CSCI 5380 – Network Virtualization and Orchestration

Lab 4
Amazon Web Services (AWS)

University of Colorado Boulder
Department of Computer Science
Network Engineering

Professor Levi Perigo, Ph.D.

Summary

Amazon Web Services (AWS) is a cloud-computing platform offered by Amazon. Cloud technology and tools are popular for network engineering, and the skills learned in this lab will enhance students' resumes in a desirable skillset in the current market. The AWS tools and skills used throughout this lab include EC2, security policies, S3 buckets, SNS, and Cloud Watch. Although these services could be managed using the AWS Management Console, AWS also offers a powerful tool (Software Development Kit) known as Boto3. This gives the power to manage the above-mentioned services for a large number of resources using simple Python scripts integrated with the Boto module.

Objectives

1. Learn about Amazon Web Services (AWS) tools
2. Lean how to deploy EC2 server instances
3. Learn how to deploy applications on those instances
4. Learn how to create security policies/firewall rules
5. Learn how to backup configurations into S3 buckets
6. Learn how to setup and use Simple Notification Service (SNS)
7. Learn how to setup Cloud Watch monitoring system
8. Manage AWS resources using Boto3 (Python based SDK)

Part 1

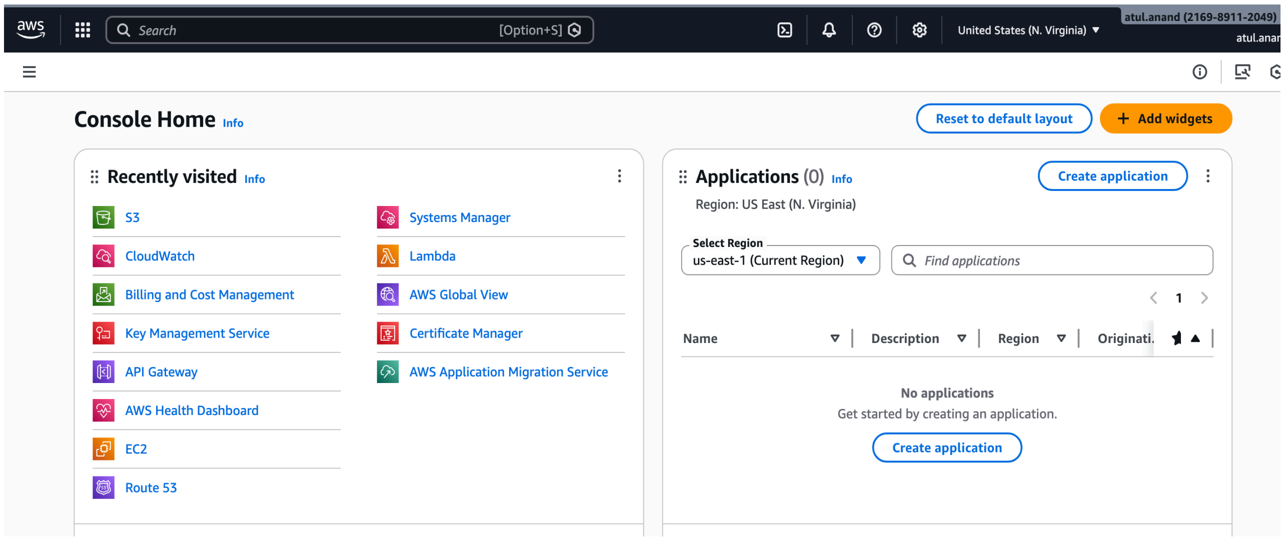
Objective 1.1 – Create an AWS account

In order to use Amazon Web Services, you must first create a user account. Amazon provides one year of free usage (limited resources).

1. Create an AWS Educate account at https://aws.amazon.com/education/awseducate/ to get AWS promotional credits. You can also use your amazon.com credentials to log in, but please ensure you create a student account to get the AWS credits.

[**NOTE:** You might need to provide your debit/credit card number for future billing]

2. After creating the account, you will be directed to AWS management console which presents the user with a number of AWS tools and services. Provide a screenshot of your management console. [**1 point**]



Objective 1.2 – Tweaking AWS settings

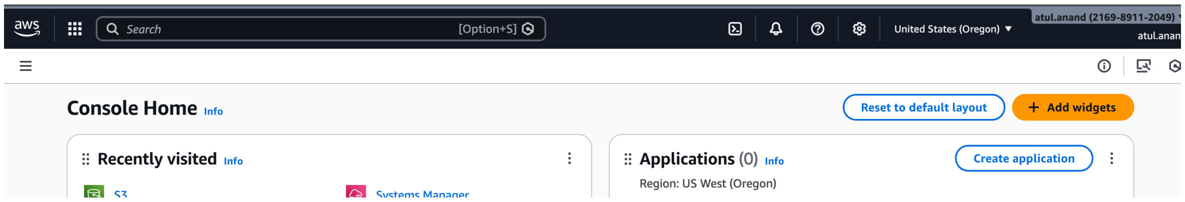
We shall now look at tweaking the AWS management settings to make efficient use of these resources.

1. We do not want Amazon to charge us for either using their premium services (Multi-Core CPU's, Terabytes of data, CDN services, etc.) or breaching their ONE-year free usage tier.

   To prevent extra charges on your account please shut down and remove all instances at the end of the lab.

   To ensure you get the most efficient performance, we need to change our region to any of the US West regions (N. California/Oregon). You can do this from the AWS management console. Provide a screenshot of the region you selected [**1 point**].

I selected the West-2 region - Oregon


United States (Oregon)

atul.anand (2169-8911-2049)

Applications (0)

Region: US West (Oregon)

2. Set up security credentials for your account: https://aws.amazon.com/premiumsupport/knowledge-center/create-access-key/ (Please ensure that you download and save the keys on your machine, as it will be required for further objectives).

Part 2

Objective 2.1 – Deploying EC2 Instances

1. What is EC2? [**5 points**]

EC2 stands for the elastic compute, and this is a service offered by the Amazon which provides on-demand virtual machines or instances of different configurations as per requirement and allows for dynamic scaling as per server load.

2. Briefly explain the below types of EC2 instances. [**15 points**]

- Spot instances

This type of instance allows users to use the unused cloud resources and usually offered at deep discounts, and this is ideal for tasks which are fault tolerant. These resources can be reclaimed by the AWS by providing a short 2 minute notice.

- Reserved instances

These types are instances are usually leased for longer term such as 3 years or such. This allows AWS for very predicable load and provides user with good discount due to upfront commitment for the longer term.

- Dedicated hosts

The dedicated host refers to the VPC running on dedicated physical host for single customer. This provides the isolation to host up to hardware level for customer.

- Dedicated instances

Dedicated instances are virtual machines which are deployed on shared hardware by one or multiple customers and does not offer any hardware level isolation. The virtual instance is just dedicated to the customer and share the underlaying infrastructure.

- Elastic GPUs

Elastic GPU provides way to accelerate graphic based performance for EC2 instances based on the workload requirement.

To deploy an EC2 instance, click on Services> EC2 > Launch Instance.

Step 1: Search for ubuntu and select Ubuntu Server 16.04 LTS 64-bit (x86) image. [Or you can select an image of your choice.]

Step 2: t2.micro (Free tier eligible)

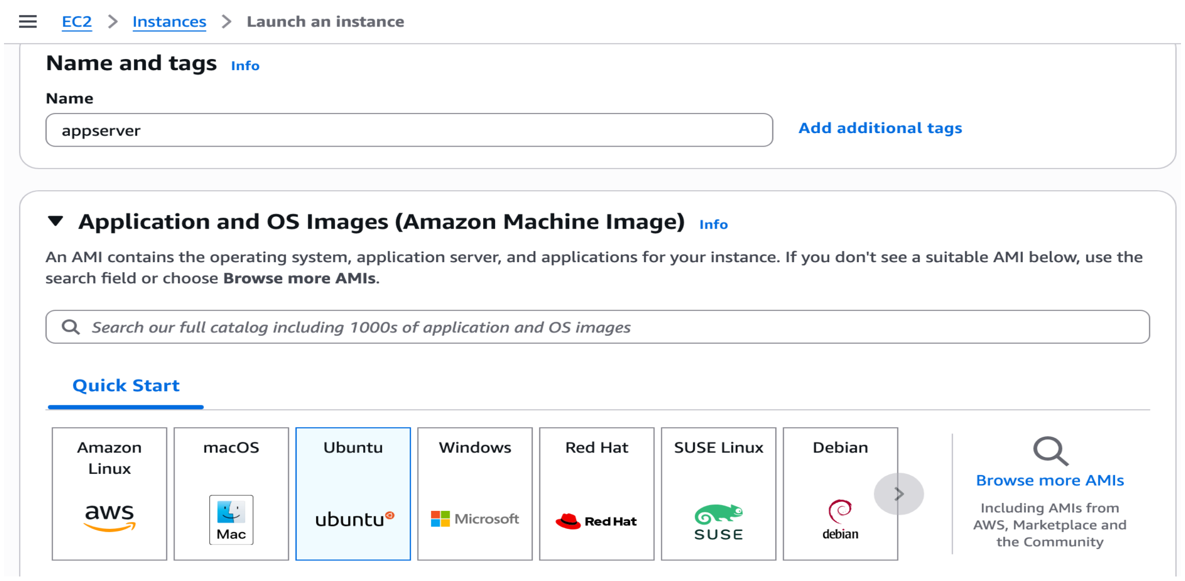
NOTE: Selecting anything else will result in OS costs.

Step 3: Deploy **TWO** instances.

Step 4: Add storage (8 GB is fine, can add more later)

Step 5: Add name as NVO-Lab

This uniquely identifies the instance and will come handy in a clustered environment. You can define something like a **“webserver”** if you are deploying a webserver or **“appserver”** for an application server, etc.


EC2 > Instances > Launch an instance

Name and tags Info

Name

appserver

Add additional tags

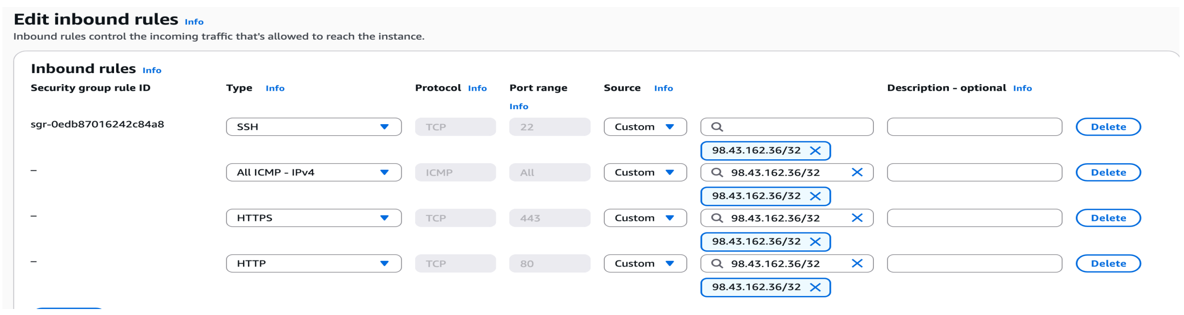
Application and OS Images (Amazon Machine Image) Info

An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose **Browse more AMIs**.

Search our full catalog including 1000s of application and OS images

Quick Start

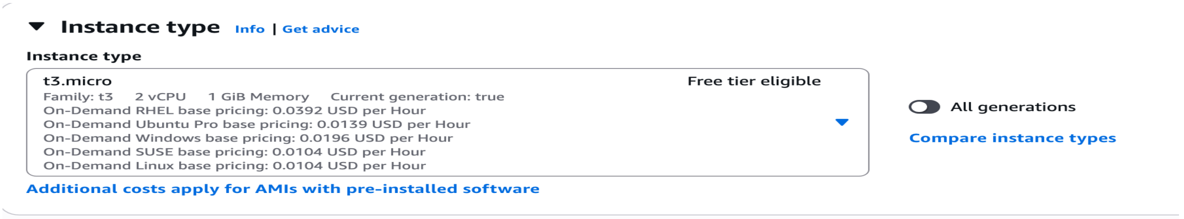
Step 6: Create a new security group and select the source to be your IP address for SSH rule. Add security rules allowing ICMP, HTTP and HTTPS traffic from your IP address.


Edit inbound rules Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules Info

Security group rule ID	Type	Protocol	Port range	Source		Description - optional	
sgr-0edb87016242c84a8	SSH	TCP	22	Custom	98.43.162.36/32		Delete
–	All ICMP - IPv4	ICMP	All	Custom	98.43.162.36/32		Delete
–	HTTPS	TCP	443	Custom	98.43.162.36/32		Delete
–	HTTP	TCP	80	Custom	98.43.162.36/32		Delete

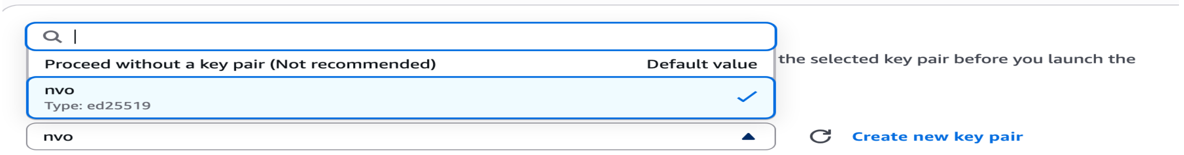

Instance type Info | Get advice

Instance type

t3.micro Free tier eligible

Family: t3 2 vCPU 1 GiB Memory Current generation: true
On-Demand RHEL base pricing: 0.0392 USD per Hour
On-Demand Ubuntu Pro base pricing: 0.0139 USD per Hour
On-Demand Windows base pricing: 0.0196 USD per Hour
On-Demand SUSE base pricing: 0.0104 USD per Hour
On-Demand Linux base pricing: 0.0104 USD per Hour

Additional costs apply for AMIs with pre-installed software

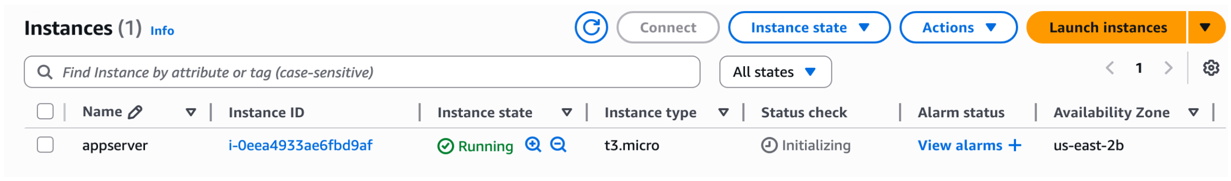

Proceed without a key pair (Not recommended) Default value

nvo
Type: ed25519

nvo

the selected key pair before you launch the

Step 7: Review the configuration parameters before launching the instance. Create a new key-pair and click on download. [**NOTE:** If you miss this step, you will have to repeat the whole process again.] After the download completes, click on the **"Launch"** button.



Instances (1) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
appserver	i-0eea4933ae6fbd9af	Running	t3.micro	Initializing	View alarms	us-east-2b

3. Explain how an AMI is related to the instance. [**3 points**]

AMI (Amazon Machine Image) is snapshot of the OS being deployed on the EC2 and in this case we deployed UBUNTU, so we used an AMI to deploy the EC2 instance – Its is like ISO file when we install the OS on physical servers.

4. What are the disadvantages of allowing SSH traffic into the server from anywhere? Is this a security concern? [**5 points**]

Since, this instance we deployed is reachable on internet and if we allow SSH traffic from anywhere from internet, it will cause DOS attack and attackers may try to gain unauthorized access to the VM's. This may also affect CPU utilization which may cause a unexpected bill increase.

5. What are some best practices one should follow to secure the north-south and east-west traffic in the cloud? [**5 points**]

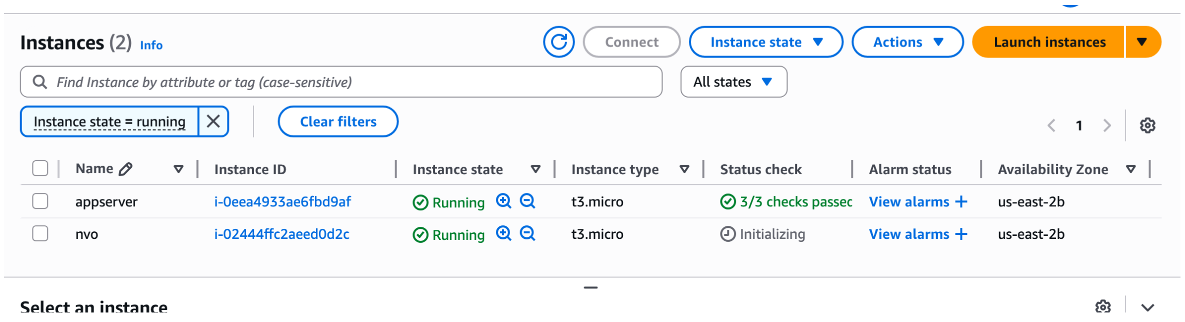
North/South traffic –

- Strict control over ports and service being exposed
- Stricter perimeter and zero trust approach
- NGFW and web-application firewall to ensure filtering out unwanted traffic or attempts
- Limiting the access to bare minimum IP address.

East-West Traffic –

- Zero Trust system to ensure internal traffic are not allow attackers to gain un-interrupted access
- Segmentation of services to allow for containing any breaches
- Limited to minimum access to users for services.

6. Paste a screenshot showing the running instances. [**10 points**]



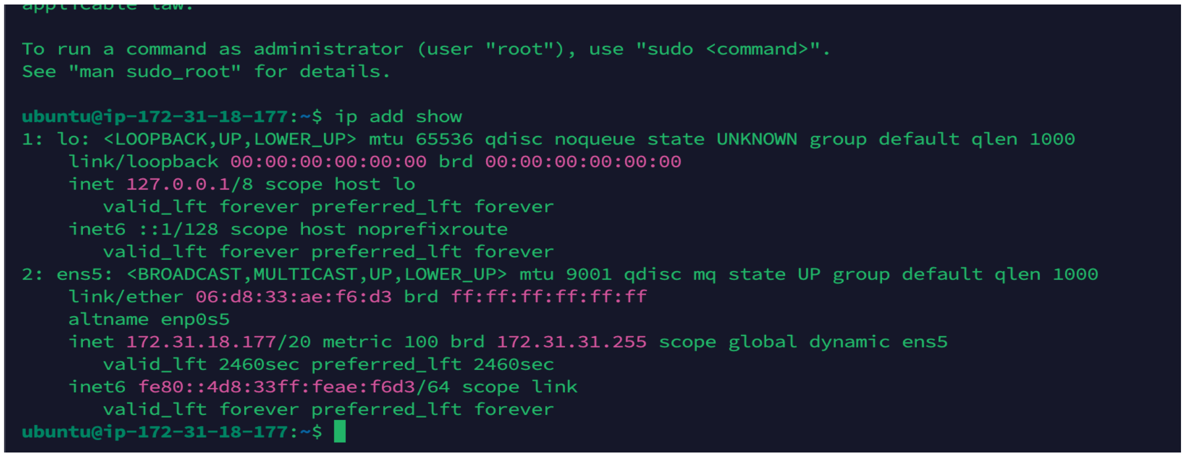
Instances (2) Info

Instance state = running

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
appserver	i-0eea4933ae6fbd9af	Running	t3.micro	3/3 checks passec	View alarms	us-east-2b
nvo	i-02444ffc2aeed0d2c	Running	t3.micro	Initializing	View alarms	us-east-2b

Select an instance

7. Select an instance and click on connect. Follow the instructions and SSH into one of the EC2 instances. Paste a screenshot showing the bash prompt. [Login for the Ubuntu instance is: ubuntu] [**5 points**]



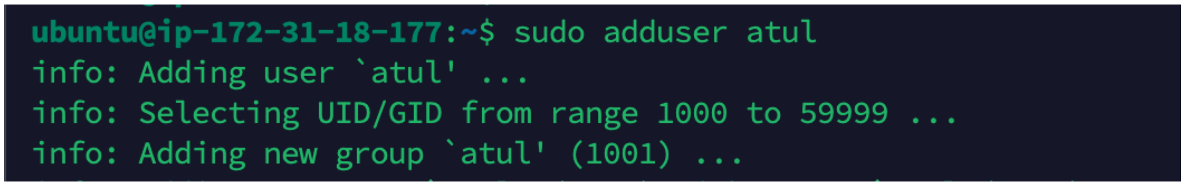
```
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-18-177:~$ ip add show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: ens5: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc mq state UP group default qlen 1000
    link/ether 06:d8:33:ae:f6:d3 brd ff:ff:ff:ff:ff:ff
    altname enp0s5
    inet 172.31.18.177/20 metric 100 brd 172.31.31.255 scope global dynamic ens5
       valid_lft 2460sec preferred_lft 2460sec
    inet6 fe80::4d8:33ff:feae:f6d3/64 scope link
       valid_lft forever preferred_lft forever
ubuntu@ip-172-31-18-177:~$
```

8. What IP address did you SSH to? And what is the IP address on the interface of the instance? Explain the flow of traffic from your laptop to the EC2 instance. [**5 points**]

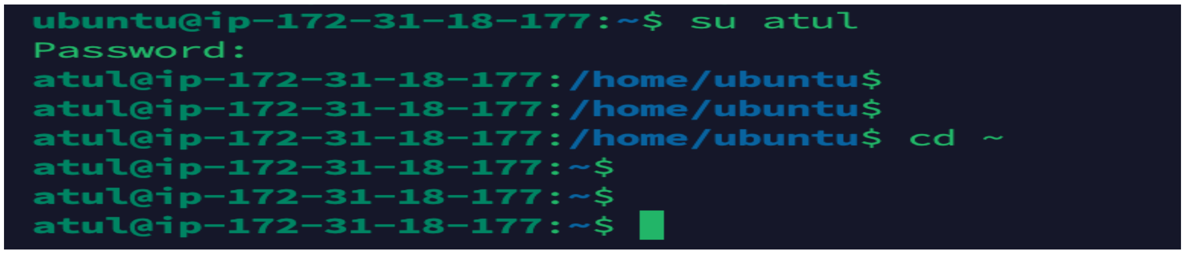
I used the IP: 18.119.98.212 which is public IP address, and I see a 172 range IP on the ubuntu machine where I logged in. Since, I allows the SSH port 22 from my public IP address, My SSH request was routed via my ISP to the AWS Edge and then AWS must be using some sort of NAT service to translate this public request to private instances.

9. Create another user in your instance with the username being your name and enable SSH with password for just your user. Explain how you achieved this. [**3 points**]

I used following command to create the user –

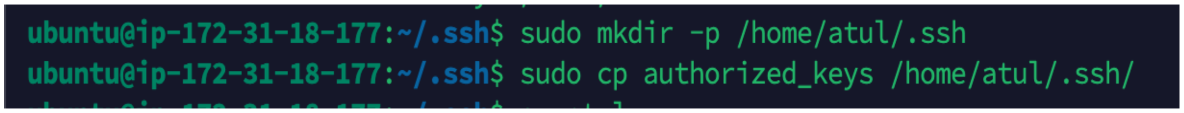


```
ubuntu@ip-172-31-18-177:~$ sudo adduser atul
info: Adding user `atul' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `atul' (1001) ...
```



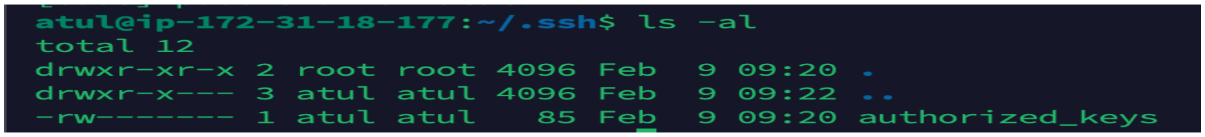
```
ubuntu@ip-172-31-18-177:~$ su atul
Password:
atul@ip-172-31-18-177:/home/ubuntu$
atul@ip-172-31-18-177:/home/ubuntu$
atul@ip-172-31-18-177:/home/ubuntu$ cd ~
atul@ip-172-31-18-177:~$
atul@ip-172-31-18-177:~$
atul@ip-172-31-18-177:~$
```

Since, this is public facing server, I will enable the key based authentication which we created when creating the instance for my user as well.

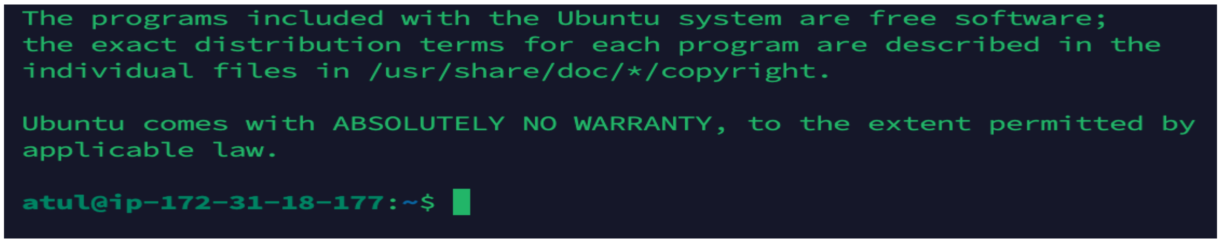


```
ubuntu@ip-172-31-18-177:~/.ssh$ sudo mkdir -p /home/atul/.ssh
ubuntu@ip-172-31-18-177:~/.ssh$ sudo cp authorized_keys /home/atul/.ssh/
```

Once the keys were saved under user .ssh directory – I was able to login using SSH- -



```
atul@ip-172-31-18-177:~/.ssh$ ls -al
total 12
drwxr-xr-x 2 root root 4096 Feb  9 09:20 .
drwxr-x--- 3 atul atul 4096 Feb  9 09:22 ..
-rw------- 1 atul atul   85 Feb  9 09:20 authorized_keys
```

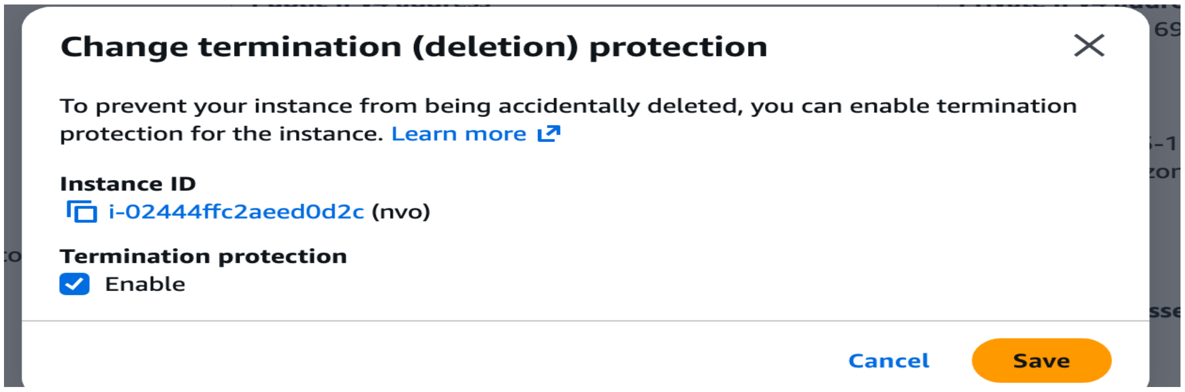


```
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

atul@ip-172-31-18-177:~$
```

10. What is the difference between stopping and terminating an instance? [**2 points**]

Stopping an instance means temporary shutdown and it can be resumed when needed but termination means permanently deleting an instance which can be recovered once completed.

11. Protect the second instance that you created from unauthorized termination. How did you achieve this? [**2 points**]

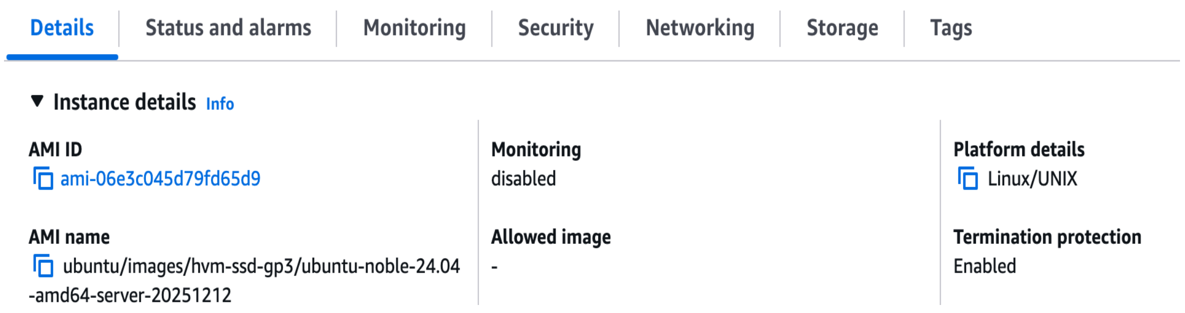


Change termination (deletion) protection

To prevent your instance from being accidentally deleted, you can enable termination protection for the instance. Learn more

Instance ID
i-02444ffc2aeed0d2c (nvo)

Termination protection
☑ Enable



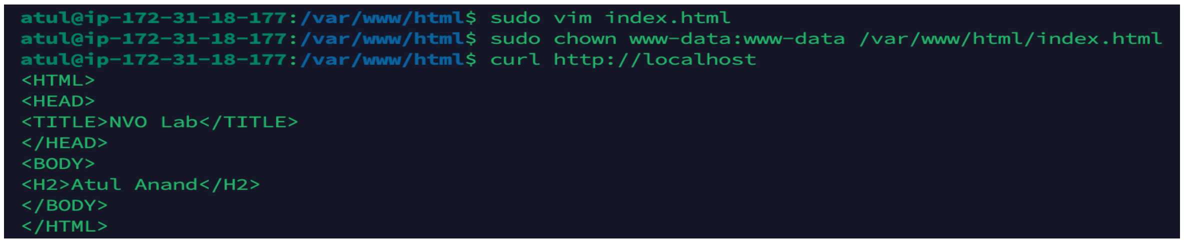
AMI ID ami-06e3c045d79fd65d9	Monitoring disabled	Platform details Linux/UNIX
AMI name ubuntu/images/hvm-ssd-gp3/ubuntu-noble-24.04-amd64-server-20251212	Allowed image -	Termination protection Enabled

Objective 2.2 – Deploying an application on your instance

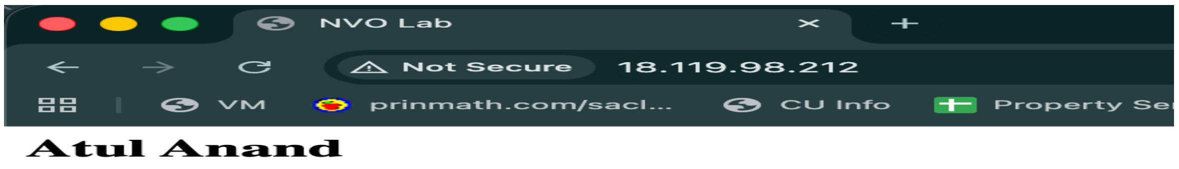
Install Apache web server on one of the instances. Create an index.html file in the /var/www/html/ directory which displays your name and 'NVO Lab'. You can use the below sample.

```
<HTML>
<HEAD>
<TITLE>NVO Lab</TITLE>
</HEAD>
<BODY>
<H2>Your_Name</H2>
</BODY>
</HTML>
```

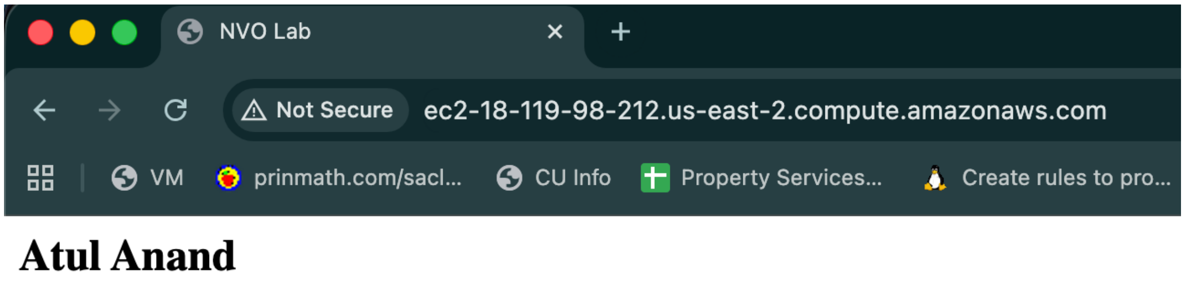
How do you access this webpage hosted on the instance from your laptop? Paste relevant screenshots. Can you access it using an IP address or DNS name or both? [**20 points**]



```
atul@ip-172-31-18-177:/var/www/html$ sudo vim index.html
atul@ip-172-31-18-177:/var/www/html$ sudo chown www-data:www-data /var/www/html/index.html
atul@ip-172-31-18-177:/var/www/html$ curl http://localhost
<HTML>
<HEAD>
<TITLE>NVO Lab</TITLE>
</HEAD>
<BODY>
<H2>Atul Anand</H2>
</BODY>
</HTML>
```



Atul Anand



Atul Anand

Part 3

Objective 3.1 – Deploying S3 Backups

NOTE: It is preferred to use the course VM provided to complete this objective.

1. What is S3? [**5 points**]

S3 is amazon's storage service which is designed to store data and offer very low latency and high availability architecture.

2. You shall now backup some of our router configurations to the cloud. Before you proceed, ensure you have configuration files of your routers stored in a separate folder (Eg. /home/nvo/routerConfigs)

   [**NOTE:** If you do not have configuration files present on your system, feel free to use other files (images, files, text,etc.)]

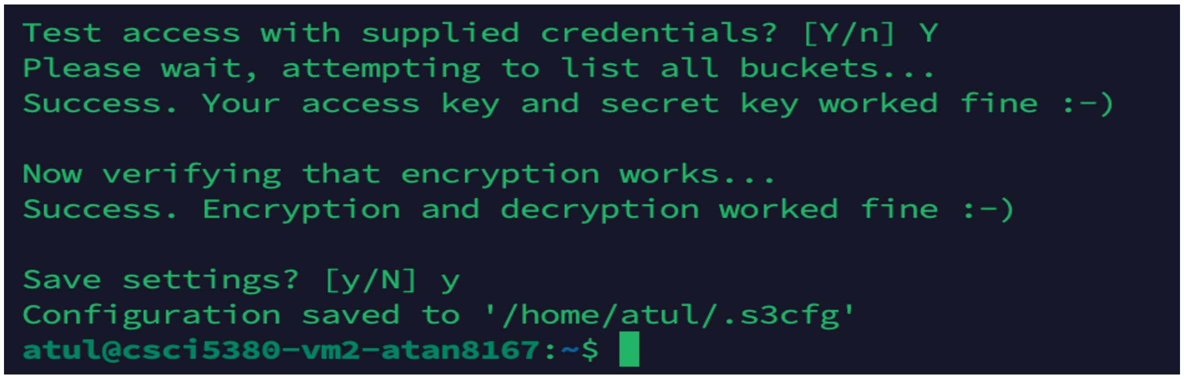
3. Update your VM and install S3 command-line utility

   (VM)# sudo apt-get install s3cmd -y

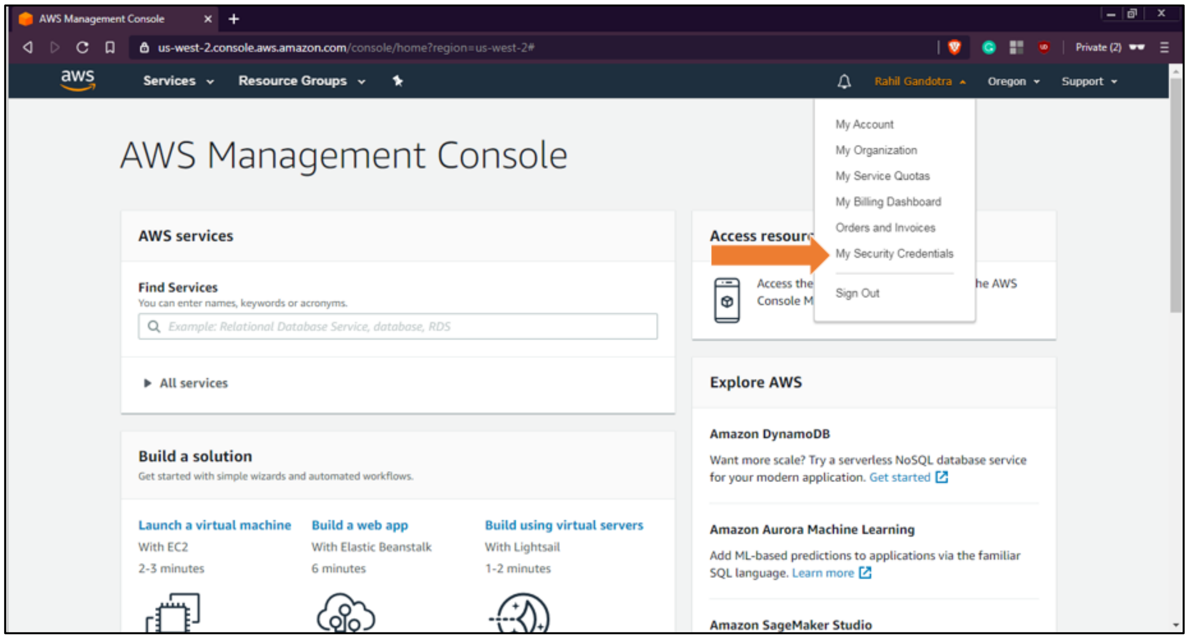
4. Configure S3 parameters by entering your AWS access/secret credentials and enter the encryption password (Your choice) when prompted. Save the settings.

   (VM)# s3cmd –configure

   You can create/find an AWS access/secret key on your "username" > Security Credentials > Continue > Access Keys > Create/Use



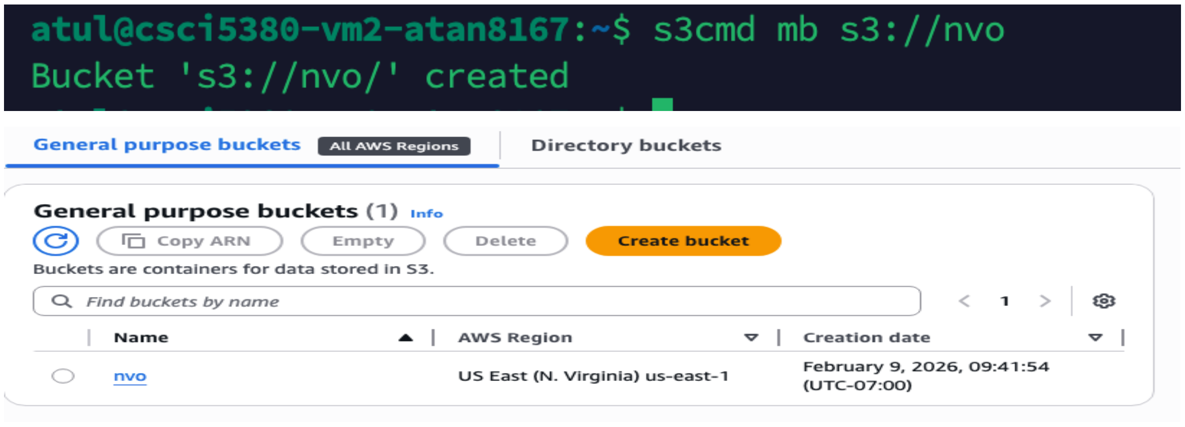
```
Test access with supplied credentials? [Y/n] Y
Please wait, attempting to list all buckets...
Success. Your access key and secret key worked fine :-)

Now verifying that encryption works...
Success. Encryption and decryption worked fine :-)

Save settings? [y/N] y
Configuration saved to '/home/atul/.s3cfg'
atul@csci5380-vm2-atan8167:~$
```


AWS Management Console
us-west-2.console.aws.amazon.com/console/home?region=us-west-2#
Services
Resource Groups
Rahil Gandotra
Oregon
Support
My Account
My Organization
My Service Quotas
My Billing Dashboard
Orders and Invoices
My Security Credentials
Sign Out
AWS services
Find Services
You can enter names, keywords or acronyms.
Example: Relational Database Service, database, RDS
All services
Build a solution
Get started with simple wizards and automated workflows.
Launch a virtual machine
With EC2
2-3 minutes
Build a web app
With Elastic Beanstalk
6 minutes
Build using virtual servers
With Lightsail
1-2 minutes
Access resour
Explore AWS
Amazon DynamoDB
Want more scale? Try a serverless NoSQL database service for your modern application. Get started
Amazon Aurora Machine Learning
Add ML-based predictions to applications via the familiar SQL language. Learn more
Amazon SageMaker Studio

5. Create an S3 bucket

(VM)# s3cmd mb s3://<S3_BUCKET_NAME_CREATED>

Confirm by issuing **"s3cmd ls"**. You can also check using the AWS Management Console. S3 is located under Storage and Content Delivery.



```
atul@csci5380-vm2-atan8167:~$ s3cmd mb s3://nvo
Bucket 's3://nvo/' created
```

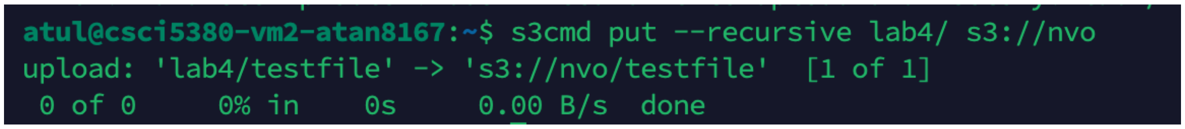
Buckets are containers for data stored in S3.

Name	AWS Region	Creation date
nvo	US East (N. Virginia) us-east-1	February 9, 2026, 09:41:54 (UTC-07:00)

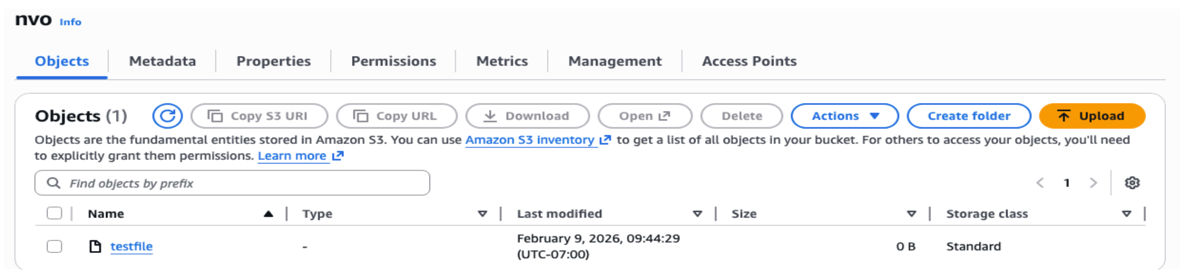
6. Push the configuration folder into the bucket

(VM)# s3cmd put <PATH_TO_LOCAL_CONFIG_FOLDER> s3://<S3_BUCKET_NAME_CREATED>

Verify that the files are updated. Paste a screenshot showing the same. [**10 points**]



```
atul@csci5380-vm2-atan8167:~$ s3cmd put --recursive lab4/ s3://nvo
upload: 'lab4/testfile' -> 's3://nvo/testfile'  [1 of 1]
 0 of 0     0% in    0s     0.00 B/s  done
```



Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

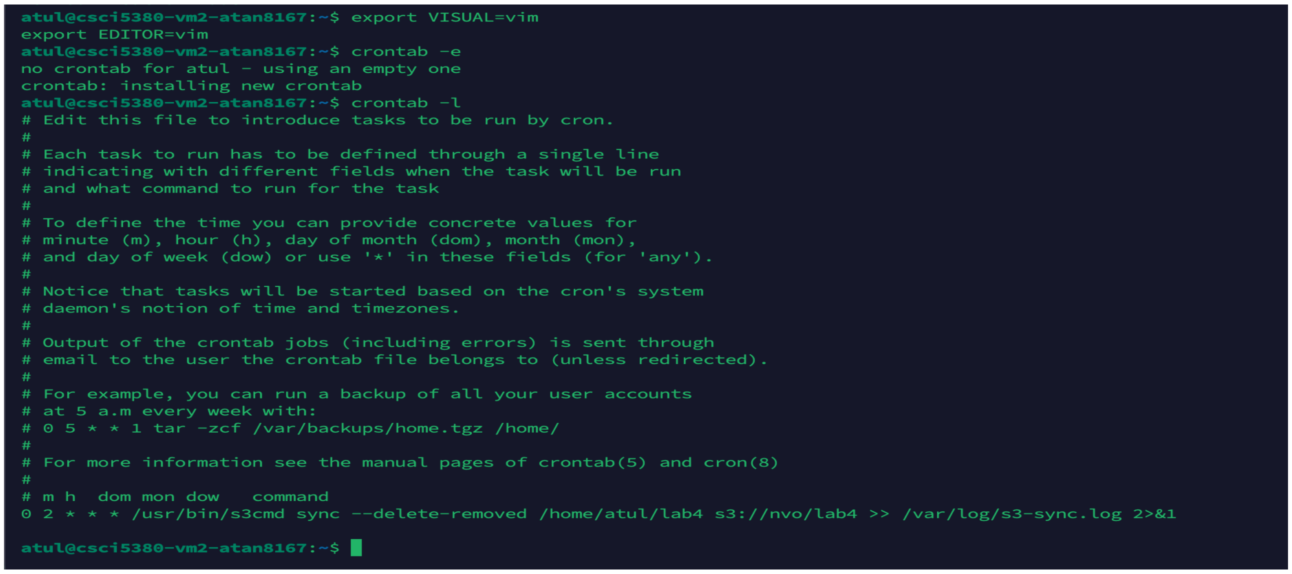
Find objects by prefix

Name	Type	Last modified	Size	Storage class
testfile	-	February 9, 2026, 09:44:29 (UTC-07:00)	0 B	Standard

7. What is another way of transferring data to your Amazon S3 bucket? [**2 points**]

The data can be manually uploaded to the s3 bucket as well as shown on the screenshot above as well as we can write a script to upload the data to the s3.

8. Create a **cronjob** to sync every night. Paste a screenshot of the **cronjob** created. [**5 points**]



```
atul@csci5380-vm2-atan8167:~$ export VISUAL=vim
export EDITOR=vim
atul@csci5380-vm2-atan8167:~$ crontab -e
no crontab for atul - using an empty one
crontab: installing new crontab
atul@csci5380-vm2-atan8167:~$ crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
0 2 * * * /usr/bin/s3cmd sync --delete-removed /home/atul/lab4 s3://nvo/lab4 >> /var/log/s3-sync.log 2>&1

atul@csci5380-vm2-atan8167:~$ 
```

9. What is Amazon Glacier and how is it different from S3? [**3 points**]

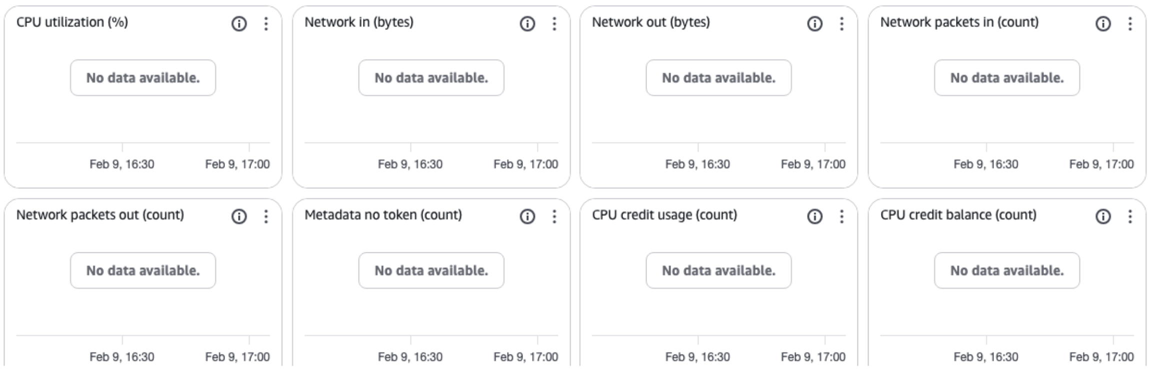
Glacier is also a storage solution by AWS but it provides a very low-cost alternative to S3 and is designed for long-term storage and data is not accessed very frequently and retrieval time could range from minutes to hours. S3 is very frequent and immediate access with sub-second latency and Glacier service is way cheaper than S3

Part 4

Objective 4.1 – Monitoring using CloudWatch

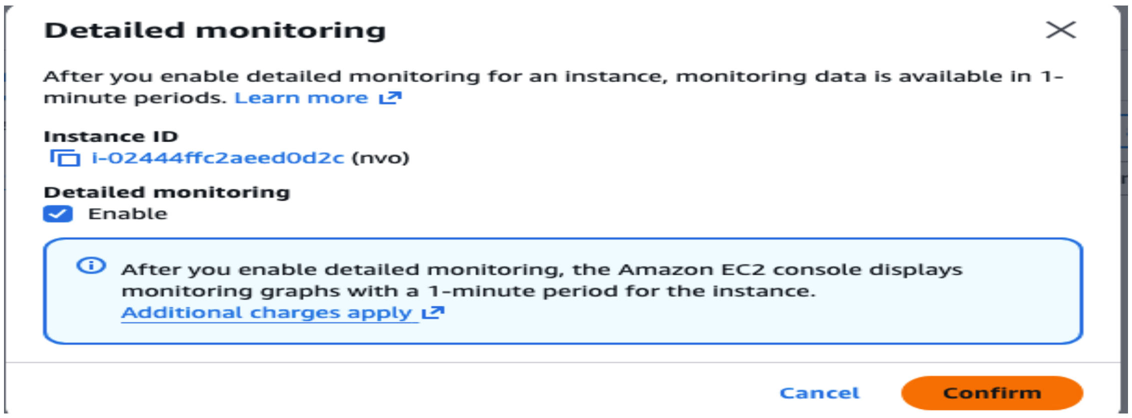
1. Click on any of the running EC2 instances and check the monitoring tab for EC2 metrics. What are the key metrics that you see and why is this important for an organization? [**5 points**]

I see CPU utilization, Network monitoring in terms of how much data is in and out and this allows for better resource management and cost predictability as instances can dynamically adjust based on the load, so it also provides a way to scale quickly if the organization expects that load is going to increase due to certain event.



2. Enable detail monitoring for CloudWatch metrics. What is the difference between Basic monitoring and Detailed monitoring? [**5 points**]

Detailed monitoring provides faster and granular as well as detailed insight in the logging as we can see from below screenshot the detailed monitoring provides details up to per minute period.



Detailed monitoring

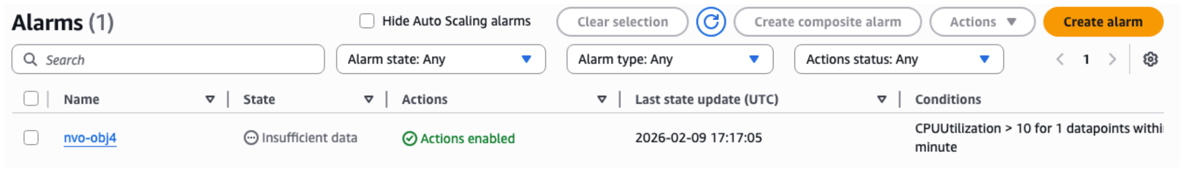
After you enable detailed monitoring for an instance, monitoring data is available in 1-minute periods. Learn more

Instance ID
i-02444ffc2aeed0d2c (nvo)

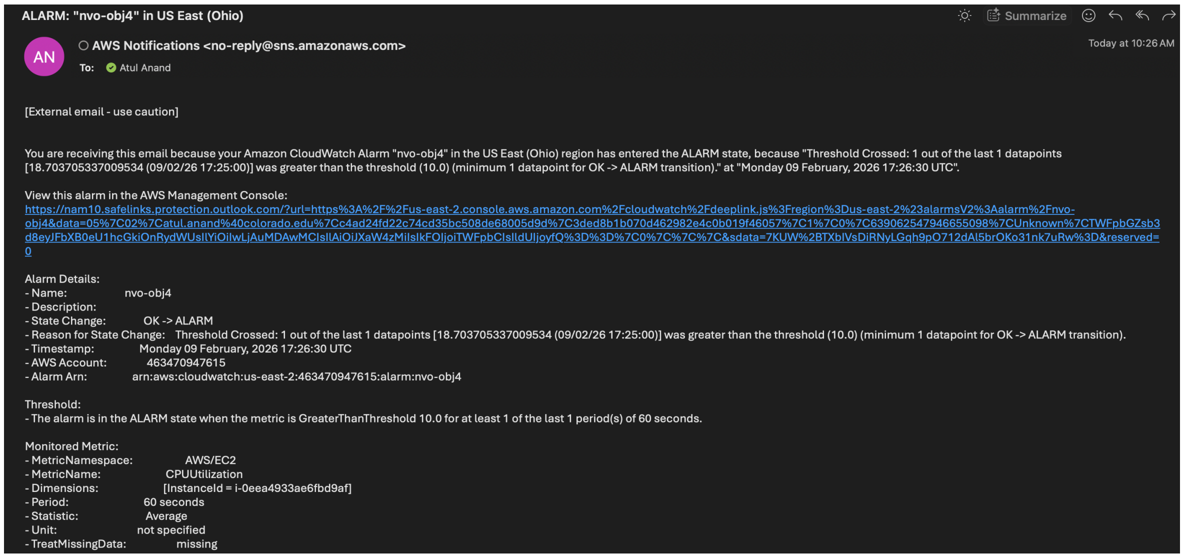
Detailed monitoring
✓ Enable

After you enable detailed monitoring, the Amazon EC2 console displays monitoring graphs with a 1-minute period for the instance.
Additional charges apply

3. Create a new **CloudWatch alarm** to monitor **average CPU utilization**. The alarm should take effect when average CPU utilization is greater than a user-defined threshold. Alarm should send an **E-Mail** to the recipients entered during the alarm configuration. Paste a screenshot of the alarm created and the Email that you received. [**10 points**]
[**NOTE:** For simplicity, enter the threshold to be less than 1%. Or you can use the Linux stress tool to generate artificial stress on your instance to increase its CPU utilization.]



Name	State	Actions	Last state update (UTC)	Conditions
nvo-obj4	Insufficient data	Actions enabled	2026-02-09 17:17:05	CPUUtilization > 10 for 1 datapoints withi minute



ALARM: "nvo-obj4" in US East (Ohio)

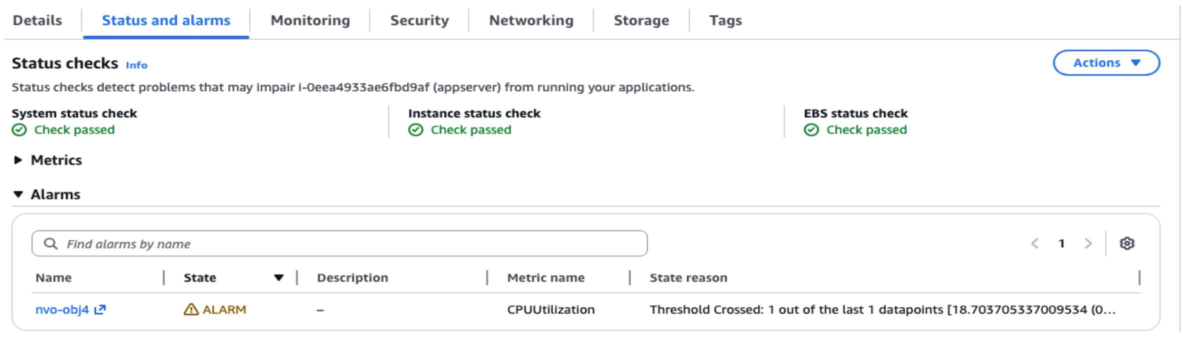
AWS Notifications <no-reply@sns.amazonaws.com>
To: Atul Anand
Today at 10:26 AM

[External email - use caution]

You are receiving this email because your Amazon CloudWatch Alarm "nvo-obj4" in the US East (Ohio) region has entered the ALARM state, because "Threshold Crossed: 1 out of the last 1 datapoints [18.703705337009534 (09/02/26 17:25:00)] was greater than the threshold (10.0) (minimum 1 datapoint for OK -> ALARM transition)." at "Monday 09 February, 2026 17:26:30 UTC".

View this alarm in the AWS Management Console:
https://nam10.safelinks.protection.outlook.com/?url=https%3A%2F%2Fus-east-2.console.aws.amazon.com%2Fcloudwatch%2Fdeeplink.js%3Fregion%3Dus-east-2%23alarmsV2%3Aalarm%2Fnvo-obj4&data=05%7C02%7Catul.anand%40colorado.edu%7Cc4ad24fd22c74cd35bc508de68005d9d%7C3ded8b1b070d462982e4c0b019f46057%7C1%7C0%7C639062547946655098%7CUnknown%7CTWFpbGZsb3d8eyJFbXB0eU1hcGkiOnRydWUsIlYiOiIwLjAuMDAwMCIsIlAiOiJXaW4zMiIsIkFOIjoiTWFpbCIsIldUIjoyfQ%3D%3D%7C0%7C%7C%7C&sdata=7KUW%2BTXbIVsDiRNyLGqh9pO712dAl5brOKo31nk7uRw%3D&reserved=0

```
Alarm Details:
- Name:              nvo-obj4
- Description:
- State Change:         OK -> ALARM
- Reason for State Change:   Threshold Crossed: 1 out of the last 1 datapoints [18.703705337009534 (09/02/26 17:25:00)] was greater than the threshold (10.0) (minimum 1 datapoint for OK -> ALARM transition).
- Timestamp:            Monday 09 February, 2026 17:26:30 UTC
- AWS Account:           463470947615
- Alarm Arn:           arn:aws:cloudwatch:us-east-2:463470947615:alarm:nvo-obj4

Threshold:
- The alarm is in the ALARM state when the metric is GreaterThanThreshold 10.0 for at least 1 of the last 1 period(s) of 60 seconds.

Monitored Metric:
- MetricNamespace:            AWS/EC2
- MetricName:               CPUUtilization
- Dimensions:               [InstanceId = i-0eea4933ae6fbd9af]
- Period:                 60 seconds
- Statistic:               Average
- Unit:                 not specified
- TreatMissingData:            missing
```



Status checks Info

Status checks detect problems that may impair i-0eea4933ae6fbd9af (appserver) from running your applications.

System status check	Instance status check	EBS status check
Check passed	Check passed	Check passed

Name	State	Description	Metric name	State reason
nvo-obj4	ALARM	–	CPUUtilization	Threshold Crossed: 1 out of the last 1 datapoints [18.703705337009534 (0...

4. Which is the service used by CloudWatch to send out E-Mail notifications? [**5 points**]

SNS – Simple notification service

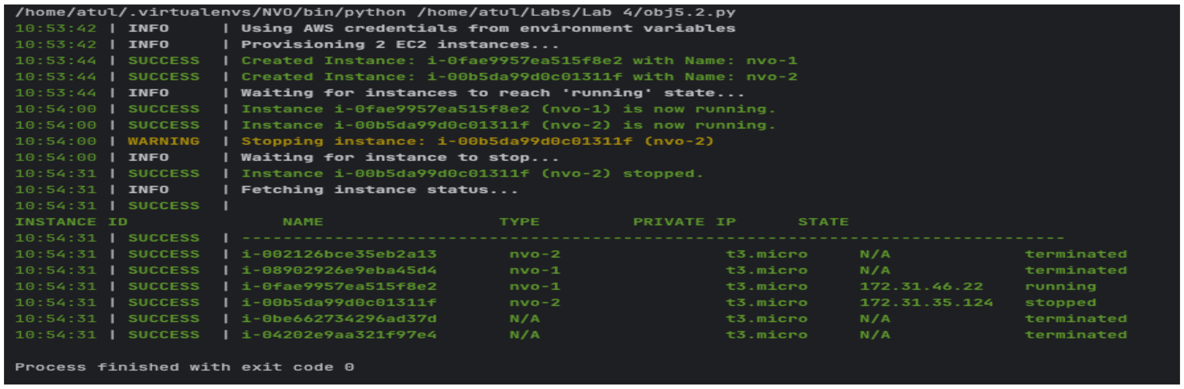
Part 5

Objective 5.1 Setting up BOTO3 for AWS resource automation

1. Install Boto 3 on your machine using “sudo -H pip install boto3” to download the required packages.
2. Before we start using Boto, it is mandatory to setup the necessary authentication to the AWS management console. In order to do this, we would need to download the AWS CLI and put in the AWS Access Keys which we have already downloaded in objective 1.3.
3. You may install the awscli for ease of authentication using “sudo -H pip install awscli” command. Further steps to achieve this are found in the below mentioned link: https://boto3.amazonaws.com/v1/documentation/api/latest/guide/quickstart.html#installation.
4. If you opt to not set up one-time authentication with AWS CLI, ensure you add Var_name = Session (aws_access_key_id = ' ', aws_secret_access_key = ‘ ‘ , region_name = ' ') in your python code for further objectives.

Objective 5.2 Controlling EC2 using Boto3

1. Write a python script to launch two new EC2 instances, stop one of the instances, and then fetch the details of all instances using the Boto3 module. Sample output: **[Instance Id] [Instance_type] [Instance_ip_address] “Running/Stopped”** [Hint: refer https://boto3.readthedocs.org/en/latest/guide/migrationec2.html]
2. Submit the .py file that you created to accomplish this objective, screenshots showing the new instances created on the console, and the instance details as specified in the sample output. [**30 points**]



```
/home/atul/.virtualenvs/NVO/bin/python /home/atul/Labs/Lab 4/obj5.2.py
10:53:42 | INFO     | Using AWS credentials from environment variables
10:53:42 | INFO     | Provisioning 2 EC2 instances...
10:53:44 | SUCCESS  | Created Instance: i-0fae9957ea515f8e2 with Name: nvo-1
10:53:44 | SUCCESS  | Created Instance: i-00b5da99d0c01311f with Name: nvo-2
10:53:44 | INFO     | Waiting for instances to reach 'running' state...
10:54:00 | SUCCESS  | Instance i-0fae9957ea515f8e2 (nvo-1) is now running.
10:54:00 | SUCCESS  | Instance i-00b5da99d0c01311f (nvo-2) is now running.
10:54:00 | WARNING  | Stopping instance: i-00b5da99d0c01311f (nvo-2)
10:54:00 | INFO     | Waiting for instance to stop...
10:54:31 | SUCCESS  | Instance i-00b5da99d0c01311f (nvo-2) stopped.
10:54:31 | INFO     | Fetching instance status...
10:54:31 | SUCCESS  |
INSTANCE ID          NAME                  TYPE          PRIVATE IP       STATE
10:54:31 | SUCCESS  | -------------------------------------------------------------------------------------
10:54:31 | SUCCESS  | i-002126bce35eb2a13       nvo-2                t3.micro     N/A              terminated
10:54:31 | SUCCESS  | i-08902926e9eba45d4       nvo-1                t3.micro     N/A              terminated
10:54:31 | SUCCESS  | i-0fae9957ea515f8e2       nvo-1                t3.micro     172.31.46.22     running
10:54:31 | SUCCESS  | i-00b5da99d0c01311f       nvo-2                t3.micro     172.31.35.124    stopped
10:54:31 | SUCCESS  | i-0be662734296ad37d       N/A                  t3.micro     N/A              terminated
10:54:31 | SUCCESS  | i-04202e9aa321f97e4       N/A                  t3.micro     N/A              terminated

Process finished with exit code 0
```

The script fetches all the instances and show the status if its running or deleted or stopped.

Objective 5.3 Fetching Cloudwatch metrics using Boto3

1. Write a python script to create a new AWS session using access keys (refer objective 1), create a cloudwatch session, and fetch the following metrics for one running EC2 instance over a specific time period (at least 30 minutes): Status_Check, CPU_Utilization, Network_In and Network_Out. Sample output:

   Instance ID: <value>

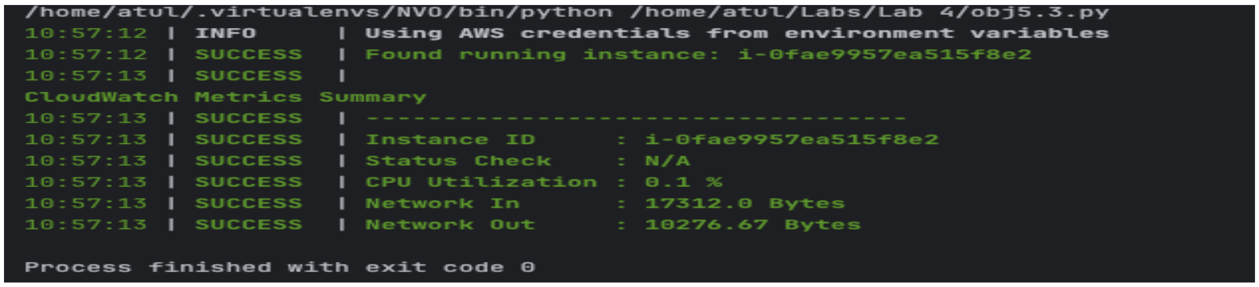
   Status Check: <value>

   CPU Utilization: <value>

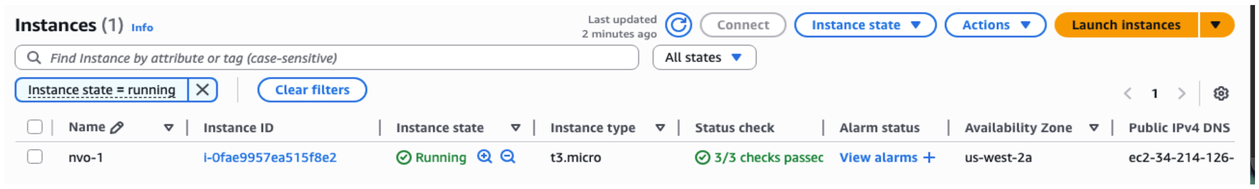
   Network In: <value>

   Network Out: <value>

2. Submit the .py file that you created to accomplish this objective and screenshots of the details as specified in the sample output. [**20 points**]



```
/home/atul/.virtualenvs/NVO/bin/python /home/atul/Labs/Lab 4/obj5.3.py
10:57:12 | INFO     | Using AWS credentials from environment variables
10:57:12 | SUCCESS  | Found running instance: i-0fae9957ea515f8e2
10:57:13 | SUCCESS  |
CloudWatch Metrics Summary
10:57:13 | SUCCESS  | -----------------------------------
10:57:13 | SUCCESS  | Instance ID      : i-0fae9957ea515f8e2
10:57:13 | SUCCESS  | Status Check     : N/A
10:57:13 | SUCCESS  | CPU Utilization  : 0.1 %
10:57:13 | SUCCESS  | Network In       : 17312.0 Bytes
10:57:13 | SUCCESS  | Network Out      : 10276.67 Bytes

Process finished with exit code 0
```



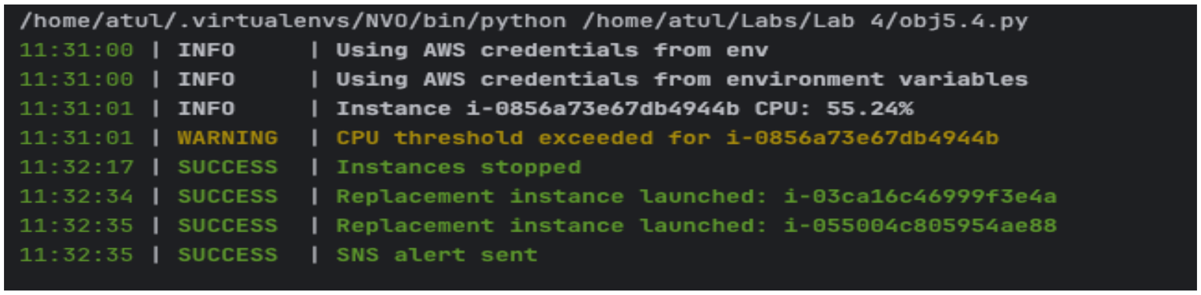
Instances (1) Info

Last updated 2 minutes ago

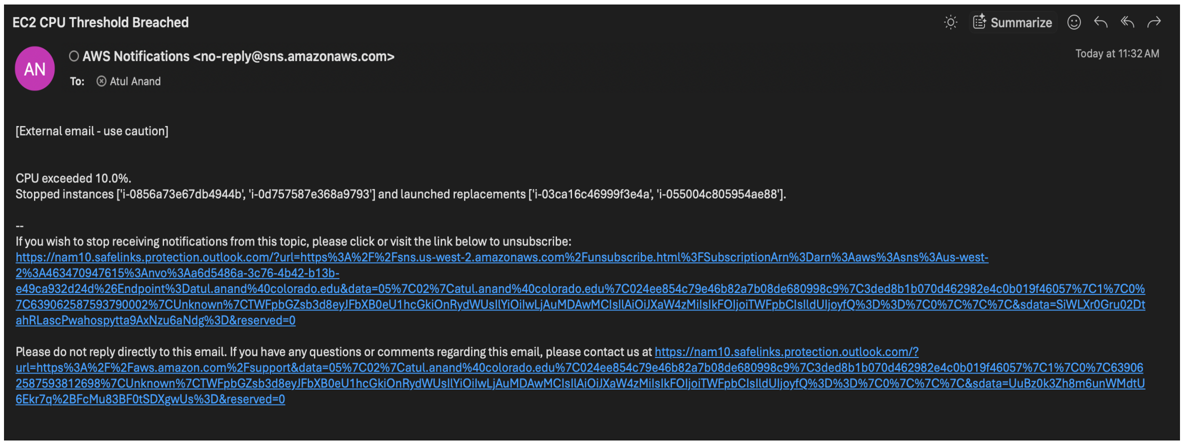
	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
	nvo-1	i-0fae9957ea515f8e2	Running	t3.micro	3/3 checks passec	View alarms +	us-west-2a	ec2-34-214-126-

Objective 5.4 Spinning new instances based on CPU utilization

Write a python script that uses Boto3 to continuously fetch CPU utilization of two running EC2 instances. When a specific threshold is reached, the script should automatically shut the instances down, spin up identical new instances and send out an alert email to your email id. Submit the .py file that you created to accomplish this and relevant screenshots. [**20 points**]



```
/home/atul/.virtualenvs/NVO/bin/python /home/atul/Labs/Lab 4/obj5.4.py
11:31:00 | INFO     | Using AWS credentials from env
11:31:00 | INFO     | Using AWS credentials from environment variables
11:31:01 | INFO     | Instance i-0856a73e67db4944b CPU: 55.24%
11:31:01 | WARNING  | CPU threshold exceeded for i-0856a73e67db4944b
11:32:17 | SUCCESS  | Instances stopped
11:32:34 | SUCCESS  | Replacement instance launched: i-03ca16c46999f3e4a
11:32:35 | SUCCESS  | Replacement instance launched: i-055004c805954ae88
11:32:35 | SUCCESS  | SNS alert sent
```



EC2 CPU Threshold Breached

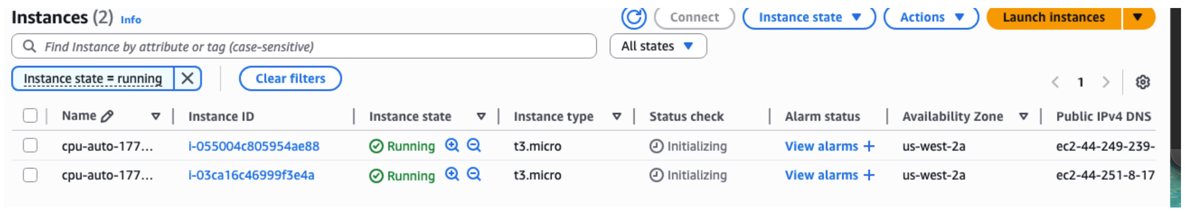
AWS Notifications <no-reply@sns.amazonaws.com>
To: Atul Anand
Today at 11:32 AM

[External email - use caution]

CPU exceeded 10.0%.
Stopped instances ['i-0856a73e67db4944b', 'i-0d757587e368a9793'] and launched replacements ['i-03ca16c46999f3e4a', 'i-055004c805954ae88'].

--
If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe:
https://nam10.safelinks.protection.outlook.com/?url=https%3A%2F%2Fsns.us-west-2.amazonaws.com%2Funsubscribe.html%3FSubscriptionArn%3Darn%3Aaws%3Asns%3Aus-west-2%3A463470947615%3Anvo%3Aa6d5486a-3c76-4b42-b13b-e49ca932d24d%26Endpoint%3Datul.anand%40colorado.edu&data=05%7C02%7Catul.anand%40colorado.edu%7C024ee854c79e46b82a7b08de680998c9%7C3ded8b1b070d462982e4c0b019f46057%7C1%7C0%7C639062587593790002%7CUnknown%7CTWFpbGZsb3d8eyJFbXB0eU1hcGkiOnRydWUsIlYiOiIwLjAuMDAwMCIsIlAiOiJXaW4zMiIsIkFOIjoiTWFpbCIsIldUIjoyfQ%3D%3D%7C0%7C%7C%7C&sdata=SiWLXr0Gru02DtahRLascPwahospytta9AxNzu6aNdg%3D&reserved=0

Please do not reply directly to this email. If you have any questions or comments regarding this email, please contact us at https://nam10.safelinks.protection.outlook.com/?url=https%3A%2F%2Faws.amazon.com%2Fsupport&data=05%7C02%7Catul.anand%40colorado.edu%7C024ee854c79e46b82a7b08de680998c9%7C3ded8b1b070d462982e4c0b019f46057%7C1%7C0%7C639062587593812698%7CUnknown%7CTWFpbGZsb3d8eyJFbXB0eU1hcGkiOnRydWUsIlYiOiIwLjAuMDAwMCIsIlAiOiJXaW4zMiIsIkFOIjoiTWFpbCIsIldUIjoyfQ%3D%3D%7C0%7C%7C%7C&sdata=UuBz0k3Zh8m6unWMdtU6Ekr7q%2BFcMu83BF0tSDXgwUs%3D&reserved=0



Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
cpu-auto-177…	i-055004c805954ae88	Running	t3.micro	Initializing	View alarms +	us-west-2a	ec2-44-249-239-
cpu-auto-177…	i-03ca16c46999f3e4a	Running	t3.micro	Initializing	View alarms +	us-west-2a	ec2-44-251-8-17

Python script is uploaded and I have used env files and each script uses each other like cpu utilization is fetched from the 5.3 and used in 5.4

I used linux stress tool to spike the CPU usage.

Reflection:

1. Now that you have learnt the basics, what do you think are the most important reasons for an organization to use AWS? [**5 points**]

AWS provides very fast scalable solutions along with on demand services such as S3 and EC2 and for an organization they can focus the effort on the core business and use AWS to quickly scale their business opportunities. So, Netflix uses AWS as their core infrastructure which allows them to react quickly to customer needs.

2. Suggest any other AWS modules that you would like to learn about. Why?

I would like to learn how to host container which will allow me to host my website without managing the environment and any update can be simply new container being pushed to the AWS using cli or boto.

3. Suggest any other public cloud platforms that you would like to learn about. Why?

I would like to try Google cloud as well as it seems to understand their services as I have used Azure for little while.

Total Points _____________ / 207