

[Menu](#)[Browse Library](#)

Training Library / Backup, Restore, and Upgrade a Kubernetes Cluster

Upgrading Kubernetes Clusters with kubeadm

40m 46s left

Open Cloud Environment

100%

Setup completed
Average setup time: 2m 31s

Open Code Environment

100%

Setup completed

Credentials

Account ID ⓘ

451412129 [Copy](#)

Username ⓘ

student [Copy](#)

Password ⓘ

Ca1_HcGbV [Copy](#)

Region ⓘ

US West 2 [Copy](#)

PEM ⓘ

PPK ⓘ

[Download](#) [Download](#)

Bastion Public Ip ⓘ

35.92.49. [Copy](#)

Cluster SSH ⓘ

ssh ubuntu [Copy](#)

Introduction

kubeadm supports upgrading Kubernetes clusters. In this lab step, you will be upgrading Kubernetes from version 1.28.1 to version 1.28.2. Although upgrading is supported, you should always take care to understand any changes between releases by [reading the release notes](#) and how they could impact your workloads. You should always backup important data before upgrading, and test upgrades before deploying them to production.

The upgrade process follows the general procedure of:

1. Upgrading the Kubernetes control plane with kubeadm (Kubernetes components and add-ons excluding the CNI)
2. Manually upgrading the CNI network plugin, if applicable
3. Upgrading the Kubernetes packages (kubect l, kubeadm, kubelet) on the control-plane and worker nodes
4. Upgrading the kubelet config on worker nodes with kubeadm

Instructions

1. To begin the upgrade, kubeadm needs to be updated to 1.28.2 :

[Copy code](#)

```
1 sudo apt-get update
2 sudo apt-get install -y --allow-change-held-packages kubeadm=1.2
```

2. Generate an upgrade plan for upgrading Kubernetes to version 1.28.2:

[Copy code](#)

```
1 sudo kubeadm upgrade plan 1.28.2
```

Support

[Skip to content](#)

Press [option](#) + [Q](#) to open this menu



Lab Steps

- 1 Connecting to the Kubernetes Cluster
- 2 Backing Up and Restoring Kubernetes Clusters
- 3 Upgrading Kubernetes Clusters with kubeadm

? Need help? Contact our support team

```
[upgrade/versions] kubeadm version: v1.28.2
[upgrade/versions] Target version: 1.28.2
[upgrade/versions] Latest version in the v1.28 series: 1.28.2

Components that must be upgraded manually after you have upgraded the control plane with 'kubeadm upgrade apply':
COMPONENT      CURRENT    TARGET
kubelet         3 x v1.28.1  1.28.2

Upgrade to the latest version in the v1.28 series:

COMPONENT      CURRENT    TARGET
kube-apiserver  v1.28.1    1.28.2
kube-controller-manager v1.28.1    1.28.2
kube-scheduler  v1.28.1    1.28.2
kube-proxy      v1.28.1    1.28.2
CoreDNS         v1.10.1    v1.10.1
etcd            3.5.9-0     3.5.9-0

You can now apply the upgrade by executing the following command:

    kubeadm upgrade apply 1.28.2

The table below shows the current state of component configs as understood by this version of kubeadm.
Configs that have a "yes" mark in the "MANUAL UPGRADE REQUIRED" column require manual config upgrade or
resetting to kubeadm defaults before a successful upgrade can be performed. The version to manually
upgrade to is denoted in the "PREFERRED VERSION" column.

API GROUP      CURRENT VERSION  PREFERRED VERSION  MANUAL UPGRADE REQUIRED
kubeproxy.config.k8s.io  v1alpha1        v1alpha1           no
kubelet.config.k8s.io    v1beta1         v1beta1            no
```

As the output explains, several checks are performed, and the requirements for upgrading the cluster are first verified. A reminder that you need to manually upgrade the kubelet on each node in the cluster is then displayed. Future versions may remove this manual step. Finally, a summary of the planned version changes for all the cluster components (**COMPONENT**) is presented.

3. Apply the upgrade plan by issuing the following command and entering y when prompted:

Copy code

```
1 | sudo kubeadm upgrade apply 1.28.2
```

kubeadm begins upgrading the cluster components on the control-plane node. Read through the output to understand what steps are being performed. It takes approximately four minutes to complete.

Enter y and press *Enter* to proceed:

```
[upgrade/version] You have chosen to change the cluster version to "v1.28.2"
[upgrade/versions] Cluster version: v1.28.1
[upgrade/versions] kubeadm version: v1.28.2
[upgrade] Are you sure you want to proceed? [y/N]:
```

You will see the following success message to know everything went as expected:

```
[upgrade/successful] SUCCESS! Your cluster was upgraded to "v1.28.2". Enjoy!
```

Note: If the upgrade procedure times out, you can safely try again until it

Skip to content

Press + to open this menu

to run the command



4. Prepare to upgrade the control-plane node's kubelet by draining the node:

[Copy code](#)

```
1 # Get control-plane node name
2 control_plane=$(kubectl get nodes | grep control-plane | cut -d'
3 # Drain the control-plane node
4 kubectl drain $control_plane --ignore-daemonsets --delete-emptyc
```

node/ip-10-0-0-100.us-west-2.compute.internal drained

The `--delete-emptydir-data` is used to drain nodes that contain local storage. In this lab, we know there isn't any important data on them so we can safely perform this. However, it is important to evaluate and restore data if needed prior to using this in a production environment.

The `--disable-eviction` is used to bypass checking [PodDisruptionBudgets](#). This should be used with caution and will cause downtime.

5. Upgrade the kubelet, and kubectl apt packages:

[Copy code](#)

```
1 sudo apt-get update
2 sudo apt-get upgrade -y --allow-change-held-packages \
3     kubelet=1.28.2-00 kubectl=1.28.2-00
```

The upgrade may take a few minutes to complete.

Click *Enter* for any **Package configuration** pop-ups:

Package configuration

6. Uncordon the control-plane to allow pods to be scheduled on it now that it has been upgraded:

[Copy code](#)

```
1 kubectl uncordon $control_plane
```

7. Get the node information to confirm that the version of the control-plane is 1.28.2.



```
ip-10-0-0-10.us-west-2.compute.internal Ready <none> 35m v1.28.1
ip-10-0-0-100.us-west-2.compute.internal Ready control-plane 26d v1.28.2
ip-10-0-0-11.us-west-2.compute.internal Ready <none> 34m v1.28.1
```

8. Drain the worker node 1 to prepare it for upgrading:

[Copy code](#)

```
1 # Get the worker's name
2 worker1=$(kubectl get nodes | grep 10.u | cut -d' ' -f1)
3 # Drain the worker node
4 kubectl drain $worker1 --ignore-daemonsets --delete-emptydir-dat
```

```
node/ip-10-0-0-10.us-west-2.compute.internal cordoned
Warning: Ignoring DaemonSet-managed Pods: kube-system/calico-node-jxc2g, kube-system/ebc-csi-node-kcg7h, kube-system/kube-proxy-xvcr5
evicting pod kube-system/kube-proxy-xvcr5
evicting pod kube-system/ebc-csi-controller-75b7b645cc-vnlls
evicting pod kube-system/calico-kube-controllers-f88756749-tw9kt
evicting pod kube-system/coredns-5558847f94-c4knd
evicting pod kube-system/metrics-server-77b976bb4b-2kxmh
evicting pod kube-system/ebc-csi-controller-75b7b645cc-rtjt6
evicting pod kube-system/kube-proxy-xvcr5
error when evicting pod/ebc-csi-controller-75b7b645cc-vnlls -n "kube-system" (will retry after 5s): Cannot evict pod as it would violate the pod's disruption budget.
pod/metrics-server-77b976bb4b-2kxmh evicted
pod/kube-system/kube-proxy-xvcr5 evicted
pod/dashboard-metrics-scraper-77d78b7997-jjfb4 evicted
pod/calico-kube-controllers-f88756749-tw9kt evicted
evicting pod kube-system/ebc-csi-controller-75b7b645cc-vnlls
error when evicting pod/ebc-csi-controller-75b7b645cc-vnlls -n "kube-system" (will retry after 5s): Cannot evict pod as it would violate the pod's disruption budget.
pod/ebc-csi-controller-75b7b645cc-rtjt6 evicted
pod/coredns-5558847f94-c4knd evicted
evicting pod kube-system/ebc-csi-controller-75b7b645cc-vnlls
error when evicting pod/ebc-csi-controller-75b7b645cc-vnlls -n "kube-system" (will retry after 5s): Cannot evict pod as it would violate the pod's disruption budget.
evicting pod kube-system/ebc-csi-controller-75b7b645cc-vnlls
pod/ebc-csi-controller-75b7b645cc-vnlls evicted
node/ip-10-0-0-10.us-west-2.compute.internal drained
```

9. SSH into worker node 1:

[Copy code](#)

```
1 ssh worker1 -oStrictHostKeyChecking=no
```

10. Drain the node and upgrade the Kubernetes packages:

[Copy code](#)

```
1 sudo apt-get update
2 sudo apt-get upgrade -y --allow-change-held-packages \
3     kubelet=1.28.2-00 kubeadm=1.28.2-00 kubectl=1.28.2-00
```

Click **Enter** for any **Package configuration** pop-ups:

Package configuration

11. Restart the worker node's kubelet and return to the control-plane :

[Copy code](#)

[Copy code](#)

```
1 | kubectl uncordon $worker1
```

```
node/ip-10-0-0-10.us-west-2.compute.internal uncordoned
```

13. Confirm the worker node is ready and running version 1.28.2:

[Copy code](#)

```
1 | kubectl get nodes
```

NAME	STATUS	ROLES	AGE	VERSION
ip-10-0-0-10.us-west-2.compute.internal	Ready	<none>	37m	v1.28.2
ip-10-0-0-100.us-west-2.compute.internal	Ready	control-plane	26d	v1.28.2
ip-10-0-0-11.us-west-2.compute.internal	Ready	<none>	36m	v1.28.1

The upgrade process is now complete. You can create some pods if you want to further test whether the upgrade succeeded.

14. Drain the worker node 2 to prepare it for upgrading:

[Copy code](#)

```
1 | # Get the worker's name
2 | worker2=$(kubectl get nodes | grep 11.u | cut -d' ' -f1)
3 | # Drain the worker node
4 | kubectl drain $worker2 --ignore-daemonsets --delete-emptydir-dat
```

```
node/ip-10-0-0-10.us-west-2.compute.internal drained
```

15. SSH into worker node 2:

[Copy code](#)

```
1 | ssh worker2 -oStrictHostKeyChecking=no
```

16. Drain the node and upgrade the Kubernetes packages:

[Copy code](#)

```
1 | sudo apt-get update
2 | sudo apt-get upgrade -y --allow-change-held-packages \
3 |     kubelet=1.28.2-00 kubeadm=1.28.2-00 kubectl=1.28.2-00
```

17. Restart the worker node's kubelet and return to the control-plane node :

[Copy code](#)

```
1 sudo systemctl restart kubelet
2 # Return to control-plane node
3 exit
```

18. Uncordon the worker node 2:

[Copy code](#)

```
1 kubectl uncordon $worker2
```

node/ip-10-0-0-55 uncordoned

19. Confirm all nodes are running version 1.28.2:

[Copy code](#)

```
1 kubectl get nodes
```

NAME	STATUS	ROLES	AGE	VERSION
ip-10-0-0-10.us-west-2.compute.internal	Ready	<none>	51m	v1.28.2
ip-10-0-0-100.us-west-2.compute.internal	Ready	control-plane	26d	v1.28.2
ip-10-0-0-11.us-west-2.compute.internal	Ready,SchedulingDisabled	<none>	50m	v1.28.2

The upgrade process is now complete. You can create some pods if you want to further test the success of the upgrade.

Summary

In this lab step, you went through the process of upgrading the cluster using kubeadm.

VALIDATION CHECKS

1 Checks

[Start check](#)

Did you like
this step?

[✕ End Lab](#)[< Back](#)[Start check](#)

ABOUT US

[About Cloud Academy](#)[About QA](#)[About Circus Street](#)

COMMUNITY

[Join Discord Channel](#)

HELP

[Help Center](#)

Copyright © 2024 Cloud Academy Inc. All rights reserved.

[Terms and Conditions](#)[Privacy Policy](#)[Sitemap](#)[System Status](#)[Manage your cookies](#)