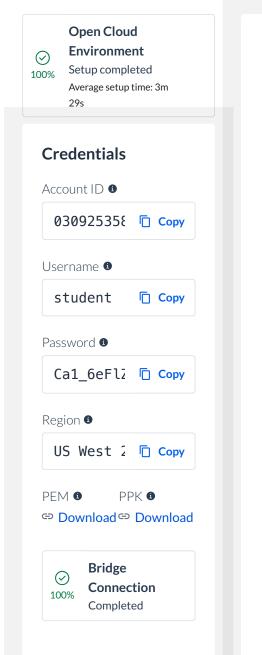




Training Library / Create and Manage a Kubernetes Cluster from Scratch

Installing kubeadm and Its Dependencies

52m 29s left



Introduction

kubeadm is a tool that allows you to easily create Kubernetes clusters that adhere to best practices. It can also perform a variety of cluster lifecycle functions, such as upgrading and downgrading the version of Kubernetes on nodes in the cluster. You will use kubeadm to create a Kubernetes cluster from scratch in this lab. Creating clusters with kubeadm is the recommended way for learning Kubernetes, creating small clusters, and as a piece of a more complex systems for more enterprise-ready clusters.

The Cloud Academy lab environment includes three EC2 instances running the Ubuntu 18.04 distribution of Linux. You will configure the instance named instance-a as a Kubernetes control-plane and the other instances as worker nodes in the cluster. In this lab step, you will install kubeadm and its dependencies, including containerd, on instance-a. The remaining nodes already been have kubeadminstalled to save you time.

Instructions

1. Enter the following command to update the system's apt package manager index and update packages required to install containerd:



- # Update the package index sudo apt-get update
- # Update packages required for HTTPS package repository access
- 4 sudo apt-get install -y apt-transport-https ca-certificates curl

2. Allow forwarding IPv4 by loading the br_netfilter module with the follow commands:

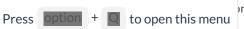


Lab Steps



























Instance Connect

- Installing kubeadm 3 and Its **Dependencies**
- Initializing the a Kubernetes Master Node
- Joining a Worker Node to the Kubernetes Cluster
- Need help? Contact our support team

3. Allow the Linux node's iptables to correctly view bridged traffic with the following commands:

```
Copy code
```

```
# sysctl params required by setup, params persist across reboots
   cat <<EOF | sudo tee /etc/sysctl.d/99-kubernetes-cri.conf</pre>
   net.bridge.bridge-nf-call-iptables = 1
   net.ipv4.ip forward
                                        = 1
   net.bridge.bridge-nf-call-ip6tables = 1
6
   E0F
   # Apply sysctl params without reboot
  sudo sysctl --system
```

4. Install containerd using the DEB package distributed by Docker with the following commands:



```
# Add Docker's official GPG key
   curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo c
   # Set up the repository
4
   echo \
     "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-
     $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/
  # Install containerd
8
  sudo apt-get update
   sudo apt-get install -y containerd.io=1.6.18-1
```

Note: This is only one way of installing containerd. Please refer here for the other options.

5. Configure the systemd cgroup driver with the following commands:



```
# Configure the systemd cgroup driver
sudo mkdir -p /etc/containerd
containerd config default | sudo tee /etc/containerd/config.toml
sudo sed -i 's/SystemdCgroup = false/SystemdCgroup = true/' /etc
sudo systemctl restart containerd
```

This is required to mitigate the instability of having two cgroup managers. Please refer here for futher explanation.

6. Install kubeadm, kubectl, and kubelet from the official Kubernetes package repository:



netes repository

```
# Add the Google Cloud packages GPG key
curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg |
# Add the Kubernetes release repository
sudo add-ant-renository "deh http://ant_kuhernetes.io/ kubernete
```

















7. Prevent automatic updates to the installed packages with the following command: Copy code 1 | sudo apt-mark hold kubelet kubeadm kubectl The version of all the packages is set to 1.28.1 for consistency in lab experiences, and so that you can perform a cluster upgrade in a later lab step. 8. Display the help page for kubeadm: Copy code 1 kubeadm Create a two-machine cluster with one control-plane no (which controls the cluster), and one worker node (where your workloads, like Pods and Deployments run). On the first machine: control-plane# kubeadm init On the second machine: worker# kubeadm join <arguments-returned-from-init> You can then repeat the second step on as many other machines as you like. any command

mmand in order to set up the Kubernetes control p

n any machine you wish to join an existing cluster

file utilities

best effort ter smoothly to a newer version with this command of kubeadm If true, adds the file directory to the header of the log messages help for kubeadm dd-dir-header

Read through the output to get a high-level overview of how a cluster is created and the commands that are available in kubeadm.

Summary

In this lab step, you installed kubeadm and its dependencies required for











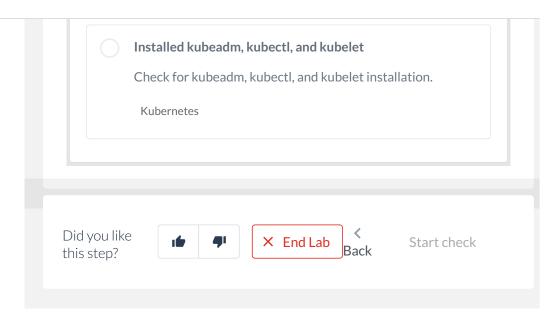






















ABOUT US

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