

Integrating Jenkins ... Update

Search drafts...

- Static Website Hosting on AWS S3
- Deploying WordPress and MySQL
- Use NFS for Persistent Volumes
- Running Kubernetes locally on Linux
- Kubernetes Deployment
- kubeadm Installation.**
- CI/CD Pipeline | Parameters Input
- CI/CD Pipeline Scripts
- Dump MySQL Server DB File in RDS
- Installation of Jenkins on Docker
- Integrating Jenkins with GitHub
- Jenkins Installation in Ubuntu
- Create EC2 instance in AWS

DELETED ARTICLES
Go to the [dashboard](#) to view your deleted articles.

[New draft](#)

kubeadm Installation.

```
node1 [Running] - Oracle VM VirtualBox
Login incorrect
node1 login: generic
Password:
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-71-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

System information as of Tue May 9 07:58:35 AM UTC 2023

System load: 0.3447265625 Processes: 124
Usage of /: 46.7% of 9.75GB Users logged in: 0
Memory usage: 11% IPv4 address for enp0s3: 192.168.1.13 → MY HOST MACHINE NETWORK
Swap usage: 0% IPv4 address for enp0s8: 192.168.10.23 → UNDERLAY NETWORK

Expanded Security Maintenance for Applications is not enabled.

43 updates can be applied immediately.
To see these additional updates run: apt list --upgradable.

node1 [Running] - Oracle VM VirtualBox
```

```
work1 [Running] - Oracle VM VirtualBox
[ 16.923426] cloud-init[1273]: 2023-05-09 08:20:18,891 - cc_final_message.py[WARNIN]: Used fake datasource

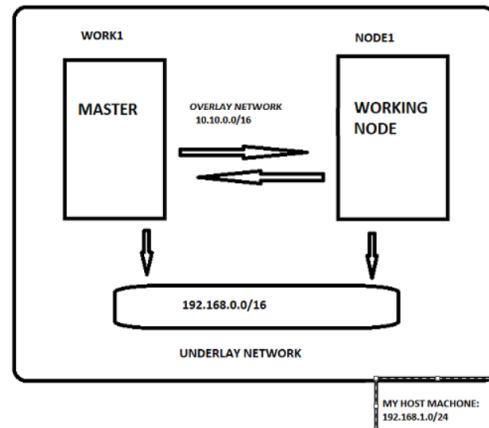
work1 login: generic
Password:
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-71-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

System information as of Tue May 9 08:41:37 AM UTC 2023

System load: 1.2421875 Users logged in: 1
Usage of /: 78.9% of 9.75GB IPv4 address for docker0: 172.17.0.1
Memory usage: 27% IPv4 address for enp0s3: 192.168.1.12
Swap usage: 0% IPv4 address for enp0s8: 192.168.10.22 → UNDERLAY NETWORK
Processes: 207

work1 [Running] - Oracle VM VirtualBox
```



```
root@work1:~# kubectl get no
NAME     STATUS   ROLES      AGE     VERSION
node1   Ready    <none>    2m40s   v1.27.1
work1   Ready    control-plane   48m    v1.27.1
root@work1:~#
root@work1:~#
root@work1:~#
root@work1:~#
```

```
root@ip-172-31-26-133:/home/ubuntu# vim /etc/hosts root@ip-172-31-26-133:/home/ubuntu# sudo bash root@ip-172-31-26-133:/home/ubuntu# swapoff -a
```

```
root@ip-172-31-26-133:/home/ubuntu# cat /etc/hosts →  
127.0.0.1 localhost ←  
# The following lines are desirable for IPv6 capable hosts  
::1 ip6-localhost ip6-loopback  
fe00::0 ip6-localnet  
ff00::0 ip6-mcastprefix  
ff02::1 ip6-allnodes  
ff02::2 ip6-allrouters  
ff02::3 ip6-allhosts → make an entry of local host with ip address  
root@ip-172-31-26-133:/home/ubuntu# vim /etc/hosts  
root@ip-172-31-26-133:/home/ubuntu# vim /etc/hosts →  
root@ip-172-31-26-133:/home/ubuntu# vim /etc/hosts
```

```
# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix 0cf5e52812a7d354 > Connect to instance
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts
```

```
root@ip-172-31-26-133:/home/ubuntu# cat <<EOF | sudo tee /etc/modules-load.d/k8s.conf
overlay
overlayfs
br_netfilter
EOF
EOF cat <<EOF | sudo tee /etc/modules-load.d/k8s.conf
overlay
overlayfs
br_netfilter
EOF
root@ip-172-31-26-133:/home/ubuntu# cat <<EOF | sudo tee /etc/sysctl.d/k8s.conf
net.bridge.bridge-nf-call-iptables = 1
net.bridge.bridge-nf-call-ip6tables = 1
net.ipv4.ip_forward = 1
EOF
EOF cat <<EOF | sudo tee /etc/sysctl.d/k8s.conf
net.bridge.bridge-nf-call-iptables = 1
net.bridge.bridge-nf-call-ip6tables = 1
net.ipv4.ip_forward = 1
EOF
EOF
EOF
```

```
root@ip-172-31-26-133:/home/ubuntu# sudo sysctl --system
* Applying /etc/sysctl.d/10-console-messages.conf ...
kernel.printk = 4 1 7
* Applying /etc/sysctl.d/10-ipv6-privacy.conf ...
net.ipv6.conf.all.use_tempaddr = 2
net.ipv6.conf.default.use_tempaddr = 2
* Applying (/etc/sysctl.d/10-kernel-hardening.conf)
```

```
root@ip-172-31-26-133:/home/ubuntu# sudo modprobe overlay  
root@ip-172-31-26-133:/home/ubuntu# sudo modprobe br_netfilter
```

```
root@ip-17-2-31-26-133:/home/ubuntu# apt-get install -y apt-transport-https ca-certificates curl gnupg lsb-release
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'apt' instead of 'apt-transport-https'
apt is already the newest version (2.4.8).
```

```
root@ip-172-31-26-131:~/home/ubuntu# mkdir /etc/apt/keyrings
root@ip-172-31-26-131:~/home/ubuntu# curl -s https://apt.kubernetes.io/keys/kubernetes-archive-keyring.gpg https://packages.cloud.google.com/apt/doc/apt-key.gpg
mkdir: cannot create directory '/etc/apt/keyrings': File exists
root@ip-172-31-26-131:~/home/ubuntu# curl -s https://apt.kubernetes.io/keys/kubernetes-archive-keyring.gpg https://apt.kubernetes.io/kubernetes-xenial main | sudo tee /etc/apt/keyrings/kubernetes-xenial.gpg
curl: [downloaded 1 file]
root@ip-172-31-26-131:~/home/ubuntu# curl -s https://apt.kubernetes.io/keys/kubernetes-archive-keyring.gpg https://apt.kubernetes.io/kubernetes-xenial main | sudo tee /etc/apt/keyrings/kubernetes-xenial.gpg
```

```
Root@ip-172-31-26-13:/home/ubuntu# apt update
Hit:1 https://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease [119 kB]
Hit:2 https://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [108 kB]
Get:3 https://security.ubuntu.com/ubuntu jammy-security InRelease [10 kB]          .io/kubernetes-xenial
Get:4 https://packages.cloud.google.com/apt kubernetes-xenial InRelease [1893 kB]
Get:6 https://packages.cloud.google.com/apt kubernetes-xenial/main amd64 Packages [65.7 kB]
Get:7 https://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [641 kB]
```

```
root@172-31-26-133:/home/ubuntu# apt-get install -y kubelet kubeadm kubectl docker.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
bridge-utils conntrack contained cri-tools dns-root-data dnsmasq-base ebtables kubernetes-cni pigz runc socat ubuntu-fan
Suggested packages:
  ifdown aufs-tools cgroupfs-mount | cgroup-lite debootstrap docker-doc rinse zfs-fuse | zfsutils
The Following NEW packages will be installed:
  bridge-utils conntrack contained cri-tools dns-root-data dnsmasq-base docker.io ebtables kubeadm kubectl kubernetes
0 upgraded, 10 newly installed, 0 to remove and 40 not upgraded.
Need to get 158 MB of archives.
After this operation, 615 MB of additional disk space will be used.
```

```
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/driver-free amdgpu-f102 and4 > 2.16 kB [65.8 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 bridge-utils amd64 1.7~ubuntus [34.4 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 conctrack amd64 1:1.4.6-2~build2 [33.5 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 run amd64 1.4.1~ubuntous-22.04.1 [4241 kB]
```

```
root@ip-172-31-26-133:/home/ubuntu# apt-mark hold kubelet kubeadm kubectl docker.io
kubelet set on hold.
kubeadm set on hold.
kubectl set on hold.
docker.io set on hold.
root@ip-172-31-26-133:/home/ubuntu#
```

```
root@ip-172-31-26-133:~/home/ubuntu          04-master-node installation > Microsoft Q
root@ip-172-31-26-133:~/home/ubuntu# mkdir /etc/containerd
root@ip-172-31-26-133:~/home/ubuntu# containerd config default >> /etc/containerd/config.toml
root@ip-172-31-26-133:~/home/ubuntu# vim /etc/containerd/config.toml
```

```
[plugins."io.containerd.grpc.v1.cri".containerd.runtimes]
  base_runtime_spec = ""
  cni_conf_dir = ""
  cni_max_conf_num = 0 Kubernetes Bootcamp(SIR.BABR)
  container_annotations = []
  pod_annotations = [] rings/kubernetes-archive-keyring.grb
    privileged_without_host_devices = false
    runtime_engine = ""
  echo runtime_path = "/etc/capt/keyrings/kubernetes-archive-keyr
  /etc/capt/runtime_root = "/kubernetes.list
  runtime_type = "io.containerd.runc.v2"

  [plugins."io.containerd.grpc.v1.cri".containerd.runtimes]
    BinaryName = ""
    CriuImagePath = ""
    CriuPath = "" kubeadm kubectl docker.io
    CriuWorkPath = ""
    IoGid = 0 kubeadm kubectl docker.io
    IoUid = 0
    NoNewKeyring = false
    NoPivotRoot = false
    Root = "ierd"
    ShimCgroup = "" >> /etc/containerd/config.toml
    SystemdCgroup = true

[plugins."io.containerd.grpc.v1.cri".containerd.untrusted_runtimes]
  base_runtime_spec = ""
  cni_conf_dir = ""
  cni_max_conf_num = 0
  container_annotations = []
  pod_annotations = []
  privileged_without_host_devices = false
  runtime_engine = ""
  runtime_path = "">>
  runtime_root = ""
  runtime_type = "" work.cidr=10.10.0.0/16
```

```
root@ip-172-31-26-133:~/home/ubuntu# systemctl restart containerd_k8s
root@ip-172-31-26-133:~/home/ubuntu# systemctl restart kubelet
root@ip-172-31-26-133:~/home/ubuntu# kubectl get no --export=kubernetes-archive-keyring.go https://packages.cloud.google.com/apt/doc/apt-key.gpg
```

```
[addons] Applied essential addon: coredns
[addons] Applied essential addon: kube-proxy

Your Kubernetes control-plane has initialized successfully!

To start using your cluster, you need to run the following as a 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

Alternatively, if you are the root user, you can run:
export KUBECONFIG=/etc/kubernetes/admin.conf

You should now deploy a pod network to the cluster.
Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:
https://kubernetes.io/docs/concepts/cluster-administration/addons/

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

kubeadm join 10.31.26.133:6443 --token f9fxky6.aaksj9m4ib1s1045 \
    --discovery-token-ca-cert-sha sha256:84d5454e02bd068108221349a2d5b5de3672befa9abdcd7881bae567ddd4c
```

MKDIR .KUBE;

```
root@ip-172-31-26-133:~# mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config
root@ip-172-31-26-133:~#
root@ip-172-31-26-133:~#
```

```
root@ip-172-31-26-131:~# kubectl create -f https://raw.githubusercontent.com/projectcalico/calico/v3.25.0/manifests/tigera-operator.yaml
namespace/tigera-operator created
customresourcedefinition.apirextensions.k8s.io/configurations.crd.projectcalico.org created
customresourcedefinition.apirextensions.k8s.io/boppers.crd.projectcalico.org created
customresourcedefinition.apirextensions.k8s.io/blockaffinities.crd.projectcalico.org created
customresourcedefinition.apirextensions.k8s.io/cidrconciliations.crd.projectcalico.org created
customresourcedefinition.apirextensions.k8s.io/clusterinformations.crd.projectcalico.org created
customresourcedefinition.apirextensions.k8s.io/gatewayconfigurations.crd.projectcalico.org created
customresourcedefinition.apirextensions.k8s.io/gatewaynetworks.crd.projectcalico.org created
customresourcedefinition.apirextensions.k8s.io/helixworkspaces.crd.projectcalico.org created
```

```
root@ip-172-31-26-133:~# curl https://raw.githubusercontent.com/projectcalico/calico/v3.25.0/manifests/custom-resources.yaml --output /tmp/calico.yaml
Total: 827 100 827 0 0 5586 0 0:--:-- --:--:--:--:--:-- 5625
root@ip-172-31-26-133:~#
root@ip-172-31-26-133:~#
root@ip-172-31-26-133:~# kubectl apply -f custom-resources.yaml
installations.operator.tigera.io/default created
apiserver.operator.tigera.io/default created →
root@ip-172-31-26-133:~#
```

Every 5.0s: kubectl get po -A

NAMESPACE	NAME	READY	STATUS	RESTARTS	AGE
calico-system	calico-kube-controllers-59d9cb8fb-5hzrp	0/1	ContainerCreating	0	2m25s
calico-system	calico-node-mpzfm	0/1	PodInitializing	0	2m25s
calico-system	calico-typha-55df88b96b-wfbxc	1/1	Running	0	2m25s
calico-system	csi-node-driver-np2tk	0/2	ContainerCreating	0	40s
kube-system	coredns-5d78c9869d-qb74p	0/1	ContainerCreating	0	5m53s
kube-system	coredns-5d78c9869d-s5f6p	0/1	ContainerCreating	0	5m53s
kube-system	etcd-work1	1/1	Running	0	6m6s
kube-system	kube-apiserver-work1	1/1	Running	0	6m6s
kube-system	kube-controller-manager-work1	1/1	Running	0	6m6s
kube-system	kube-proxy-v2cz9	1/1	Running	0	5m53s
kube-system	kube-scheduler-work1	1/1	Running	0	6m6s
tigera-operator	tigera-operator-59c686f986-qkjbv	1/1	Running	0	5m21s

```
root@work1:~# kubectl get no
NAME STATUS ROLES AGE VERSION
work1 Ready control-plane 6m49s v1.27.1
root@work1:~# ignore name / its a master
root@work1:~#
```

Repeat the same process for the working node (node1 till kubeadm init command.

In the working node, we need to put the key and make a cluster with master(work1).

```
root@node1:~# kubeadm join 192.168.10.22:6443 --token xzz5vo.4kxyi042my84f8ck --discovery-token-ca-cert-hash sha256:078aa2002217c9dafb429145f901195c9fb932
[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`.
[kubeadm-start] Writing kubeadm configuration file to file `/var/lib/kubelet/config.yaml`
[kubeadm-start] Writing kubelet environment file to file `/var/lib/kubelet/kubeadm-flags.env`
[kubeadm-start] Starting the kubelet
[kubeadm-start] Waiting for the kubelet to perform the TLS Bootstrap...
```

DUE TO SOME REASON (MACHINE CRASH) - CONSIDER THE FOLLOWING NAMING FOR THE MACHINES:

MASTER --> WORK1

WORKING NODE --> NODE1

```
root@work1:~# kubectl get no
NAME STATUS ROLES AGE VERSION
node1 Ready <none> 2m40s v1.27.1 → working (NODE-1)
work1 Ready control-plane 40m v1.27.1 → MASTER ( WORK1 )
root@work1:~#
root@work1:~#
root@work1:~#
```

```
Every 5.0s: kubectl get po -A
root@work1:~# kubectl get po -A
work1: Tue May  9 08:26:22 2023
NAMESPACE   NAME          READY   STATUS    RESTARTS   AGE
calico-apiserver  calico-apiserver-7ddff04478-f2p4f  1/1   Running  1 (6m10s ago)  33m
calico-node   calico-node-777ff04478-r29qj  1/1   Running  1 (6m10s ago)  33m
calico-system  calico-kube-controllers-59d9cb8fb-5hzrp  1/1   Running  0 (6m10s ago)  47m
calico-system  calico-node-1jdr2  1/1   Running  0  4m39s
calico-system  calico-node-mpzfm  1/1   Running  2 (6m10s ago)  47m
calico-system  csi-node-driver-55df88b96b-wfbxc  1/1   Running  0 (6m24s ago)  47m
calico-system  csi-node-driver-np2tk  2/2   Running  0  2m57s
calico-system  coredns-5d78c9869d-qb74p  1/1   Running  2 (6m10s ago)  50m
kube-system   coredns-5d78c9869d-s5f6p  1/1   Running  2 (6m10s ago)  50m
kube-system   etcd-work1  1/1   Running  2 (6m11s ago)  50m
kube-system   kube-apiserver-work1  1/1   Running  2 (6m11s ago)  50m
kube-system   kube-controller-manager-work1  1/1   Running  2 (6m10s ago)  50m
kube-system   kube-proxy-v2cz9  1/1   Running  0  4m39s
kube-system   kube-scheduler-work1  1/1   Running  2 (6m10s ago)  50m
tigera-operator tigera-operator-59c686f986-qkjbv  0/1   Completed 4 (5m21s ago)  49m
tigera-operator tigera-operator-59c686f986-qkjbv  1/1   Running  0  39s
```

=====

