**AdvDevOps Lab**

**Experiment No. :-4**

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

# Theory:

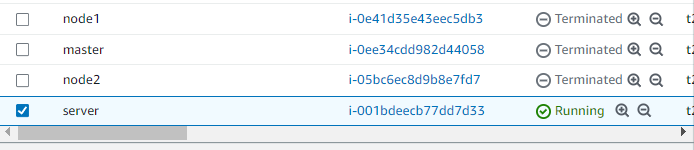
Originally developed by Google, Kubernetes is an open-source container orchestration platform designed to automate the deployment, scaling, and management of containerized applications. In fact, Kubernetes has established itself as the defacto standard for container orchestration and is the flagship project of the Cloud Native Computing Foundation (CNCF), backed by key players like Google, AWS, Microsoft, IBM, Intel, Cisco, and Red Hat.

# Kubernetes Deployment

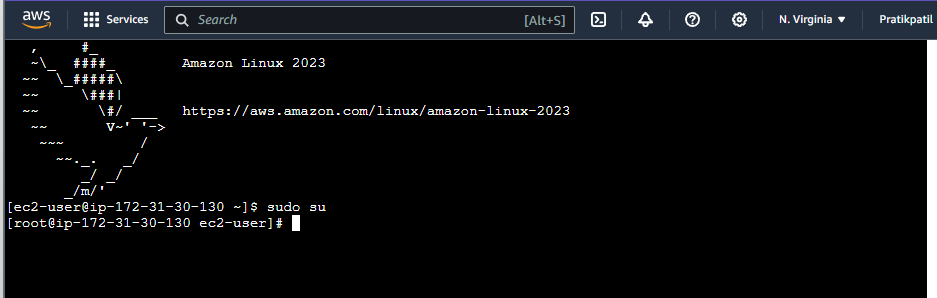
A Kubernetes Deployment is used to tell Kubernetes how to create or modify instances of the pods that hold a containerized application. Deployments can scale the number of replica pods, enable the rollout of updated code in a controlled manner, or roll back to an earlier deployment version if necessary.

# Steps:

1. Create an EC2 Ubuntu Instance on AWS.



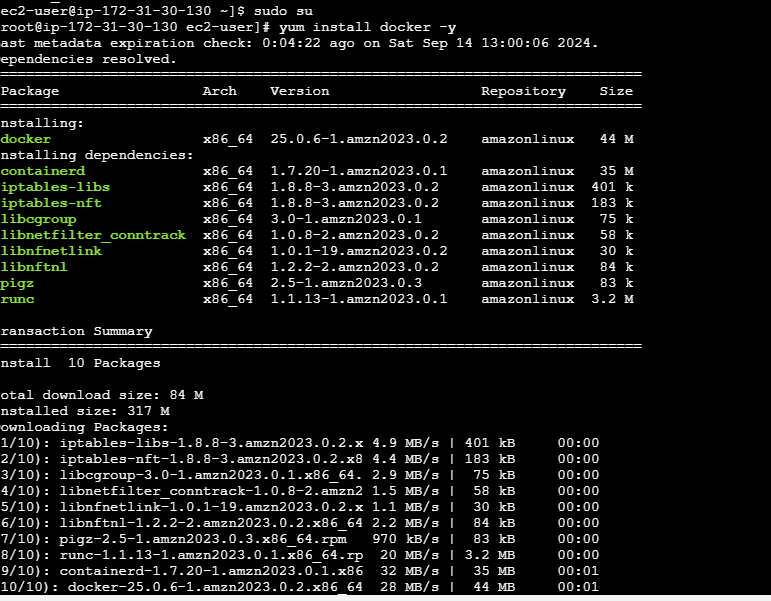
1. SSH into the machine



1. Install Docker

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb\_release -cs) stable" sudo apt-get update

sudo apt-get install -y docker-ce

Then, configure cgroup in a daemon.json file.

cd /etc/docker

cat <<EOF | sudo tee /etc/docker/daemon.json

{

"exec-opts": ["native.cgroupdriver=systemd"]

}

EOF

sudo systemctl enable docker sudo systemctl daemon-reload sudo systemctl restart docker

1. Install Kubernetes

*# Set SELinux in permissive mode (effectively disabling it)*

sudo setenforce 0

sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config

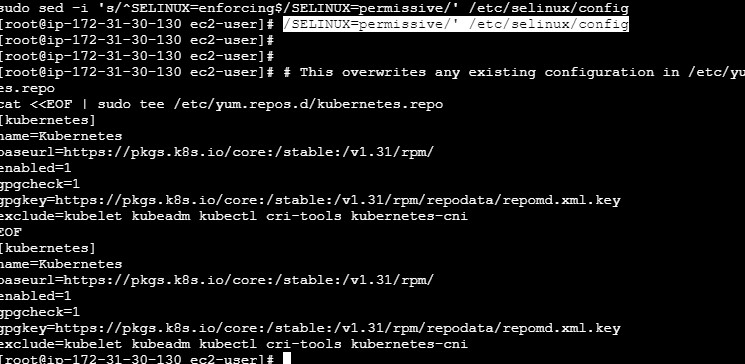
*# This overwrites any existing configuration in /etc/yum.repos.d/kubernetes.repo*

cat <<EOF | sudo tee /etc/yum.repos.d/kubernetes.repo [kubernetes]

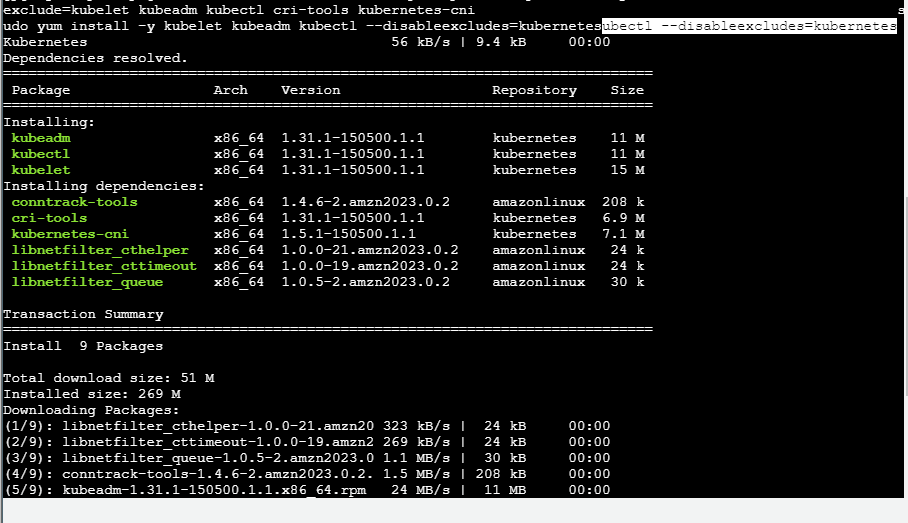
name=Kubernetes baseurl=https://pkgs.k8s.io/core:/stable:/v1.31/rpm/ enabled=1

gpgcheck=1 gpgkey=https://pkgs.k8s.io/core:/stable:/v1.31/rpm/repodata/repomd.xml.key exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni

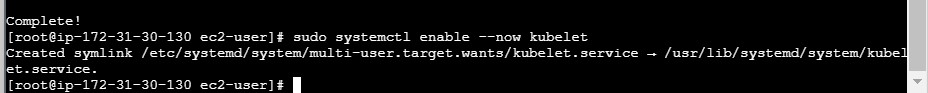
EOF



sudo yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes

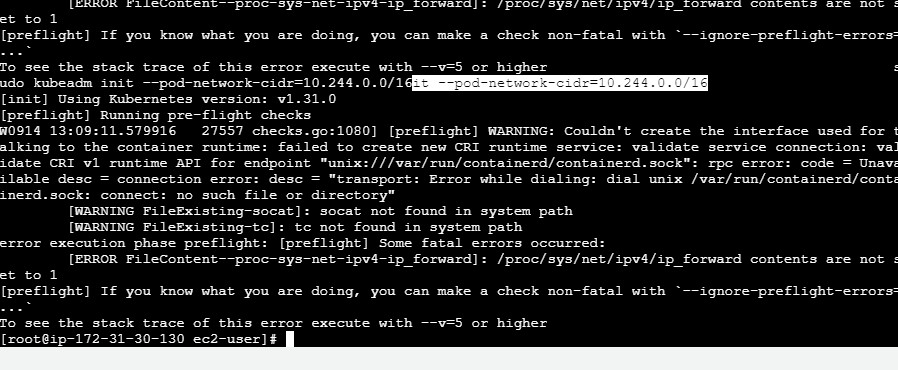


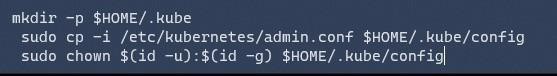
sudo systemctl enable --now kubelet



1. Initialize the Kubecluster

sudo kubeadm init --pod-network-cidr=10.244.0.0/16

=

Copy the mkdir and chown commands from the top and execute them

Then, add a common networking plugin called flannel as mentioned in the code.

kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation

/ k ube-flannel.yml



1. Now that the cluster is up and running, we can deploy our nginx server on this cluster.

Apply this deployment file using this command to create a deployment

kubectl apply -f https://k8s.io/examples/application/deployment.yaml



Use ‘kubectl get pods’ to verify if the deployment was properly created and the pod is working correctly.

Next up, create a name alias for this pod. POD\_NAME=$(kubectl get pods -l app=nginx -o jsonpath="{.items[0].metadata.name}")

1. Lastly, port forward the deployment to your localhost so that you can view it.

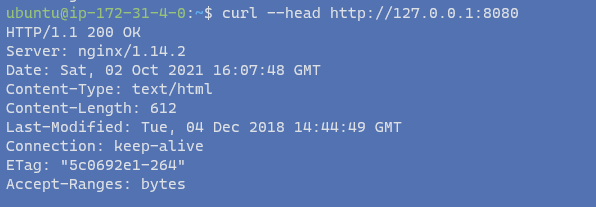
kubectl port-forward $POD\_NAME 8080:80

1. Verify your deployment

Open up a new terminal and ssh to your EC2 instance.

Then, use this curl command to check if the Nginx server is running.

curl --head http://127.0.0.1:8080



If the response is 200 OK and you can see the Nginx server name, your deployment was successful.

We have successfully deployed our Nginx server on our EC2 instance.

**Conclusion:** open-source platform originally developed by Google, has become the industry standard for managing containerized applications. It automates the deployment, scaling, and orchestration of containers, simplifying the process of maintaining complex application environments. With support from major industry players, Kubernetes ensures reliable, scalable, and efficient application management.

By using Kubernetes Deployments, developers can efficiently manage application lifecycles, scale replica pods, roll out updates in a controlled manner, and easily revert to previous versions if needed, making it a powerful tool for modern cloud-native application management.