K8 EKS SETUP ON AWS CLOUD

Please complete the DAY1 Session before this HANDSON https://youtu.be/k5hLOoANjWA

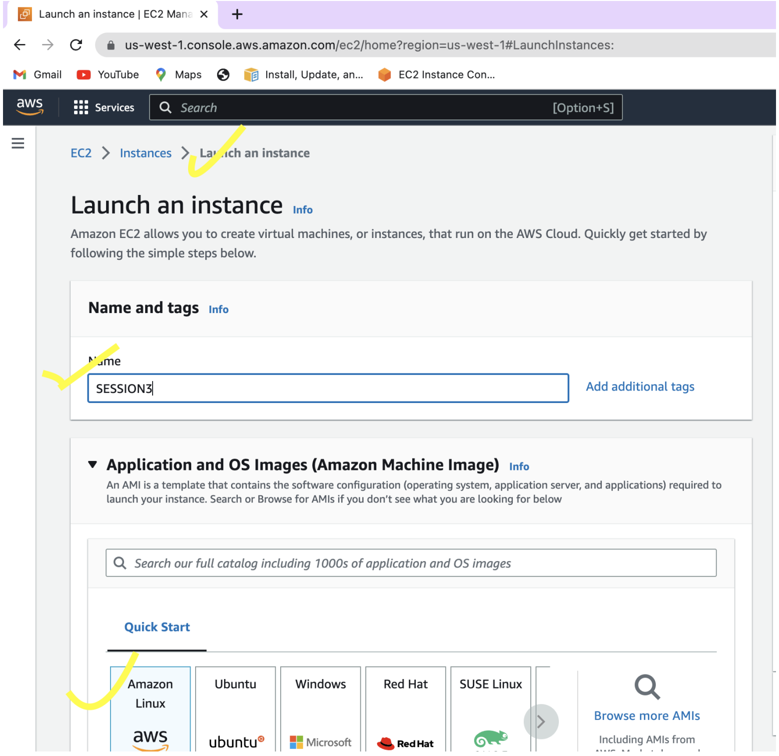
NOTE – COST WILL BE INCURRED FOR AWS EKS SETUP SO MAKE SURE YOU ARE DOING THE PROJECT ON YOUR ON INTEREST

Step 1 – LOGIN to the AWS Console

https://aws.amazon.com/console/

STEP 1.1 - Create the IAM User

Step 2 – Select EC2 LINUX 2 AMI and create T2.MEDIUM INSTANCE IN USE-WEST-1



us-west-1.console.aws.amazon.com/ec2/home?region=us-west-1#LaunchInstances:

EC2 > Instances > Launch an instance

Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name: SESSION3

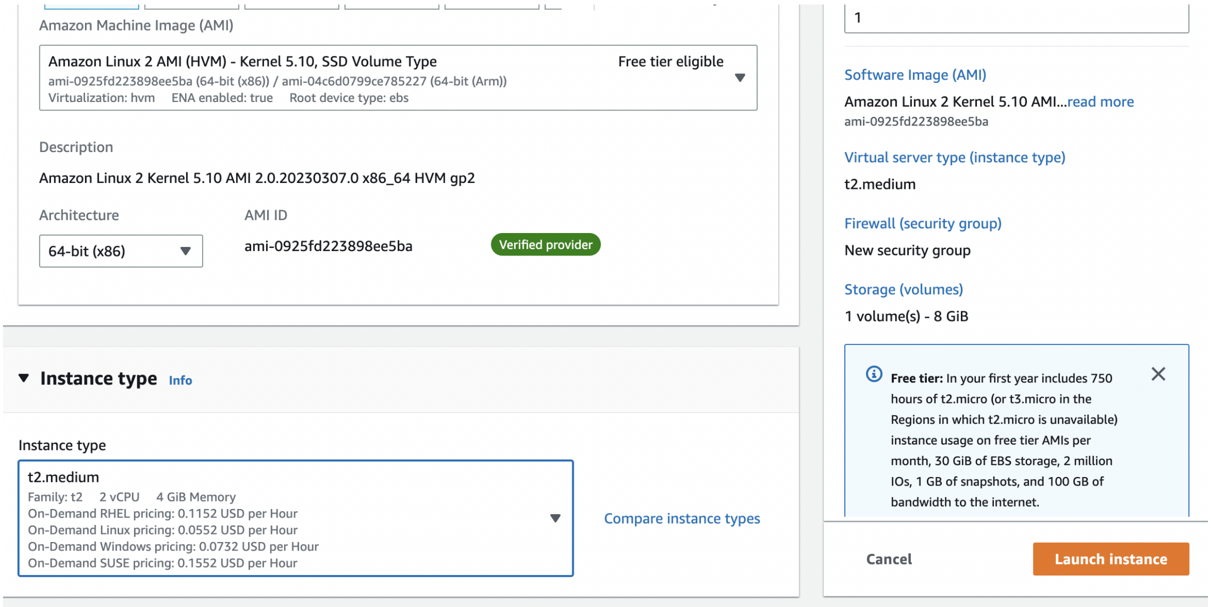
Add additional tags

Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

STEP3 – Select AMAZON LINUX 2 AMI



Amazon Machine Image (AMI)

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type — Free tier eligible
ami-0925fd223898ee5ba (64-bit (x86)) / ami-04c6d0799ce785227 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Amazon Linux 2 Kernel 5.10 AMI 2.0.20230307.0 x86_64 HVM gp2

Architecture: 64-bit (x86)

AMI ID: ami-0925fd223898ee5ba

Verified provider

Instance type

Instance type: t2.medium
Family: t2 2 vCPU 4 GiB Memory
On-Demand RHEL pricing: 0.1152 USD per Hour
On-Demand Linux pricing: 0.0552 USD per Hour
On-Demand Windows pricing: 0.0732 USD per Hour
On-Demand SUSE pricing: 0.1552 USD per Hour

1

Software Image (AMI)
Amazon Linux 2 Kernel 5.10 AMI...read more
ami-0925fd223898ee5ba

Virtual server type (instance type)
t2.medium

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Step 4 – Install all tools Prerequisites

- Install Git

yum install git -y

- Install Java

yum install java -y

- Install Maven

cd /opt/
wget http://mirrors.estointernet.in/apache/maven/maven-3/3.6.3/binaries/apache-maven-3.6.3-bin.tar.gz

```
tar xvzf  apache-maven-3.6.3-bin.tar.gz
vi /etc/profile.d/maven.sh
export MAVEN_HOME=/opt/apache-maven-3.6.3
export PATH=$PATH:$MAVEN_HOME/bin
```

- Install Docker

```
yum install docker -y
usermod -aG docker jenkins [ Add jenkins user to docker group ]
systemctl start docker
systemctl enable docker
```

- Install Python

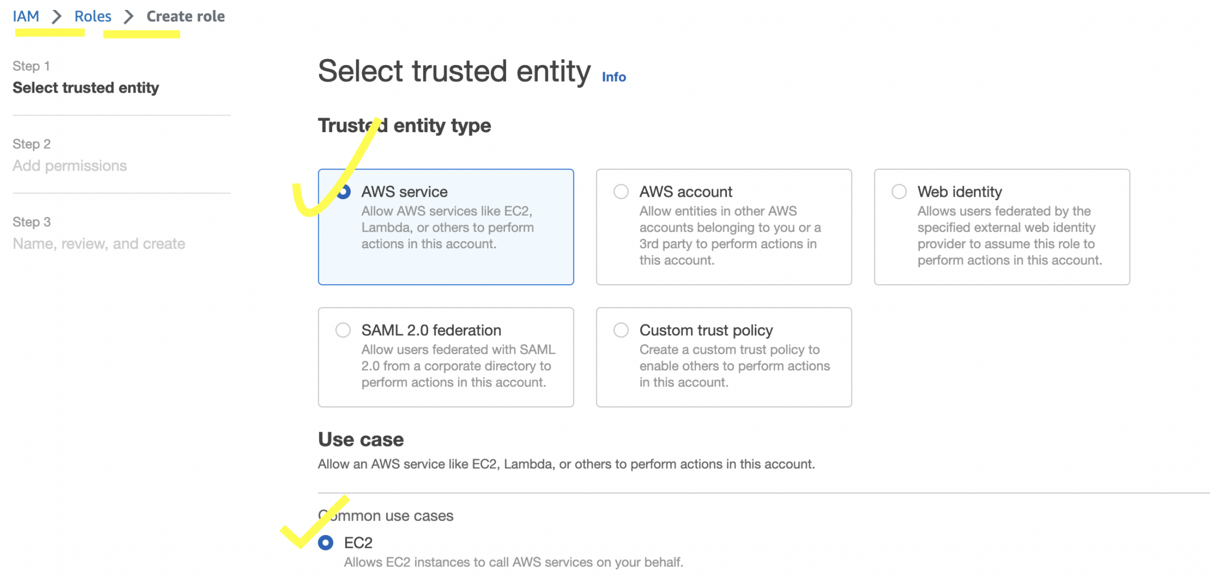
```
yum install python3 -y
```

- Install Ansible

```
amazon-linux-extras install ansible2 -y
```

Step4.1 - ATTACH THE IAM ROLE

Go to IAM -> CLICK CREATE NEW IAM ROLE -> SELECT EC2 -> CLICK ON ADMINISTRATOR ACCESS -> CREATE ROLE



IAM > Roles > Create role

Step 1
Select trusted entity

Step 2
Add permissions

Step 3
Name, review, and create

Select trusted entity Info

Trusted entity type

● AWS service
Allow AWS services like EC2, Lambda, or others to perform actions in this account.

○ AWS account
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.

○ Web identity
Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.

○ SAML 2.0 federation
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.

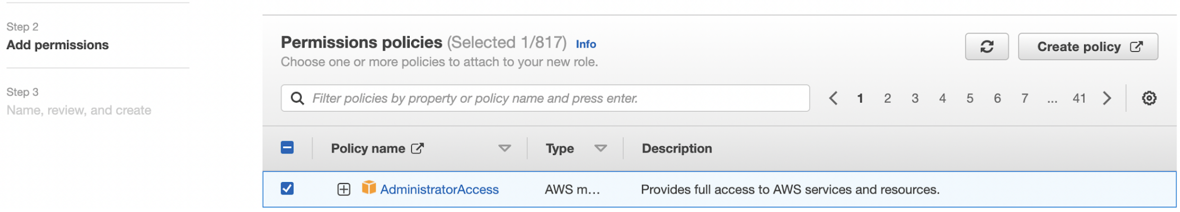
○ Custom trust policy
Create a custom trust policy to enable others to perform actions in this account.

Use case

Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

Common use cases

● EC2
Allows EC2 instances to call AWS services on your behalf.



Step 2
Add permissions

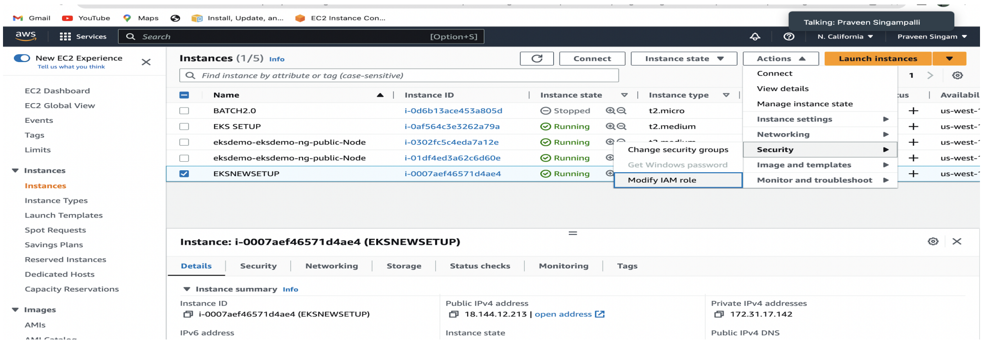
Step 3
Name, review, and create

Permissions policies (Selected 1/817) Info

Choose one or more policies to attach to your new role.

☐	Policy name	Type	Description
☑	AdministratorAccess	AWS m...	Provides full access to AWS services and resources.

Step 4.2 - Go toEC2 instance you have created -> Click on ACTIONS -> SECURITY -> MODIFY IAM ROLE -> ATTACH YOUR NEW ROLE



Instances (1/5) Info

	Name	Instance ID	Instance state	Instance type	Availabil
☐	BATCH2.0	i-0d6b13ace453a805d	Stopped	t2.micro	us-west-
☐	EKS SETUP	i-0af564c3e3262a79a	Running	t2.medium	us-west-
☐	eksdemo-eksdemo-ng-public-Node	i-0302fc5c4eda7a12e	Running		us-west-
☐	eksdemo-eksdemo-ng-public-Node	i-01df4ed3a62c6d60e	Running		us-west-
☑	EKSNEWSETUP	i-0007aef46571d4ae4	Running		us-west-

Instance: i-0007aef46571d4ae4 (EKSNEWSETUP)

Details | Security | Networking | Storage | Status checks | Monitoring | Tags

Instance summary Info

Instance ID
i-0007aef46571d4ae4 (EKSNEWSETUP)

Public IPv4 address
18.144.12.213 | open address

Private IPv4 addresses
172.31.17.142

IPv6 address

Instance state

Public IPv4 DNS

Step 5 – INSTALL SETUP FOR EKS

- Install kubectl

```
curl -o kubectl
https://amazon-eks.s3-us-west-2.amazonaws.com/1.14.6/2019
-08-22/bin/linux/amd64/kubectl
chmod +x ./kubectl
mkdir -p $HOME/bin
cp ./kubectl $HOME/bin/kubectl
export PATH=$HOME/bin:$PATH
echo 'export PATH=$HOME/bin:$PATH' >> ~/.bashrc
source $HOME/.bashrc
kubectl version --short –client
```

- Install eksctl

```
curl --silent --location
"https://github.com/weaveworks/eksctl/releases/latest/downlo
ad/eksctl_$(uname -s)_amd64.tar.gz" | tar xz -C /tmp

sudo mv /tmp/eksctl /usr/bin
eksctl version
```

- MASTER Cluster creation [Change the master cluster name eksdemo **as per your wish and select region as us-west-1]**

```
eksctl create cluster --name=eksdemo \
                --region=us-west-1 \
                --zones=us-west-1b,us-west-1a \
                --without-nodegroup
```

- Add Iam-Oidc-Providers

```
eksctl utils associate-iam-oidc-provider \
    --region us-west-1 \
    --cluster eksdemo \
    --approve
```

- WORKER NODE Create node-group [Change the PEM key ssh-public-key to your key]

```
eksctl create nodegroup --cluster=eksdemo \
                   --region=us-west-1 \
                   --name=eksdemo-ng-public \
                   --node-type=t2.medium \
                   --nodes=2 \
                   --nodes-min=2 \
                   --nodes-max=4 \
                   --node-volume-size=10 \
                   --ssh-access \
                   --ssh-public-key=key-test \
                   --managed \
                   --asg-access \
                   --external-dns-access \
                   --full-ecr-access \
                   --appmesh-access \
                   --alb-ingress-access
```

STEP 6 – DELETE NODE AND THEN THE CLUSTER

DELETE NODE

```
eksctl delete nodegroup --cluster=eksdemo
--region=us-west-1 --name=eksdemo-ng-public
```

DELETE CLUSTER

```
eksctl delete cluster --name=eksdemo --region=us-west-1
```