Semester: V **Name of Student:Tanmay padule**
Academic Year: 2025-26 **Student ID:23104156**
Class / Branch: TE IT B
Subject: Advanced Devops Lab (ADL)
Name of Instructor:Prof. Manjusha K.

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:

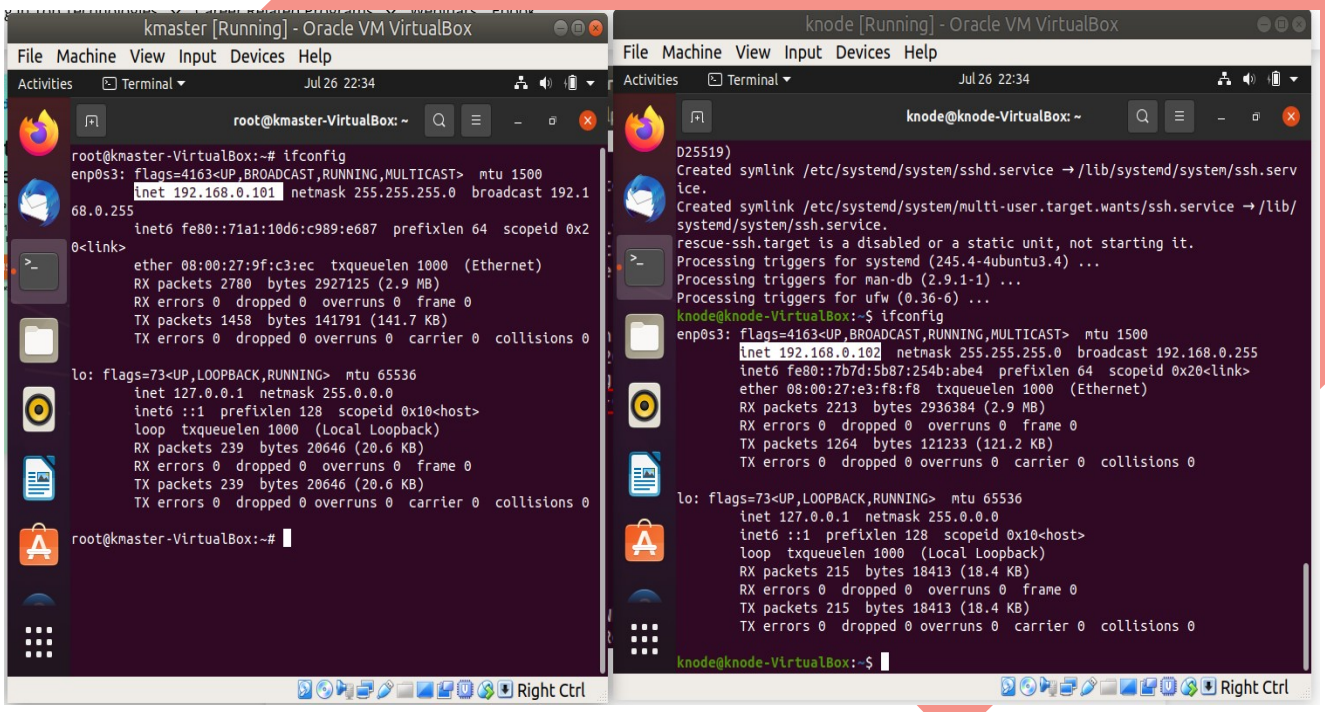
Master: 192.168.0.101 (kmaster)
Slave: 192.168.0.102 (knode)

PARSHVANATH CHARITABLE TRUST'S

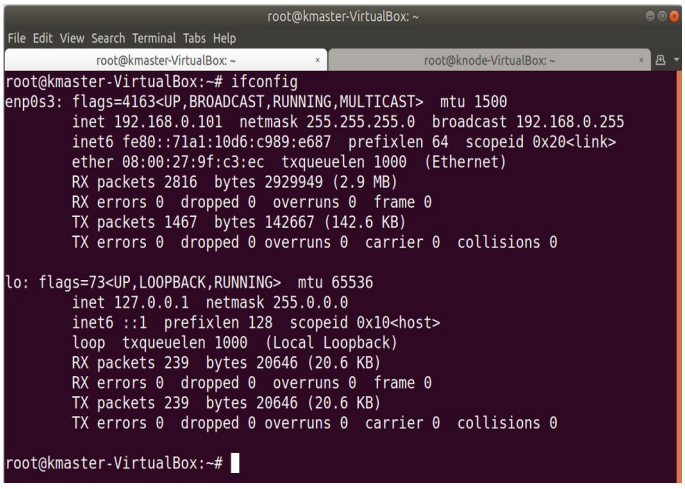
A. P. SHAH INSTITUTE OF TECHNOLOGY

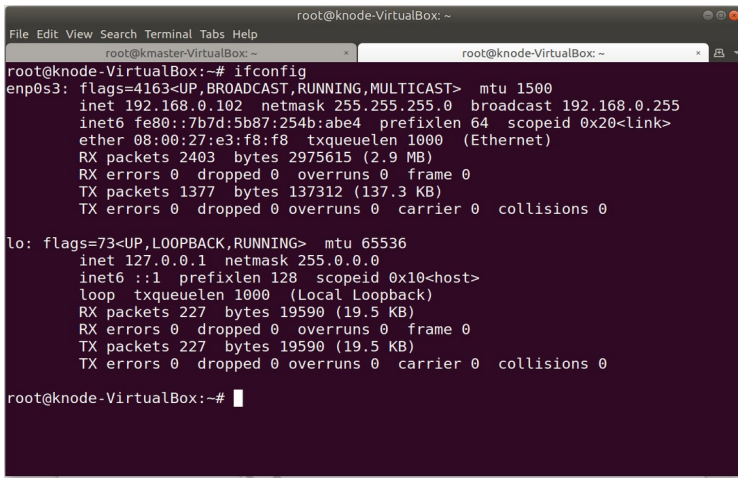
Department of Information Technology

(NBA Accredited)



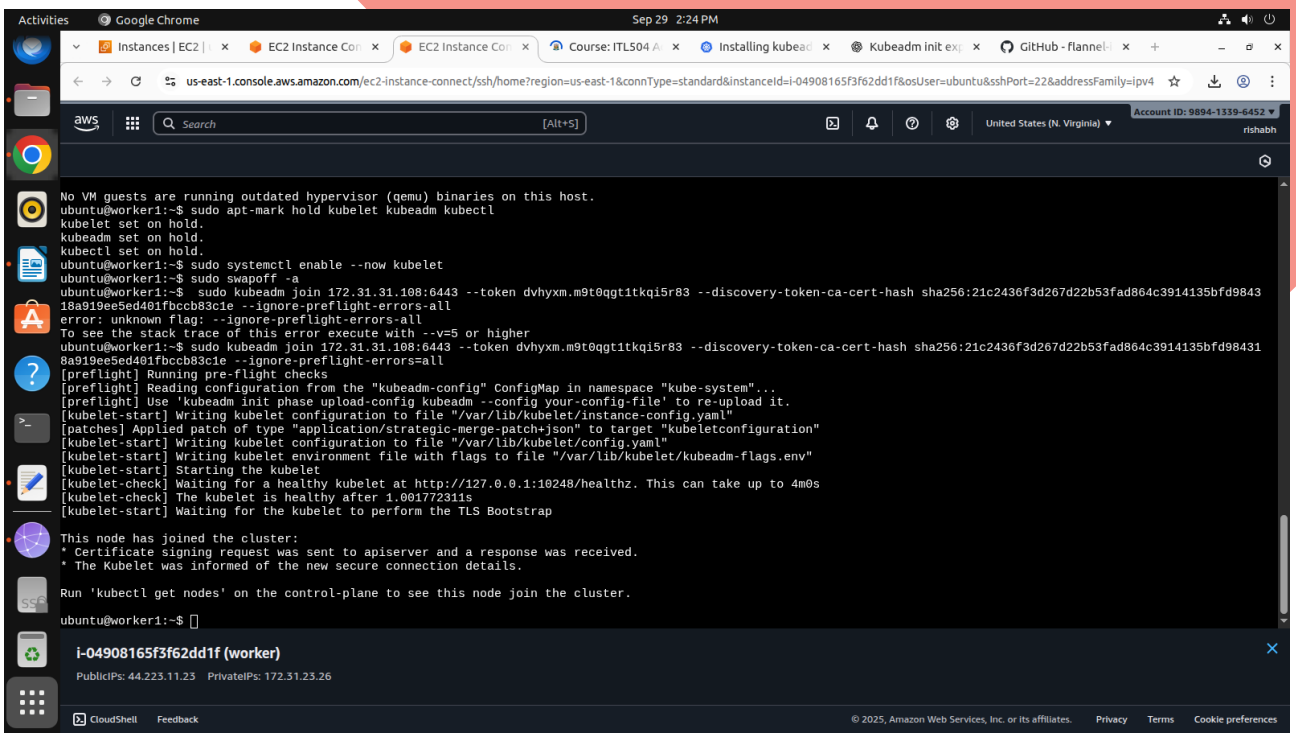
Master: 192.168.0.101 (kmaster) **Slave: 192.168.0.102 (knode)**

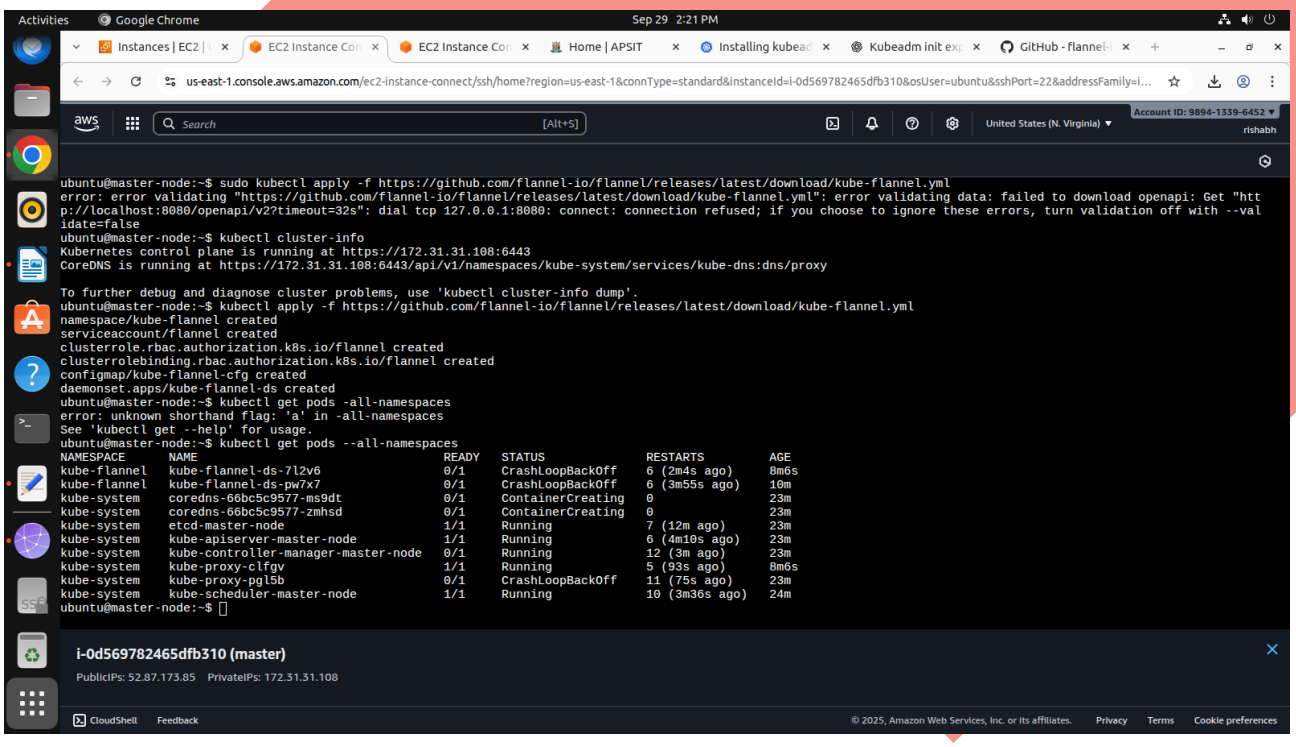




For easy to connect and perform i have taken access of VMs on my machine for further steps.

Compiled By: Prof.Manjusha K. **Information Technology Department**


Activities
Google Chrome
Sep 29 2:24 PM
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
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@worker1:~$ sudo apt-mark hold kubelet kubeadm kubectl
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.m9t0qgt1tkqi5r83 --discovery-token-ca-cert-hash sha256:21c2436f3d267d22b53fad864c3914135bfd9843
18a919ee5ed401fbccb83c1e --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.m9t0qgt1tkqi5r83 --discovery-token-ca-cert-hash sha256:21c2436f3d267d22b53fad864c3914135bfd98431
8a919ee5ed401fbccb83c1e --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 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@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



Conclusion: we sucessfully installed kubernetes and kubectl an deployed application