

FULL STACK



Certified Kubernetes Administrator

FULL STACK

Kubernetes: Cluster Maintenance



Learning Objectives

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

- ➊ Define the nodes of Kubernetes
- ➋ Work with kube-api server
- ➌ Determine how to drain a node
- ➍ Work with kubelet, kube-proxy, and kubectl
- ➎ Determine how to backup resource configs
- ➏ Determine how to backup and restore etcd



FULL STACK

OS Upgrades

Node Maintenance

Performing node maintenance (kernel upgrade, libc upgrade and hardware repair) can be done by rebooting the node.

Before performing maintenance on the node, run the **kubectl drain nodename** command to safely evict all the pods.

To make the node schedulable again, use the **kubectl uncordon nodename** command.

Node Maintenance

Characteristics of node maintenance:

- The kubelet restarts if the rebooting of node is done to perform node maintenance
- In case of replica controller, a new copy of pods must be started on a different node as upgrades can be done without special coordination
- In case of pods with no replica set, a new copy of node must be added which is not a part of service, and the clients must be redirected to it.

Drain Node

Draining a node means safely removing or evicting all the pods from the node before performing any maintenance on it.

kubectl drain <node name> command can be used to evict all the pods from the node.

kubectl drain ignores the pods that cannot be killed and once it returns successful value, all the pods can be powered down.

Cordon Node

Cordoning a node refers to marking a node as unaccessible or unschedulable. This prevents any new node from being scheduled to that node.

kubectI cordon <node name> command can be used to cordon or unschedule a node.

Unscheduling a node does not affect the existing pods of the node and is an important step before rebooting a node.

Uncordon Node

Un-cordoning a node refers to marking a node schedulable. This tells Kubernetes that the node can resume scheduling new pods.

kubectl uncordon <node name> command can be used to mark a node schedulable.

During maintenance operation, when a node is left in the cluster, **uncordon** command can be used to indicate that the node may schedule new pods later.

Drain Vs. Cordon Node

| Drain Node | Cordon Node |
|--|---|
| Used to evict or delete a node | Used to unschedule a node |
| Performed before the maintenance operation | Performed before rebooting a node |
| Powers down all the pods | Unschedules all the pods |
| Drains all the existing pods of the node | Does not affect the existing pods of the node |

Cordon Node



Problem Statement: You are given a project to demonstrate the workflow of cordon node.

ASSISTED PRACTICE

Uncordon Node



Problem Statement: You are given a project to demonstrate the workflow of uncordon node.

ASSISTED PRACTICE

Draining a Node



Problem Statement: You are given a project to demonstrate the workflow of draining a node.

ASSISTED PRACTICE

FULL STACK

Kubernetes Software Versions

Kubernetes Software Versions

| The versions of Kubernetes: |
|----------------------------------|
| ➤ v1.15 – Current version |
| ➤ v1.14 |
| ➤ v1.13 |
| ➤ v1.12 |
| ➤ v1.11 |

| The notable features of Kubernetes version v1.15: |
|---|
| ➤ v1.15.0-rc.1 |
| ➤ V1.15.0-beta-2 |
| ➤ V1.15.0-beta-1 |
| ➤ v1.15.0-alpha.3 |
| ➤ v1.15.0-alpha.2 |
| ➤ v1.15.0-alpha.1 |

FULL STACK

Cluster Upgrade Process

Working of kube-api Server



Problem Statement: You are given a project to demonstrate the workflow of kube-api server.

ASSISTED PRACTICE

Controller Manager

Controller manager is used to implant the control loops that regulate the state of the system.

The controller examines the shared state through apiserver and makes the changes for it to move to the desired state.

Kube-controller-manager [flags] command is used to embed the control loops.

Kube-scheduler

It is a workload specific function that impacts the performance, capacity, and availability.

Kube-scheduler [flags] command is used to expose the workload-specific requirements.

Factors that scheduler needs to consider:

- Individual and collective resource requirement
- Quality of service requirement
- Hardware and software policy constraints
- Affinity and anti-affinity specifications
- Data locality
- Inter-workload interference
- Deadline

Working with Controller Manager and Kube-Scheduler



Problem Statement: You are given a project to demonstrate the workflow of controller manager and kube-scheduler.

ASSISTED PRACTICE

Working of Kubelet, Kube-proxy, and Kubectl



Problem Statement: You are given a project to demonstrate the workflow of kubelet, kube-proxy, and kubectl.

ASSISTED PRACTICE

Master and Worker Upgrade

Command to upgrade kubeadm on master node:

```
kubeadm upgrade plan
```

Commands to upgrade kubectl on master node:

```
apt-mark unhold kubectl && \
```

```
apt-get update && apt-get install -y kubectl=1.13.x-00 && \
```

```
apt-mark hold kubectl
```

Master and Worker Upgrade

Command to upgrade kubeadm on worker node:

```
kubeadm upgrade node config --kubelet-version v1.13.x
```

Commands to upgrade kubectl on worker node:

```
# replace x in 1.13.x-00 with the latest patch version  
apt-mark unhold kubelet kubeadm  
apt-get update  
apt-get install -y kubelet=1.13.x-00 kubeadm=1.13.x-00  
apt-mark hold kubelet kubeadm
```


FULL STACK

Backup and Restore Methods

Backup Resource Configs



Problem Statement: You are given a project to demonstrate the backing up of the resource configs.

ASSISTED PRACTICE

Backup and Restore etcd



Problem Statement: You are given a project to demonstrate the backing up and restoring of etcd.

ASSISTED PRACTICE

Key Takeaways

You are now able to:

- 🕒 Define the nodes of Kubernetes
- 🕒 Work with kube-api server
- 🕒 Determine how to drain a node
- 🕒 Work with kubelet, kube-proxy, and kubectl
- 🕒 Determine how to backup resource configs
- 🕒 Determine how to backup and restore etcd



FULL STACK



Knowledge Check

Knowledge Check

1

Which of the following commands is used to mark a node schedulable?

- a. `kubectl drain`
- b. `kubectl cordon`
- c. `kubectl uncordon`
- d. `kubectl schedule`



Knowledge Check

1

Which of the following commands is used to mark a node schedulable?

- a. kubectl drain
- b. kubectl cordon
- c. kubectl uncordon
- d. kubectl schedule



The correct answer is **c**

kubectl uncordon command is used to mark a node schedulable.

Knowledge Check

2

Which of the following commands kills all the pods of a node?

- a. `kubectl drain`
- b. `kubectl cordon`
- c. `kubectl uncordon`
- d. `kubectl schedule`



Knowledge Check

2

Which of the following commands kills all the pods of a node?

- a. `kubectl drain`
- b. `kubectl cordon`
- c. `kubectl uncordon`
- d. `kubectl schedule`



The correct answer is **a**

Kubectl drain command kills all the pods of the node.

Knowledge Check

3

_____ command is used to expose the workload specific requirements.

- a. Kube-controller-manager [flags]
- b. Kube-scheduler [flags]
- c. Kubectl uncordon <node name>
- d. Kubectl drain <node name>



Knowledge Check

3

_____ command is used to expose the workload specific requirements.

- a. Kube-controller-manager [flags]
- b. Kube-scheduler [flags]
- c. Kubectl uncordon <node name>
- d. Kubectl drain <node name>



The correct answer is **b**

Kube-scheduler [flags] command is used to expose the workload specific requirements.

Knowledge Check

4

Which of the following components is used to embed the control loops?

- a. Kube-Scheduler
- b. Controller Manager
- c. Kube-Proxy
- d. Kubelet



Knowledge
Check

4

Which of the following components is used to embed the control loops?

- a. Kube-Scheduler
- b. Controller Manager
- c. Kube-Proxy
- d. Kubelet



The correct answer is **b**

Controller manager is used to embed the control loops.

**Knowledge
Check**

5

Which of the following is a network that runs on each node?

- a. Kube-apiserver
- b. Kubelet
- c. Kube-proxy
- d. Kube-scheduler



Knowledge
Check

5

Which of the following is a network that runs on each node?

- a. Kube-apiserver
- b. Kubelet
- c. Kube-proxy
- d. Kube-scheduler



The correct answer is **c**

Kube-proxy is a network that runs on each node.



Problem Statement: In agile methodology, mostly companies use the CI/CD (Continuous Integration and Continuous Deployment) approach. Daily deployment and update of a new build are time-consuming processes and may sometimes lead to a delay.

Considering this scenario, how can you use Kubernetes for deployments/updates that take place with zero downtime?

Objective: Ensure rolling updates using Kubernetes take place with zero downtime.