Certified Kubernetes Administrator

Kubernetes: Core Concepts

```
#include <iostream>
#include <cstdlib>
using namespace std;
int main()
{
  cout << "Hello, world!" << endl;
  return 0;
}
```

Learning Objectives

By the end of this lesson, you will be able to:

- Install etcd
- Install kubelet in worker nodes using kubeadm method
- Create pods using the YAML file and kubectl command and on specific namespace
- Create ReplicaSet using the YAML file and kubectl command
- Create deployment using the YAML file and kubectl command
- Create service using the YAML file and kubectl command

etcd

Introduction to etcd

etcd is a consistent and distributed key value store used for discovering services.

Here are the characteristics of etcd:

- Written in **Go** programming language
- Can be configured externally
- Used to store configuration details

etcd Requirements

Usually etcd can be run only with limited resources for development or testing purposes, but when running etcd clusters in production, the following hardware guidelines are required:

- High CPU capacity
- 8GB memory for small deployment
- 16GB to 64GB for heavy deployments
- Fast disk of about 7200 RPM
- 1GbE network for common etcd deployments and 10GbE network for large etcd clusters

Assisted Practice: etcd Installation

Duration: 30 min.

Problem Statement:

Install an etcd.

Introduction to etcdCTL Tools

Given below are a few etcdCTL tools:
➢ etcdctl
➢ etcd-backup
➢ etcd-dump
➢ etcd-fs
➢ etcddir
➢ etcd-browser
➢ etcd-lock
➢ etcd-console
➢ etcd-viewer
➢ etcd-rest
➢ etcdsh

Introduction to etcdCTL Tools

etcdctl is a command line client for etcd.

Here are the operations of etcdctl:

- PUT
- GET
- DEL
- TXN
- COMPACTION
- WATCH
- LEASE

kube-api Server

What Is kube-api Server?

kube-api server is the front-end of Kubernetes which also exposes the Kubernetes API.

It is a gateway to the Kubernetes cluster that implements a RESTful API over HTTP.

It validates the API requests and updates the corresponding objects in the database.

Features of kube-api Server

Implements a RESTful API

Stateless

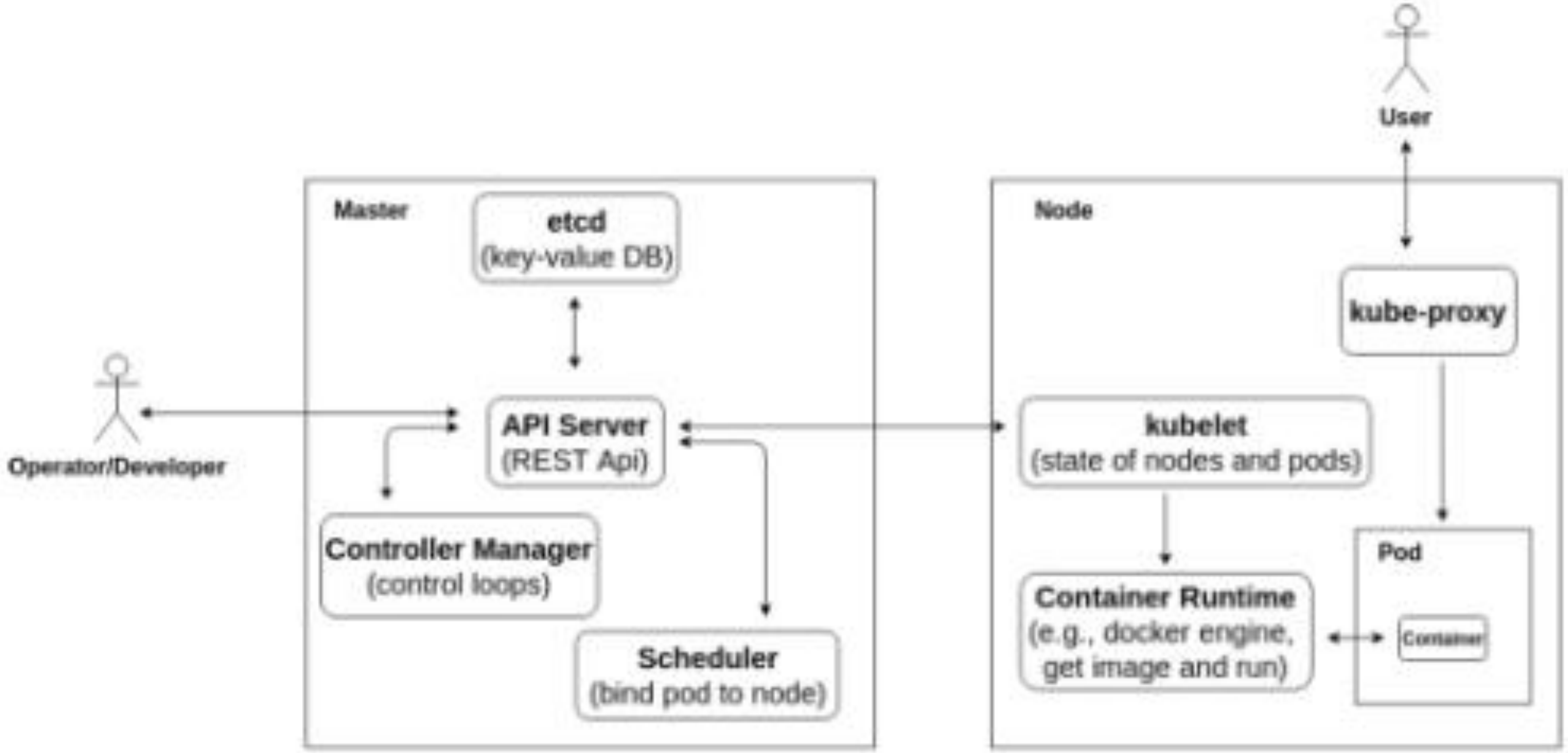
Can be connected to etcd

Fault tolerant as it can be replicated

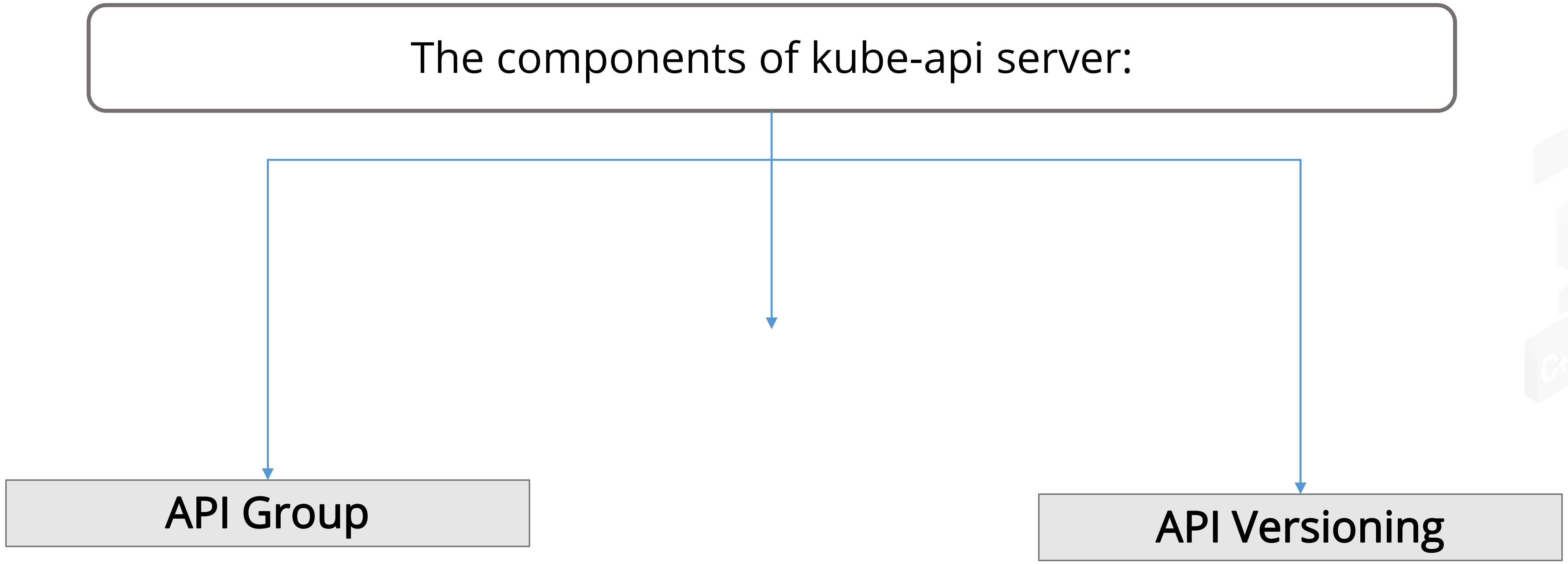
Authorizes and authenticates various mechanisms

Working of kube-api Server

Here is a diagram explaining the working of kube-api server:



Kube-api Server Components



The components of kube-api server:

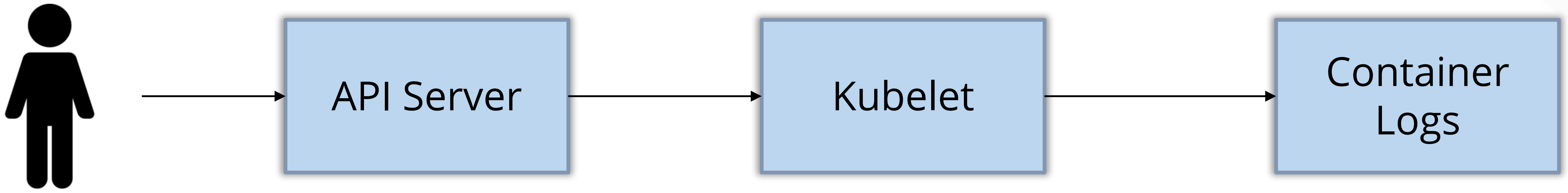
API Group

API Versioning

Criticality of kube-api in K8s Cluster

Here are the criticalities of kube-api in K8s cluster:

- Is the front-end of Kubernetes control plane
- Exposes Kubernetes API
- Scales up by deploying more instances



Controller Manager

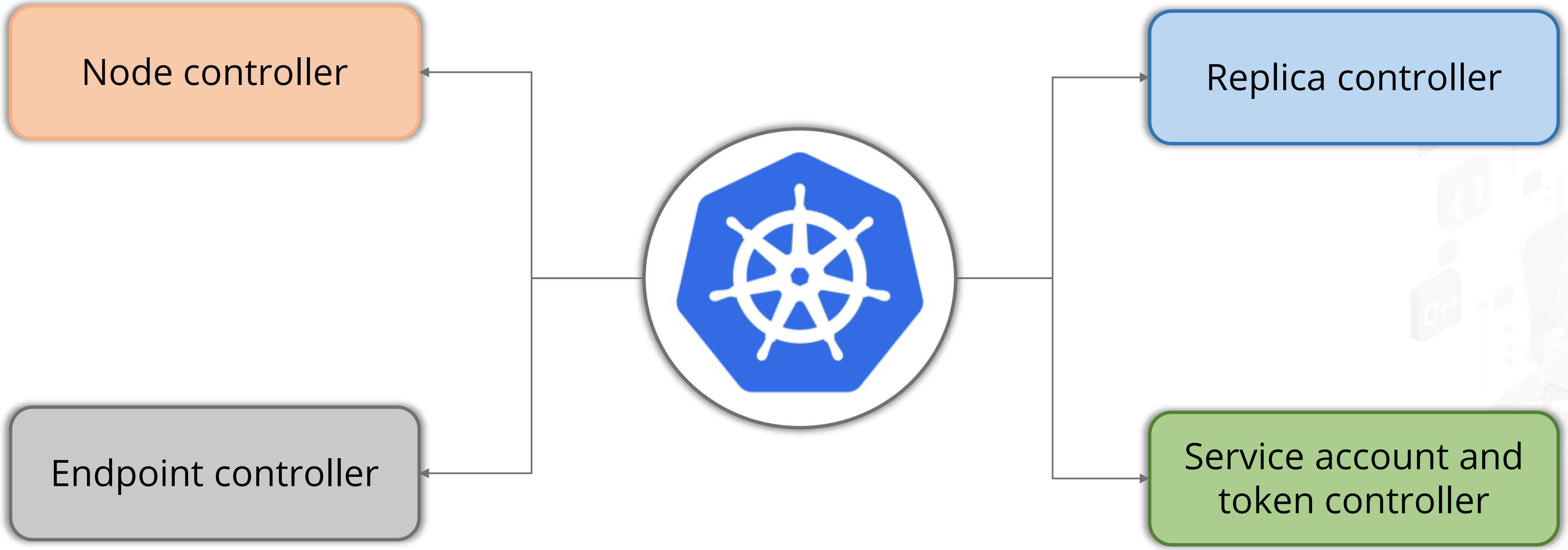
What Is a Controller Manager?

The components of kubernetes that are used to manage the non-terminating control loops and regulate the Kubernetes cluster

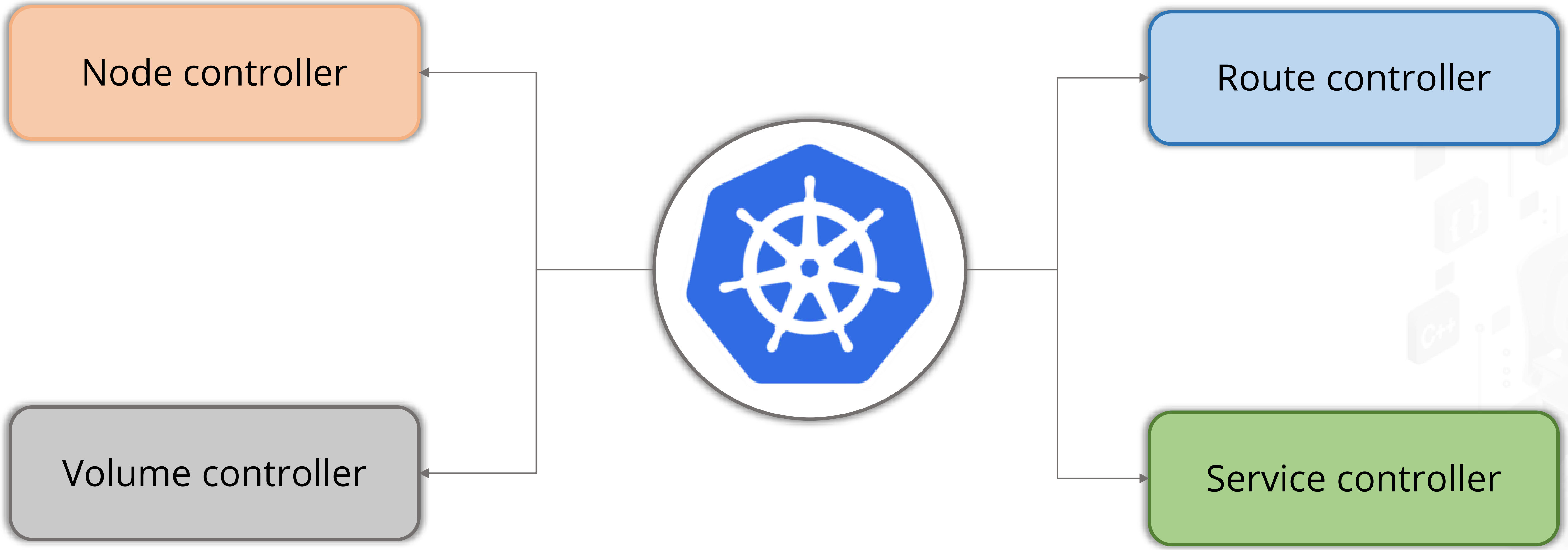
The two types of controller managers:

- Kube-Controller-Manager
- Cloud-Controller-Manager

Types of Kube-Controller Manager



Types of Cloud-Controller Manager



Node controller

Volume controller

Route controller

Service controller

Node Controller

When the nodes crash down, the node controller is responsible for checking and taking action on them.

The uses of a node controller:

- Manages various aspects of nodes
- Assigns CIDR block to the node
- Keeps internal list of nodes updated
- Monitors node health

Replication Controller

It is a node that is used to maintain the number of pods for every object in the system.

Some uses of a replication controller:

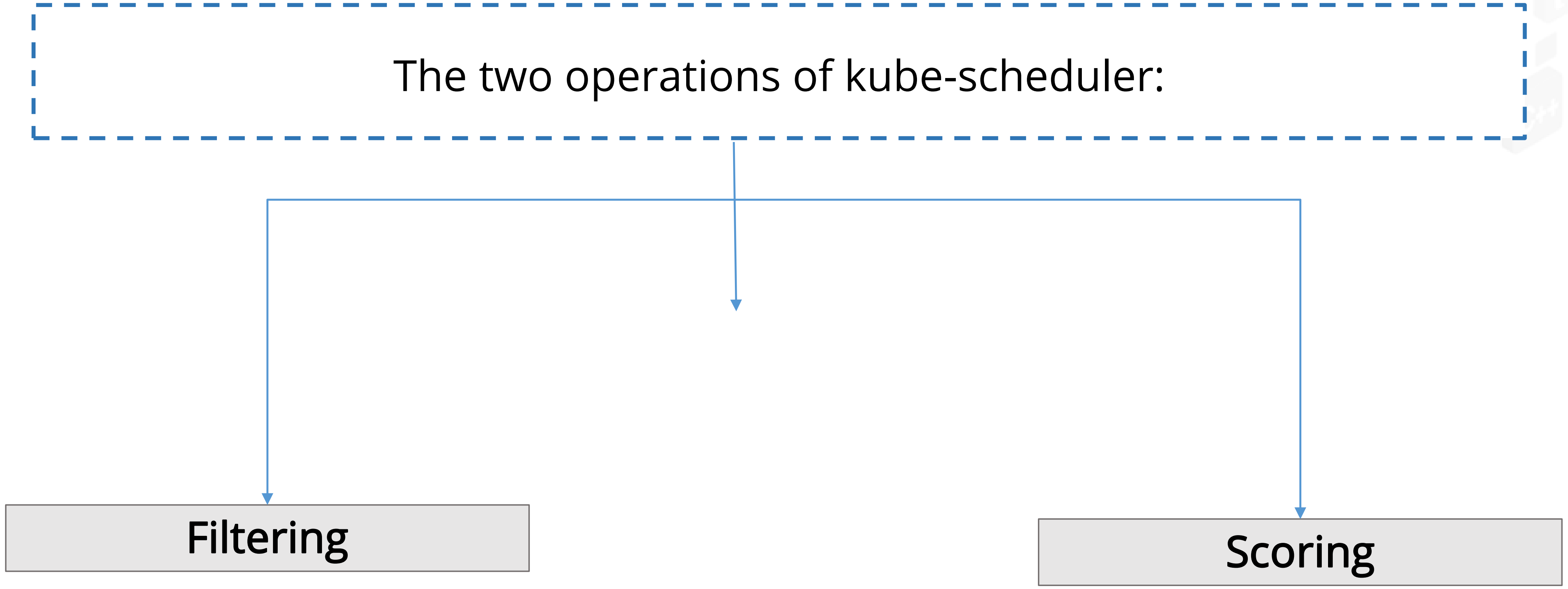
- Ensures that replication pods are up and running
- Terminates the extra pods
- Automatically replaces the failed or terminated pods

Scheduler

What Is Kubernetes Scheduler?

It is a component of the master node that is responsible for scheduling tasks for the slave nodes and storing resource information of each node.

When a pod is created, the scheduler finds the best node to run that pod.



The two operations of kube-scheduler:

Filtering

Scoring

Uses of Scheduler

A few uses of a Kubernetes scheduler:

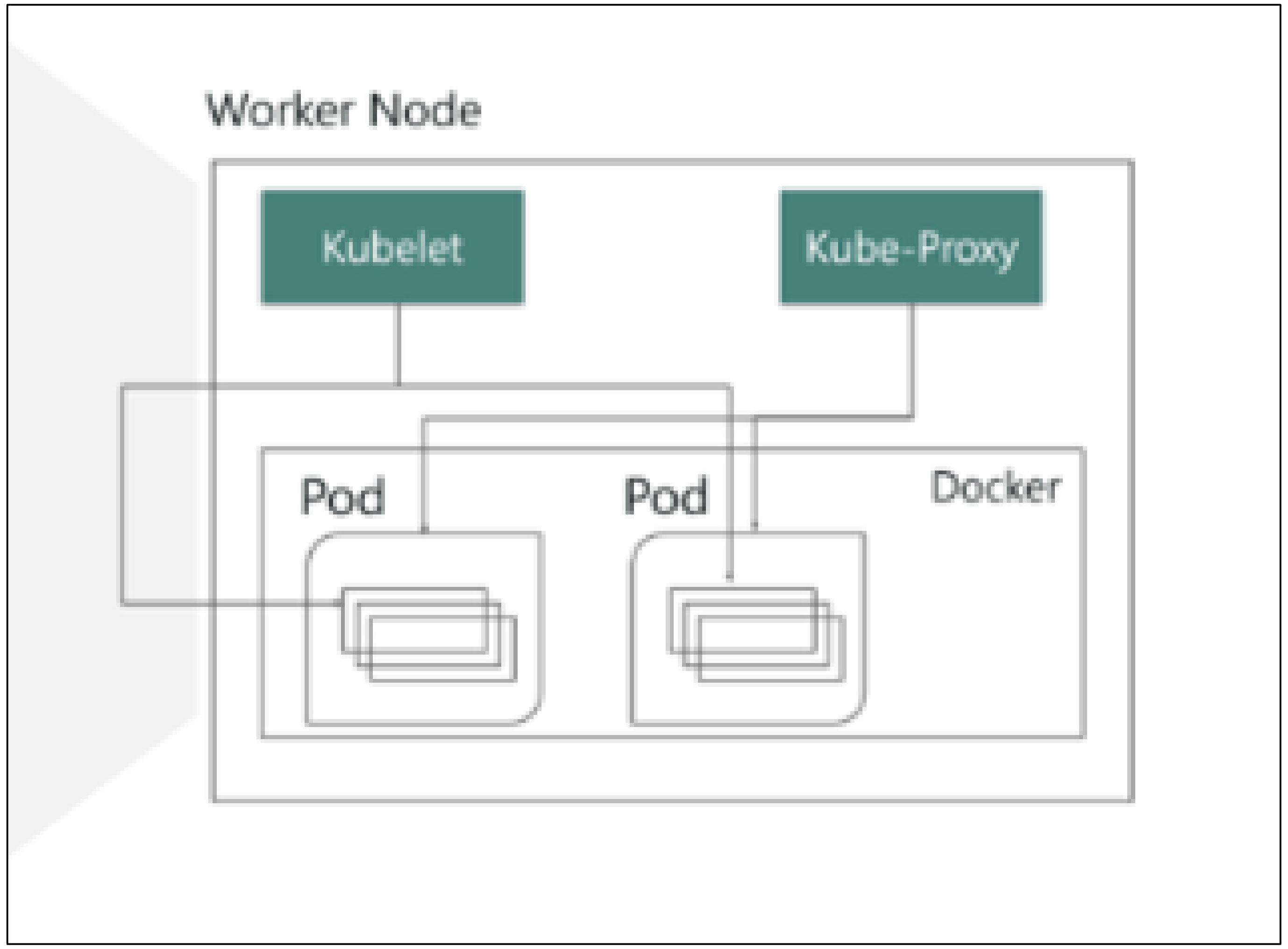
- Controls the performance and availability of nodes
- Matches each pod to the resource of the nodes
- Allows user to implement a chosen scheduler
- Allows multiple schedulers to work simultaneously

Kubelet

What Is Kubelet?

It is an agent which runs on each node and is responsible for ensuring that all containers are running.

It uses the API server to get the configurations of the pods and ensures the containers described are up and running.



Uses of Kubelet

Some uses of a Kubernetes scheduler:

- Authenticates the cluster
- Interacts with etcd store to read configurations
- Maintains the state of work
- Maintains the containers’ launch time
- Destroys the container whenever needed

Assisted Practice: Installation of Kubelet in Worker Nodes Using the Kubeadm Method

Duration: 30 min.

Problem Statement:

Install kubelet in worker nodes using the kubeadm method.

Kube Proxy

What Is Kube Proxy?

It is a network proxy and a load balancer that runs on each node of the cluster and is responsible for forwarding requests.

It allows TCP or UDP stream forwarding and is responsible for ensuring that the network is accessible.

Options of Kube Proxy

Here are a few options that can be used with kube-proxy:
➢ --azure-container-registry-config string
➢ --bind-address 0.0.0.0   Default: 0.0.0.0
➢ --cleanup
➢ --cleanup-ipvs   Default: true
➢ --cluster-cidr string
➢ --config string
➢ --config-sync-period duration   Default: 15m0s
➢ --conntrack-max-per-core int32   Default: 32768
➢ --conntrack-min int32   Default: 131072
➢ --conntrack-tcp-timeout-close-wait duration   Default: 1h0m0s
➢ --conntrack-tcp-timeout-established duration   Default: 24h0m0s
➢ --feature-gates mapStringBool
➢ --h, --help
➢ --proxy-mode ProxyMode

Uses of Kube Proxy

The uses of kube-proxy:

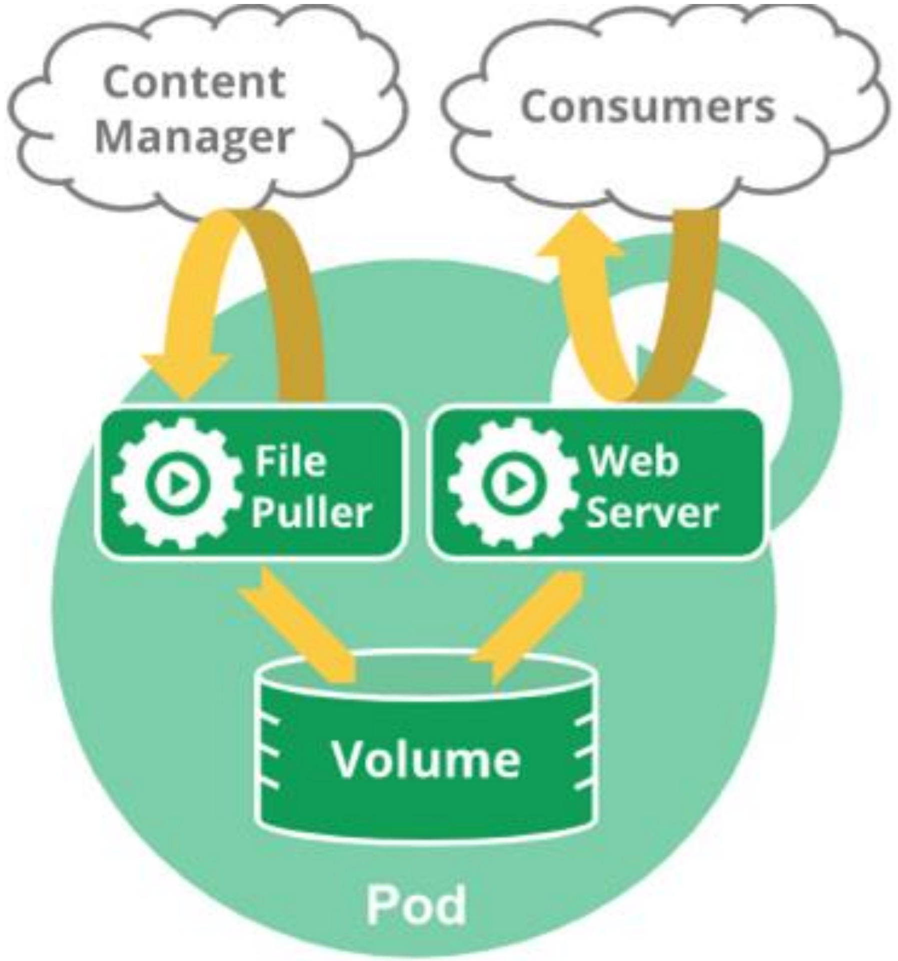
- Enables Kubernetes service abstraction
- Performs connection forwarding
- Allows TCP and UDP stream forwarding
- Performs load balancing

Pods

What Is a Pod?

It is the smallest unit of an application that represents a process running on the cluster.

It is a group of cluster that has shared network due to which the pod contents are always co-scheduled.



Uses of Pod

A few uses of pods:

- Loads files and data
- Manages cache
- Logs backup system
- Hosts integrated application
- Acts as a proxy, bride, or adapter
- Manages the data change, log trails
- Acts as a publisher

Assisted Practice: Pod Creation

Duration: 40 min.

Problem Statement:

Create the following:

- Pods using the YAML file
- Pods using the command kubectl
- Pods on specific namespace

ReplicaSets

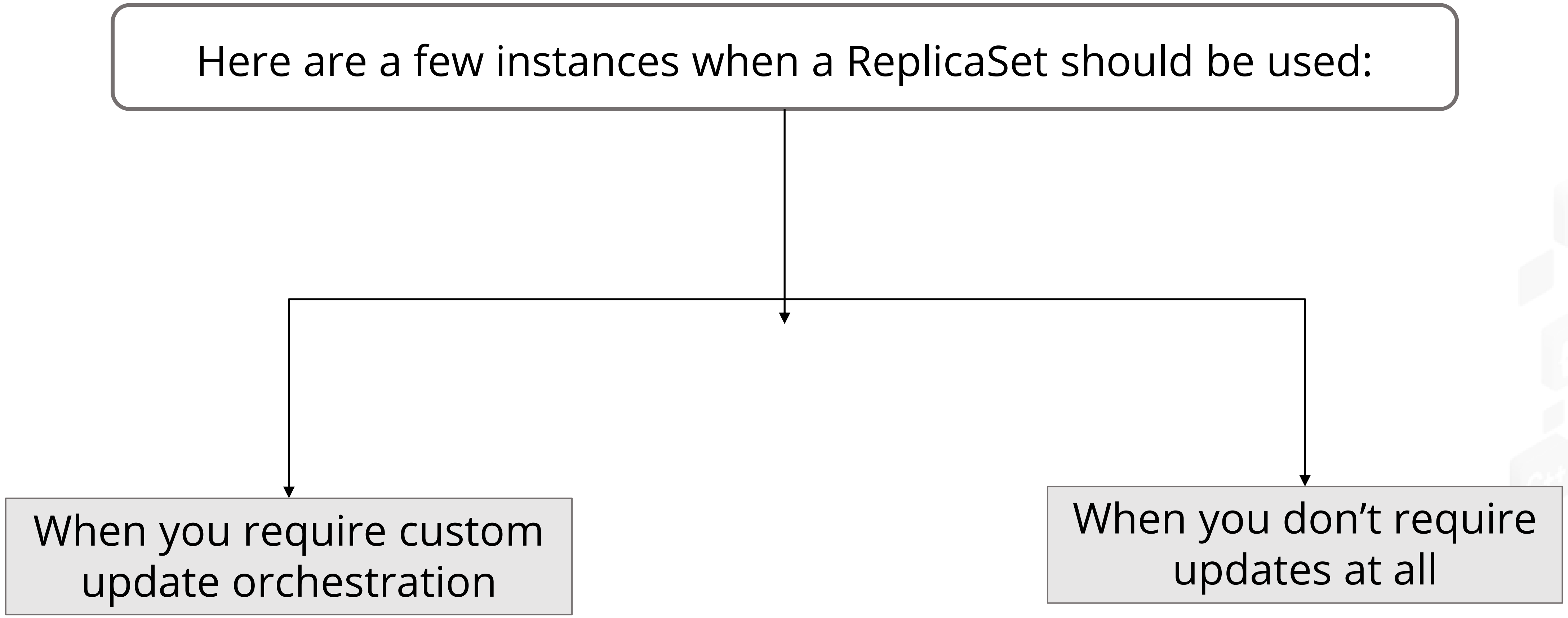
What Is ReplicaSet?

It is a method used to maintain a set of replica pods running and to identify the identical pods.

The ReplicaSet uses the selector to identify the pods running and based on the result, it creates or deletes the pods.

The ReplicaSet acquires the pod if the pod does not have an OwnerReference and matches the selector of ReplicaSet.

When to Use ReplicaSet



Here are a few instances when a ReplicaSet should be used:

When you require custom update orchestration

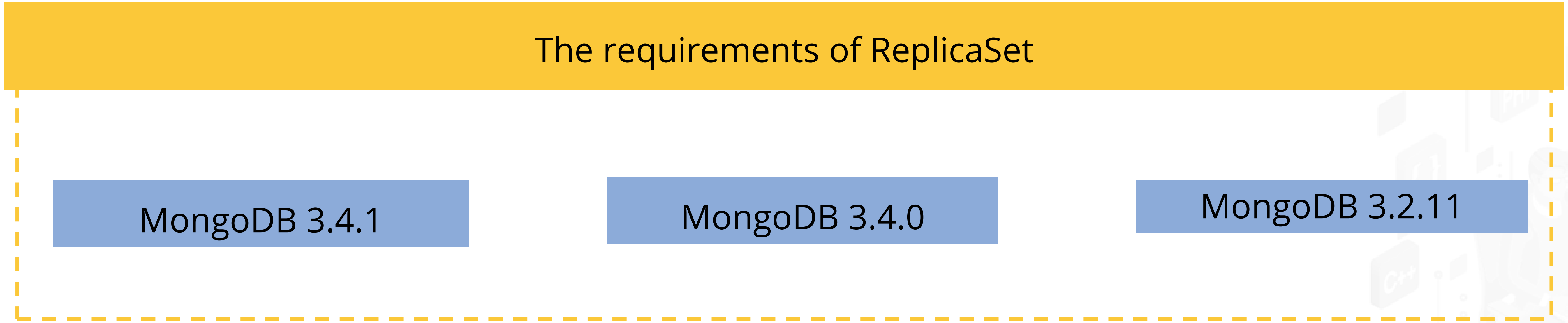
When you don’t require updates at all

Example of ReplicaSet

```
controllers/frontend.yaml

apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: frontend
  labels:
    app: sample_myapp
    tier: frontend
spec:
  # modify replicas according to your case
  replicas: 2
  selector:
    matchLabels:
      tier: frontend
  template:
    metadata:
      labels:
        tier: frontend
    spec:
      containers:
      - name: php-redis
        image: gcr.io/google_samples/gb-frontend:v3
```

ReplicaSet Requirements



The requirements of ReplicaSet

MongoDB 3.4.1

MongoDB 3.4.0

MongoDB 3.2.11

ReplicaControllers

It is a method that is responsible for ensuring that the pods are up and running.

A few characteristics of ReplicaController:

- Terminates the extra pods
- Automatically replaces the failed or terminated pods
- Supervises multiple pods at multiple nodes

Example of ReplicaControllers

controllers/replication.yaml

```
apiVersion: v1
kind: ReplicationController
metadata:
  name: nginx
spec:
  replicas: 2
  selector:
    app: nginx
  template:
    metadata:
      name: nginx
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx
        ports:
        - containerPort: 80
```

Assisted Practice: ReplicaSet Creation

Duration: 40 min.

Problem Statement:

You are given a project to implement the following:

- Create a ReplicaSet using the YAML file.
- Create a ReplicaSet using the command kubectl.

FULL STACK

Deployments

What Is Deployment?

A deployment is a controller that provides updates for pods and ReplicaSets.

The main role of deployment is to change the actual state to the desired state as specified.

Operations that can be performed on deployments:

- Creating a deployment
- Updating a deployment
- Rolling back a deployment
- Scaling a deployment
- Pausing and resuming a deployment
- Cleaning up deployment policy

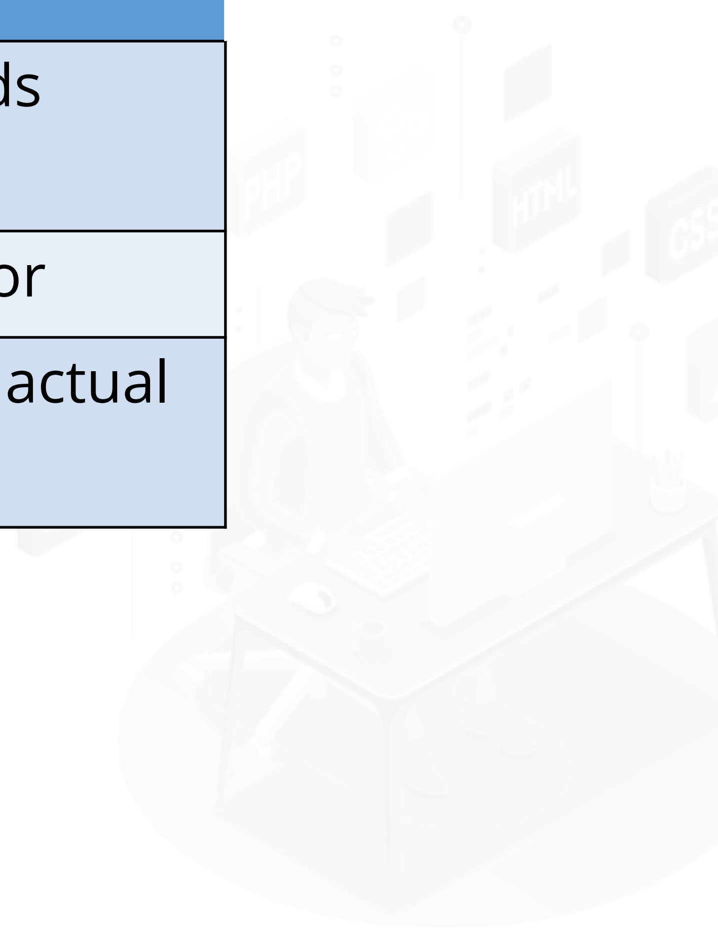
Uses of Deployment

Deployment can be used for the following purposes:

- To roll out a ReplicaSet
- To declare the new state of pods
- To roll back to an earlier deployment revision
- To facilitate more load
- To apply multiple fixes to pods
- To indicate that a rollout is stuck
- To clean up older ReplicaSets

Difference Between ReplicaSet and Deployments

ReplicaSet	Deployment
It maintains a set of replica pods running	It provides updates for pods and ReplicaSets
It uses set-based selector	It uses equity-based selector
It uses a selector to identify if the pods are running	It uses a selector to change the actual state to the desired state



Assisted Practice: Deployment Creation

Duration: 40 min.

Problem Statement:

Create the following:

- Deployment using the YAML file
- Deployment using the command kubectl

FULL STACK

Services and Service Cluster IP

What Is a Service?

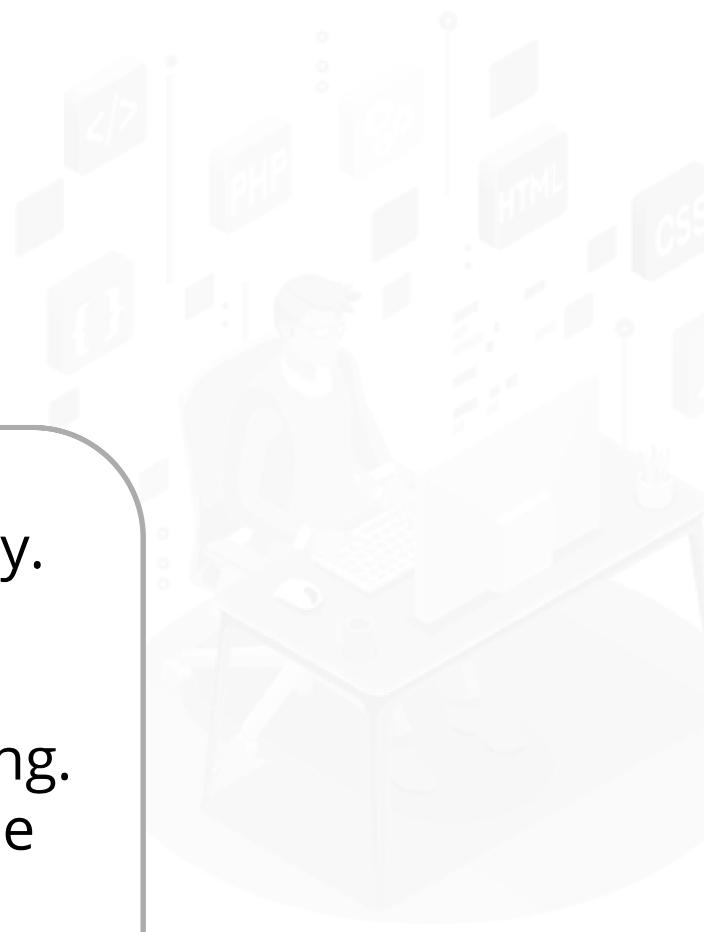
A service is a way of exposing an application that runs on a set of pods as a network service.

To do so, pods are given an IP address and a DNS name, through which they are being load balanced.

```
//Defining a service

apiVersion: v1
kind: Service
metadata:
  name: my-service
spec:
  selector:
    app: MyApp
  ports:
  - protocol: TCP
    port: 80
    targetPort: 9376
```

Service Resource



A service defines a set of pods that accesses other pods using a policy. It is also called micro-service.

This service determines the pods using a selector. It enables decoupling. **For example:** In an application containing backend, the pods keep the track of backend, whereas the front-end is unaware about which backend is being used.

Access Types in Service

Here are the access types in services:

- **ClusterIP**: The packets sent are never NAT’d
- **NodePort**: The packets sent are NAT’d by default
- **LoadBalancer**: The packets sent are NAT’d by default

Assisted Practice: Service Creation

Duration: 40 min.

Problem Statement:

Create the following:

- A service using the YAML file
- A service using the command kubectl

Key Takeaways

You are now able to:

- Install etcd
- Install kubelet in worker nodes using kubeadm method
- Create pods using the YAML file and kubectl command and on specific namespace
- Create ReplicaSet using the YAML file and kubectl command
- Create deployment using the YAML file and kubectl command
- Create service using the YAML file and kubectl command

Knowledge Check

Knowledge Check

1

What is the network requirement for large etcd cluster?

a. 1GbE

b. 5GbE

c. 10GbE

d. 15GbE

Knowledge Check

1

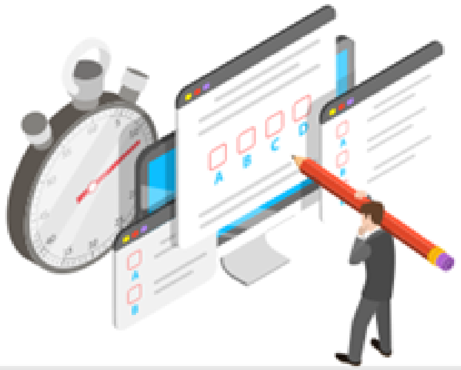
What is the network requirement for large etcd cluster?

a. 1GbE

b. 5GbE

c. 10GbE

d. 15GbE



The correct answer is c

10GbE network is required for large etcd cluster.

Knowledge Check

2

Which of the following statements is NOT true about kube-api?

a. It implements a RESTful API

b. It can be connected to etcd

c. It cannot be connected to etcd

d. It is stateless

Knowledge Check

2

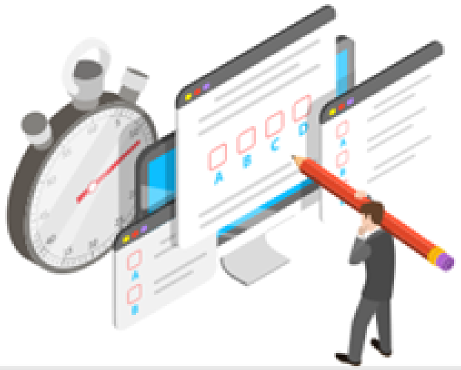
Which of the following statements is NOT true about kube-api?

a. It implements a RESTful API

b. It can be connected to etcd

c. It cannot be connected to etcd

d. It is stateless



The correct answer is c

Kube-api implements a RESTful API, is stateless, and can be connected to etcd.

Knowledge Check

3

When the nodes crash down, which of the following controllers is responsible for checking and taking actions on them?

a. Replica controller

b. Node controller

c. Service controller

d. Token controller

Knowledge Check

3

When the nodes crash down, which of the following controllers is responsible for checking and taking actions on them?

a. Replica controller

b. Node controller

c. Service controller

d. Token controller

The correct answer is b

When the nodes crash down, the node controller is responsible for checking and taking actions on them.

Knowledge Check 4	Which of the following operations of scheduler is responsible for finding the set of nodes which will be suitable to schedule a pod?

a. Filtering

b. Scoring

c. Sourcing

d. Searching

Knowledge Check

4

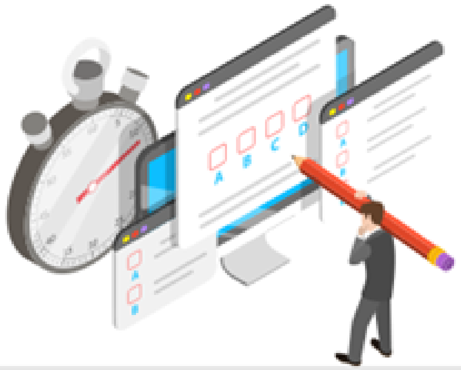
Which of the following operations of scheduler is responsible for finding the set of nodes which will be suitable to schedule a pod?

a. Filtering

b. Scoring

c. Sourcing

d. Searching



The correct answer is a

Filtering is responsible for finding the set of nodes which will be suitable to schedule a pod.

Knowledge Check

5

Which of the following statements is false about kubelet?

a. Authenticates the cluster

b. Manages the cluster created by Kubernetes

c. Manages the cluster not created by Kubernetes

d. Maintains the state of work

Knowledge Check

5

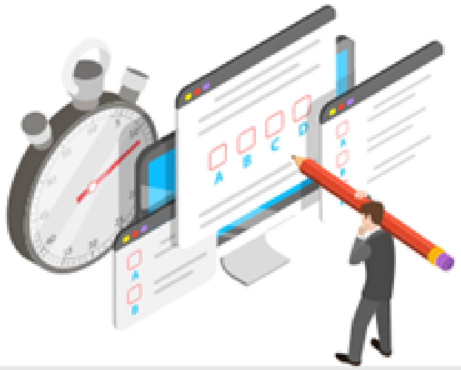
Which of the following statements is false about kubelet?

a. Authenticates the cluster

b. Manages the cluster created by Kubernetes

c. Manages the cluster not created by Kubernetes

d. Maintains the state of work



The correct answer is c

Kubelet does not manage the cluster that is not created by Kubernetes.

Knowledge Check

6

Which of the following options of kube-proxy specifies the IP address for the proxy server to serve on?

a. --azure-container-registry-config string

b. --bind-address 0.0.0.0

c. --cleanup-ipvs

d. --cluster-cidr string

Knowledge Check

6

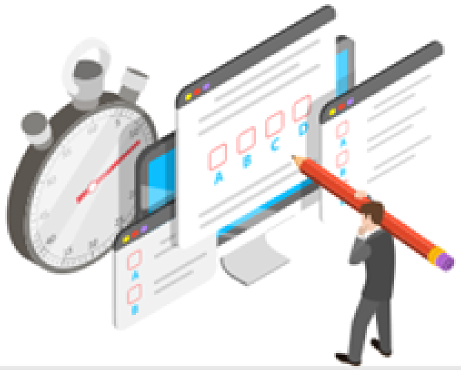
Which of the following options of kube-proxy specifies the IP address for the proxy server to serve on?

a. --azure-container-registry-config string

b. --bind-address 0.0.0.0

c. --cleanup-ipvs

d. --cluster-cidr string



The correct answer is b

--bind-address 0.0.0.0 option of kube-proxy specifies the IP address for the proxy server to serve on.

Knowledge Check

7

Which of the following is NOT true about pods?

a. Load files and data

b. Act as a proxy, bride, or adapter

c. Manage cache, data change, and log trails

d. Host non-integrated applications

Knowledge Check

7

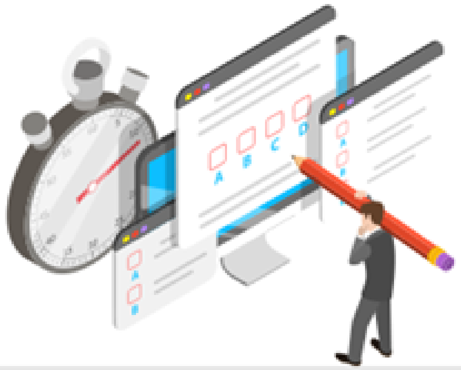
Which of the following is NOT true about pods?

a. Load files and data

b. Act as a proxy, bride, or adapter

c. Manage cache, data change, and log trails

d. Host non-integrated applications



The correct answer is d

Pods host integrated applications.

Knowledge Check

8

Which of the following is NOT true about ReplicaController?

a. Stores the extra pods

b. Automatically replaces the failed or terminated pods

c. Supervises multiple pods at multiple nodes

d. Maintains the number of pods for every object

Knowledge Check

8

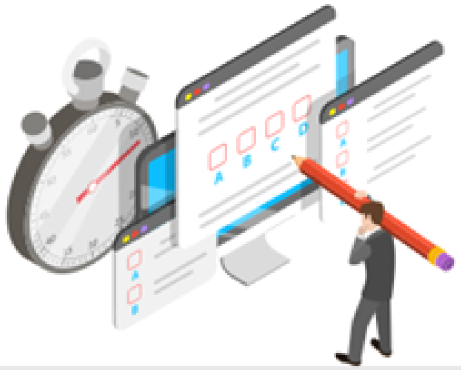
Which of the following is NOT true about ReplicaController?

a. Stores the extra pods

b. Automatically replaces the failed or terminated pods

c. Supervises multiple pods at multiple nodes

d. Maintains the number of pods for every object



The correct answer is a

ReplicaController terminates the extra pods.

Knowledge Check

9

Which of the following is a use case of deployment?

a. Rolling out a ReplicaSet

b. Declaring the new state of pods

c. Applying multiple fixes to pods

d. All of the above

Knowledge Check

9

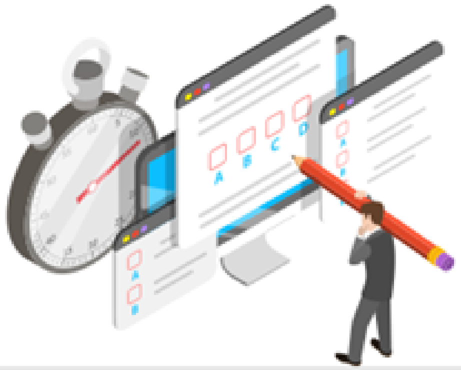
Which of the following is a use case of deployment?

a. Rolling out a ReplicaSet

b. Declaring the new state of pods

c. Applying multiple fixes to pods

d. All of the above



The correct answer is d

Rolling out a ReplicaSet, declaring the new state of pods, and applying multiple fixes to pods are the use cases of deployment.

Knowledge Check

10

In which of the access types are the packets sent to cluster never NAT'd?

a. ClusterIP

b. NodePort

c. LoadBalancer

d. ClusterPort

Knowledge Check

10

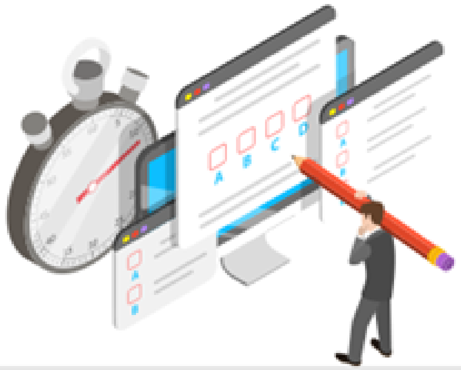
In which of the access types are the packets sent to cluster never NAT'd?

a. ClusterIP

b. NodePort

c. LoadBalancer

d. ClusterPort



The correct answer is a

In ClusterIP, the packets sent to cluster are never NAT'd.

Lesson-End Project

Problem Statement: A cyber security company whose product development is working on Real time Threat Detection and Incident Reporting has several analysts who do the analysis on a daily basis and report the incidents. The incidents that get generated are numerous. Being a Kubernetes expert, how would you automate the process of allocating the alerts to an analyst periodically?

Objective: Use Kubernetes Scheduler to automate the alert allocation for an analyst on a regular interval of time.

LESSON-END PROJECT