Northeastern University

CS 6620 Cloud Computing

Project #1 [200 points]

EC2 Instances Creation and Operations

Guidelines

Project #1 should be electronically submitted to Canvas by midnight on the due date. A submission link is provided on Canvas.

Assignment Overview

For this project, you will get familiar with the cretaion of multiple EC2 instances on AWS and conducting different operations on them, via programming as well as using the AWS cosnole. Part I requires programming. Part II is basic preparation via the cosnole, for the next project assignment (i. e., Project 2).

Please post any questions to Piazza and attend TA office hours to address any questions. Get started early.

PART I: Basic EC2 Operations via Programming

1) Using an appropriate method you need to set up an EC2 instance
2) Modify the EC2 instance initiated from #1 above by as signing a tag to the instance
3) Further modify Q1 to reboot the EC2 instance that you had set up at the beginning
4) When the entire process of set up has been completed then stop the EC2 instance.
5) Start 5 to 6 EC2 instances and assign tags randomly ensuring there are 2 different tags in total
6) Find all the instance id that are actively running and report the same
7) Filter the EC2 instance based on tags
8) Describe the properties of all the EC2 instances
9) Update the tag to make sure that there is only one tag that is assigned to all the ec2 instance
10) List all the instance id and their tags
11) Stop all the instances using the instance id's

PART II: Study the concepts for the next assignment (needs to do be done in console)

1) Set up the EC2 instance, using t2.micro, add tag to it and it has to be accessed via any ip address
2) Explain in brief the process by setting up a routing table
3) Set up an Application Load Balancer in the console
4) Create a VPC
5) Write a 'hello world program' in Python using AWS

Explain using screenshots, code and text as needed to fully answer these questions.

Other notes:

Please see TA announcement regarding project submitting guidelines. A PDF report showing the results via screenshots and brief explanations are required. In addition, submit the code file as appropriate.

Evaluation

Your work will be evaluated on how well the code, results including screenshots and explanations (text) are in conformance to the requirements above.

Executive Summary

Part of your completed assignment submission should be an executive summary containing an "Assignment overview" (1 paragraph, up to about 250 words) explaining what you understand to be the purpose and scope of the assignment and a "technical impression" (1–2 paragraphs, about 200–500 words) describing your experiences while carrying out the assignment. The assignment overview shows how well you understand the assignment; the technical impression section helps to determine what parts of the assignment need clarification, improvement, etc., for the future.

Evaluation

The grade for your executive summary is based on the effort you put into the assignment overview and technical impression. In general, if you put some effort into your writing, you will receive full credit for your executive summary (provided that it is properly formatted and submitted as a plain text file).

Project Deliverables

The following items should be archived together, e.g., placed in a `.zip` file or tarball file (*.tgz or *.tar.gz), and electronically submitted via the link is provided on the course Moodle page.

1. All source code files plus any additional support code you developed to answer Part 1.
2. Your executive summary and the PDF report answering all questions.

References

https://aws.amazon.com/ec2/getting-started/
https://www.youtube.com/watch?v=bm-6D7iAVS8

ANS (Qichen An)

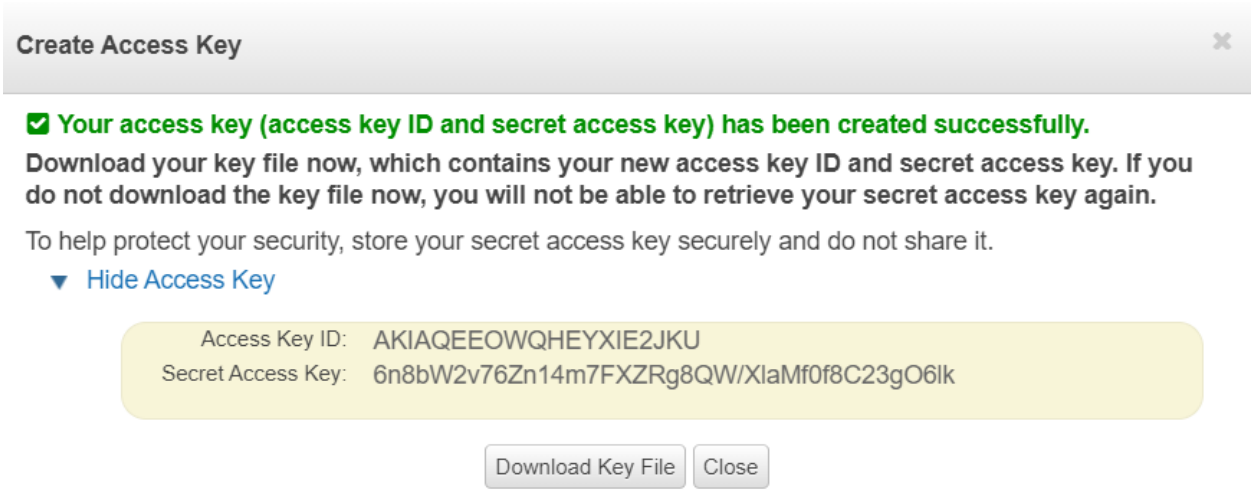
Part 1

1. Using an appropriate method you need to set up an EC2 instance

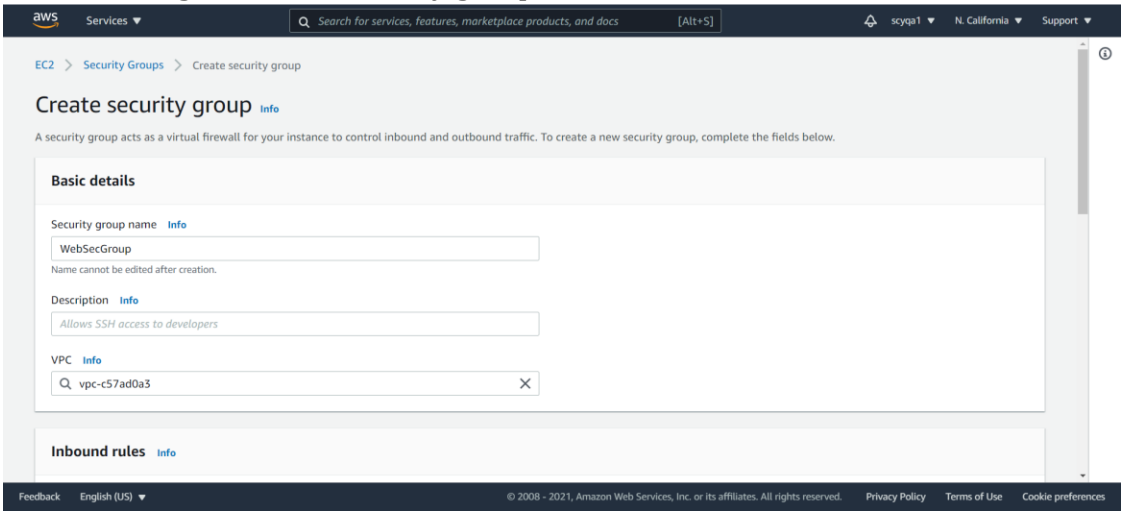
Configure the environment (boto3)

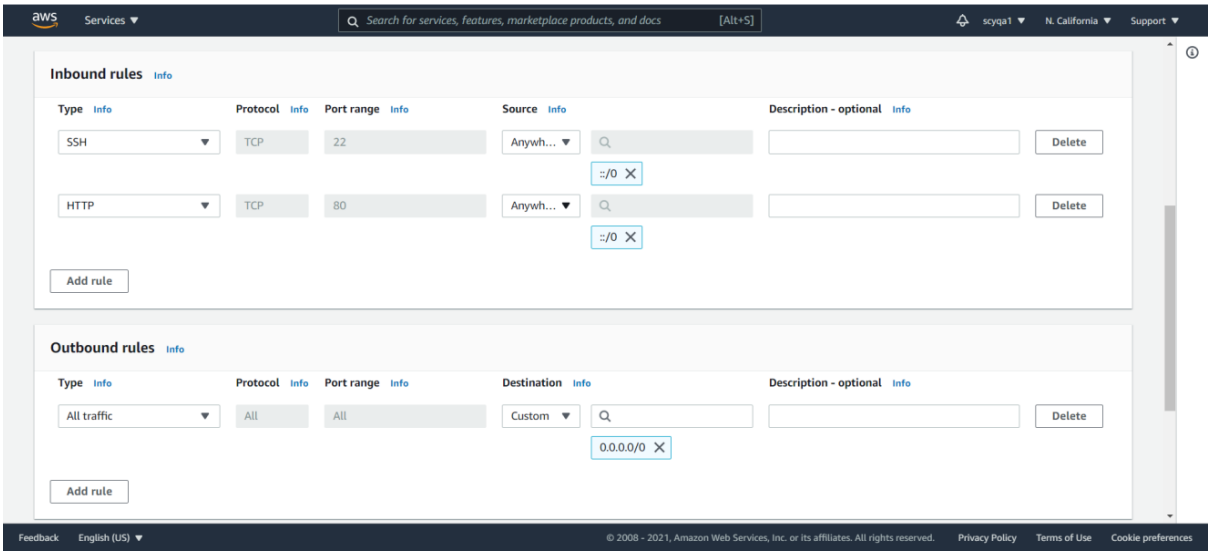
```
(base) D:\Year21-22\CS 6650 Fundamentals of Cloud Computing\CW\project-individual1>pip install boto3
Collecting boto3
  Downloading boto3-1.18.50-py3-none-any.whl (131 kB)
     |████████████████████████████████| 131 kB 1.3 MB/s
Collecting botocore<1.22.0,>=1.21.50
  Downloading botocore-1.21.50-py3-none-any.whl (8.0 MB)
     |████████████████████████████████| 8.0 MB 1.6 MB/s
Collecting jmespath<1.0.0,>=0.7.1
```

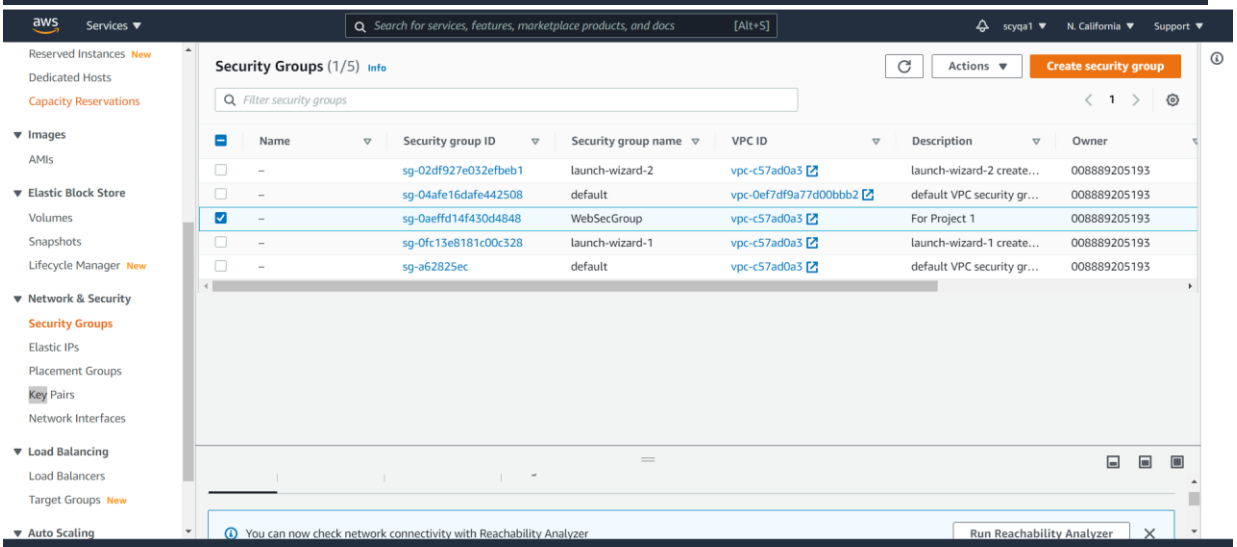
Create Access Key

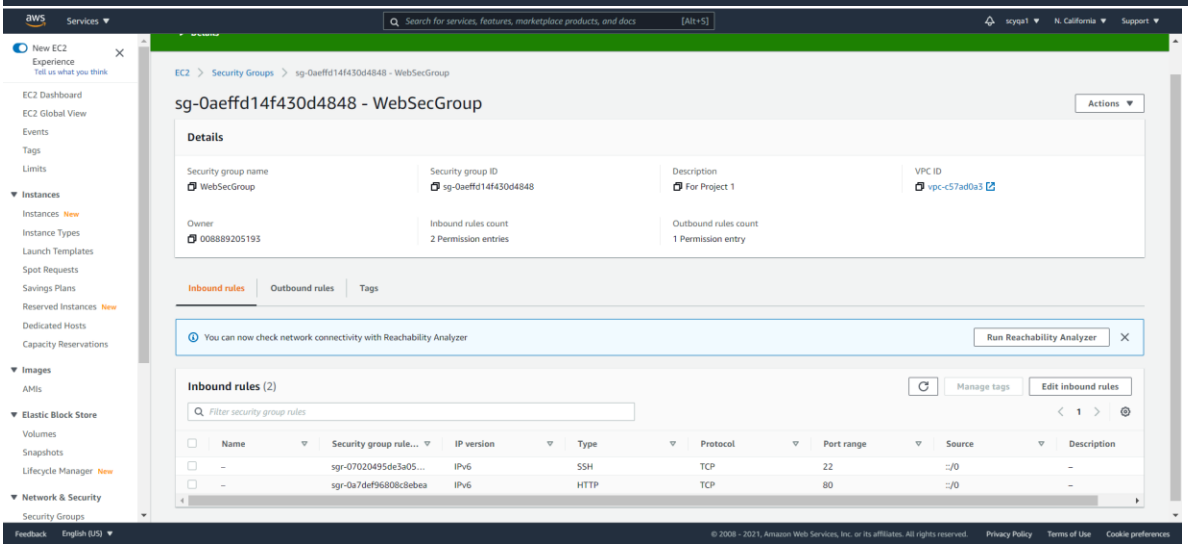


Before coding, I create a security group in case it is needed for instance creation

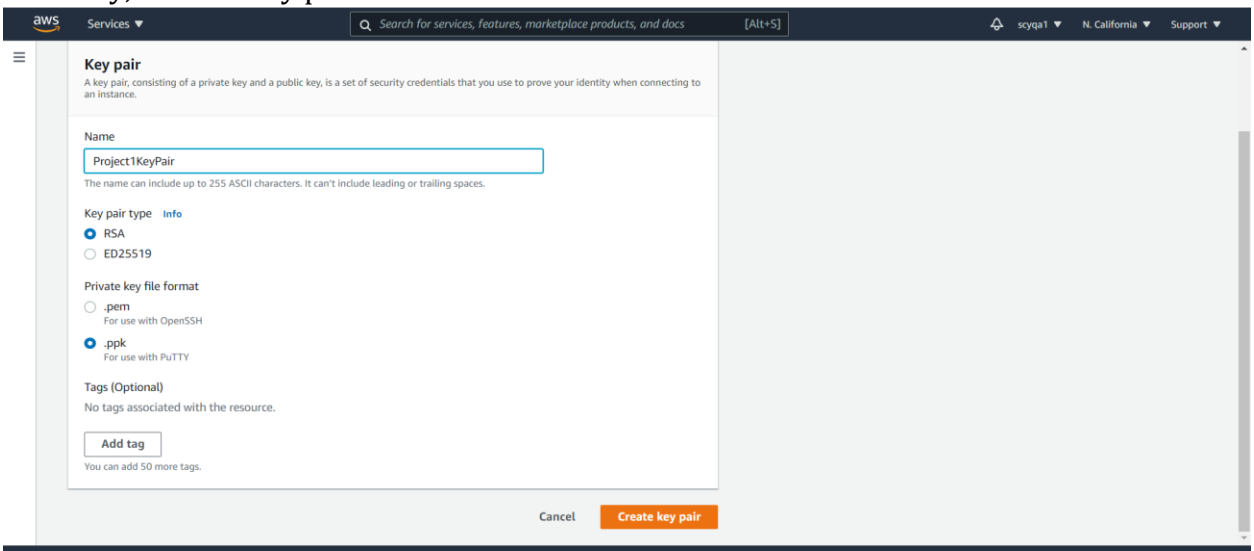


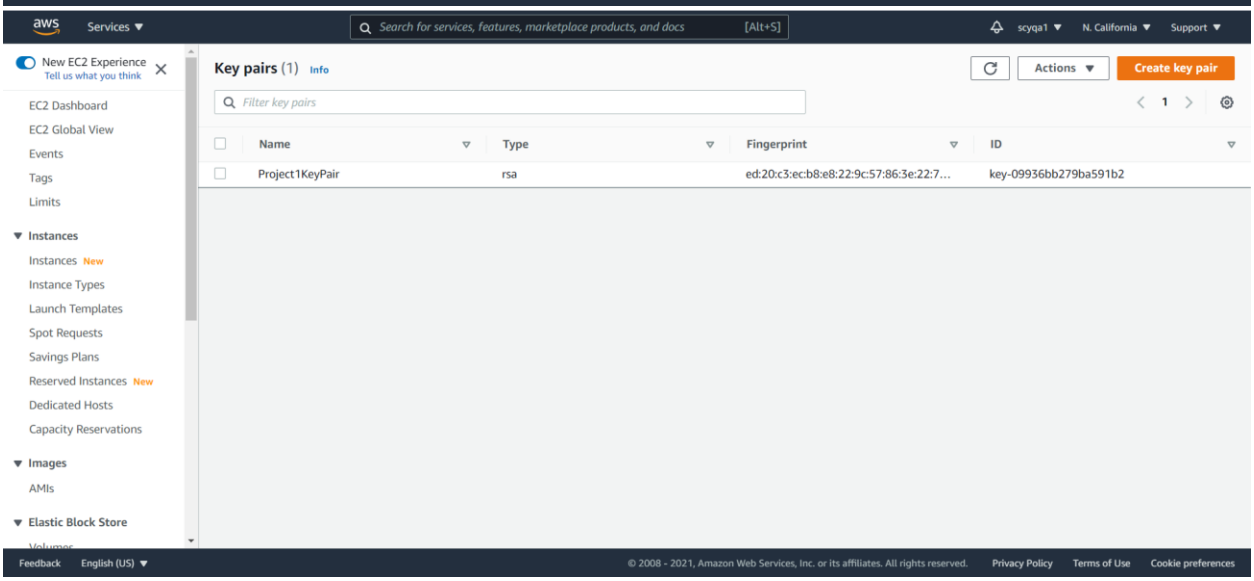

Services
Search for services, features, marketplace products, and docs [Alt+S]
scyqa1
N. California
Support
Inbound rules Info
Type Info
Protocol Info
Port range Info
Source Info
Description - optional Info
SSH
TCP
22
Anywh...
::/0
Delete
HTTP
TCP
80
Anywh...
::/0
Delete
Add rule
Outbound rules Info
Type Info
Protocol Info
Port range Info
Destination Info
Description - optional Info
All traffic
All
All
Custom
0.0.0.0/0
Delete
Add rule
Feedback
English (US)
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Cookie preferences


Services
Search for services, features, marketplace products, and docs [Alt+S]
scyqa1
N. California
Support
Reserved Instances New
Dedicated Hosts
Capacity Reservations
Images
AMIs
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager New
Network & Security
Security Groups
Elastic IPs
Placement Groups
Key Pairs
Network Interfaces
Load Balancing
Load Balancers
Target Groups New
Auto Scaling
Security Groups (1/5) Info
Actions
Create security group
Filter security groups
Name
Security group ID
Security group name
VPC ID
Description
Owner
sg-02df927e032efbeb1 launch-wizard-2 vpc-c57ad0a3 launch-wizard-2 create... 008889205193
sg-04afe16dafe442508 default vpc-0ef7df9a77d00bbb2 default VPC security gr... 008889205193
sg-0aeffd14f430d4848 WebSecGroup vpc-c57ad0a3 For Project 1 008889205193
sg-0fc13e8181c00c328 launch-wizard-1 vpc-c57ad0a3 launch-wizard-1 create... 008889205193
sg-a62825ec default vpc-c57ad0a3 default VPC security gr... 008889205193
You can now check network connectivity with Reachability Analyzer
Run Reachability Analyzer
Feedback
English (US)
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Cookie preferences

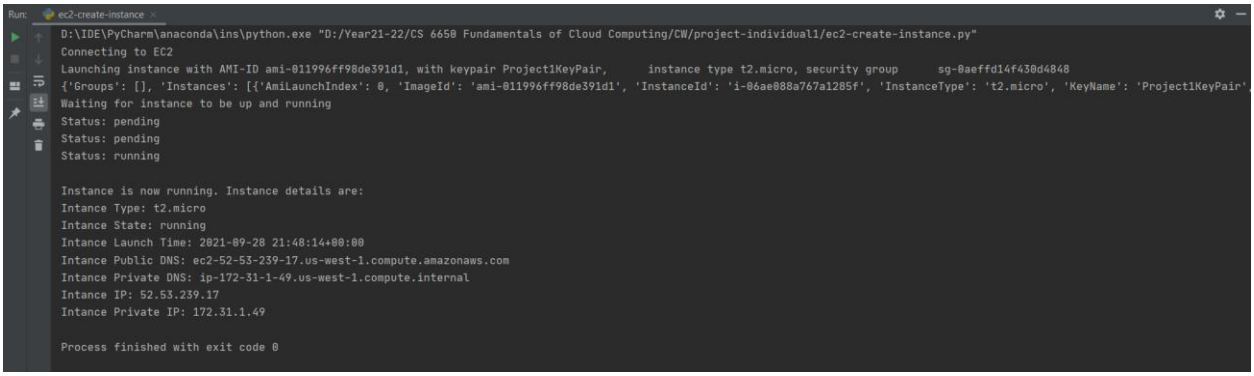

Services
Search for services, features, marketplace products, and docs [Alt+S]
scyqa1
N. California
Support
New EC2 Experience
Tell us what you think
EC2 Dashboard
EC2 Global View
Events
Tags
Limits
Instances
Instances New
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances New
Dedicated Hosts
Capacity Reservations
Images
AMIs
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager New
Network & Security
Security Groups
EC2 > Security Groups > sg-0aeffd14f430d4848 - WebSecGroup
sg-0aeffd14f430d4848 - WebSecGroup
Actions
Details
Security group name
WebSecGroup
Security group ID
sg-0aeffd14f430d4848
Description
For Project 1
VPC ID
vpc-c57ad0a3
Owner
008889205193
Inbound rules count
2 Permission entries
Outbound rules count
1 Permission entry
Inbound rules
Outbound rules
Tags
You can now check network connectivity with Reachability Analyzer
Run Reachability Analyzer
Inbound rules (2)
Manage tags
Edit inbound rules
Filter security group rules
Name
Security group rule...
IP version
Type
Protocol
Port range
Source
Description
sgr-07020495de3a05...
IPv6
SSH
TCP
22
::/0
sgr-0a7def96808c8ebea
IPv6
HTTP
TCP
80
::/0
Feedback
English (US)
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Cookie preferences

Similarly, I create key pair

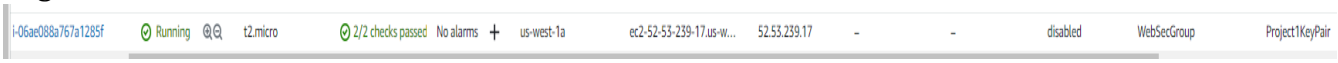




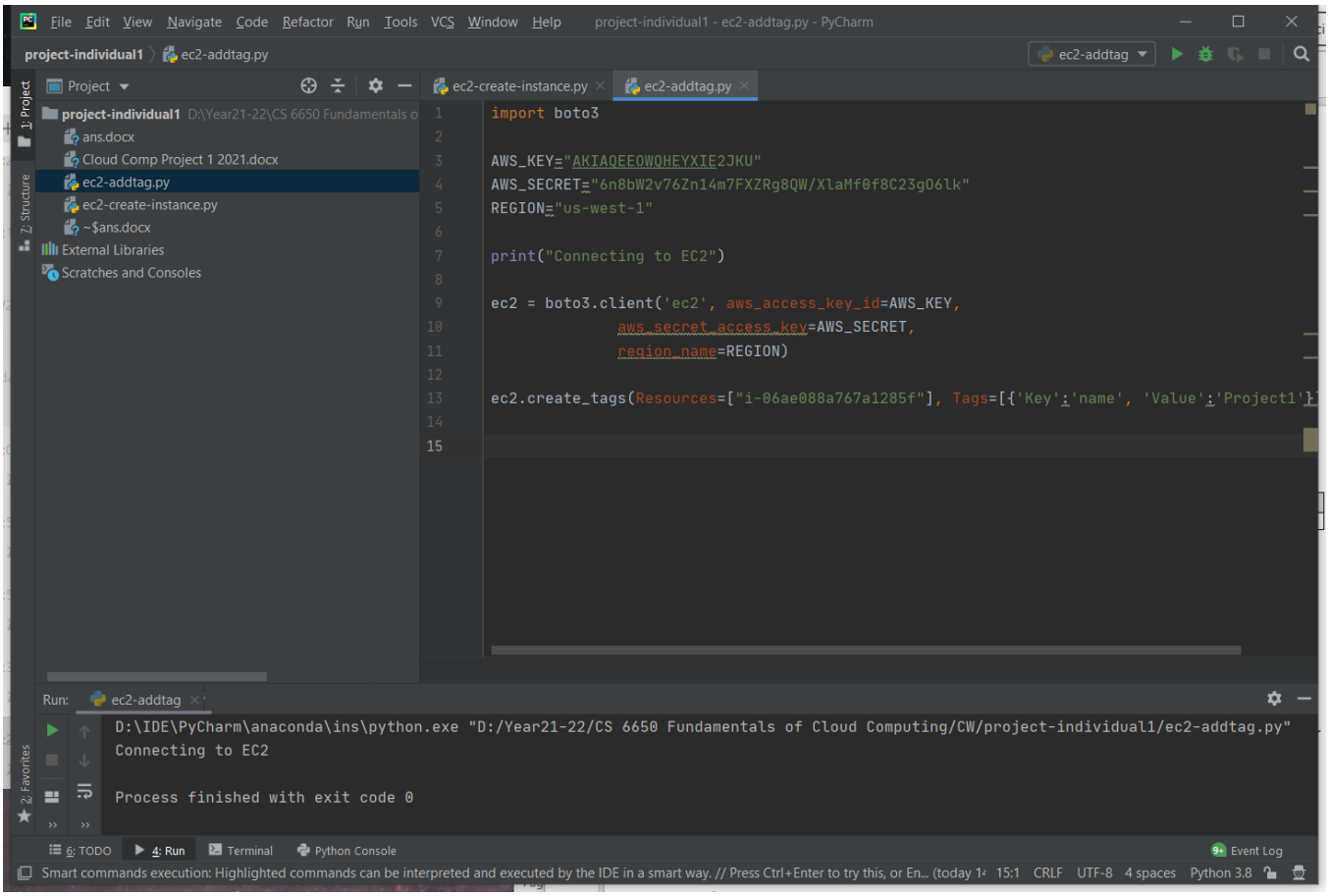
Final Result



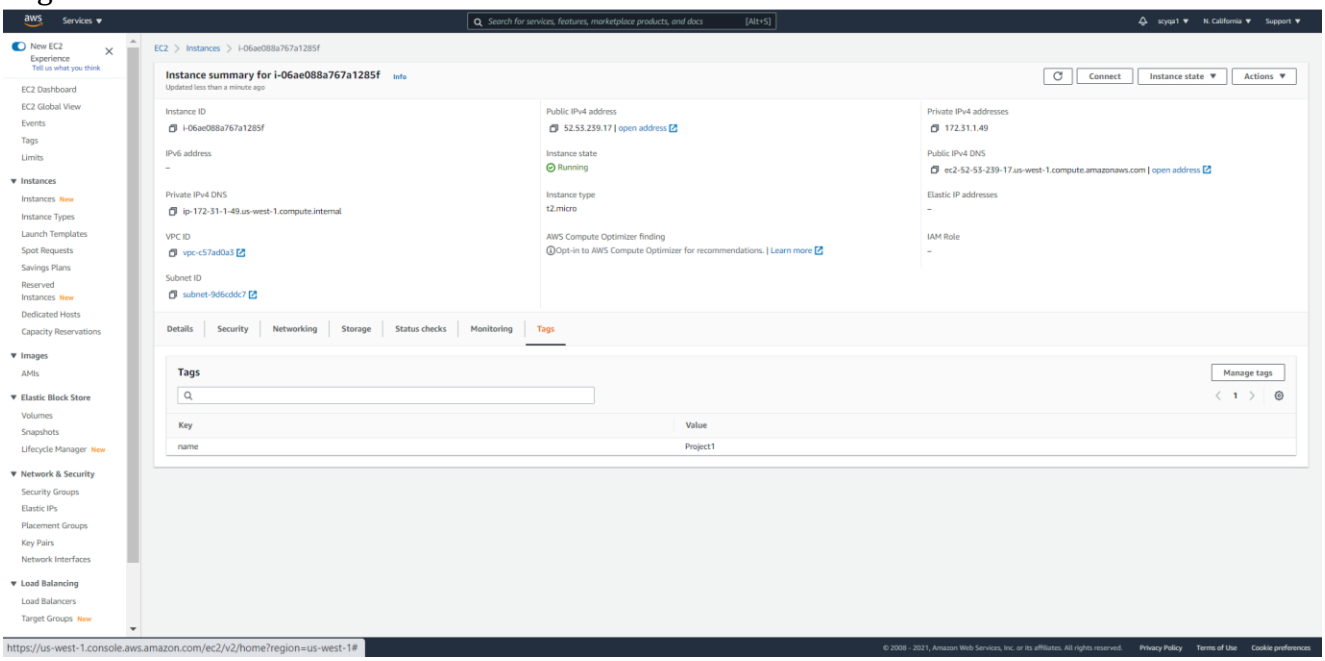
Log on the aws



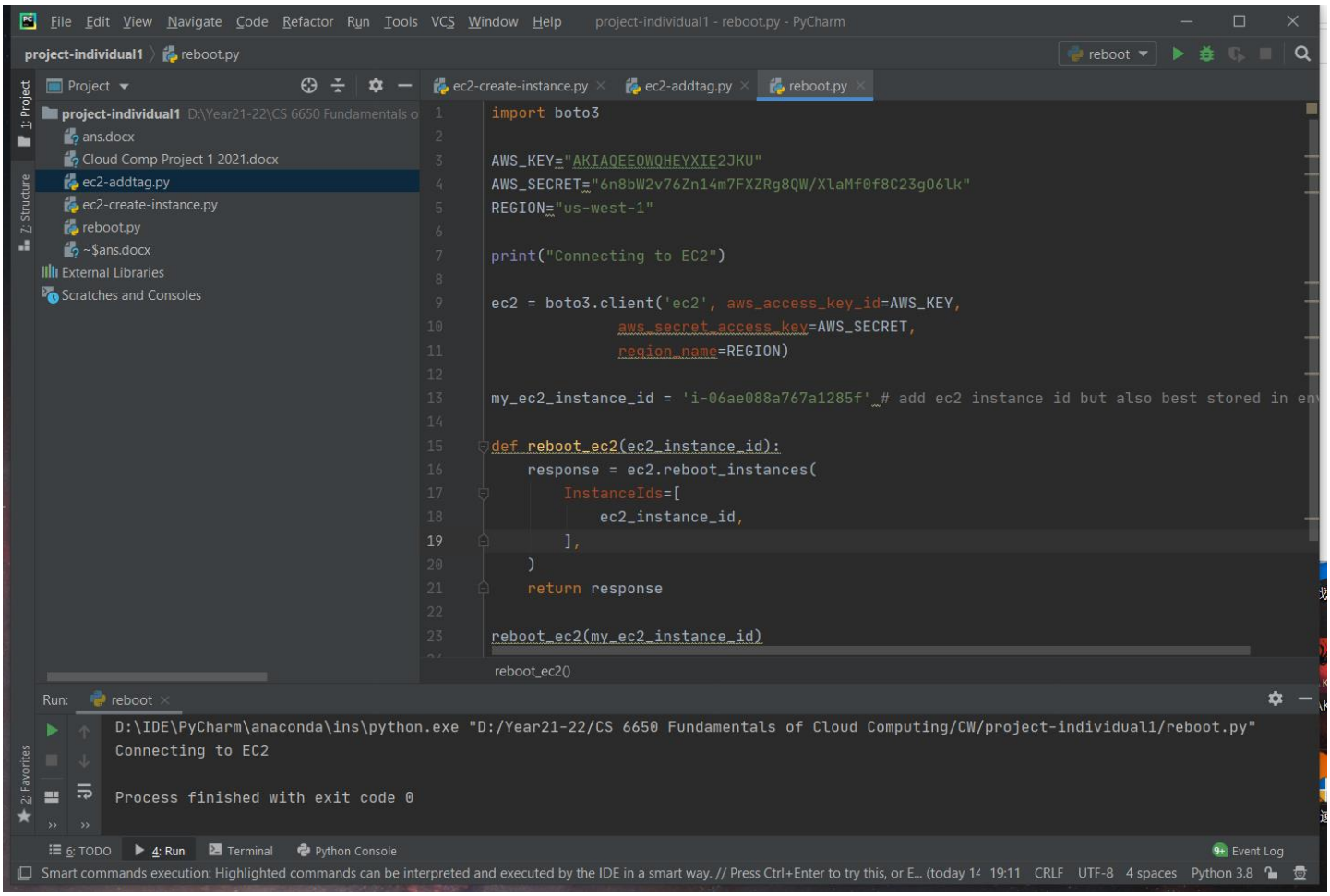
2. Modify the EC2 instance initiated from #1 above by assigning a tag to the instance

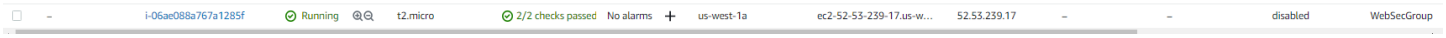


Log on the aws

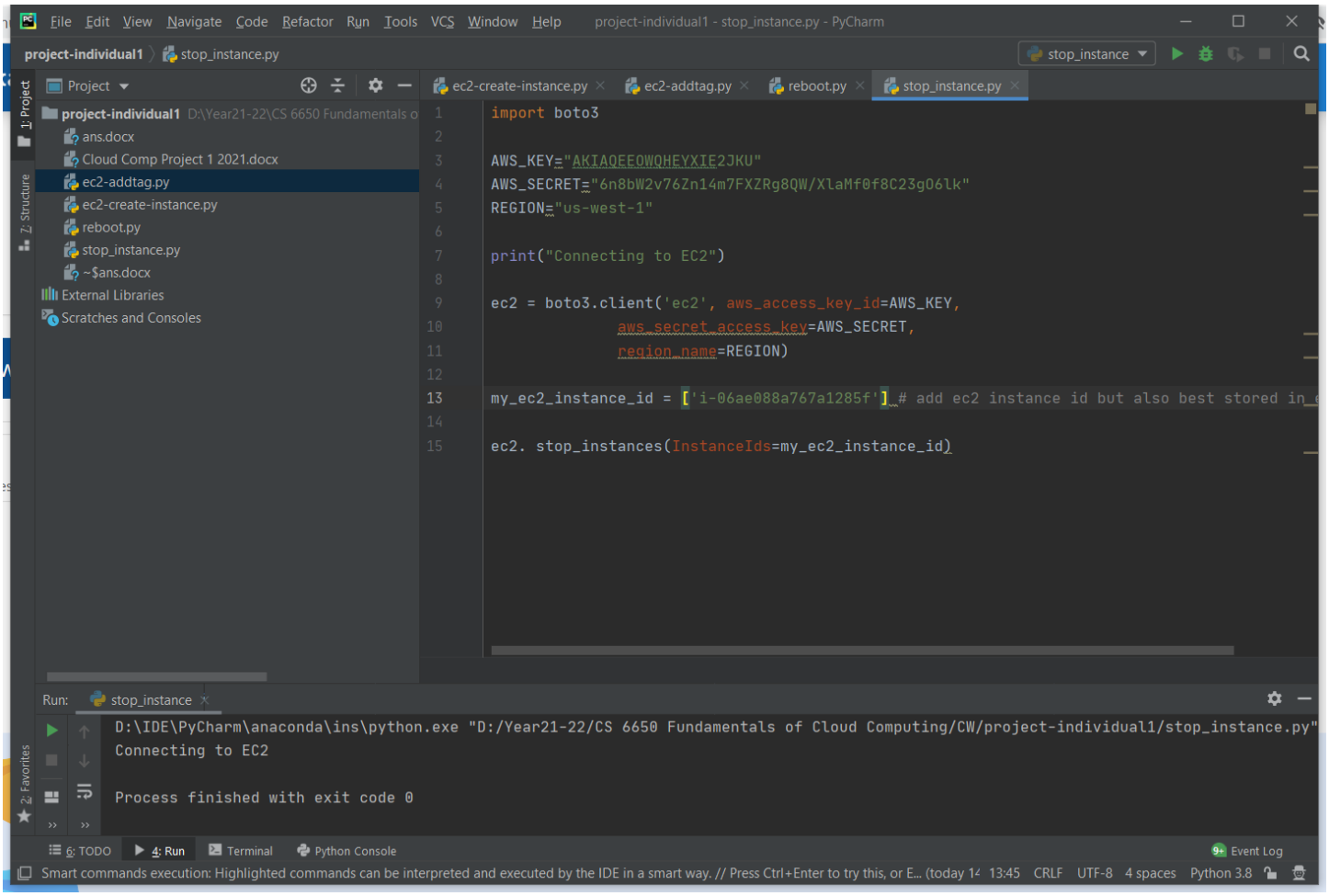


3. Further modify Q1 to reboot the EC2 instance that you had set up at the beginning



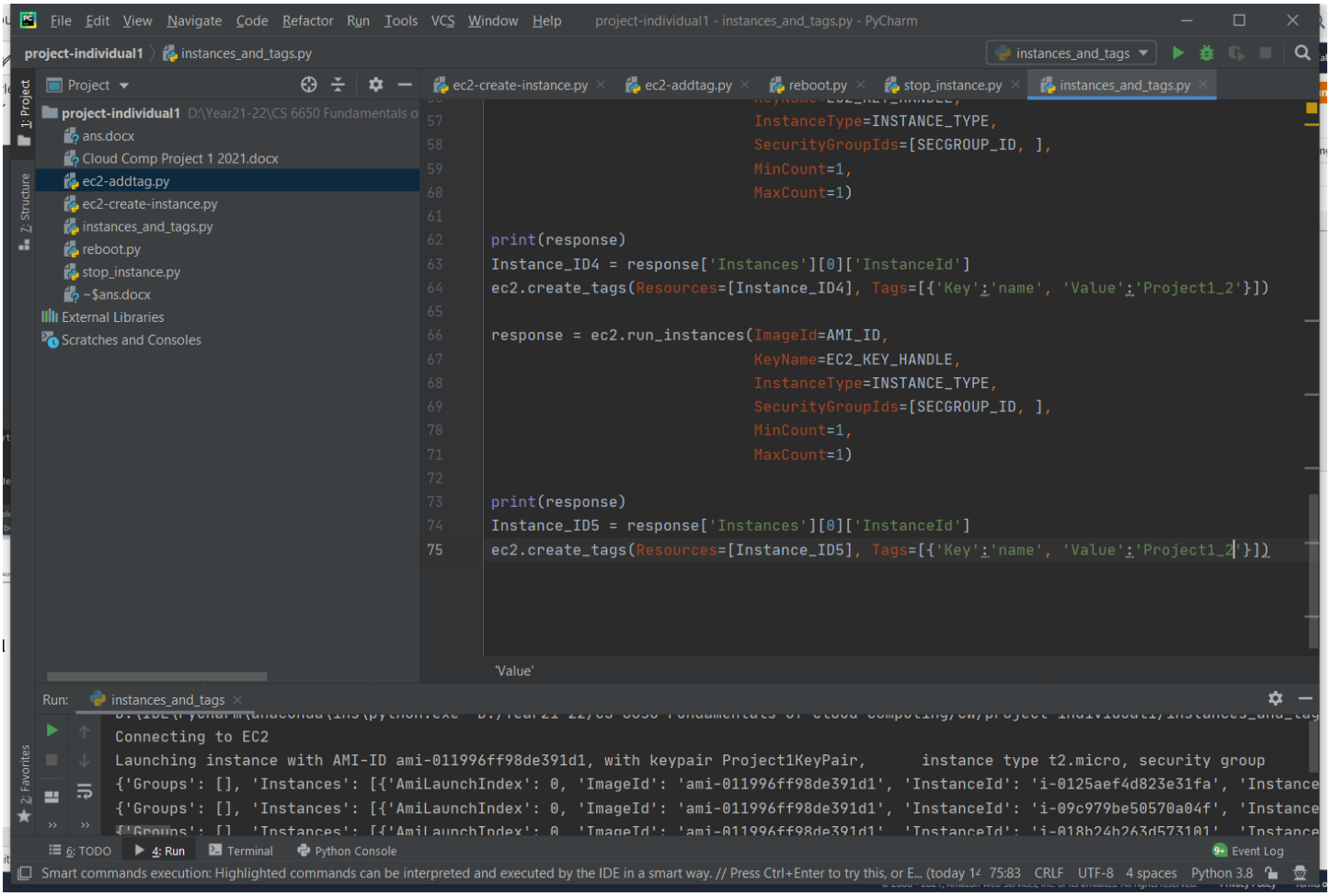


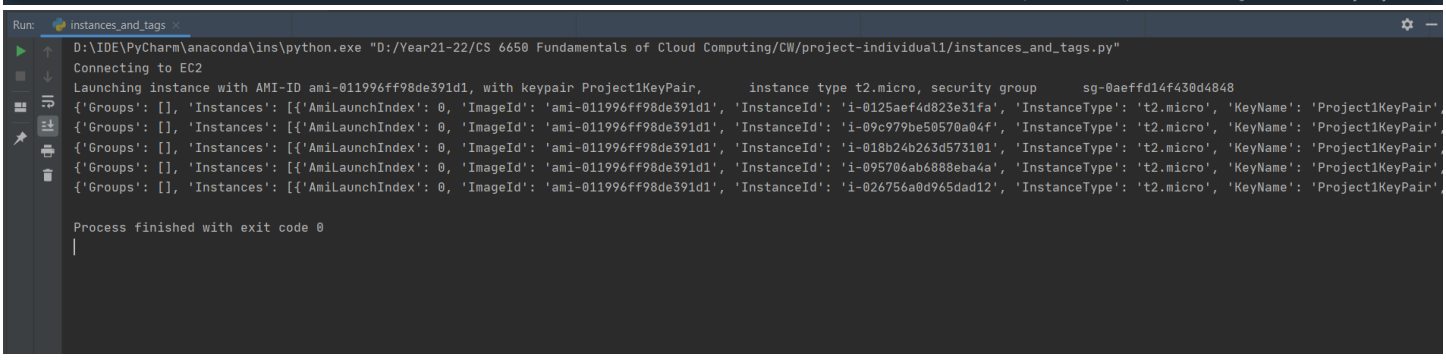
4. When the entire process of set up has been completed then stop the EC2 instance.



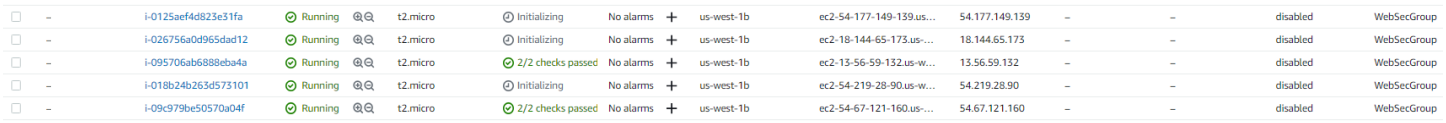
5. Start 5 to 6 EC2 instances and assign tags randomly ensuring there are 2 different tags in total

For this part, I launch 5 EC2 instaces, and the first three are with Tag ‘Project1_1’ and the last two are with Tag ‘Project1_2’

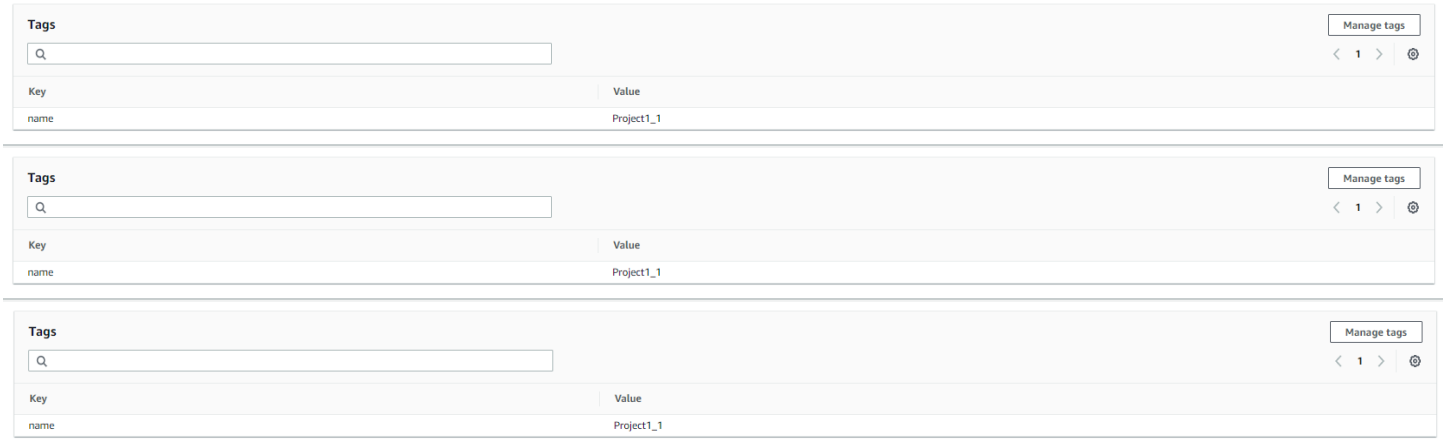




Log on the aws

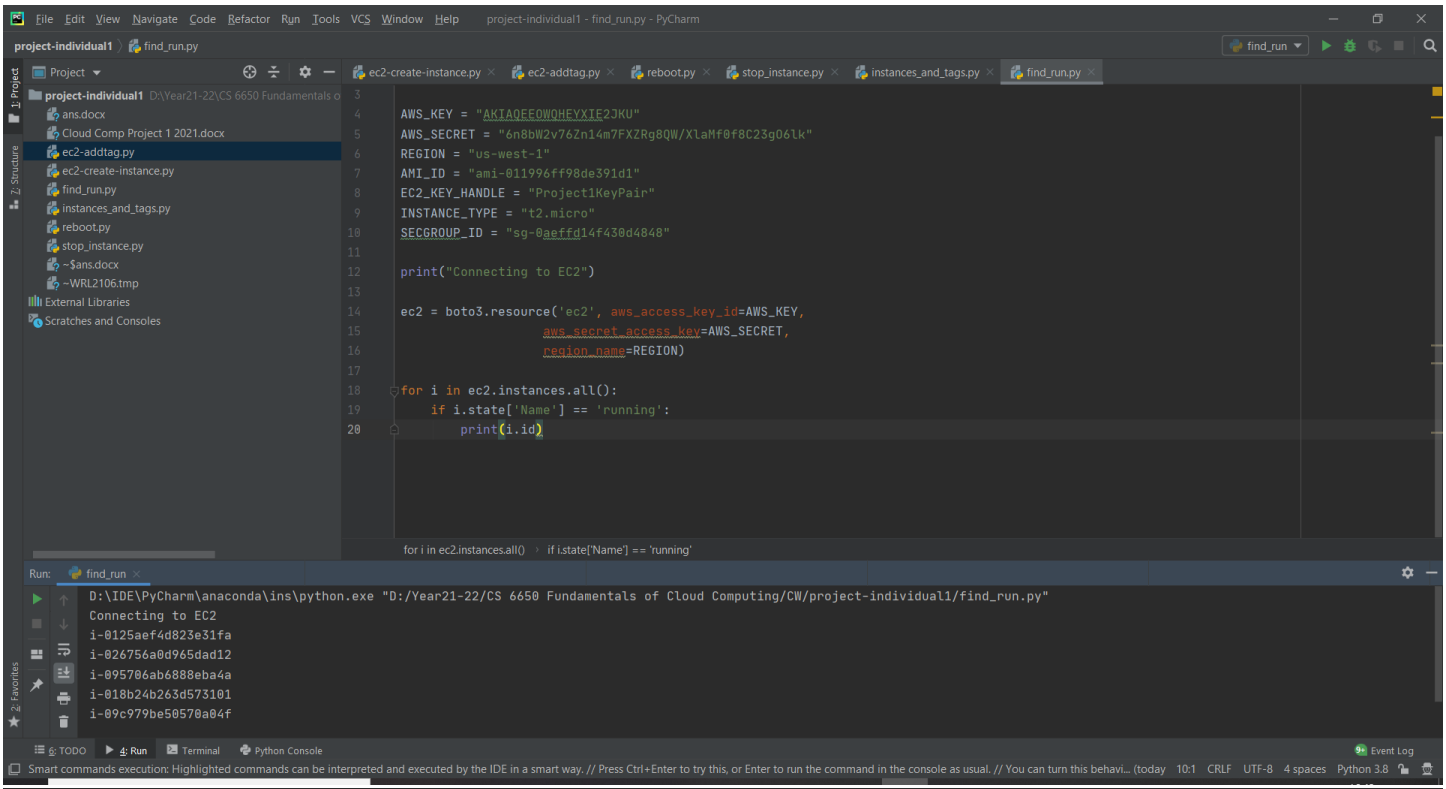


Project1_1



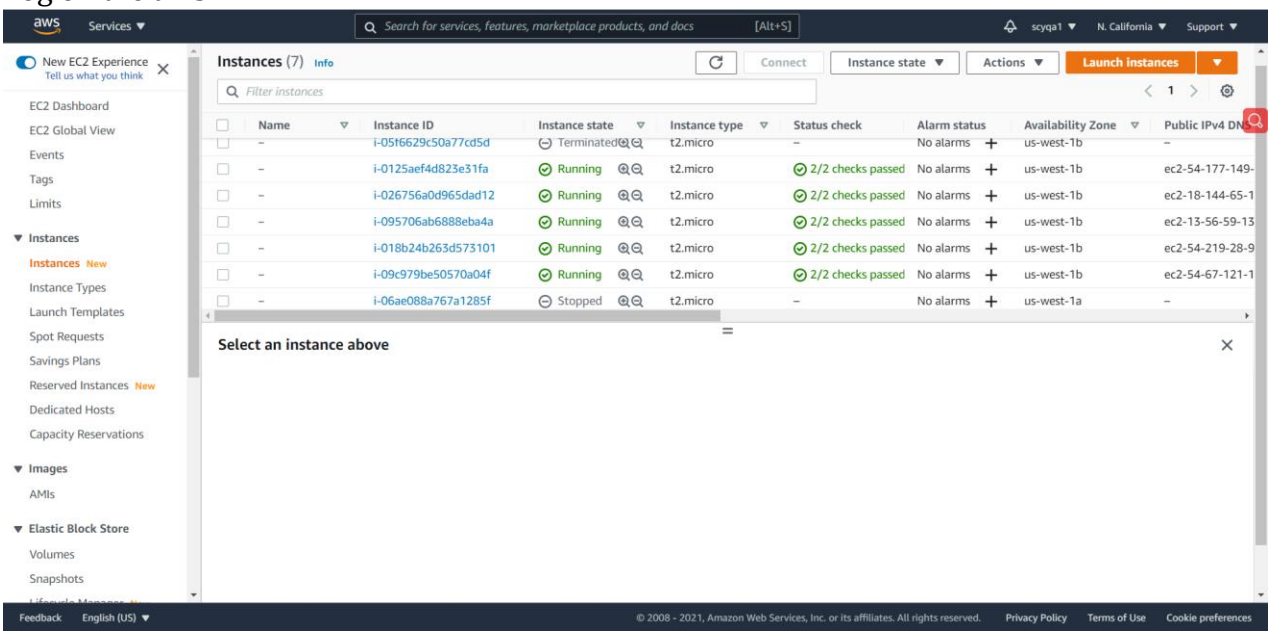
Project1_2

6. Find all the instance id that are actively running and report the same

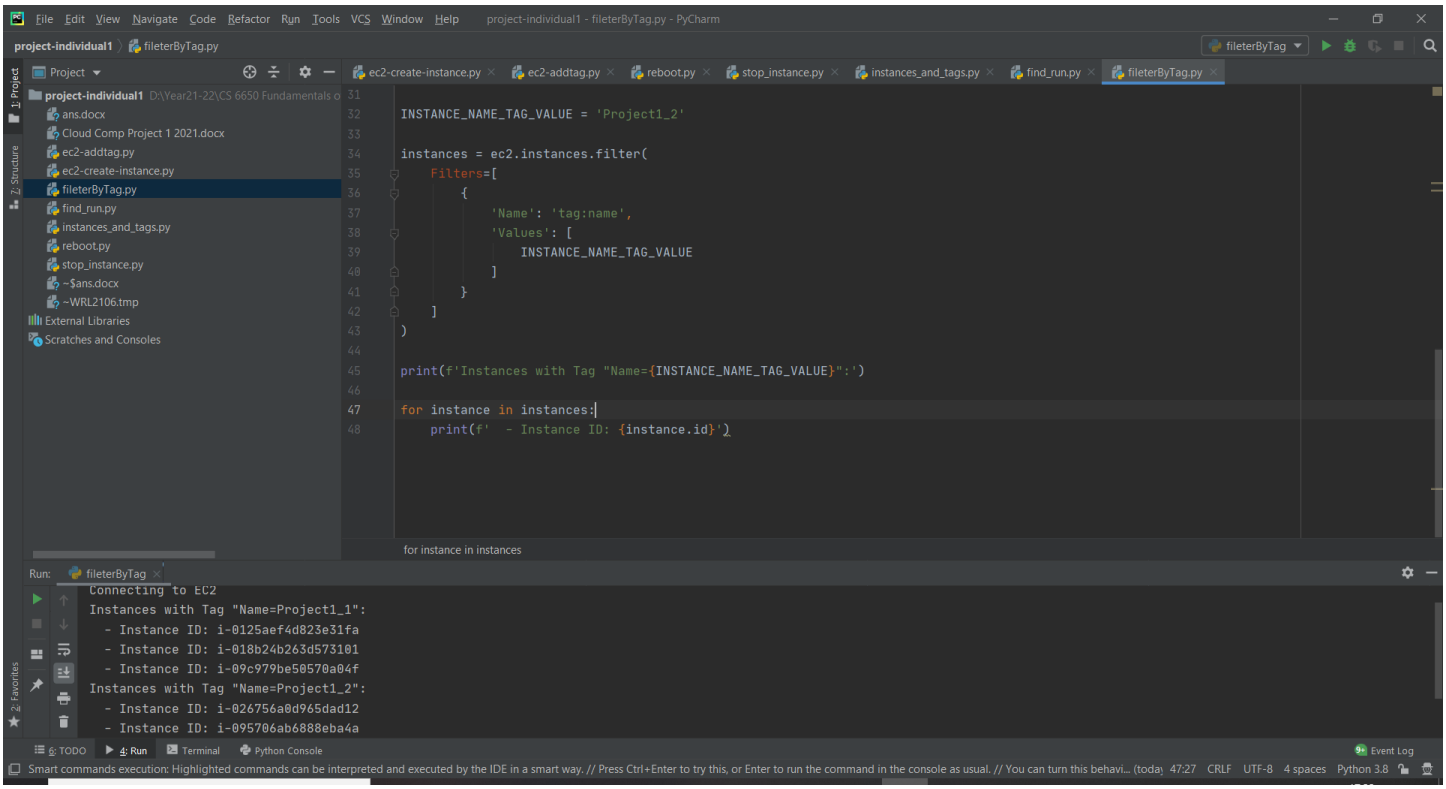


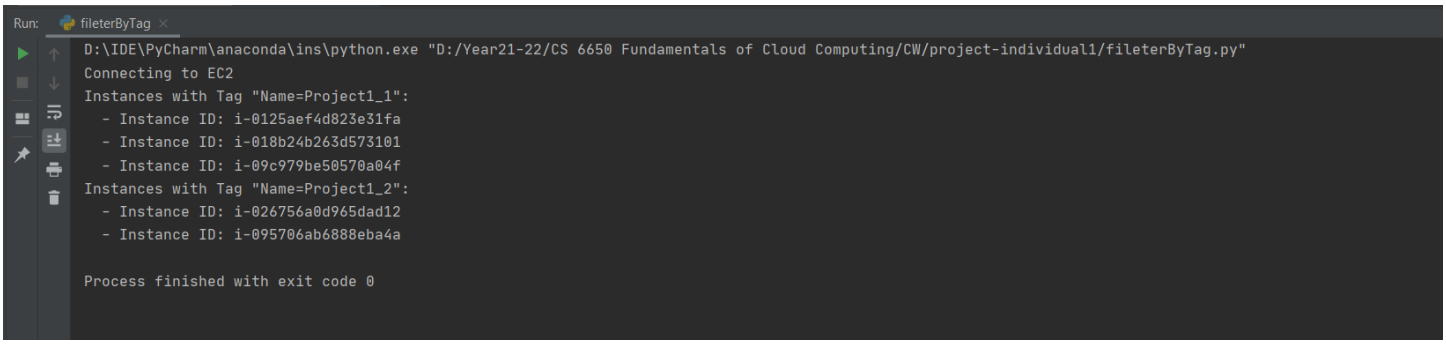

```
D:\IDE\PyCharm\anaconda\ins\python.exe "D:/Year21-22/CS 6650 Fundamentals of Cloud Computing/CW/project-individual1/find_run.py"
Connecting to EC2
i-0125aef4d823e31fa
i-026756a0d965dad12
i-095706ab6888eba4a
i-018b24b263d573101
i-09c979be50570a04f

Process finished with exit code 0
```

Log on the aws

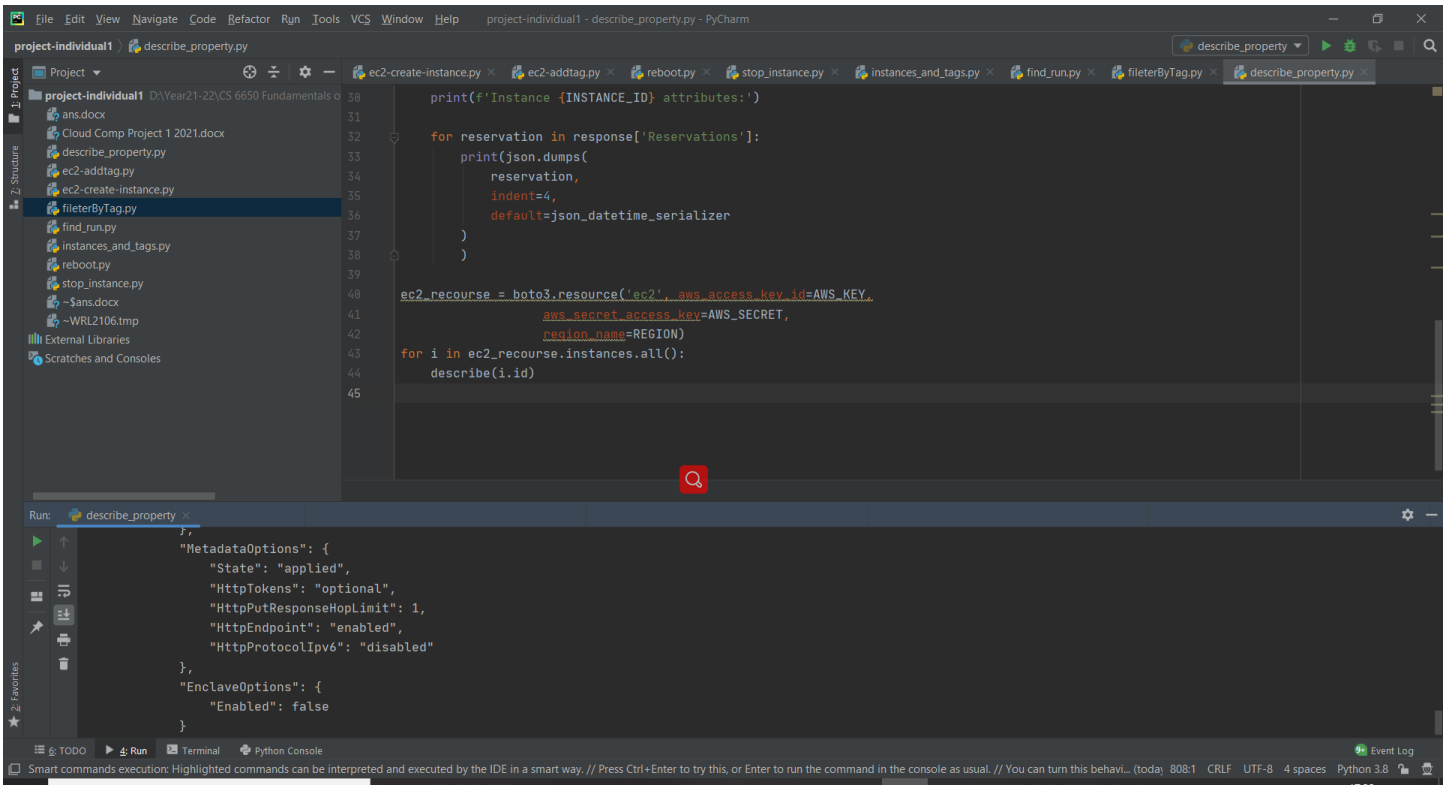


7. Filter the EC2 instance based on tags

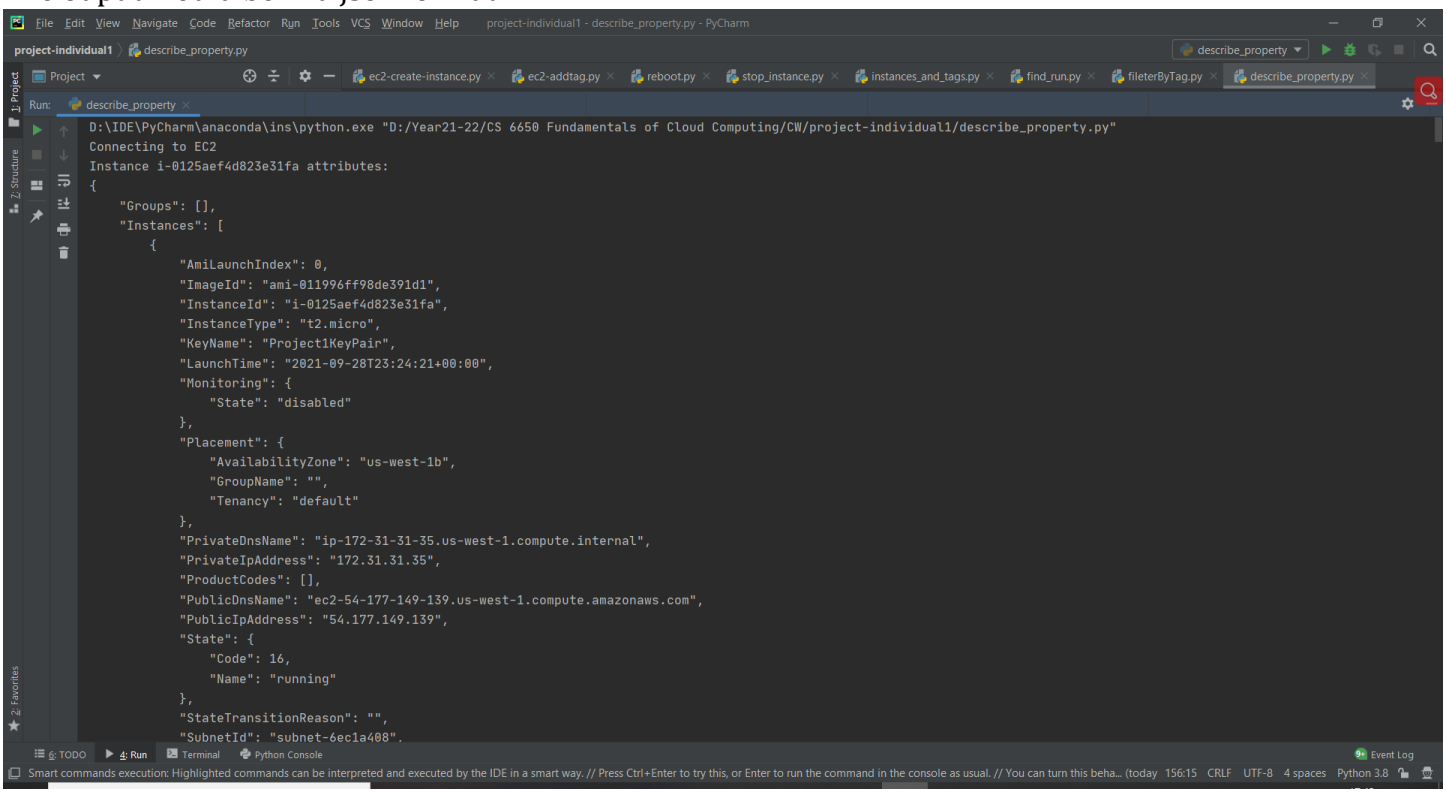


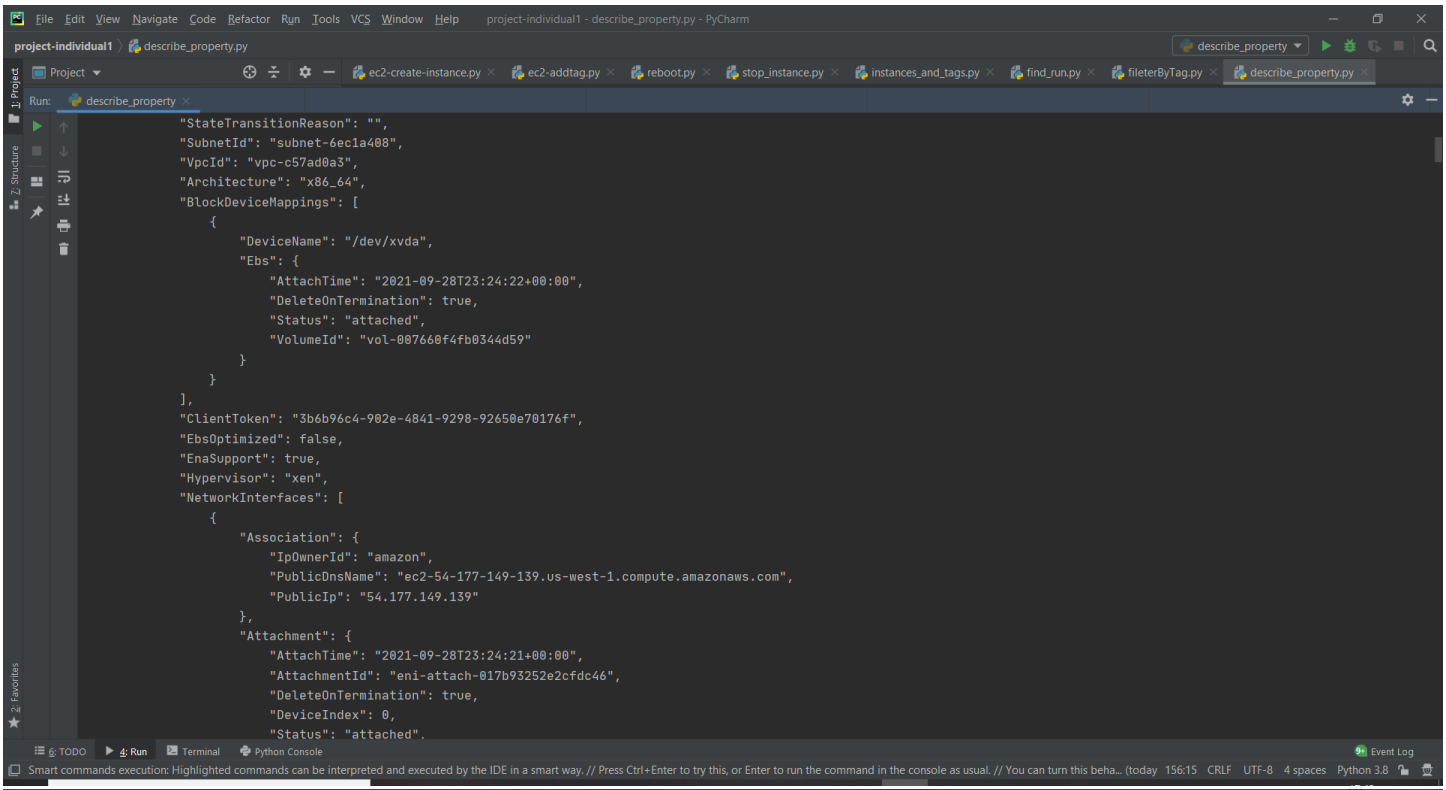


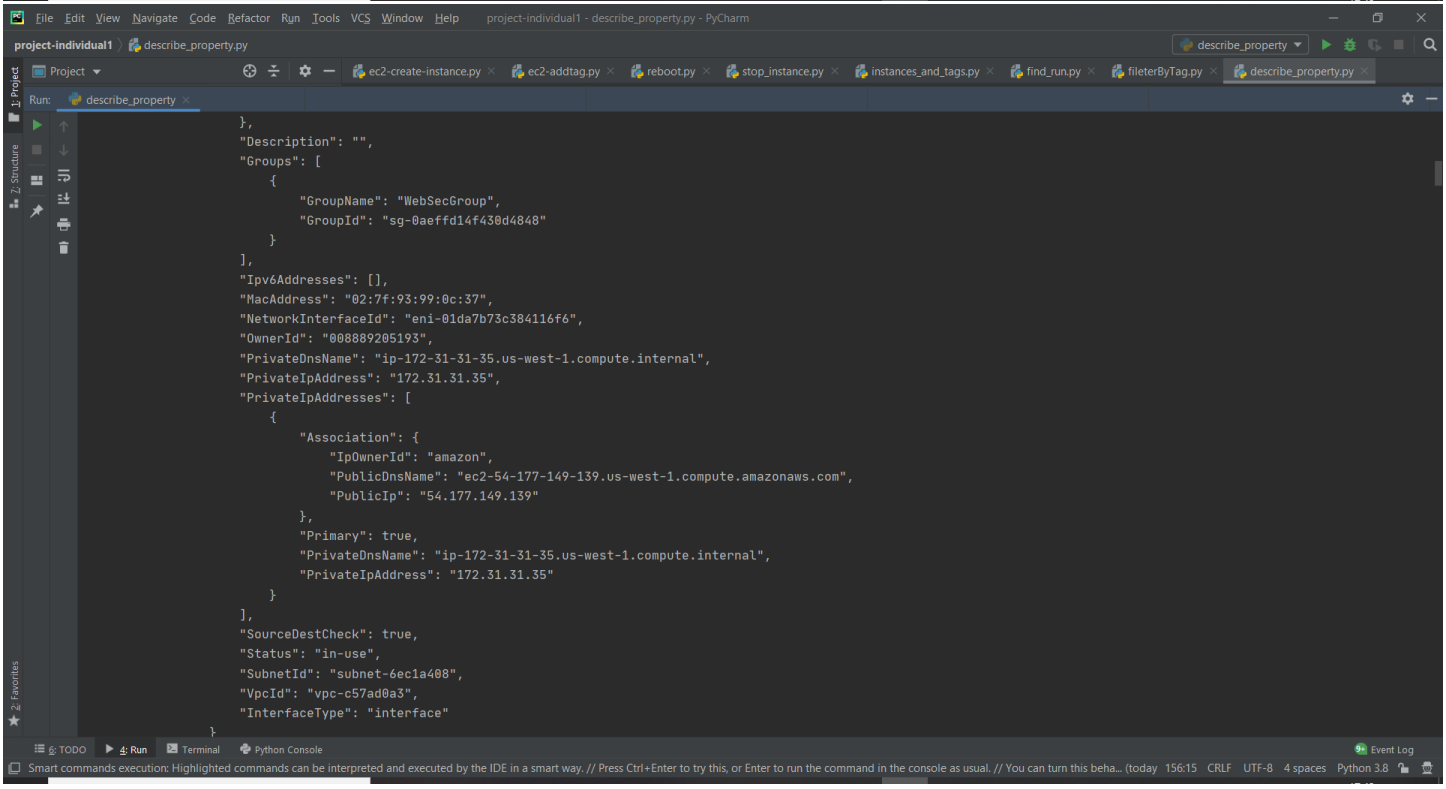
8. Describe the properties of all the EC2 instances

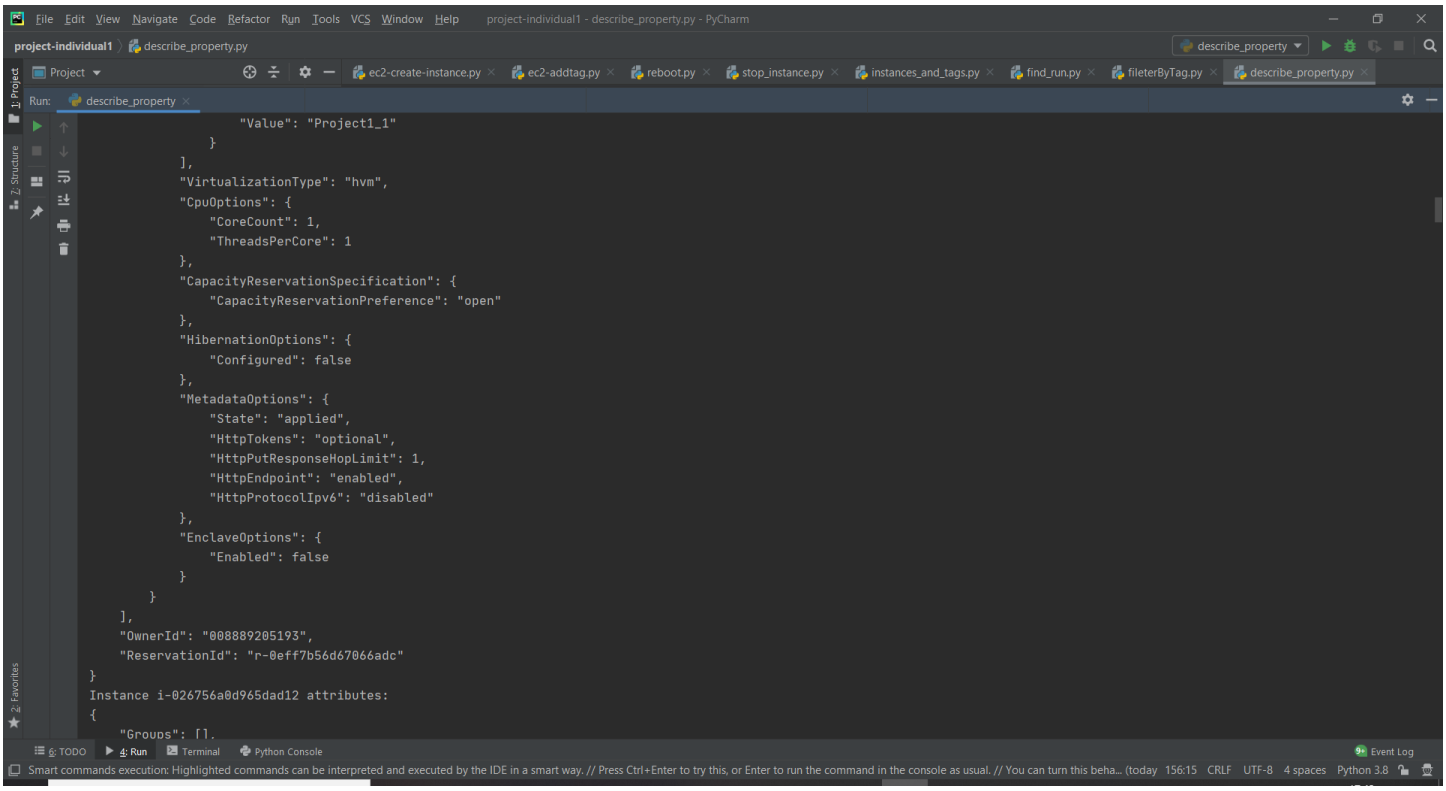


The ouput would be in a json format



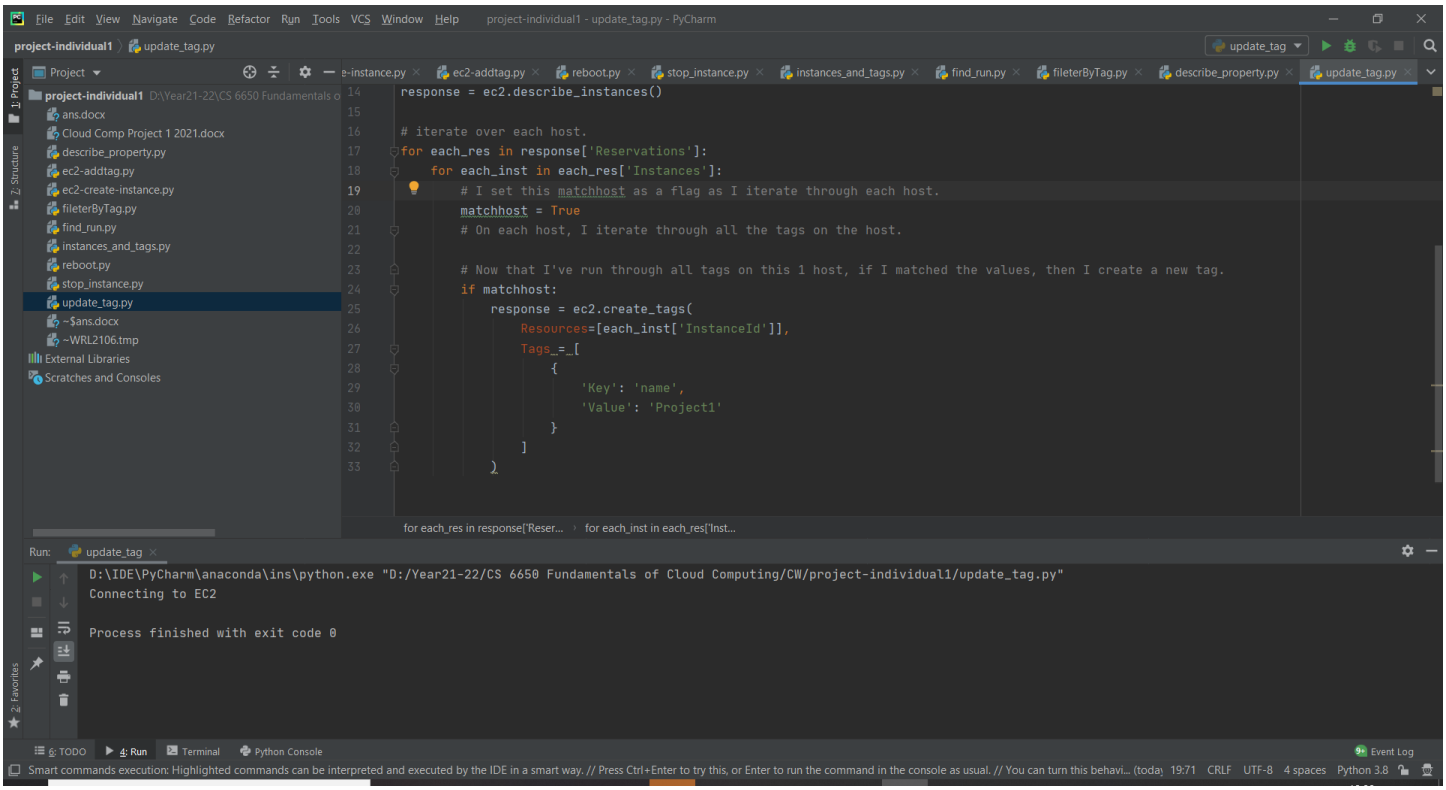

File Edit View Navigate Code Refactor Run Tools VCS Window Help
project-individual1 - describe_property.py - PyCharm
project-individual1 describe_property.py
describe_property
Project
ec2-create-instance.py
ec2-addtag.py
reboot.py
stop_instance.py
instances_and_tags.py
find_run.py
fileterByTag.py
describe_property.py
Run: describe_property
"StateTransitionReason": "",
"SubnetId": "subnet-6ec1a408",
"VpcId": "vpc-c57ad0a3",
"Architecture": "x86_64",
"BlockDeviceMappings": [
{
"DeviceName": "/dev/xvda",
"Ebs": {
"AttachTime": "2021-09-28T23:24:22+00:00",
"DeleteOnTermination": true,
"Status": "attached",
"VolumeId": "vol-007660f4fb0344d59"
}
}
],
"ClientToken": "3b6b96c4-902e-4841-9298-92650e70176f",
"EbsOptimized": false,
"EnaSupport": true,
"Hypervisor": "xen",
"NetworkInterfaces": [
{
"Association": {
"IpOwnerId": "amazon",
"PublicDnsName": "ec2-54-177-149-139.us-west-1.compute.amazonaws.com",
"PublicIp": "54.177.149.139"
},
"Attachment": {
"AttachTime": "2021-09-28T23:24:21+00:00",
"AttachmentId": "eni-attach-017b93252e2cfdc46",
"DeleteOnTermination": true,
"DeviceIndex": 0,
"Status": "attached",
6: TODO
4: Run
Terminal
Python Console
Event Log
Smart commands execution: Highlighted commands can be interpreted and executed by the IDE in a smart way. // Press Ctrl+Enter to try this, or Enter to run the command in the console as usual. // You can turn this beha... (today 156:15 CRLF UTF-8 4 spaces Python 3.8


File Edit View Navigate Code Refactor Run Tools VCS Window Help
project-individual1 - describe_property.py - PyCharm
project-individual1 describe_property.py
describe_property
Project
ec2-create-instance.py
ec2-addtag.py
reboot.py
stop_instance.py
instances_and_tags.py
find_run.py
fileterByTag.py
describe_property.py
Run: describe_property
},
"Description": "",
"Groups": [
{
"GroupName": "WebSecGroup",
"GroupId": "sg-0aeffd14f430d4848"
}
],
"Ipv6Addresses": [],
"MacAddress": "02:7f:93:99:0c:37",
"NetworkInterfaceId": "eni-01da7b73c384116f6",
"OwnerId": "008889205193",
"PrivateDnsName": "ip-172-31-31-35.us-west-1.compute.internal",
"PrivateIpAddress": "172.31.31.35",
"PrivateIpAddresses": [
{
"Association": {
"IpOwnerId": "amazon",
"PublicDnsName": "ec2-54-177-149-139.us-west-1.compute.amazonaws.com",
"PublicIp": "54.177.149.139"
},
"Primary": true,
"PrivateDnsName": "ip-172-31-31-35.us-west-1.compute.internal",
"PrivateIpAddress": "172.31.31.35"
}
],
"SourceDestCheck": true,
"Status": "in-use",
"SubnetId": "subnet-6ec1a408",
"VpcId": "vpc-c57ad0a3",
"InterfaceType": "interface"
}
6: TODO
4: Run
Terminal
Python Console
Event Log
Smart commands execution: Highlighted commands can be interpreted and executed by the IDE in a smart way. // Press Ctrl+Enter to try this, or Enter to run the command in the console as usual. // You can turn this beha... (today 156:15 CRLF UTF-8 4 spaces Python 3.8

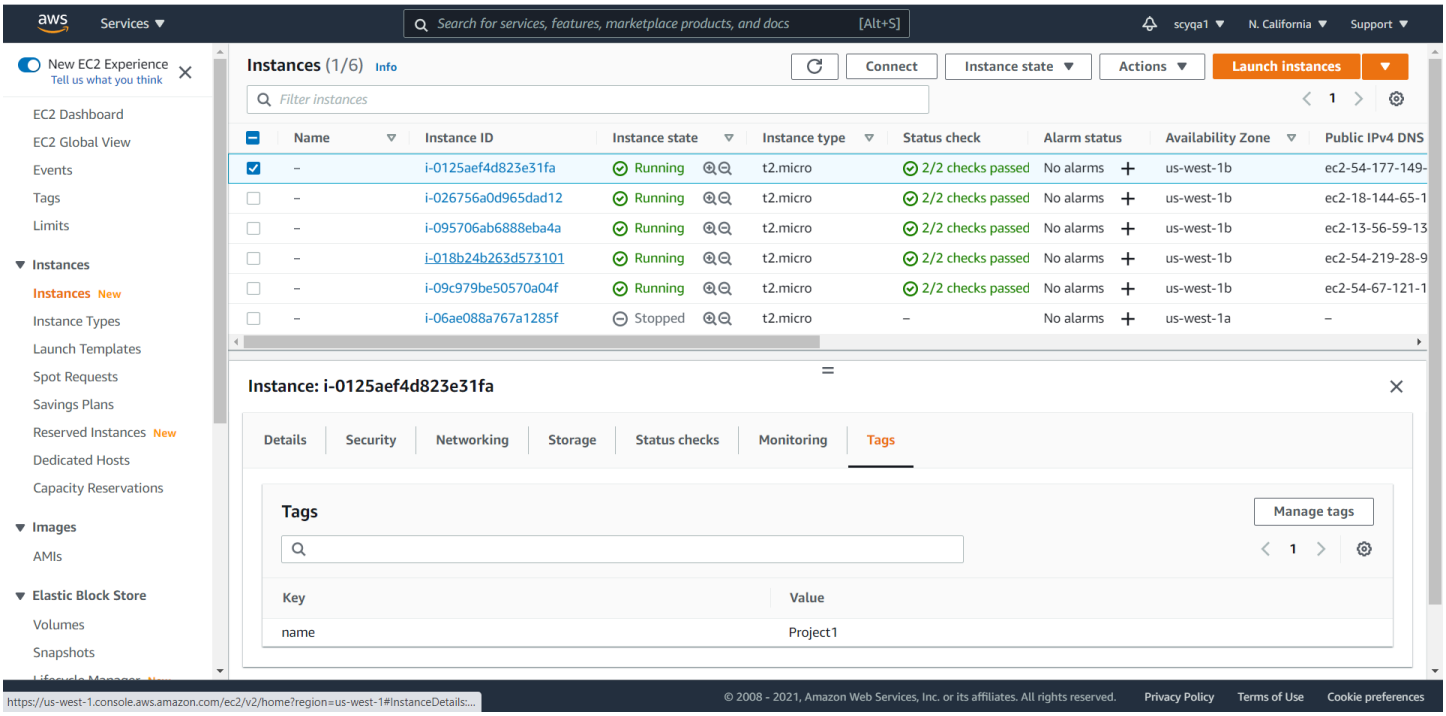


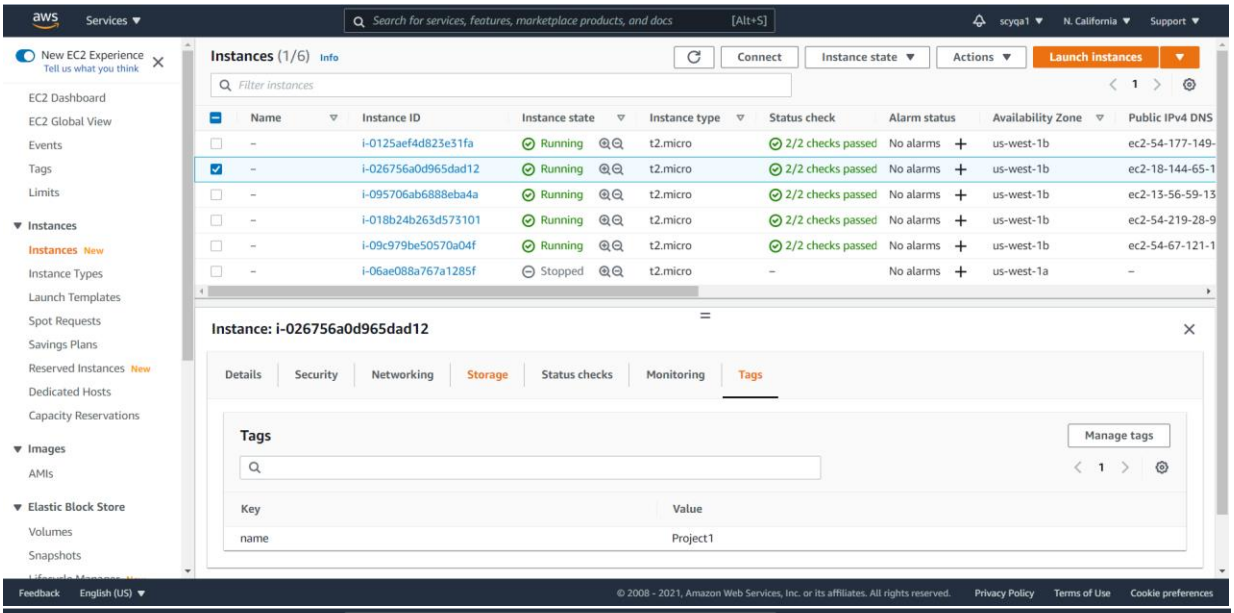
As you can see from the progress bar on the right side of the terminal, there are too many properties, so that I don't include all the screenshots.

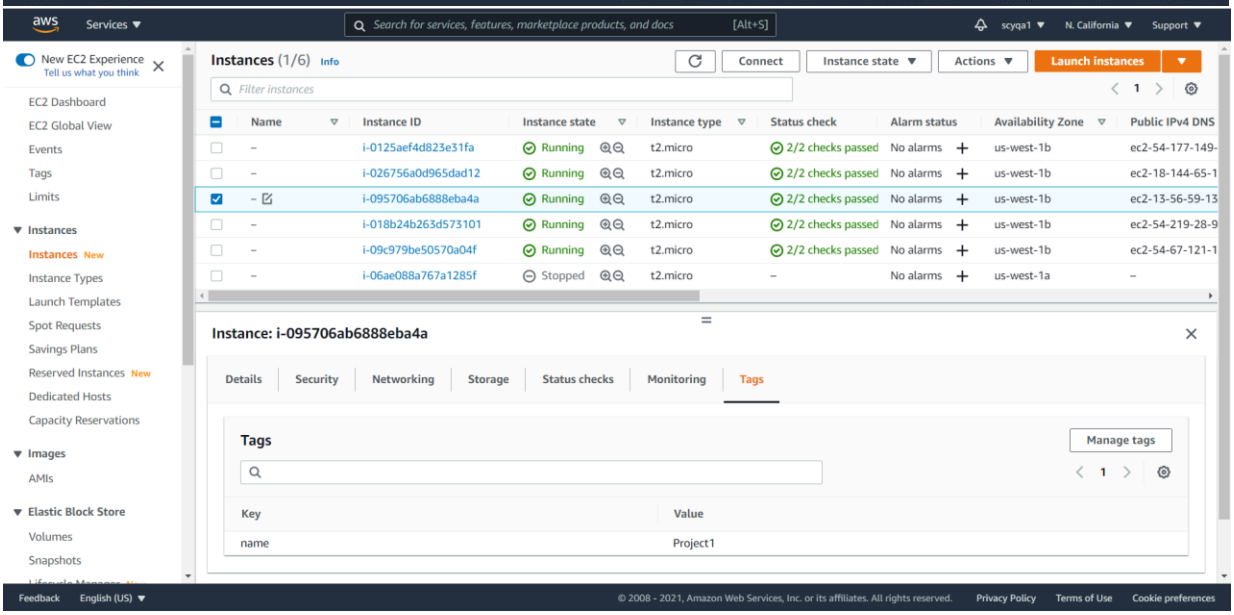
9. Update the tag to make sure that there is only one tag that is assigned to all the ec2 instance

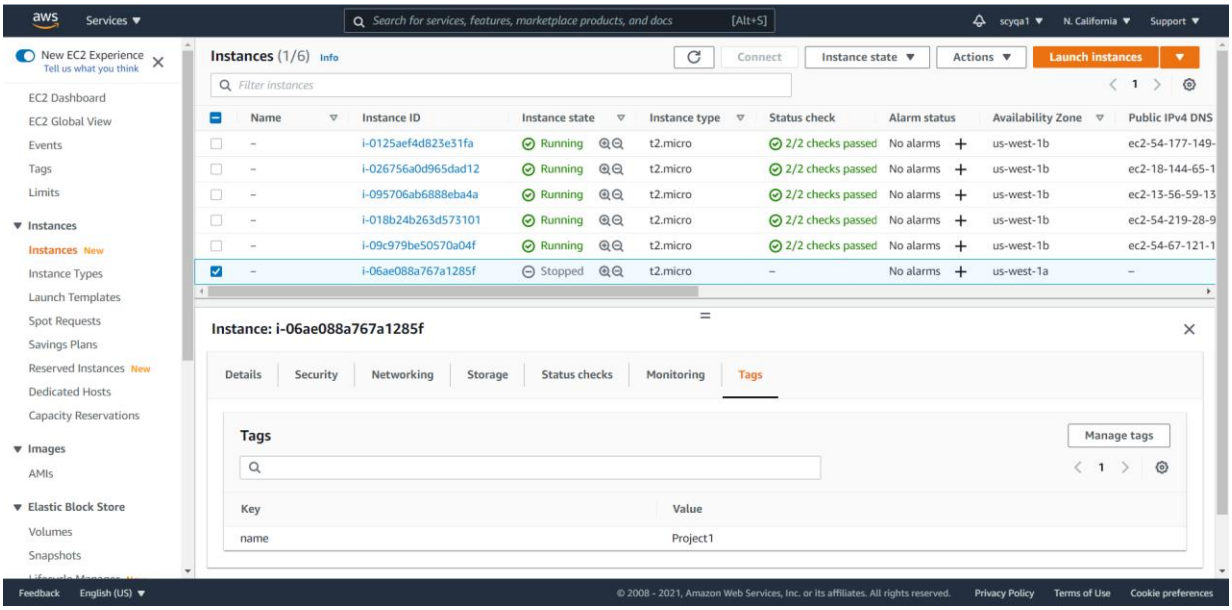


By the function of 'create_tags', all tags could be updated to new tag 'Project1'

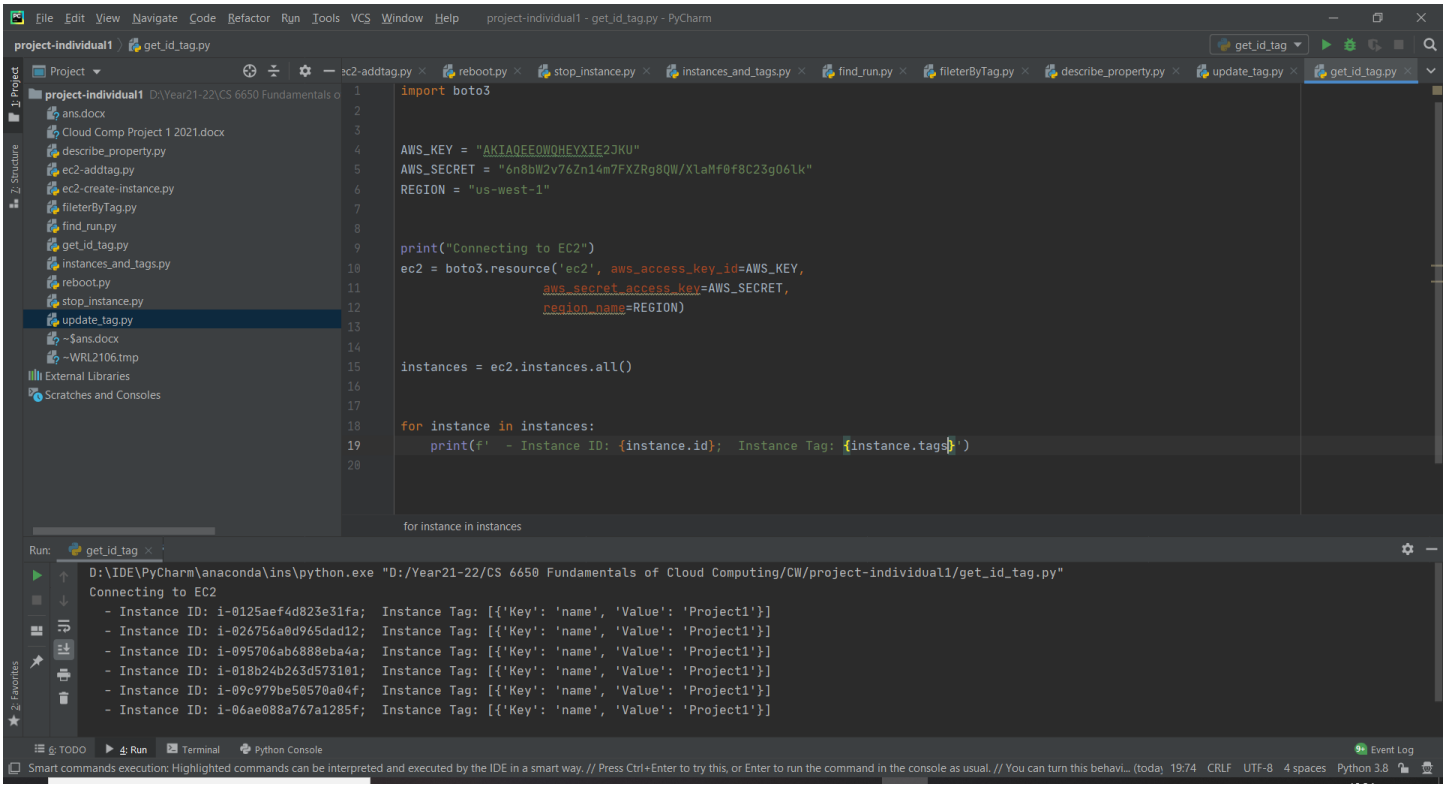



Services
Search for services, features, marketplace products, and docs [Alt+S]
scyqa1 N. California Support
New EC2 Experience
Tell us what you think
EC2 Dashboard
EC2 Global View
Events
Tags
Limits
Instances
Instances New
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances New
Dedicated Hosts
Capacity Reservations
Images
AMIs
Elastic Block Store
Volumes
Snapshots
Instances (1/6) Info
Connect
Instance state
Actions
Launch instances
Filter instances
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
Public IPv4 DNS
i-0125aef4d823e31fa Running t2.micro 2/2 checks passed No alarms us-west-1b ec2-54-177-149-
i-026756a0d965dad12 Running t2.micro 2/2 checks passed No alarms us-west-1b ec2-18-144-65-1
i-095706ab6888eba4a Running t2.micro 2/2 checks passed No alarms us-west-1b ec2-13-56-59-13
i-018b24b263d573101 Running t2.micro 2/2 checks passed No alarms us-west-1b ec2-54-219-28-9
i-09c979be50570a04f Running t2.micro 2/2 checks passed No alarms us-west-1b ec2-54-67-121-1
i-06ae088a767a1285f Stopped t2.micro – No alarms us-west-1a –
Instance: i-026756a0d965dad12
Details
Security
Networking
Storage
Status checks
Monitoring
Tags
Tags
Manage tags
Key
Value
name
Project1
Feedback
English (US)
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Cookie preferences

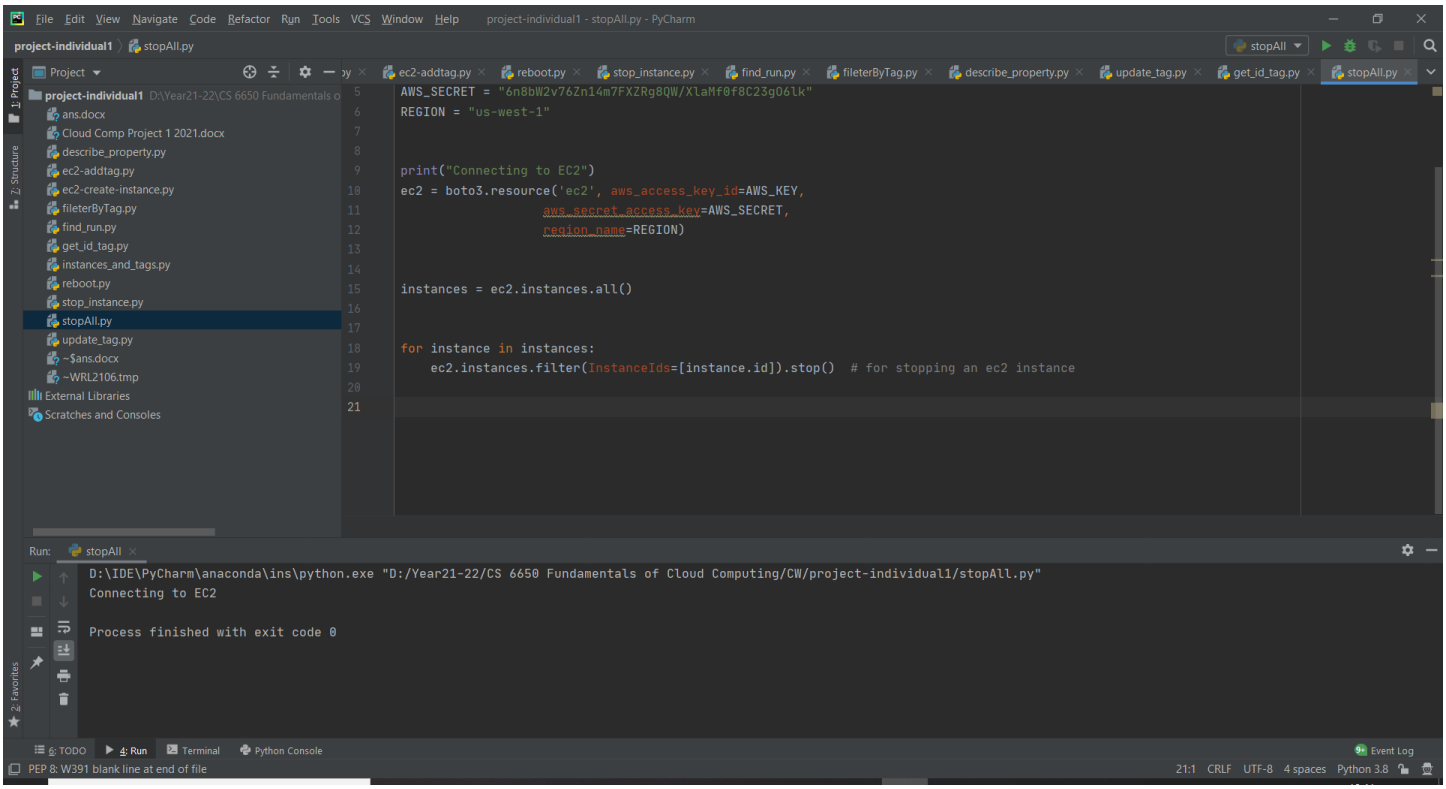

Services
Search for services, features, marketplace products, and docs [Alt+S]
scyqa1 N. California Support
New EC2 Experience
Tell us what you think
EC2 Dashboard
EC2 Global View
Events
Tags
Limits
Instances
Instances New
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances New
Dedicated Hosts
Capacity Reservations
Images
AMIs
Elastic Block Store
Volumes
Snapshots
Instances (1/6) Info
Connect
Instance state
Actions
Launch instances
Filter instances
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
Public IPv4 DNS
i-0125aef4d823e31fa Running t2.micro 2/2 checks passed No alarms us-west-1b ec2-54-177-149-
i-026756a0d965dad12 Running t2.micro 2/2 checks passed No alarms us-west-1b ec2-18-144-65-1
i-095706ab6888eba4a Running t2.micro 2/2 checks passed No alarms us-west-1b ec2-13-56-59-13
i-018b24b263d573101 Running t2.micro 2/2 checks passed No alarms us-west-1b ec2-54-219-28-9
i-09c979be50570a04f Running t2.micro 2/2 checks passed No alarms us-west-1b ec2-54-67-121-1
i-06ae088a767a1285f Stopped t2.micro – No alarms us-west-1a –
Instance: i-095706ab6888eba4a
Details
Security
Networking
Storage
Status checks
Monitoring
Tags
Tags
Manage tags
Key
Value
name
Project1
Feedback
English (US)
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Cookie preferences


Services
Search for services, features, marketplace products, and docs [Alt+S]
scyqa1 N. California Support
New EC2 Experience
Tell us what you think
EC2 Dashboard
EC2 Global View
Events
Tags
Limits
Instances
Instances New
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances New
Dedicated Hosts
Capacity Reservations
Images
AMIs
Elastic Block Store
Volumes
Snapshots
Instances (1/6) Info
Connect
Instance state
Actions
Launch instances
Filter instances
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
Public IPv4 DNS
i-0125aef4d823e31fa Running t2.micro 2/2 checks passed No alarms us-west-1b ec2-54-177-149-
i-026756a0d965dad12 Running t2.micro 2/2 checks passed No alarms us-west-1b ec2-18-144-65-1
i-095706ab6888eba4a Running t2.micro 2/2 checks passed No alarms us-west-1b ec2-13-56-59-13
i-018b24b263d573101 Running t2.micro 2/2 checks passed No alarms us-west-1b ec2-54-219-28-9
i-09c979be50570a04f Running t2.micro 2/2 checks passed No alarms us-west-1b ec2-54-67-121-1
i-06ae088a767a1285f Stopped t2.micro – No alarms us-west-1a –
Instance: i-06ae088a767a1285f
Details
Security
Networking
Storage
Status checks
Monitoring
Tags
Tags
Manage tags
Key
Value
name
Project1
Feedback
English (US)
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Cookie preferences

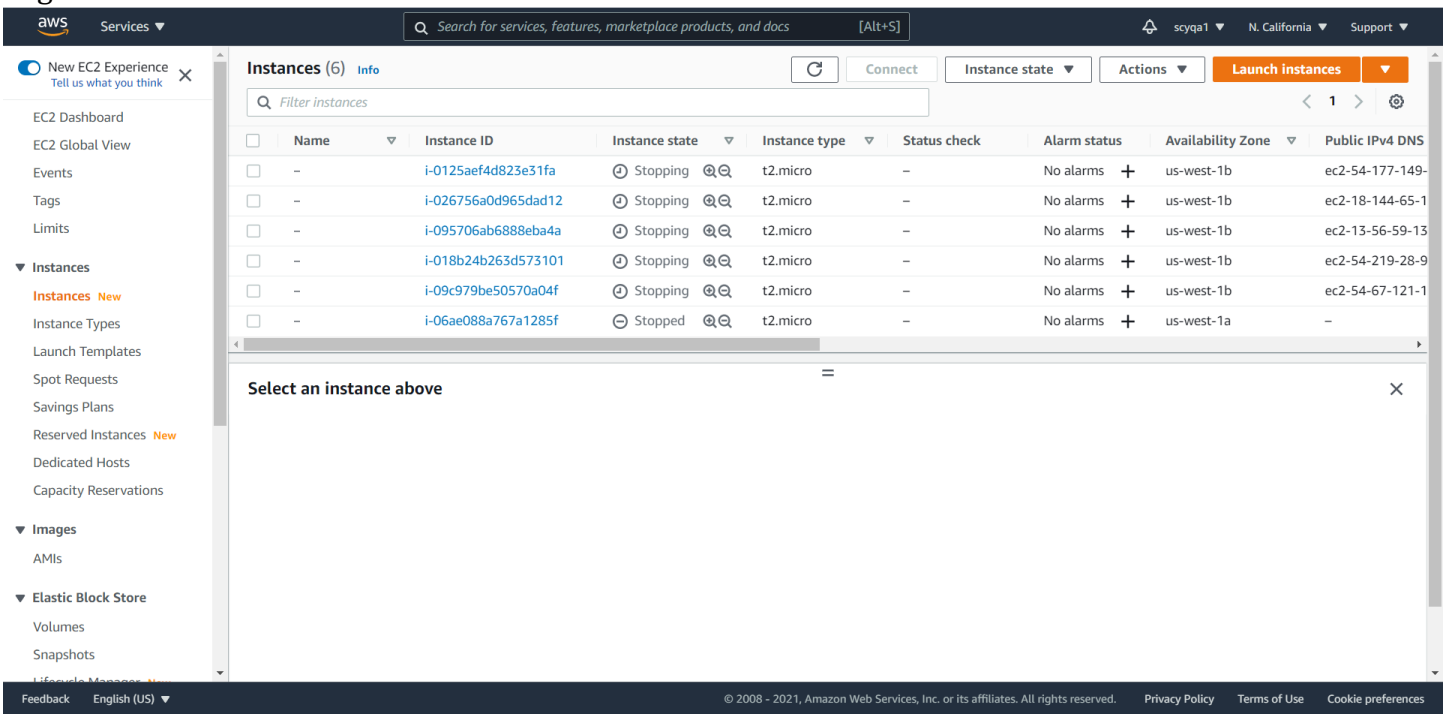
10. List all the instance id and their tags



11. Stop all the instances using the instance id’s



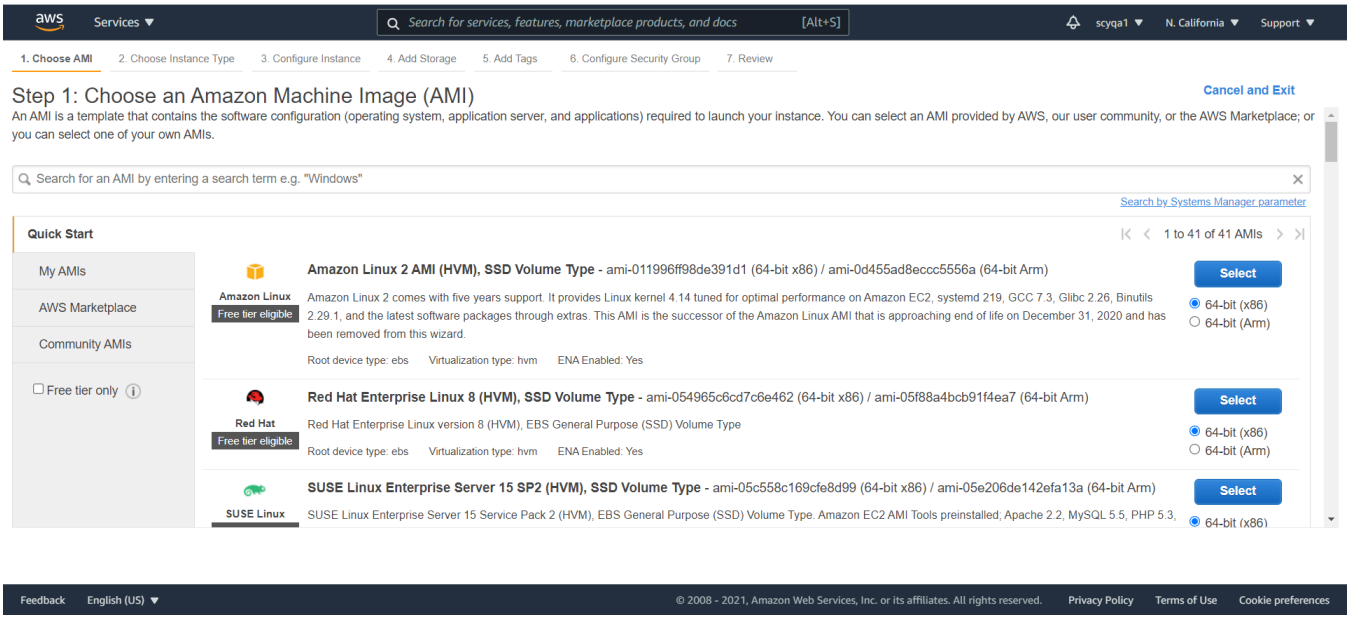
Log on the aws



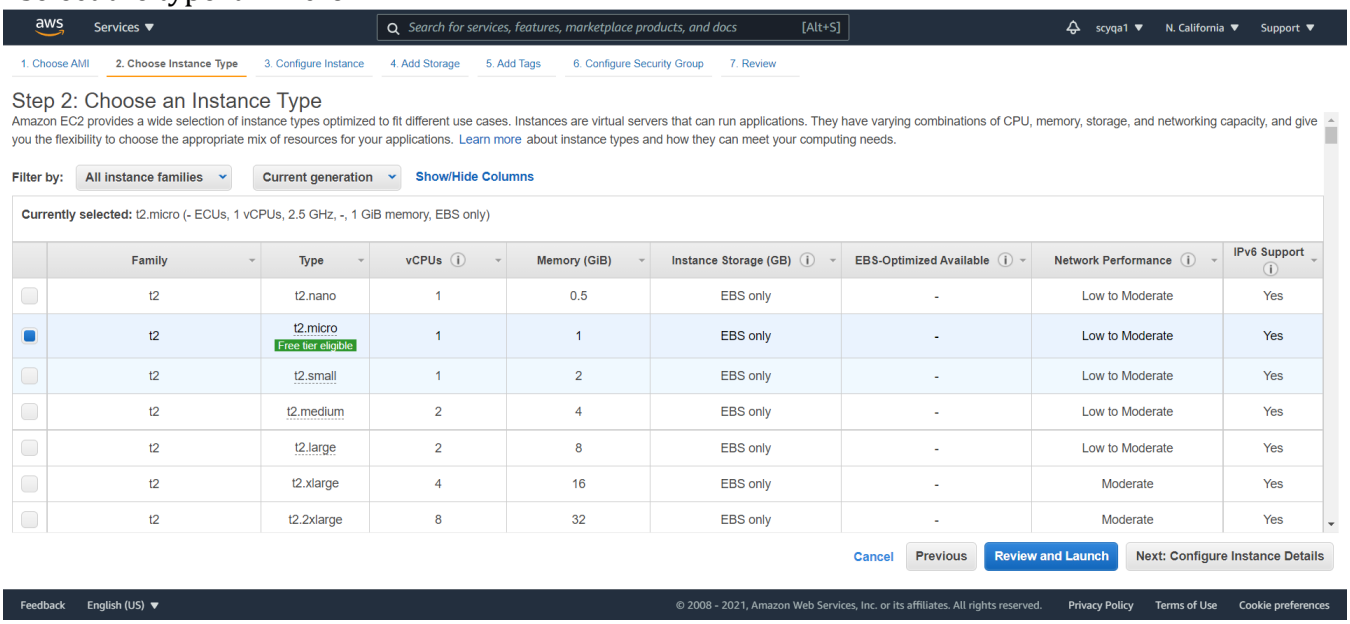
Part 2

1)Set up the EC2 instance, using t2.micro, add tag to it and it has to be accessed via any IP address

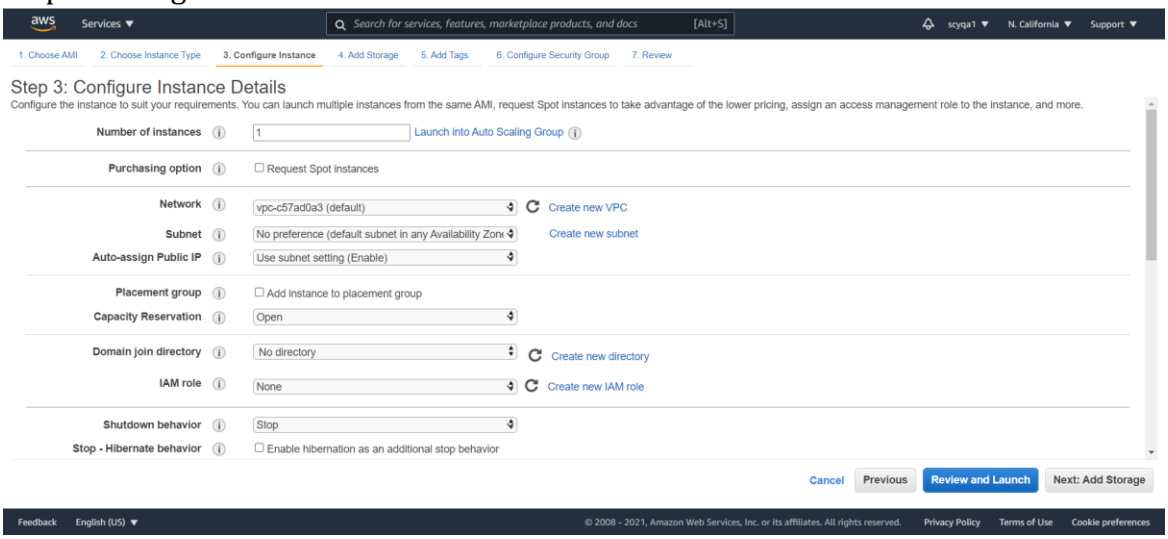
Step 1: choose an Amazon Machine Image (AMI)
I select the Amazon Linux 2 AMI.



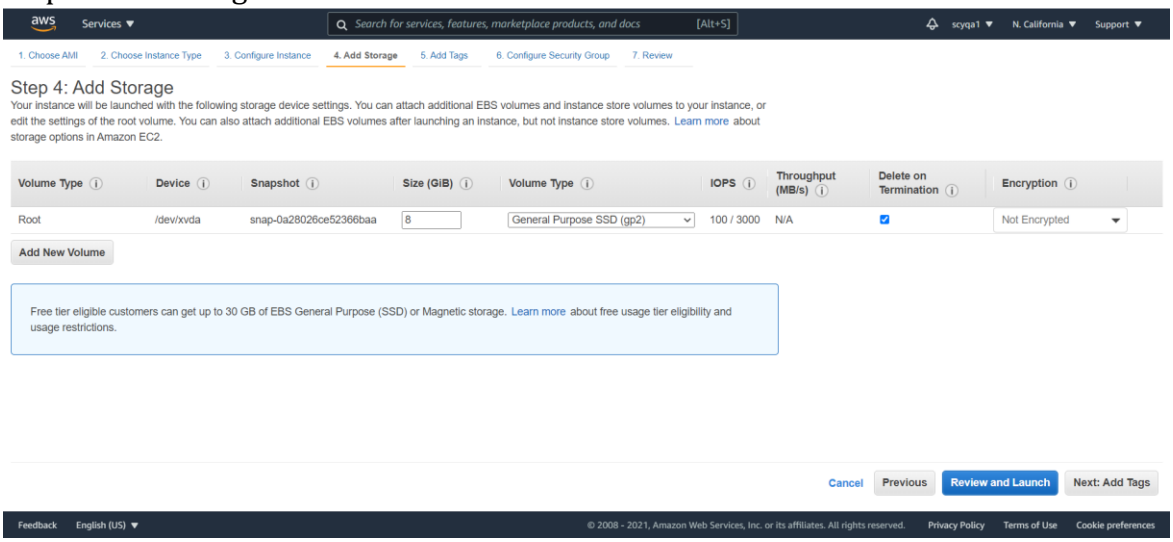
Step 2: choose an instance type
I select the type 't2.micro'.



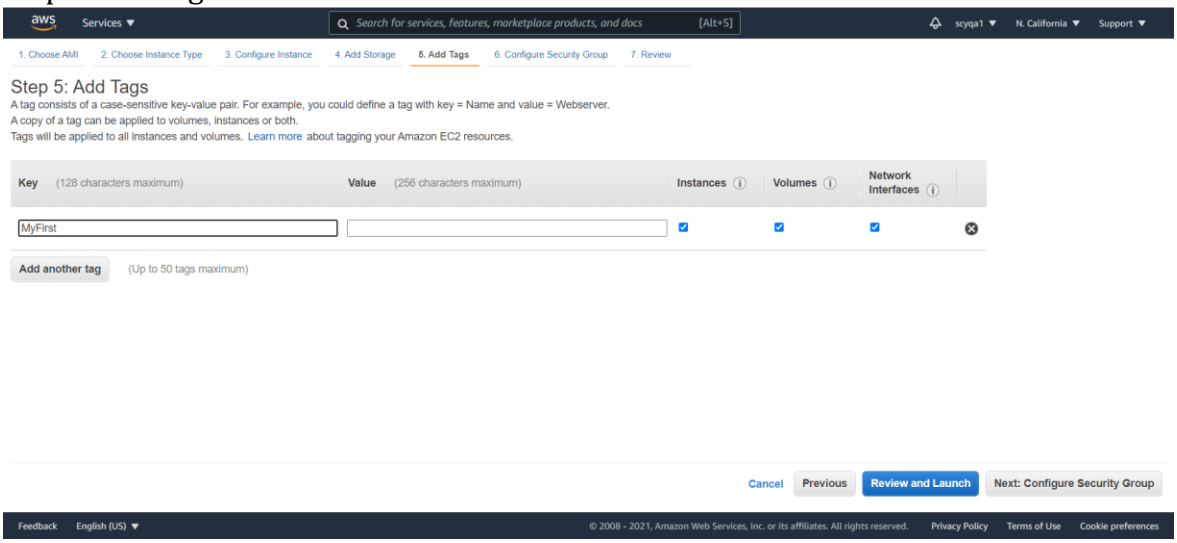
Step 3: configure instance details.



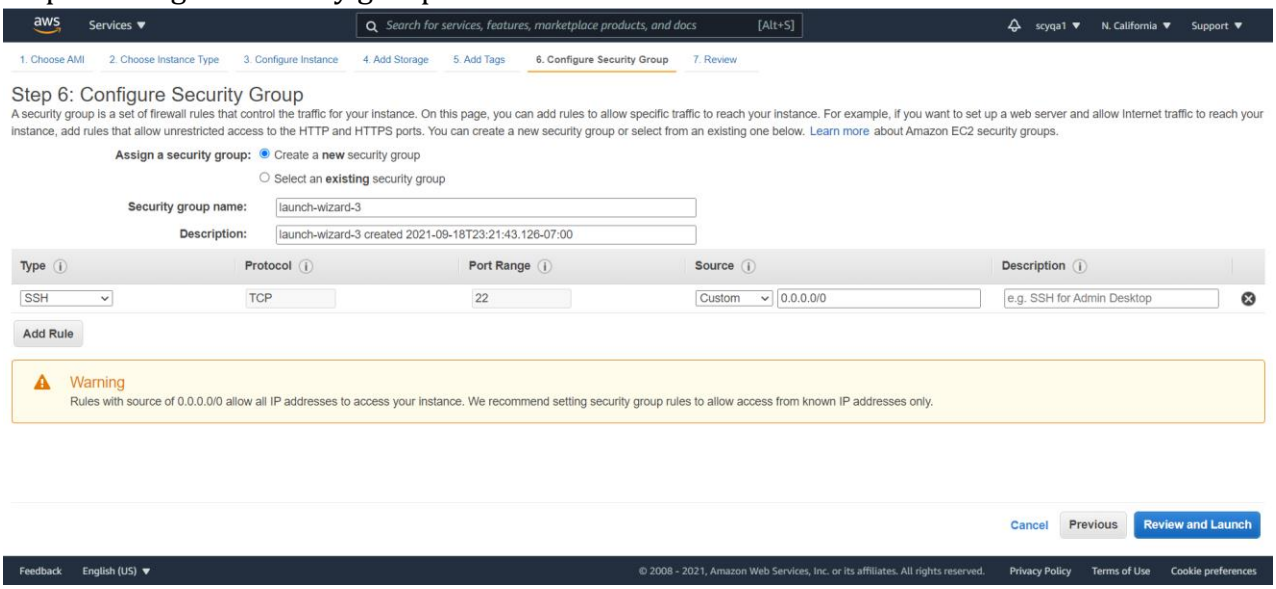
Step 4: add storage.

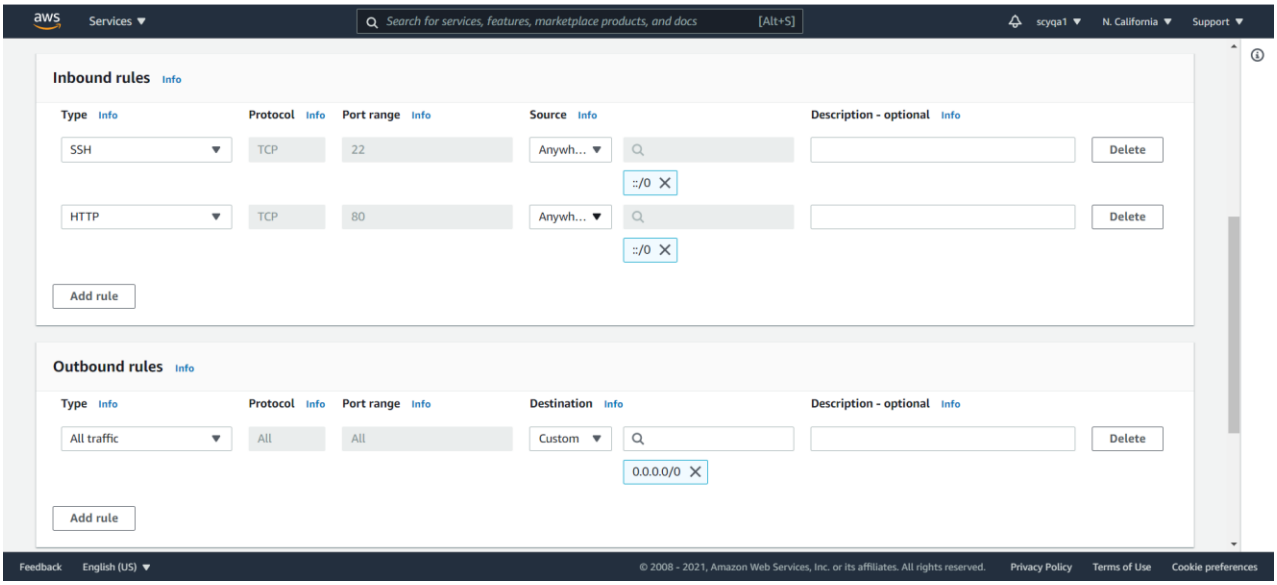


Step 5: add tags



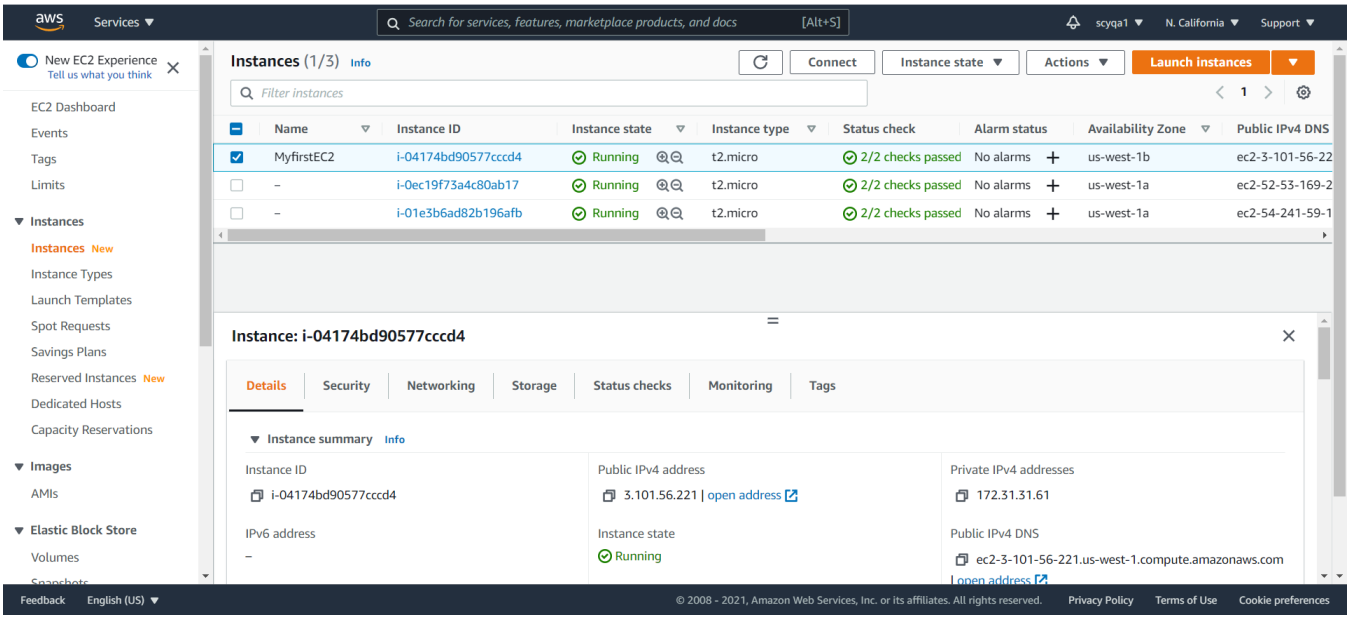
Step 6: configure security group





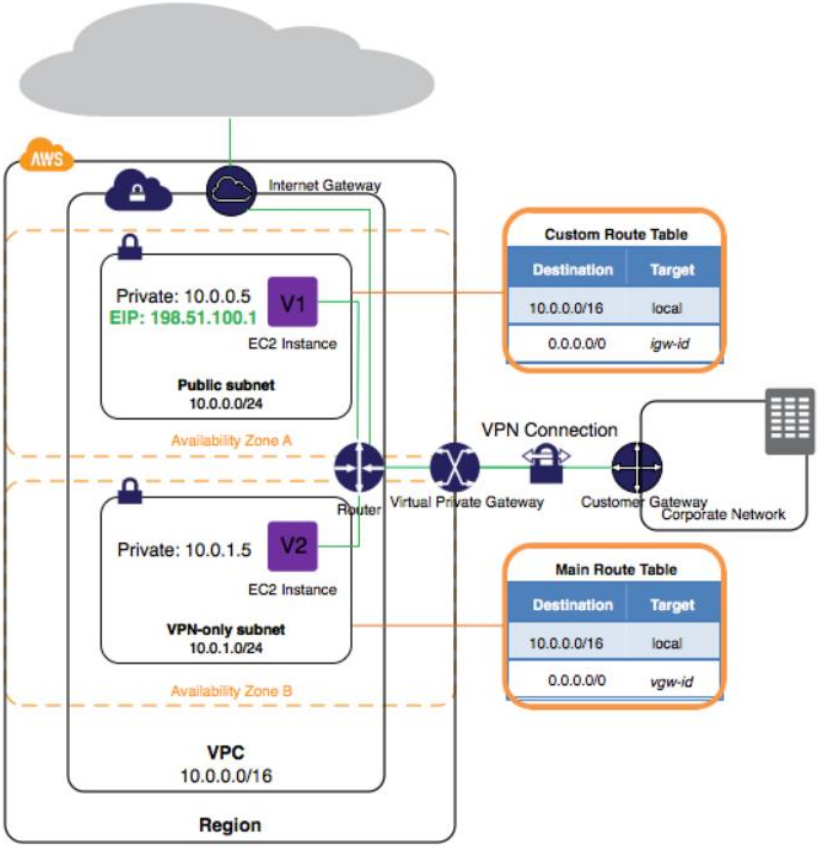
According to the figure about security group configuration above, the inbound rules of source is anywhere, and the outbound rules is all traffic. So it can be accessed via any IP address.

Final result

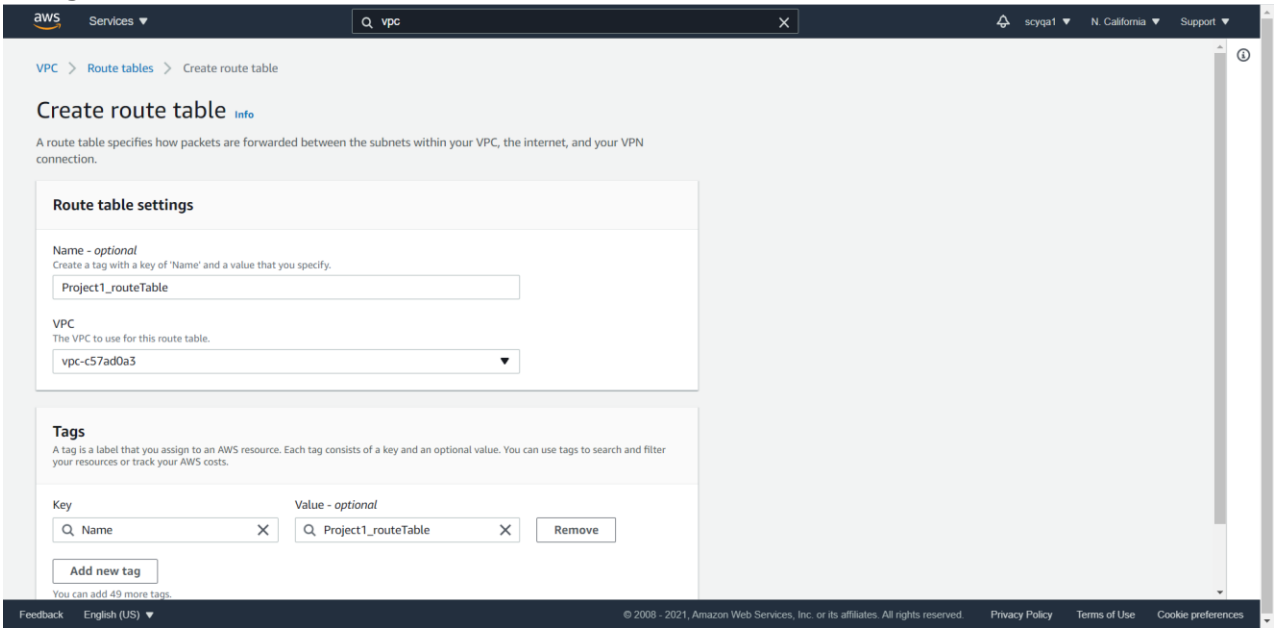


2)Explain in brief the process by setting up a routing table

The following figure shows the routing of a VPC with both an Internet gateway and a virtual private gateway, as well as a public subnet and a VPN-only subnet.



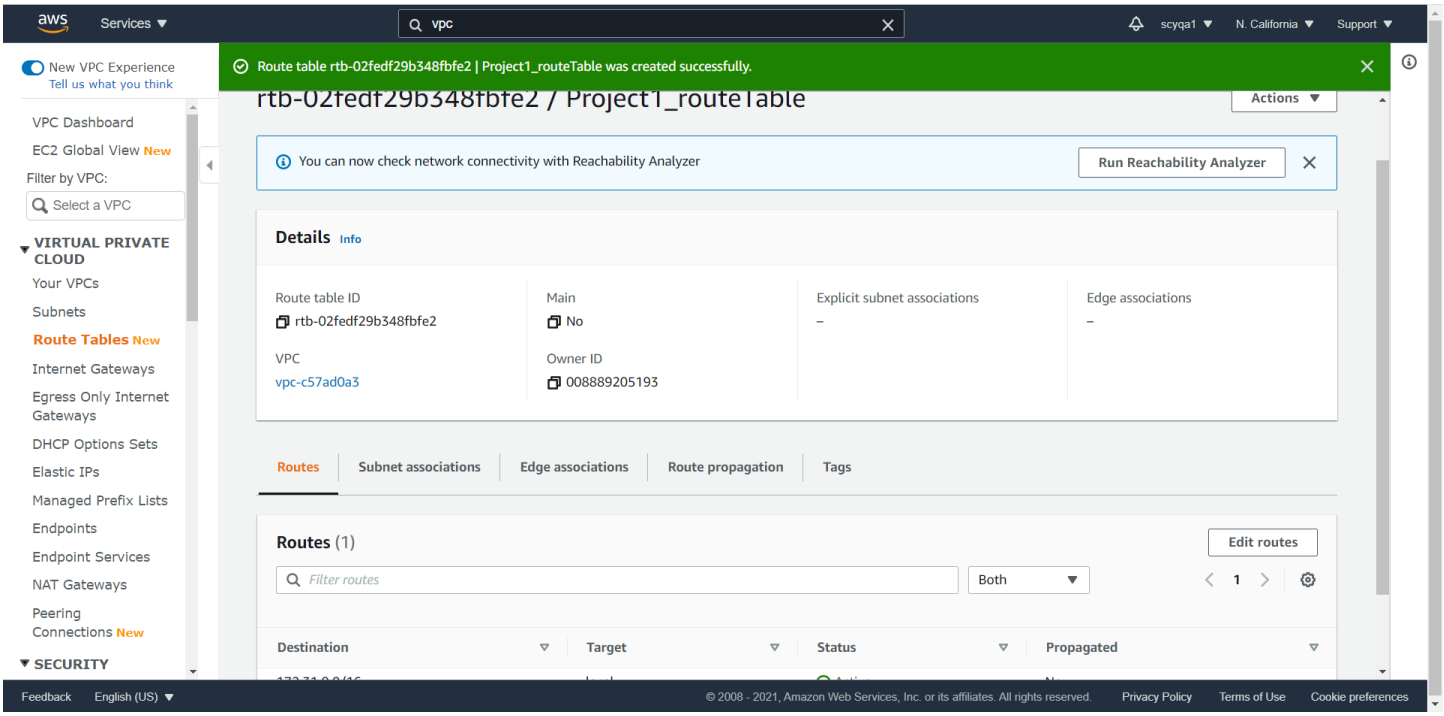
Using Amazon Web Services, I can create route table in VPC services.



Create route table

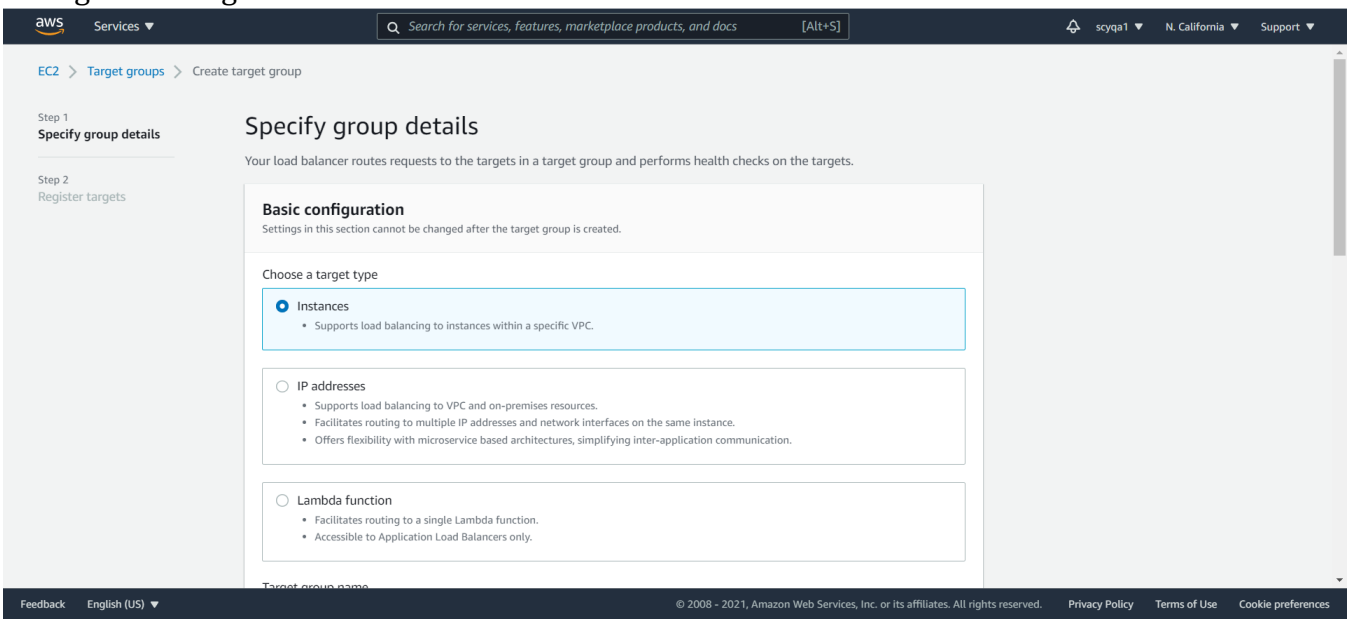
In the process, I need to choose a Virtual Private Cloud line to be used for the route table.

Result:

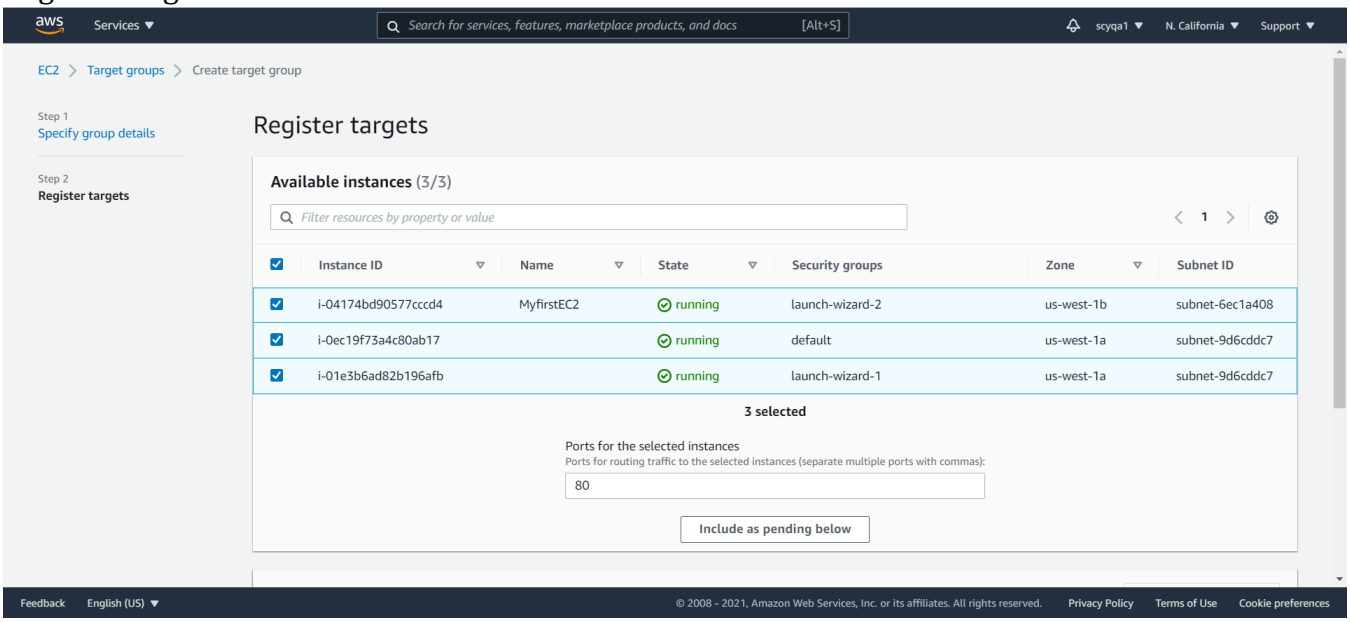


3)Set up an Application Load Balancer in the console

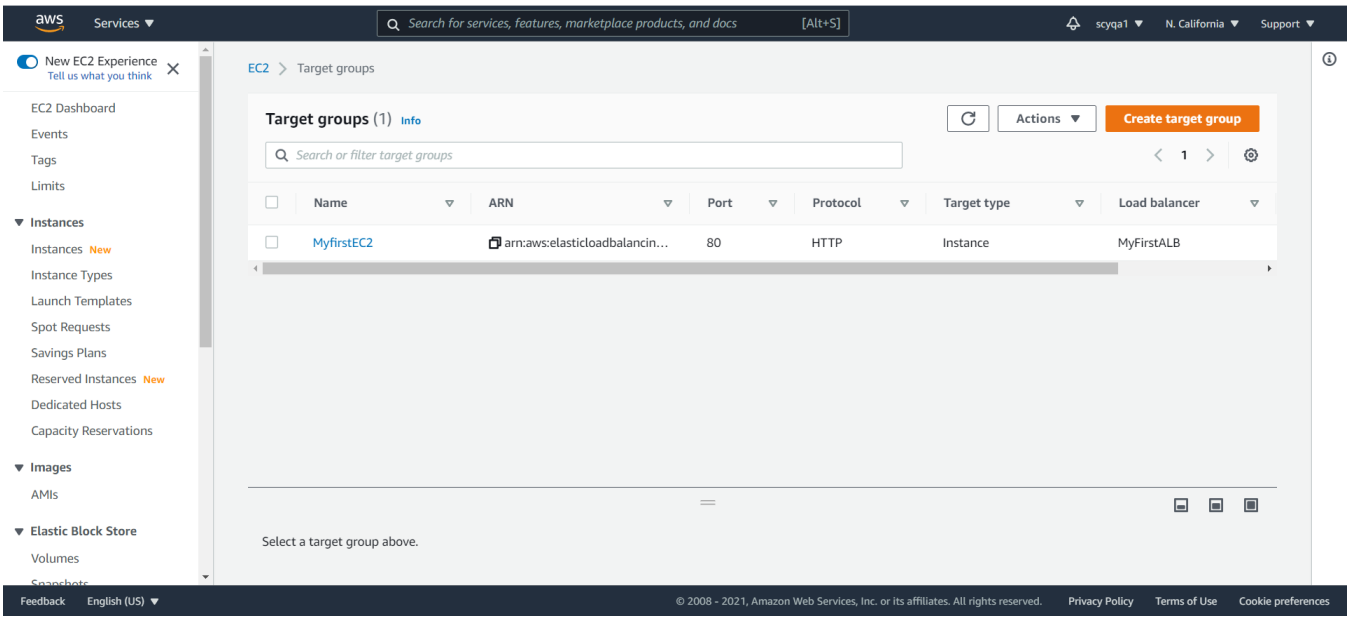
Step 1: create target group
Configure settings.



Register targets.

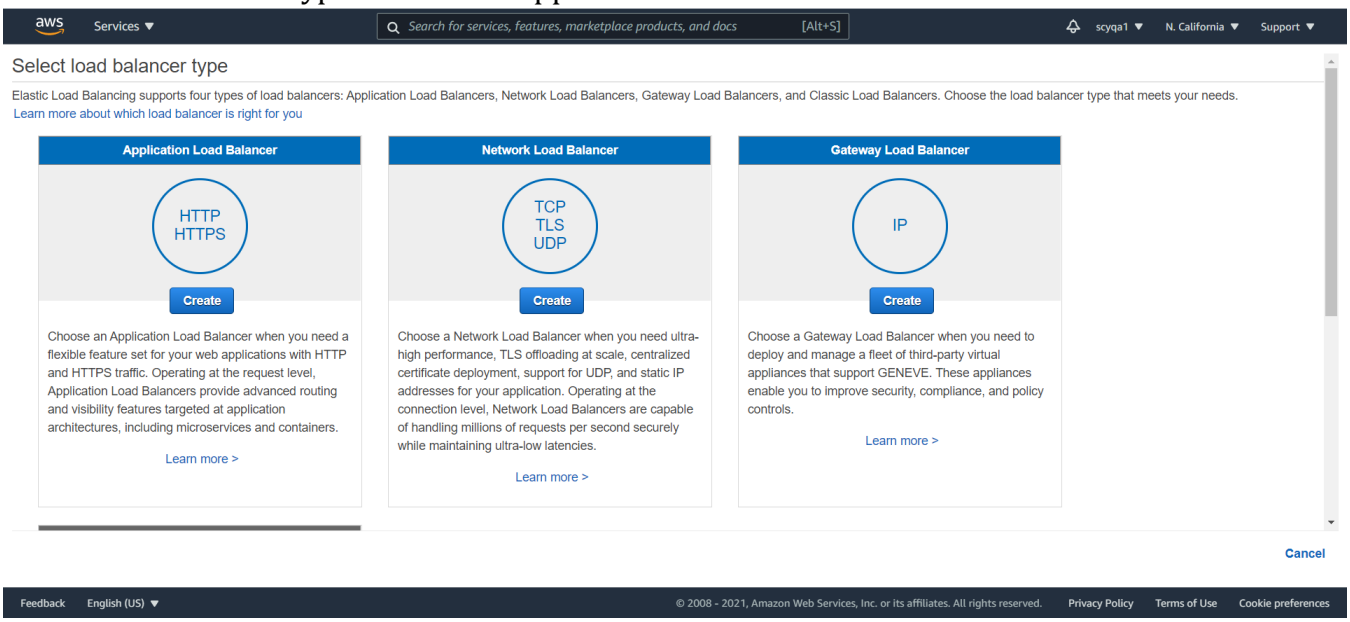


Result

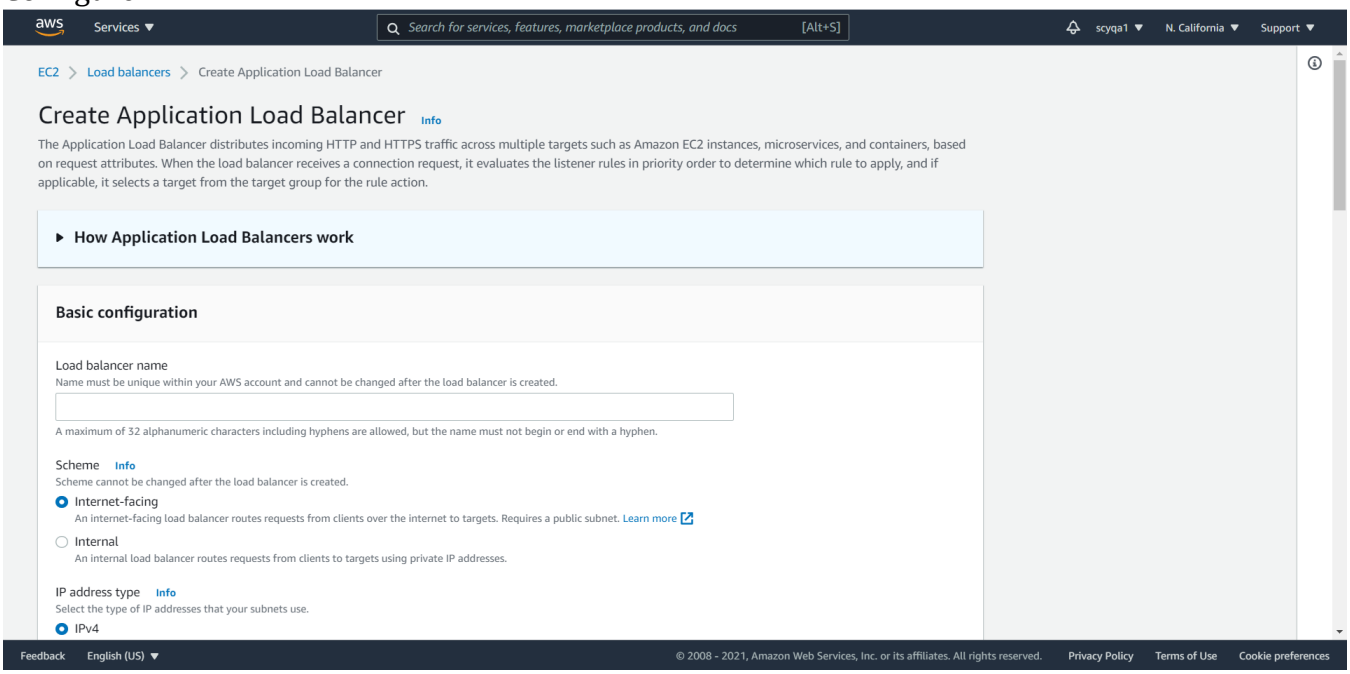


Step 2: create load balancer

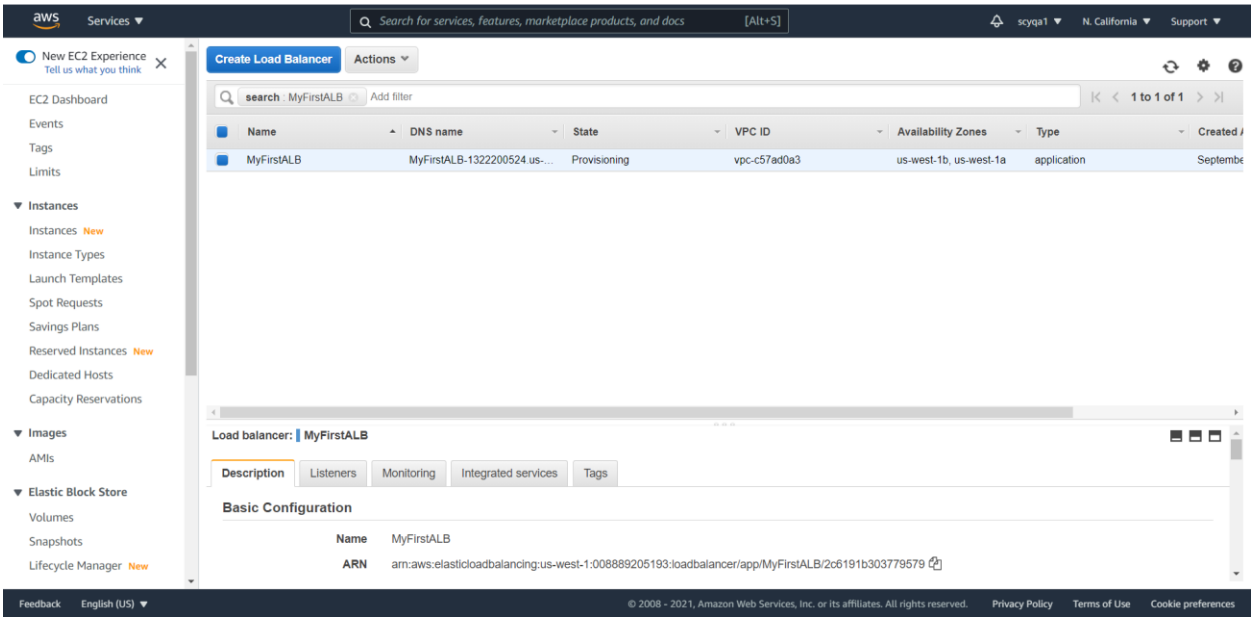
Create load balancer type. I select the application load balancer.



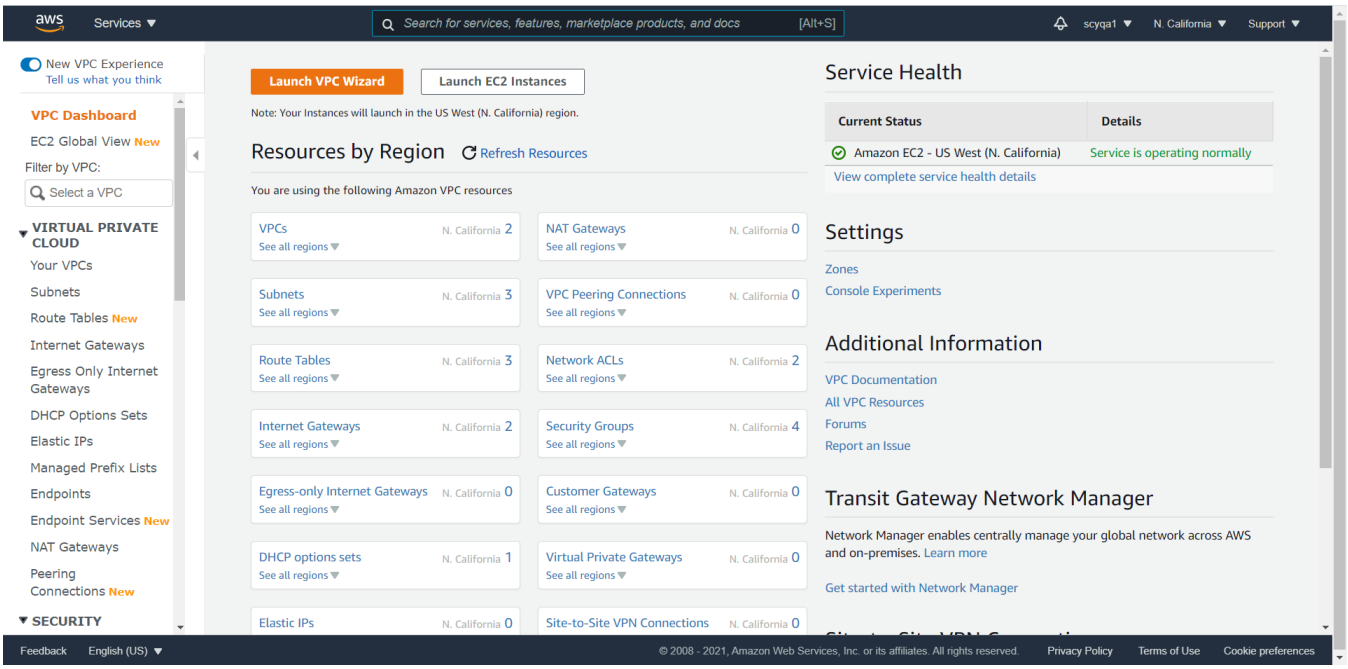
Configure.



Final result



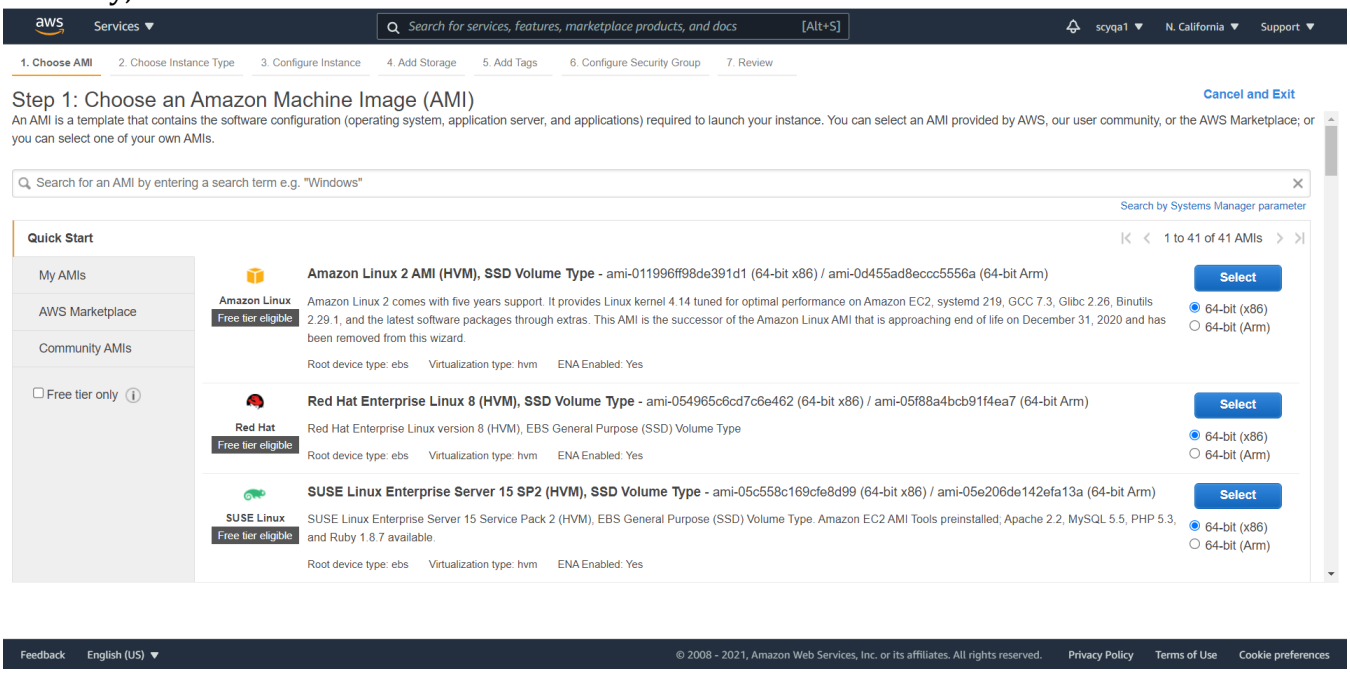
4)Create a VPC



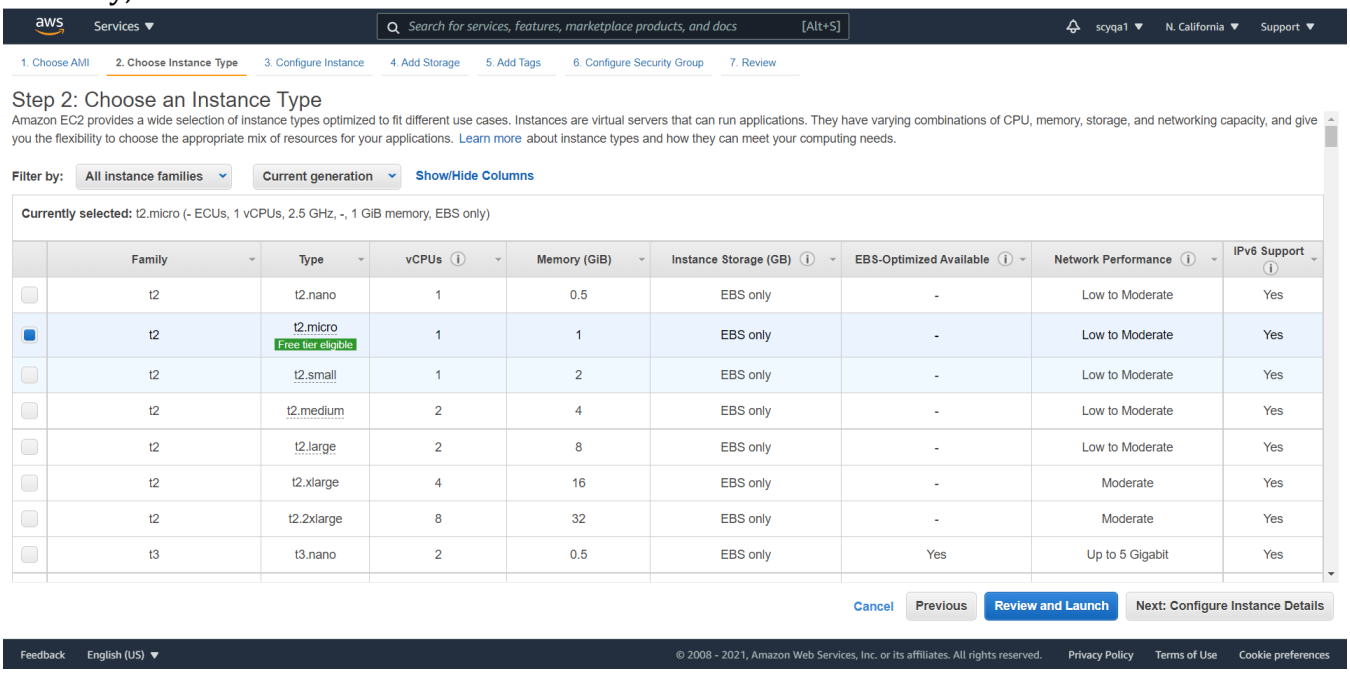
Launch EC2 instances

Step 1: choose AMI

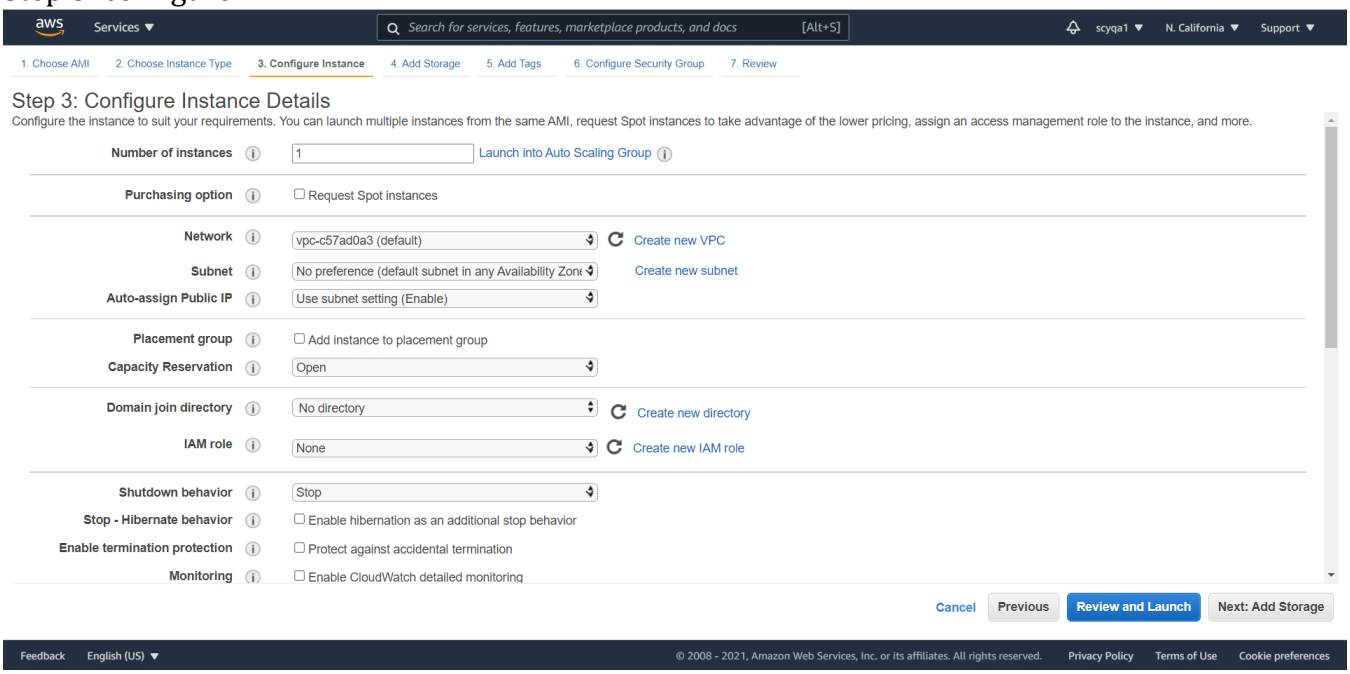
Similarly, I choose Amazon Linux 2 AMI.



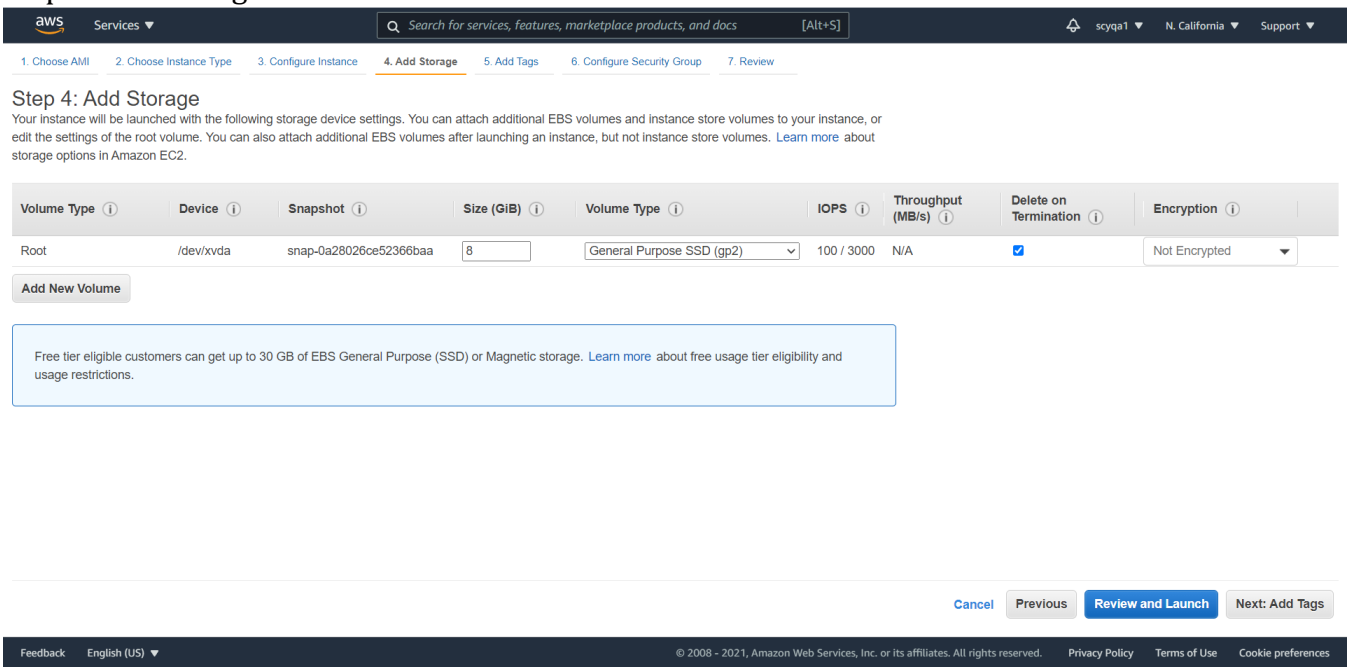
Step 2: choose an instance type
Similarly, I select t2.micro.



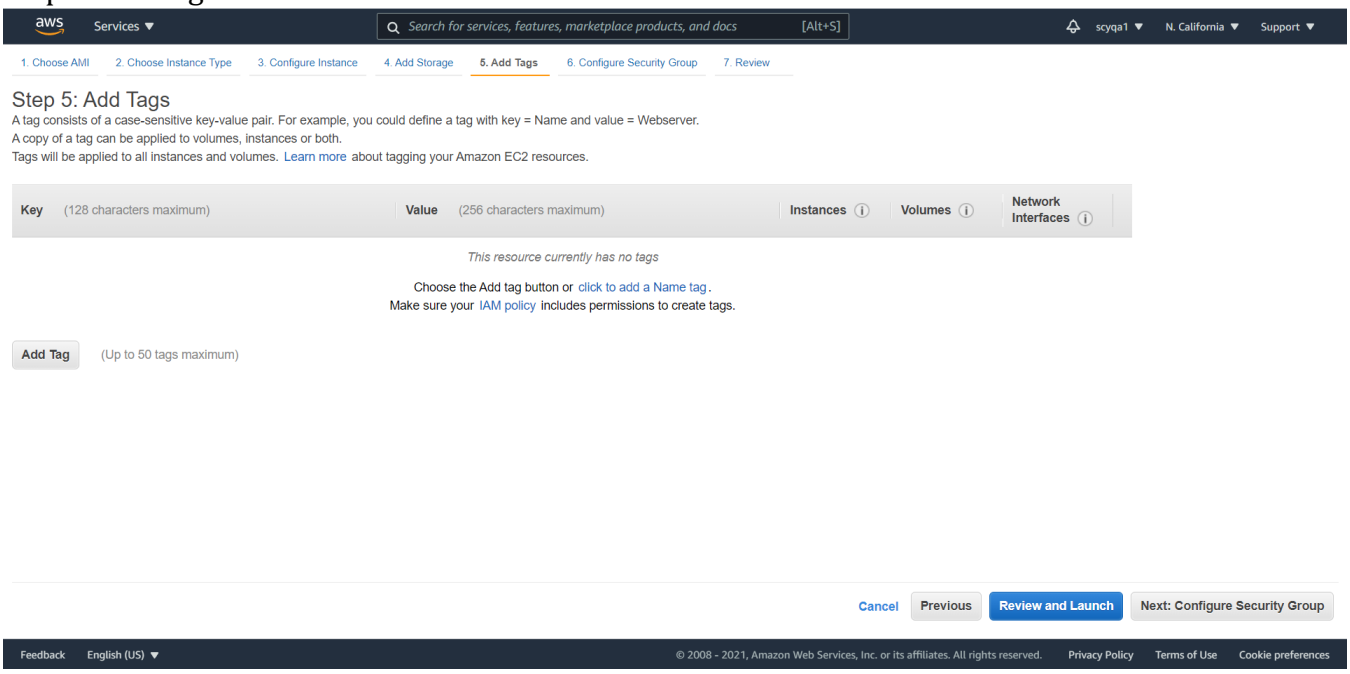
Step 3: configure



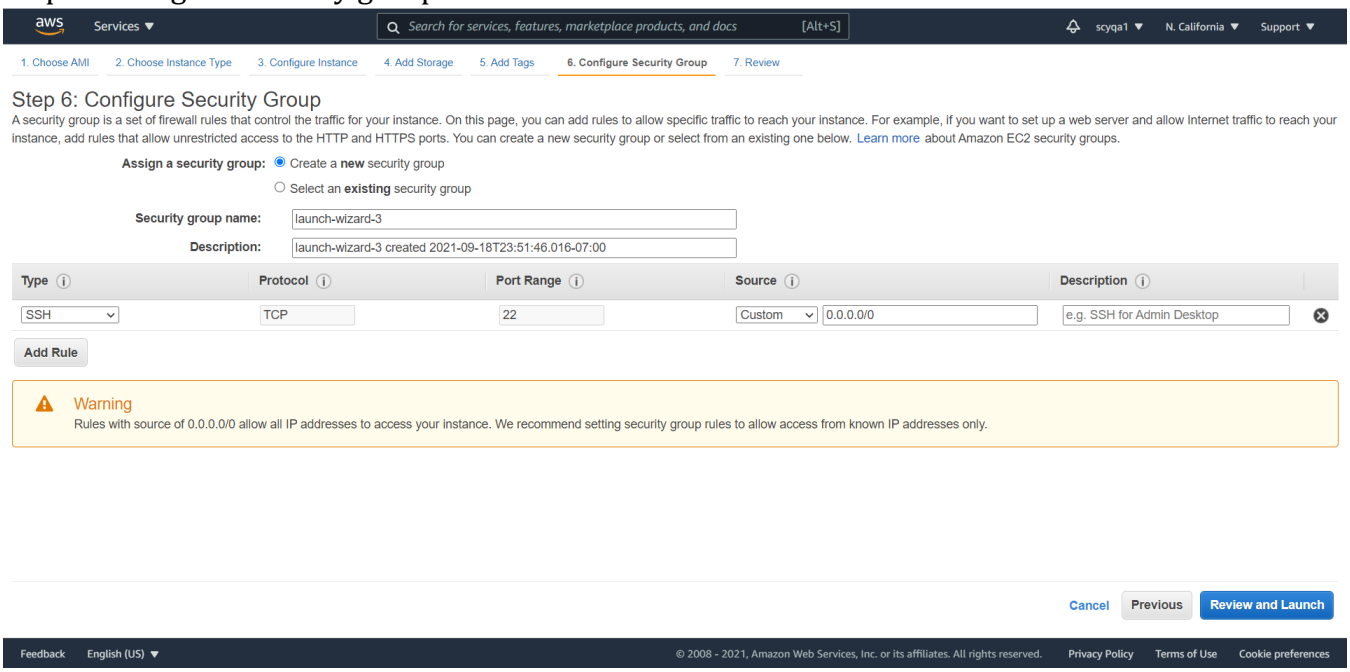
Step 4: add storage



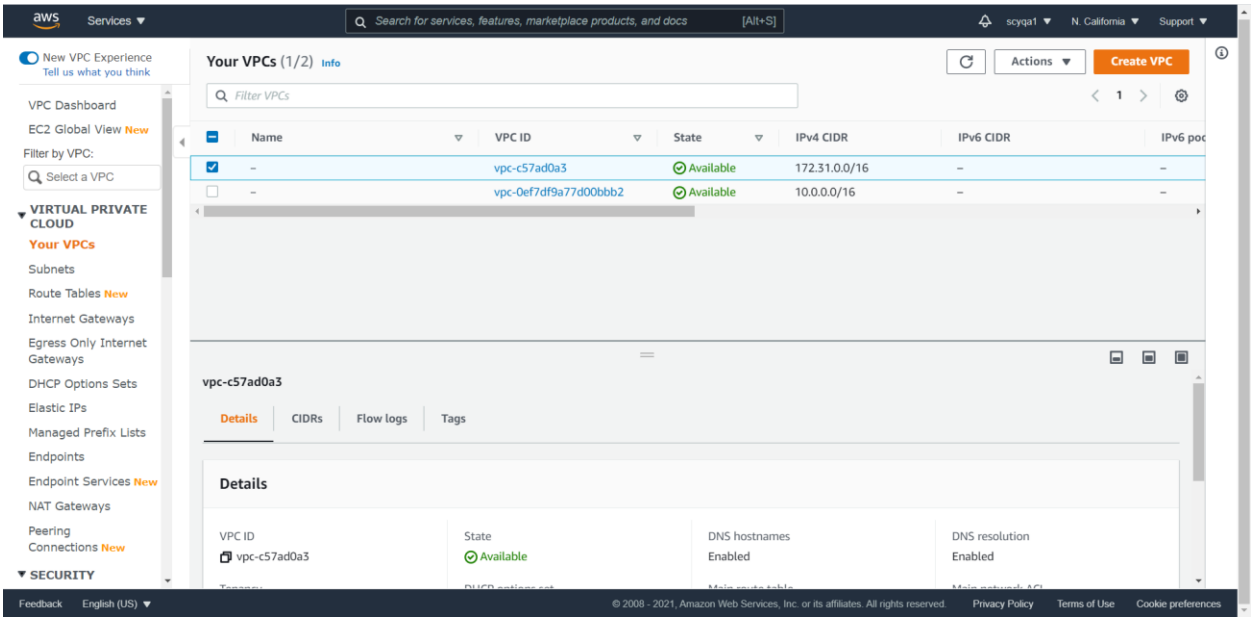
Step 5: add tags



Step 6: configure security group

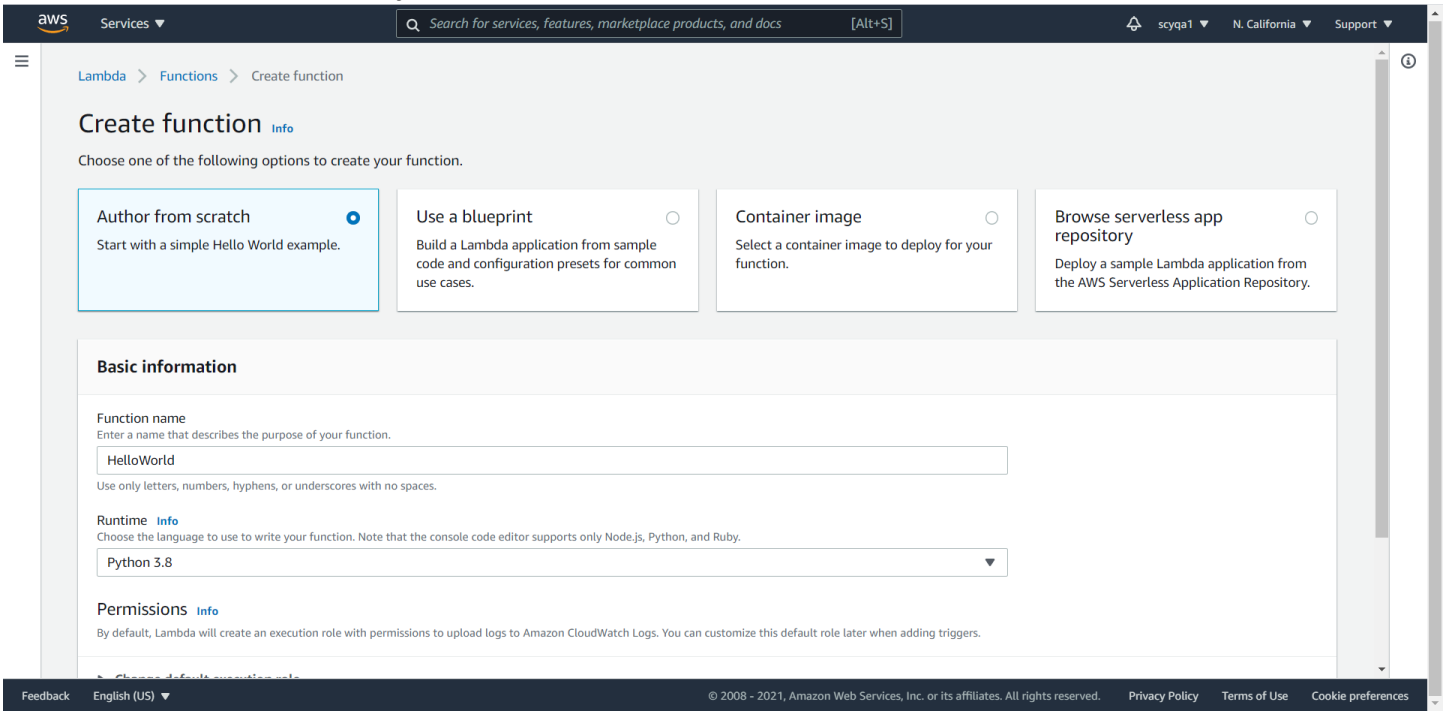


Final result

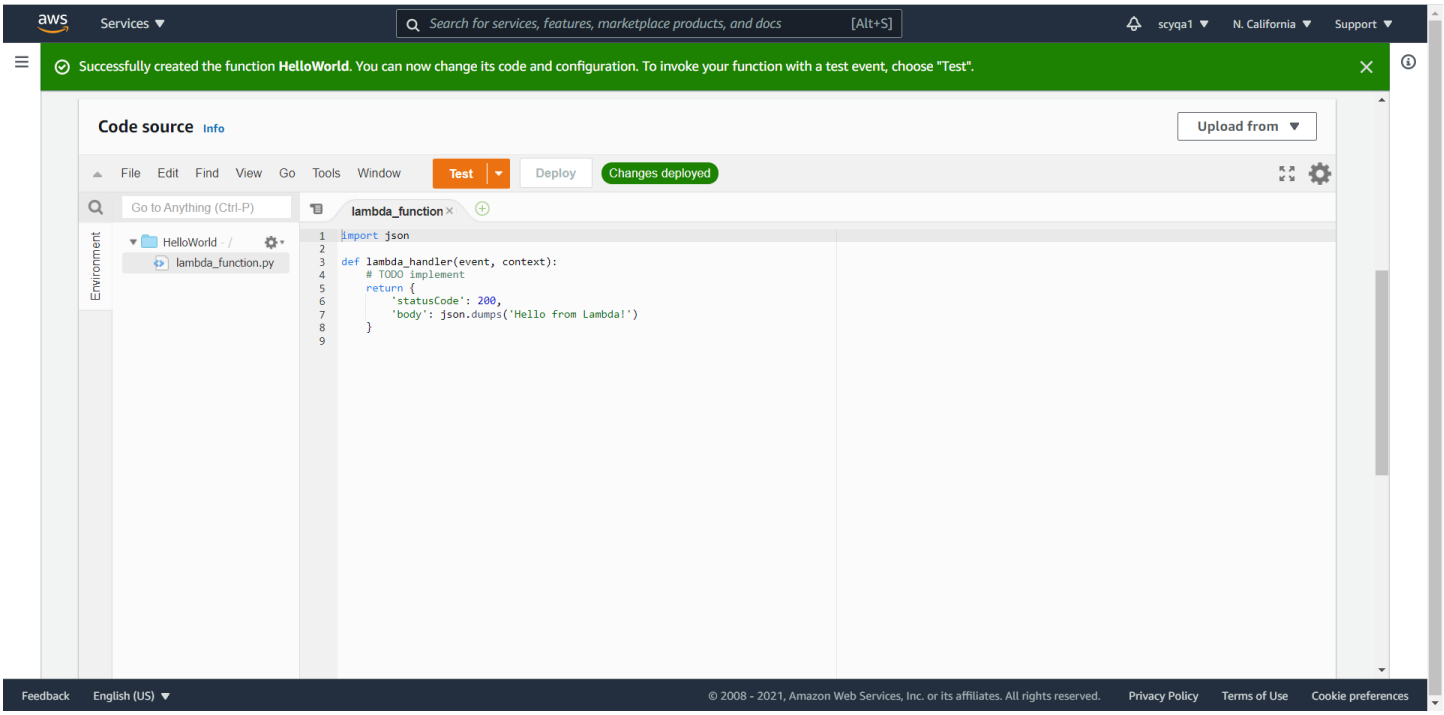


5)Write a ‘hello world program’ in Python using AWS

Create Lambda Function in Python

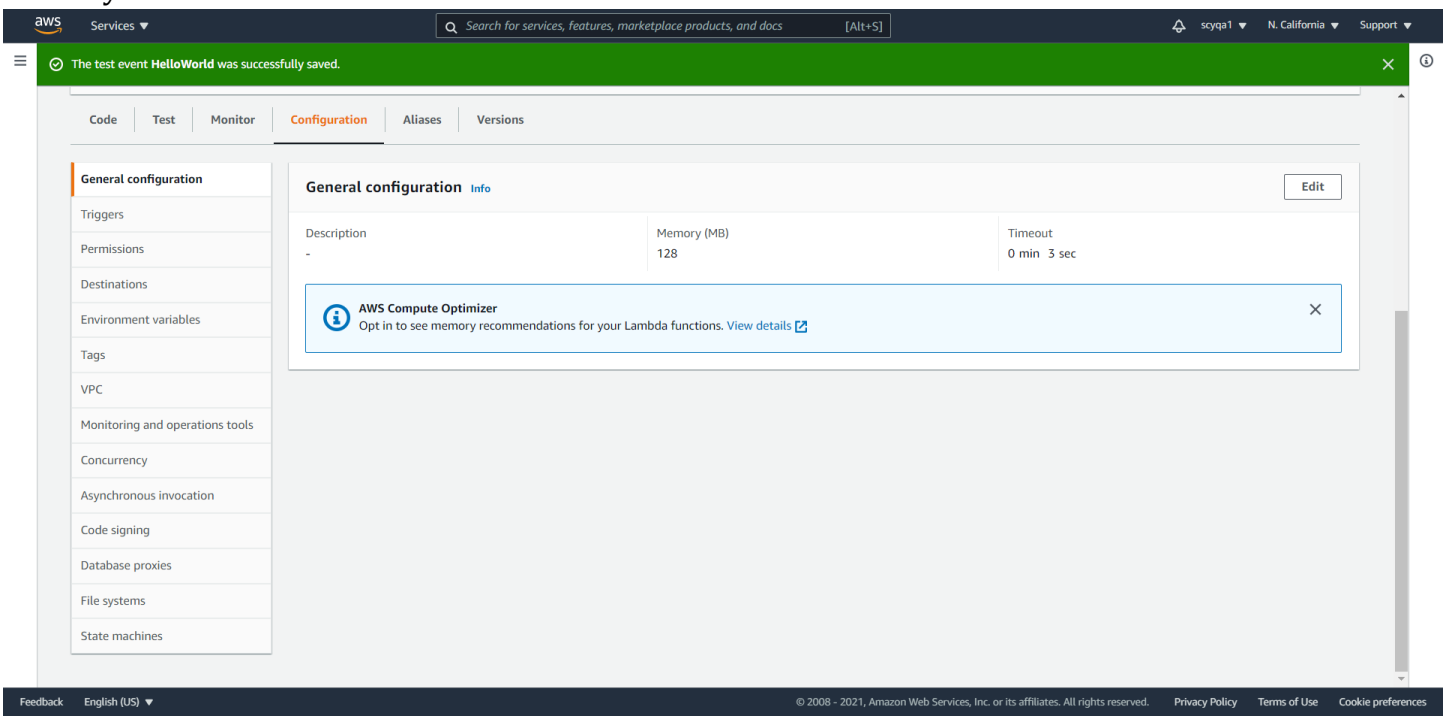


Choose Author from scratch and configure environment of the python 3.8

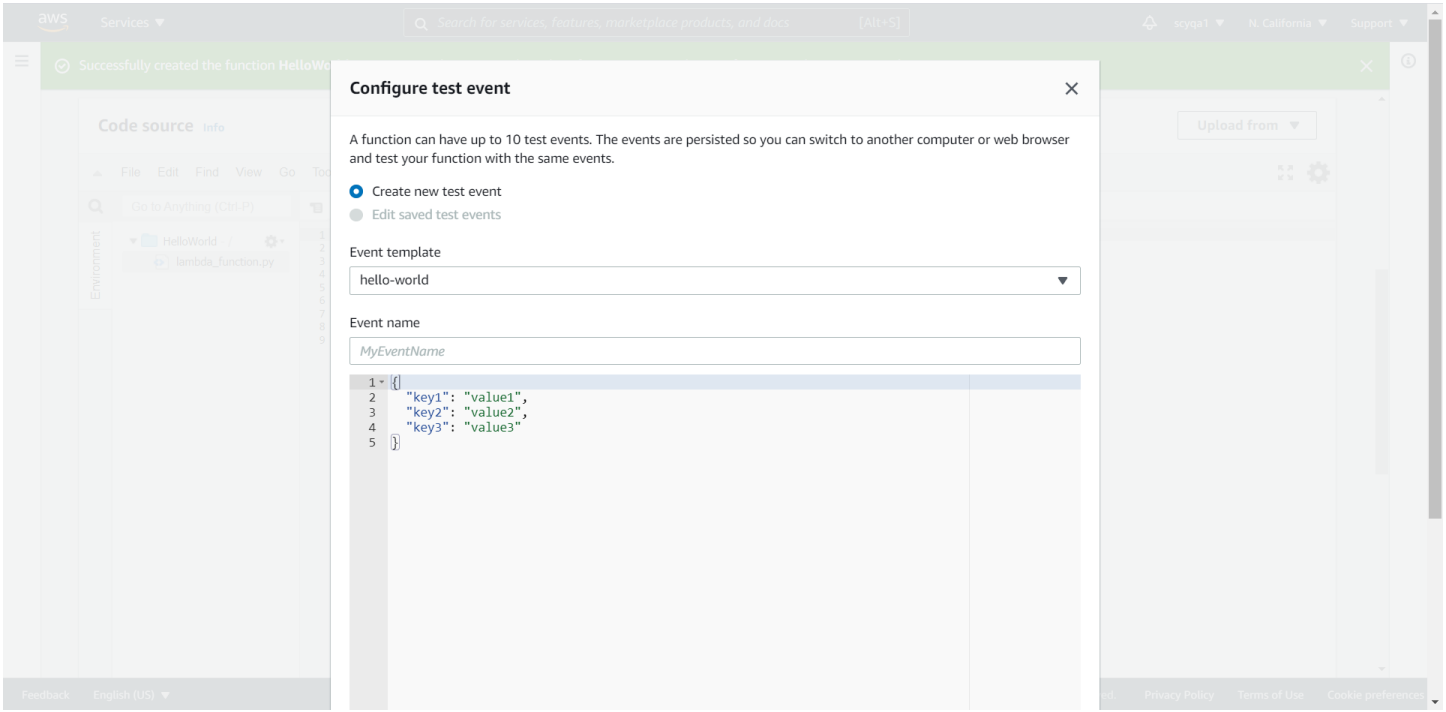


Configurations

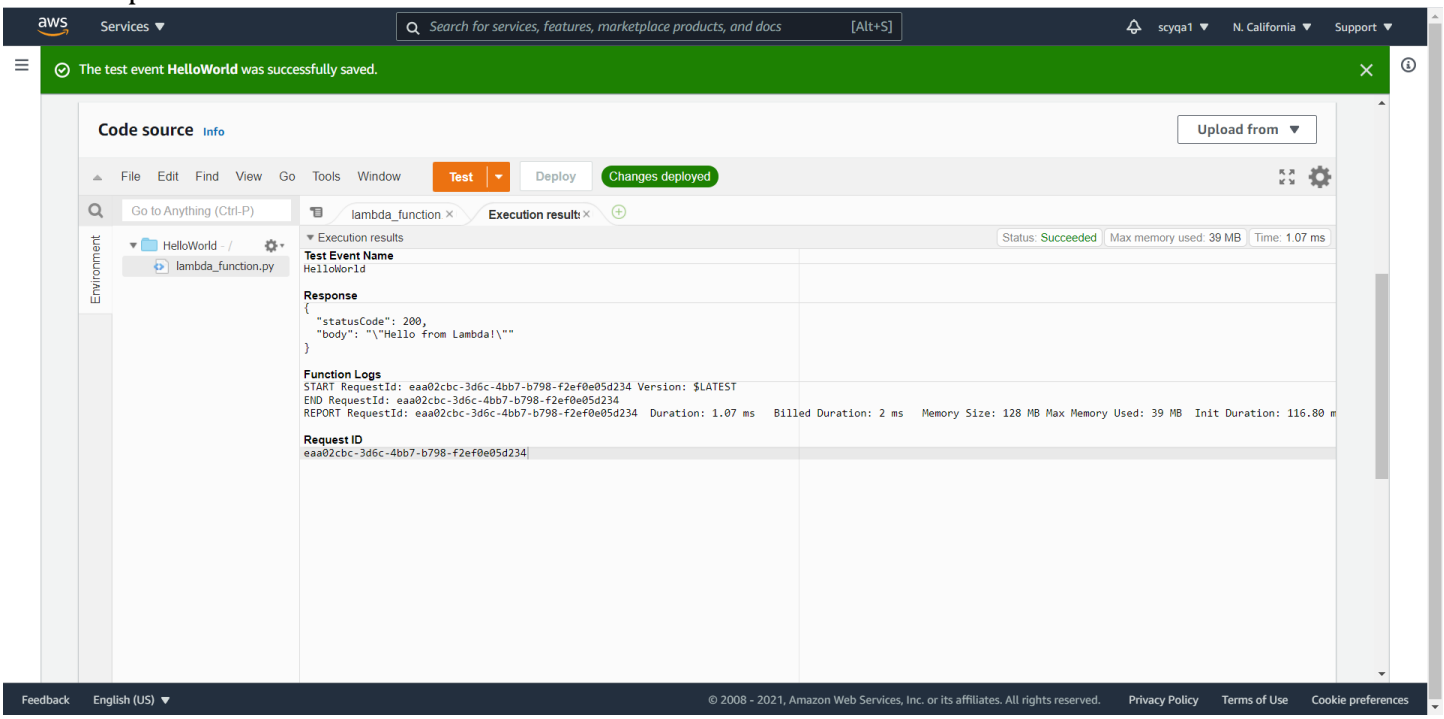
I can change the configurations for your Lambda function. For Example, set the request time out and memory.



Configure test event



Final Response:



Summary:

As an overview of this project, I have to say I put a lot of work into it, especially in the condition we have an assignment and a project in just one week. Through this project, for the part1, I learnt a lot about python coding for amazon web service using the library ‘boto3’. And for the part 2, I reviewed the work of first week, and learnt some techniques about Lambda on AWS. This helps with understanding and long-term memory. More specific details about such as AWS and EC2 VPC will be covered in the next section.

The Amazon Web Services Global Cloud Infrastructure offers cloud computing services globally through its secure, flexible, effective platform. One of the remarkable advantages of the AWS is to lower the cost needed to establish the up-front capital infrastructure, which allows users to spin up numbers of servers instantly. There are also some other benefits by AWS. Firstly, AWS provides a vast global cloud infrastructure for users to make innovations, do experiments and iterate computations quickly. It is easy to build new applications and adapt them to the required scales of the workload. Secondly, AWS provides different use of services such as development platform or programming model for different users, which enables users to choose corresponding modes and experience the flexibility according to their work demand. Moreover, AWS is also a durable and reliable technology with official certifications and audits. In order to guarantee the integrity and safety of user's data, the AWS system is built with multiple layers of operational and physical security.

Through a lot of programming, I understand what the AWS can serve better. For example, the instances, the tags, the operations of reboot, creation, stop, update and so on. Through the comprehensive application of them, I believe, in the future, I can cooperate with team members of the class could computing, my friends or colleagues to complete larger projects in a more efficient way. I'm grateful for the experience