## **EXPERIMENT NO: 03**

**AIM :** To understand the Kubernetes cluster architecture, install and spin up a kubernetes clusters on linux machine/ cloud platforms.

**LO :** LO1 &LO2

**THEORY:**

**What is kubernetes cluster?**

A Kubernetes cluster is a set of nodes that run containerized applications. Containerizing

applications packages an app with its dependences and some necessary services. They are more

lightweight and flexible than virtual machines. In this way, Kubernetes clusters allow for

applications to be more easily developed, moved and managed.

Kubernetes clusters allow containers to run across multiple machines and environments:

virtual, physical, cloud-based, and on-premises. Kubernetes containers are not restricted to a

specific operating system, unlike virtual machines. Instead, they are able to share operating

systems and run anywhere.

Kubernetes clusters are comprised of one master node and a number of worker nodes. These

nodes can either be physical computers or virtual machines, depending on the cluster.

The master node controls the state of the cluster; for example, which applications are running

and their corresponding container images. The master node is the origin for all task

assignments. It coordinates processes such as:

1) Scheduling and scaling applications

2)Maintaining a cluster’s state

3)Implementing updates

The worker nodes are the components that run these applications. Worker nodes perform tasks

assigned by the master node. They can either be virtual machines or physical computers, all

operating as part of one system.

There must be a minimum of one master node and one worker node for a Kubernetes cluster to

be operational. For production and staging, the cluster is distributed across multiple worker

nodes. For testing, the components can all run on the same physical or virtual node.

A namespace is a way for a Kubernetes user to organize many different clusters within just one

physical cluster. Namespaces enable users to divide cluster resources within the physical

cluster among different teams via resource quotas. For this reason, they are ideal in situations

involving complex projects or multiple teams.

**A Kubernetes cluster contains six main components:**

1. API server: Exposes a REST interface to all Kubernetes resources. Serves as the front end

of the Kubernetes control plane.

2. Scheduler: Places containers according to resource requirements and metrics. Makes note

of Pods with no assigned node, and selects nodes for them to run on.

3. Controller manager: Runs controller processes and reconciles the cluster’s actual state with

its desired specifications. Manages controllers such as node controllers, endpoints

controllers and replication controllers.

4. Kubelet: Ensures that containers are running in a Pod by interacting with the Docker engine,

the default program for creating and managing containers. Takes a set of provided PodSpecs

and ensures that their corresponding containers are fully operational.

5. Kube-proxy: Manages network connectivity and maintains network rules across nodes.

Implements the Kubernetes Service concept across every node in a given cluster.

6. Etcd: Stores all cluster data. Consistent and highly available Kubernetes backing store.

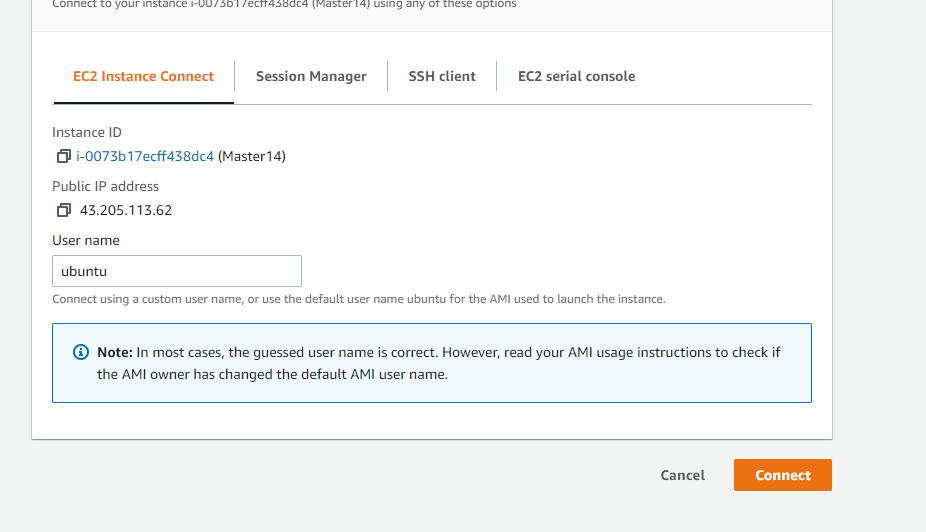
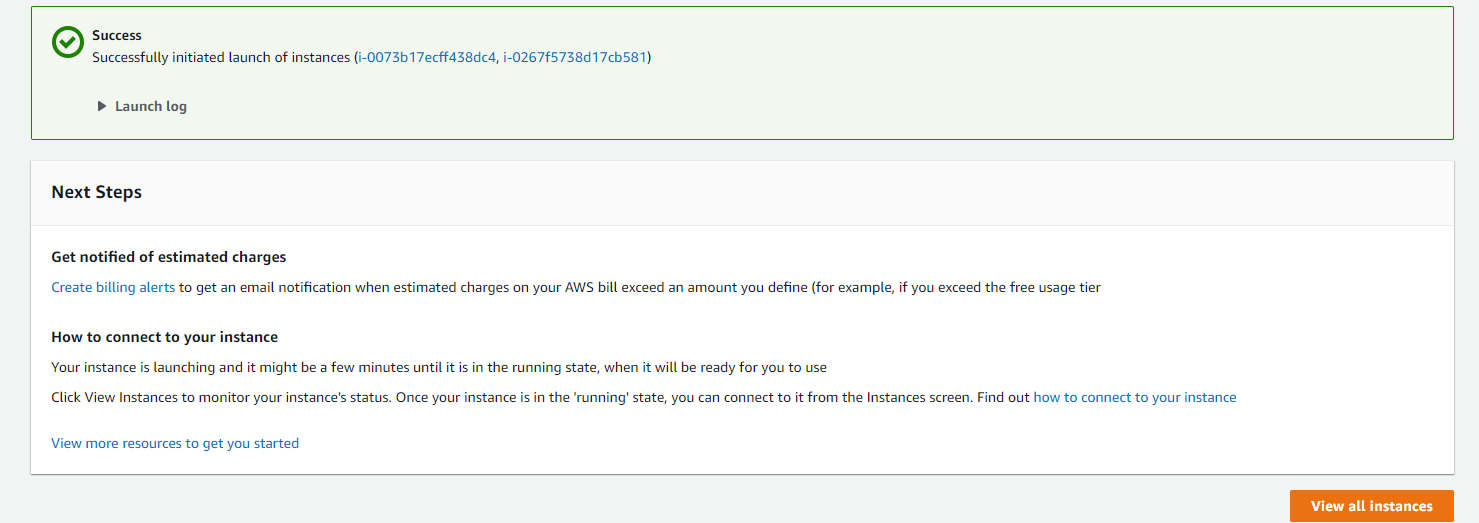
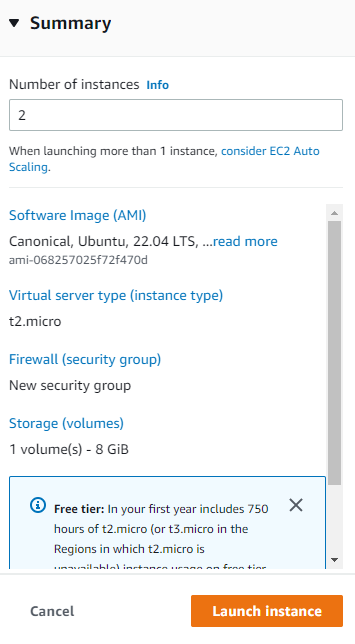
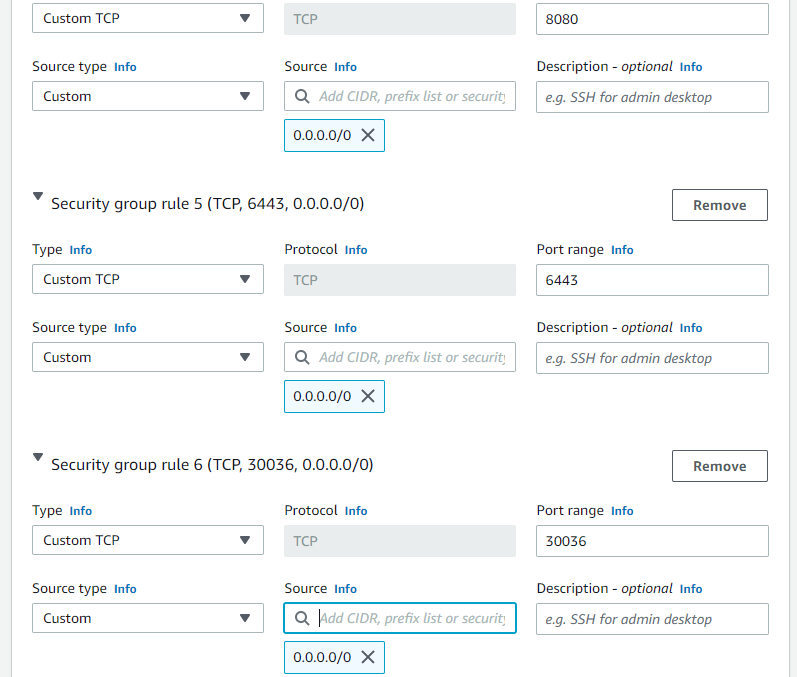
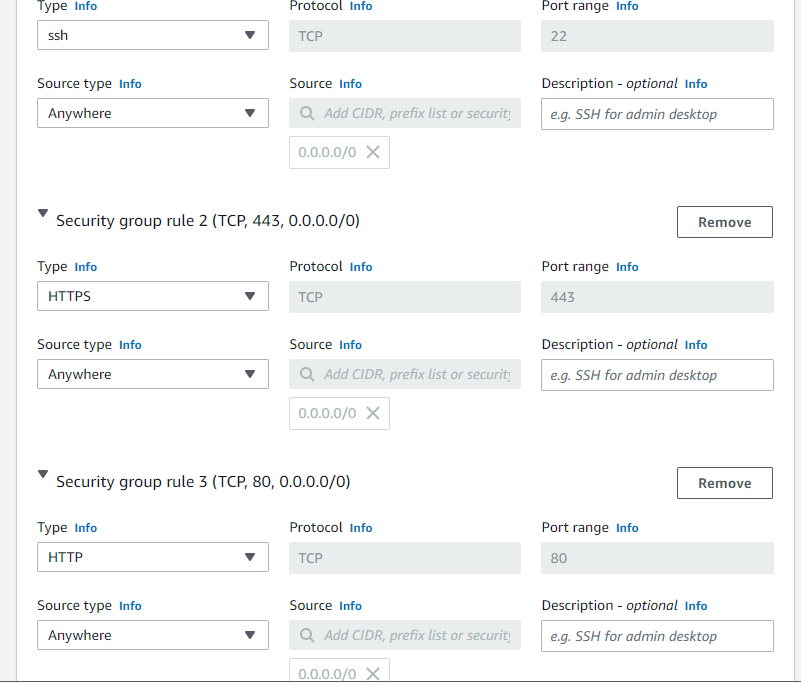
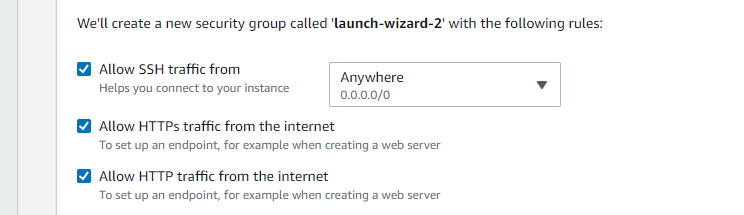
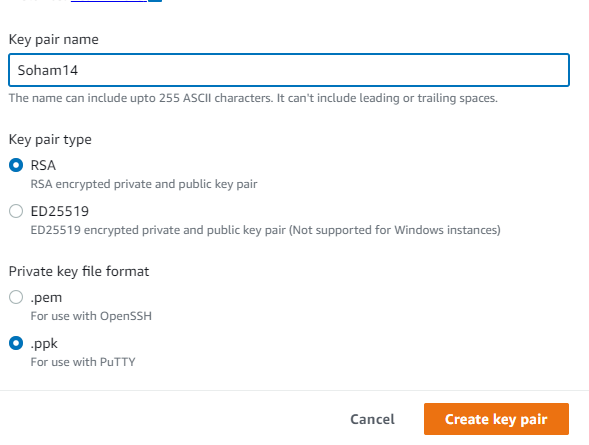
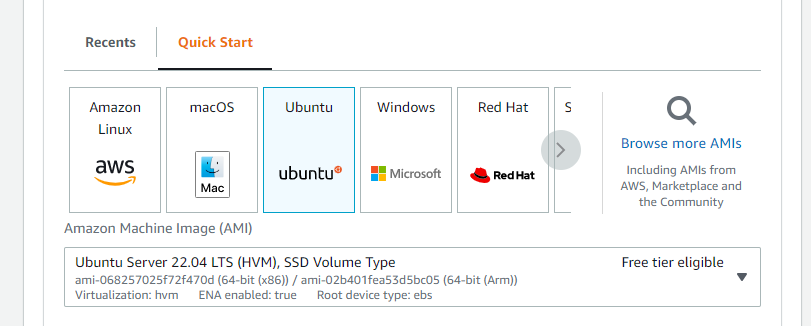
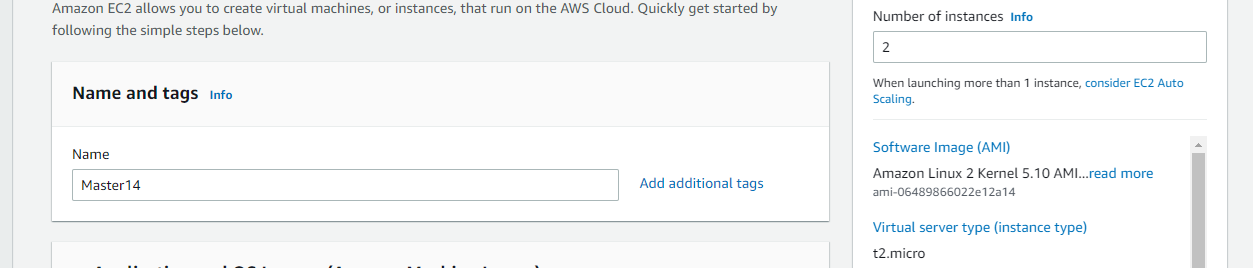
These six components can each run on Linux or as Docker containers. The master node runs

the API server, scheduler and controller manager, and the worker nodes run the kubelet and

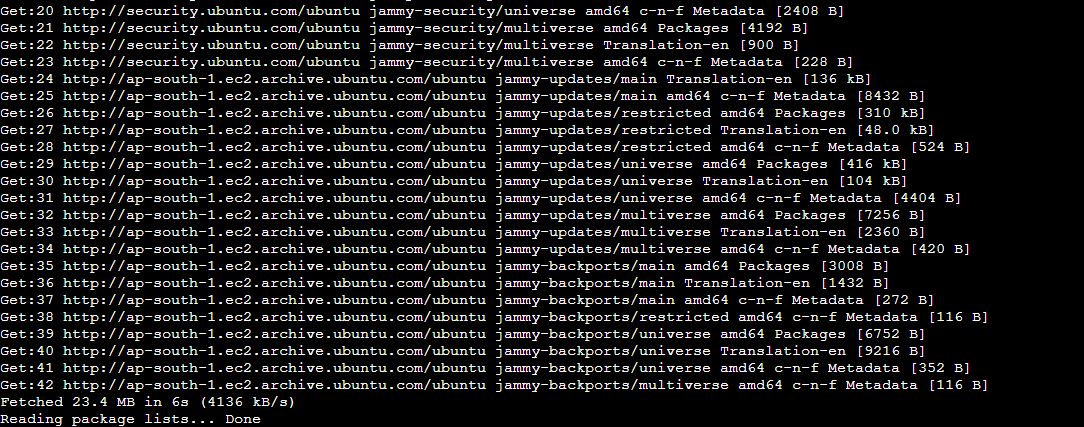
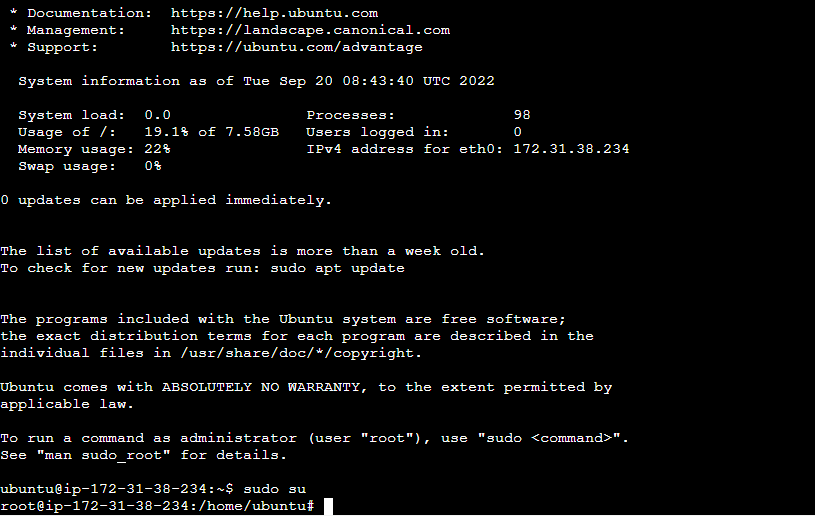
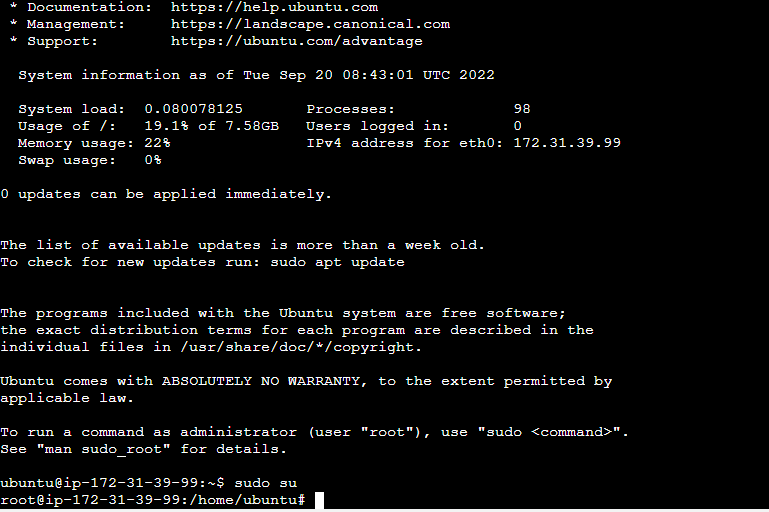
kube-proxy.

**Steps:**

1. Create an instance of linux machine.



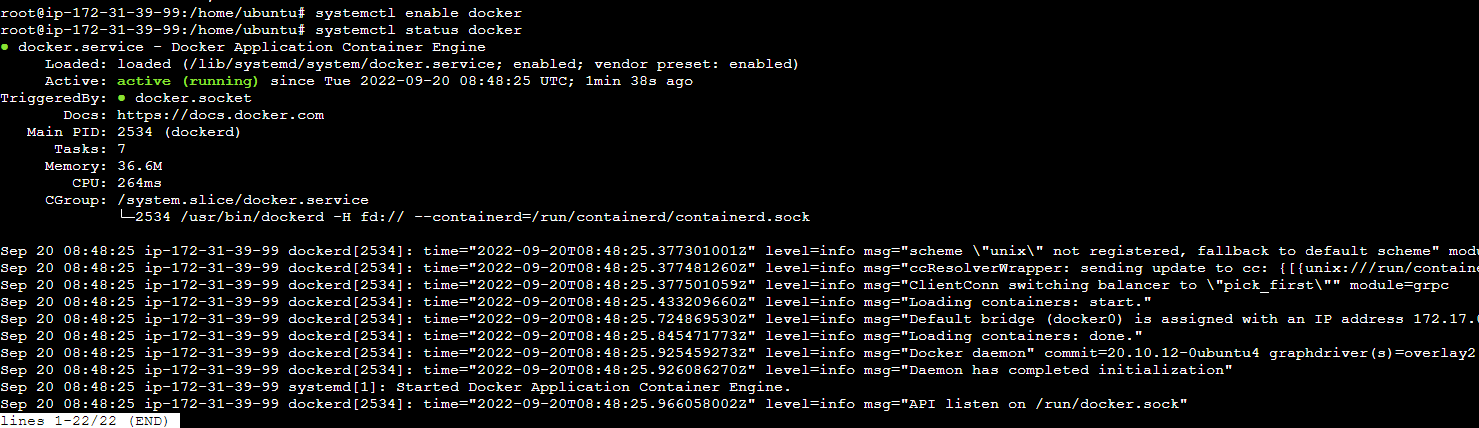
2. Connect instance with the help of PuTTY.



3. Check the docker version



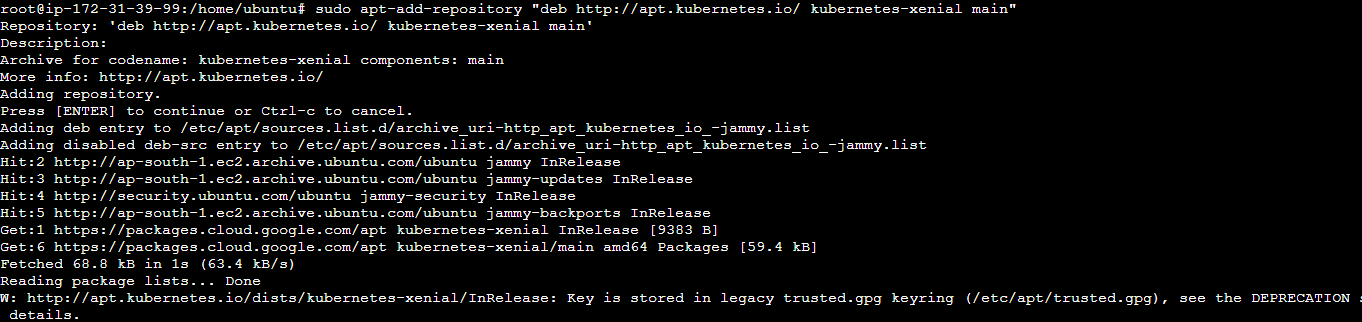
4. Enable and check Status



6. Using curl command :: curl -s hhtps://packages.cloud.google.com



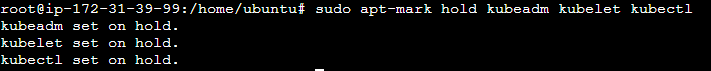
7. Add Repository:



8. Install kubernetes admin



9. To set on hold:



10. To check the kubeadm version :



**Conclusion:**

From this experiment, we have learnt about the Kubernetes Cluster and its Architecture and also we have installed and spun up the kubernetes on our linux machine by using AWS cloud service