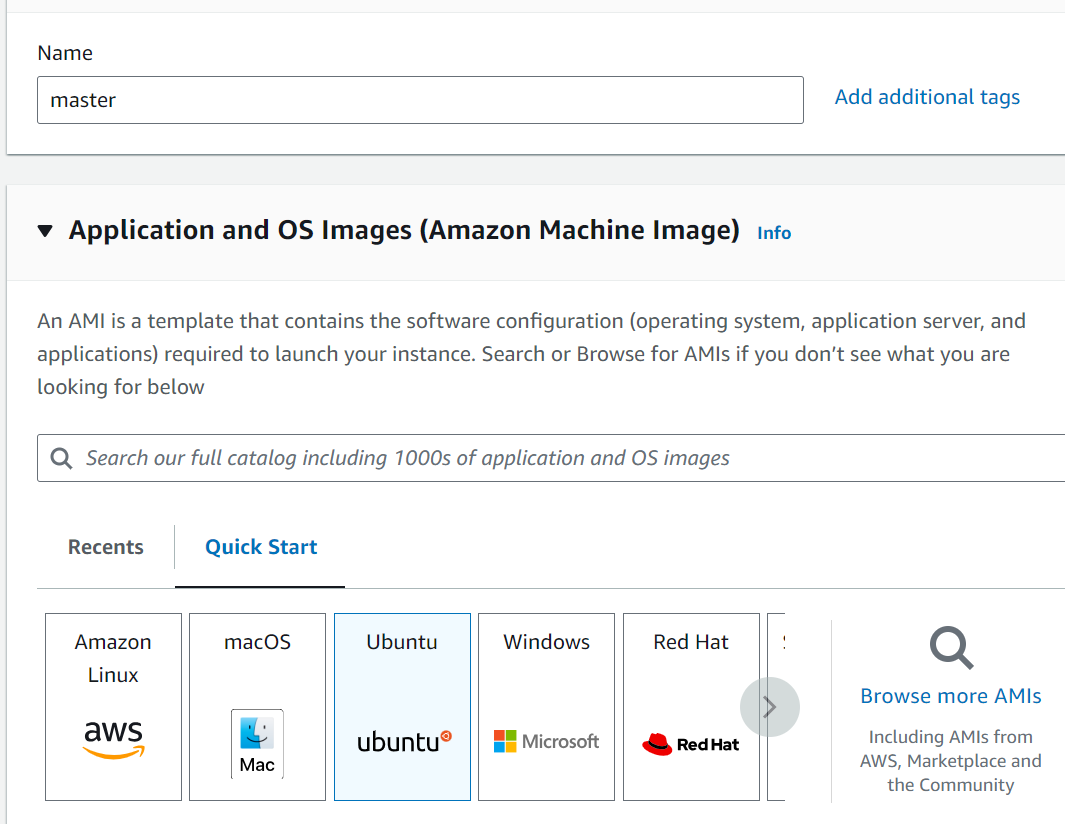
**ADVANCE DEVOPS EXPERIMENT 3**

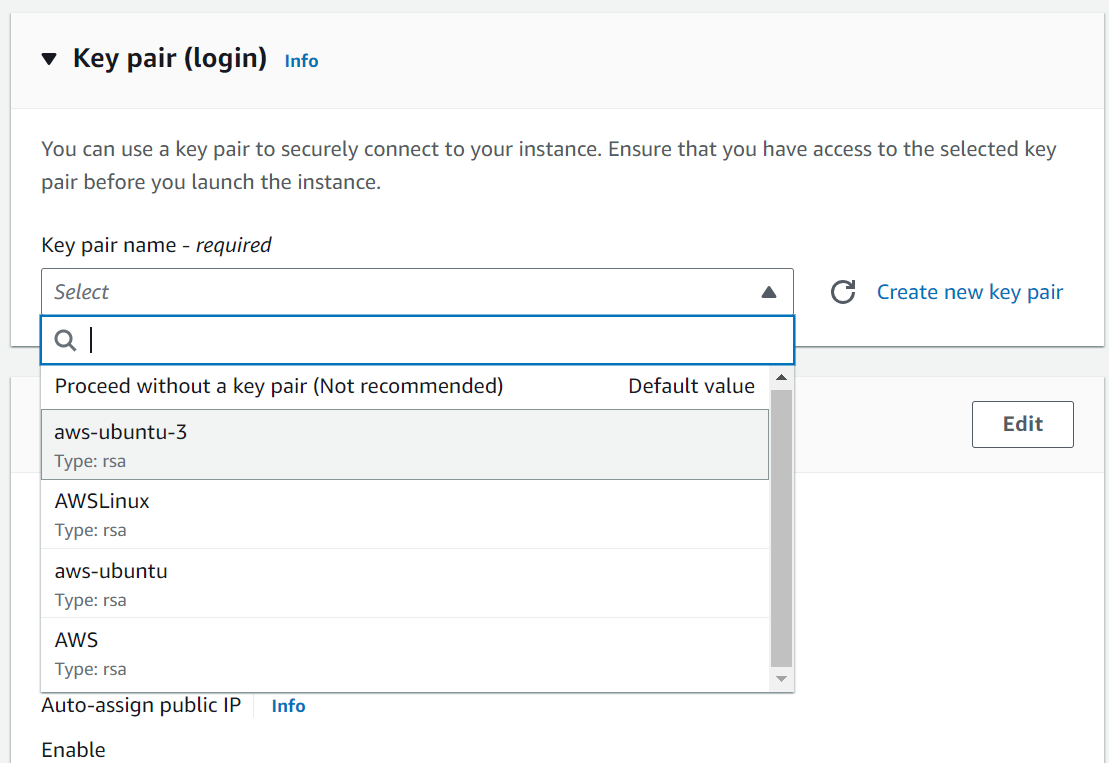
Prajjwal Pandey – D15A/33

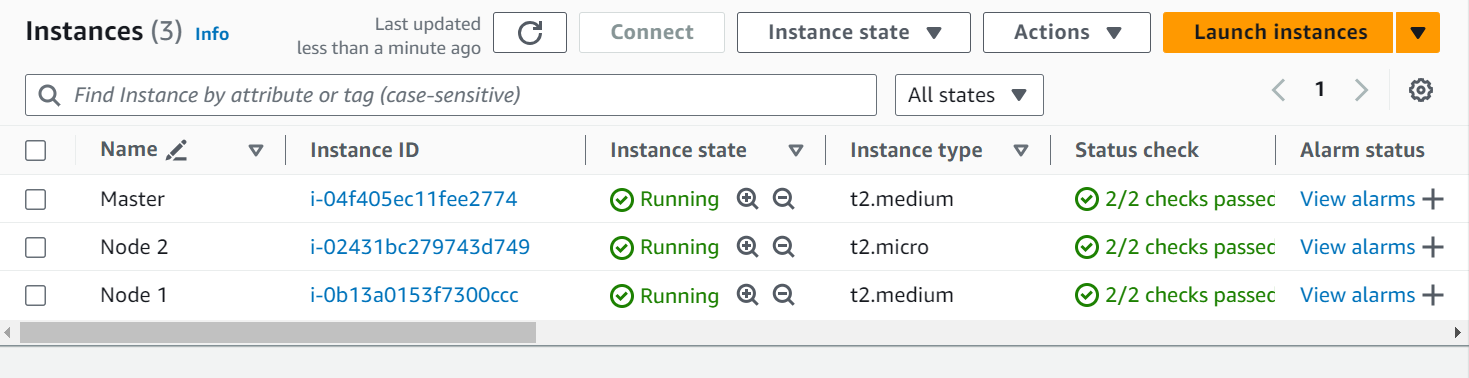
**Aim**:To understand the Kubernetes Cluster Architecture, install and Spin Up a Kubernetes Cluster on Linux Machines/Cloud Platforms.

* 1. Create 3 EC2 instances master, node 1 and node 2.

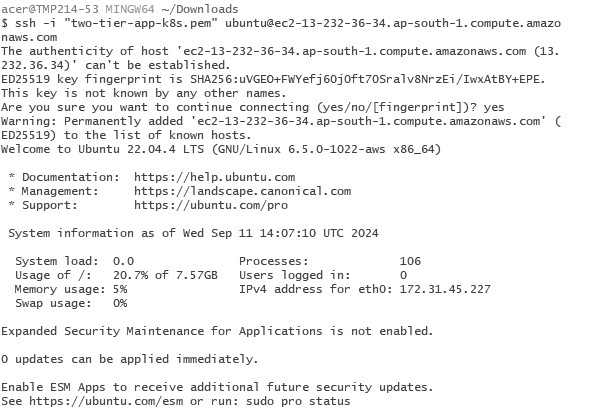
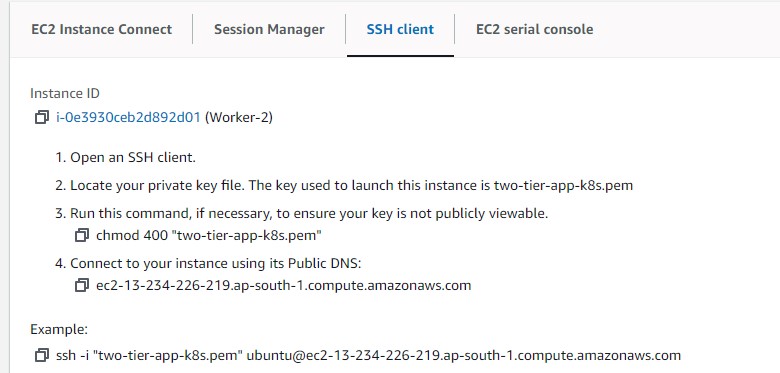


* 1. Create a new key pair (use the same key pair for all 3 instances)





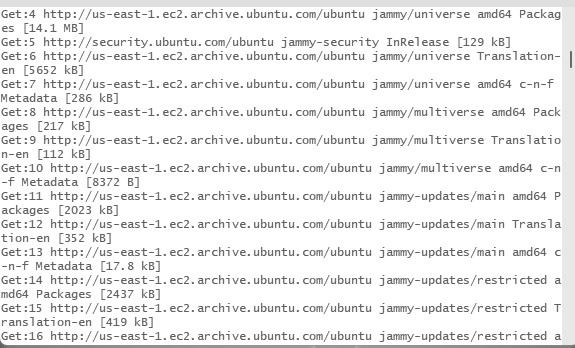
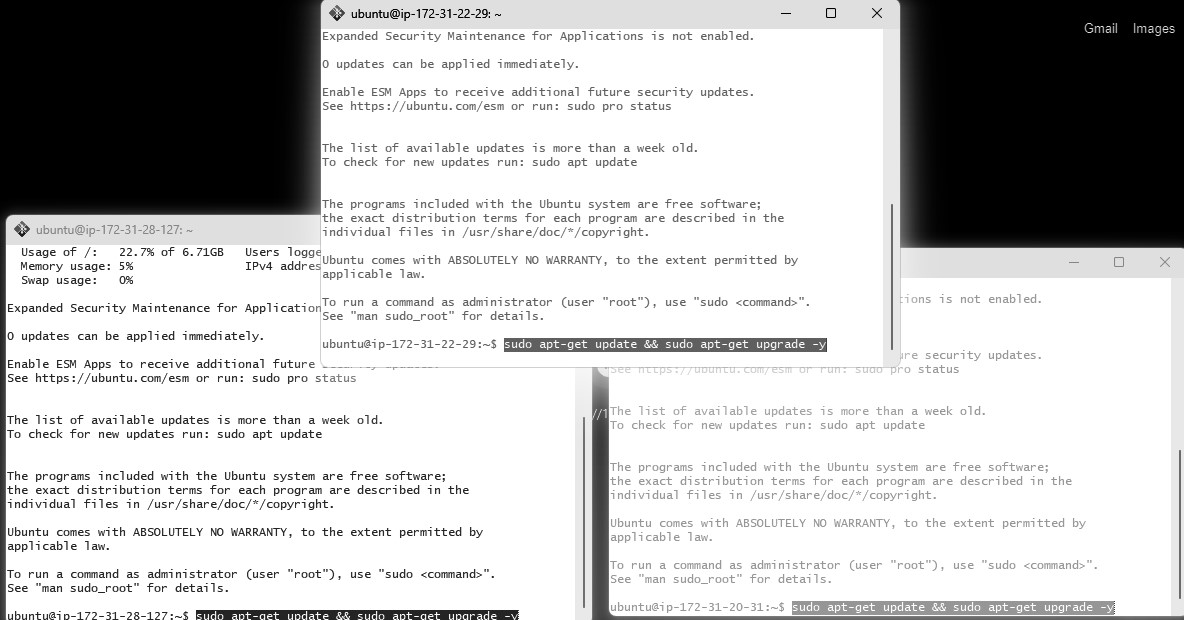
* 1. After the instances have been created, copy the text given in the example part of each of the three instances into git bash.



1.3 After the instances have been created,copy the text given in the example part of each of the three instances into git bash.

**Step 2:Prepare Nodes**

2.1. Update the package manager on all nodes: sudo apt-get update && sudo apt-get upgrade -y



2.2. Disable Swap (Kubernetes requires swap to be off): sudo swapoff -a sudo sed -i '/ swap / s/^/#/' /etc/fstab



2.3. Load necessary kernel modules for networking and iptables:

cat <<EOF | sudo tee /etc/modules-load.d/k8s.conf overlay br\_netfilter

EOF

sudo

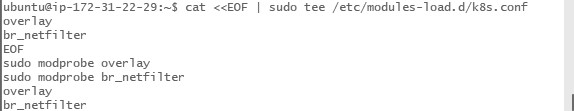
modprobe

overlay

sudo

modprobe

br\_netfilter

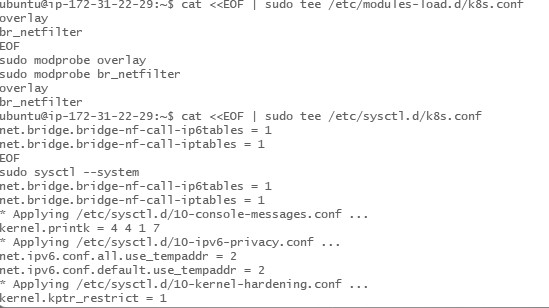


2.4. Configure sysctl settings for Kubernetes networking:

cat <<EOF | sudo tee /etc/sysctl.d/k8s.conf net.bridge.bridge-nf-call-ip6tables = 1 net.bridge.bridge-nf-call-iptables = 1

EOF

sudo sysctl --system



**Step 3: Install Docker**

Kubernetes uses container runtimes like Docker. Install Docker on all nodes.

sudo apt-get update sudo apt-get install -y apt-transport-https ca-certificates curl software-properties-common 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 docker-ce-cli containerd.io



Configure Docker for Kubernetes:

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

{

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

"log-driver": "json-file",

"log-opts": {

"max-size": "100m"

},

"storage-driver": "overlay2"

}

EOF

sudo

systemctl

restart

docker



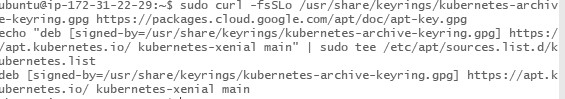
**Step 4: Install kubeadm, kubelet, kubectl**

Install Kubernetes tools on all nodes.

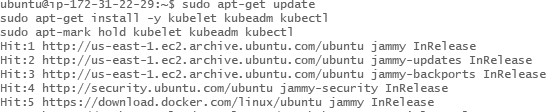
4.1. Add Kubernetes APT repository:

sudo curl -fsSLo /usr/share/keyrings/kubernetes-archive-keyring.gpg https://packages.cloud.google.com/apt/doc/apt-key.gpg echo "deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring.gpg] https://apt.kubernetes.io/ kubernetes-xenial main" | sudo tee

/etc/apt/sources.list.d/kubernetes.list



4.2. Install kubeadm, kubelet, and kubectl: sudo apt-get update sudo apt-get install -y kubelet kubeadm kubectl sudo apt-mark hold kubelet kubeadm kubectl



**Step 5: Initialize the Kubernetes Cluster on Master Node** On the master node:

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



5.1. Set up kubectl on the master node:

mkdir -p $HOME/.kube sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config sudo chown $(id -u):$(id -g) $HOME/.kube/config

**Step 6: Install a Pod Network Add-on**

To enable communication between pods, install a pod network plugin like Flannel or Calico.



**Install Flannel:**

kubectl apply -f

<https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml>

**Step 7: Join Worker Nodes to the Cluster**

On the **worker nodes**, run the command provided by the master node during initialization . It looks something like this:

sudo kubeadm join <master-ip>:6443 --token <token> --discovery-token-ca-cert-hash sha256:<hash>



**Step 8: Verify the Cluster**

Once the worker node joins, check the status on the **master node**

