Kubernetes(Primary) Installation Guide

# Step 01 : Update the /etc/hosts File for Hostname Resolution

Take below command :

* sudo nano /etc/hosts
* cat /etc/hosts -> set 127.0.1.1 k8smaster
* ping -c 4 k8smaster

Updating the /etc/hosts file allows the system to resolve the hostname 'k8smaster' to its IP address. This ensures that the hostname can be pinged and resolved correctly.

# Step 02 : Install kubectl

To install kubectl, go to this URL : https://kubernetes.io/releases/download/

1. Select 'Linux': Choose the Linux option to get installation instructions specific to the Linux operating system.
2. Select 'Install using native package management': Follow the instructions to use your system’s package manager for the installation.

Run the following commands :

* sudo apt-get update
* sudo apt-get install -y apt-transport-https ca-certificates curl gnupg

These commands update your package list and install necessary packages for secure transport and certificate handling.

Add the Kubernetes apt repository key :

* curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.28/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg

This command downloads and adds the key for the Kubernetes repository to your system.

Set the correct permissions for the keyring :

* sudo chmod 644 /etc/apt/keyrings/kubernetes-apt-keyring.gpg

This ensures the keyring file is readable by the package manager.

Add the Kubernetes apt repository to your sources list :

* echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.28/deb/ /' | sudo tee /etc/apt/sources.list.d/kubernetes.list

This command adds the Kubernetes repository to your system’s sources list.

Update your package list:

* sudo apt-get update

Refreshes the package list to include the Kubernetes repository.

Install kubectl :

* sudo apt-get install -y kubectl

Installs kubectl, the command-line tool for interacting with your Kubernetes cluster.

Verify the installation :

* kubectl version

Checks and displays the installed version of kubectl.

# Step 03 : Install kubeadm, kubelet

To install kubeadm and kubelet, go to this URL :

1. https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/install-kubeadm/
2. Select 'Installing kubeadm (Kubernetes v1.28)': Follow the instructions specific to version 1.28 of Kubernetes.

Run the following commands :

* sudo apt-get update
* sudo apt-get install -y apt-transport-https ca-certificates curl gpg

These commands update your package list and install necessary packages for secure transport and certificate handling.

Add the Kubernetes apt repository key :

* curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.28/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg

This command downloads and adds the key for the Kubernetes repository to your system.

Add the Kubernetes apt repository to your sources list :

* echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.28/deb/ /' | sudo tee /etc/apt/sources.list.d/kubernetes.list

This command adds the Kubernetes repository to your system’s sources list.

Update your package list :

* sudo apt-get update

Refreshes the package list to include the Kubernetes repository.

Install kubelet and kubeadm:

* sudo apt-get install -y kubelet kubeadm

Installs kubelet, the Kubernetes node agent, and kubeadm, the tool for bootstrapping a Kubernetes cluster.

Mark packages to hold :

* sudo apt-mark hold kubelet kubeadm kubectl

Prevents these packages from being automatically updated.

Verify the installation :

* kubectl version
* kubelet --version
* kubeadm version

Checks and displays the installed versions of kubectl, kubelet, and kubeadm.

# Step 04 : Install Docker

Install Docker :

* sudo apt install docker.io

Installs Docker, a container runtime necessary for running containers in Kubernetes.

Remove containerd.io (If installed) :

* sudo apt remove containerd.io

Removes containerd.io if it is already installed to avoid conflicts.

Create the containerd directory :

* sudo mkdir /etc/containerd

Creates a directory for containerd configuration files.

Generate the default containerd configuration file :

* sudo sh -c 'containerd config default > /etc/containerd/config.toml'

Creates the default containerd configuration file.

Enable systemd as the cgroup driver :

* sudo sed -i 's/ SystemdCgroup = false/ SystemdCgroup = true/' /etc/containerd/config.toml

Modifies the containerd configuration to use systemd as the cgroup driver, which is recommended for Kubernetes.

Restart the containerd and kubelet services :

* sudo systemctl restart containerd.service
* sudo systemctl restart kubelet.service

Restarts the containerd and kubelet services to apply the configuration changes.

Enable the kubelet service :

* sudo systemctl enable kubelet.service

Enables the kubelet service to start automatically on system boot.

# Step 05 : Configure iptables

Enable bridged IPv4 traffic to iptables :

* echo 'net.bridge.bridge-nf-call-iptables=1' | sudo tee -a /etc/sysctl.conf
* sudo sysctl -p

Configures the system to pass bridged IPv4 traffic to iptables for processing. This allow IP forwarding and bridge network traffic.

# Step 06 : Initialize the Kubernetes Cluster

Pull necessary images :

* sudo kubeadm config images pull

Pulls the necessary container images for Kubernetes components.

Initialize the cluster :

* sudo kubeadm init

Bootstraps the Kubernetes control-plane node.

If the initialization fails, reset and re-initialize :

* sudo kubeadm reset
* sudo kubeadm init

Resets the cluster configuration and reinitializes it with a specified pod network CIDR.

# Step 06 : Post-Initialization Setup

Set up kubectl for the regular user :

* mkdir -p $HOME/.kube
* sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
* sudo chown $(id -u):$(id -g) $HOME/.kube/config

Copies the admin configuration file to the user’s home directory and sets the correct permissions.

For root users :

* export KUBECONFIG=/etc/kubernetes/admin.conf

Sets the KUBECONFIG environment variable to use the admin configuration file.

# Step 07 : Install Calico (CNI)

Download the Calico manifest :

* curl -O https://raw.githubusercontent.com/projectcalico/calico/v3.25.0/manifests/calico.yaml

This command downloads the Calico manifest file from GitHub, which contains the necessary configuration to deploy Calico as the Container Network Interface (CNI) plugin for your Kubernetes cluster.

Calico provides scalable networking and robust network security policies for Kubernetes, enabling efficient pod communication and enhancing cluster security.

Apply the Calico manifest :

* sudo kubectl apply -f calico.yaml

Deploys the Calico network plugin to the Kubernetes cluster.

# Step 08 : Verify the Cluster

Check the status of the nodes :

* kubectl get nodes

Displays the status of the nodes in the cluster.

Expected Output :

NAME STATUS ROLES AGE VERSION

k8smaster Ready control-plane 20s v1.28.12

Happy Kubernetes Setup