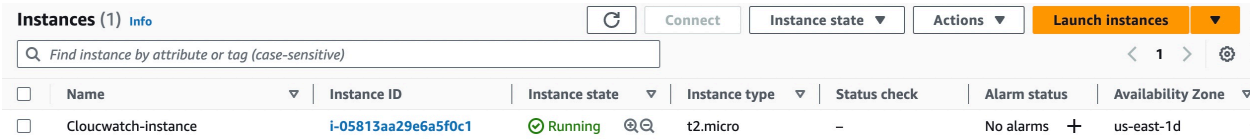
Cloudwatch Alarm Events, and Logging

Part 1

Launching an Instance

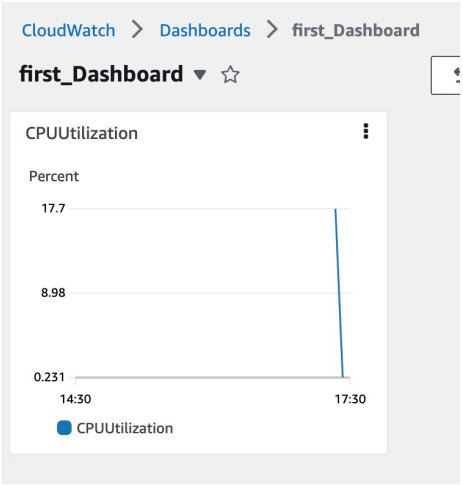
I created an EC2 instance named Cloudwatch-instance , AMI : Amazon Linux 2, t2.micro.
I checked “enable CloudWatch detailed monitoring’ under Advanced Details and pasted the code was given. I chose http and ssh open to anywhere under the security group section.



Part 2

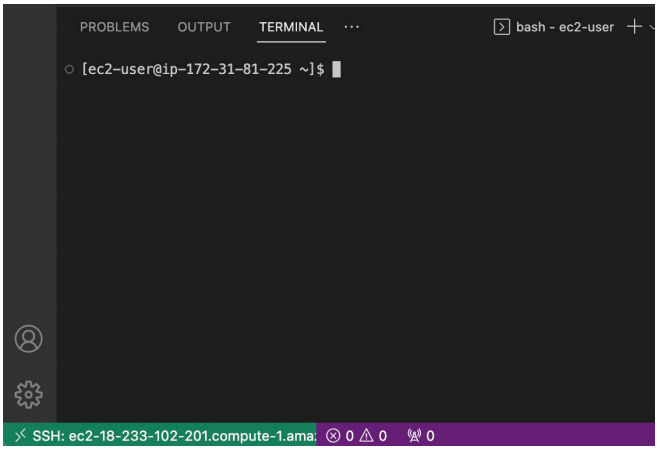
Creating a Cloudwatch dashboard

I selected Dashboard on the Cloudwatch and created one named ‘first_Dashboard’. Then I chose line, metrics, ec2 as a metrics. I selected ‘per-instance’ and CPUUtilization and then clicked ‘create widget’.



Upload Stress tool on EC2

I connected to my EC2 via ssh.

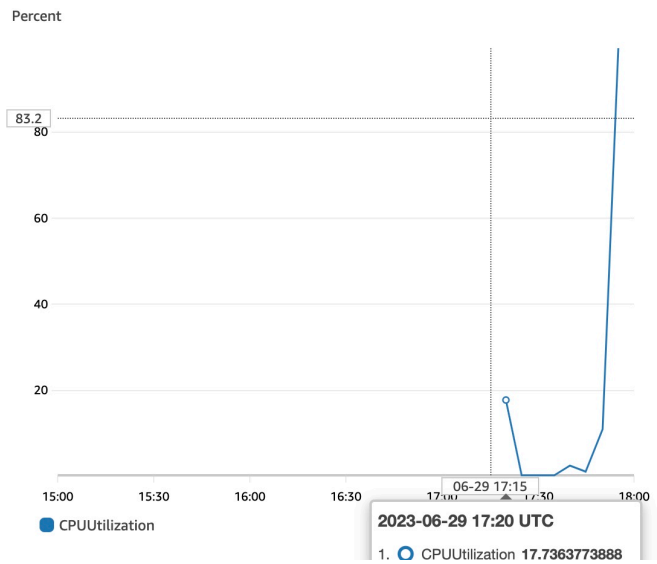


I uploaded the stress tool with the code ‘sudo yum install -y stress’.

```
Installed:
  stress-1.0.4-28.amzn2023.0.2.x86_64

Complete!
[ec2-user@ip-172-31-81-225 ~]$
: ec2-18-233-102-201.compute-1.amaz  0  0  0
```

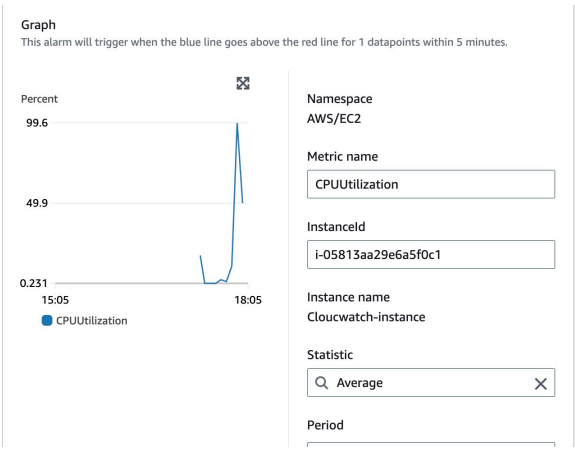
I want to force the CPU. In order to force I used the code given. “stress --cpu 80 --timeout 20000’



Part 3

Create an Alarm

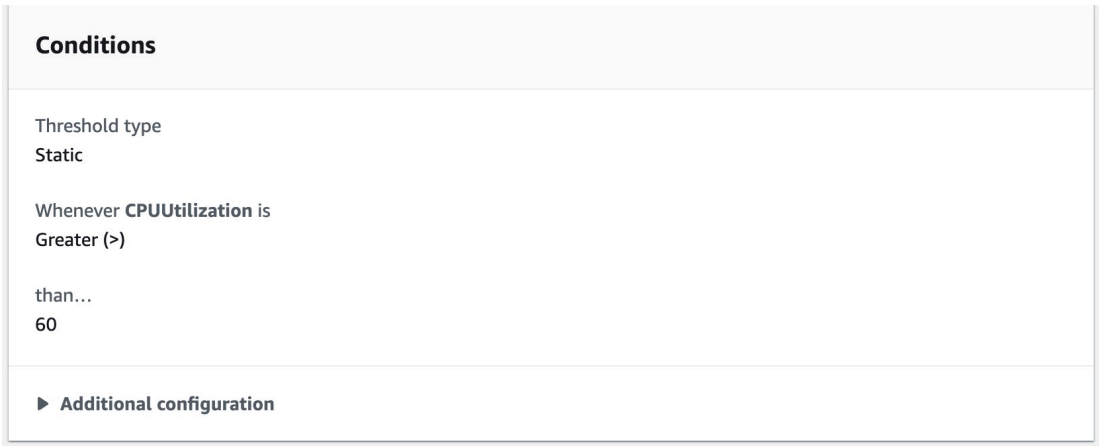
I selected ‘create alarm’ and ‘select metric’
I selected EC2, Per Instance Metrics, CPUUtilization and select metric.

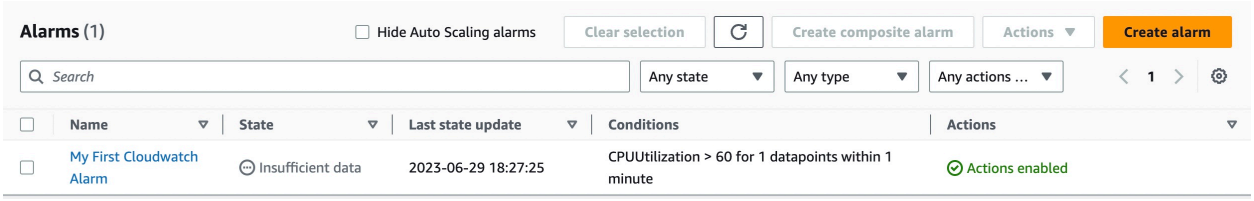


I chose period for 1 minute and kept remaining as default. Threshold Type is static, whenever CPUUtilization is greater, than 60.

Alarm state trigger is ‘In alarm’ , created a new topic as ‘my-alarm’, email endpoints I put my email address.

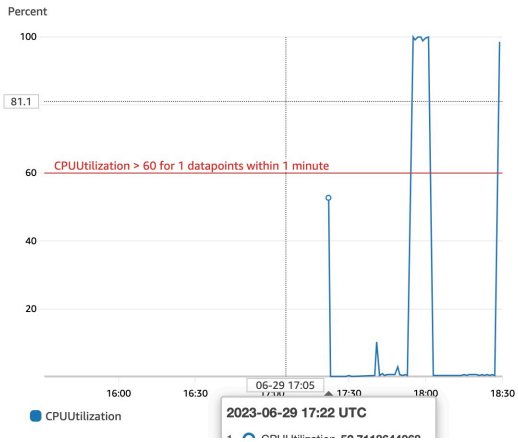
Under ec2 action; alarm state trigger ; in alarm ; select ‘ Stop this instance’ and next.
Alarm named as my first cloud watch alarm.

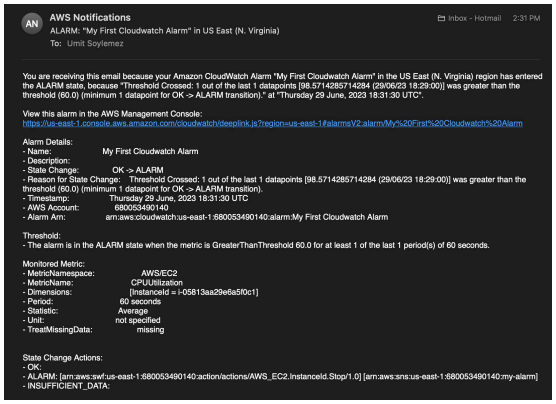


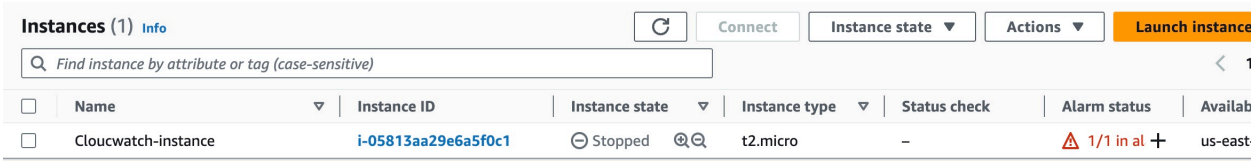


Now again the same code I run ‘stress --cpu 80 --timeout 20000’ on VS code.

I received an alarm message to my email and this message triggered to stop the ec2 instance.







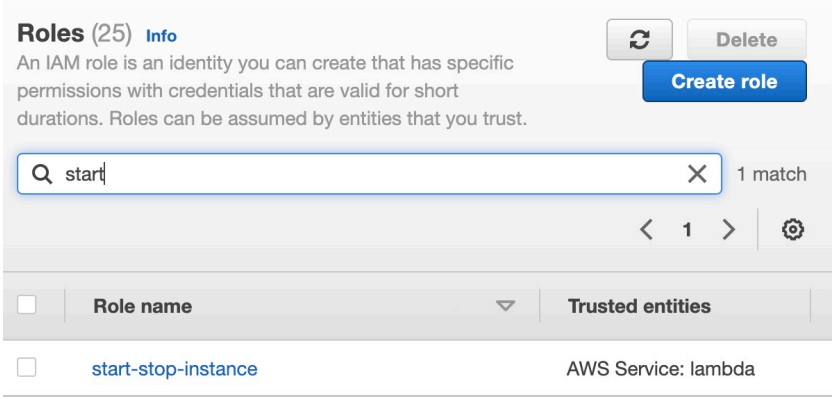
Part 4

CloudWatch Events with Lambda

Create Role

I created a Policy named ‘start-stop-instance’ and pasted the json script.

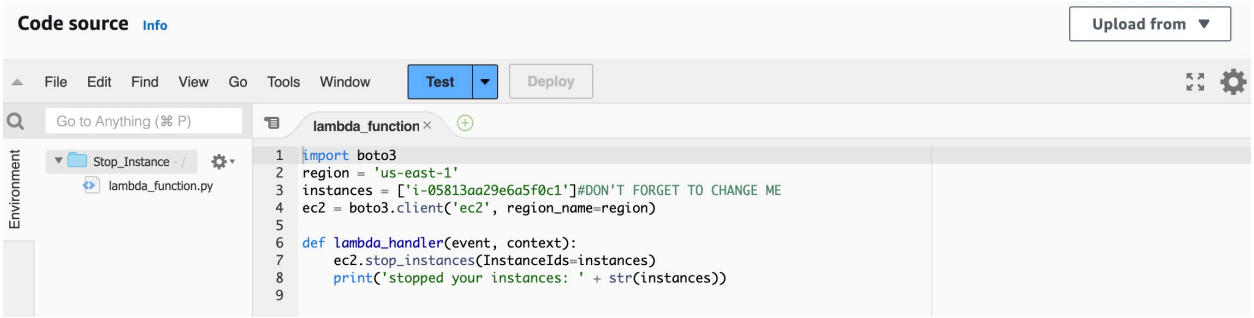
Than I created an IAM Role that will be used in Lambda Function. Role named start-stop-instance.



Creating Stop Lambda Functions

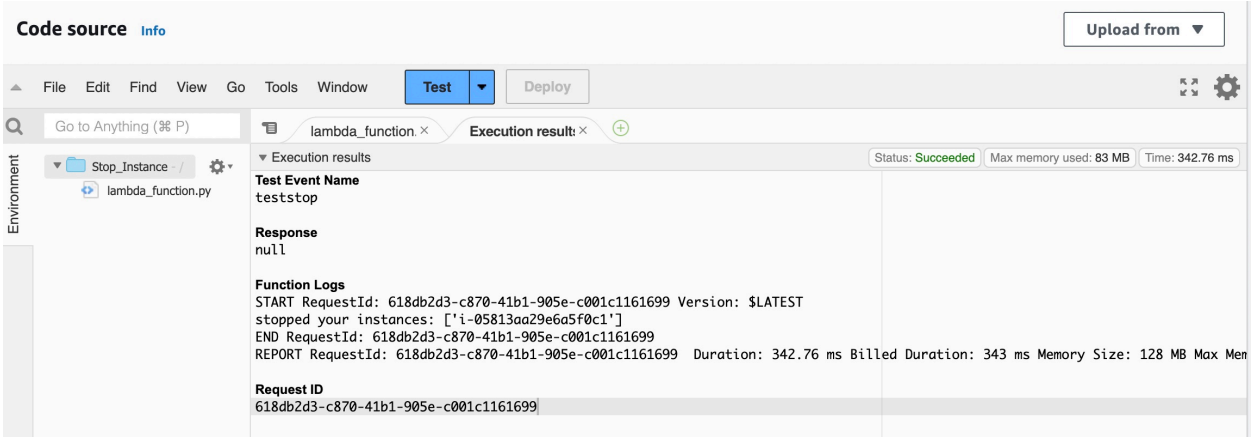
I create lambda function under functions. I selected author from scratch named Stop_Instance, runtime python 3.9.

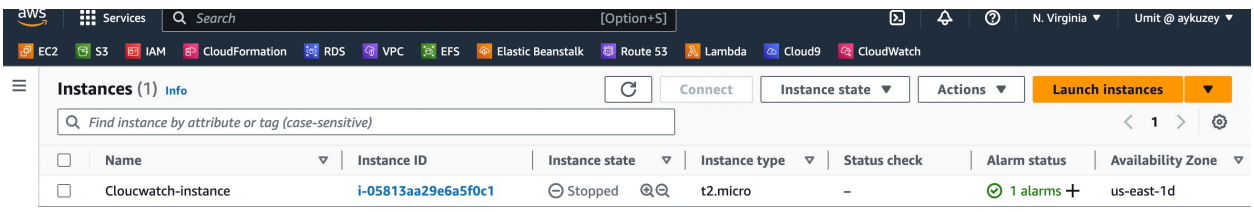
I chose existing role ‘start-stop-instance’ under Change Default execution Role and than hit the create function. I pasted the function code given but I changed the instance ID in the Code and than clicked DEPLOY



Testing the function- Create test event

I selected ‘Create new test event’. Event name is teststop, event template is ‘hello-world. Than I clicked test the instance had stopped.

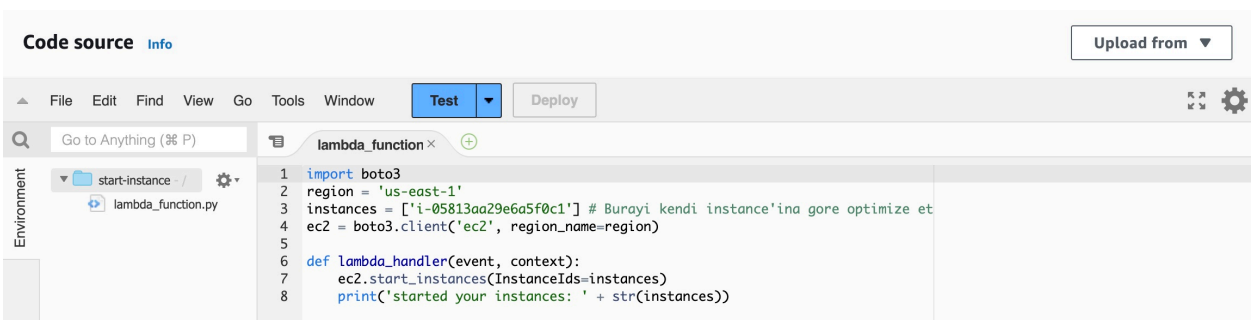




Creating Start Lambda Functions

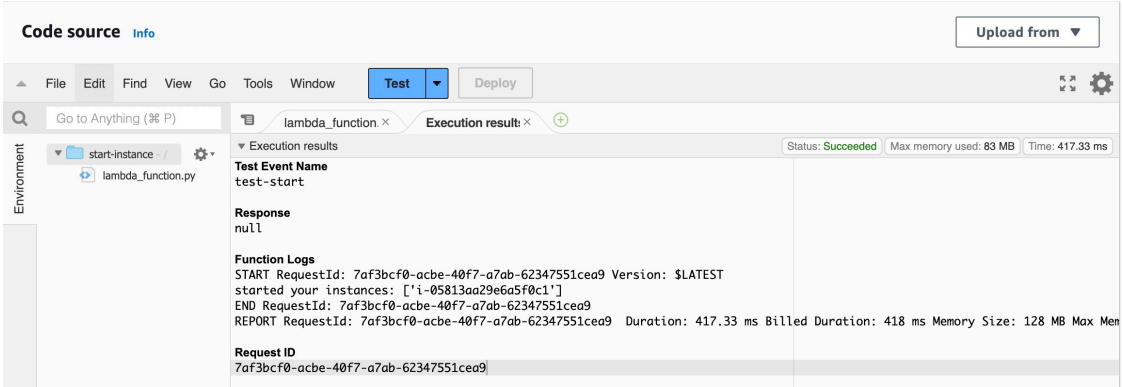
I create lambda function under functions. I selected author from scratch named Start_Instance, runtime python 3.9.

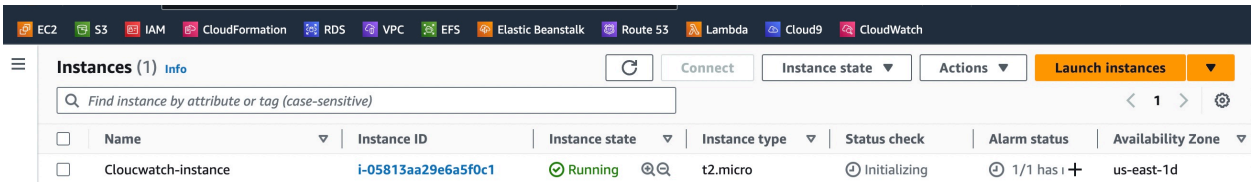
I chose existing role ‘start-stop-instance’ under Change Default execution Role and than hit the create function. I pasted the function code given but I changed the instance ID in the Code and than clicked DEPLOY



Testing the function- Create test event

I selected ‘Create new test event’. Event name is teststart, event template is ‘hello-worl. Than I clicked test the instance had started.



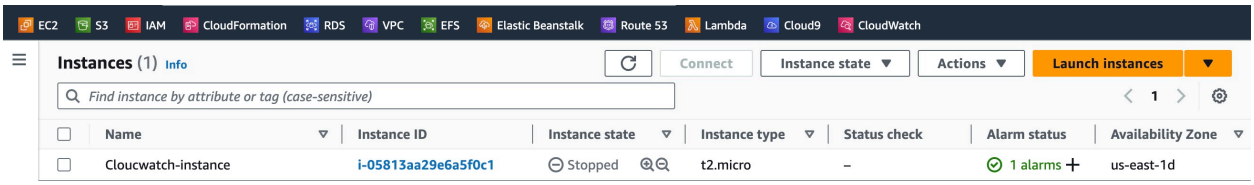


Creating Stop-Cloudwatch Event

I went to the CloudWatch Console and under the Event sub-section
Rules and Create Rules.

I selected ‘Schedule’ and 1 mins for cron expression under event source. Targets; Function: Stop_Instance.

I named Event_Stop under the configure details. Description: This event provides stop action state I chose enabled.



Creating Terminate- Cloudwatch Event

I went to the CloudWatch Console and under the Event sub-section
Rules and Create Rules.

I selected ‘Schedule’ and 1 mins for cron expression under event source. Targets; Function: Start_Terminate.

I named Event_Terminate under the configure details. Description: This event provides start action state I chose enabled.