

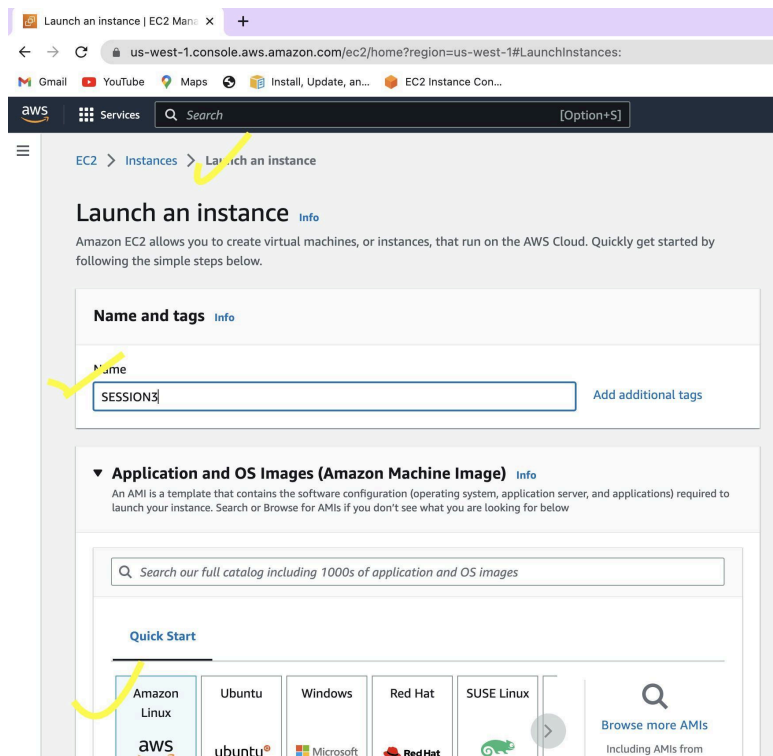
AWS EKS PROJECT SETUP CLOUD

NOTE – COST WILL BE INCURRED FOR AWS EKS SETUP SO MAKE SURE YOU ARE DOING THE PROJECT ON YOUR OWN INTEREST AND ALSO YOU CAN DO THE SAME PROJECT WITH MINIKUBE SETUP

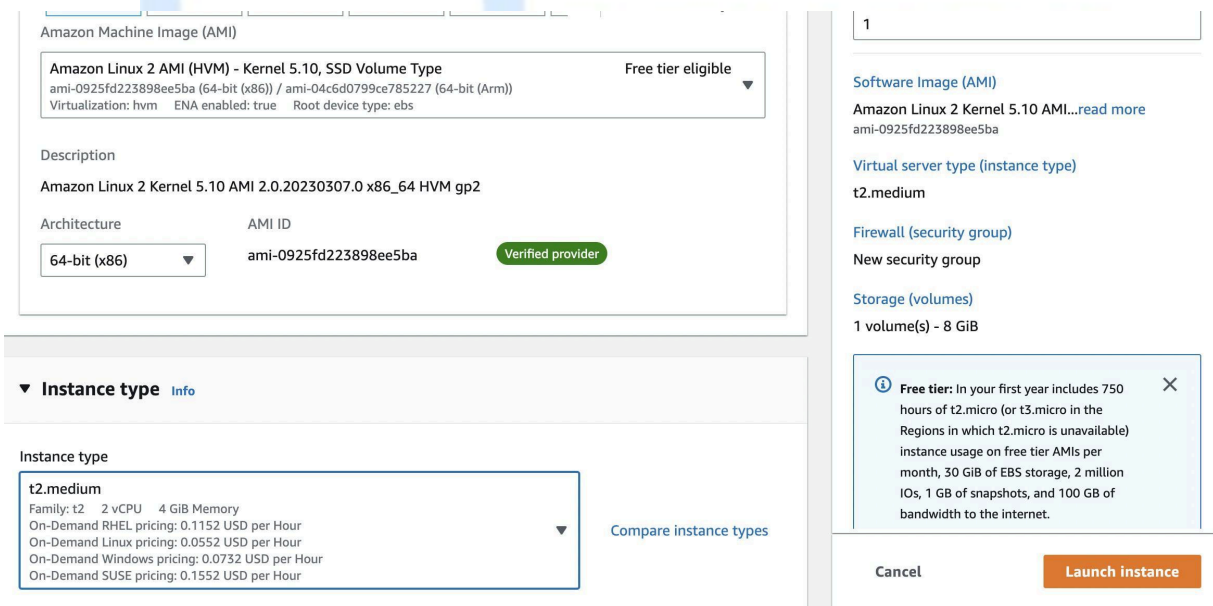
[REFER MINIKUBE SETUP Session 6 (Live 21st July 9am) Kubernetes HANDS-ON PART1 [BASICS] project

Step 1 – LOGIN to the AWS Console <https://aws.amazon.com/console/>

Step 2 – Select EC2 and create T2.MEDIUM INSTANCE IN US WEST1



STEP3 – Select AMAZON LINUX 2 AMI 30 GB



STEP3.1 – Make sure to create the PEM file in the EC2 instance with name key-test

Step 4 – Install all tools Pre-requisites

- Install Git

```
yum install git -y
```

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
```

OR

```
curl -O
https://s3.us-west-2.amazonaws.com/amazon-eks/1.24.11/2023-03-17/bin/linux
/ amd64/kubectl.sha256
chmod +x ./kubectl
mkdir -p $HOME/bin && cp ./kubectl $HOME/bin/kubectl && export
PATH=$PATH:$HOME/bin
echo 'export PATH=$PATH:$HOME/bin' >> ~/.bashrc
kubectl version --short --client
```

- Install eksctl

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

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

Step6 - 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.

Permissions policies (Selected 1/817) [Info](#)
Choose one or more policies to attach to your new role.

Q Filter policies by property or policy name and press enter.

	Policy name ↗	Type	Description
<input checked="" type="checkbox"/>	 AdministratorAccess	AWS m...	Provides full access to AWS services and resources.

STEP 7 –

Go to EC2 instance you have created -> Click on
ACTIONS -> SECURITY -> MODIFY IAM ROLE ->
ATTACH YOUR NEW ROLE

The screenshot shows the AWS Management Console interface. On the left, there's a navigation menu with options like 'EC2 Dashboard', 'EC2 Global View', 'Events', 'Tags', 'Limits', 'Instances', 'Instance Types', 'Launch Templates', 'Spot Requests', 'Savings Plans', 'Reserved Instances', 'Dedicated Hosts', 'Capacity Reservations', 'Images', 'AMIs', and 'AMI Catalog'. The main content area displays the 'Instances (1/5)' table. The table has columns for Name, Instance ID, Instance state, and Instance type. The instance 'EKSNEWSETUP' with ID 'i-0007aef46571d4ae4' is in a 'Running' state and is of type 't2.micro'. The 'Actions' menu for this instance is open, showing various options. The 'Security' option is selected, and a dropdown menu is visible with options like 'Change security groups', 'Get Windows password', and 'Modify IAM role'. Below the table, the details for the selected instance are shown, including the Instance ID, Public IPv4 address, Private IPv4 addresses, and Public IPv4 DNS.

Step 8 -

- **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-1c \
  --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 created in step 3.1**]

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

STEP 9 –

**** Clone the repository**

https://github.com/praveen1994dec/Custom_Resource_Definition.git

**** cd Custom_Resource_Definition/ and hit the below command**

```
kubectl apply -f crd.yml
```

**** Once the CRD is registered, verify that by running the**

```
kubectl api-resources | grep myplatform
```

**** Creating the custom resource**

```
kubectl apply -f cr.yml
```

**** Hit the below command**

```
kubectl get myp
```

STEP 10 – DELETE NODE AND THEN THE CLUSTER

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

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

