Kubernetes(Secondary) 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.

**Create a token in master node as below :**

ubuntu@k8smaster**:~$ sudo kubeadm token create --print-join-command**

**Output :**

kubeadm join 10.0.1.164:6443 --token ef8czb.us1o8xzeim1obiwc --discovery-token-ca-cert-hash sha256:364c7b82e5da4c8e83da7c512c7cba176a083f04e3feda53e3ef7368fedcd746

ubuntu@k8smaster:~$

**Run the token in worker node to join into master node as below**

ubuntu@k8sworker2:~$ sudo kubeadm join 10.0.1.164:6443 --token ef8czb.us1o8xzeim1obiwc --discovery-token-ca-cert-hash sha256:364c7b82e5da4c8e83da7c512c7cba176a083f04e3feda53e3ef7368fedcd746

Console Output :

[preflight] Running pre-flight checks

[preflight] Reading configuration from the cluster...

[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'

[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"

[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"

[kubelet-start] Starting the kubelet

[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap...

This node has joined the cluster:

\* Certificate signing request was sent to apiserver and a response was received.

\* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the control-plane to see this node join the cluster.

ubuntu@k8sworker2:~$

**After joined in worker node run the below command in “Master Node” :**

* kubectl get nodes

**Output :**

ubuntu@k8smaster:/etc/sample$ kubectl get nodes

NAME STATUS ROLES AGE VERSION

k8smaster Ready control-plane 18h v1.28.12

k8sworker Ready <none> 20s v1.28.12

ubuntu@k8sworker2:~$

Happy Kubernetes Setup