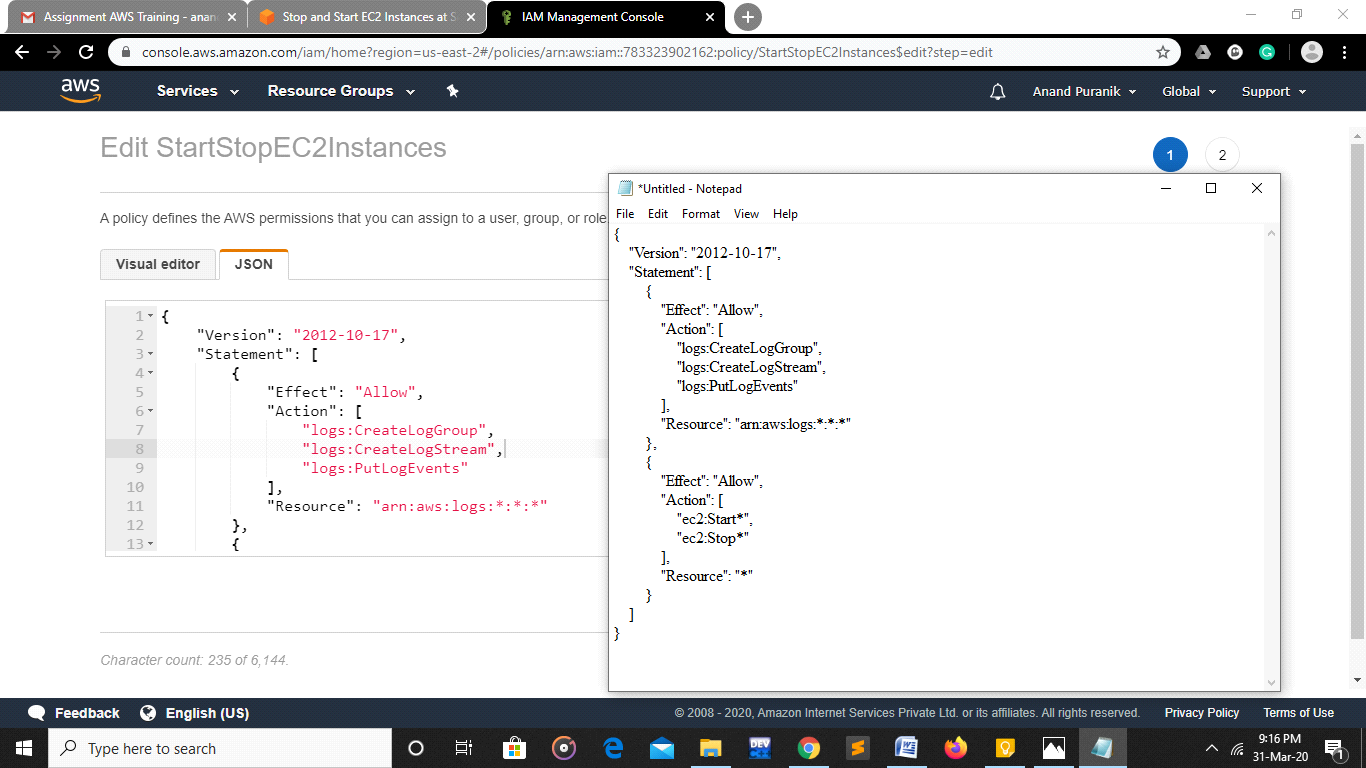
**Assignment 1 - Write lambda function using role to create and Start EC2 instace**

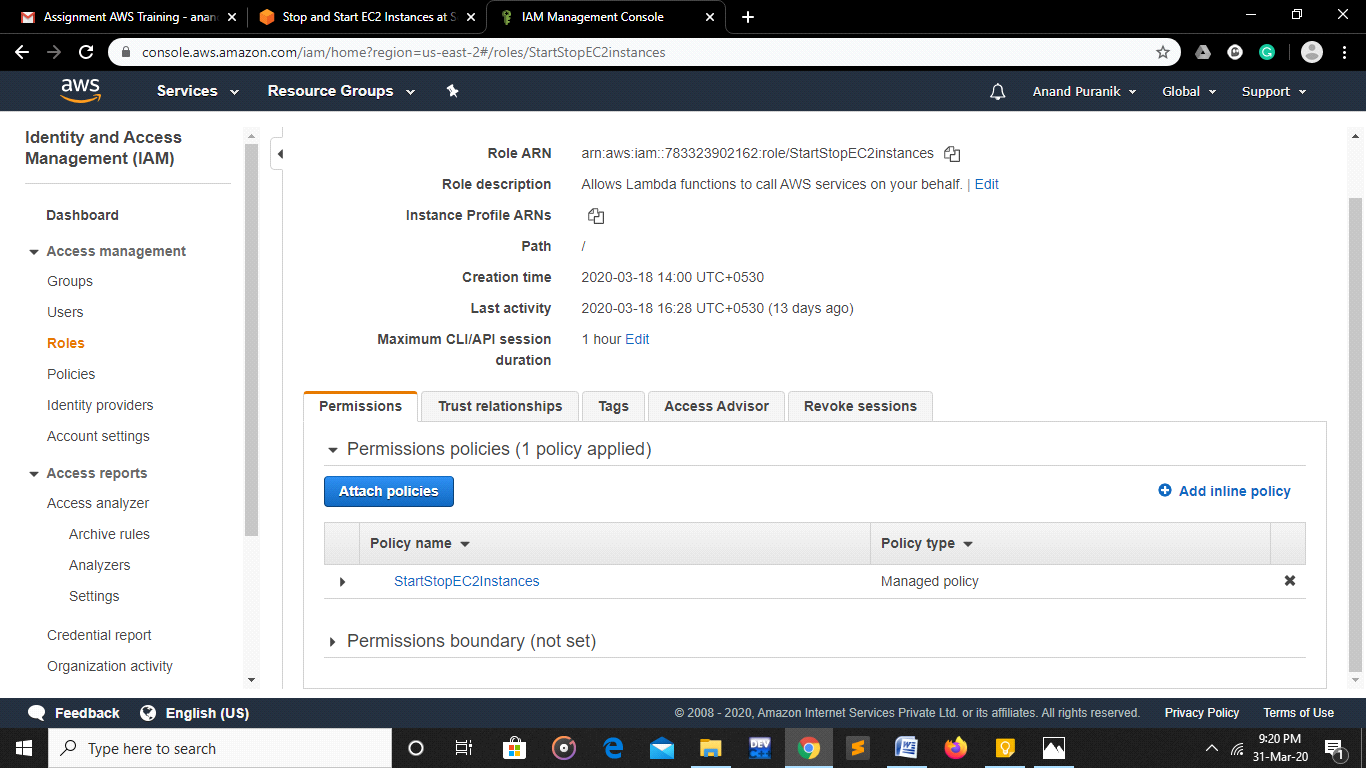
1.    Create a custom AWS Identity and Access Management (IAM) policy and execution role for your Lambda function.

2.    Create Lambda functions that stop and start your EC2 instances.

Create an IAM policy and role-



2.   Create an IAM role for Lambda. When attaching a permissions policy, search for and choose the IAM policy that you created.



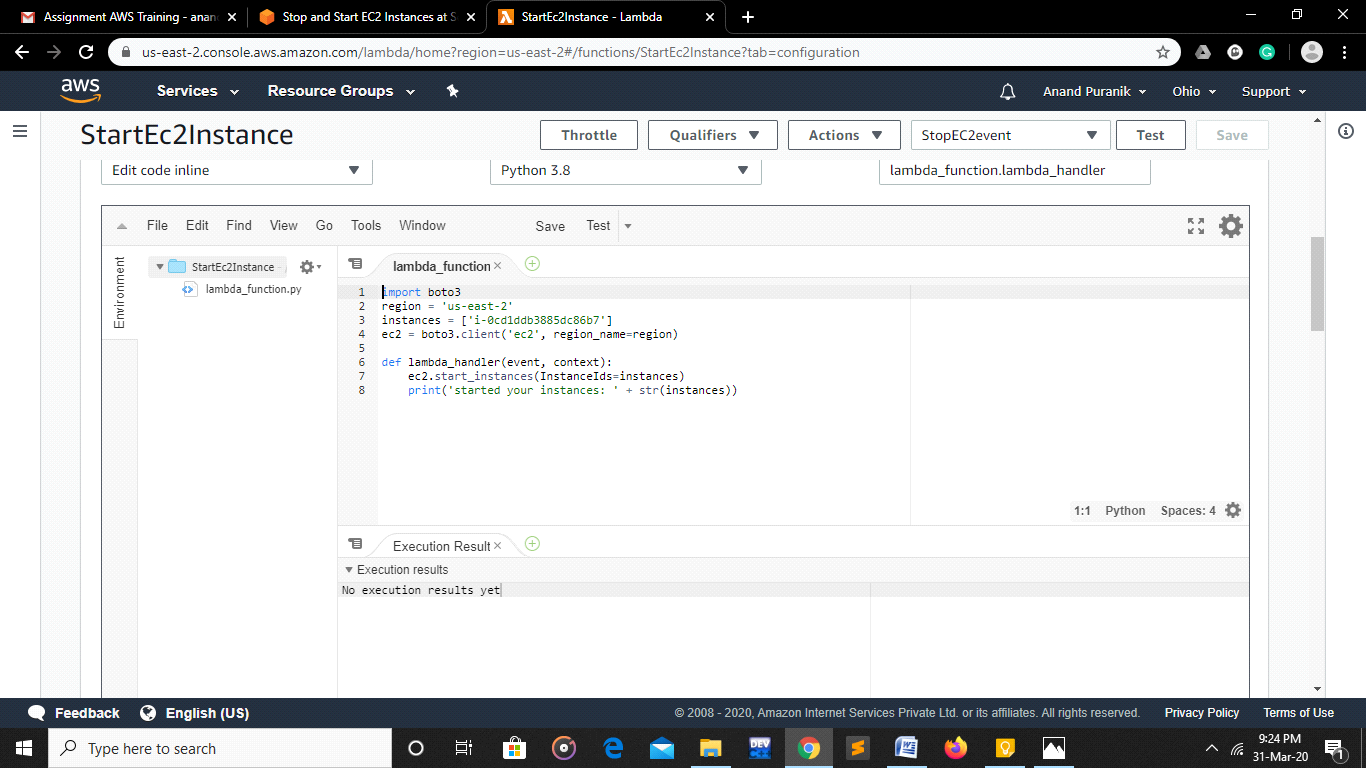
Create Lambda functions that stop and start your EC2 instances

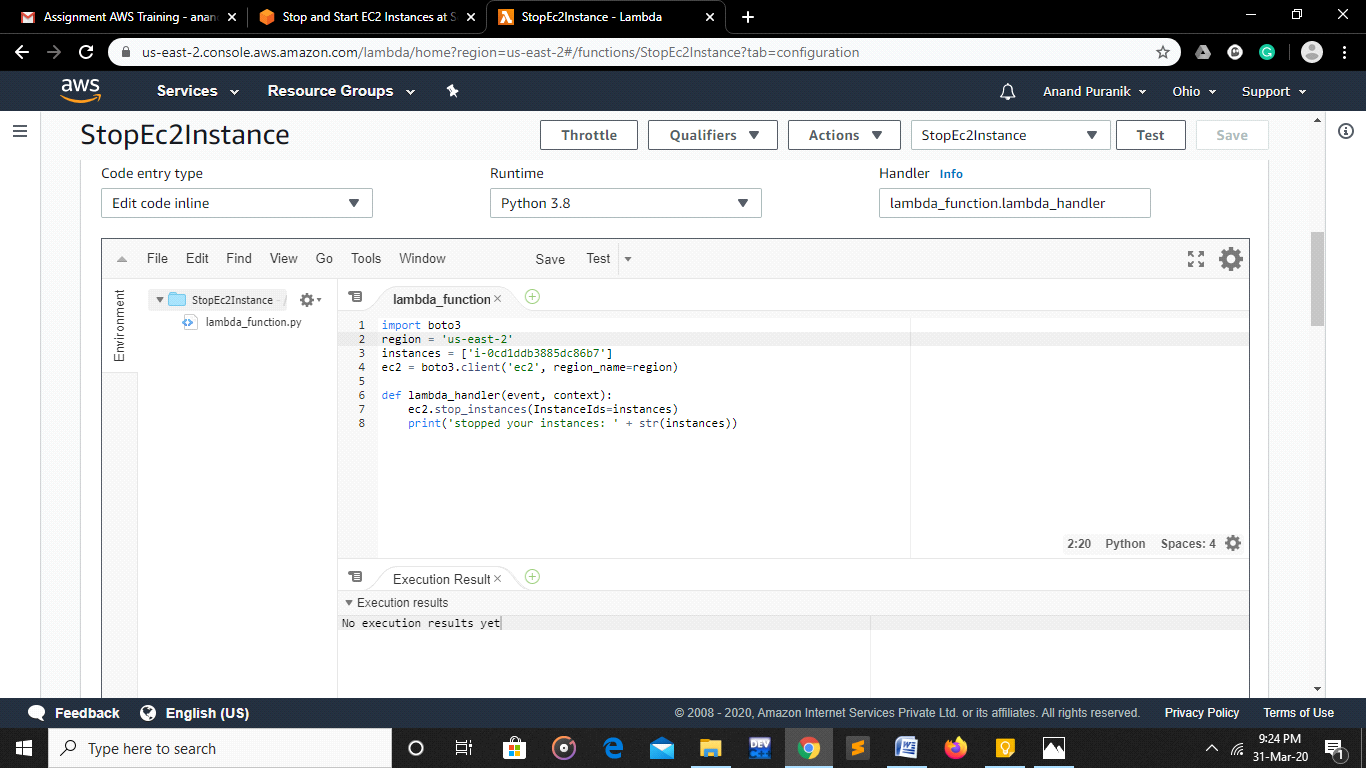
1.    In the Lambda Function, choose Create function.

2.    Choose Author from scratch.

3.    Under Basic information, add the following:  
For Function name, enter a name that identifies it as the function used to stop your EC2 instances. For example, "Start EC2Instances".  
  
Under Permissions, expand Choose or create an execution role.  
Under Execution role, choose Use an existing role.  
Under Existing role, choose the IAM role that you created.

4.    Choose Create function.





* Under Basic settings, set Timeout to 10 seconds.

Testing ofr Lambda functions

1.    In the Lambda Function, choose Functions.

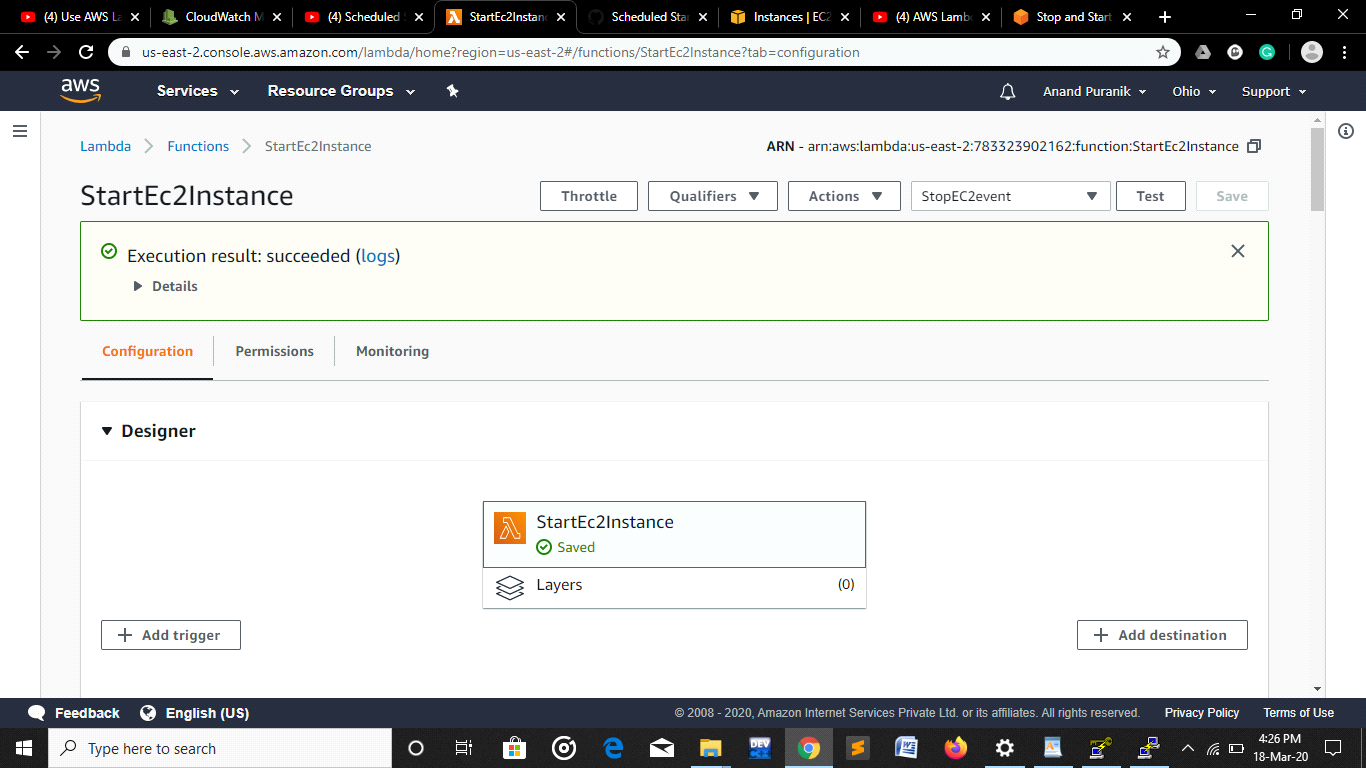
2.    Select one of the functions that you created.

3.    Choose Actions, and then choose Test.

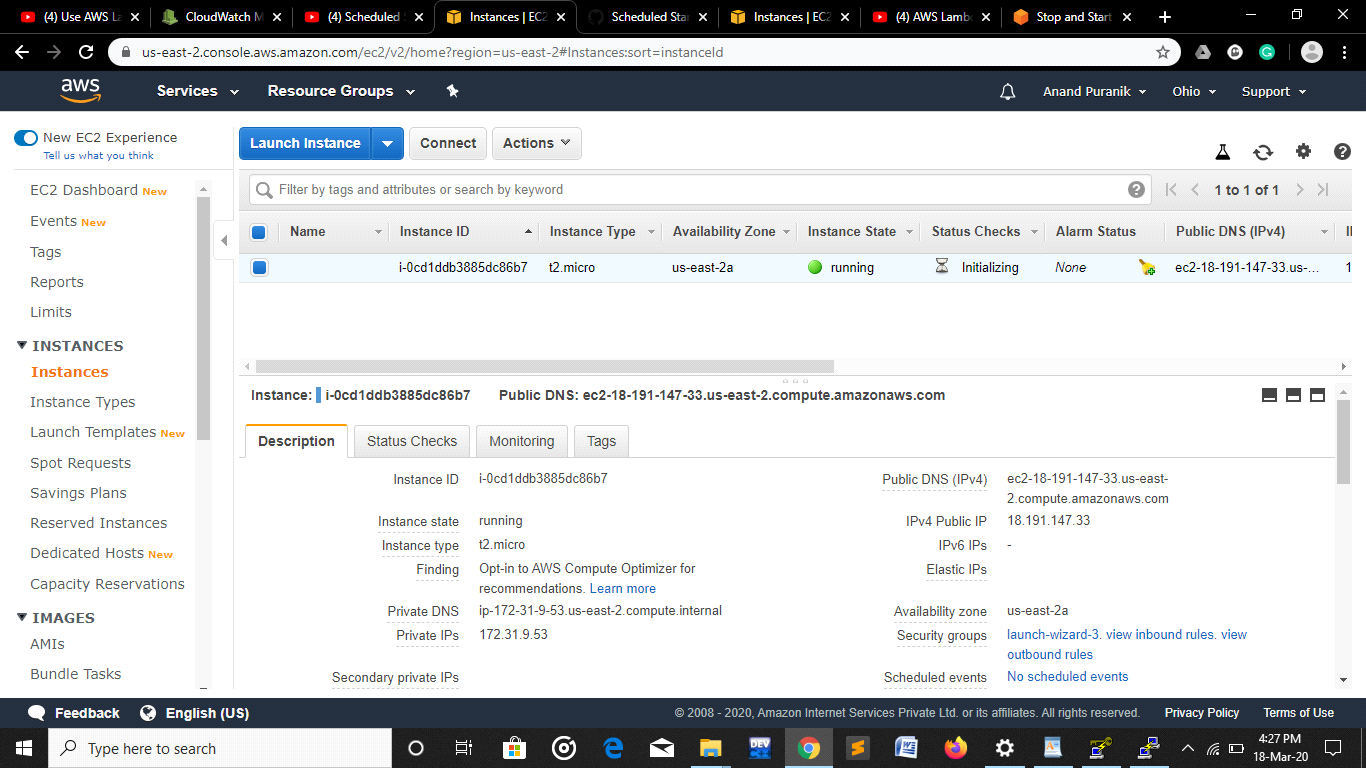
4.    In the Configure test event dialog, choose Create new test event.

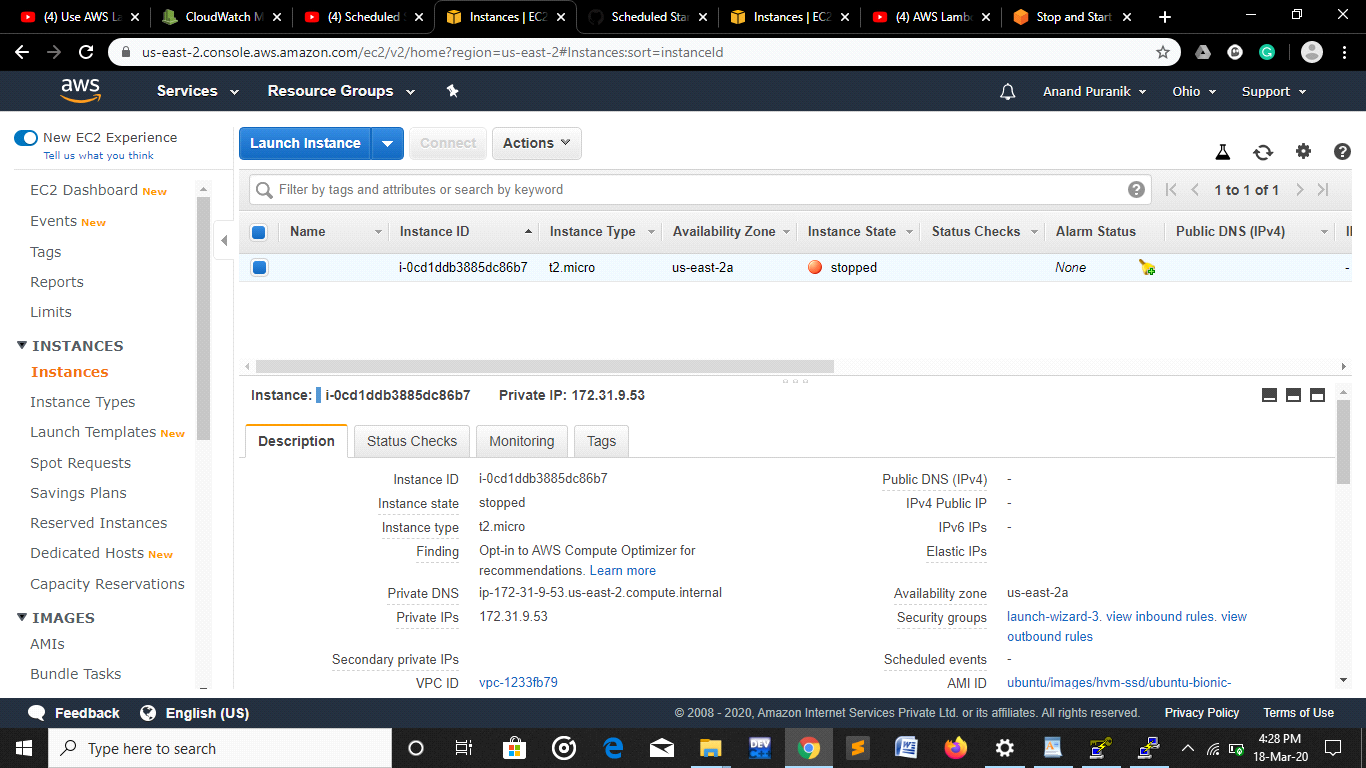
5.    Enter an Event name, and then choose Create.

6.    Choose Test to execute the function.



Output-

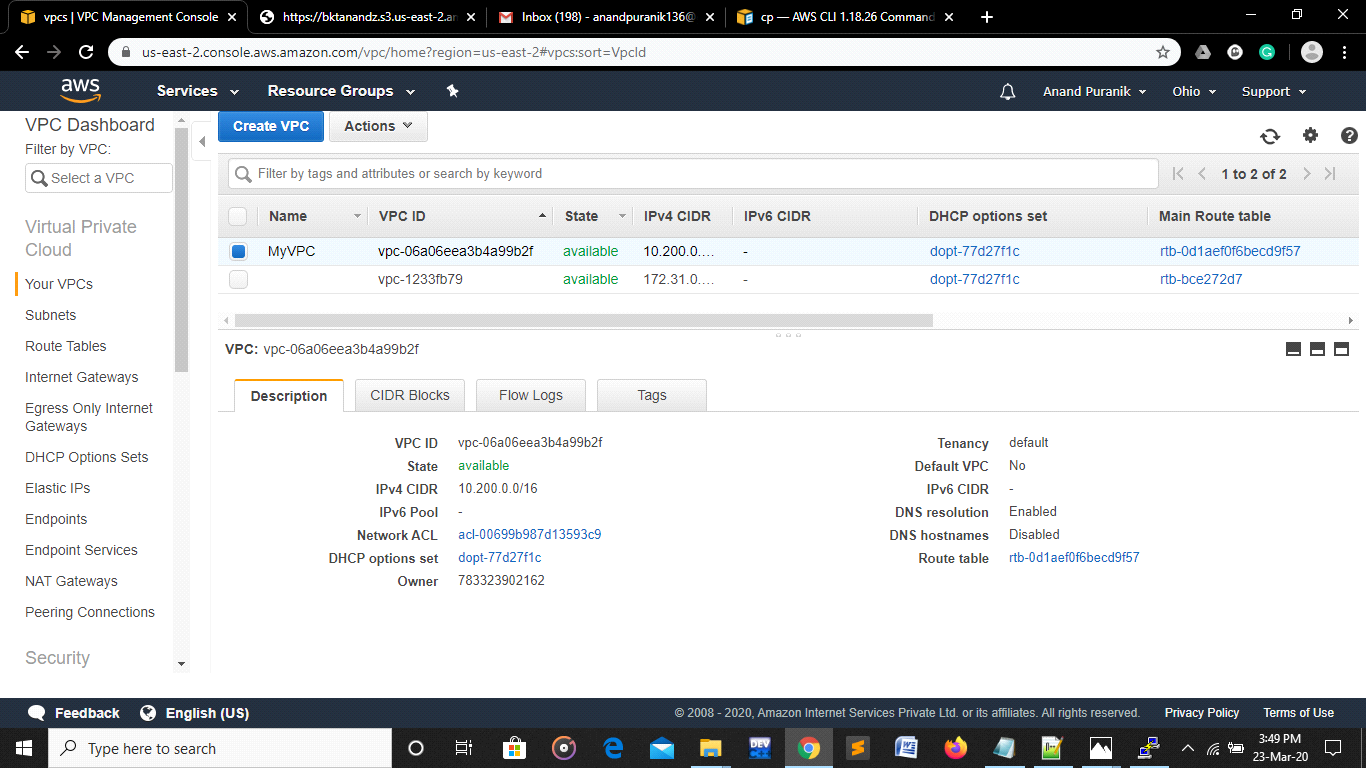


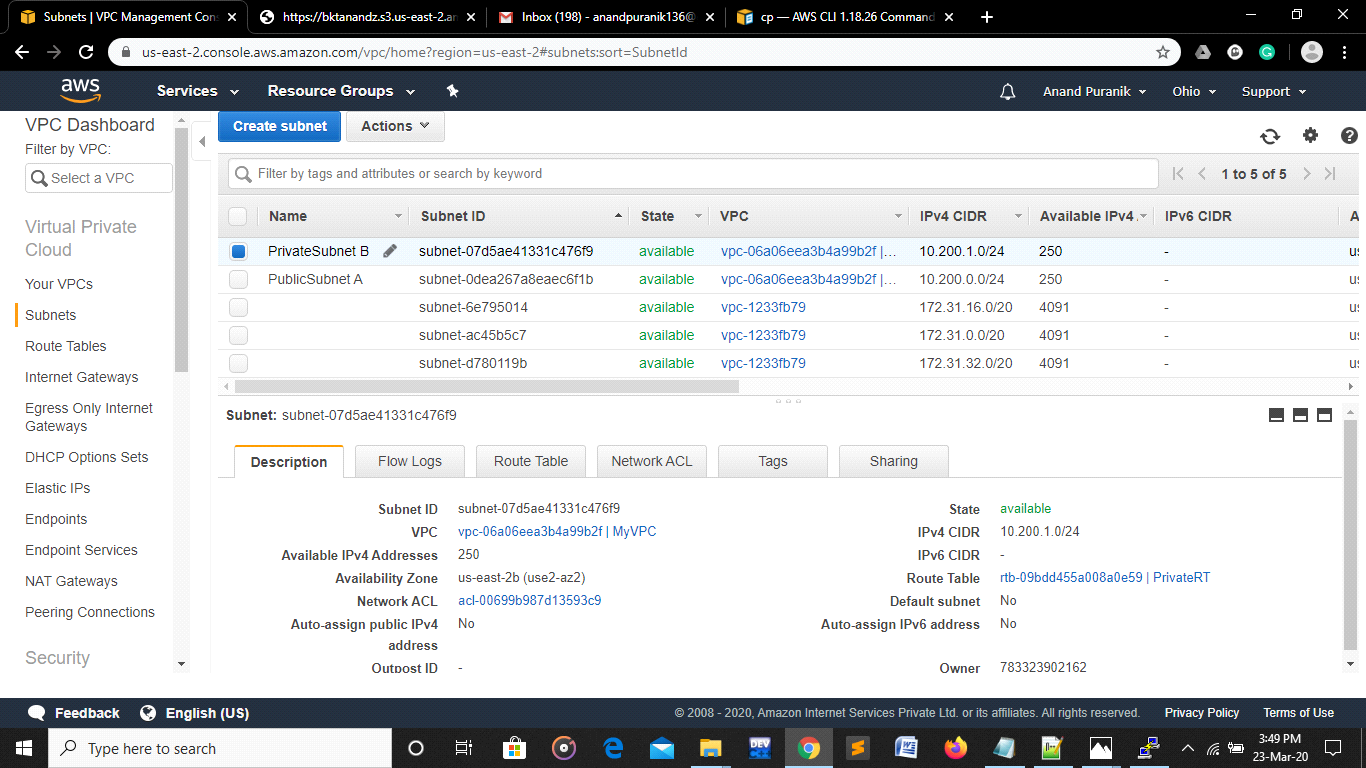


**Assignment 2**

Creating VPC with public and private subnet , configure route table , connect ec2 private instance with public ec2 instance , upload sample file to newly created s3 bucket

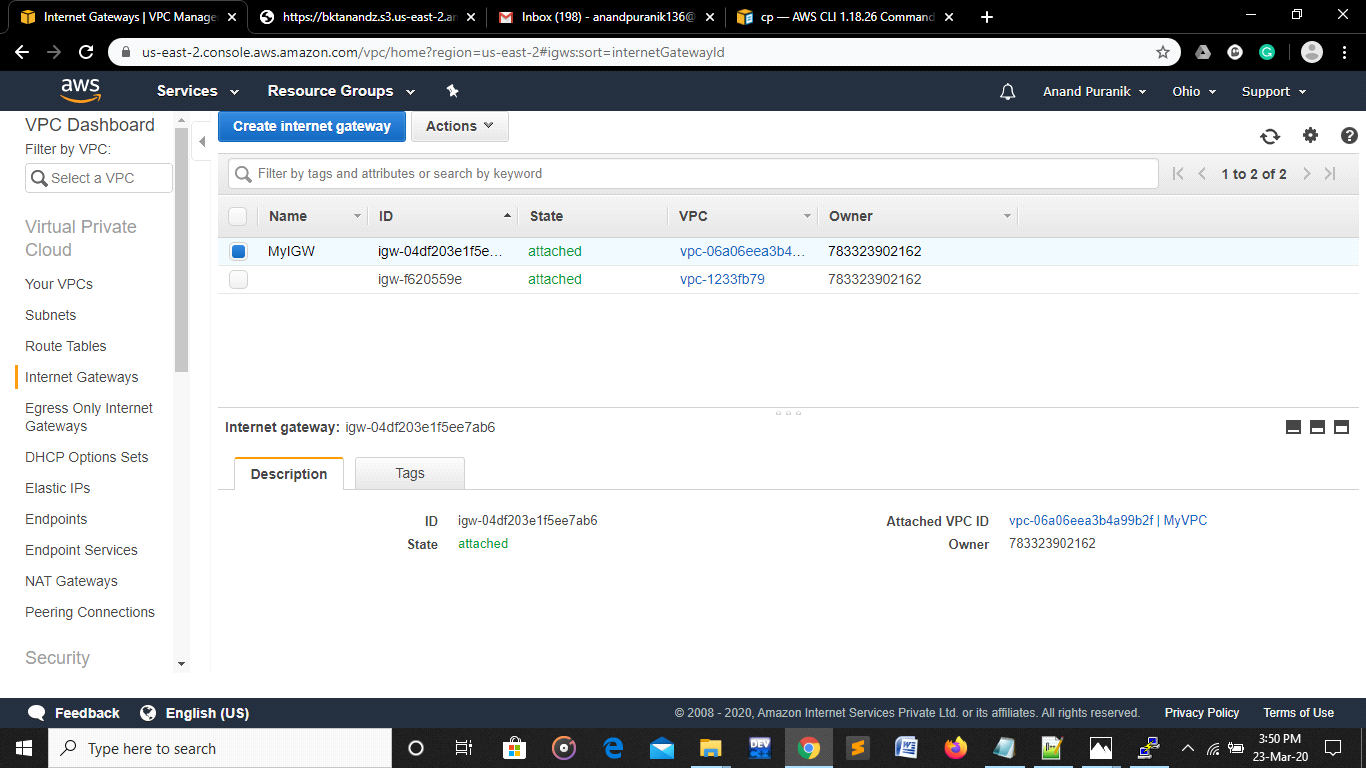
* VPC is created with CIDR block – 10.0.0.0/16
* 2 subnets are created &
* Choose custom vpc which we created
* Assign IPv4 CIDR block as giving ip range to subnet i.e 10.0.1.0/24





When creating Public subnet Modify auto assign IP - ON Auto-assign IPv4.

4. Further Internet Gateway is created and It is attached to the VPC

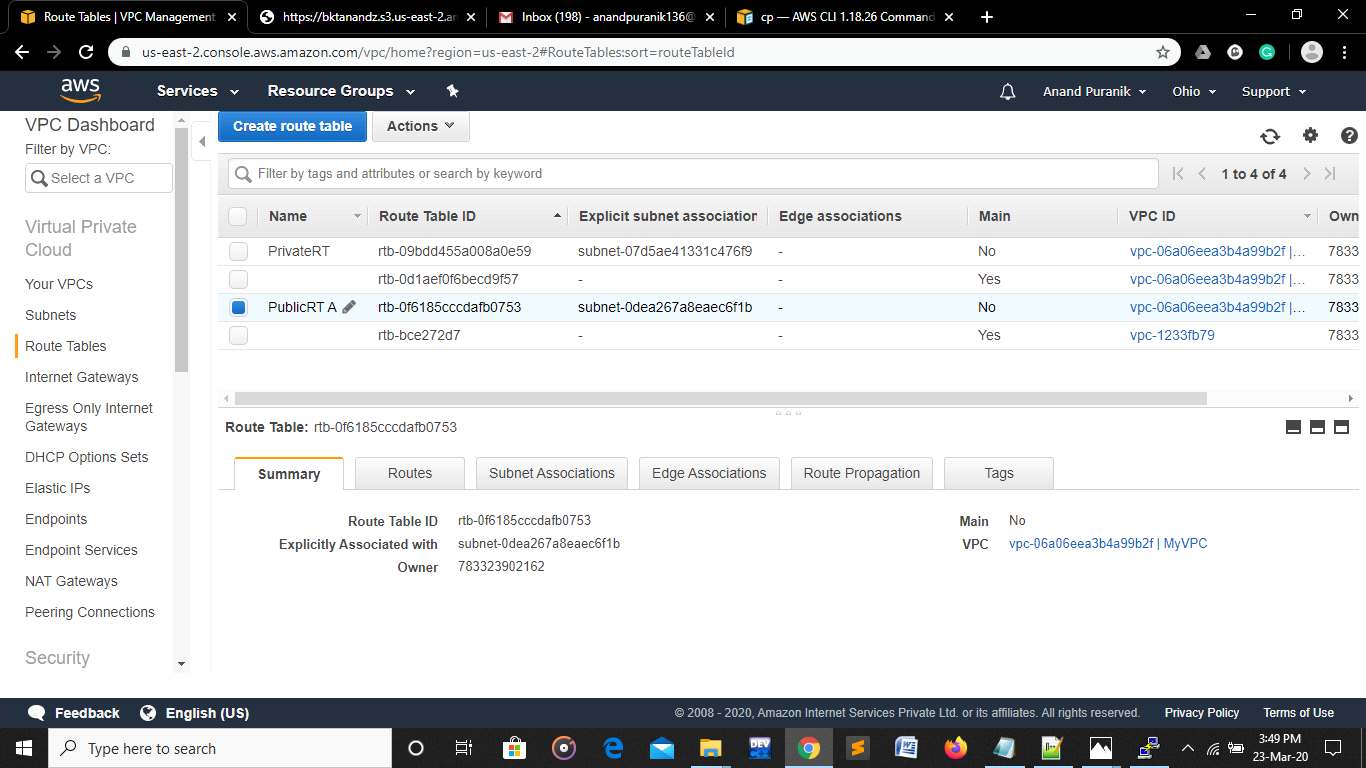


5. Creating route table

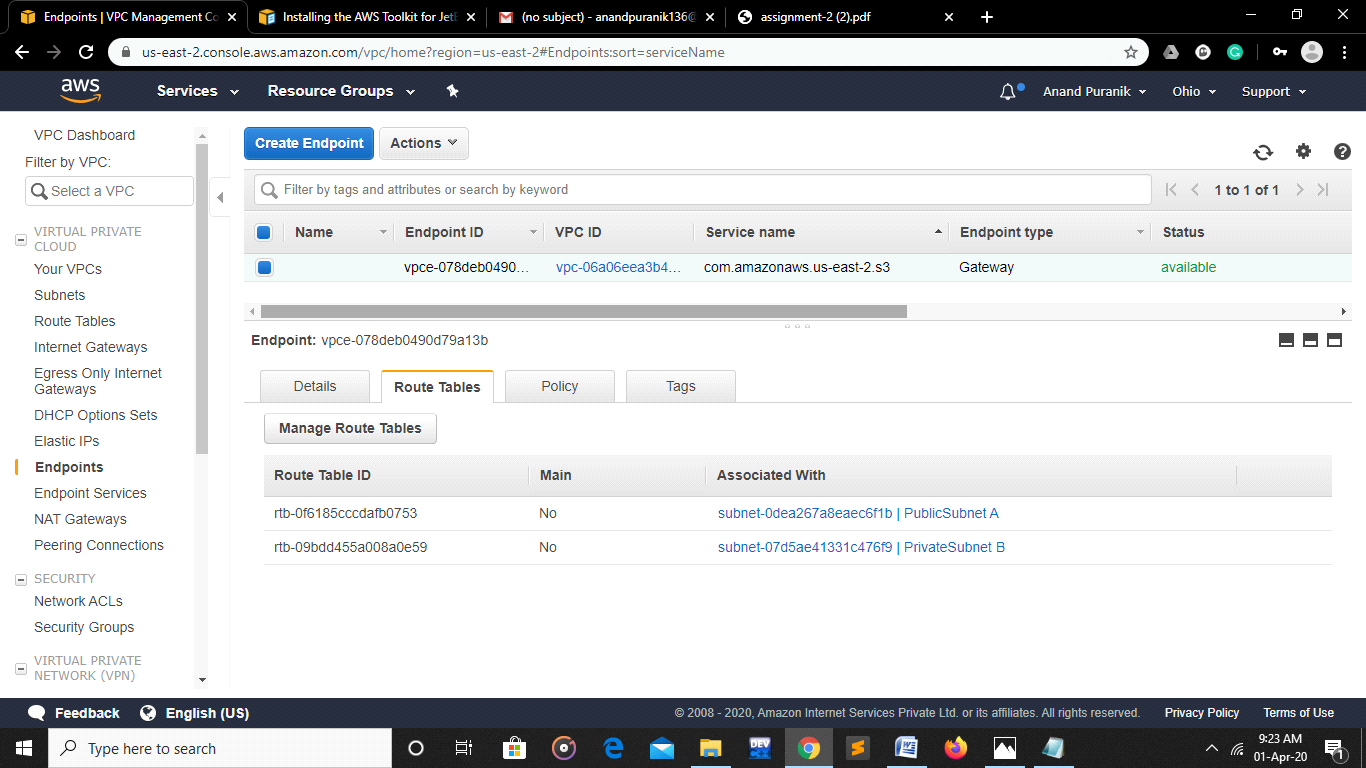
a. Select custom vpc while creating

b. After creating Route set Target as Internet Gateway

c. Associate Public Subnet to Public Route Table

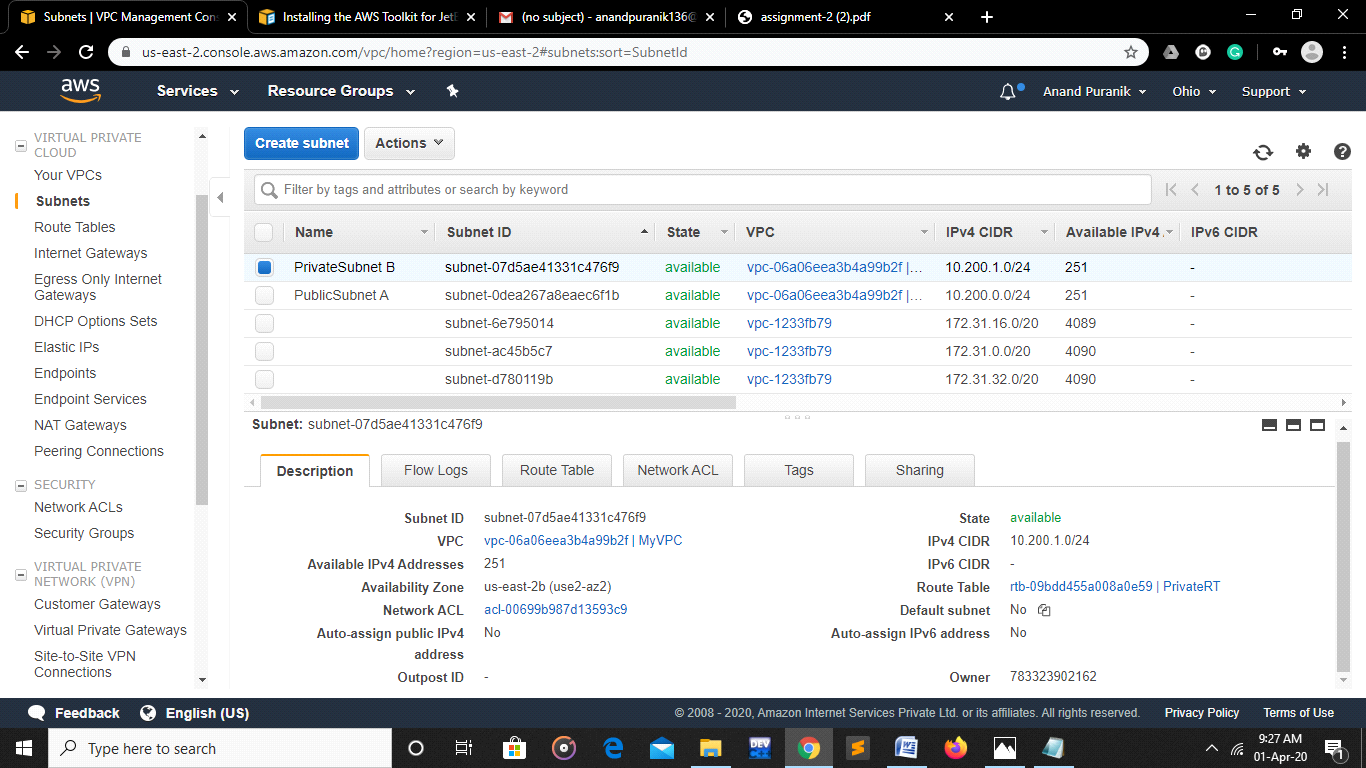


Creating end point to custom vpc and attach to private subnet

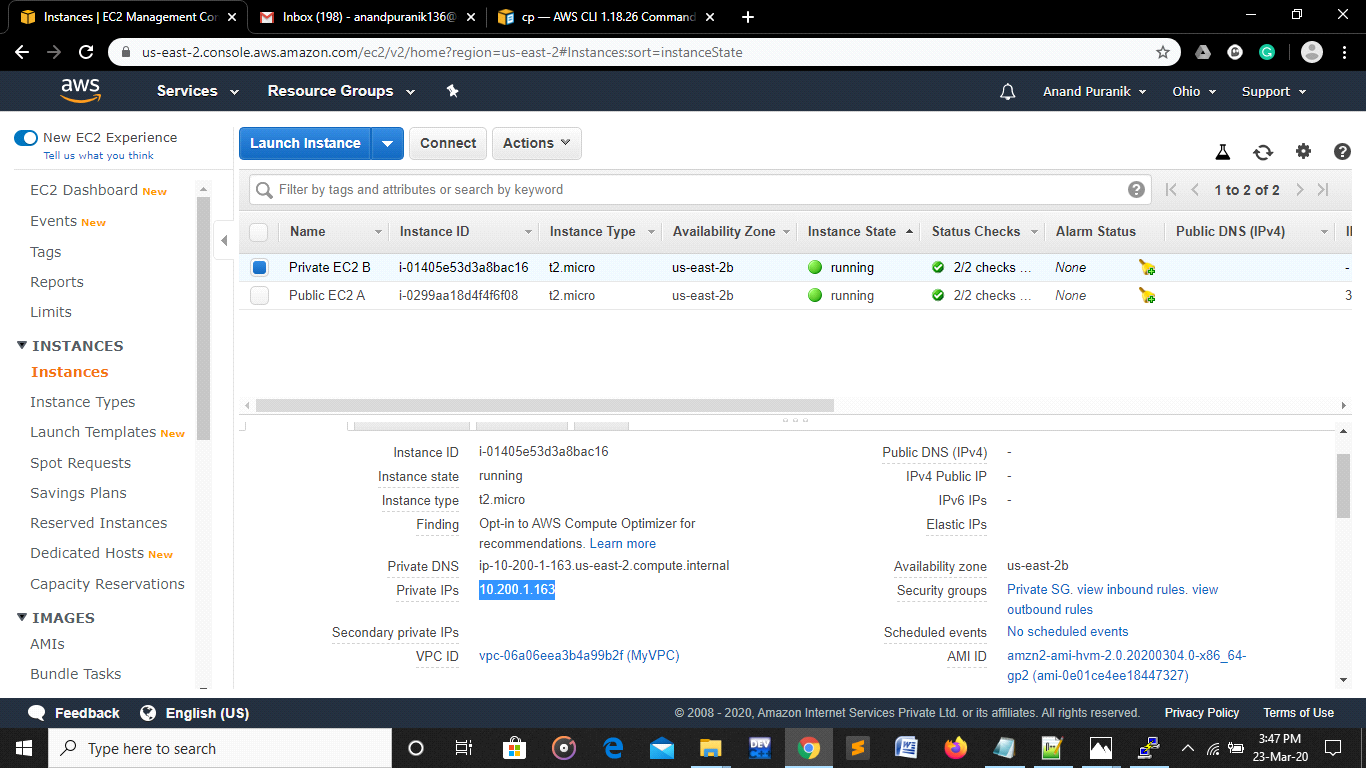


Select Custom VPC

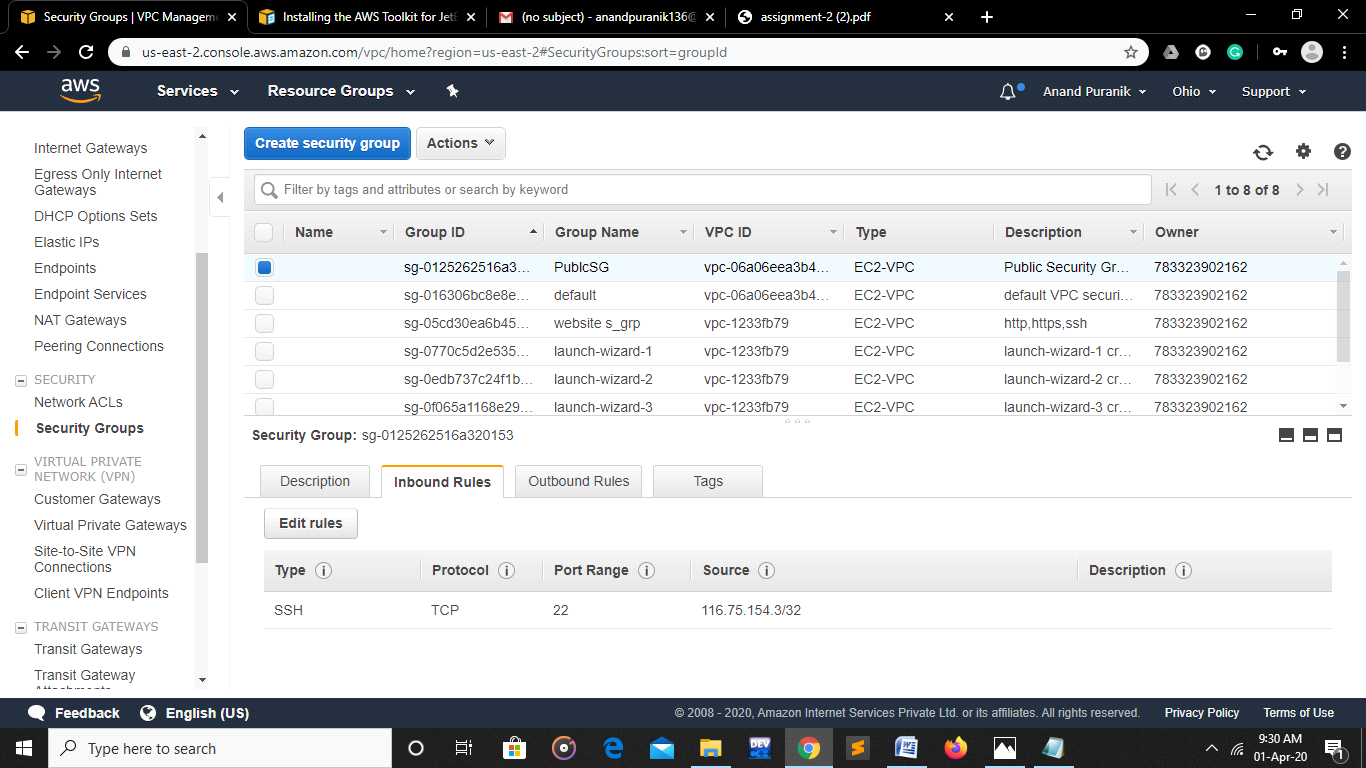
Select public/private subnet

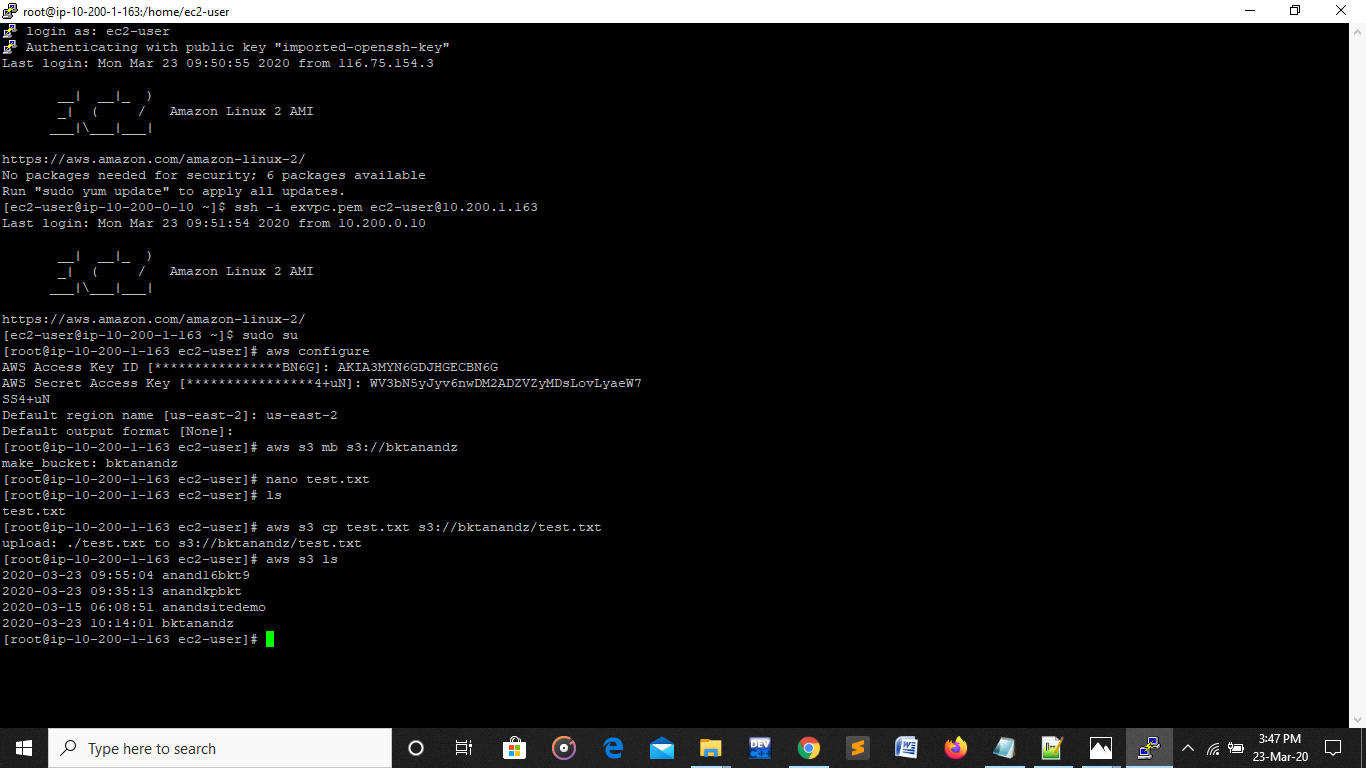


Create Two EC2 Instances-



Create Security Groups-





With the help of putty connection is established.

Connecting Public EC2 with Private EC2 using SSH cmd Command & .pem file.

Create IAM user to access s3 bucket :

a. Create user group by having S3 full (access) as a policy

b. Attach user which we created

* **Bucket is created using aws S3 mb S3://btkanandz**
* **Upload file using aws s3 cp s3://bktanandz**

**Assignment 3**

Create autoscaling of instance with min and Max load on the basis of cpu  
Attach the instance to Alb

1.creating 2 ec2 instances

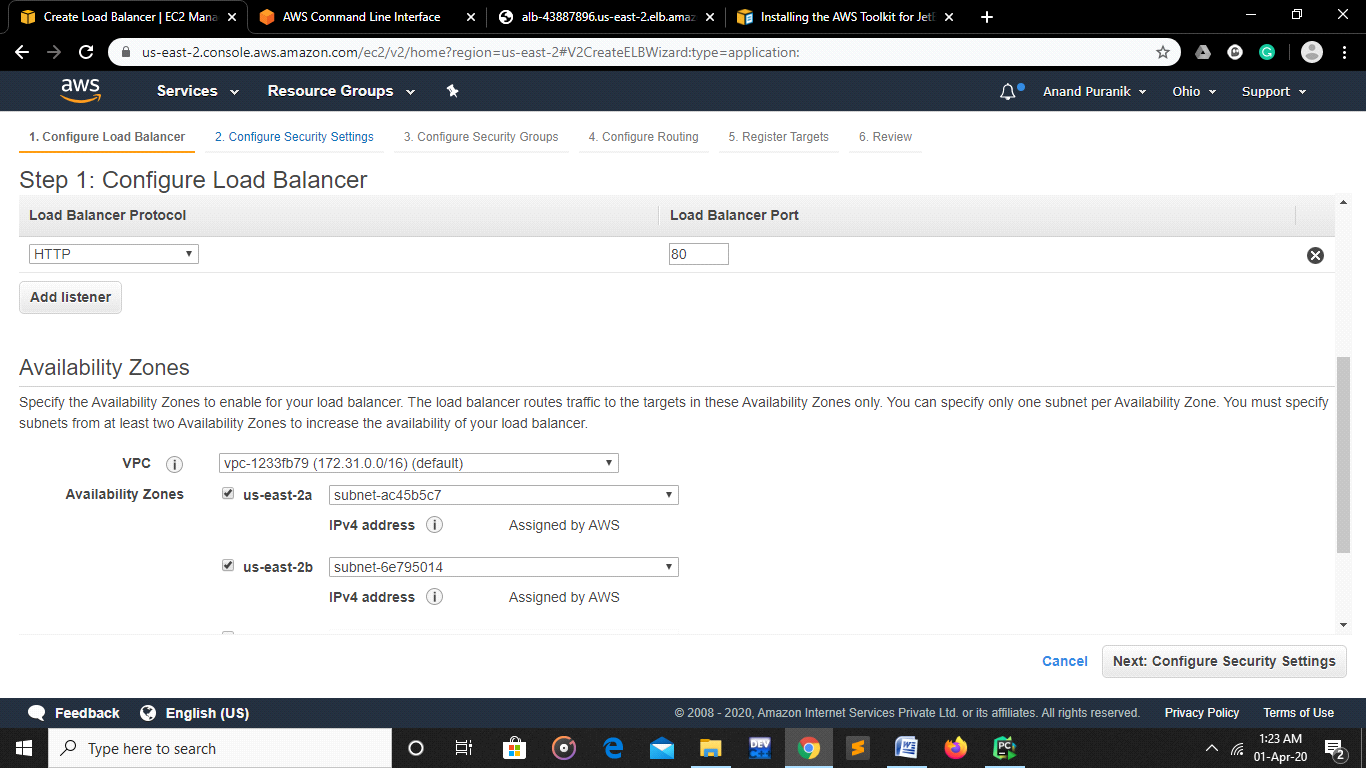


2. creating target group:

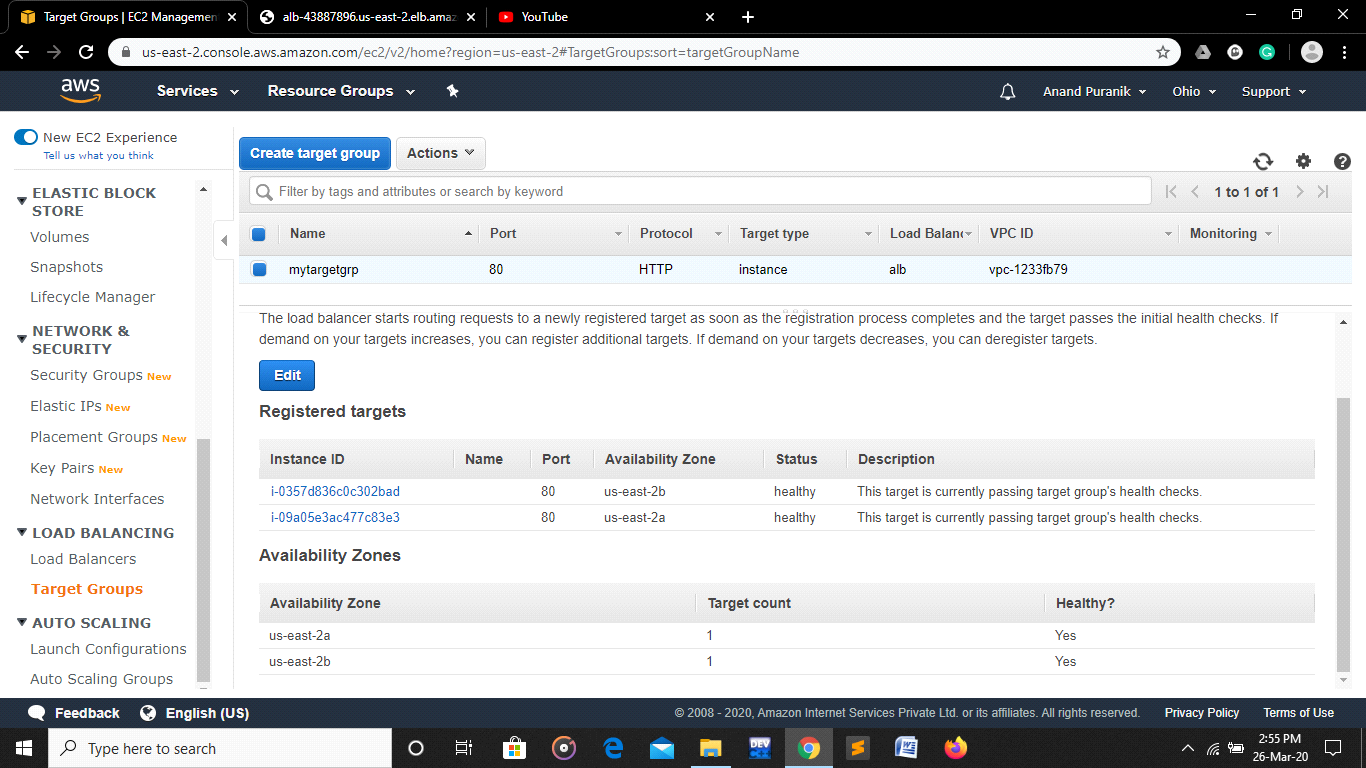


3. creating alb

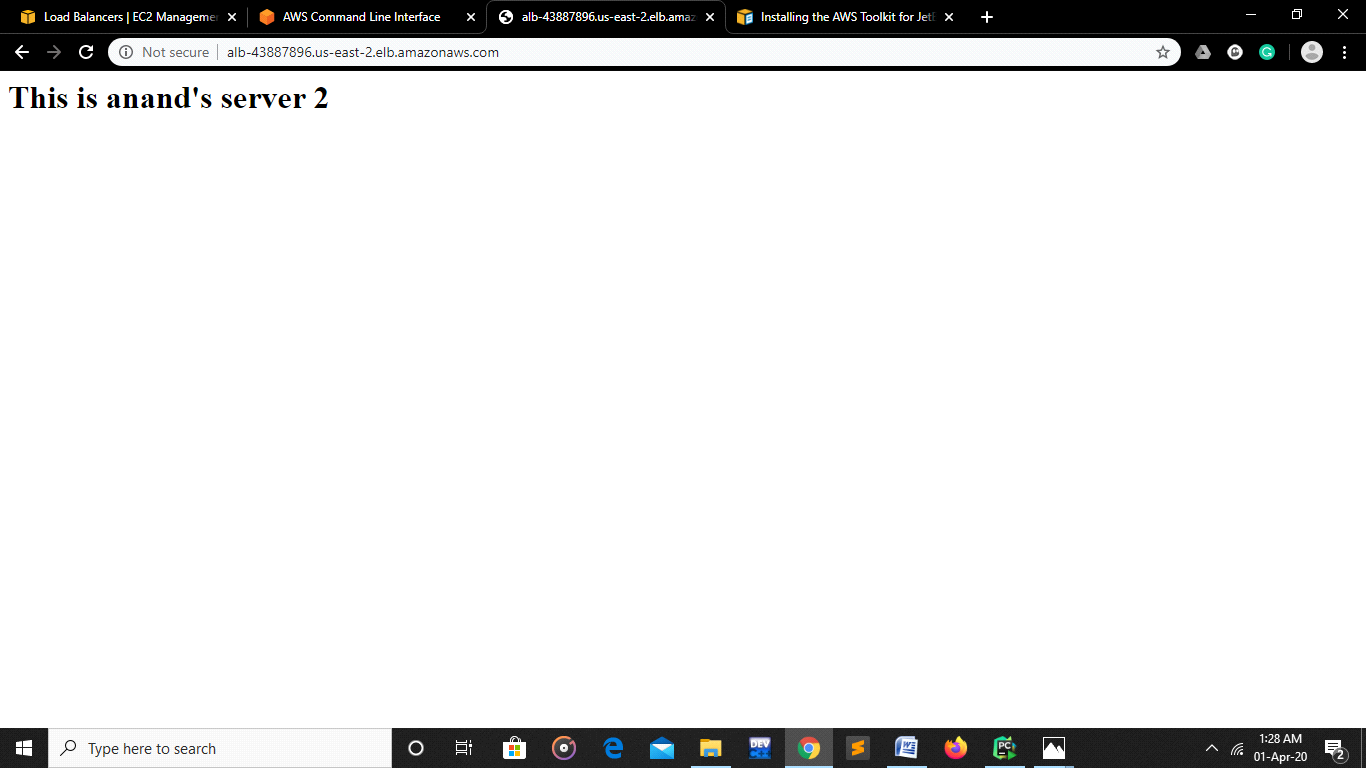
default vpc using atleat 2 subnet



Giving existing target grp-



4. Copy the DNS address and test the results

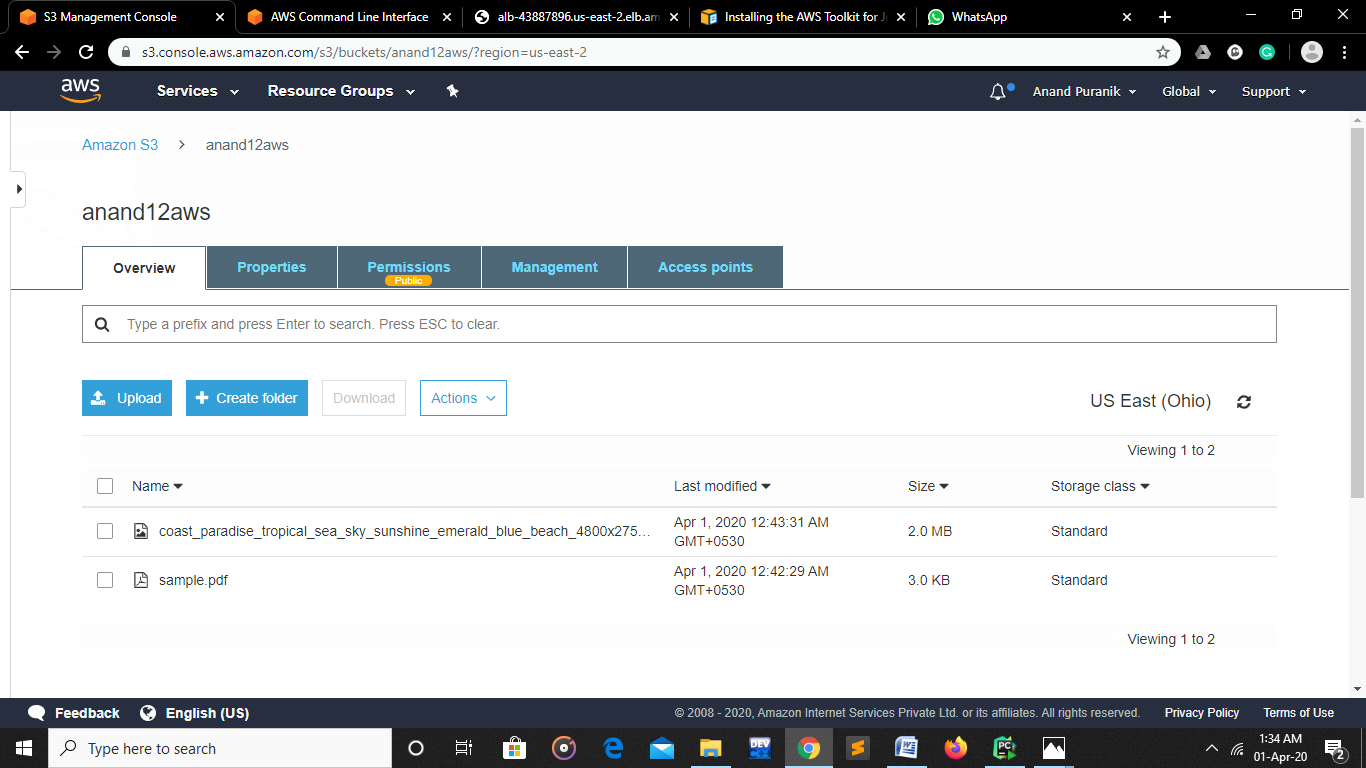


**Assignment 4**

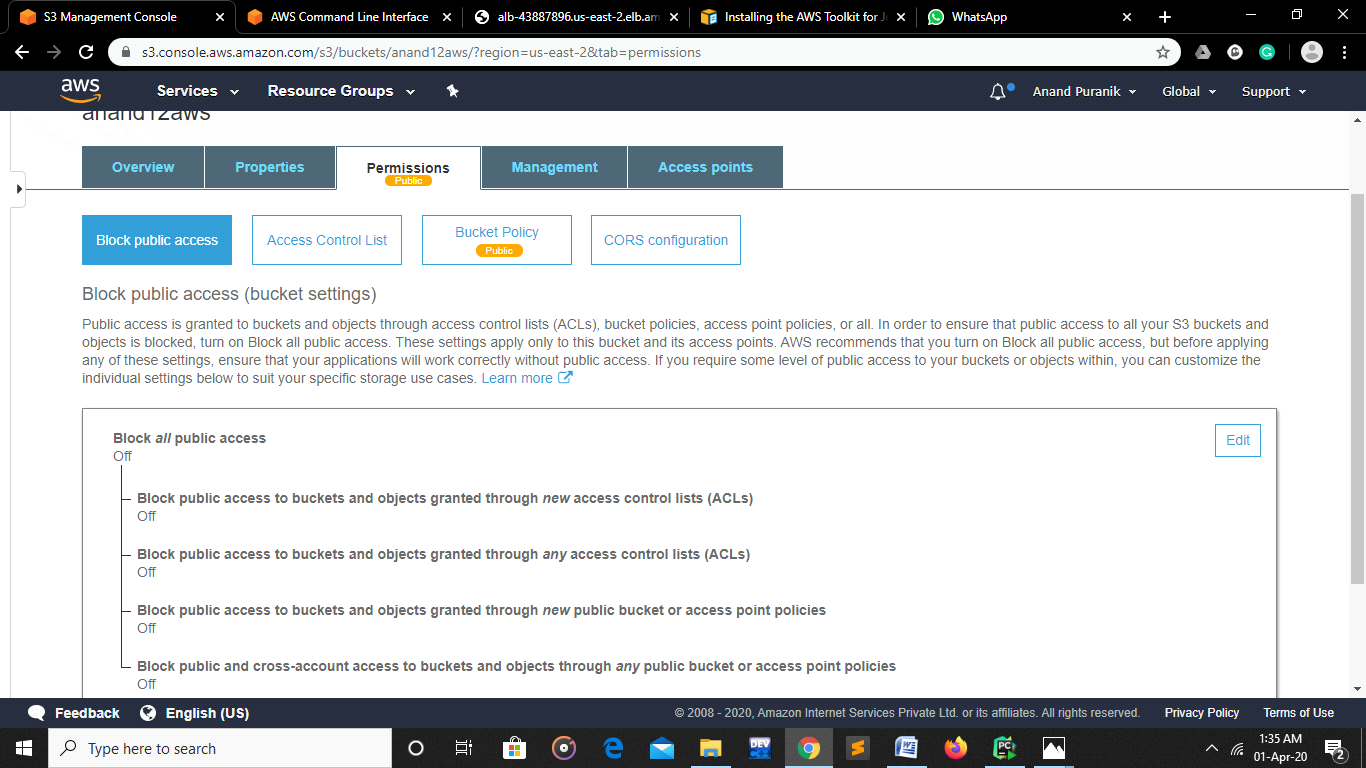
Create a bucket policy Upload some images pdf Which will give public access only to images and not to PDF

Steps

* Creating s3 bucket



1. While creating s3 bucket choose Block all public access “Off”
2. Uploading .pdf and .jpg file to newly created bucket



1. Creating bucket policy using following code:

{

    "Version": "2012-10-17",

    "Statement": [

        {

            "Sid": "PublicRead",

            "Effect": "Allow",

            "Principal": "\*",

            "Action": "s3:GetObject",

            "Resource": "arn:aws:s3:::anand12aws/\*.jpg"

        },        {

            "Sid": "PublicRead",

            "Effect": "Deny",

            "Principal": "\*",

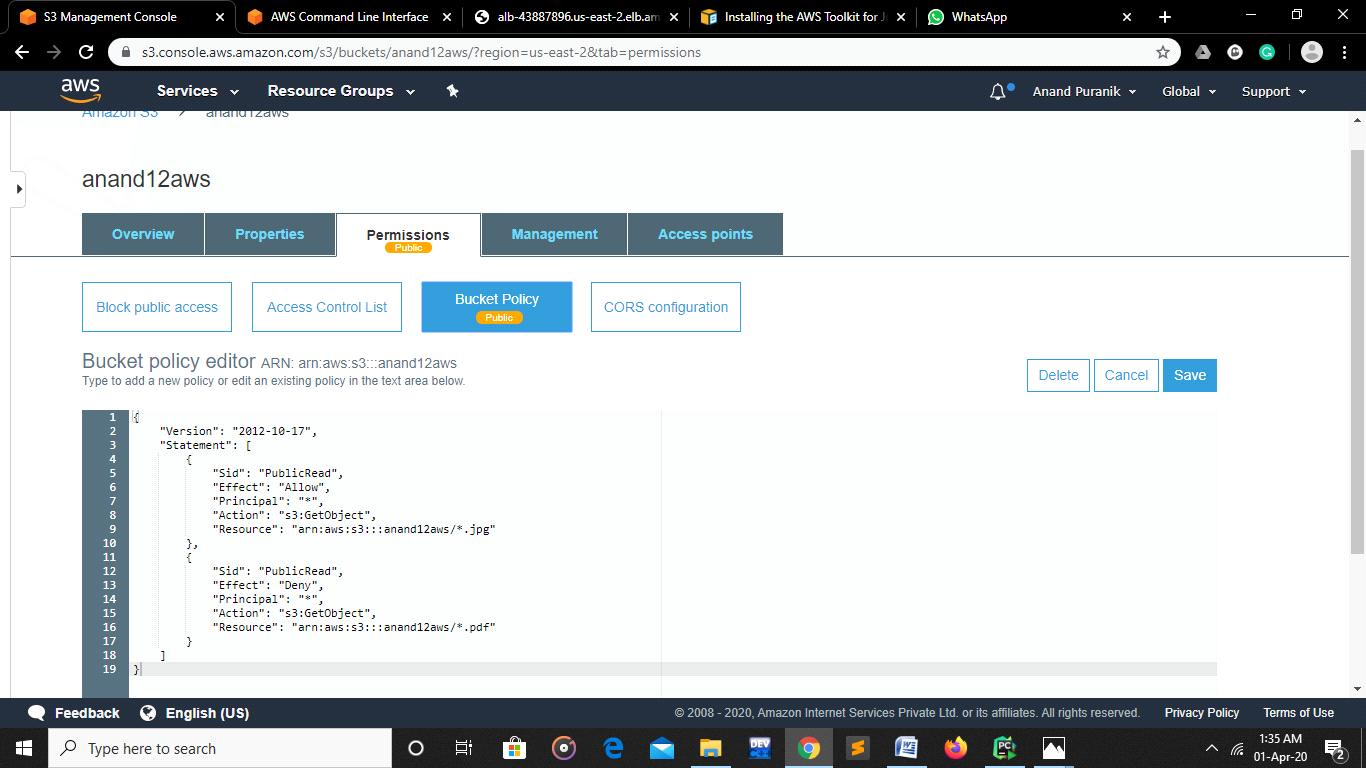
            "Action": "s3:GetObject",

            "Resource": "arn:aws:s3:::anand12aws/\*.pdf"

        }

    ]

}



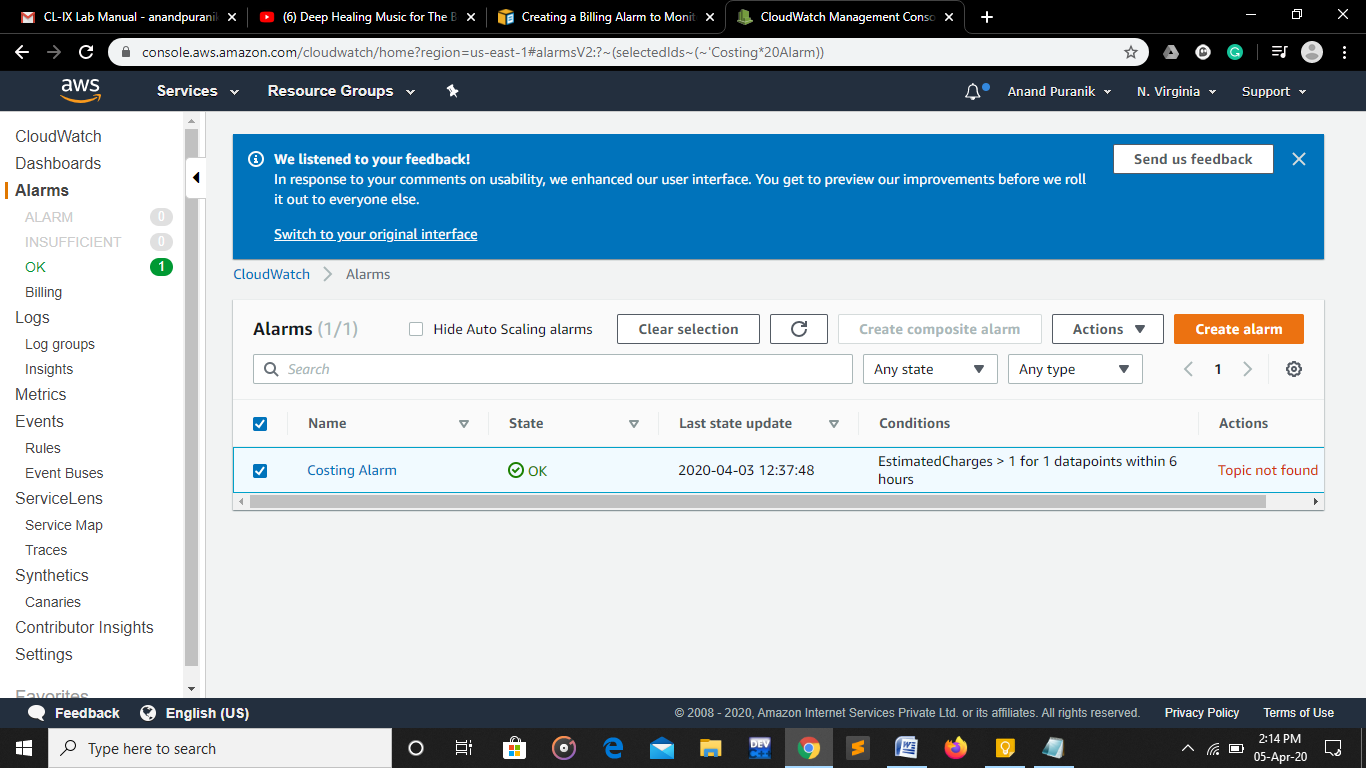
**Assignment 5  
1. Create a billing alarm on your account to check if your account billing exceeds $1 for the current month.**

**To enable the monitoring of estimated charges**

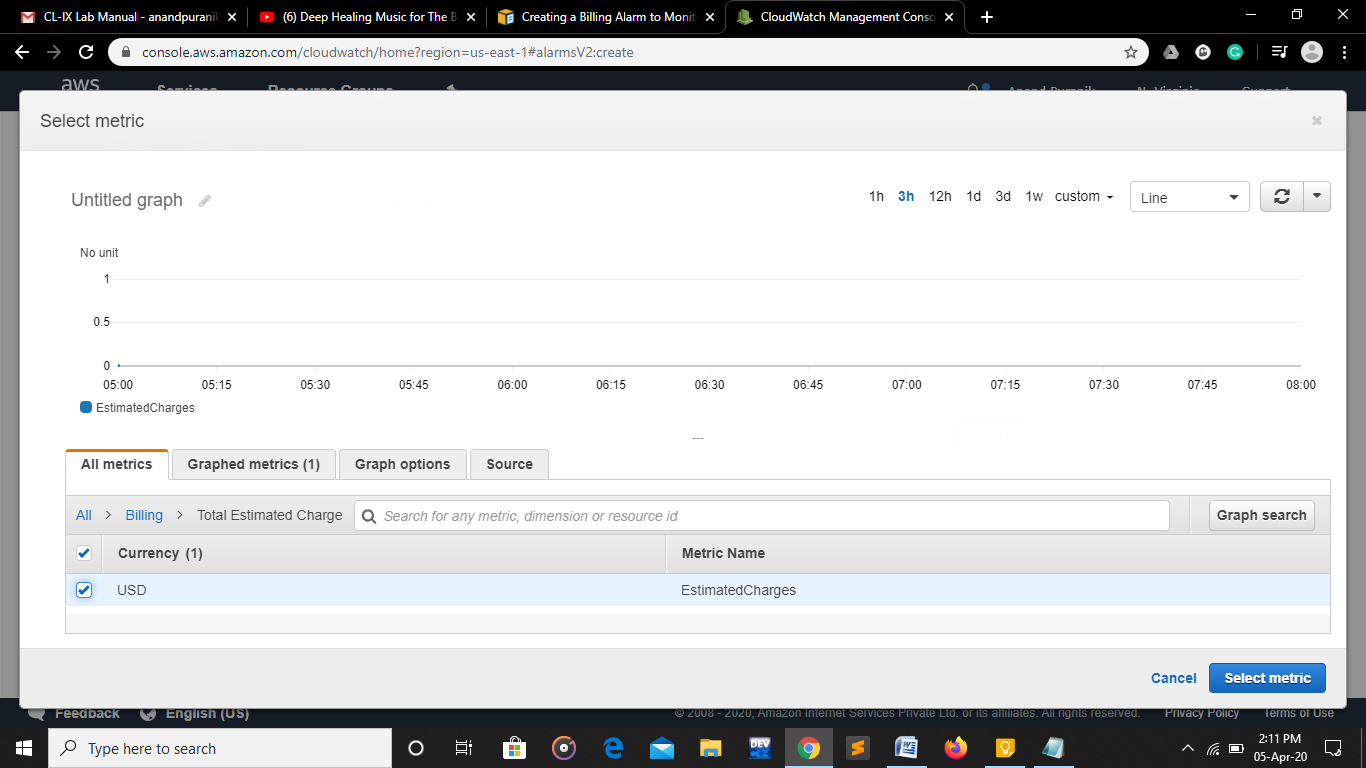
1. Open the Billing and Cost Management console
2. In the navigation pane, choose **Billing Preferences**.
3. Choose **Receive Billing Alerts**.
4. Choose **Save preferences**.

**To create a billing alarm using the CloudWatch console**

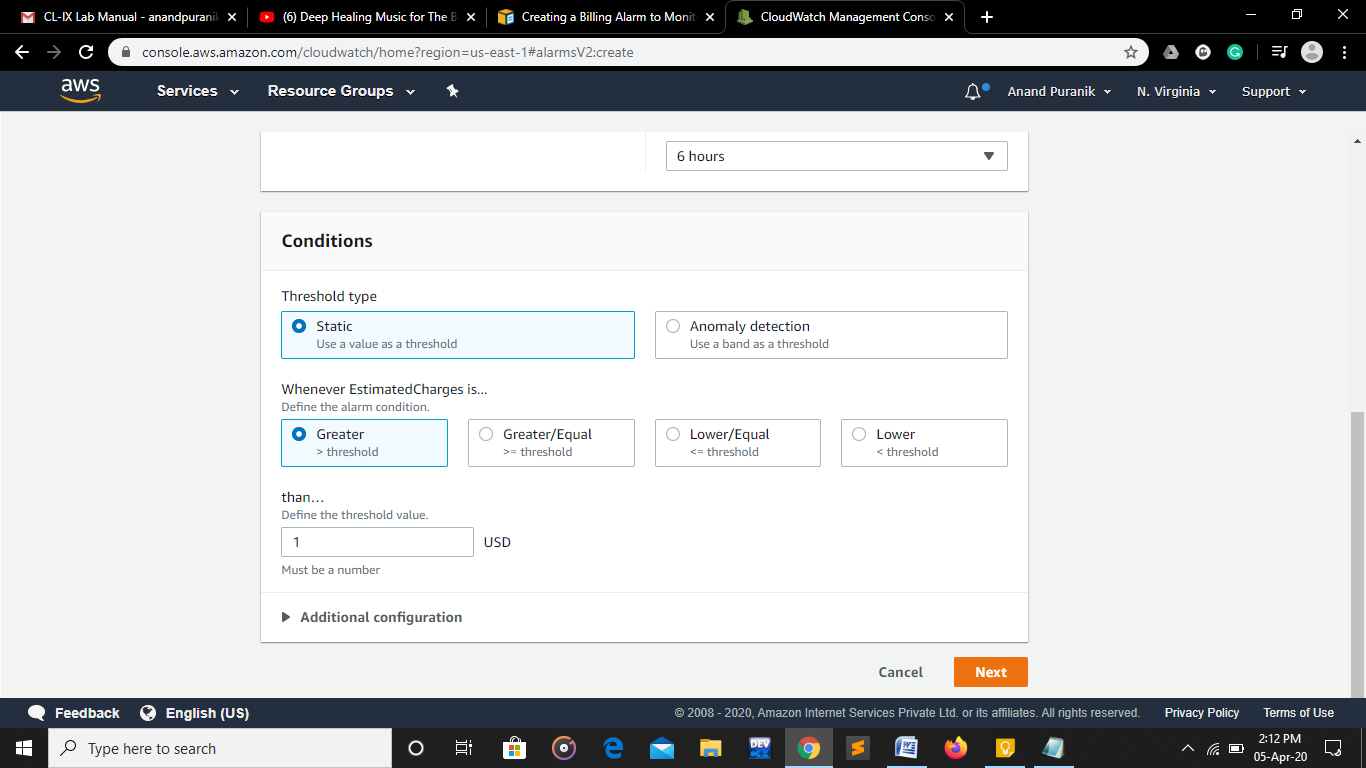
1. Select N Virginia
2. In the navigation pane, choose **Alarms**, **Create Alarm**.



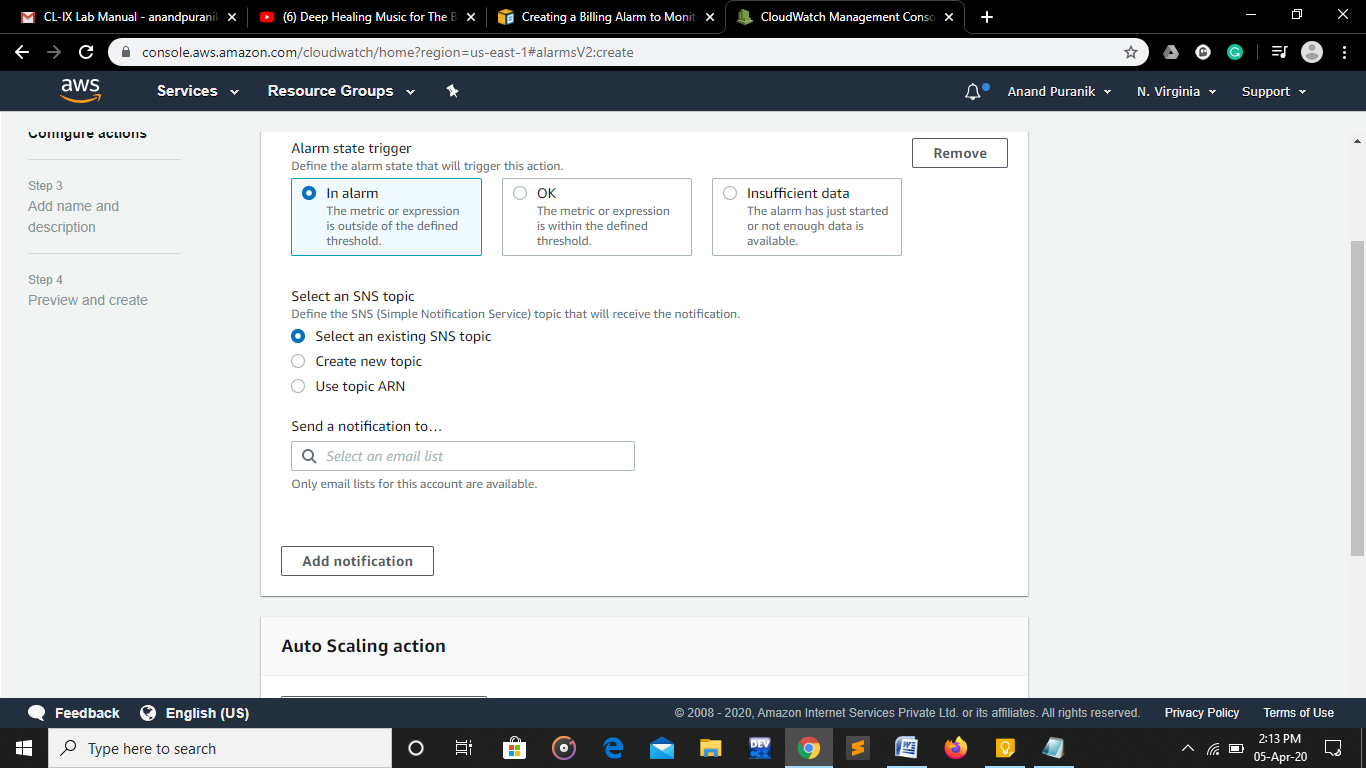
1. Choose **Select metric**. In the **All metrics** tab, choose **Billing**, **Total Estimated Charge**.
2. Select the check box next to **EstimatedCharges**, and choose **Select metric**.



1. Under **Conditions**, choose **Static**.
2. For **Whenever EstimatedCharges is**, choose **Greater**.
3. For **than**, enter the monetary amount (for example, **1$**) that must be exceeded to trigger the alarm.



1. Choose **Next**.



1. Under **Notification**, select an SNS topic to notify when the alarm is in alarm state.

To have the alarm send multiple notifications for the same alarm state or for different alarm states, choose **Add notification**.

1. When finished, choose **Next**.
2. Enter a name and description for the alarm. The name must contain only ASCII characters. Then choose **Next**.
3. Under **Preview and create**, confirm that the information and conditions are what you want, then choose **Create alarm**.

