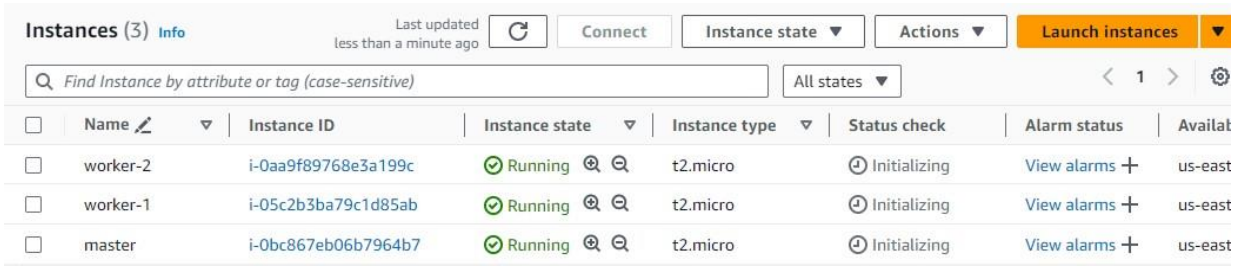
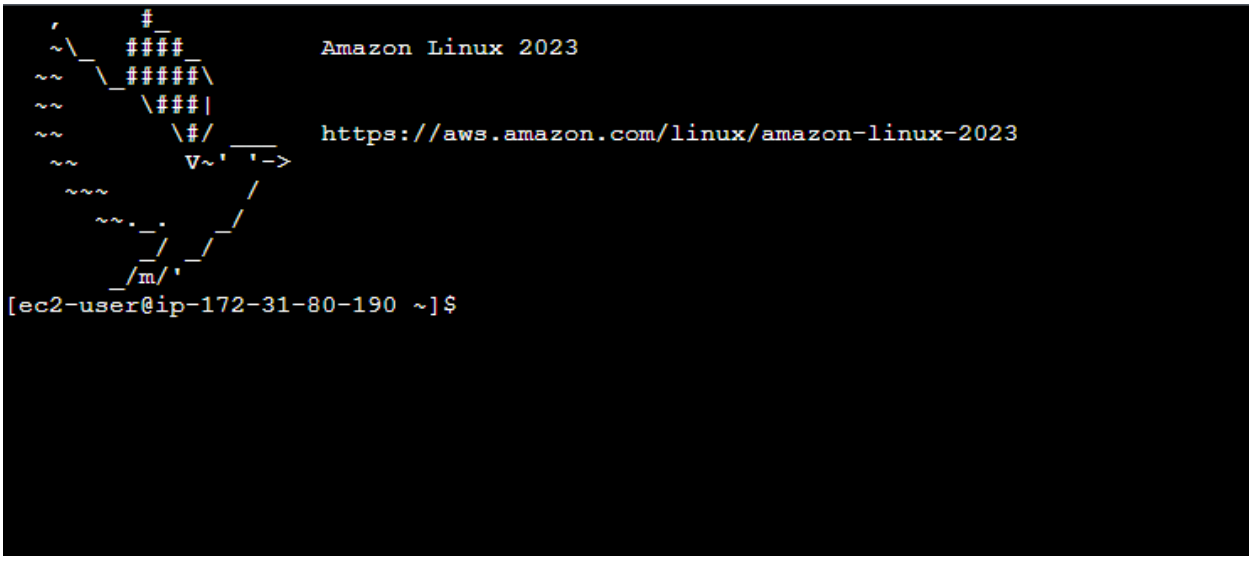
EXPERIMENT NO. 3

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

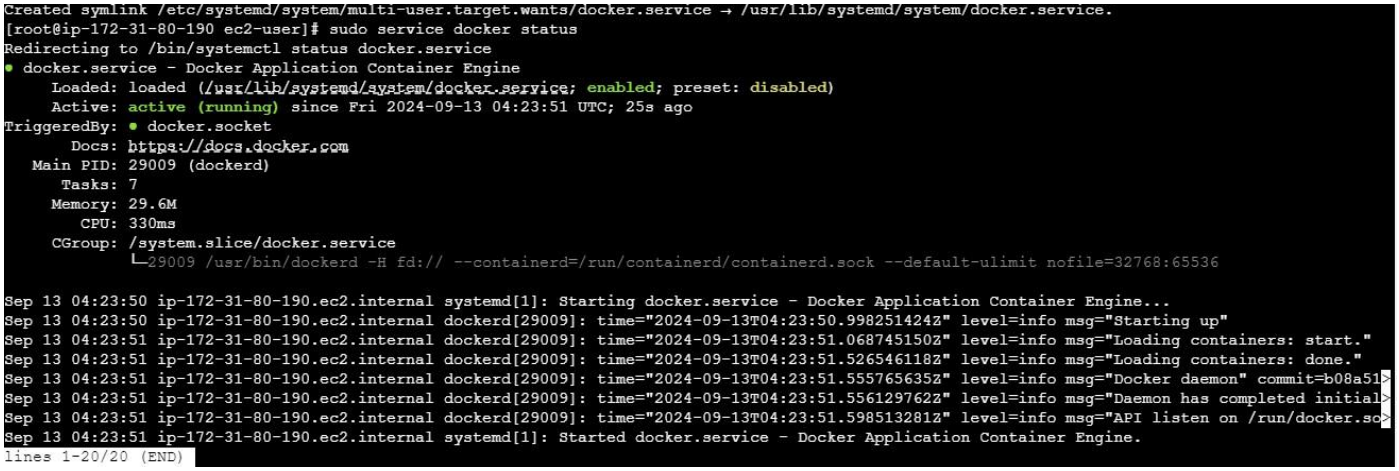
1. Launch three EC2 instances and assign unique names to each. Create and assign a key pair, ensuring SSH access is enabled.



2. SSH into each instance, open the terminal, and proceed to install Docker and Kubernetes.



3. Install Docker on all three instances. Follow the same installation steps for each one.



4.For Kubernetes installation, use kubeadm. Fetch the necessary installation commands and complete the setup on all instances.

```
  Installing       : kubeadm-1.31.1-150500.1.1.x86_64                                                    8/9
  Installing       : kubectl-1.31.1-150500.1.1.x86_64                                                    9/9
  Running scriptlet: kubectl-1.31.1-150500.1.1.x86_64                                                    9/9
  Verifying        : conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64                                         1/9
  Verifying        : libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64                                  2/9
  Verifying        : libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64                                 3/9
  Verifying        : libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64                                      4/9
  Verifying        : cri-tools-1.31.1-150500.1.1.x86_64                                                  5/9
  Verifying        : kubeadm-1.31.1-150500.1.1.x86_64                                                    6/9
  Verifying        : kubectl-1.31.1-150500.1.1.x86_64                                                    7/9
  Verifying        : kubelet-1.31.1-150500.1.1.x86_64                                                    8/9
  Verifying        : kubernetes-cni-1.5.1-150500.1.1.x86_64                                              9/9

Installed:
  conntrack-tools-1.4.6-2.amzn2023.0.2.x86_64                          cri-tools-1.31.1-150500.1.1.x86_64
  kubeadm-1.31.1-150500.1.1.x86_64                                     kubectl-1.31.1-150500.1.1.x86_64
  kubelet-1.31.1-150500.1.1.x86_64                                     kubernetes-cni-1.5.1-150500.1.1.x86_64
  libnetfilter_cthelper-1.0.0-21.amzn2023.0.2.x86_64                   libnetfilter_cttimeout-1.0.0-19.amzn2023.0.2.x86_64
  libnetfilter_queue-1.0.5-2.amzn2023.0.2.x86_64

Complete!
[root@ip-172-31-83-1 ec2-user]# sudo systemctl enable --now kubelet
Created symlink /etc/systemd/system/multi-user.target.wants/kubelet.service → /usr/lib/systemd/system/kubelet.service.
[root@ip-172-31-83-1 ec2-user]#
```

5. Verify the repositories are correctly configured

```
[root@ip-172-31-83-1 ec2-user]# yum repolist
repo id                                                    repo name
amazonlinux                                                Amazon Linux 2023 repository
kernel-livepatch                                           Amazon Linux 2023 Kernel Livepatch repository
kubernetes                                                 Kubernetes
[root@ip-172-31-83-1 ec2-user]#
```

6. Run the kubeadm command to initialize Kubernetes.

```
[root@ip-172-31-83-1 ec2-user]# kubeadm init
[init] Using Kubernetes version: v1.31.0
[preflight] Running pre-flight checks
        [WARNING FileExisting-socat]: socat not found in system path
        [WARNING FileExisting-tc]: tc not found in system path
error execution phase preflight: [preflight] Some fatal errors occurred:
        [ERROR NumCPU]: the number of available CPUs 1 is less than the required 2
        [ERROR Mem]: the system RAM (949 MB) is less than the minimum 1700 MB
[preflight] If you know what you are doing, you can make a check non-fatal with `--ignore-preflight-erro
To see the stack trace of this error execute with --v=5 or higher
[root@ip-172-31-83-1 ec2-user]#
```