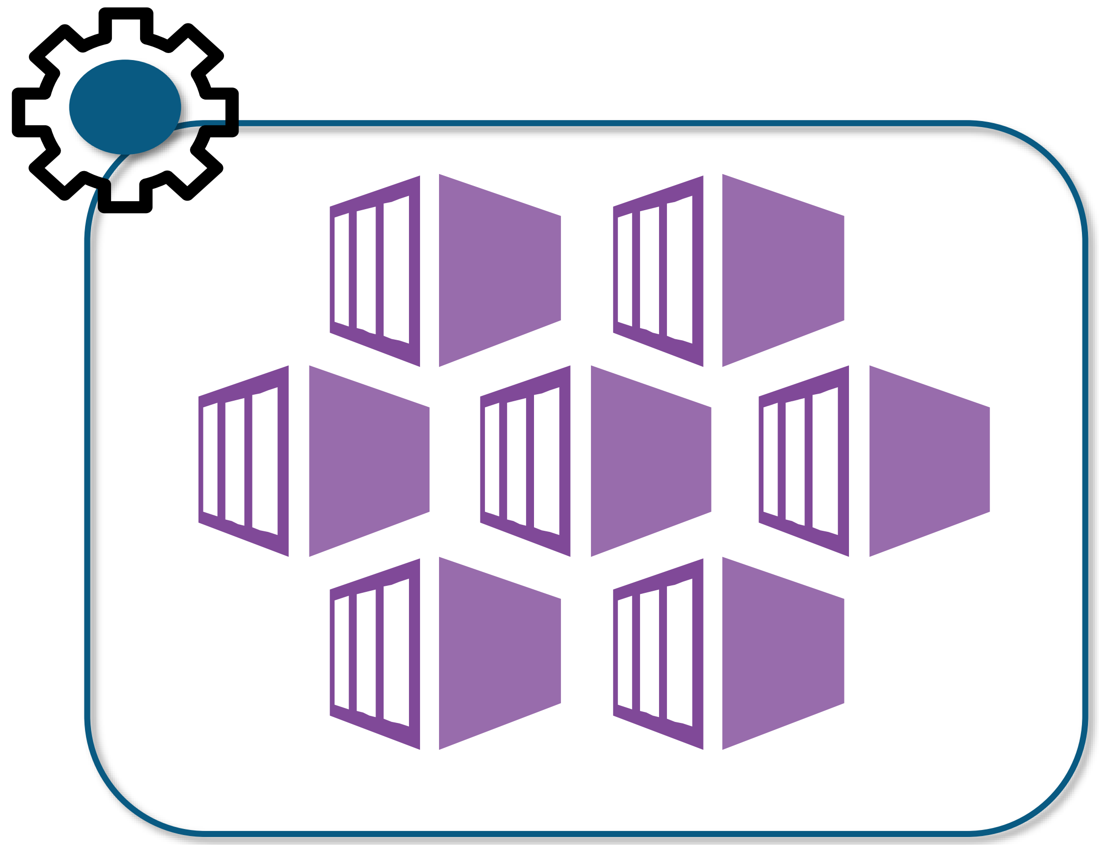
Container Management



Container orchestration

Kubernetes Core Concepts: Container Management

Container management



Process of organizing, adding, removing, or updating containers

Kubernetes manages containers through pods

Kubernetes monitors them based on health checks, load factor, etc.

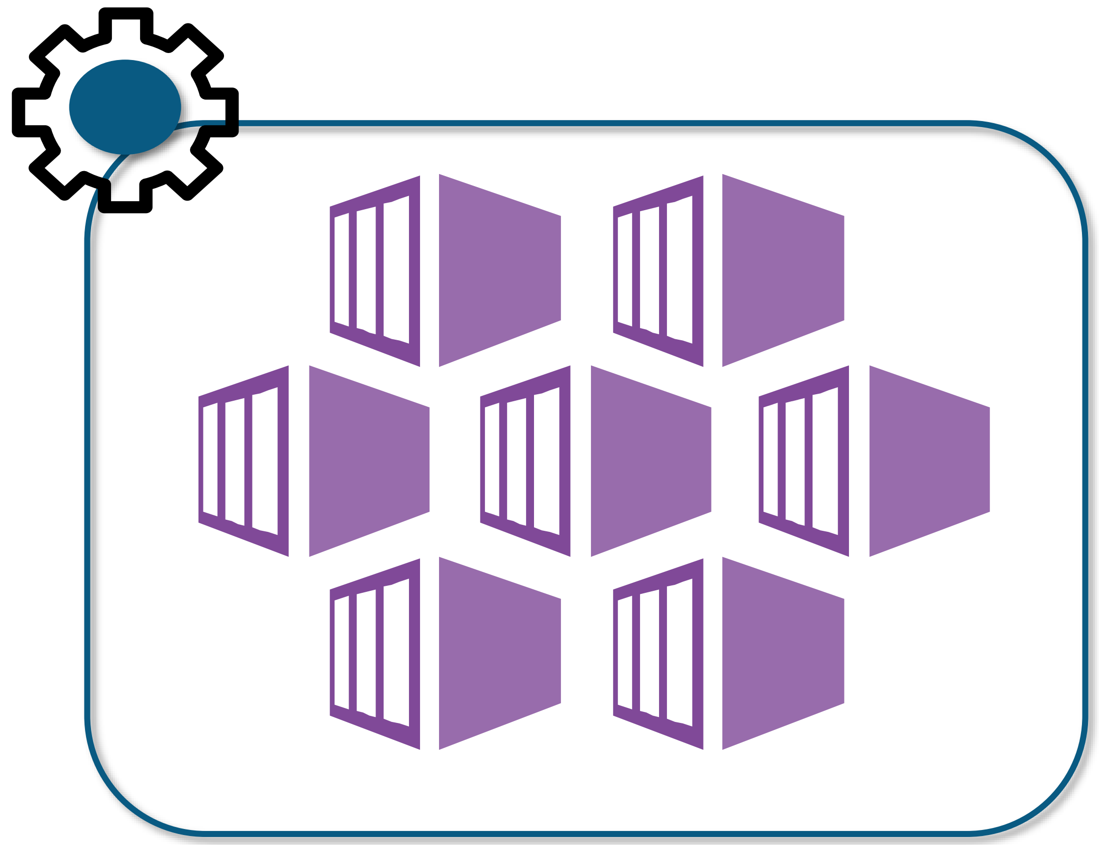
Based on these factors, Kubernetes scales the containers, starts/stops a container, etc.

Kubernetes manages all the containers hosted in it

Kubernetes Core Concepts: Container Orchestration

Container orchestration

Automatically deploys and manages containerized applications

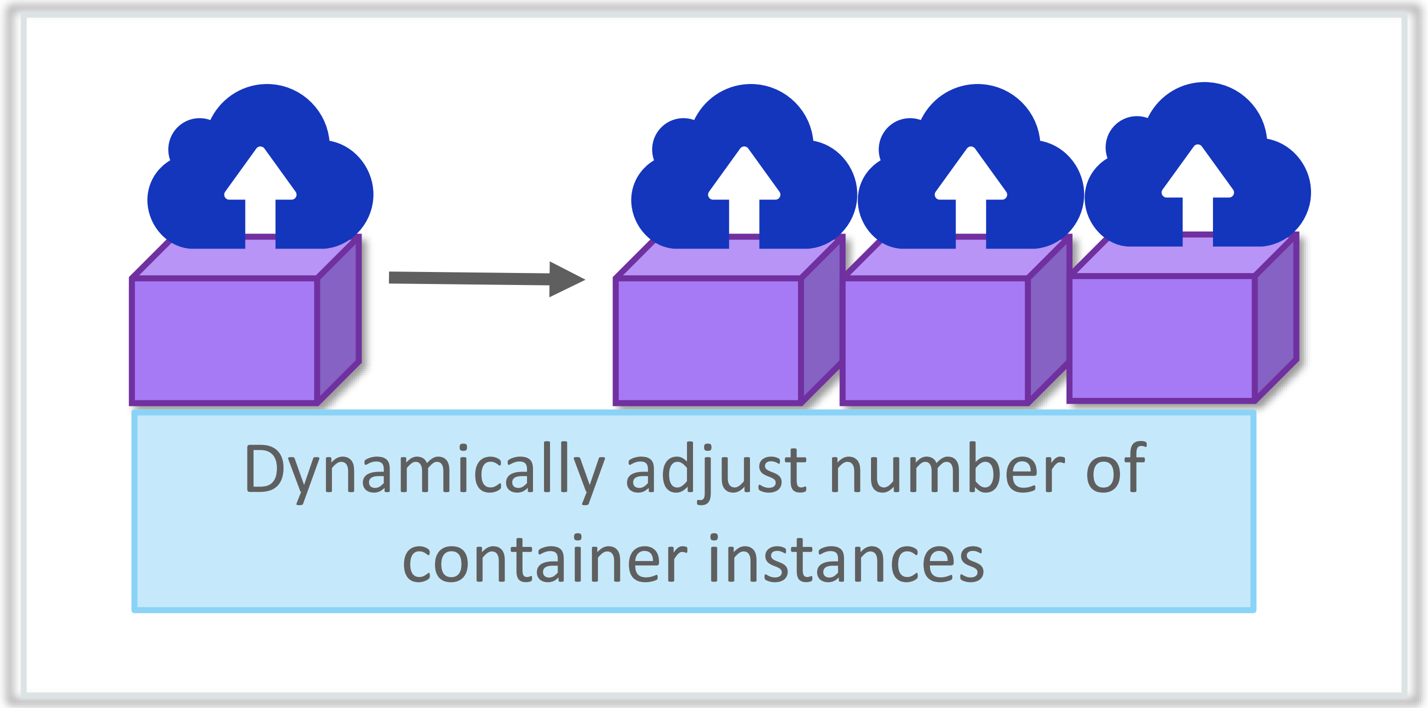


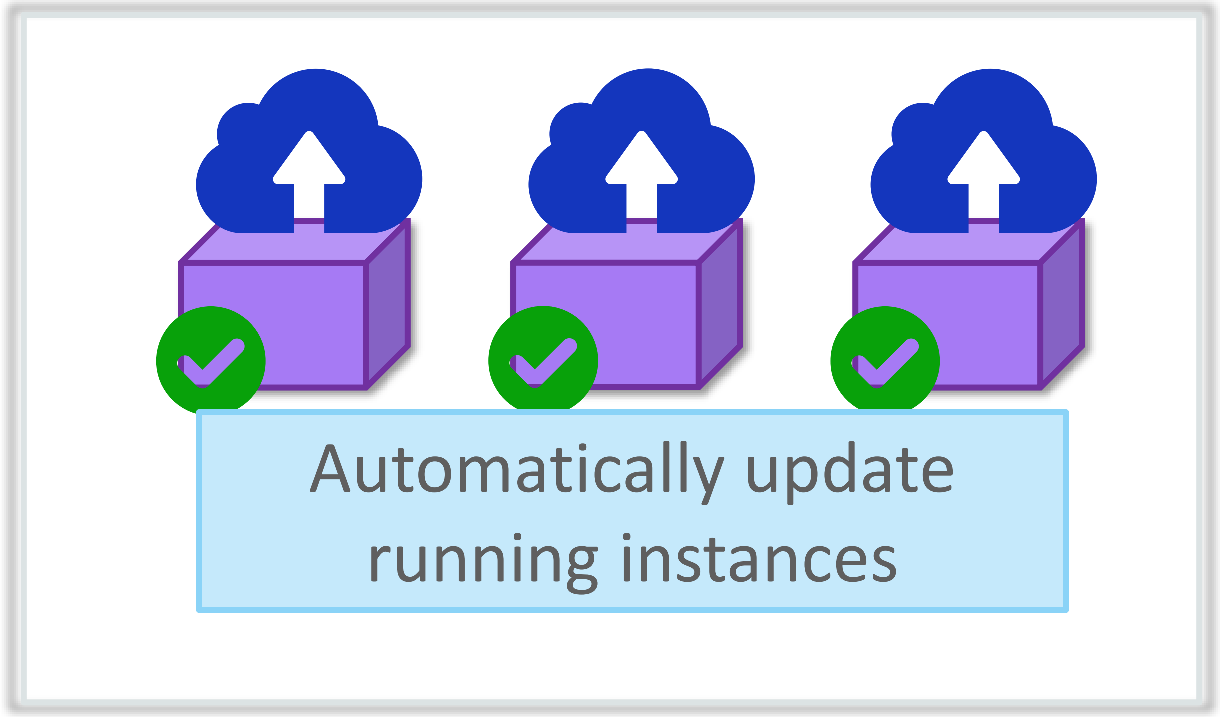
Dynamically responds to the changes in the environment and scales up or down the container

Updates the containers in case of any new patch, or any new version of service comes up

edureka!

Working of Container Orchestration

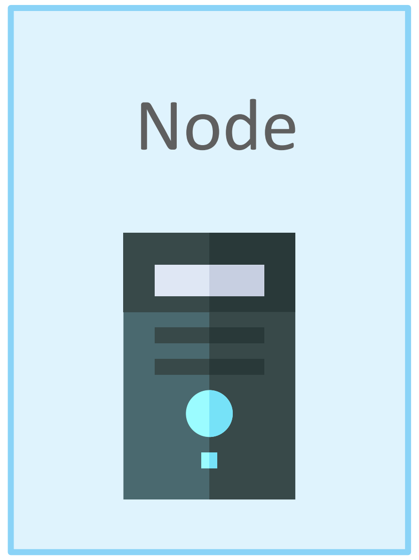


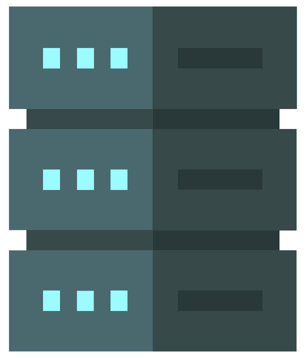


Kubernetes Cluster

A Kubernetes cluster is a cluster of nodes and a control plane.

Nodes	Where applications are deployed
Control plane	The logical layer which manages and controls the Nodes

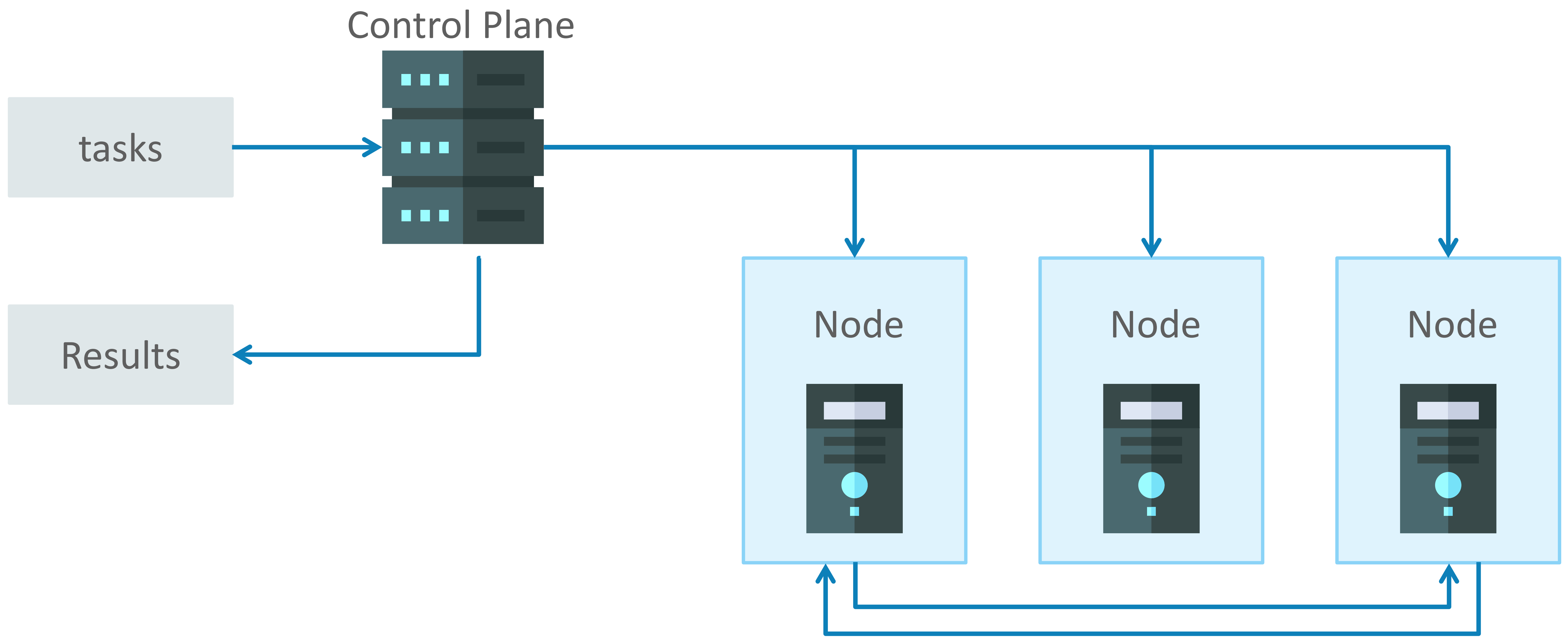




edureka!

Kubernetes Cluster (Cont.)

- A task is assigned through the control plane, and it is pushed to nodes for execution
- After execution, the result is passed back to the control plane, which in turn passes it back to the calling application/user, as shown in the arrow connecting to results

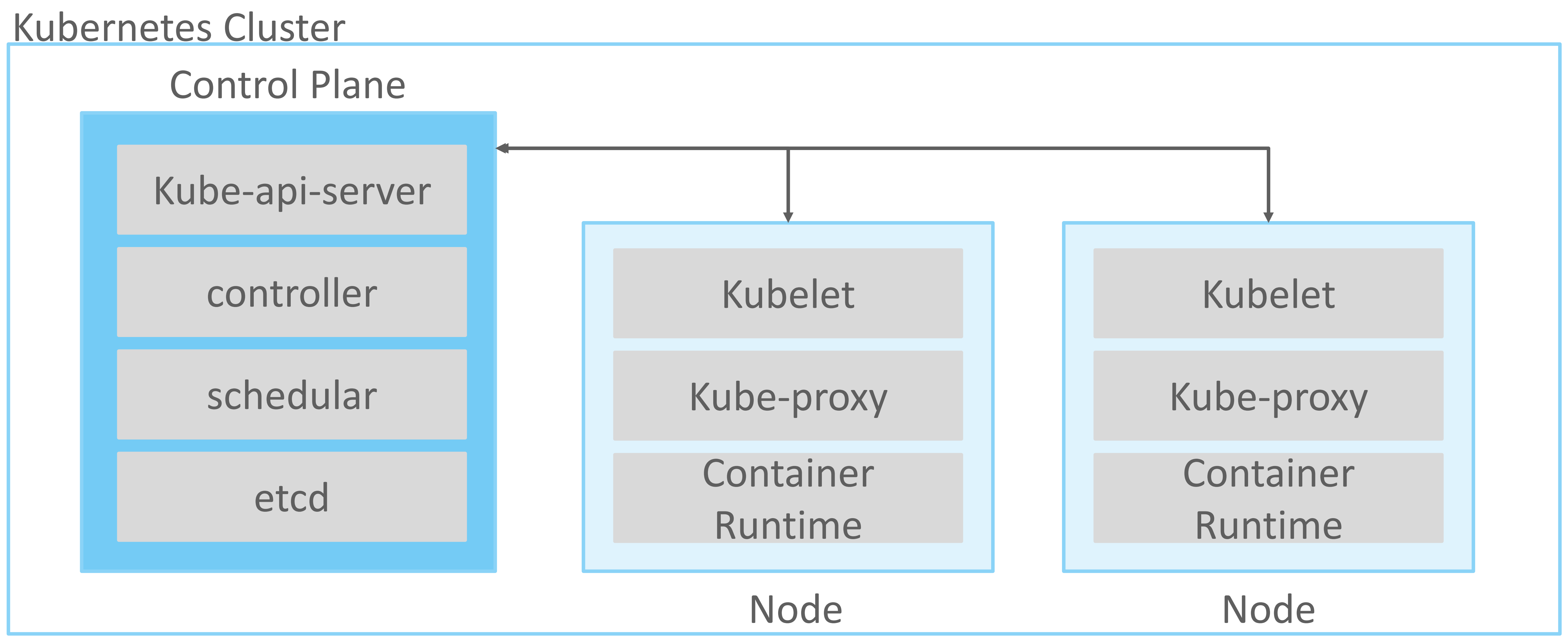


Kubernetes Architecture

edureka!

Kubernetes Architecture

- As part of Kubernetes architecture, the control plane and nodes run several services
- Control plane services manage aspects, such as cluster component communication, workload scheduling, and cluster state persistence



edureka!

Kubernetes Architecture: Control Plane Services

Kube-Api-server

- It is the front end to the control plane in your Kubernetes cluster
- All the communications are done through this API
- This API exposes a REST API, which is used for sending the commands to Kubernetes nodes

Backing Store(etcd)

- Backing store is a persistence store that Kubernetes uses to save the complete configuration of the Kubernetes cluster

Scheduler

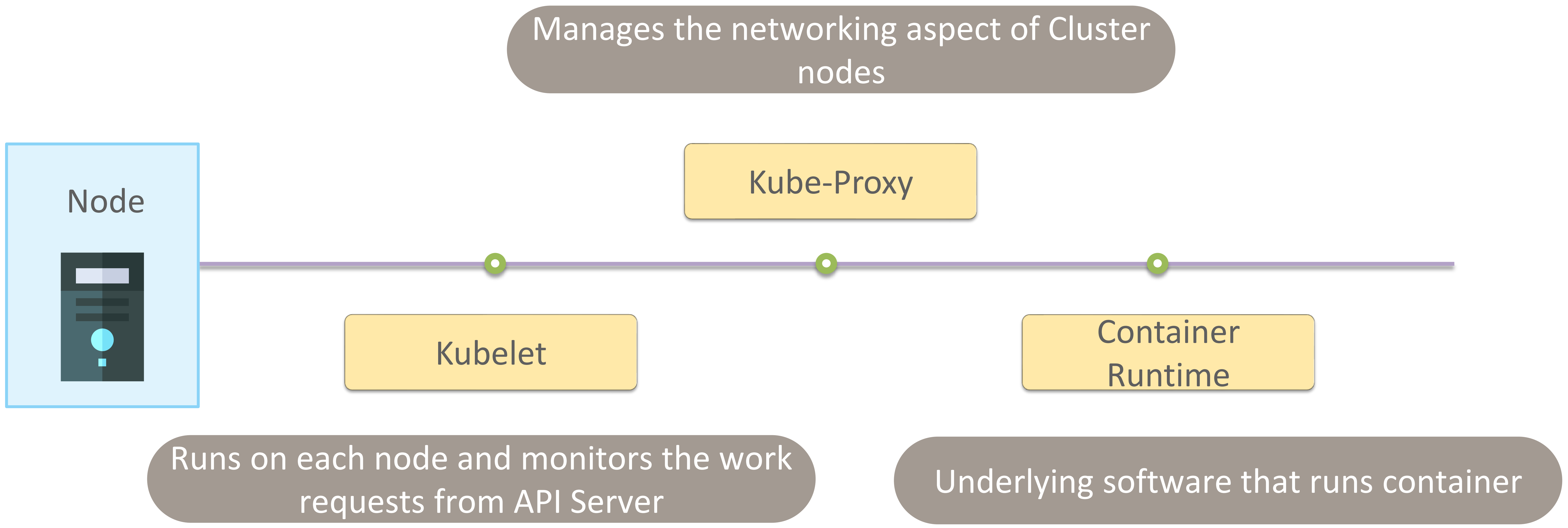
- Scheduler is responsible for the assignment of workloads across all nodes
- The scheduler monitors the cluster for newly created containers and assigns them to nodes

Controller Manager

- It monitors the state of the object in the cluster. Kubernetes uses controllers to track the state of the object in cluster

edureka!

Kubernetes Architecture: Node Services


Manages the networking aspect of Cluster nodes
Node
Kube-Proxy
Kubelet
Container Runtime
Runs on each node and monitors the work requests from API Server
Underlying software that runs container

Kubernetes Architecture: Node Services

01

Kubelet: It is an agent that processes the orchestration requests coming from the control plane.

Kube-Proxy: Kube-Proxy handles virtual networking. It manages network traffic and routing.

02

03

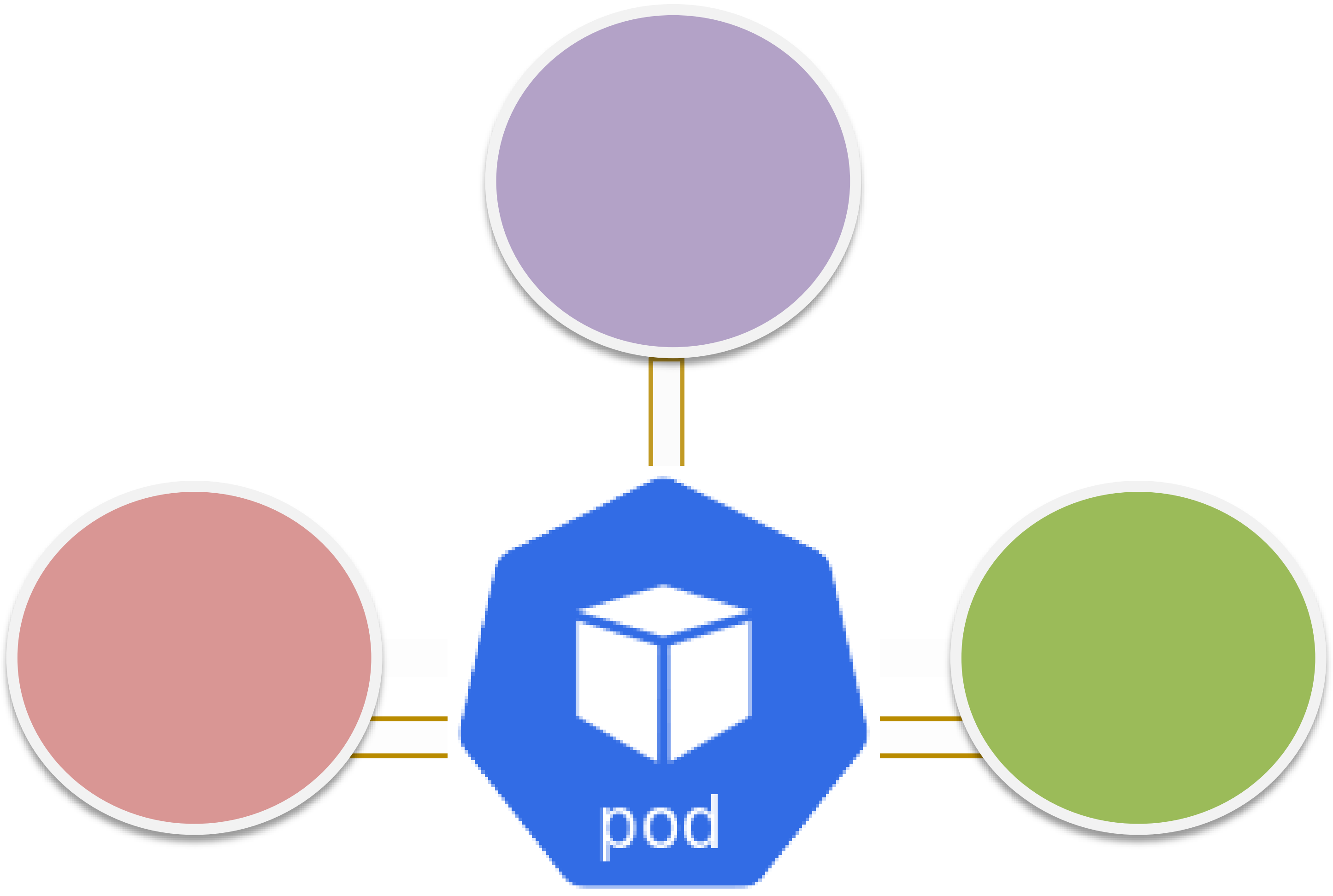
Container Runtime: It is the component that allows the containerized application to run and interact with additional resources.

edureka!

Kubernetes Architecture: Kubernetes Pods

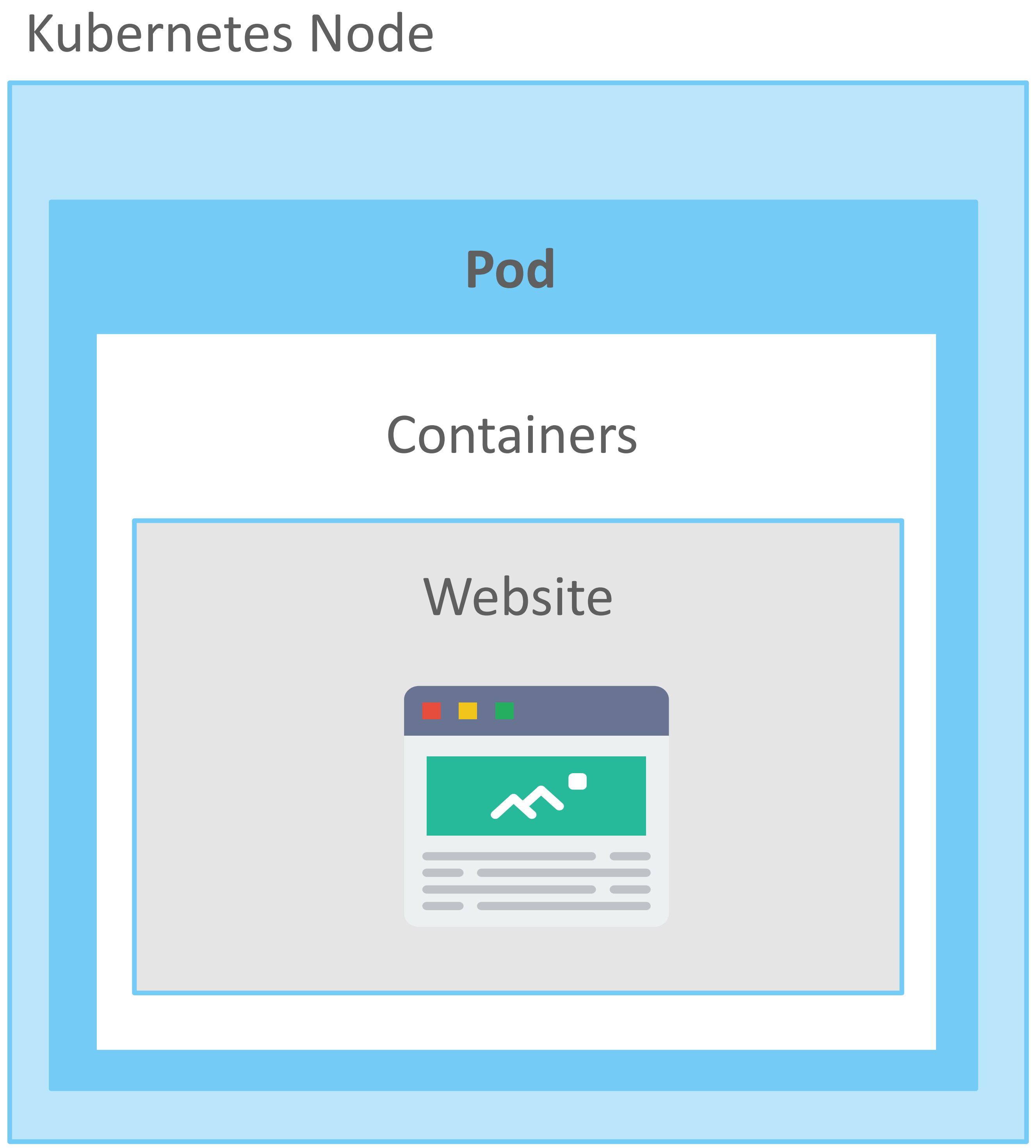
A pod is an application that runs in Kubernetes, and it holds containers

Containers are packaged in a Kubernetes object named Pod



A pod can contain multiple containers as well

Kubernetes Architecture: Node



Managing Kubernetes Cluster

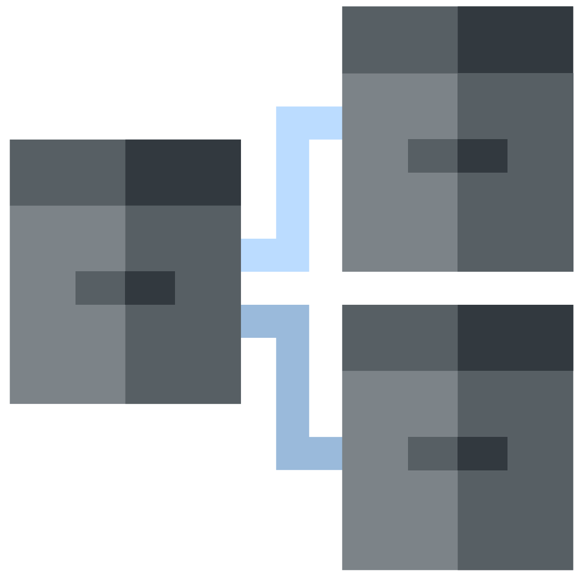
edureka!

Container Deployment in Kubernetes Cluster

After creating a Kubernetes cluster, the container can be deployed onto the clusters through the Kubernetes command-line interface Kubectl. Kubectl uses the Kubernetes API to interact with the cluster.

```
#The command to deploy a container to Kubernetes cluster is
like:

kubectl create deployment new-node –image=” Container Image”
```



edureka!

Manage Kubernetes Cluster

Managing a Kubernetes cluster involves:

Upgrading Cluster

Cluster Autoscaling

Resizing the Cluster

Node Maintenance

Azure Kubernetes Service (AKS)

Azure Kubernetes Service (AKS)

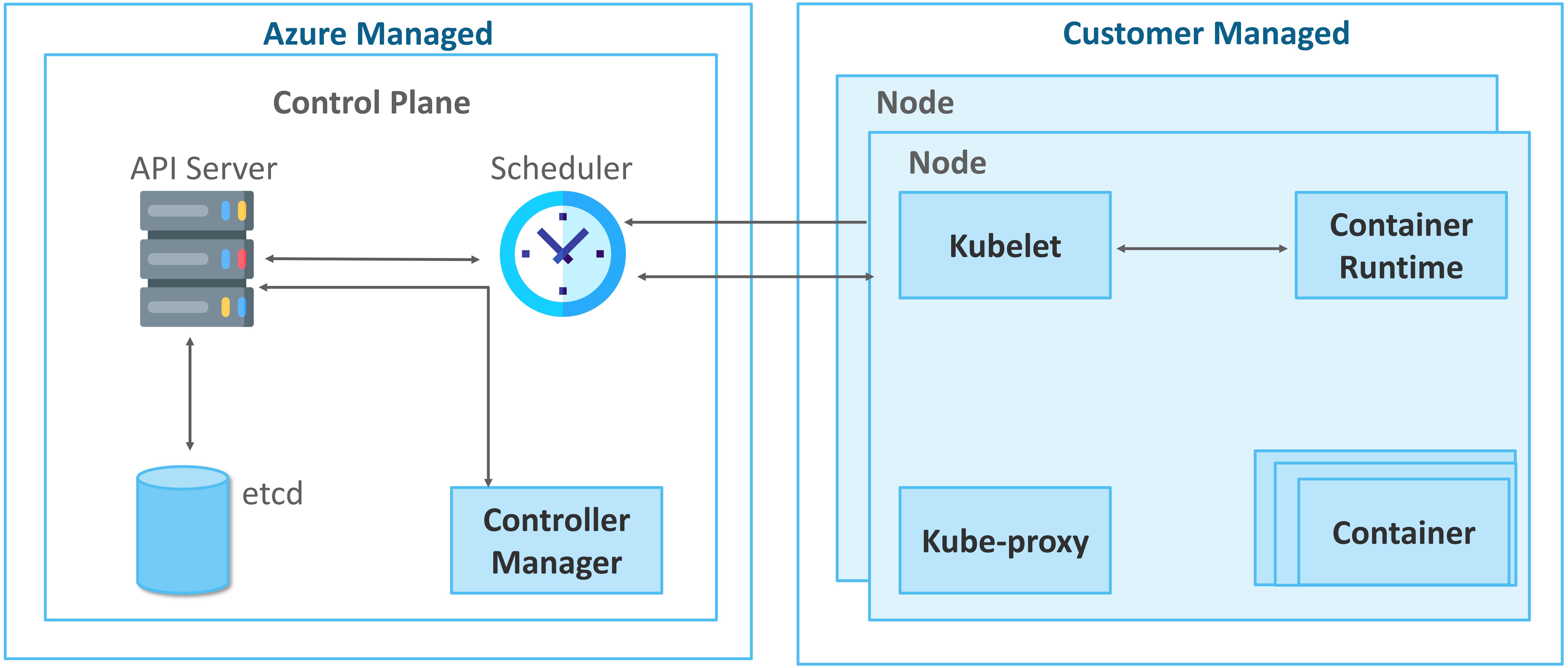
- Azure Kubernetes Service (AKS) manages the hosted Kubernetes environment and makes it simple to deploy and manage Azure's containerized applications
- AKS environment is embedded with features such as automated updates, easy scaling, and self-healing
- The Kubernetes cluster master is managed by Azure and is free
- The developer/user manages the agent nodes in the cluster and pays only for the VMs on which the nodes run

edureka!

Azure Kubernetes Service (AKS)

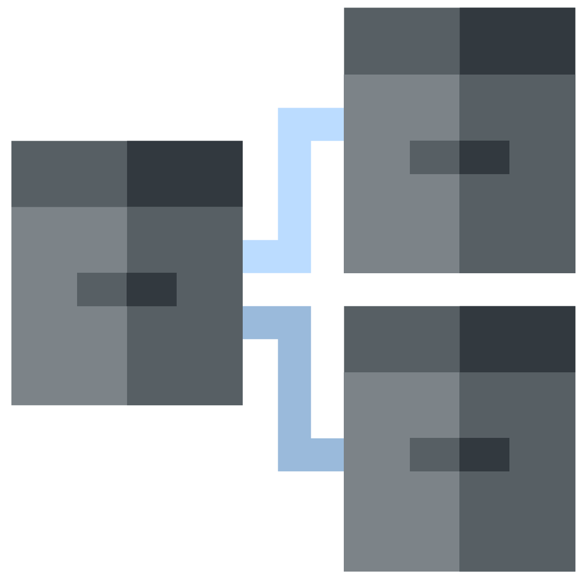
- AKS can be created either through the Azure portal or Azure CLI
- With AKS, we can get the benefit of open-source Kubernetes without the complexity, operational and management overhead
- When the application is deployed in AKS, all the Kubernetes master and other nodes are configured by AKS

AKS Architecture

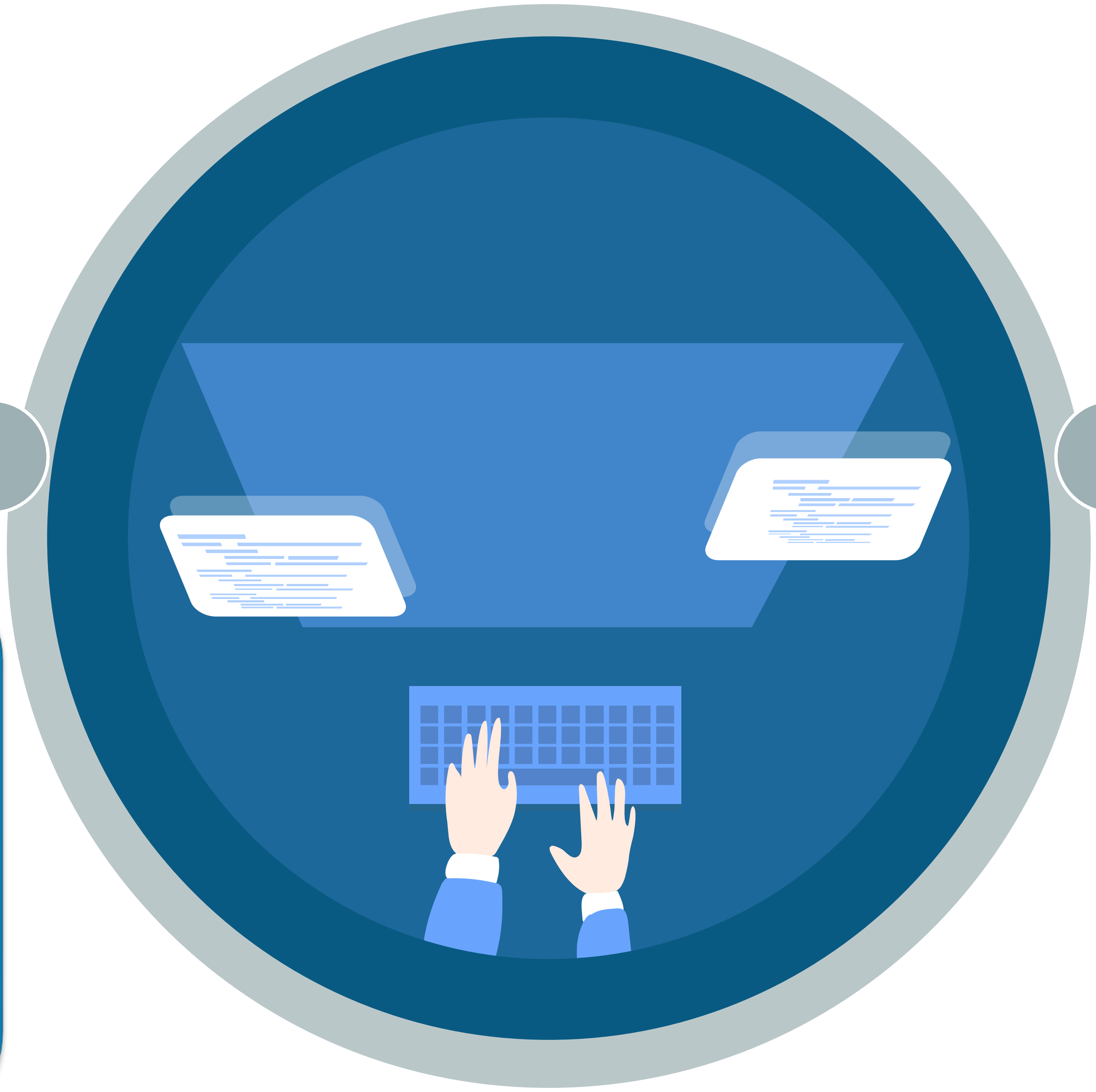


Characteristics of AKS

- On AKS creation, the control plane is automatically created, and Azure manages it
- Nodes (also known as the Kubernetes node) are created, and customers can do it all themselves
- AKS cluster can have one or more nodes. Nodes are virtual machines



AKS: Node Pools and Node Selectors



Node Pools

- Nodes of the same configuration are grouped into node pools
- A Kubernetes cluster contains one or more node pools

Node Selectors

- In an AKS cluster that contains multiple node pools, the user needs to specify Kubernetes Scheduler, which node pool uses for a given resource

edureka!

AKS Scaling Options

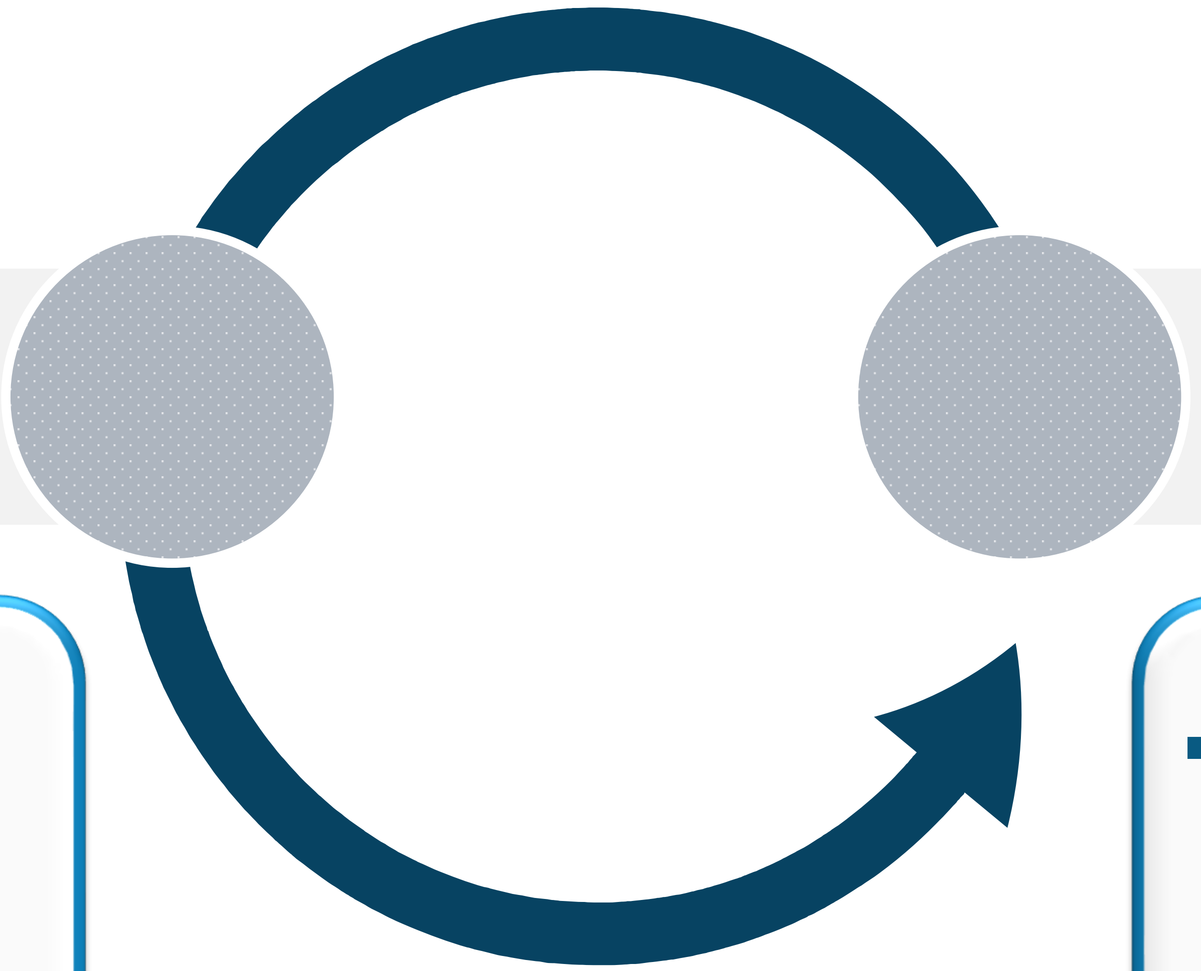
edureka!

Scaling Options in AKS

Azure Kubernetes service has the feature of scaling when it finds that it is running short of resources. Cluster Autoscalar is the service that does the Autoscaling of cluster nodes.

Azure Kubernetes Service (AKS)

Scaling Options in AKS: Autoscaler Types



Cluster Autoscaler

- Watches if any of the pods cannot be scheduled on a node due to resource constraints

Horizontal Pod Autoscaler

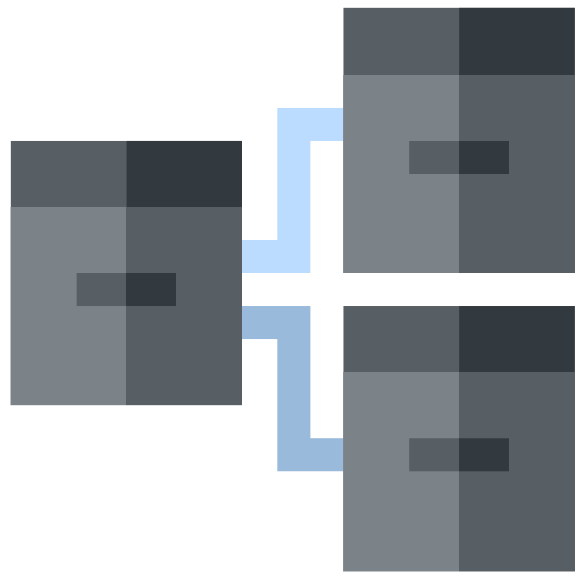
- It uses a Matrix server in the cluster to monitor the resource demand of pods

edureka!

Scaling Options in AKS: Enabling Scaling

In the Azure CLI command used below to create the AKS cluster, we have enabled the Autoscalar with a minimum node of 1 and a maximum of 5.

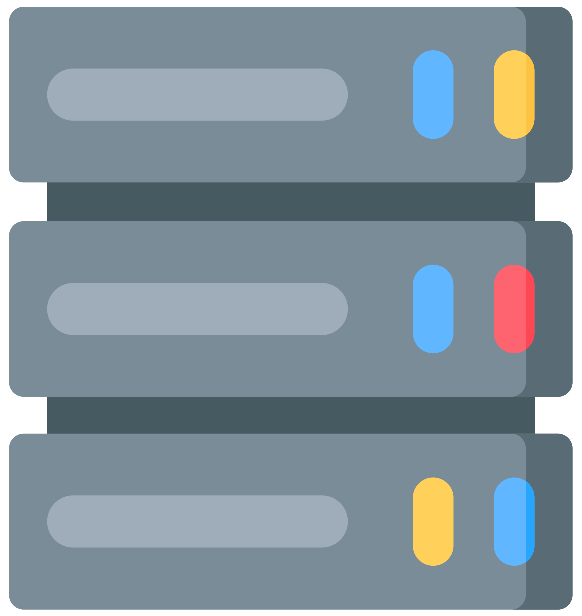
```
   #The scaling option can be enabled at the time of AKS cluster
   creation through Azure CLI like:
az aks create \
  --resource-group myResourceGroup \
  --name AKSClusterName \
  --node-count 1 \
  --vm-set-type VirtualMachineScaleSets \
  --load-balancer-sku standard \
  --enable-cluster-autoscaler \
  --min-count 1 \
  --max-count 5
```



Scaling Options in AKS: Enabling AutoScalar

In any existing cluster, the user can enable the Autoscalar through below Azure CLI command.

```
az aks update \
  --resource-group myResourceGroup \
  --name existingClusterName \
  --enable-cluster-autoscaler \
  --min-count 1 \
  --max-count 5
```



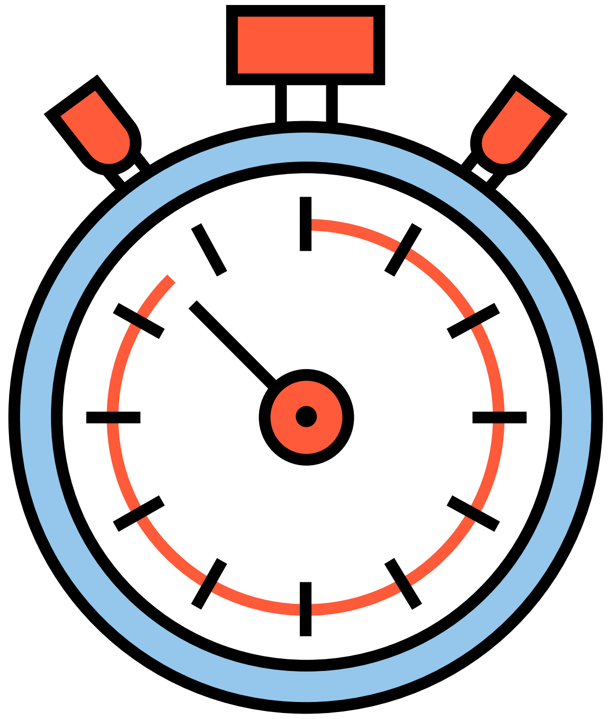
edureka!

Scaling Options in AKS: Setting Time Interval

Users can set the interval time from the last time when the action was taken for scaling.

```
# After any scaling action occurs, the cluster will wait for
# this time, and then it will take any action related to scaling.

az aks update \
     --resource-group myResourceGroup \
     --name existingClusterName \
     --cluster-autoscaler-profile scan-interval=30s
```

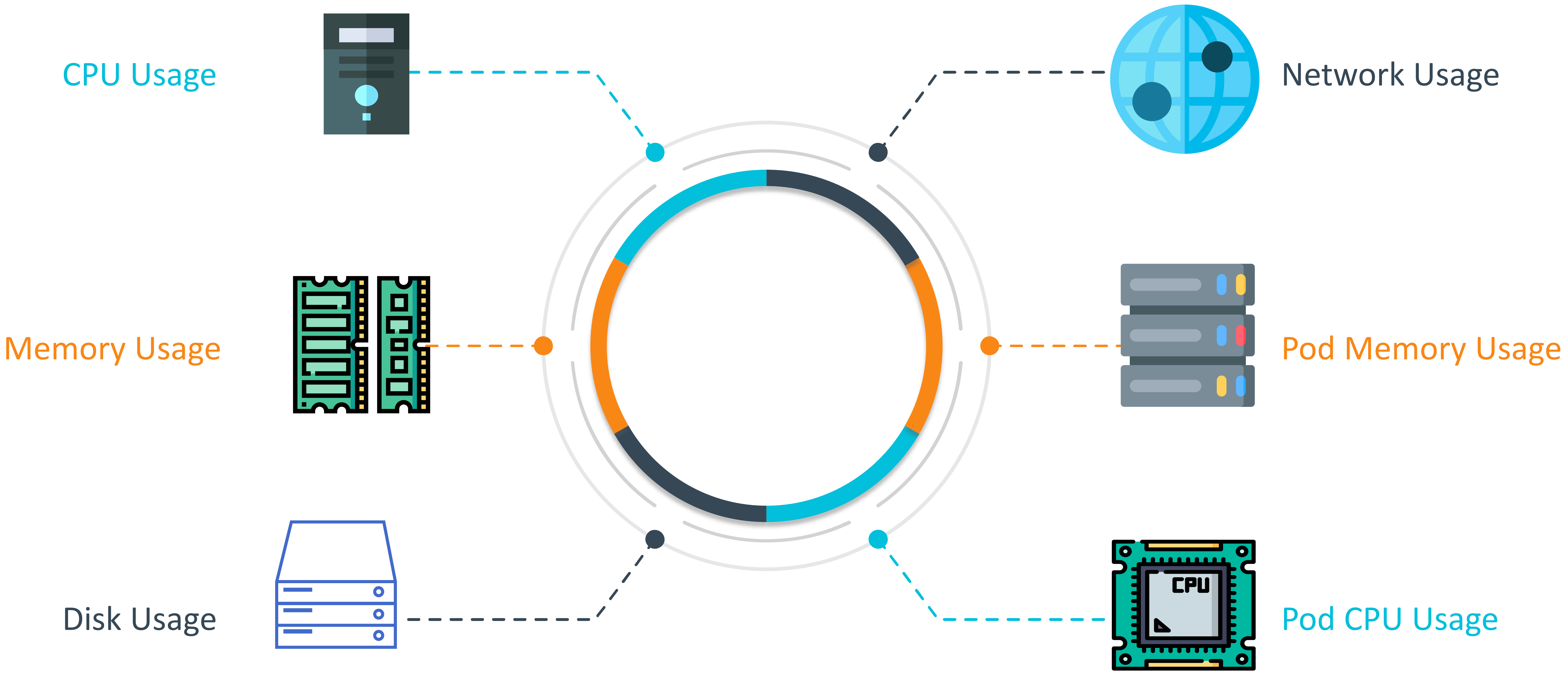


edureka!

AKS Cluster Monitoring

edureka!

Monitoring AKS Cluster



Monitoring AKS Cluster from Azure Monitor

Users can use Azure Monitor to monitor all the above parameters. Users can directly go to the Azure Monitor service.

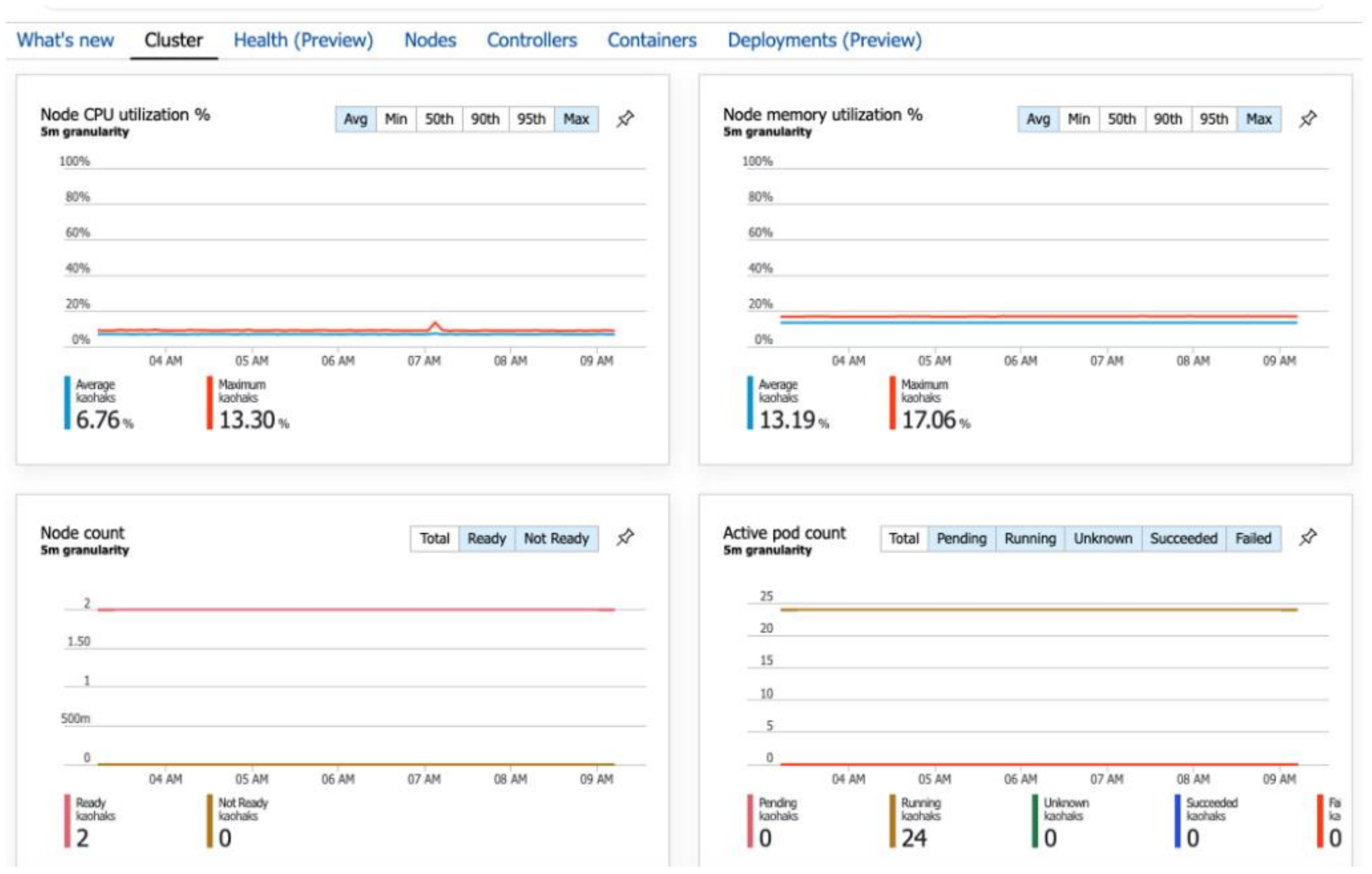
- Open Azure Monitor from Azure Portal
- Go to Insights → Containers
- Select the AKS Cluster
- Visualize all the required matrices of these clusters

Azure DevOps

Monitoring AKS Cluster from AKS

Alternatively, users can go to the Monitor section from AKS itself.

- Open Azure Monitor from Azure Portal
- Go to Monitoring→ Insights
- The snapshot below shows the matrices



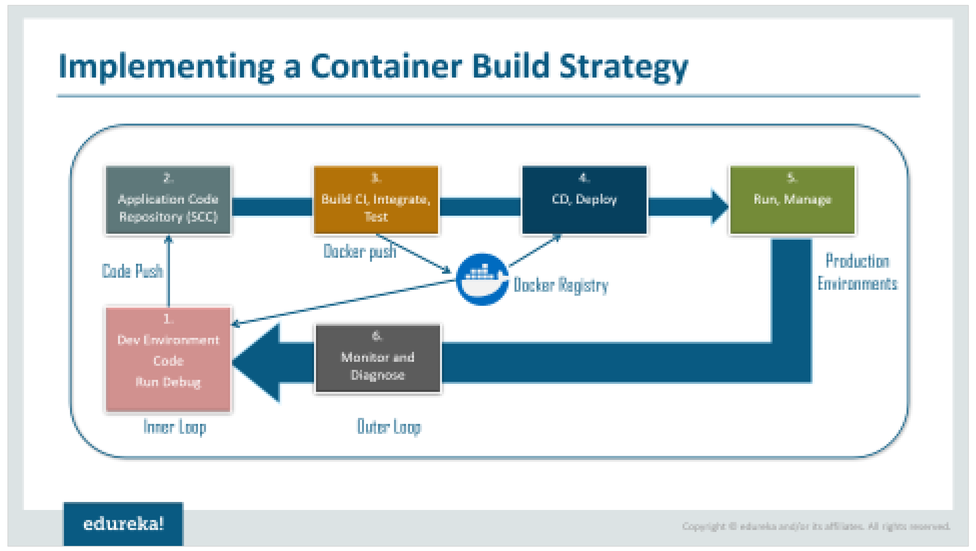
edureka!

Demonstration

- Demo 1: Deploy a multi-container application to AKS
- Demo 2: Modernize the ASP.NET application to Azure

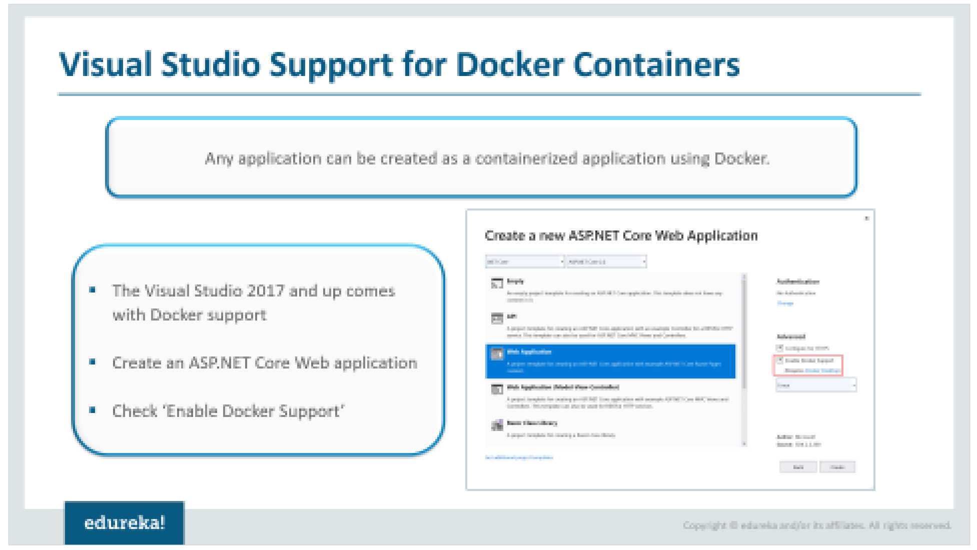
edureka!

Summary



Implementing a Container Build Strategy

edureka!

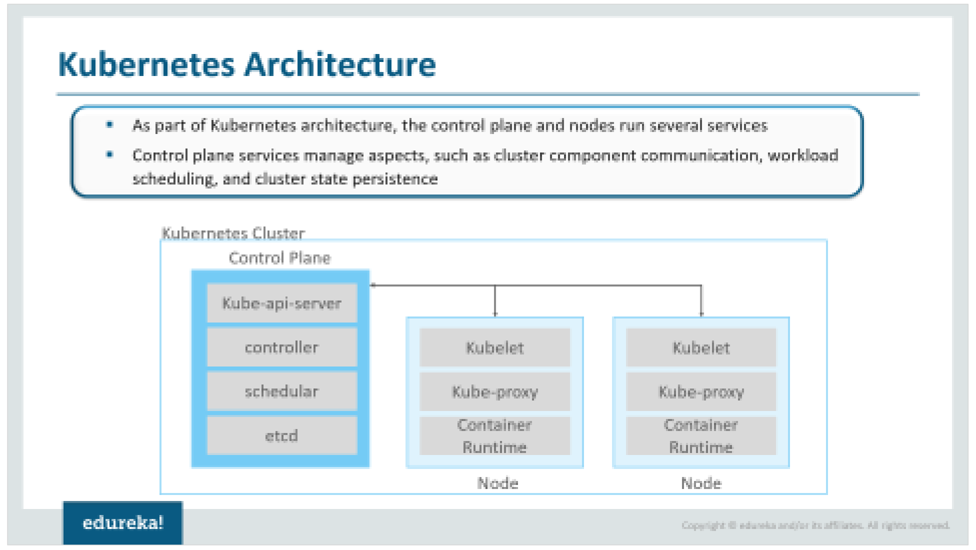


Visual Studio Support for Docker Containers

Any application can be created as a containerized application using Docker.

- The Visual Studio 2017 and up comes with Docker support
- Create an ASP.NET Core Web application
- Check 'Enable Docker Support'

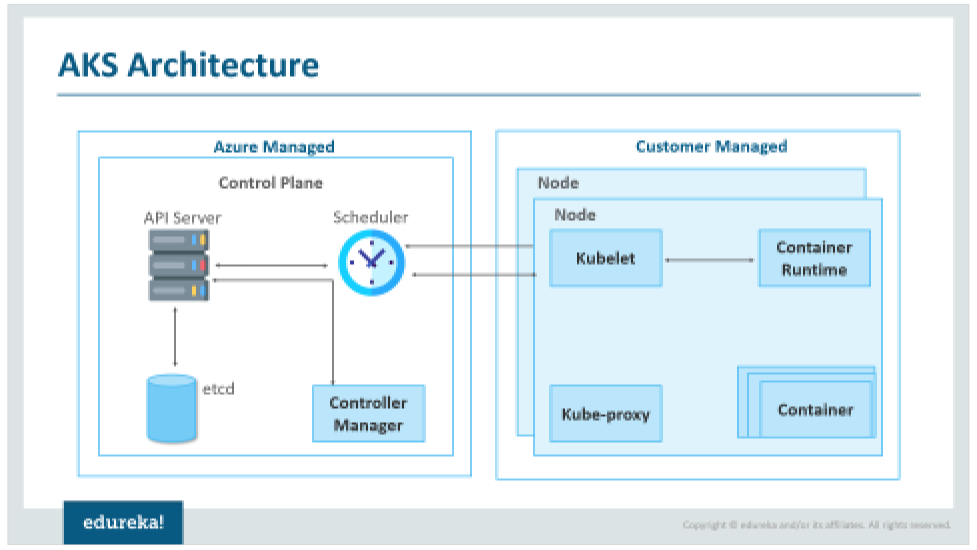
edureka!



Kubernetes Architecture

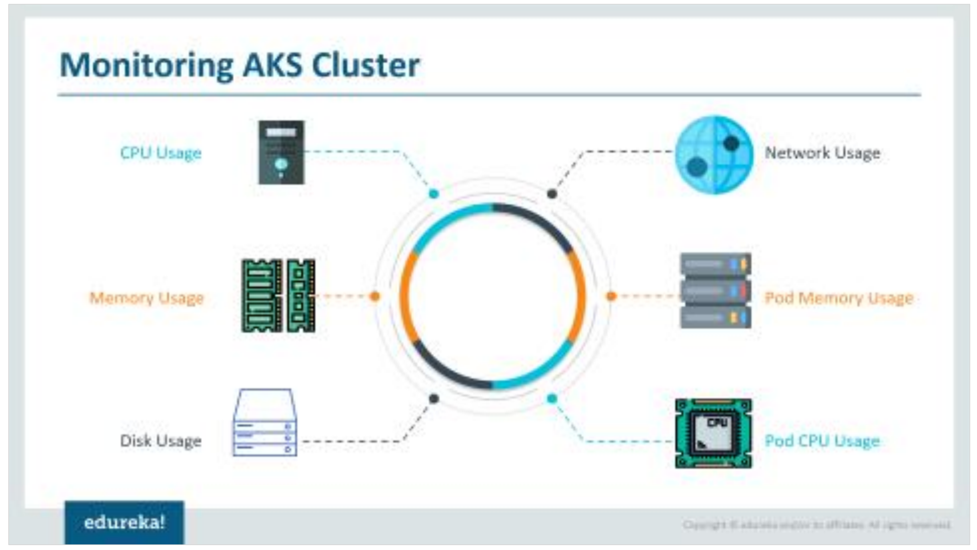
- As part of Kubernetes architecture, the control plane and nodes run several services
- Control plane services manage aspects, such as cluster component communication, workload scheduling, and cluster state persistence

edureka!



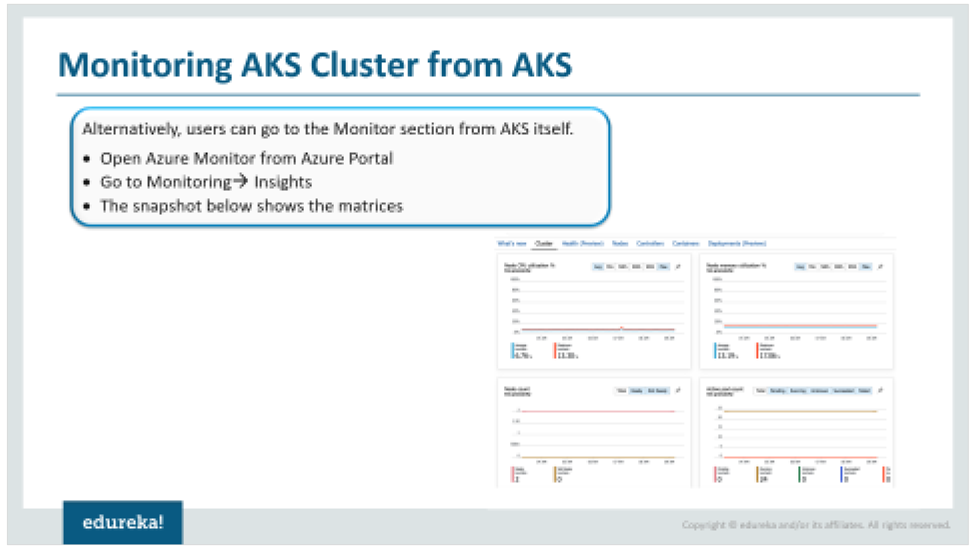
AKS Architecture

edureka!



Monitoring AKS Cluster

edureka!



Monitoring AKS Cluster from AKS

Alternatively, users can go to the Monitor section from AKS itself.

- Open Azure Monitor from Azure Portal
- Go to Monitoring→ Insights
- The snapshot below shows the matrices

edureka!

Questions

edureka!

FEEDBACK

edureka!

Thank You

For more information please visit our website
www.edureka.co