



**Semester: V**

**Academic Year: 2025-26**

**Class / Branch: TE IT B**

**Subject: Advanced Devops Lab (ADL)**

**Name of Instructor: Prof. Manjusha K.**

**Name of Student: Tanmay padule**

**Student ID: 23104156**

---

### **EXPERIMENT NO. 04**

**Aim: To install Kubectl and execute Kubectl commands to manage the Kubernetes cluster and deploy Your First Kubernetes Application.**

### **Kubernetes Deployments**

In Order to configure Kubernetes Cluster we require 2 systems we can consider it as one master and one slave.

I have launch two Virtual machines master as kmaster and slave as knode as shown in Fig.

Configure static IP as:

**Master: 192.168.0.101 (kmaster)**

**Slave: 192.168.0.102 (knode)**



PARSHVANATH CHARITABLE TRUST'S  
**A. P. SHAH INSTITUTE OF TECHNOLOGY**  
Department of Information Technology  
(NBA Accredited)



The image shows two terminal windows from Oracle VM VirtualBox. The left window is titled 'kmaster [Running] - Oracle VM VirtualBox' and shows the output of the 'ifconfig' command for the 'root@kmaster-VirtualBox' user. The right window is titled 'knode [Running] - Oracle VM VirtualBox' and shows the output of the 'ifconfig' command for the 'root@knode-VirtualBox' user. Both windows show the configuration for the 'enp0s3' interface, including IP address, netmask, broadcast, and MTU.

```
root@kmaster-VirtualBox:~# ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.101 netmask 255.255.255.0 broadcast 192.168.0.255
    inet6 fe80::71a1:10d6:c989:e687 prefixlen 64 scopeid 0x2<link>
    ether 08:00:27:9f:c3:ec txqueuelen 1000 (Ethernet)
    RX packets 2780 bytes 2927125 (2.9 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1458 bytes 141791 (141.7 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 239 bytes 20646 (20.6 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 239 bytes 20646 (20.6 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@kmaster-VirtualBox:~#
```

```
root@knode-VirtualBox:~# ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.102 netmask 255.255.255.0 broadcast 192.168.0.255
    inet6 fe80::7b7d:5b87:254b:abe4 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:e3:f8:f8 txqueuelen 1000 (Ethernet)
    RX packets 2213 bytes 2936384 (2.9 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1264 bytes 121233 (121.2 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 215 bytes 18413 (18.4 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 215 bytes 18413 (18.4 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@knode-VirtualBox:~#
```

**Master: 192.168.0.101 (kmaster)**

**Slave: 192.168.0.102 (knode)**

The image shows a terminal window titled 'root@kmaster-VirtualBox:~' showing the output of the 'ifconfig' command. The output is identical to the one shown in the previous image for the 'kmaster' VM.

```
root@kmaster-VirtualBox:~# ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.101 netmask 255.255.255.0 broadcast 192.168.0.255
    inet6 fe80::71a1:10d6:c989:e687 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:9f:c3:ec txqueuelen 1000 (Ethernet)
    RX packets 2816 bytes 2929949 (2.9 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1467 bytes 142667 (142.6 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 239 bytes 20646 (20.6 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 239 bytes 20646 (20.6 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@kmaster-VirtualBox:~#
```

The image shows a terminal window titled 'root@knode-VirtualBox:~' showing the output of the 'ifconfig' command. The output is identical to the one shown in the previous image for the 'knode' VM.

```
root@knode-VirtualBox:~# ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.102 netmask 255.255.255.0 broadcast 192.168.0.255
    inet6 fe80::7b7d:5b87:254b:abe4 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:e3:f8:f8 txqueuelen 1000 (Ethernet)
    RX packets 2403 bytes 2975615 (2.9 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1377 bytes 137312 (137.3 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 227 bytes 19590 (19.5 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 227 bytes 19590 (19.5 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@knode-VirtualBox:~#
```

**For easy to connect and perform i have taken access of VMs on my machine for further steps.**

**Compiled By: Prof.Manjusha K.**

**Information Technology Department**



PARSHVANATH CHARITABLE TRUST'S  
**A. P. SHAH INSTITUTE OF TECHNOLOGY**  
Department of Information Technology  
(NBA Accredited)



```
Activities Google Chrome Sep 29 2:24 PM
Instances | EC2 | x EC2 Instance Con x Course: ITLS04 x Installing kubead x Kubeadm init exp x GitHub - flannel- x
us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?region=us-east-1&connType=standard&instanceId=i-04908165f3f62dd1f&osUser=ubuntu&sshPort=22&addressFamily=ipv4
aws Search [Alt+s] United States (N. Virginia) Account ID: 9894-1339-6452 rishabh

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@worker1:~$ sudo apt-mark hold kubelet kubeadm kubect
kubelet set on hold.
kubeadm set on hold.
kubectl set on hold.
ubuntu@worker1:~$ sudo systemctl enable --now kubelet
ubuntu@worker1:~$ sudo swapoff -a
ubuntu@worker1:~$ sudo kubeadm join 172.31.31.108:6443 --token dvhyxm.m9t0qgt1tkq15r83 --discovery-token-ca-cert-hash sha256:21c2436f3d267d22b53fad864c3914135bfd9843
8a919ee5ed481fbccb83c1e --ignore-preflight-errors=all
error: unknown flag: --ignore-preflight-errors=all
To see the stack trace of this error execute with --v=5 or higher
ubuntu@worker1:~$ sudo kubeadm join 172.31.31.108:6443 --token dvhyxm.m9t0qgt1tkq15r83 --discovery-token-ca-cert-hash sha256:21c2436f3d267d22b53fad864c3914135bfd98431
8a919ee5ed481fbccb83c1e --ignore-preflight-errors=all
[preflight] Running pre-flight checks
[preflight] Reading configuration from the "kubeadm-config" ConfigMap in namespace "kube-system"...
[preflight] Use 'kubeadm init phase upload-config kubeadm --config your-config-file' to re-upload it.
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/instance-config.yaml"
[patches] Applied patch of type "application/strategic-merge-patch+json" to target "kubeletconfiguration"
[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-check] Waiting for a healthy kubelet at http://127.0.0.1:10248/healthz. This can take up to 4m0s
[kubelet-check] The kubelet is healthy after 1.001772311s
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap

This node has joined the cluster:
* Certificate signing request was sent to apiser 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@worker1:~$

i-04908165f3f62dd1f (worker)
PublicIPs: 44.223.11.23 PrivateIPs: 172.31.23.26
CloudShell Feedback
© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
```



PARSHVANATH CHARITABLE TRUST'S  
**A. P. SHAH INSTITUTE OF TECHNOLOGY**  
Department of Information Technology  
(NBA Accredited)



```
Activities Google Chrome Sep 29 2:21 PM
Instances | EC2 | x EC2 Instance Con x Home | APSIT x Installing kubead x Kubeadm init exp x GitHub - flannel- x +
us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?region=us-east-1&connType=standard&instanceId=i-0d569782465dfb310&osUser=ubuntu&sshPort=22&addressFamily=i...
aws Search [Alt+S] United States (N. Virginia) Account ID: 9894-1339-6452 rishabh

ubuntu@master-node:~$ sudo kubectl apply -f https://github.com/flannel-io/flannel/releases/latest/download/kube-flannel.yml
error: error validating "https://github.com/flannel-io/flannel/releases/latest/download/kube-flannel.yml": error validating data: failed to download openapi: Get "http://localhost:8080/openapi/v2?timeout=32s": dial tcp 127.0.0.1:8080: connect: connection refused; if you choose to ignore these errors, turn validation off with --validate=false
ubuntu@master-node:~$ kubectl cluster-info
Kubernetes control plane is running at https://172.31.31.108:6443
CoreDNS is running at https://172.31.31.108:6443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
ubuntu@master-node:~$ kubectl apply -f https://github.com/flannel-io/flannel/releases/latest/download/kube-flannel.yml
namespace/kube-flannel created
serviceaccount/flannel created
clusterrole.rbac.authorization.k8s.io/flannel created
clusterrolebinding.rbac.authorization.k8s.io/flannel created
configmap/kube-flannel-cfg created
daemonset.apps/kube-flannel-ds created
ubuntu@master-node:~$ kubectl get pods -all-namespaces
error: unknown shorthand flag: 'a' in -all-namespaces
See 'kubectl get --help' for usage.
ubuntu@master-node:~$ kubectl get pods --all-namespaces
NAMESPACE NAME READY STATUS RESTARTS AGE
kube-flannel kube-flannel-ds-712v6 0/1 CrashLoopBackOff 6 (2m4s ago) 8m6s
kube-flannel kube-flannel-ds-pw7x7 0/1 CrashLoopBackOff 6 (3m59s ago) 10m
kube-system coredns-66bc5c9577-ms9dt 0/1 ContainerCreating 0 23m
kube-system etcd-master-node 1/1 Running 7 (12m ago) 23m
kube-system kube-apiserver-master-node 1/1 Running 6 (4m10s ago) 23m
kube-system kube-controller-manager-master-node 0/1 Running 12 (3m ago) 23m
kube-system kube-proxy-clfgv 1/1 Running 5 (93s ago) 8m6s
kube-system kube-proxy-pg15b 0/1 CrashLoopBackOff 11 (75s ago) 23m
kube-system kube-scheduler-master-node 1/1 Running 10 (3m36s ago) 24m
ubuntu@master-node:~$

i-0d569782465dfb310 (master)
PublicIPs: 52.87.173.85 PrivateIPs: 172.31.31.108
CloudShell Feedback
© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
```

**Conclusion: we successfully installed kubernetes and kubectl and deployed an application**