**K8s Multi-Node on Multi-Cloud**



**Problem Statement:**

* Here We need to configure K8s Multi-Node Cluster on Multi-Cloud.
* For Multi-Cloud I am using AWS, Azure.
* AWS for 1 master, 1 Node.
* Azure for 1 Node.

**Solution: First we need to launch instances in AWS, Azure, for provisioning, I am using Terraform.**

**Prerequisite:**

* Have an AWS account
* Have an Azure account
* AWS CLIv2 installed and configured
* Azure CLI installed and configured

**Step-1)Launching Instances in AWS Azure**

* Let’s write code for this

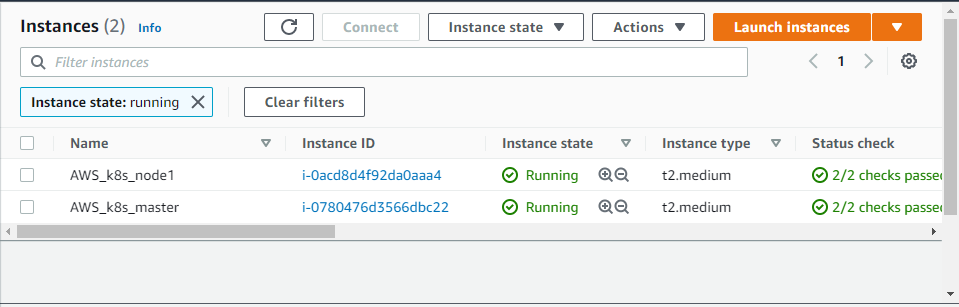


terraform init  
terraform apply

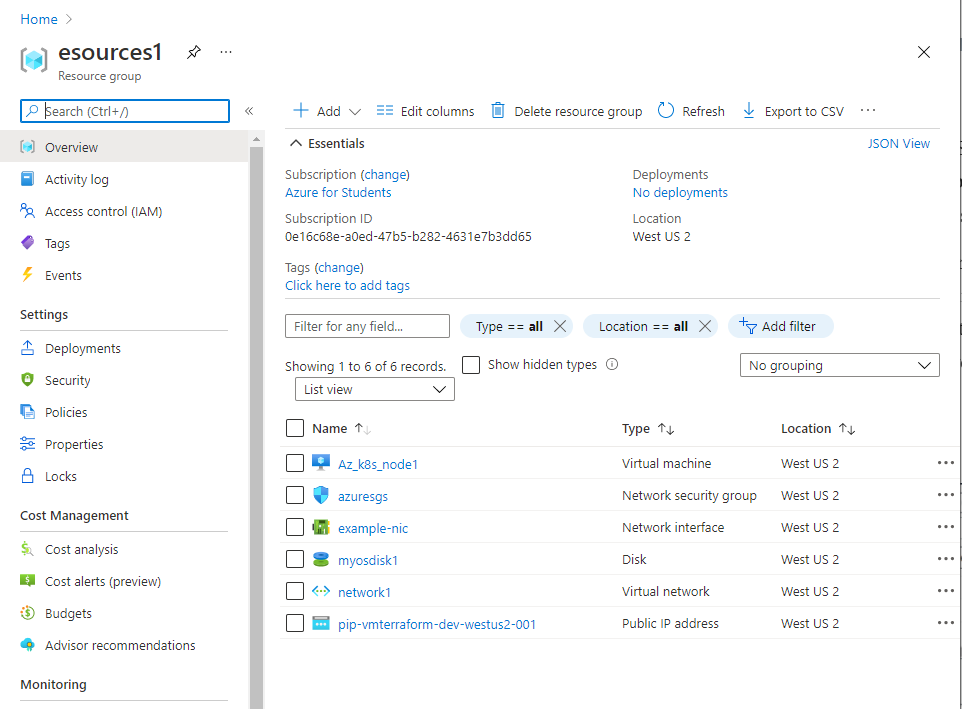
**Explanation:**

* Using resource AWS for launching instances in AWS 1 for master 1 for node, **instance\_type=t2.medium**(2vCPU,4GiB RAM) is required, I have already created SG’s, keys, so I used their values and os type is Amazon Linux 2.
* Using resource azurerm for launching 1 instance, in the azure first we need to make resources and after all, things should be in resources, in this, I am using rehl8 with**vm\_size=Standard\_B2s**(2vCPU,4GiB RAM), for logging in VM I have already created one user and assigned password

**Output**



AWS instance launched

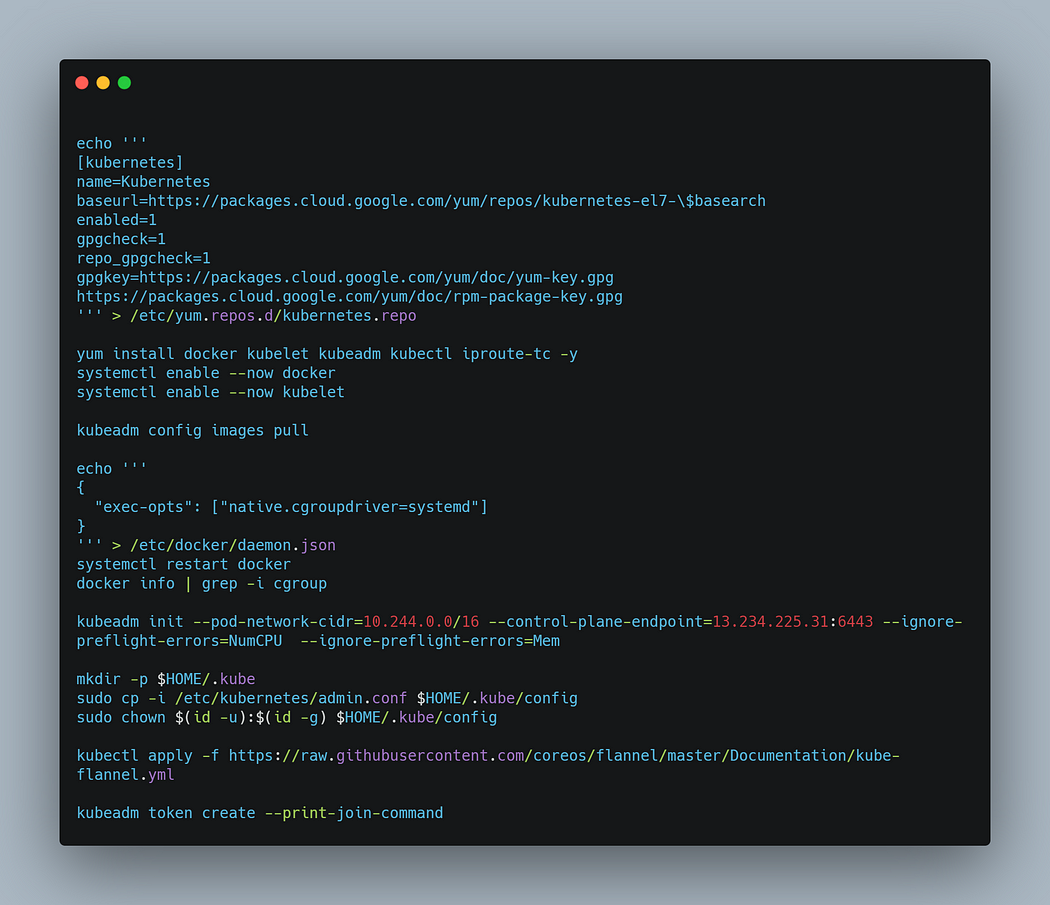


Azure VM launched

**Note: For configuring k8s Multi-Node Cluster I am using a bash script and make sure you’re login into root accunt**

**Step-2)Configure K8s Master**

* Master Node launched in AWS.
* Go and log in to that VM.
* For Configuring I am making a bash script

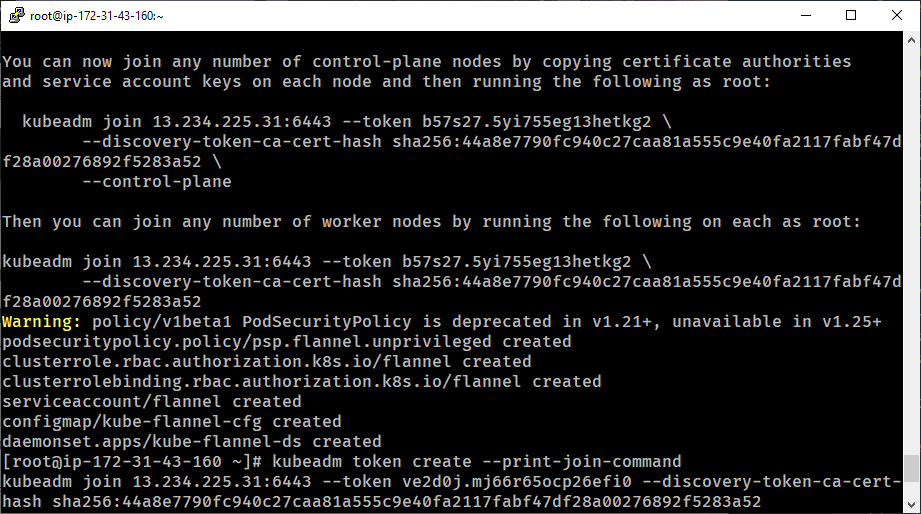


bash script.sh

**Explanation:**

* For installing kubelet kubeadm kubectl, first, we need to set up a repo for this, so in the first setup, I am configuring a repo for this.
* After that installing kubelet, kubeadm, kubectl, docker.
* Start and Enable docker and kubelet service.
* In the master, we need some images to use kubeadm config images pull
* After that, we need to change the docker cgroupdriver into systemd
* While running preflight the main thing is we need to associate the token to the public IP of instance, so that any of the other nodes can easily connect, so for this use **— control-plane-endpoint=VM\_PUBLIC\_IPv4:6443**
* After this make dir for Kube config files and give permission to them.
* The final step apply flannel.

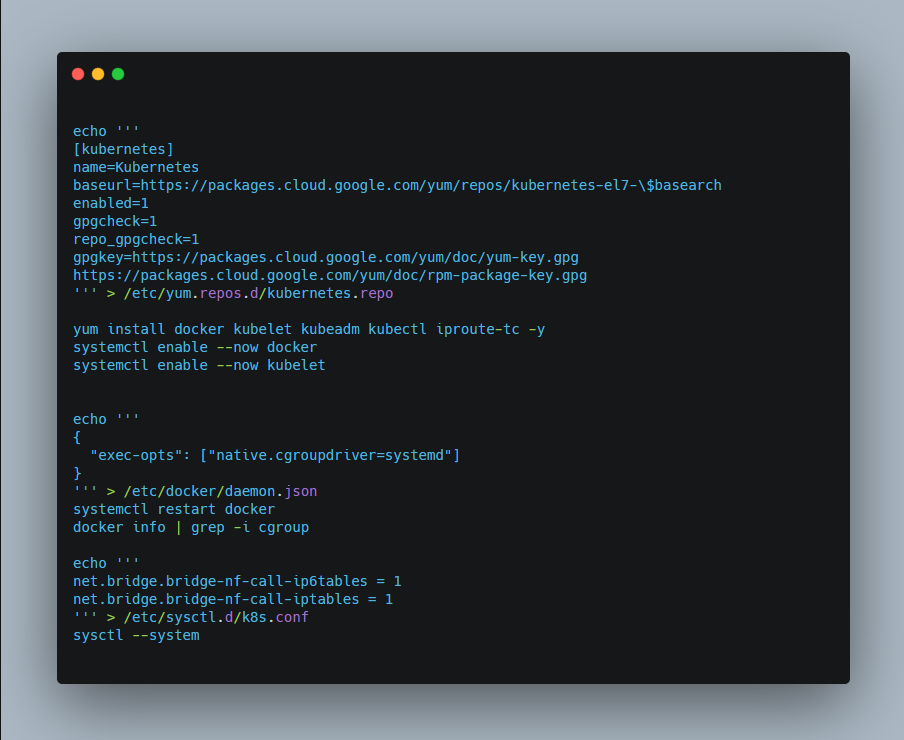
**Output**



AWS k8s Master

**Step-3)Configure Slave Nodes in AWS, Azure.**

* **Let’s configure the AWS Node**
* Again for this, I am going to write a bash script.



bash script.sh

**Explanation:**

* In the Slave Nodes again we need to configure repo for Kubernetes for installing kubelet kubeadm kubectl.
* After this install kubelet, kubeadm, kubectl, docker,
* Start and enable the service of kubelet, docker.
* Change cgroupdriver of docker into systemd
* Only one thing is change i,e we need to configure IP tables.
* One last thing we need to run the join token command

kubeadm join VM\_PUBLIC\_IPv4:6443 --token TOKEN --discovery-token-ca-cert-hash CERT\_HASH

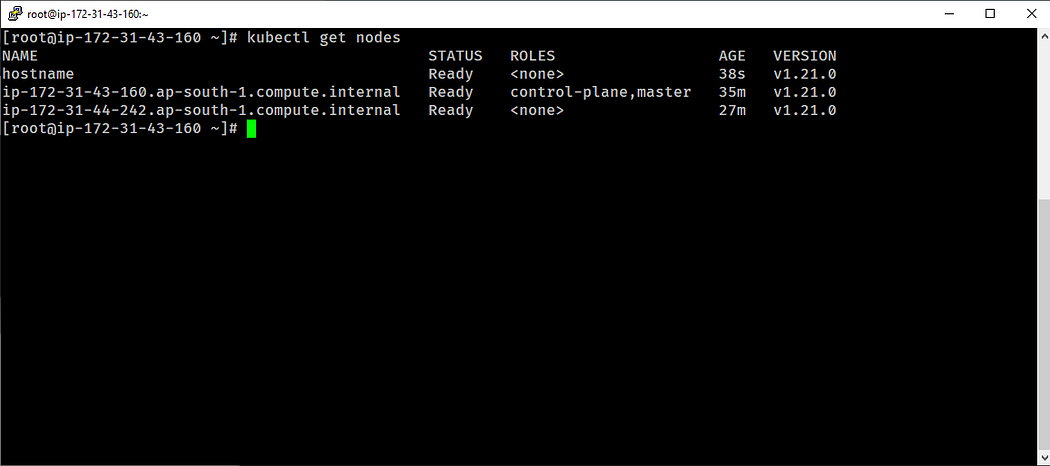
* **Now configure the Azure Node**
* Again for this, I am going to write a bash script.



* In this VM os is rehl8 and in rehl8 first, we need to setup the repo for docker and Kubernetes repo for installing kubelet, kubeadm, kubectl, docker,
* That’s it all the steps are the same as the AWS Slave node
* Last we need to run the token join command provided by the master Node.

kubeadm join VM\_PUBLIC\_IPv4:6443 --token TOKEN --discovery-token-ca-cert-hash CERT\_HASH

**Final Output**



* As you can see the first node is Azure Node
* The second one is the AWS master node
* and the third one is the AWS Slave node