

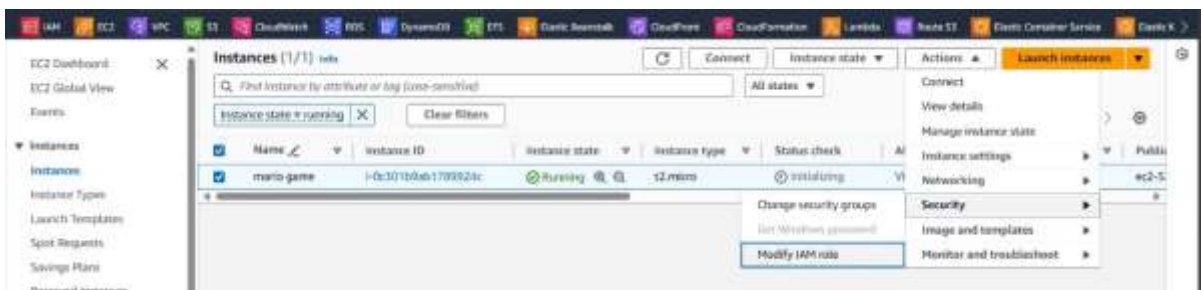
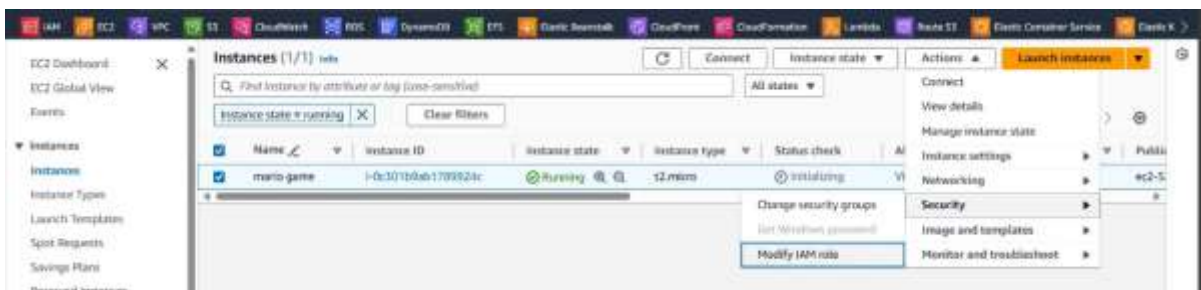
TASK = Deployment mario game

NAME = Akshay choudhary

Step 1

- Create an EC2 instance with ubuntu image.
- Create an IAM role for EC2 & give full access.
- Attache the polices (EC2 Adminstratin)

The screenshot shows the 'Launch an instance' page in the AWS Management Console. The 'Name and tags' section has 'mario game' entered in the Name field. The 'Application and OS Images (Amazon Machine Image)' section shows a search bar with the text 'Search our full catalog including 1000s of application and OS images'. The 'Summary' section on the right shows the following configuration: Number of instances: 1; Software image (AMI): Canonical, Ubuntu, 22.04 LTS, ...; Virtual server type (instance type): t2.micro; Firewall (security group): New security group; Storage (volumes): 1 volume(s) - 8 GiB. A 'Free tier' banner indicates that the first year includes 750 hours of t2.micro. The 'Launch Instance' button is highlighted in orange.



Step2

- Launch the instance
- #sudo apt update

- Sudo su
- Git clone - <http://github.com/Aj7Ay/k8s-mario>

```
ubuntu@ip-172-31-32-74:~$ sudo su
root@ip-172-31-32-74:/home/ubuntu# git clone https://github.com/awsanuragkadu/k8s-mario.git
Cloning into 'k8s-mario'...
remote: Enumerating objects: 41, done.
remote: Counting objects: 100% (41/41), done.
remote: Compressing objects: 100% (25/25), done.
remote: Total 41 (delta 14), reused 41 (delta 14), pack-reused 0
Receiving objects: 100% (41/41), 10.37 KiB | 1.30 MiB/s, done.
Resolving deltas: 100% (14/14), done.
root@ip-172-31-32-74:/home/ubuntu#
```

- Cd /k8s-mario

```
root@ip-172-31-32-74:/home/ubuntu# ls
k8s-mario
root@ip-172-31-32-74:/home/ubuntu# cd k8s-mario/
root@ip-172-31-32-74:/home/ubuntu/k8s-mario# chmod +x script.sh
root@ip-172-31-32-74:/home/ubuntu/k8s-mario# sh script.sh
```

Step3

- Script.sh (install the some file terraform, awscli kubectl client)
- Docker install docker command
- Use command #apt install docker.io

Step4

Check the all installed file in version

- aws --version
- docker --version
- terraform --version
- kubectl version --client

```
root@ip-172-31-32-74:/home/ubuntu/k8s-mario# aws --version
aws-cli/2.15.35 Python/3.11.8 Linux/6.5.0-1014-aws exe/x86_64.ubuntu.22 prompt/off
root@ip-172-31-32-74:/home/ubuntu/k8s-mario# terraform --version
Terraform v1.7.5
on linux_amd64
root@ip-172-31-32-74:/home/ubuntu/k8s-mario# kubectl version - client
error: extra arguments: [- client]
root@ip-172-31-32-74:/home/ubuntu/k8s-mario# kubectl version -client
error: unknown shorthand flag: 'c' in -client
See 'kubectl version --help' for usage.
root@ip-172-31-32-74:/home/ubuntu/k8s-mario# kubectl version --client
Client Version: v1.29.3
Kustomize Version: v5.0.4-0.20230601165947-6ce0bf390ce3
```

Step 4

Change the directory

- # cd EKE-TF
- Ls

```
root@ip-172-31-32-74:/home/ubuntu/k8s-mario# cd EKS-TF/
Command 'cdc' not found, but there are 24 similar ones.
root@ip-172-31-32-74:/home/ubuntu/k8s-mario# cd EKS-TF/
root@ip-172-31-32-74:/home/ubuntu/k8s-mario/EKS-TF#
```

Step5

- Create a s3 bucket. name = Akshay& change the backend file
- & change the region name in the perticular region

Step 6

- Provider.tf file in the change in region
- & change the region name in the perticular region

Step 7

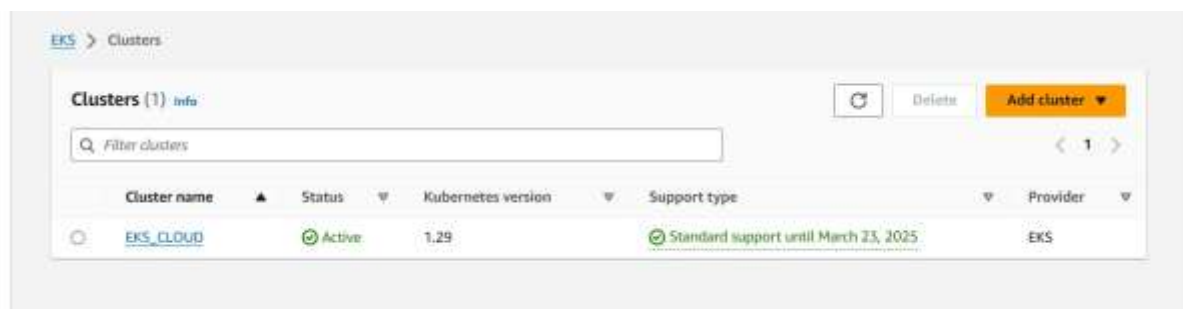
Installation in terafrom command

- Terraform init
- Terraform plan
- Terraform validate
- Terraform apply -auto-approve

```
Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
root@ip-172-31-32-74:/home/ubuntu/k8s-mario/EKS-TF# aws eks update-kubeconfig --name EKS_CLOUD --region ap-northeast-2
Added new context arn:aws:eks:ap-northeast-2:381492218806:cluster/EKS_CLOUD to /root/.kube/config
root@ip-172-31-32-74:/home/ubuntu/k8s-mario/EKS-TF# cd ..
root@ip-172-31-32-74:/home/ubuntu/k8s-mario#
```

Step 8

- Check the Aws cluster &node the particular region the is create or not



- Update the Kubernetes configuration

```
Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
root@ip-172-31-32-74:/home/ubuntu/k8s-mario/EKS-TF# aws eks update-kubeconfig --name EKS_CLOUD --region ap-northeast-2
Added new context arn:aws:eks:ap-northeast-2:381492218806:cluster/EKS_CLOUD to /root/.kube/config
root@ip-172-31-32-74:/home/ubuntu/k8s-mario/EKS-TF#
```

- Cd ..

Step 9

Let's apply the deployment and service

Deployment

- `kubectl apply -f deployment.yaml`
#to check the deployment
`kubectl get all`

Now let's apply the service

Service

- `kubectl apply -f service.yaml`
`kubectl get all`

step 10

Now let's describe the service and copy the LoadBalancer Ingress

- `kubectl describe service mario-service`

```
root@ip-172-31-32-74:/home/ubuntu/k8s-mario# kubectl get all
NAME                                READY    STATUS    RESTARTS   AGE
pod/mario-deployment-78cbc65cb-kr8tt 1/1      Running   0           74s
pod/mario-deployment-78cbc65cb-tkw6k 1/1      Running   0           74s

NAME                TYPE        CLUSTER-IP    EXTERNAL-IP    PORT(S)    AGE
service/kubernetes  ClusterIP   10.100.0.1    <none>         443/TCP    19m

NAME                                READY    UP-TO-DATE    AVAILABLE    AGE
deployment.apps/mario-deployment  2/2      2              2            74s

NAME                                DESIRED    CURRENT    READY    AGE
replicaset.apps/mario-deployment-78cbc65cb 2          2          2        74s
root@ip-172-31-32-74:/home/ubuntu/k8s-mario#
```

```

deployment.apps/mario-deployment 2/2 2 2 100%
NAME                                DESIRED  CURRENT  READY  AGE
replicaset.apps/mario-deployment-78cbc65cb 2        2        2      108s
root@ip-172-31-32-74:/home/ubuntu/k8s-mario# kubectl describe service mario-service
i-0c301b9ab1789924c (mario game)

```

Step 11

- This loadbalancer ingress link copy & past on browser
- & out put on the game mario

```

IP: 10.100.52.82
IPs: 10.100.52.82
LoadBalancer Ingress: acc7287a0df0646feb2d7be24a8f1d97-1547344463.ap-northeast-2.elb.amazonaws.com
Port: <unset> 80/TCP

```

- Out put



- Remove the services and deployment first
- `kubectl get all`
`kubectl delete service mario-service`
`kubectl delete deployment mario-deployment`
- `terraform destroy --auto-approve`