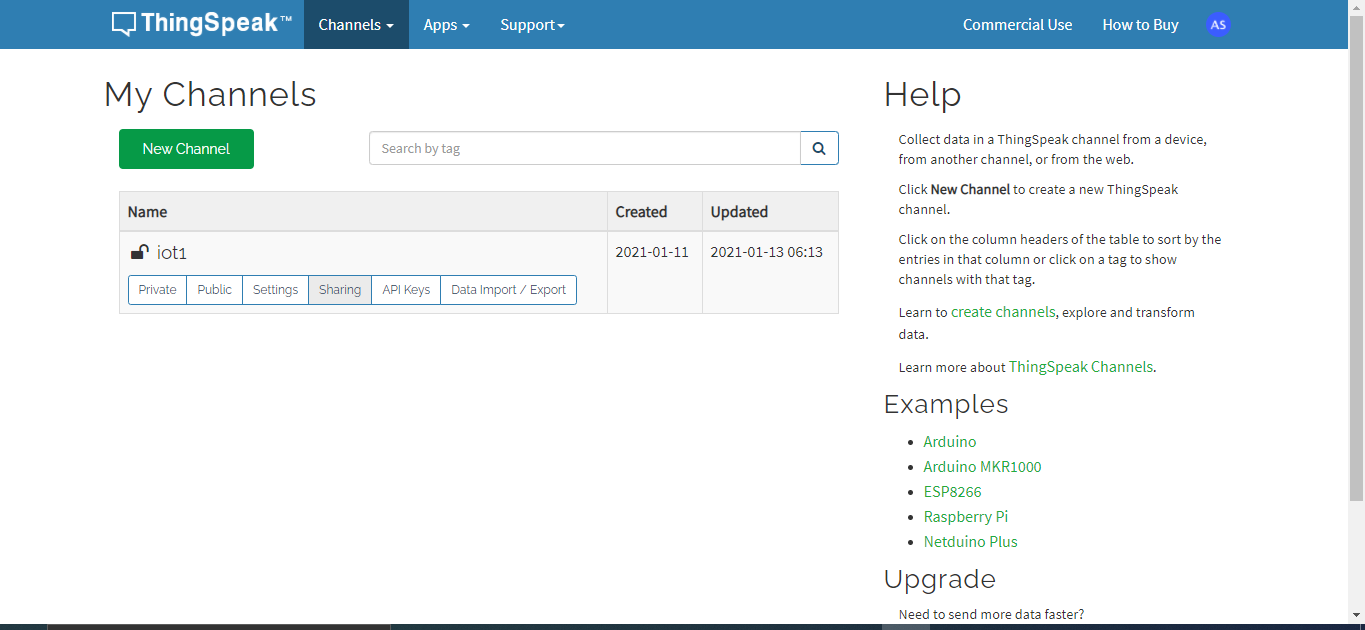
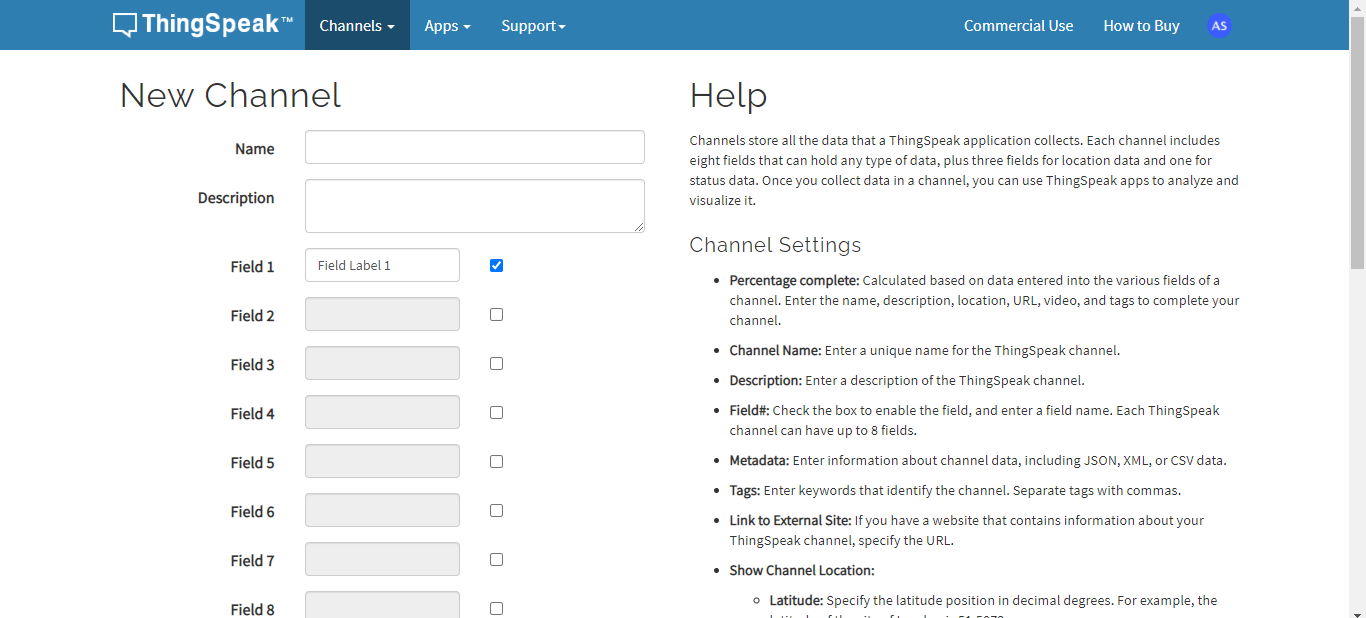
**EC2 INSTANCE AWS**

**Attempt 1:**

**API on ThinkSpeak-**

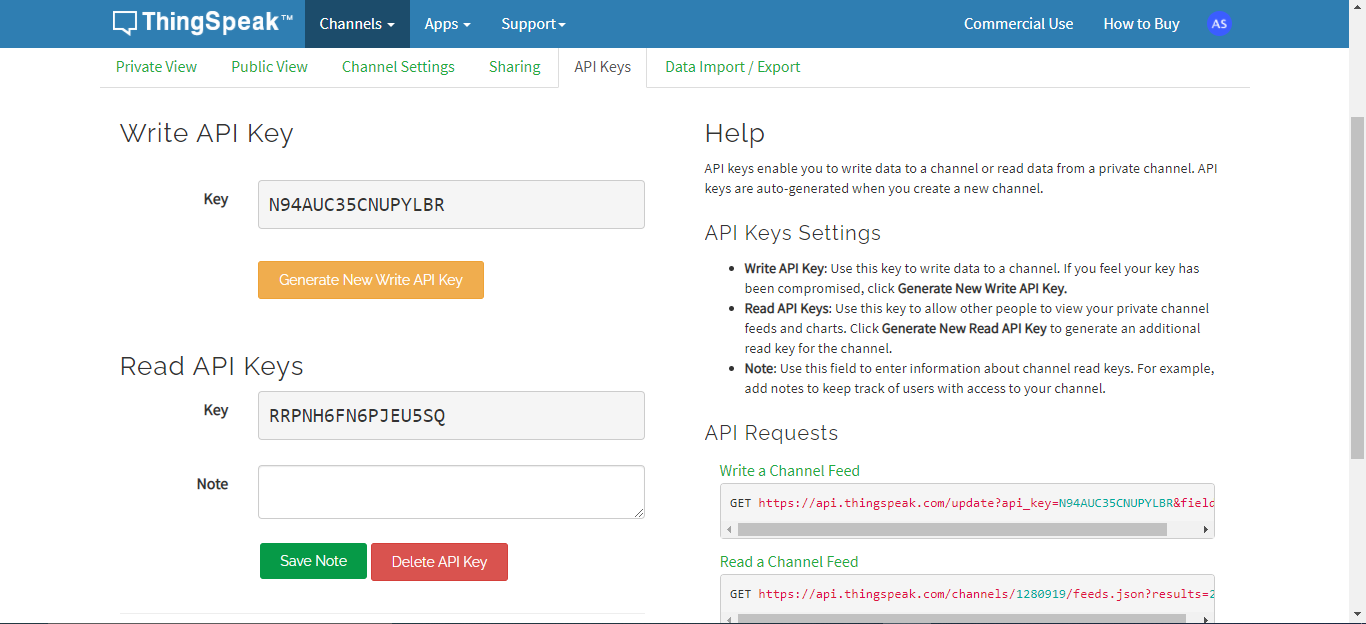
1. Login with user id and password and go to “New Channel” **Setting Channel to Public**

****

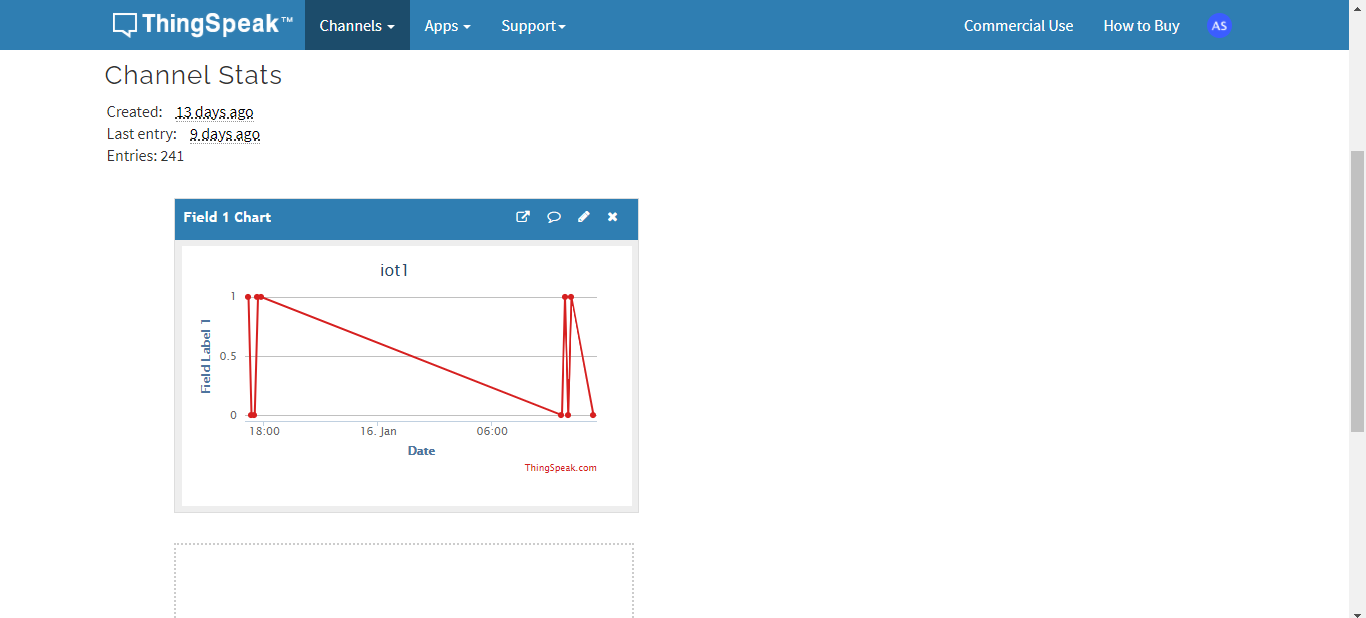
****

1. **Read & write API**

**Go to API keys, here is the link for reading and writing API.**

****

1. **Channel Status**



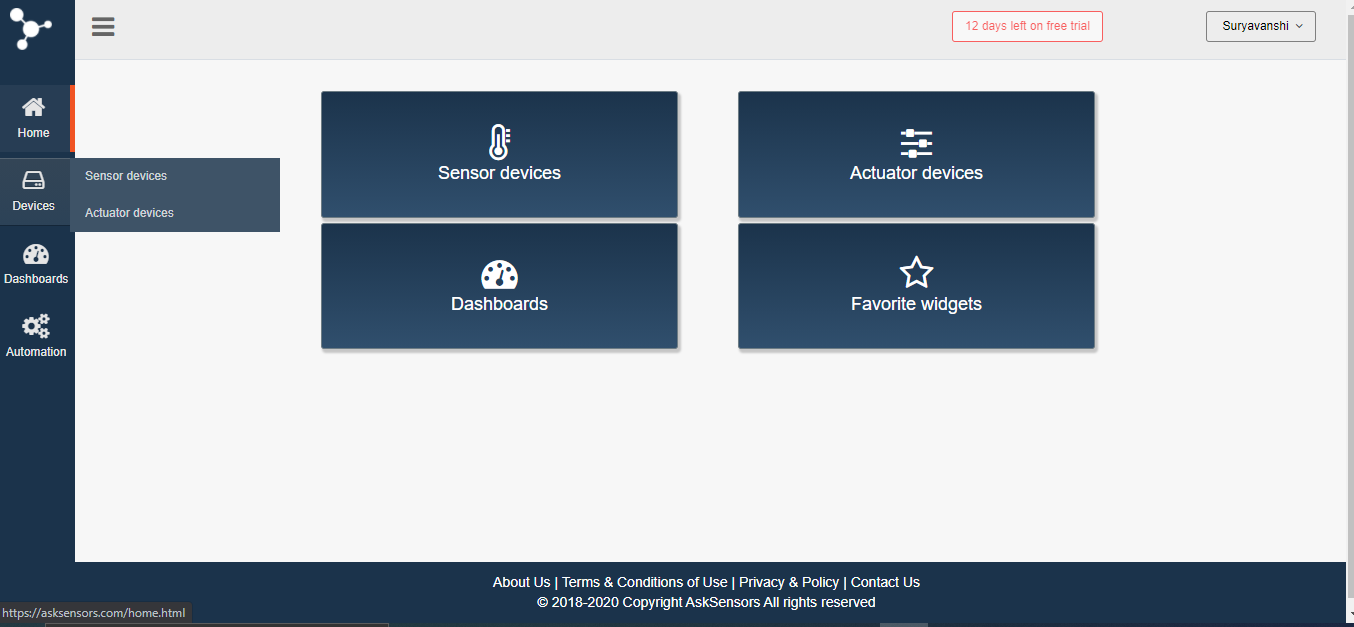
**Reason for Rejection this API platform – High Cost**

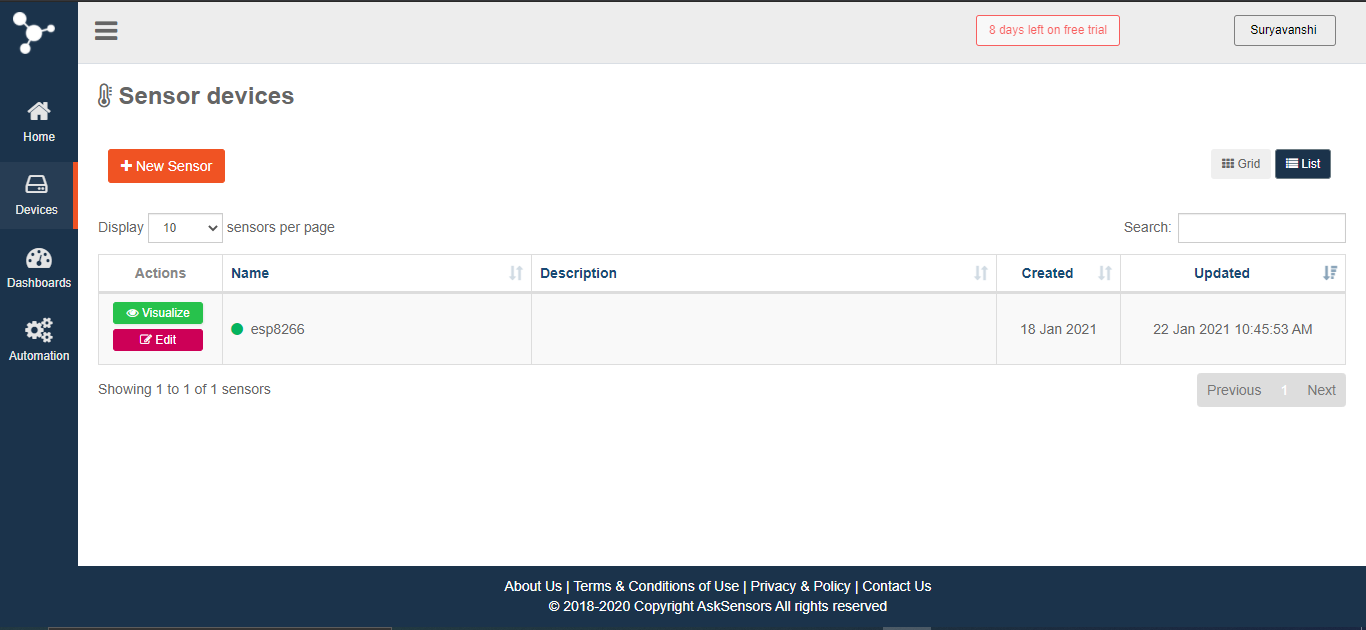
**Attempt 2:**

**API on Asksensor:**

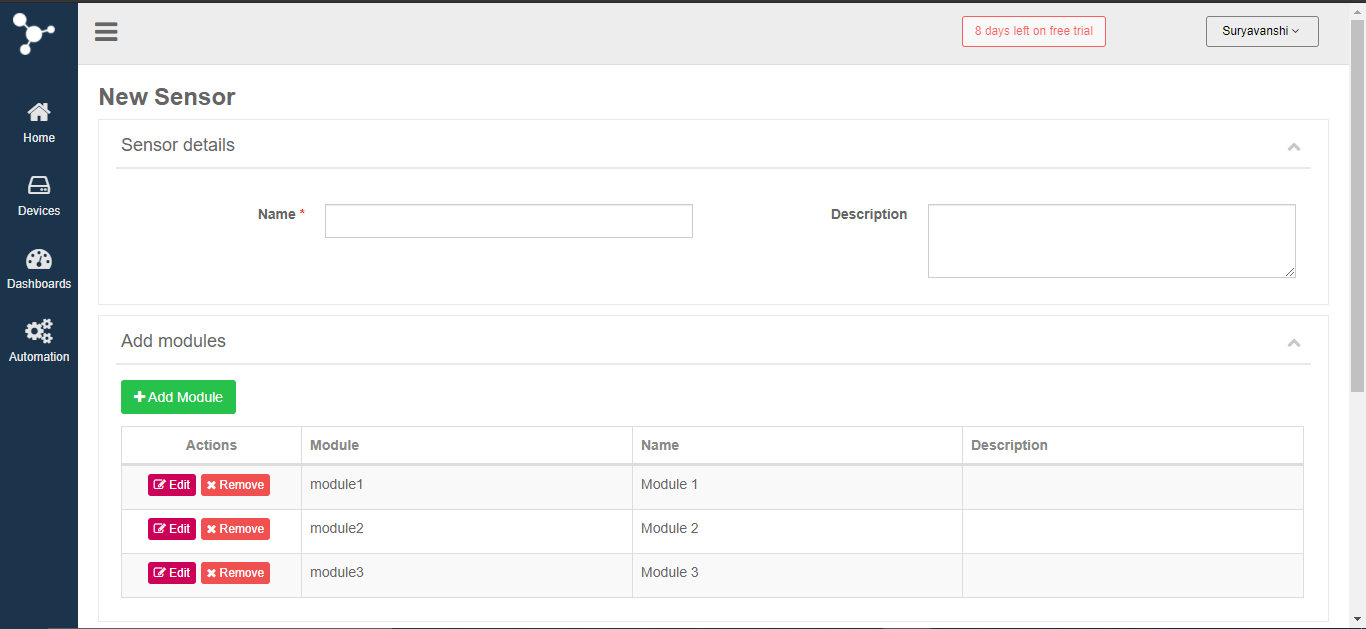
1. **Login with user id and password. Start 15 days free trial Creating API**

with “simple plan”. Go to “sensors and devices” and select “new sensor”.

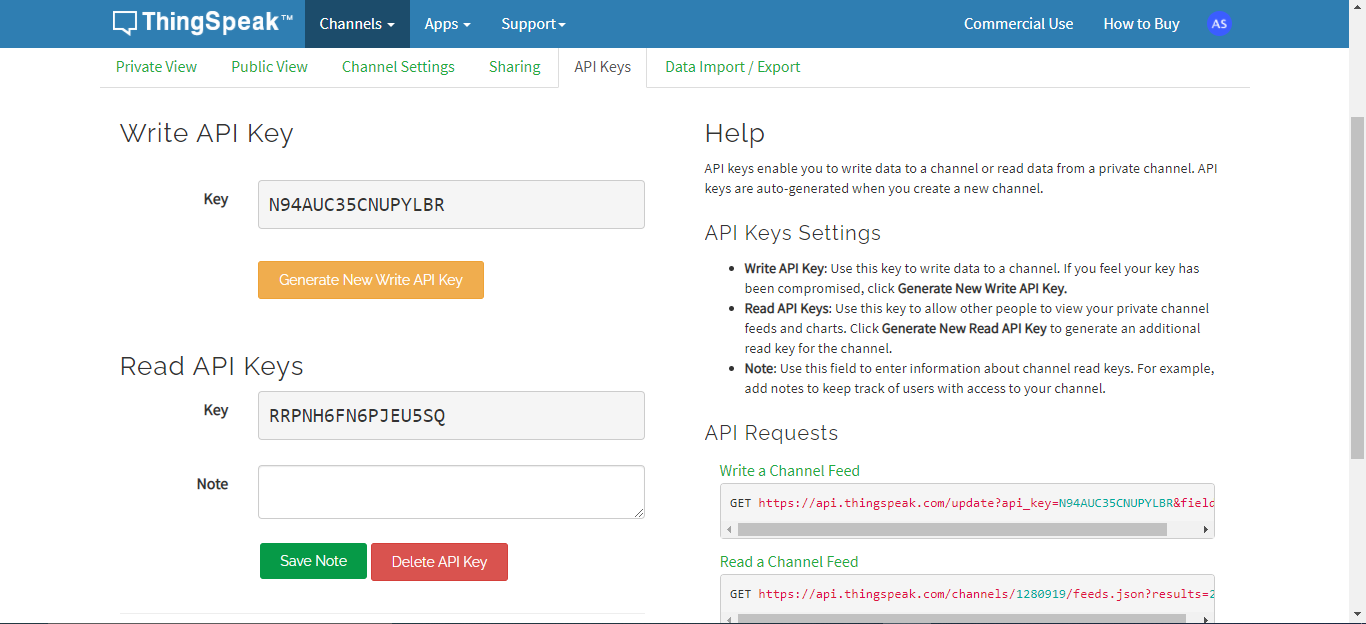




1. **Add modules**



1. **Select your sensor for details of read and write API.**

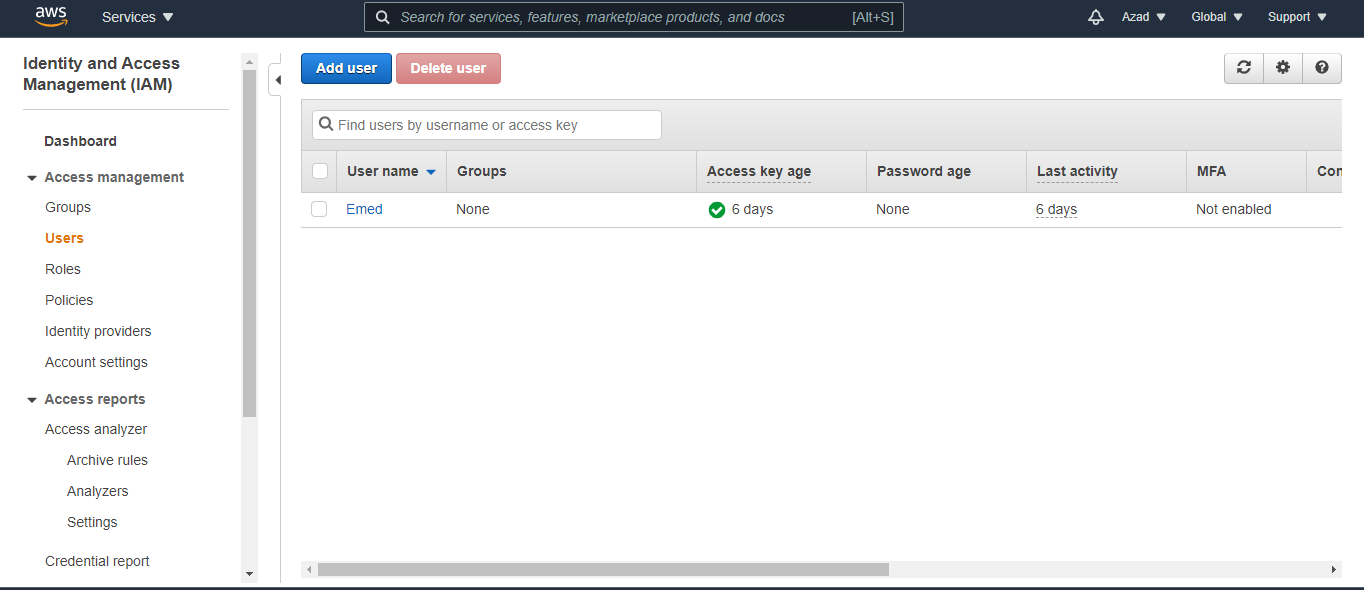


**Reason for Rejection this API platform – Cannot read all modules at a time.**

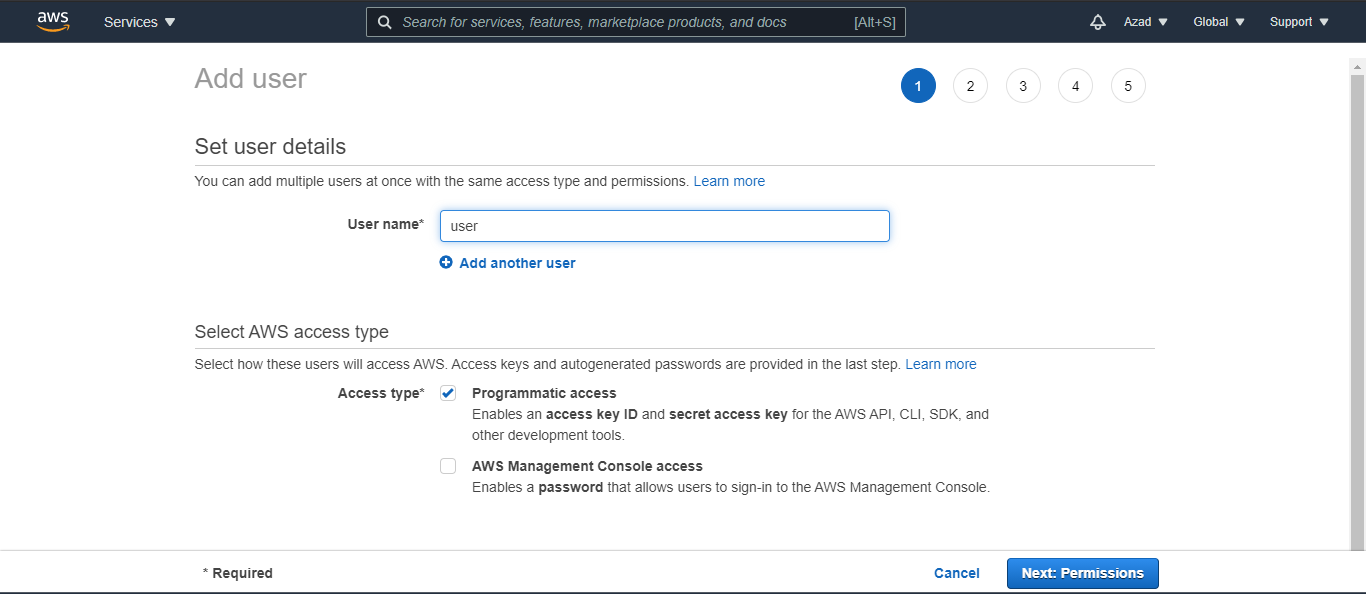
**Attempt 3:**

**Amazon S3**

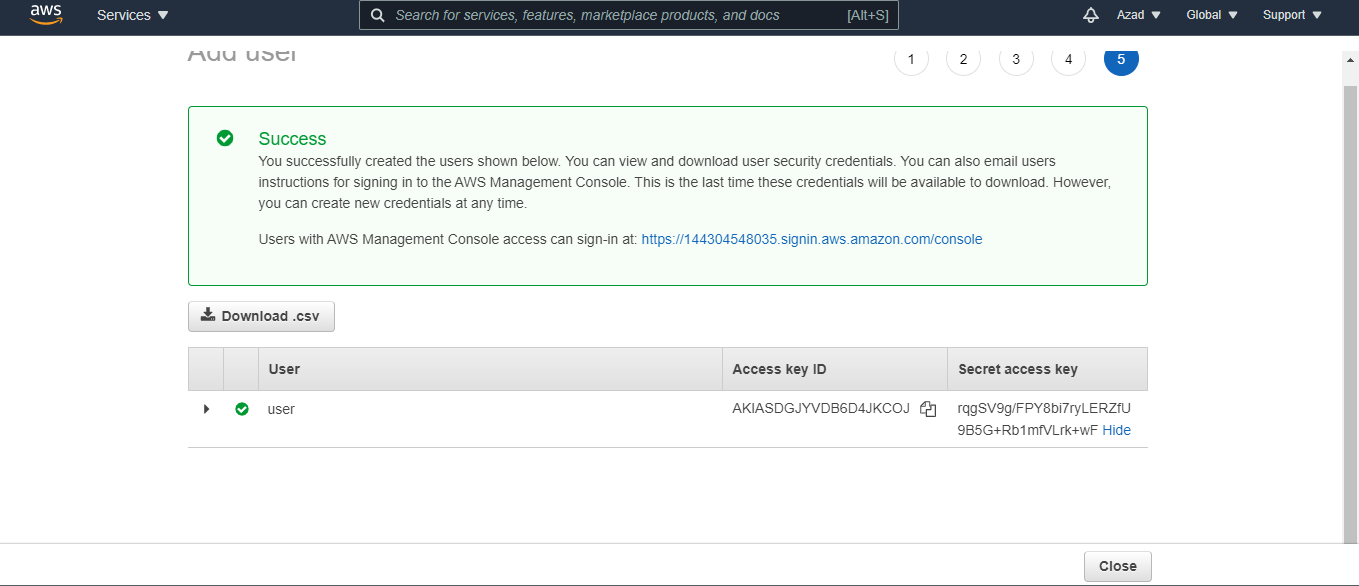
1. **Creating a free tier account and Go to services>IAM>users>add user**

****

1. **Add user name and mark tick to Programmatic access.**

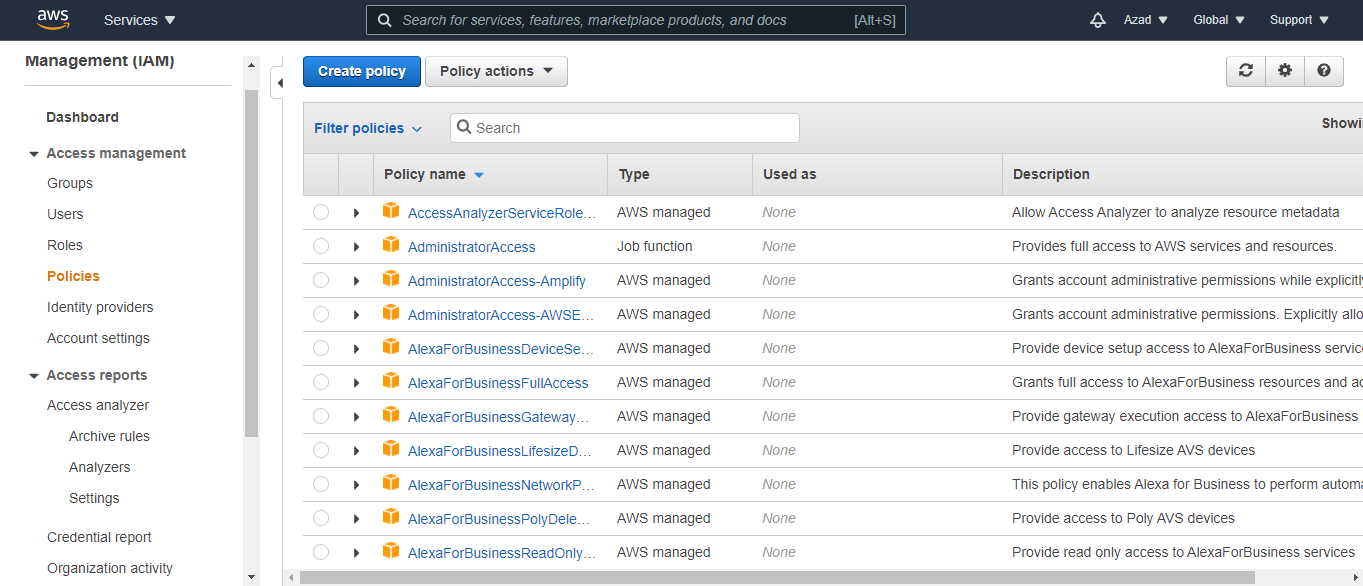
****

1. **Download csv file which contains user id and key.**

****

1. **Create policy**

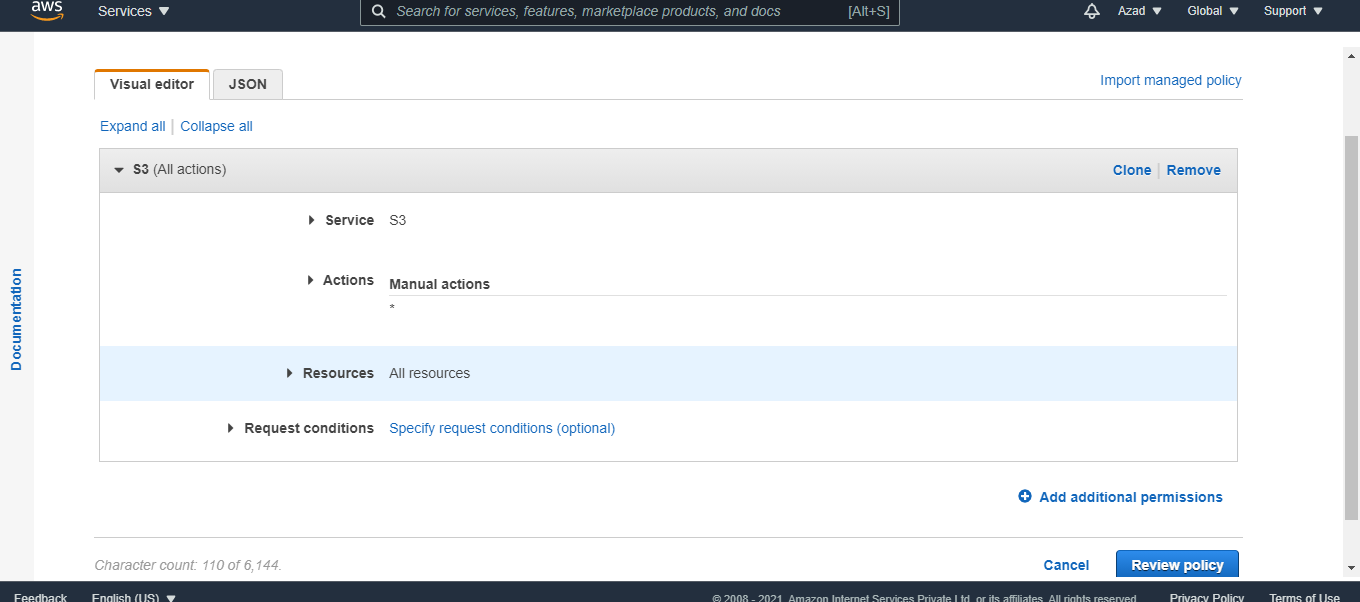
**Go to policies>create policy**

****

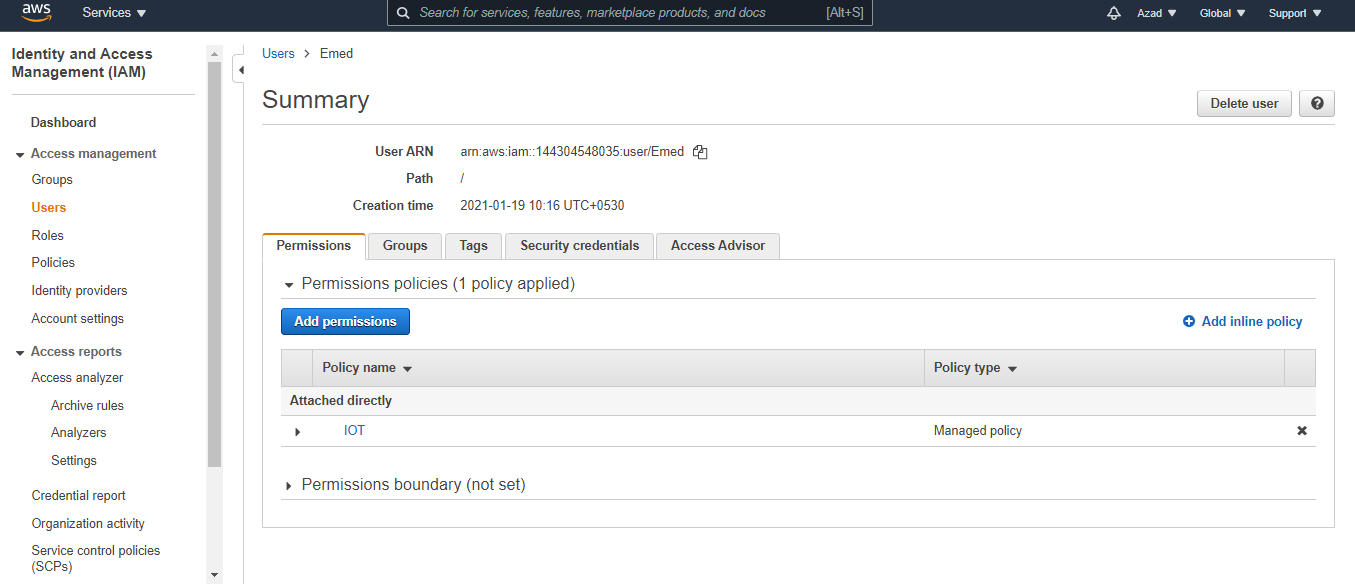
1. **Select Services -> S3**

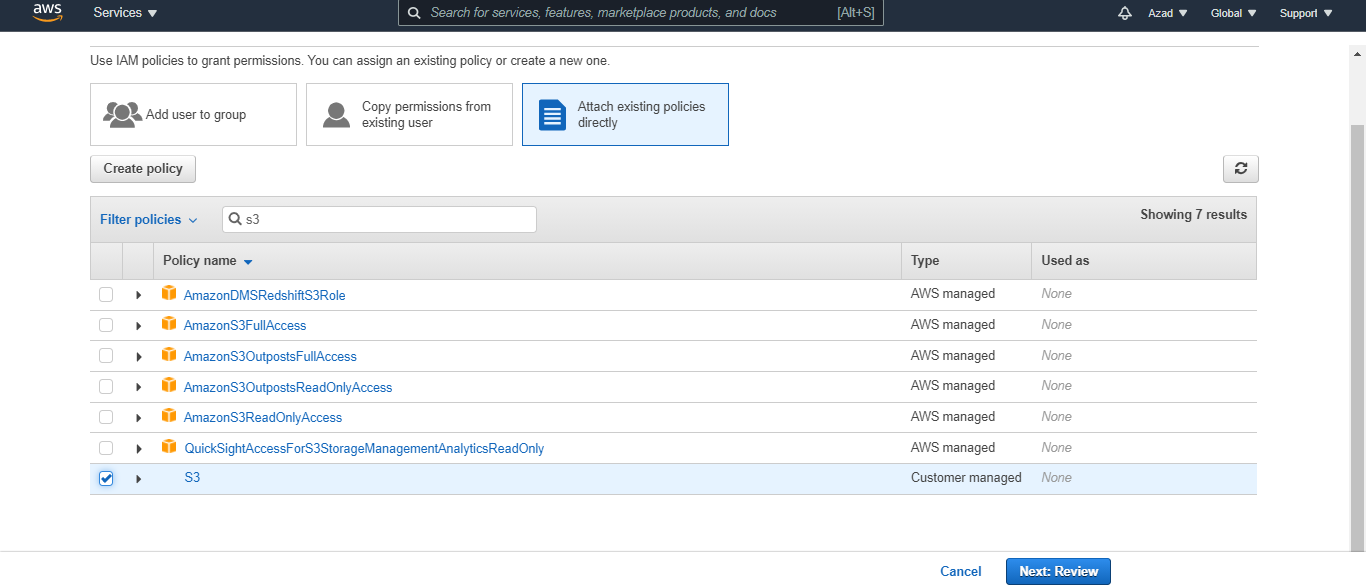
**Actions -> Manual Actions**

**Resources -> All resources**

****

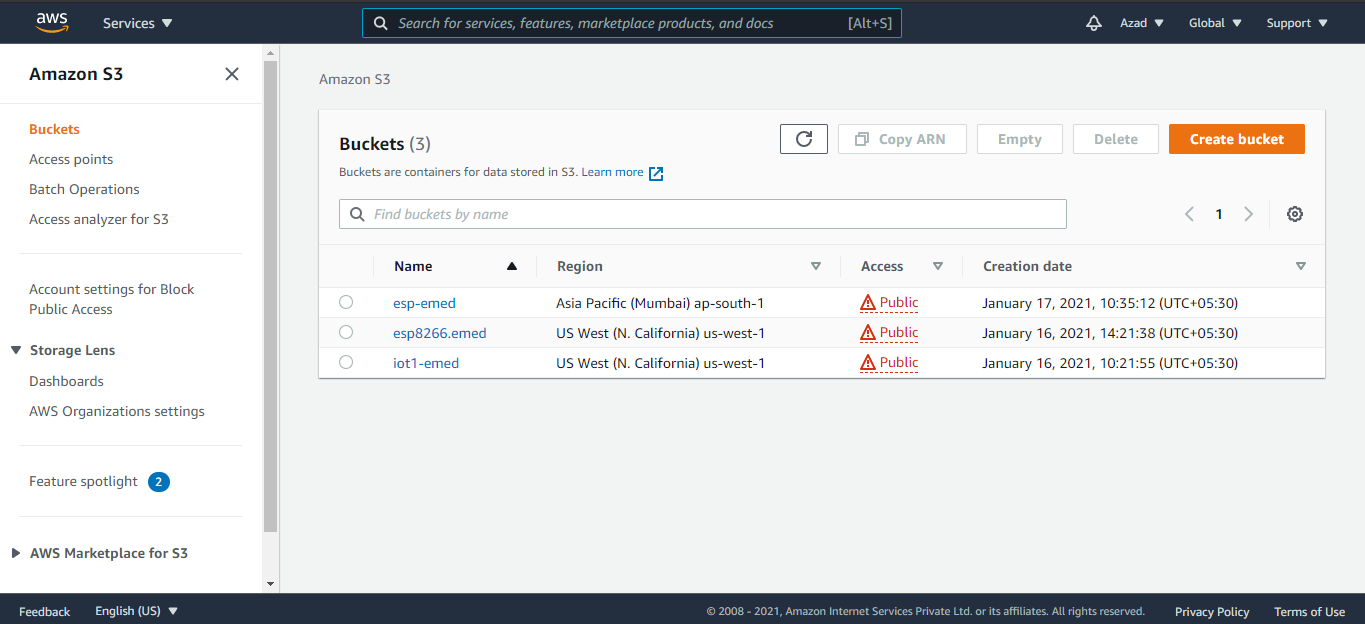
1. **Attach Policy  
   Go to IAM>users>myuser>add permissions>add existing policy and search “s3”**

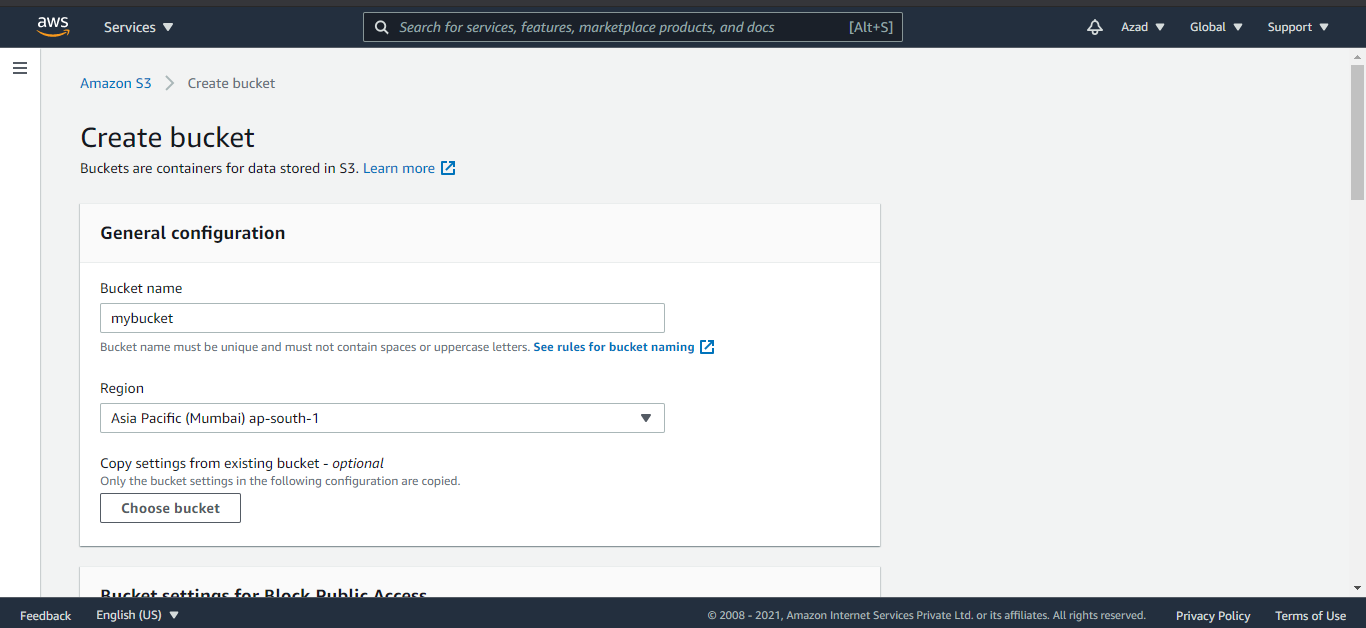
****

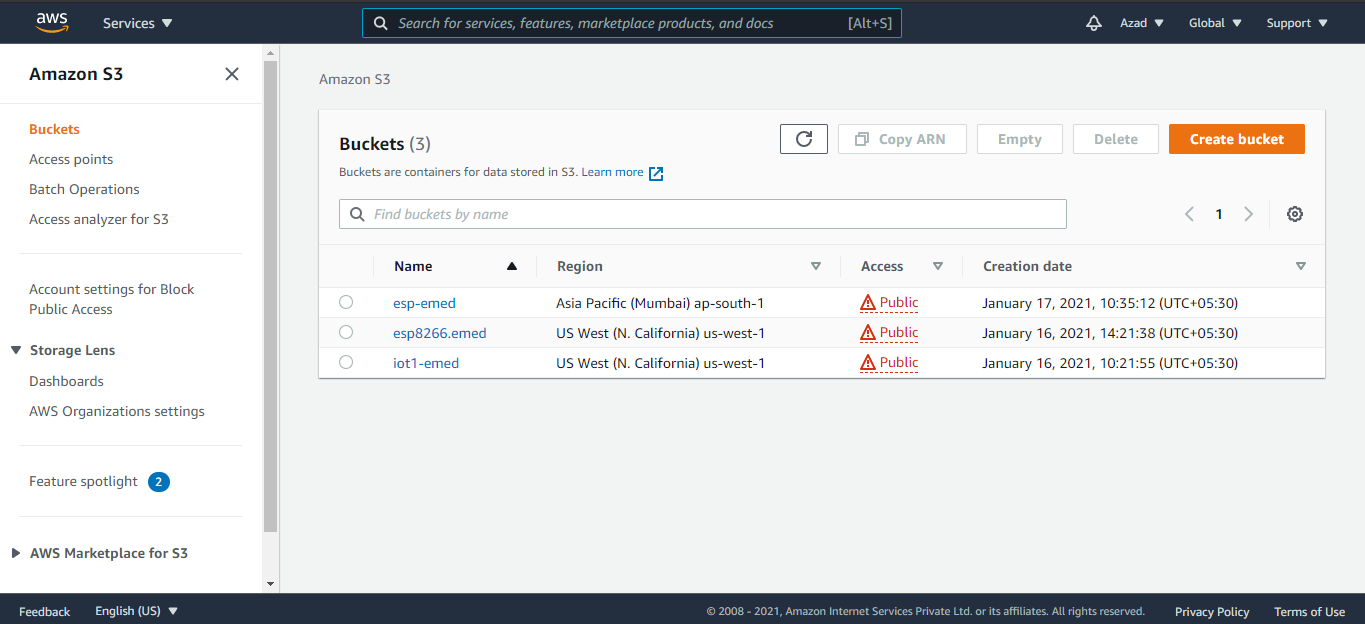
****

1. **Creating Bucket in AWS s3.**

**Go to services>s3>create bucket fil your bucket name and create.**

****





1. Upload file with Postman

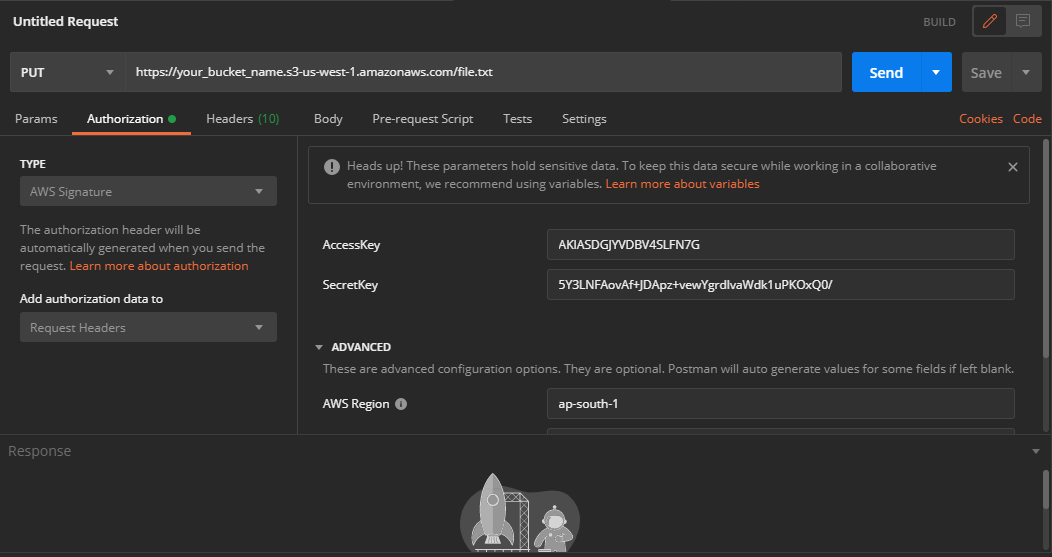
Open Postman, select request type “PUT” and type url - https://your\_bucket\_name.s3-us-west-1.amazonaws.com/file.txt

Go to type>AWS signature and type access key and secret key from csv file downloaded when IAM user was created.

In Body, type the content of file.

Using “Send” command send upload your file in Amazon s3 Bucket

For reading file change request type from PUT to GET. File content displayed in Body section.

****

**Reason for Rejection this platform – Cannot find any process or example code for reading and writing file in s3 Bucket with esp8266 Wi-Fi module.**

**Attempt 4: AWS IoT Core:**

**Go to >>**

[**https://docs.aws.amazon.com/iot/latest/developerguide/what-is-aws-iot.html**](https://docs.aws.amazon.com/iot/latest/developerguide/what-is-aws-iot.html)

**for setting up AWS IoT core account and creating Thing.**

**Library used: PubSubClient.h**

**Function used and conditions:**

1. **setCurrentTime(); for Internet time**
2. **pubSubCheckConnect(); for connecting to AWS IoT account and continuously subscribe topic. If any data available on this topic, print it immediately.**
3. **currState = digitalRead(switchStatus); check if switch condition is change or not. If condition is changed, then publish switch status to the topic.**

**Syntax>> pubSubClient.publish("outTopic", pubData);**

**where pubData is any kind of string data.**

1. **if (millis() - lastPublish > 10000) for publish data on topic in every 10 sec.**

**References:**

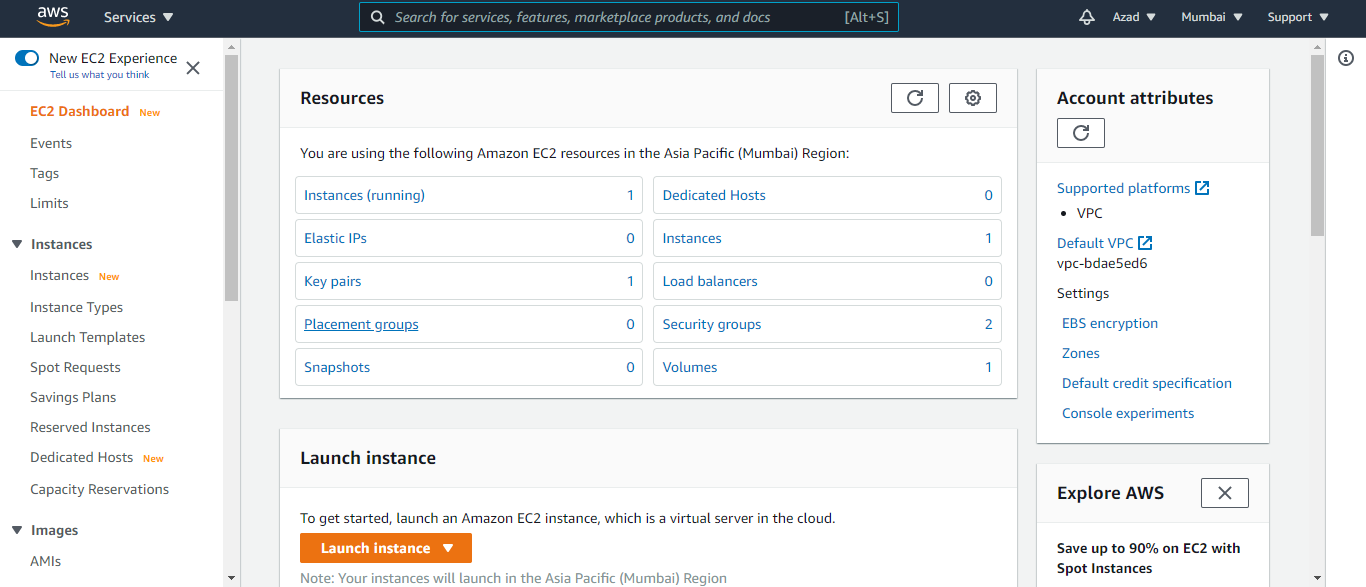
**1.** [**https://github.com/knolleary/pubsubclient**](https://github.com/knolleary/pubsubclient) **for library**

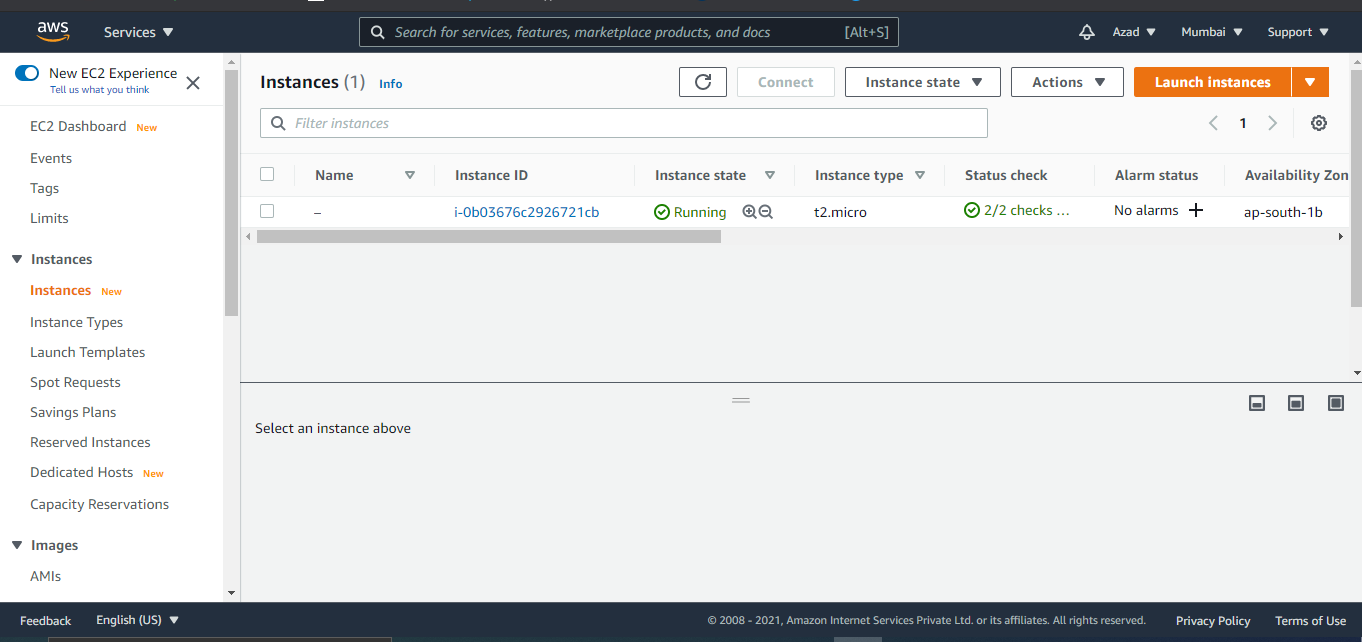
**2.** [**https://github.com/sborsay/AWS-IoT/blob/master/ESP8266-to-AWSIoT-Modified**](https://github.com/sborsay/AWS-IoT/blob/master/ESP8266-to-AWSIoT-Modified) **for example code.**

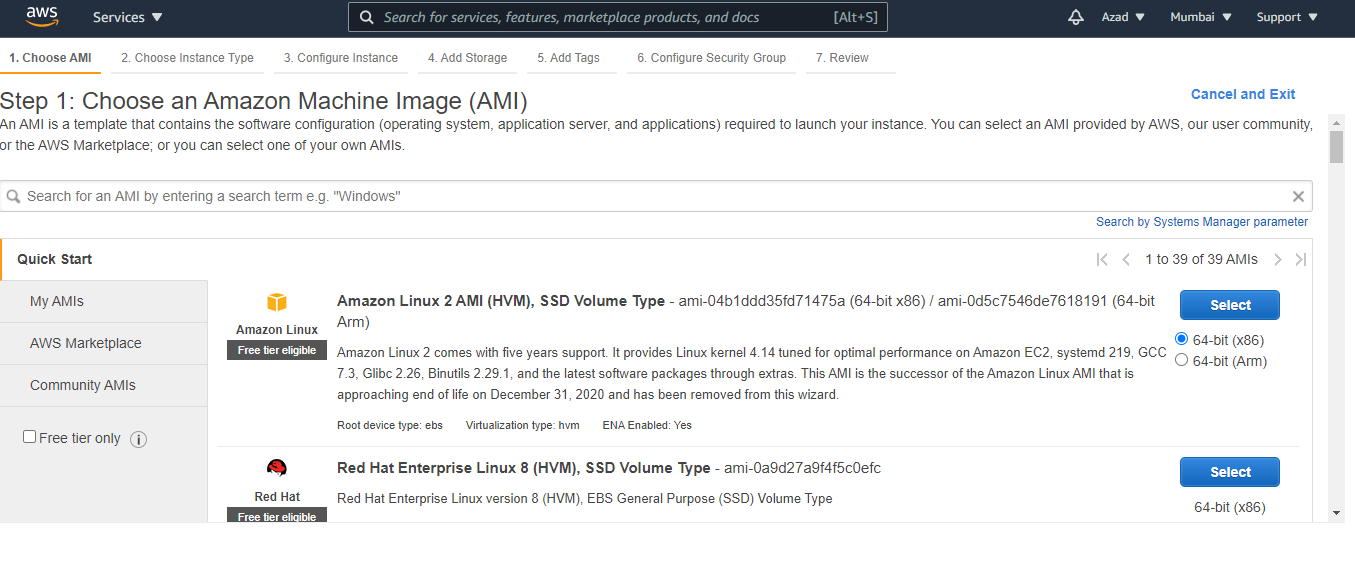
**Attempt 4: AWS EC2 INSTANCE –**

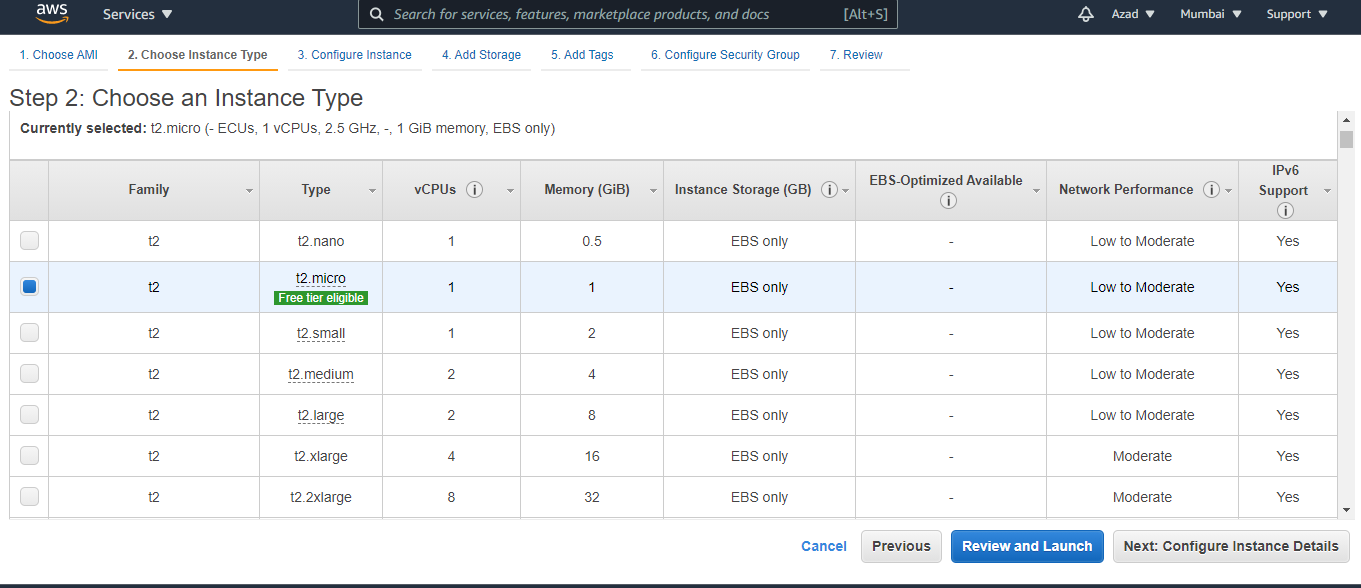
**Create INSTANCE – Go to services>INSTANCES>launch instance**

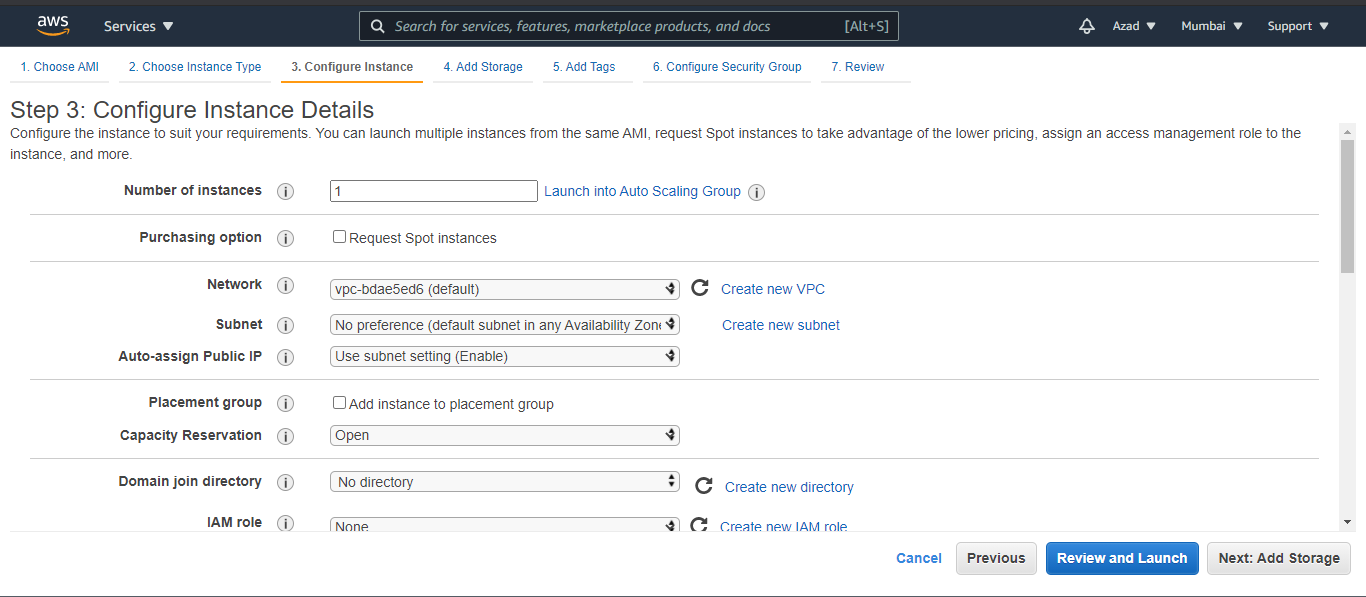
**and follow the following steps**

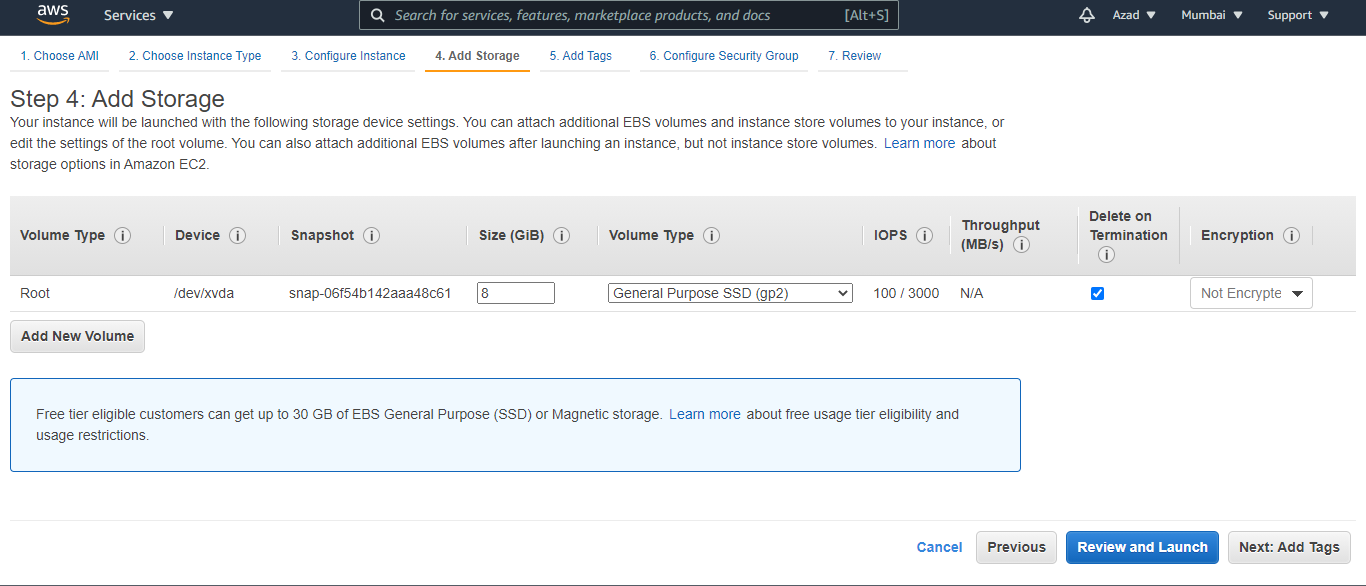
****

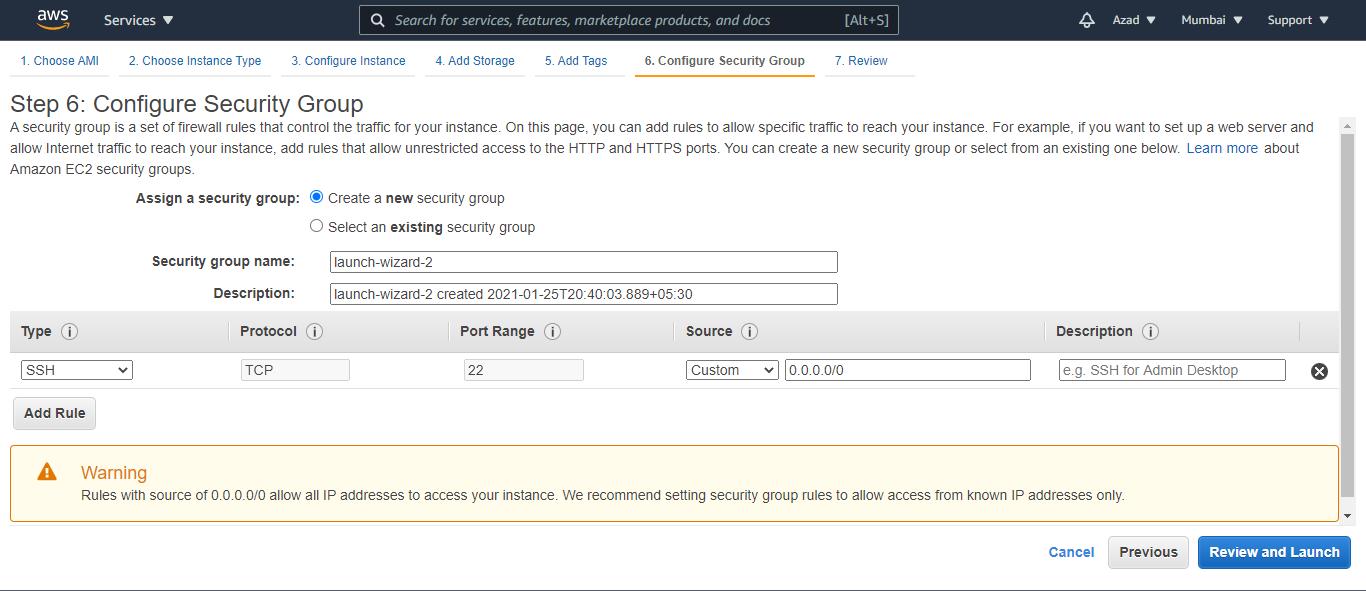
****

****



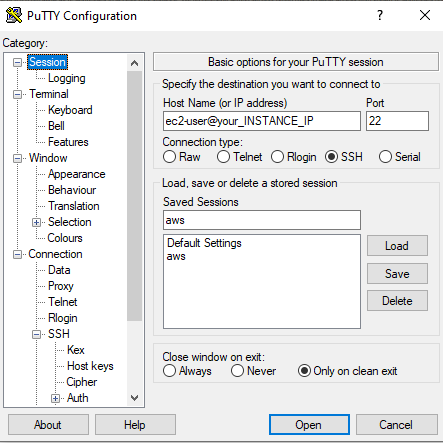


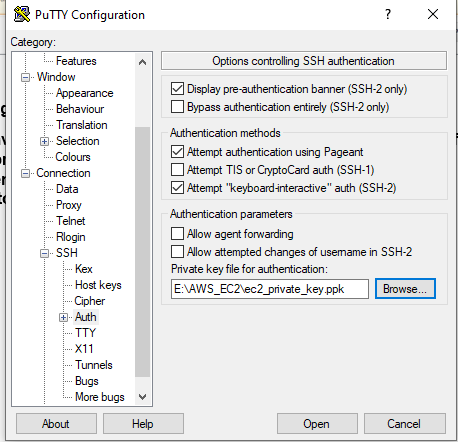




**Connecting INSTANCE to putty:**

1. **Convert ppm key to PPK – open puttygen>convert>select .pem file>save as private key**
2. **Open putty fill host id ie. ec2-user@public\_ip\_address**
3. **Go to ssh>Auth and select private.ppk putty key file.**

****

****

**For installing LAMP Server on INSTANCE follow the following link**

[**https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-lamp-amazon-linux-2.html**](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-lamp-amazon-linux-2.html)

**php script –**

**Posting Data of Device 1**

<?php

$Device1 = htmlspecialchars($\_GET["Value"]);

$myfile = fopen("Device1.txt", "w") or die("Unable to open file!");

fwrite($myfile,'{"feeds":[');

$data = array("Device1"=>$Device1);

$Jdata = json\_encode($data);

fwrite($myfile, $Jdata);

fclose($myfile);

?>

**Posting Data of Device 2**

<?php

$Device2 = htmlspecialchars($\_GET["Value"]);

$myfile = fopen("Device2.txt", "w") or die("Unable to open file!");

fwrite($myfile,',');

$data = array("Device2"=>$Device2);

$Jdata = json\_encode($data);

fwrite($myfile, $Jdata);

fclose($myfile);

?>

**Posting Data of Device 3**

<?php

$Device3 = htmlspecialchars($\_GET["Value"]);

$myfile = fopen("Device3.txt", "w") or die("Unable to open file!");

fwrite($myfile,',');

$data = array("Device3"=>$Device3);

$Jdata = json\_encode($data);

fwrite($myfile, $Jdata);

fwrite($myfile,'] }');

fclose($myfile);

?>

**Above three php scripts create three txt file which contains aal three device states. The following php read script is used to read all three txt files and display data in json format.**

<?php

$myfile1 = fopen("Device1.txt", "r") or die("Unable to open file!");

$myfile2 = fopen("Device2.txt", "r") or die("Unable to open file!");

$myfile3 = fopen("Device3.txt", "r") or die("Unable to open file!");

echo fgets($myfile1);

echo fgets($myfile2);

echo fgets($myfile3);

fclose($myfile1);

fclose($myfile2);

fclose($myfile3);

?>