

The background is a dark blue gradient. On the left, there is a large, semi-transparent circular inset showing a detailed view of a circuit board. Overlaid on the top-left of this circle are two overlapping triangles: a blue one in front and a light green one behind it. In the top-right corner, there is a 3D perspective view of a circuit board's surface, showing numerous small components and traces.

Kubernetes Installation Kubedm

About Me



Hi there, my name is Bilal and I will Welcome you to DevOps boot camp! I am thrilled to have you join us for this exciting journey of learning and discovery.

In this boot camp, we will be exploring the principles and practices of DevOps, which is a set of methodologies and tools that aims to bridge the gap between software development and operations. DevOps is an increasingly important area in the field of software engineering, as it helps organizations to streamline their processes, improve their agility, and deliver better value to their customers.

By the end of this boot camp, you will have gained a comprehensive understanding of DevOps and its key concepts, as well as practical skills in areas such as infrastructure automation, continuous integration and delivery, monitoring and logging, and more. You will be equipped with the knowledge and tools to apply DevOps principles in your own work and contribute to the success of your organization.

I am always looking to connect with other professionals in the field, share ideas and insights, and stay up to date on the latest trends and developments. I welcome the opportunity to connect with you and explore ways in which we can collaborate and support each other.

Github : <https://github.com/BilalMaz/DevOps-Architect-BootCamp>

LinkedIn : <https://www.linkedin.com/in/bilalmazhar-cyber-security-consultant/>



Installation

Kubernetes can be install in three ways

- **MicroK8s** is a lightweight and easy-to-use Kubernetes distribution developed by Canonical, the company behind Ubuntu. It is designed to run on desktops, servers, and IoT devices, and provides a complete Kubernetes environment on a single node.
- **Minikube** is a tool that allows users to run a single-node Kubernetes cluster on their local machine. It is designed for developers and individuals who want to test and develop Kubernetes applications locally before deploying them to a production Kubernetes cluster
- **Kubeadm** is a tool for bootstrapping a Kubernetes cluster. It is designed for advanced users and administrators who want to set up a custom Kubernetes cluster from scratch. Kubeadm provides a set of commands for initializing and joining nodes to a Kubernetes cluster, and can be used to set up highly available and secure Kubernetes clusters.



Kubeadm installation

→ To setup a cluster we required “**1 master**” and “**1 worker**” nodes,

Master :

- 2 GB RAM
- 2 Cores of CPU

Worker :

- 1 GB RAM
- 1 Core of CPU



Pre-Installation Steps On Both Master & worker nodes

→ The following steps have to be executed on both the master and node machines. Let's call the master '**bilalmaster**' and node as '**bilalworker**'.

→ First, login as '**sudo**' user because the following set of commands need to be executed with 'sudo' permissions. Then, update your 'apt-get' repository.

```
$ sudo su
```

```
# apt-get update
```

-> Turn Off Swap Space : Next, we have to turn off the swap space because Kubernetes will start throwing random errors otherwise. After that you need to open the 'fstab' file and comment out the line which has mention of swap partition.

```
# swapoff -a
```

```
# nano /etc/fstab
```



- disable the swap
- Comment out the /swap file command
- to save Ctrl + o and then Ctrl + x

```
root@bilalmaster-virtual-machine: /home/bilalmaster
GNU nano 6.2 /etc/fstab *
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point> <type> <options> <dump> <pass>
# / was on /dev/sda3 during installation
UUID=aafa898a-8135-4e4a-b1eb-c8a84f706976 / ext4 errors=remount
# /boot/efi was on /dev/sda2 during installation
#UUID=26DA-9E66 /boot/efi vfat umask=0077 0 1
/swapfile none swap sw
/dev/fd0 /media/floppy0 auto rw,user,noauto,exec,utf8 0 0
```

[Recording a macro...]

^G Help	^O Write Out	^W Where Is	^K Cut	^T Execute	^C Location
^X Exit	^R Read File	^N Replace	^U Paste	^J Justify	^_ Go To Line

Update The Hostnames

→ To change the hostname of both machines, run the below command to open the file and subsequently rename the master machine to 'kmaster' and your node machine to 'knode'.

nano /etc/hostname

```
root@bilalmaster-virtual-n
GNU nano 6.2 /etc/hostname
bilalmaster
```

ifconfig

```
RX bytes:1967910 (1.9 MB) TX bytes:62801 (62.8 KB)
enp0s8 Link encap:Ethernet HWaddr 08:00:27:54:bf:cd
inet addr:192.168.56.101 Bcast:192.168.56.255 Mask:
inet6 addr: fe80::ab4b:e95e:9dd1:5d49/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:16 errors:0 dropped:0 overruns:0 frame:0
TX packets:90 errors:0 dropped:0 overruns:0 carrier:
collisions:0 txqueuelen:1000
RX bytes:2761 (2.7 KB) TX bytes:11699 (11.6 KB)
```

→ nano /etc/hosts

```
root@bilalmaster: /home/bilalmaster
GNU nano 6.2 /etc/hosts
27.0.0.1      localhost
27.0.1.1      bilalmaster-virtual-machine
92.168.60.148  bilalmaster
92.168.60.149  bilalworker
The following lines are desirable for IPv6 capable hosts
::1          ip6-localhost ip6-loopback
fe00::0      ip6-localnet
ff00::0      ip6-mcastprefix

root@bilalworker: /home/bilalworker
GNU nano 6.2 /etc/hosts
127.0.0.1      localhost
127.0.1.1      bilalworker-virtual-machine
192.168.60.149  bilalworker
192.168.60.148  bilalmaster
% The following lines are desirable for IPv6 capable hosts
::1          ip6-localhost ip6-loopback
fe00::0      ip6-localnet
```


Install OpenSSH-Server : Now we have to install openssh-server. Run the following command:

```
root@bilalmaster: /home/bilalmaster
root@bilalmaster:/home/bilalmaster# sudo apt-get install openssh-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer required:
  systemd-hwe-hwdb
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  ncurses-term openssh-client openssh-sftp-server ssh-import-id
```

Install Docker : Now we have to install Docker because Docker images will be used for managing the containers in the cluster. Run the

- # sudo su
- # apt-get update
- # apt-get install -y docker.io

```
root@bilalmaster: /home/bilalmaster
root@bilalmaster:/home/bilalmaster# apt-get install -y docker.io -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer required:
  systemd-hwe-hwdb
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
```

→ Next we have to install these 3 essential components for setting up Kubernetes environment: kubeadm, kubectl, and kubelet on master and worker nodes

Run the following commands before installing the Kubernetes environment :

- # apt-get update && apt-get install -y apt-transport-https curl

```
root@bilalmaster:/home/bilalmaster# apt-get update && apt-get install -y apt-transport-https curl
Hit:1 http://pk.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://pk.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:4 http://pk.archive.ubuntu.com/ubuntu jammy-backports InRelease [108 kB]
Fetched 337 kB in 2s (185 kB/s)
Reading package lists... Done
```

- # curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | apt-key add -

```
root@bilalmaster:/home/bilalmaster# curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | apt-key add -
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
OK
root@bilalmaster:/home/bilalmaster#
```

- # cat <<EOF >/etc/apt/sources.list.d/kubernetes.list
- deb http://apt.kubernetes.io/ kubernetes-xenial main
- EOF

```
root@bilalmaster: /home/bilalmaster

root@bilalmaster:/home/bilalmaster# cat <<EOF >/etc/apt/sources.list.d/kubernetes.list
deb http://apt.kubernetes.io/ kubernetes-xenial main
EOF
root@bilalmaster:/home/bilalmaster#
```

- # apt-get update

```
root@bilalmaster: /home/bilalmaster


root@bilalmaster:/home/bilalmaster# apt update
Hit:1 http://pk.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:3 http://pk.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:5 http://pk.archive.ubuntu.com/ubuntu jammy-backports InRelease [108 kB]
Get:4 https://packages.cloud.google.com/apt kubernetes-xenial InRelease [8,993 B]
Ign:6 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 Packages
Get:6 http://apt.kubernetes.io kubernetes-xenial/main amd64 Packages [64.5 kB]
88% [6 Packages 3,489 B/64.5 kB 5%]
```



Install kubeadm, Kubelet And Kubectl

→ Now it's time to install the 3 essential components. On master and worker nodes

- **Kubelet** is the lowest level component in Kubernetes. It's responsible for what's running on an individual machine.
- **Kubeadm** is used for administering the Kubernetes cluster.
- **Kubectl** is used for controlling the configurations on various nodes inside the cluster.

```
#  root@bilalmaster: /home/bilalmaster

root@bilalmaster:/home/bilalmaster# apt-get install -y kubelet kubeadm kubectl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer required:
  systemd-hwe-hwdb
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  conntrack cri-tools ebtables kubernetes-cni socat
The following NEW packages will be installed:
  conntrack cri-tools ebtables kubeadm kubectl kubelet kubernetes-cni socat
0 upgraded, 8 newly installed, 0 to remove and 316 not upgraded.
```

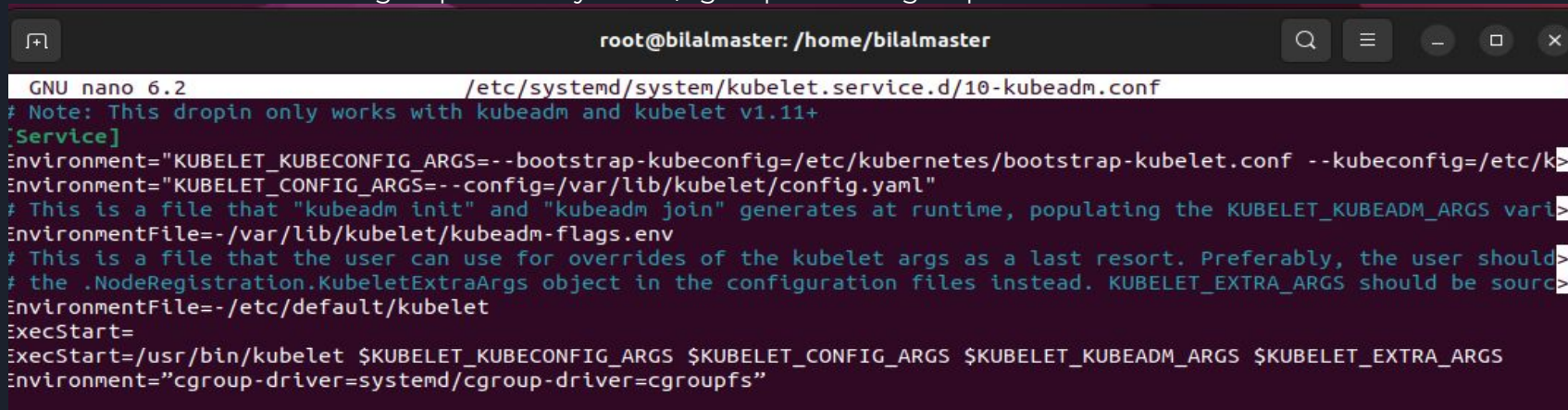
Updating Kubernetes Configuration

→ Next, we will change the configuration file of Kubernetes. Run the following command:

```
# nano /etc/systemd/system/kubelet.service.d/10-kubeadm.conf
```

→ This will open a text editor, enter the following line after the last "Environment Variable":

```
Environment="cgroup-driver=systemd/cgroup-driver=cgroupfs"
```

A terminal window titled 'root@bilalmaster: /home/bilalmaster' shows the nano 6.2 text editor editing the file '/etc/systemd/system/kubelet.service.d/10-kubeadm.conf'. The editor's status bar at the top indicates 'GNU nano 6.2' and the file path. The content of the file is displayed in a dark background with syntax highlighting. It includes a note about kubeadm and kubelet versions, a '[Service]' section, and several 'Environment=' lines. The last line shown is 'Environment="cgroup-driver=systemd/cgroup-driver=cgroupfs"', which is being edited. The terminal window has standard Linux window controls (search, menu, zoom, close) in the top right corner.

```
root@bilalmaster: /home/bilalmaster
GNU nano 6.2 /etc/systemd/system/kubelet.service.d/10-kubeadm.conf
# Note: This dropin only works with kubeadm and kubelet v1.11+
[Service]
Environment="KUBELET_KUBECONFIG_ARGS=--bootstrap-kubeconfig=/etc/kubernetes/bootstrap-kubelet.conf --kubeconfig=/etc/k
Environment="KUBELET_CONFIG_ARGS=--config=/var/lib/kubelet/config.yaml"
# This is a file that "kubeadm init" and "kubeadm join" generates at runtime, populating the KUBELET_KUBEADM_ARGS vari
EnvironmentFile=-/var/lib/kubelet/kubeadm-flags.env
# This is a file that the user can use for overrides of the kubelet args as a last resort. Preferably, the user should
# the .NodeRegistration.KubeletExtraArgs object in the configuration files instead. KUBELET_EXTRA_ARGS should be sourc
EnvironmentFile=-/etc/default/kubelet
ExecStart=
ExecStart=/usr/bin/kubelet $KUBELET_KUBECONFIG_ARGS $KUBELET_CONFIG_ARGS $KUBELET_KUBEADM_ARGS $KUBELET_EXTRA_ARGS
Environment="cgroup-driver=systemd/cgroup-driver=cgroupfs"
```



iLISTO!! YOU HAVE SUCCESSFULLY INSTALLED KUBERNETES ON BOTH THE MACHINES NOW!

Kubernetes Master VM (bilalmaster) setup

→ Note: These steps will only be executed on the master node (bilalmaster VM).

→ We will now start our Kubernetes cluster from the bilalmaster's machine. Run the following command:

```
root@bilalmaster: /home/bilalmaster

root@bilalmaster:/home/bilalmaster#
root@bilalmaster:/home/bilalmaster# kubeadm init --apiserver-advertise-address=192.168.60.148 --pod-network-cidr=192.168.0.0/16
[init] Using Kubernetes version: v1.26.3
[preflight] Running pre-flight checks
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action in beforehand using 'kubeadm config images pull'
```

\$ mkdir -p \$HOME/.kube

\$ sudo cp -i /etc/kubernetes/admin.conf \$HOME/.kube/config

\$ sudo chown \$(id -u):\$(id -g) \$HOME/.kube/config

```
root@bilalmaster: /home/bilalmaster

bilalmaster@bilalmaster:~$ sudo su
[sudo] password for bilalmaster:
root@bilalmaster:/home/bilalmaster#
root@bilalmaster:/home/bilalmaster# mkdir -p $HOME/.kube
root@bilalmaster:/home/bilalmaster# ^[[200~sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo: command not found
root@bilalmaster:/home/bilalmaster# cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
root@bilalmaster:/home/bilalmaster# sudo chown $(id -u):$(id -g) $HOME/.kube/config
root@bilalmaster:/home/bilalmaster#
```


→ To verify, if kubectl is working or not, run the following command:

```
$ kubectl get pods -o wide --all-namespaces
```

```
bilalmaster@bilalmaster: ~  
bilalmaster@bilalmaster:~$ kubectl get pods -o wide --all-namespaces  
NAMESPACE      NAME                                READY   STATUS    RESTARTS      AGE   IP              NODE      NOMINATED NODE   READINESS GATES  
kube-system     etcd-bilalmaster                   1/1     Running   23 (2m41s ago)  126m  192.168.60.150  bilalmaster   <none>           <none>  
kube-system     kube-apiserver-bilalmaster         1/1     Running   28 (3m7s ago)   128m  192.168.60.150  bilalmaster   <none>           <none>  
kube-system     kube-controller-manager-bilalmaster 0/1     CrashLoopBackOff 35 (2m41s ago)  126m  192.168.60.150  bilalmaster   <none>           <none>  
kube-system     kube-scheduler-bilalmaster         0/1     CrashLoopBackOff 34 (3m26s ago)  126m  192.168.60.150  bilalmaster   <none>           <none>  
bilalmaster@bilalmaster:~$
```

→ Installing a Pod network add-on

```
bilalmaster@bilalmaster:~$ systemctl enable --now kubelet  
cp -i bilalmaster@bilalmaster:~$ kubectl get pods  
No resources found in default namespace.  
bilalmaster@bilalmaster:~$ kubectl create -f https://raw.githubusercontent.com/projectcalico/calico/v3.25.1/manifests/tigera-operator  
namespace/tigera-operator created  
customresourcedefinition.apiextensions.k8s.io/bgpconfigurations.crd.projectcalico.org created  
customresourcedefinition.apiextensions.k8s.io/bgppeers.crd.projectcalico.org created  
customresourcedefinition.apiextensions.k8s.io/blockaffinities.crd.projectcalico.org created  
customresourcedefinition.apiextensions.k8s.io/caliconodestatuses.crd.projectcalico.org created  
customresourcedefinition.apiextensions.k8s.io/clusterinformations.crd.projectcalico.org created  
customresourcedefinition.apiextensions.k8s.io/felixconfigurations.crd.projectcalico.org created  
customresourcedefinition.apiextensions.k8s.io/globalnetworkpolicies.crd.projectcalico.org created  
customresourcedefinition.apiextensions.k8s.io/globalnetworksets.crd.projectcalico.org created  
customresourcedefinition.apiextensions.k8s.io/hostendpoints.crd.projectcalico.org created  
pod/192.168.60.150:6443 --token 91ax7u.r3lvu17zaoinehi \
```

```

tigera-operator      tigera-operator-5008450490-gz5jd      0/1      Pending      0      29s      <none>      <none>      <non
e>      <none>
bilalmaster@bilalmaster:/etc/kubernetes/manifests$ kubectl create -f https://raw.githubusercontent.com/projectcalico/calico/v3.25.1/manifests/
custom-resources.yaml
installation.operator.tigera.io/default created
apiserver.operator.tigera.io/default created
bilalmaster@bilalmaster:/etc/kubernetes/manifests$

```

Then you can join any number of worker nodes by running the following on each as root:

→ Install Calico by creating the necessary custom resource. For more information on configuration options available in this manifest

`kubectl create -f https://raw.githubusercontent.com/projectcalico/calico/v3.25.1/manifests/custom-resources.yaml`