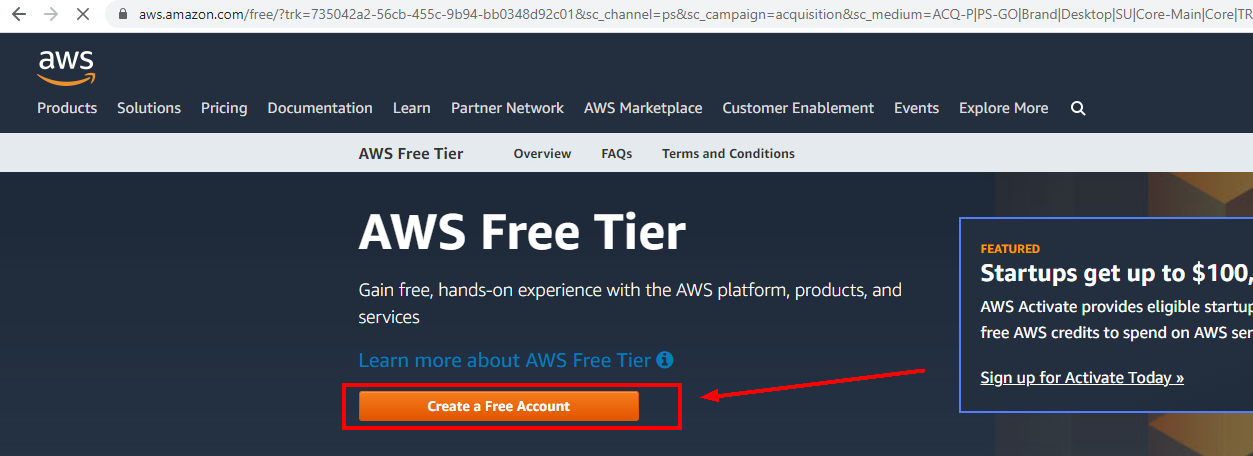
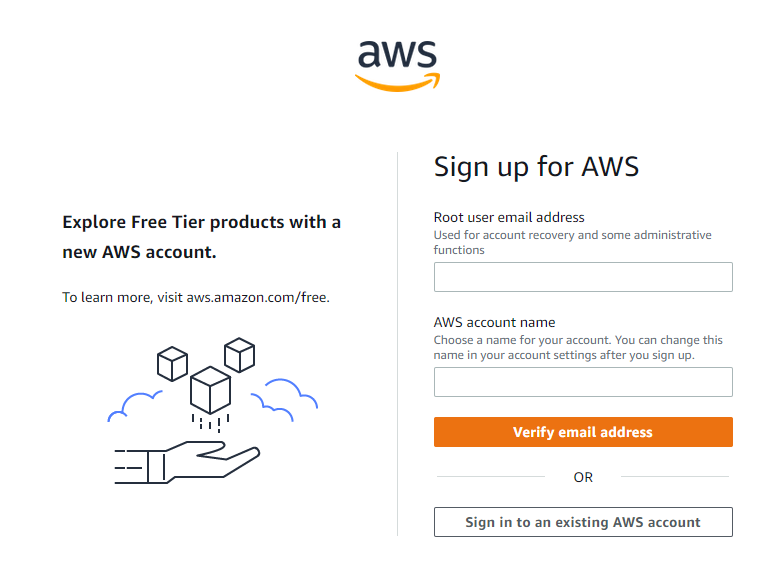
**AWS OPERATIONS (CREATE AWS ACCOUNT, CREATE NEW EC2 INSTANCE)**

1. **Create free AWS account using below URL**

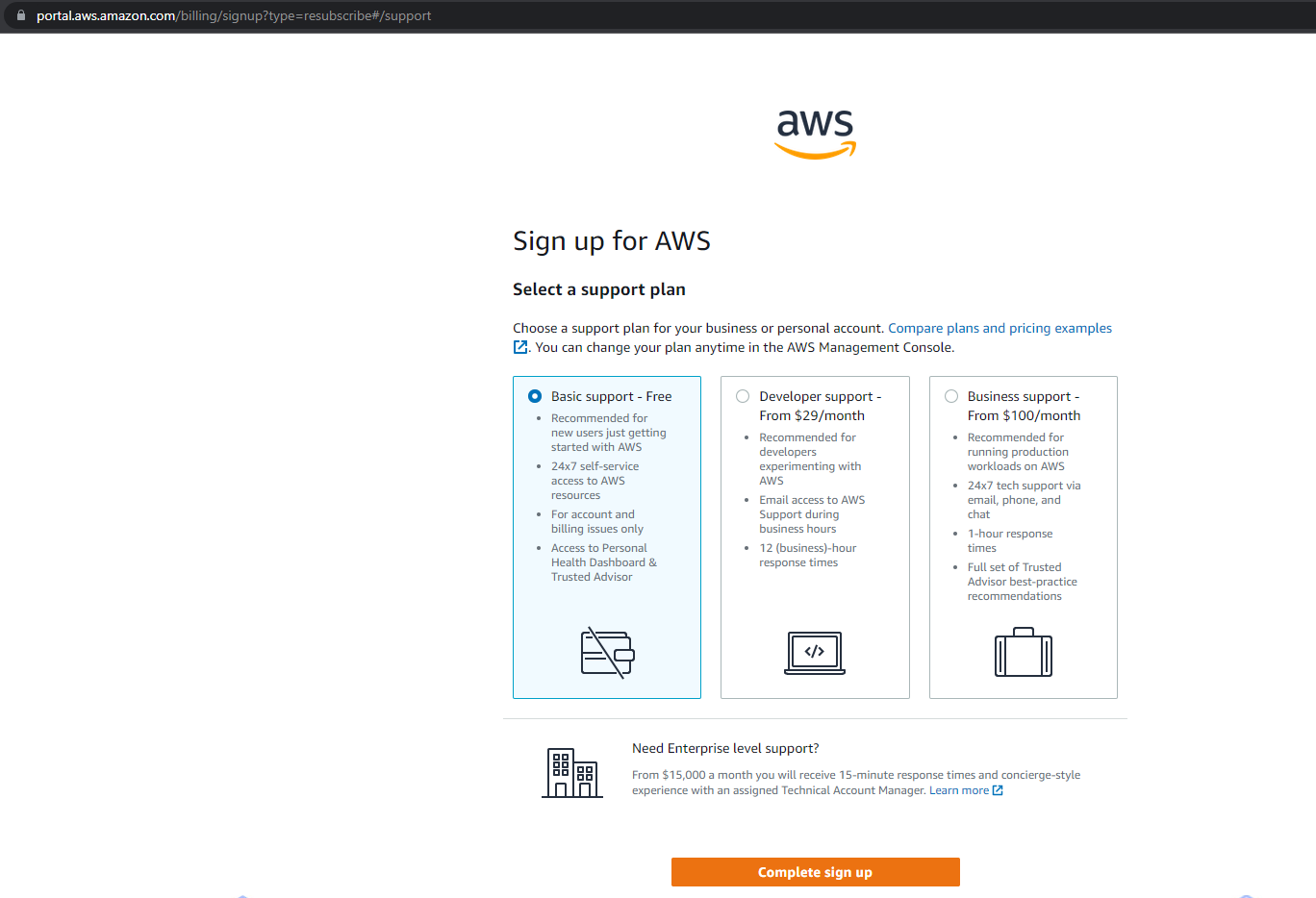
<https://aws.amazon.com/free/?trk=735042a2-56cb-455c-9b94-bb0348d92c01&sc_channel=ps&sc_campaign=acquisition&sc_medium=ACQ-P|PS-GO|Brand|Desktop|SU|Core-Main|Core|TR|EN|Text&s_kwcid=AL!4422!3!444593200938!e!!g!!aws&ef_id=EAIaIQobChMI0cKxrbOo9wIVHePmCh1fvwhiEAAYASAAEgJAkfD_BwE:G:s&s_kwcid=AL!4422!3!444593200938!e!!g!!aws>



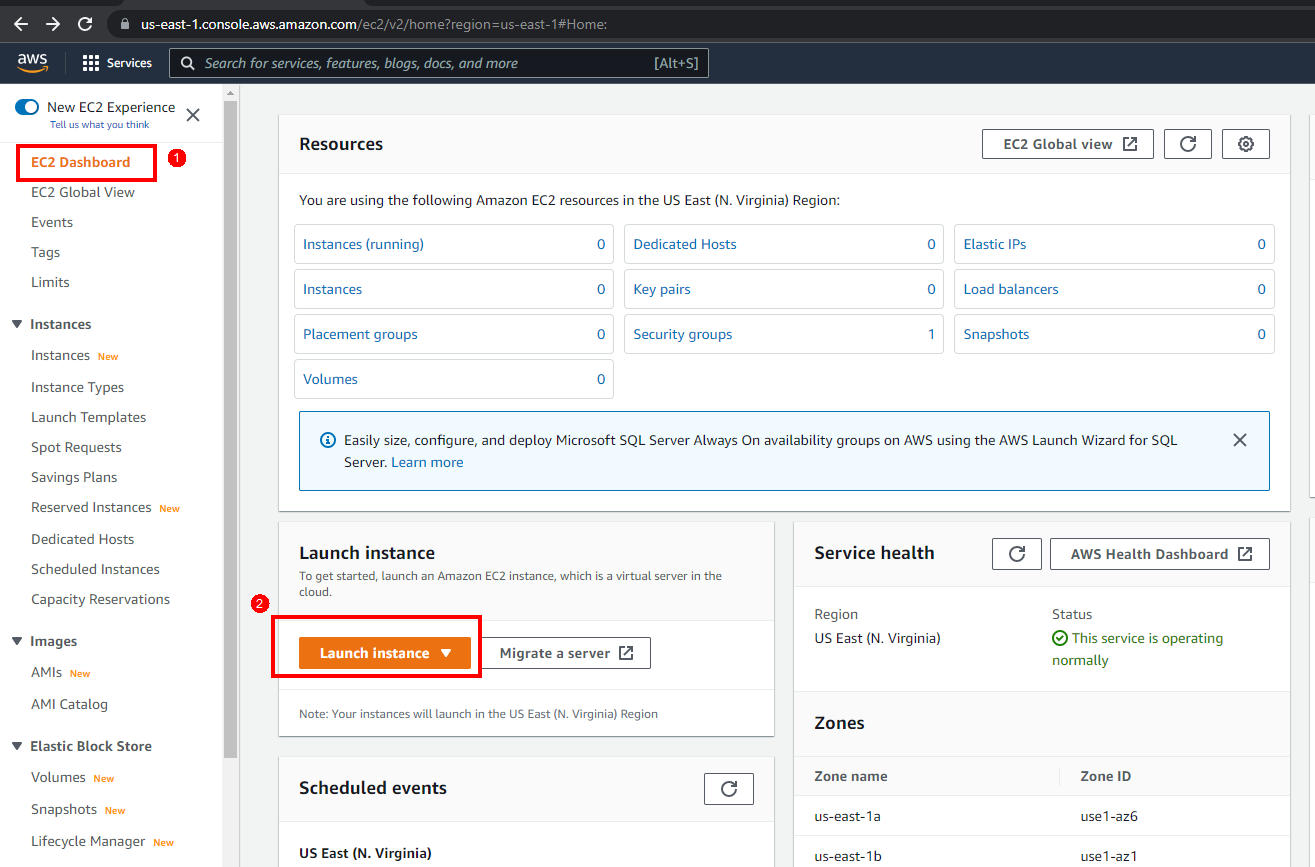
1. **Enter your mail, phone and credit card informations and confirm your application.**



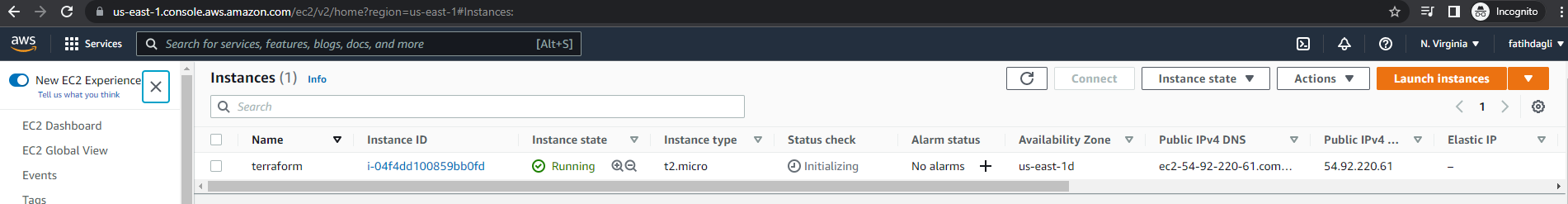
1. **After created account, Select AWS Plan**



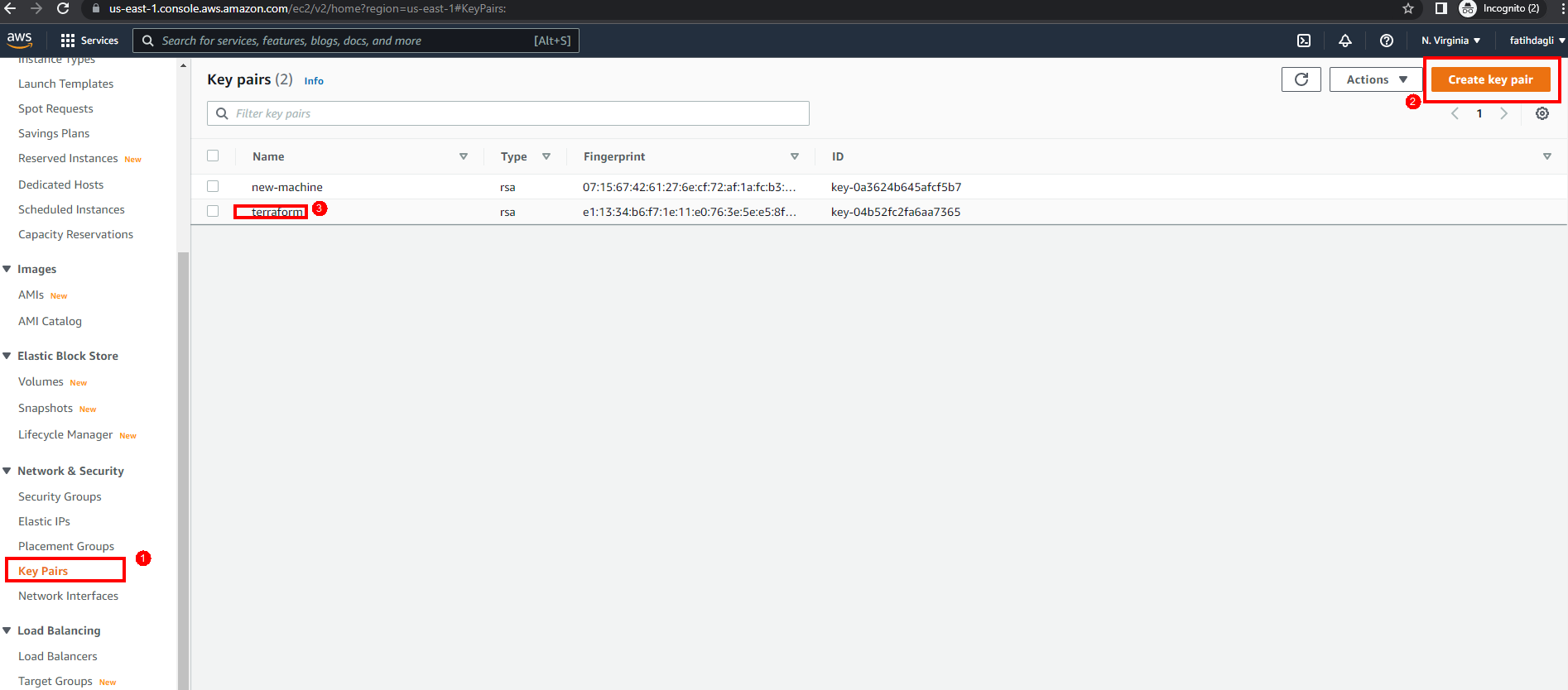
1. **Open EC2 service and create new instance**



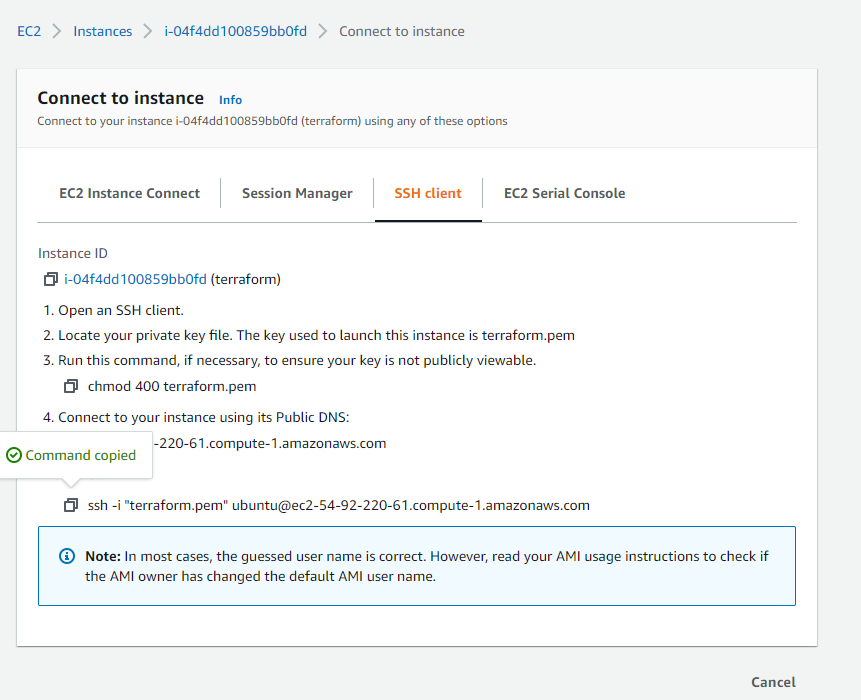
1. **Instance is ready now**



1. **Create key pair to connect to machine via ssh client. Download key to local.**

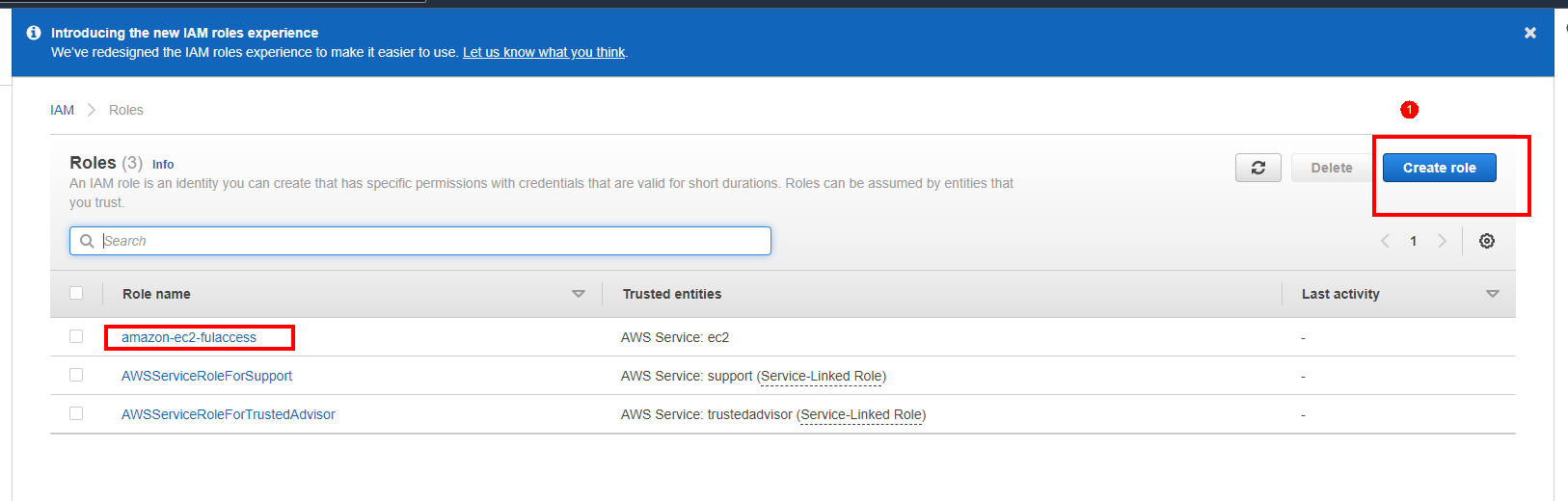


1. **Click connect button and copy ssh command to connect machine**

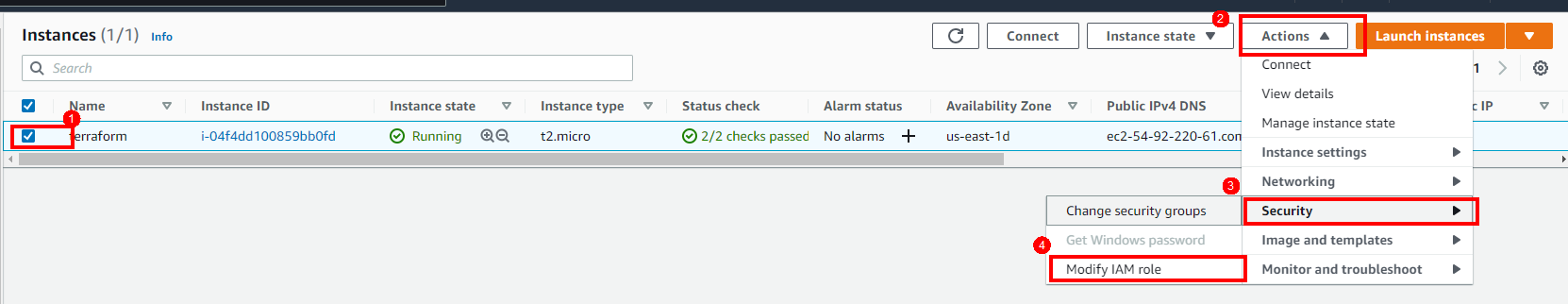


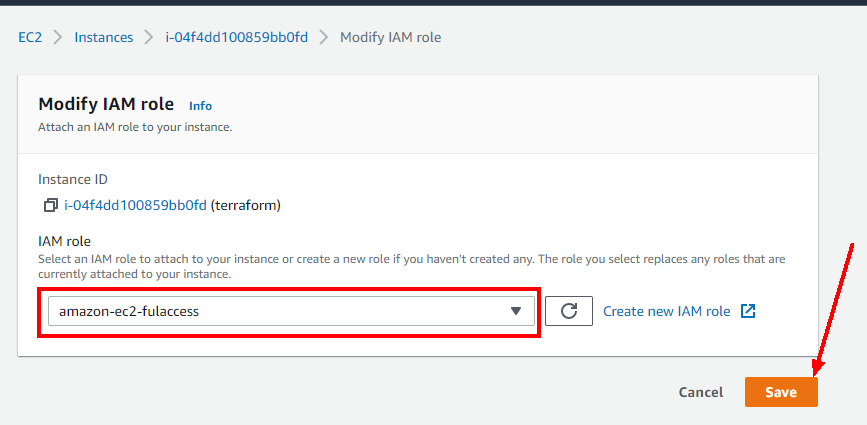
1. **Create IAM role**

Create IAM role with **AmazonEC2FullAccess** and [**AmazonEC2ContainerRegistryFullAccess**](https://us-east-1.console.aws.amazon.com/iam/home#/policies/arn:aws:iam::aws:policy/AmazonEC2ContainerRegistryFullAccess)Policy using **Create role** button to give machine create permission to our machine

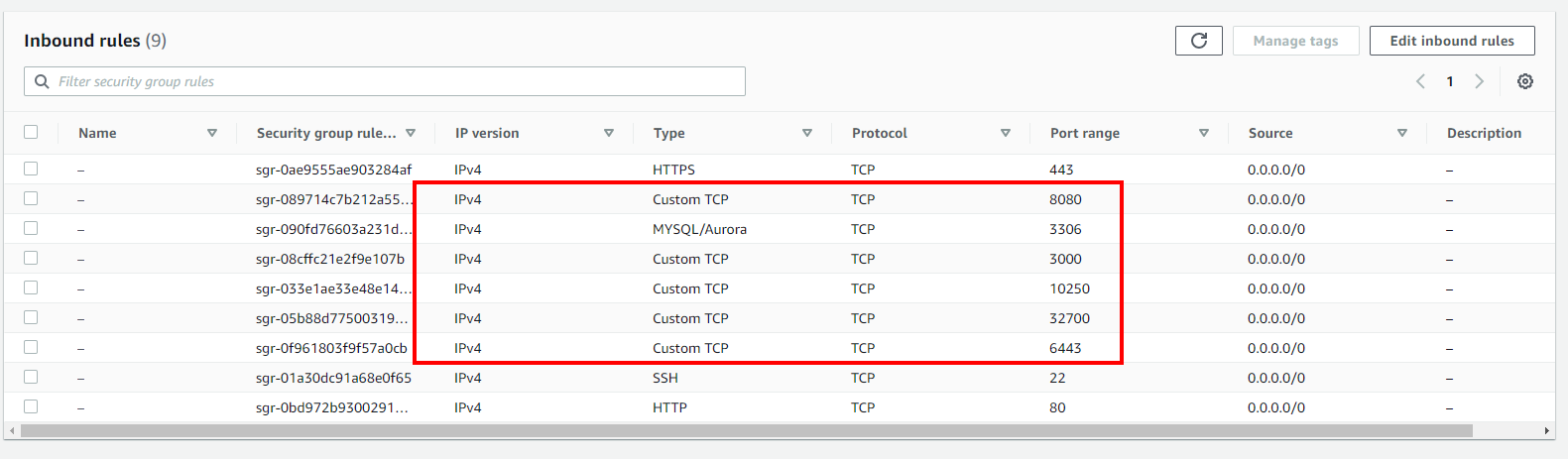


1. **Assign IAM role to EC2 instance**



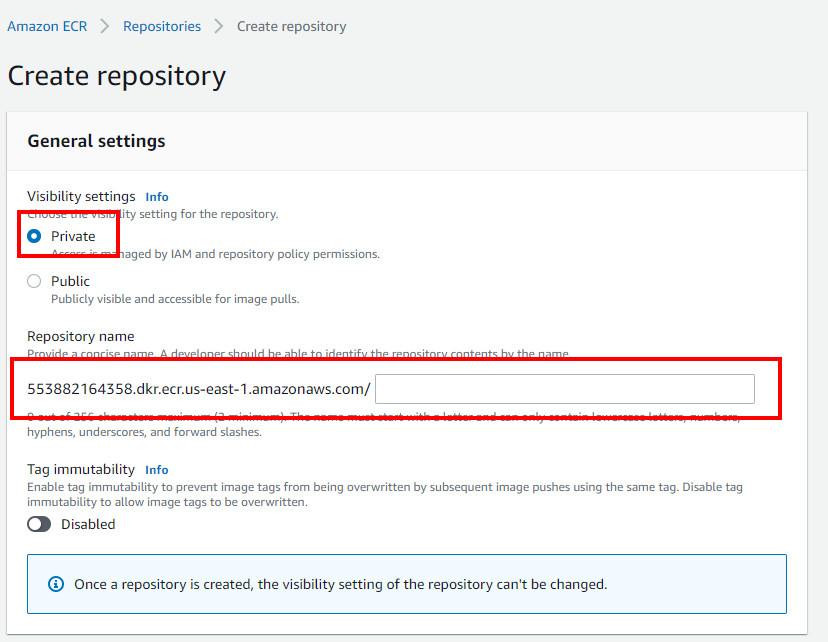


1. **Add new 6443, 10250, 8080, 3306, 3000 and 32700 port to security group**

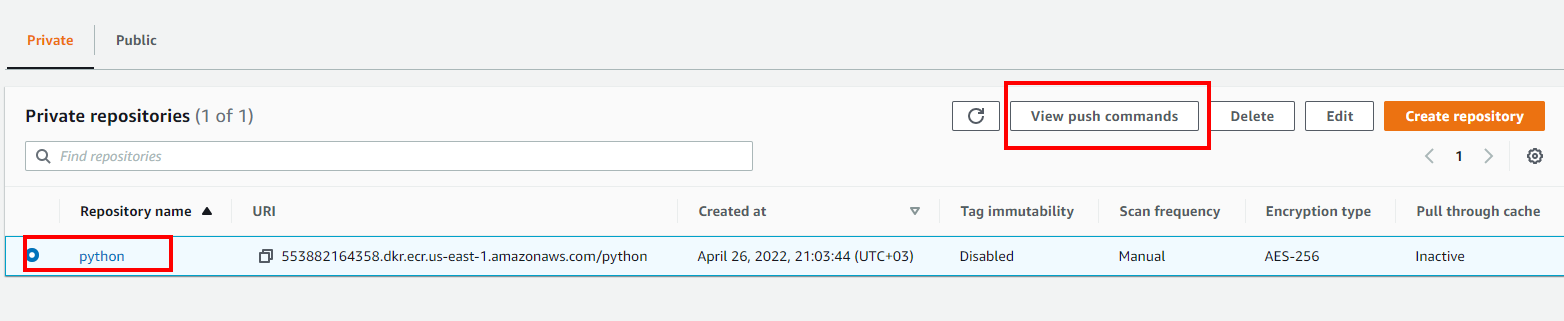


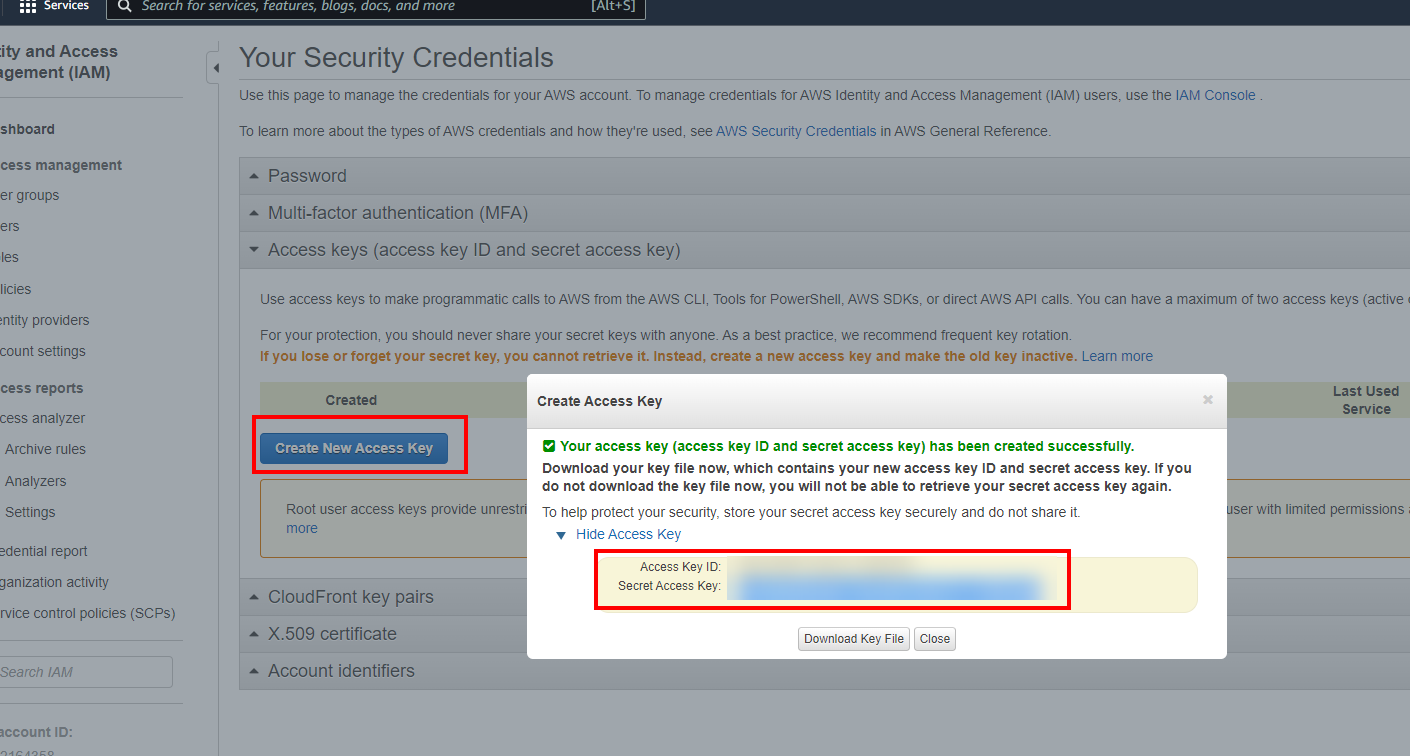
1. **Create private ECR repository image for docker images**

[**https://us-east-1.console.aws.amazon.com/ecr/create-repository?publicRepoCreate=true&region=us-east-1**](https://us-east-1.console.aws.amazon.com/ecr/create-repository?publicRepoCreate=true&region=us-east-1)



You can see push command in here after created



1. **Get secret and access key and enter it in ansible prompt**

**ANSIBLE INSTALLATION**

1. **Execute below command to install ansible**

sudo apt update

sudo apt install ansible

1. **Change root user ssh permission and change root password**

sudo su

sed -i "s/#PermitRootLogin prohibit-password/PermitRootLogin yes/g" /etc/ssh/sshd\_config

systemctl restart sshd

passwd root

1. change **PasswordAuthentication no** >>> **PasswordAuthentication yes** in/etc/ssh/sshd\_config

sed -i "s/**PasswordAuthentication no**/**PasswordAuthentication** yes/g" /etc/ssh/sshd\_config

1. Install **sshpass**, **docker** and **awscli** package

apt-get install sshpass

apt-get install docker.io

apt-get install awscli

1. Change ansible host file path (this hosts file will be generated after you cloned the repo with the name kubernetes-env-creation-with-aws/ansible/hosts) in /etc/ansible/ansible.cfg

vim /etc/ansible/ansible.cfg

**OLD: #inventory = /etc/ansible/hosts**

**NEW: inventory = /home/ubuntu/kubernetes-env-creation-with-aws/ansible/hosts**

1. Change **host\_key\_checking** parameterin/etc/ansible/ansible.cfg

vim /etc/ansible/ansible.cfg

**OLD: #host\_key\_checking = False**

**NEW: host\_key\_checking = False**

1. Install terraform plugin

ansible-galaxy collection install community.general

1. Install some required python package for jenkins job creation

apt install python3-pip

pip install docker

pip install python-jenkins

**TERRAFORM INSTALLATION**

1. **Execute below commands to install terraform**

sudo mkdir -p /opt/terraform

cd /opt/terraform

sudo wget <https://releases.hashicorp.com/terraform/1.1.7/terraform_1.1.7_linux_386.zip>

sudo apt-get install unzip -y

sudo unzip terraform\_1.1.7\_linux\_386.zip

sudo mv /opt/terraform/terraform /usr/bin/

terraform --version

**USE TERRAFORM COMMAND MANUALLY IF YOU WANT**

* Go any folder in machine and clone this repository.

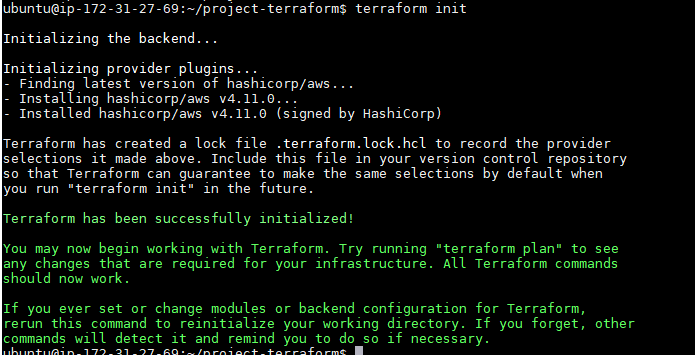
git clone https://github.com/fatih-dagli/kubernetes-env-creation-with-aws.git

* You should configure your personal informations and path in

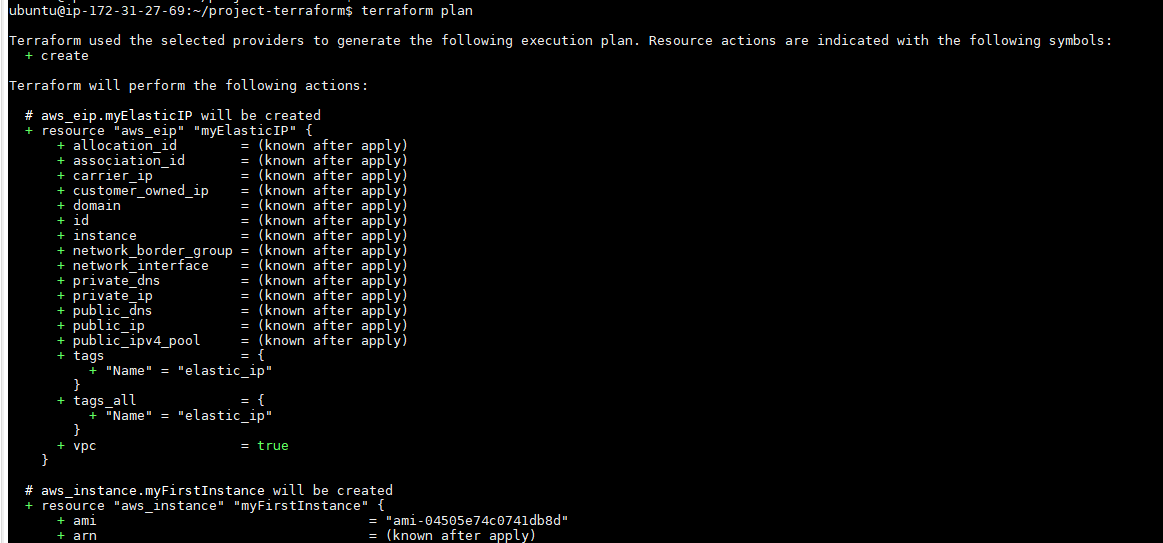
- kubernetes-env-creation-with-aws/terraform/variable.tf (Change AWS account information)

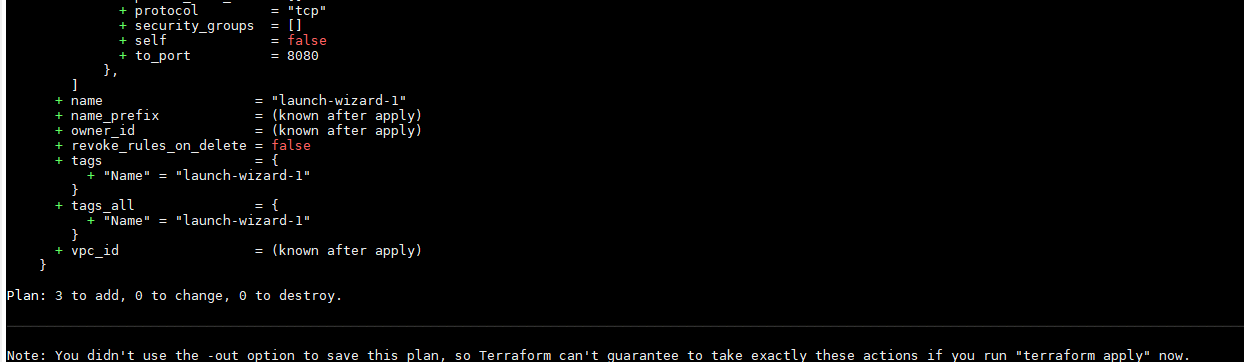
* Execute below commands in **kubernetes-env-creation-with-aws/terraform-files** folder

1. Execute **terraform init** command to initialize some aws plugin

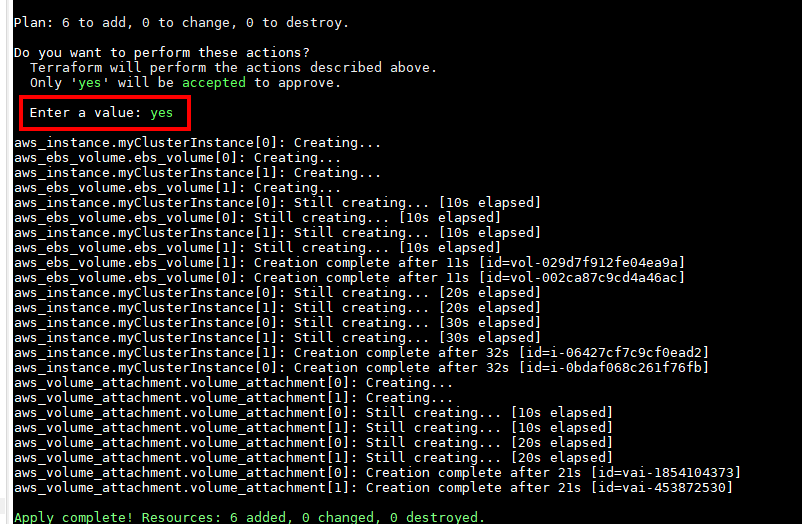


1. Execute **terraform plan** to see and ensure what tarraform will do.

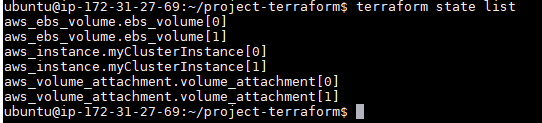




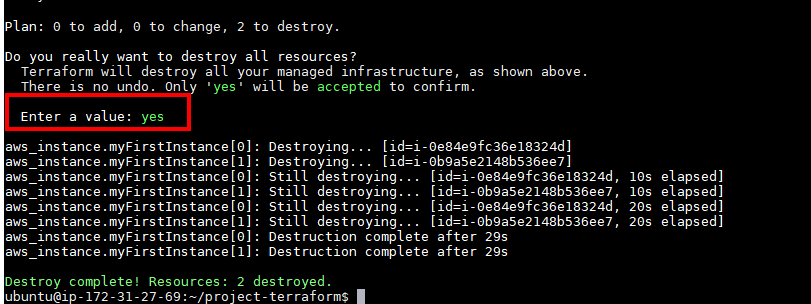
1. Execute **terraform apply** to create machine



1. You can see your terraform instance using **terraform state list** command.



1. If you want to destroy your terraform instance, you can use **terraform destroy** command



**INSTALLATION STEPS**

9. Go any folder in machine and clone this repository.

git clone https://github.com/fatih-dagli/kubernetes-env-creation-with-aws.git

10. You should configure your personal informations and path in

- kubernetes-env-creation-with-aws/terraform/variable.tf (Change AWS account information)

- kubernetes-env-creation-with-aws/ansible/project.yml (Change project paths)

- kubernetes-env-creation-with-aws/ansbile/hosts (Change IP address with your machine IP address)

**Execute below commands in kubernetes-env-creation-with-aws/ansible folder and enter password of root user.**

**Second command will ask your AWS access, secret key, AWS region and AWS account id. You should enter these parameters.**

* ansible-playbook --tags install-terraform --skip-tags "install-jenkins" project.yml -k
* ansible-playbook --tags "prepare-k8s-all, initialize-k8s-master, initialize-k8s-node, install-jenkins" project.yml -k

Also If you want to deploy/update again your deployment after first installation, you can enter to jenkins and trigger pipeline manually. Jenkins and Application information will be shown in ansible output after installation finished successfully.