**Cloud Computing-MiniProject**

**Creating an EC2-Instance with specified requirements**

|  |
| --- |
| **Team Members:**  Mohammed Wadoodh |

# Overview:

1. We have to create an EC2-Instance with YAML/JSON coding.
2. We have to create an EC2-instance with console management and connecting to session manager.

### Requirements:

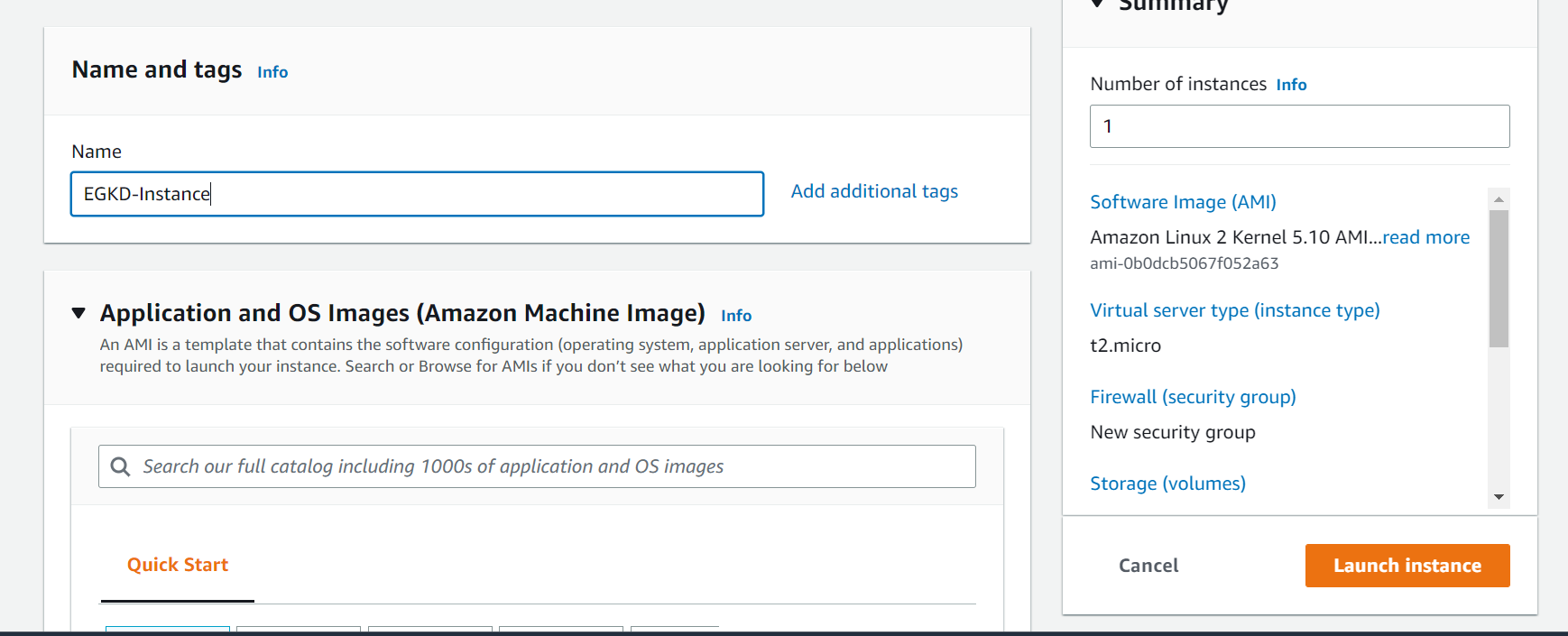
For both the way we have common requirements:

* AMI - Amazon linux
* Security group - Open 22 from 0.0.0.0
* EBS with 8 GB
* key pair - <youname>.pem
* Instance type: t2.micro

## Creating an EC2-Instance with management console and Connecting via session manager:

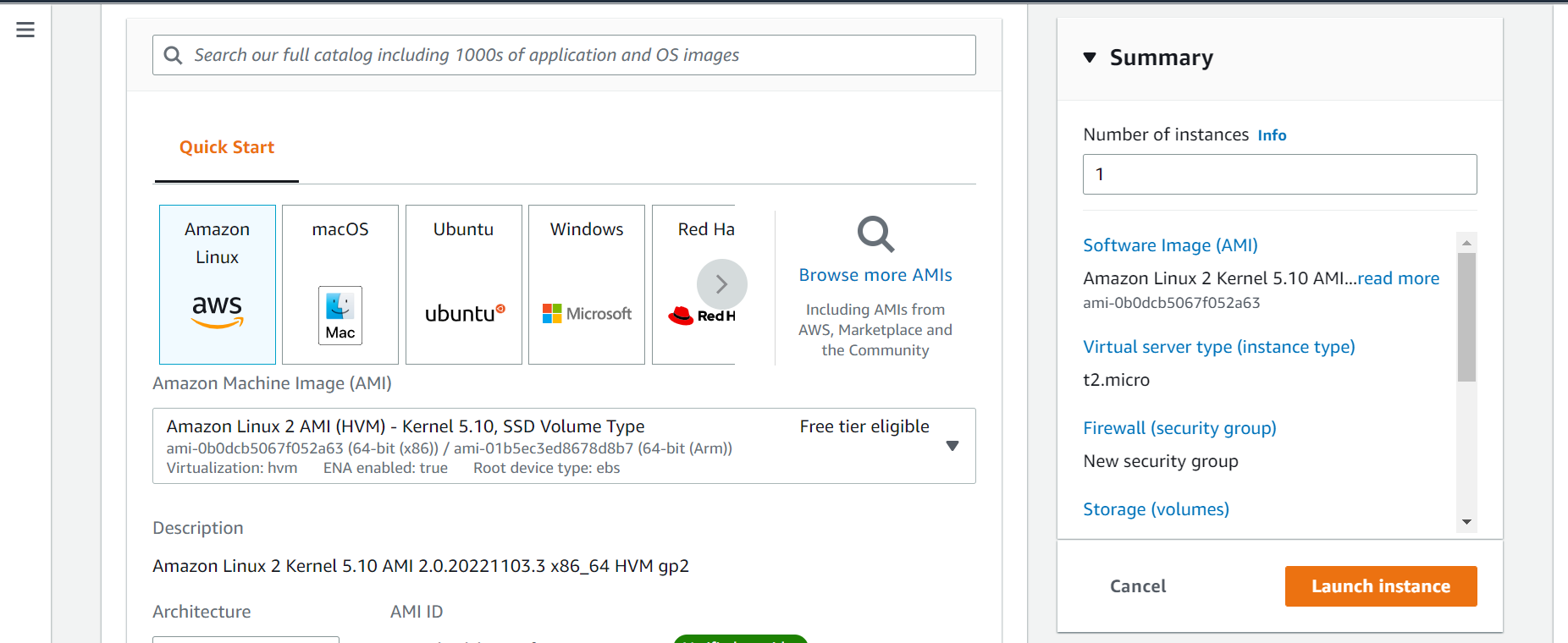
#### Step1:

Here the first step is launching instance and choosing an instance name



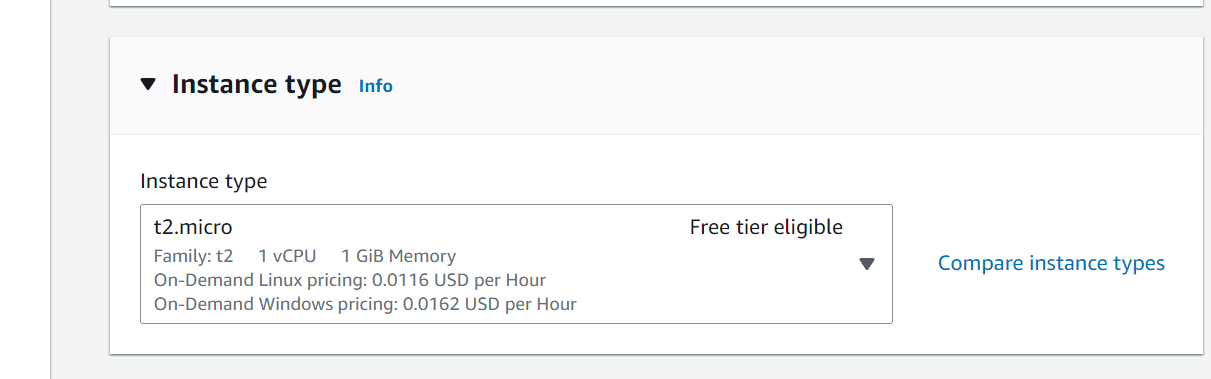
#### Step2:

As per the requirement we are choosing an Amazon Linux -AMI



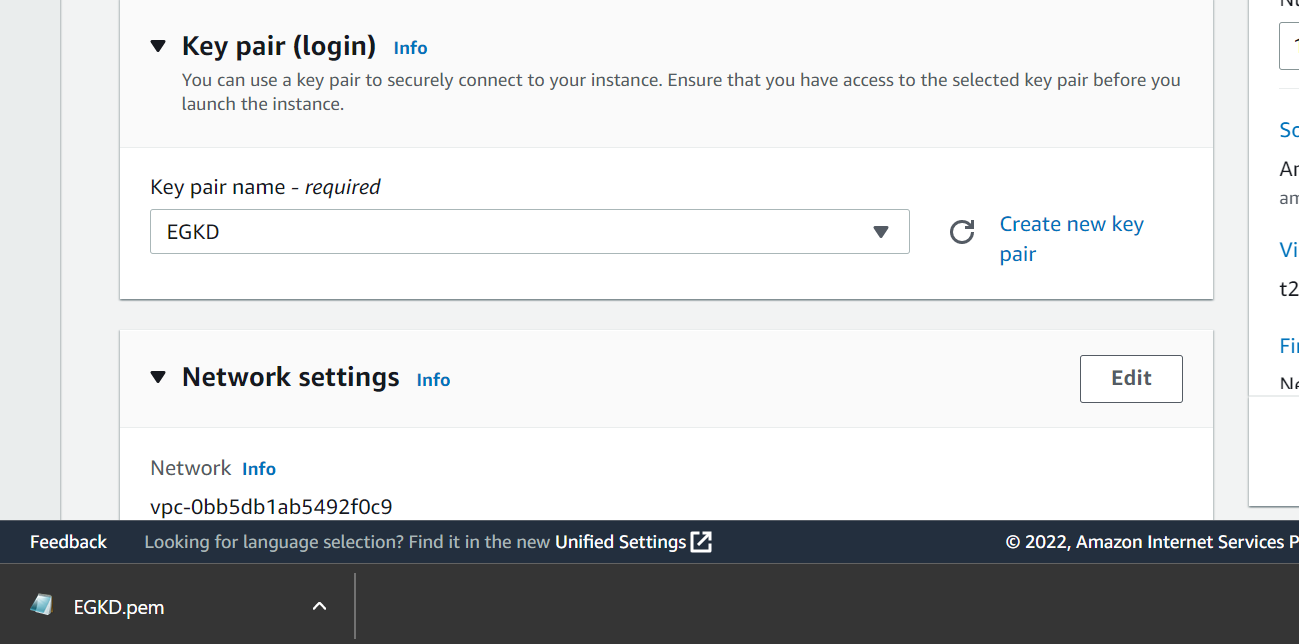
#### Step3:

As per the requirement instance type we are selecting as t2.micro



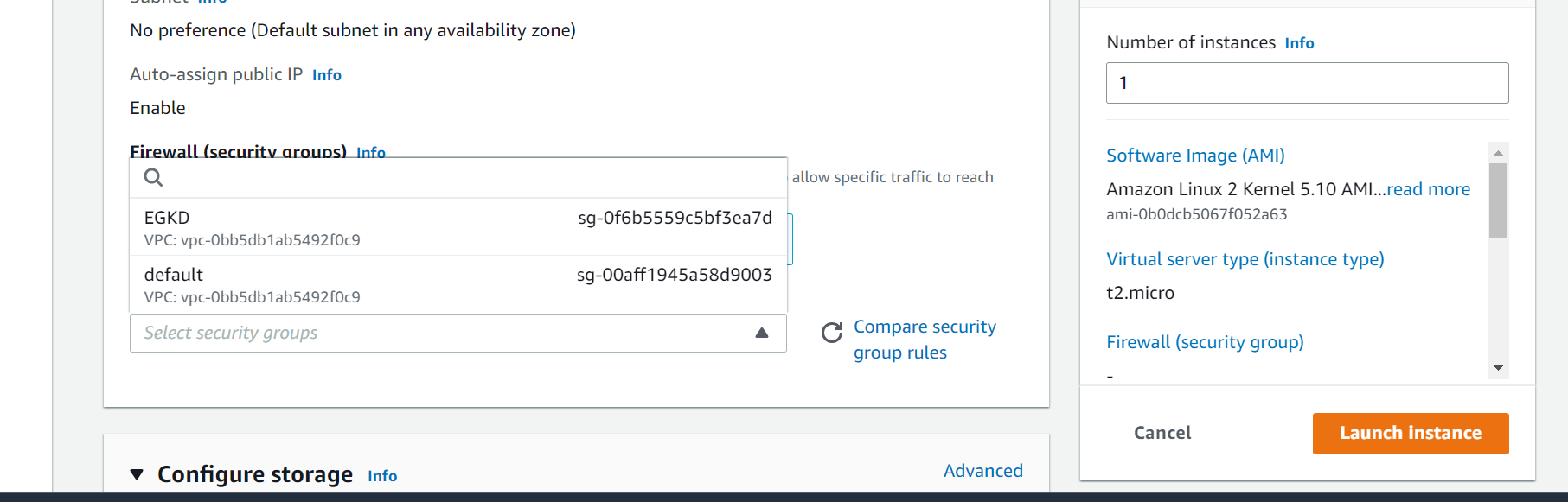
#### Step4:

Generating a key pair, with proper name with .pem extension. Once key pair is generated, we it will download automatically and we are storing in local.



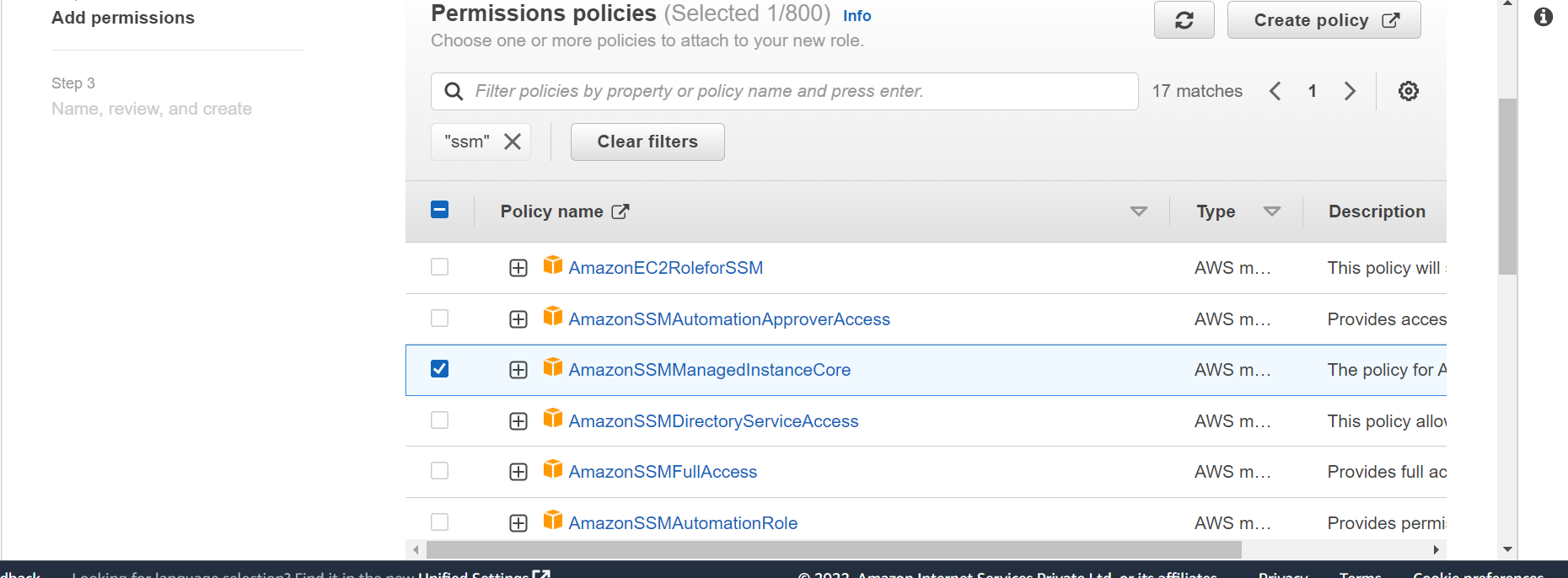
#### Step5:

Creating a security group and attaching to the instance.As per the requirement we have created a security group with port 22 allow any where



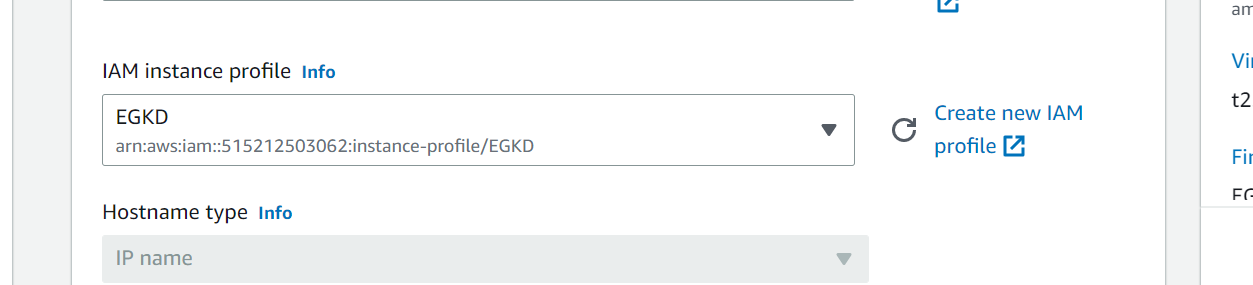
#### Step6:

As per the requirement, we have to connect to session manager with the instance. So creating an IAM role with that policy (AmazonSSMManagedInstanceCore)



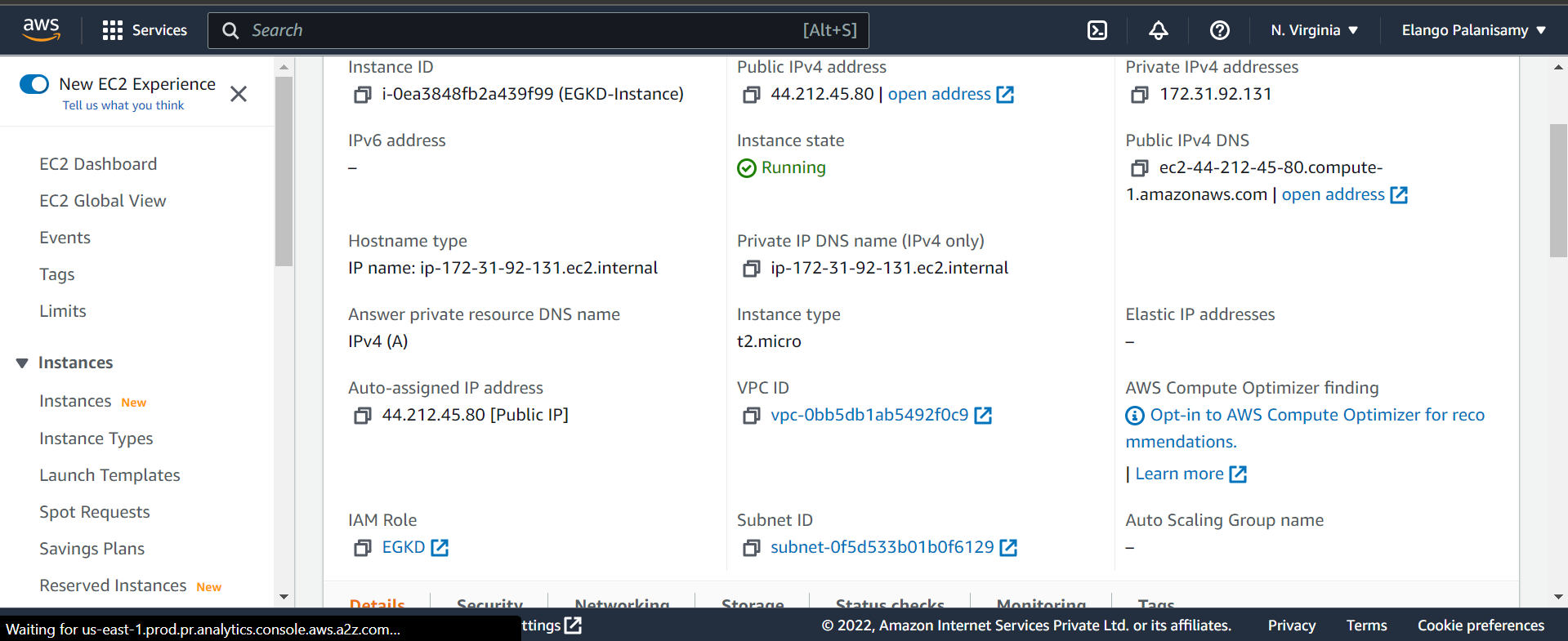
#### Step7:

Once the IAM role is created we have to attach that in to an instance that falls under Advanced Details.



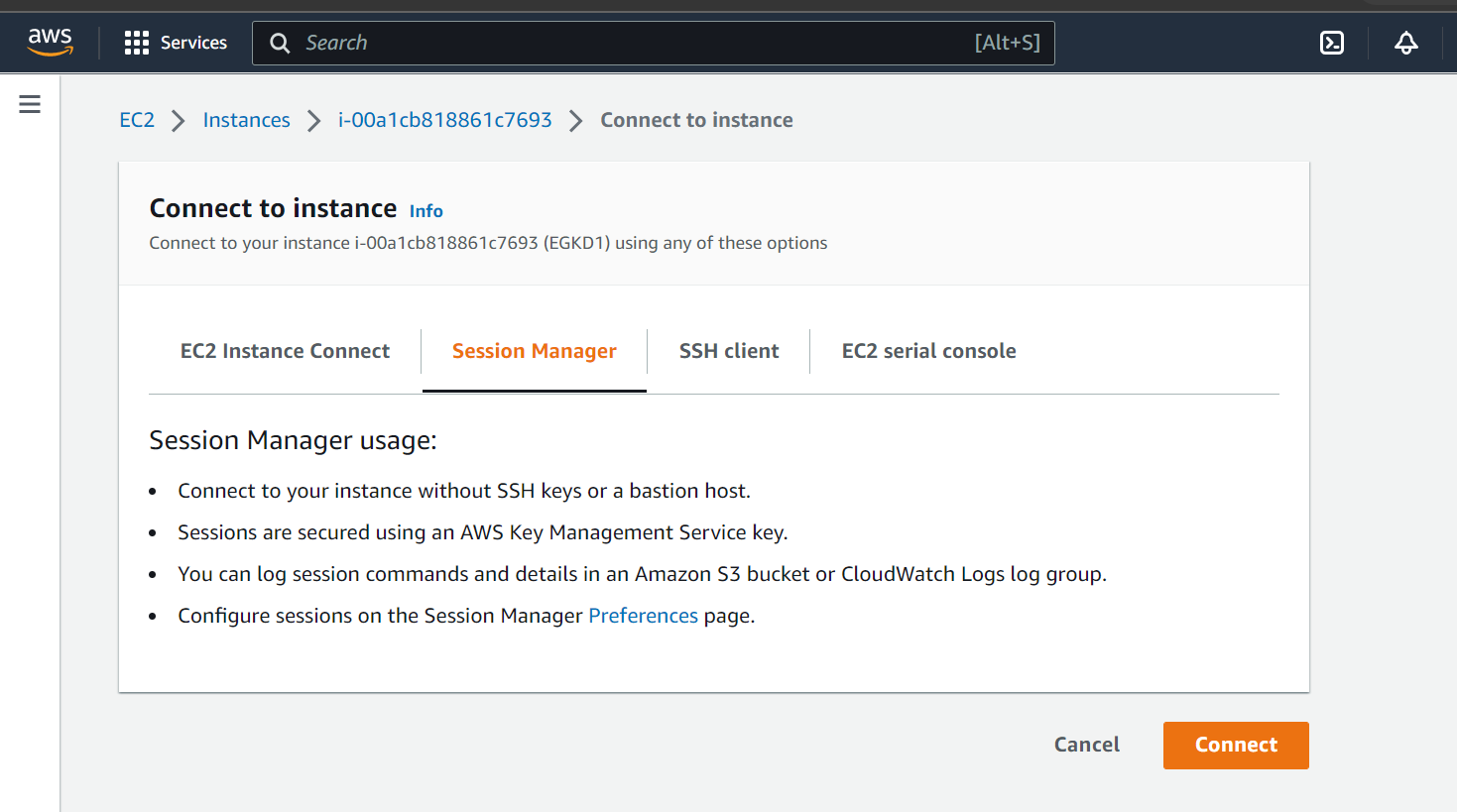
#### Step8:

Once still step-7 completed, we have to launch an instance, it will show state as running.

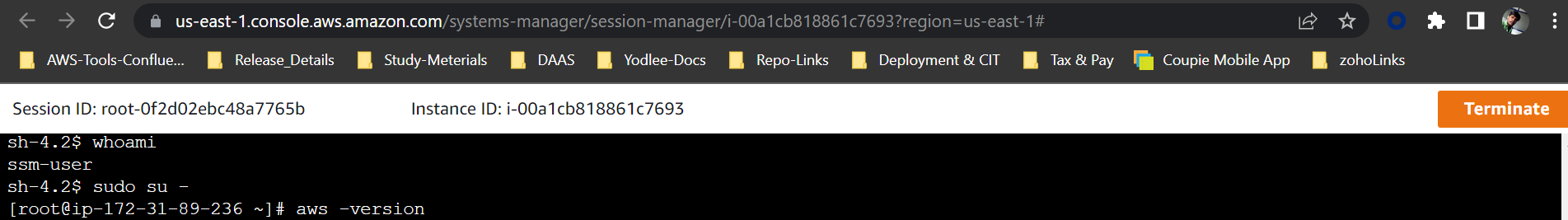


#### Step9:

Connecting to session manager

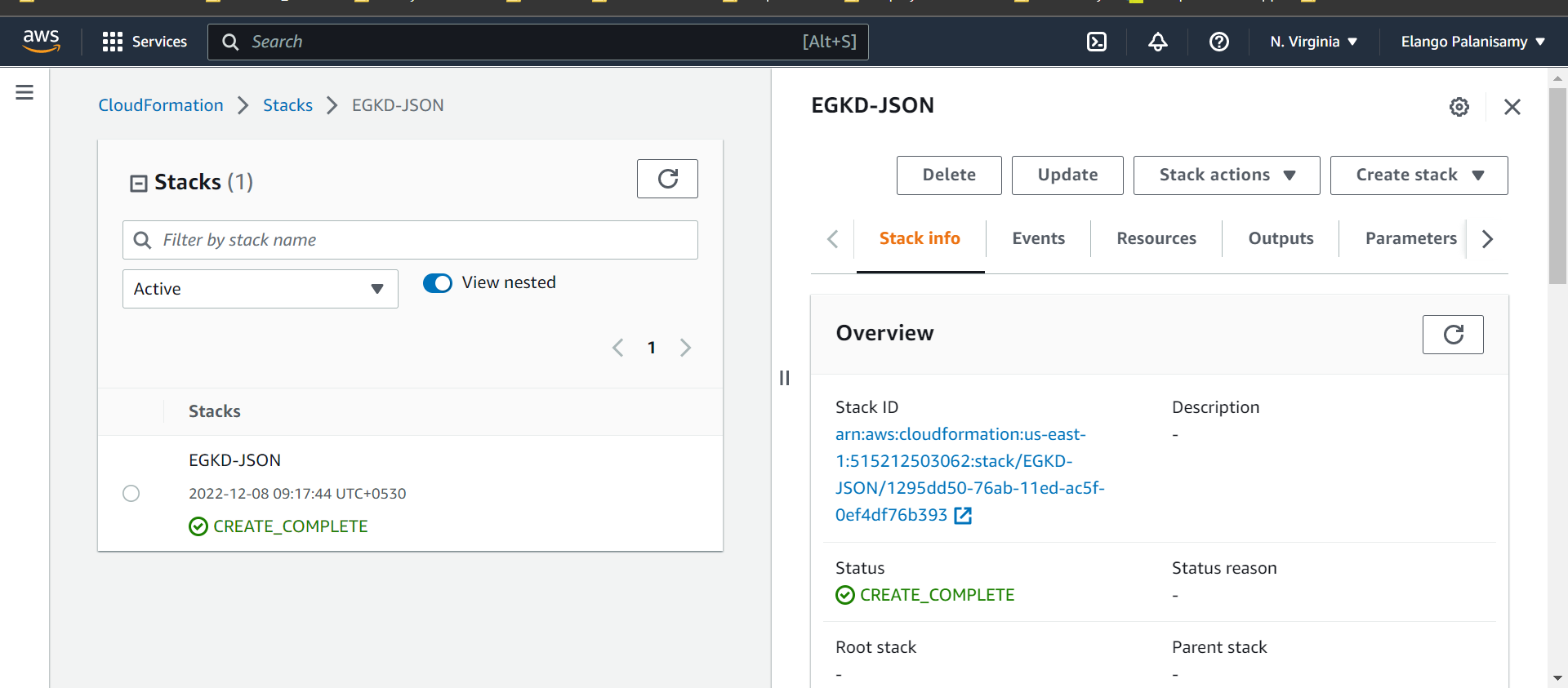


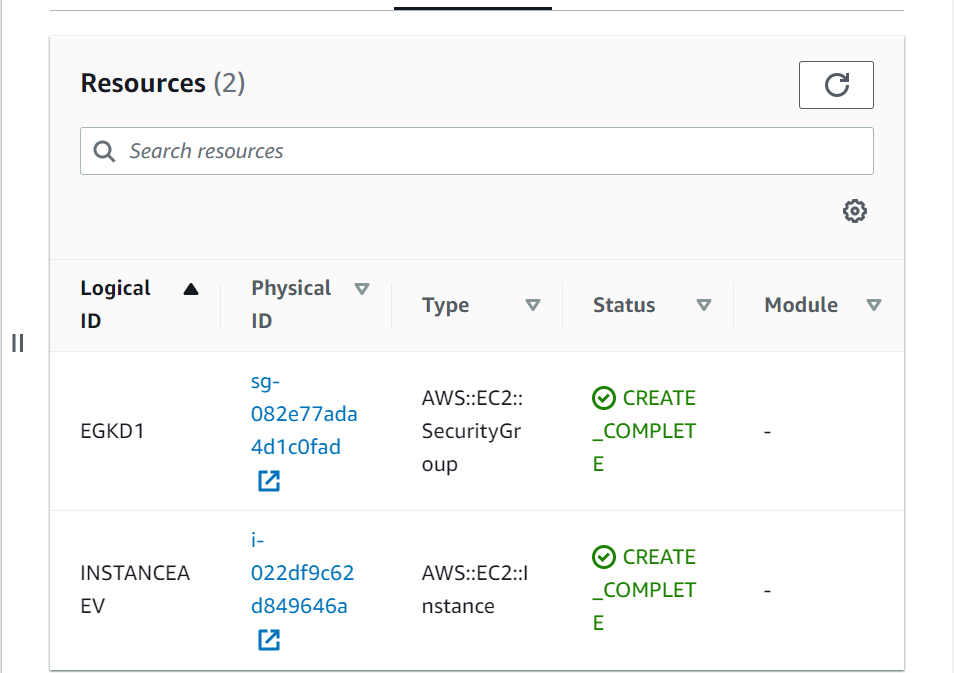
Connected to an session manager



## Creating an EC2-instance in cloud formation with JSON/YAM code:

We have created an instance and stack using JSON/YAML code, below is the references for the stack successful creation





### Coding of JSON and YAML:

|  |  |
| --- | --- |
| **JSON** | **YAML** |
| {  "Resources": {  "EGKD1": {  "Type": "AWS::EC2::SecurityGroup",  "Properties": {  "GroupDescription": "Allow ssh to client host",  "VpcId": "vpc-0bb5db1ab5492f0c9",  "SecurityGroupIngress": [  {  "IpProtocol": "tcp",  "FromPort": 22,  "ToPort": 22,  "CidrIp": "0.0.0.0/0"  }  ],  "SecurityGroupEgress": [  {  "IpProtocol": "tcp",  "FromPort": 0,  "ToPort": 65535,  "CidrIp": "0.0.0.0/0"  }  ]  }  },  "INSTANCEAEV": {  "Type": "AWS::EC2::Instance",  "Properties": {  "ImageId": "ami-0b0dcb5067f052a63",  "KeyName": "EGKD",  "BlockDeviceMappings": [  {  "DeviceName": "/dev/xvda",  "Ebs": {  "VolumeType": "gp2",  "VolumeSize": 8  }  }  ],  "InstanceType": "t2.micro",  "SecurityGroupIds": [  {  "Ref": "EGKD1"  }  ]  }  }  }  } | Resources:  EGKD1:  Type: 'AWS::EC2::SecurityGroup'  Properties:  GroupDescription: Allow ssh to client host  VpcId: vpc-0bb5db1ab5492f0c9  SecurityGroupIngress:  - IpProtocol: tcp  FromPort: 22  ToPort: 22  CidrIp: 0.0.0.0/0  SecurityGroupEgress:  - IpProtocol: tcp  FromPort: 0  ToPort: 65535  CidrIp: 0.0.0.0/0  INSTANCEAEV:  Type: 'AWS::EC2::Instance'  Properties:  ImageId: ami-0b0dcb5067f052a63  KeyName: EGKD  BlockDeviceMappings:  - DeviceName: /dev/xvda  Ebs:  VolumeType: gp2  VolumeSize: 8  InstanceType: t2.micro  SecurityGroupIds:  - Ref: EGKD1 |