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]

1. 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**]

A screenshot of a computer

AI-generated content may be incorrect.

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**].

A screenshot of a computer

AI-generated content may be incorrect.

1. 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 is the Amazon Elastic Compute Cloud. It is a compute platform that allows you to spin up VMs.

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

* Spot instances

A spot instance is a small instance that uses very little EC2 capacity that is available for a lower price.

* Reserved instances

A reserved instance is just a spot instance with a long-term commitment instead of being available immediately.

* Dedicated hosts

A physical server that is fully dedicated for the users use.

* Dedicated instances

A dedicated instance is a VM that runs on hardware dedicated to only one user.

* Elastic GPUs

Allows you to attach GPUs to an EC2 instance.

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.

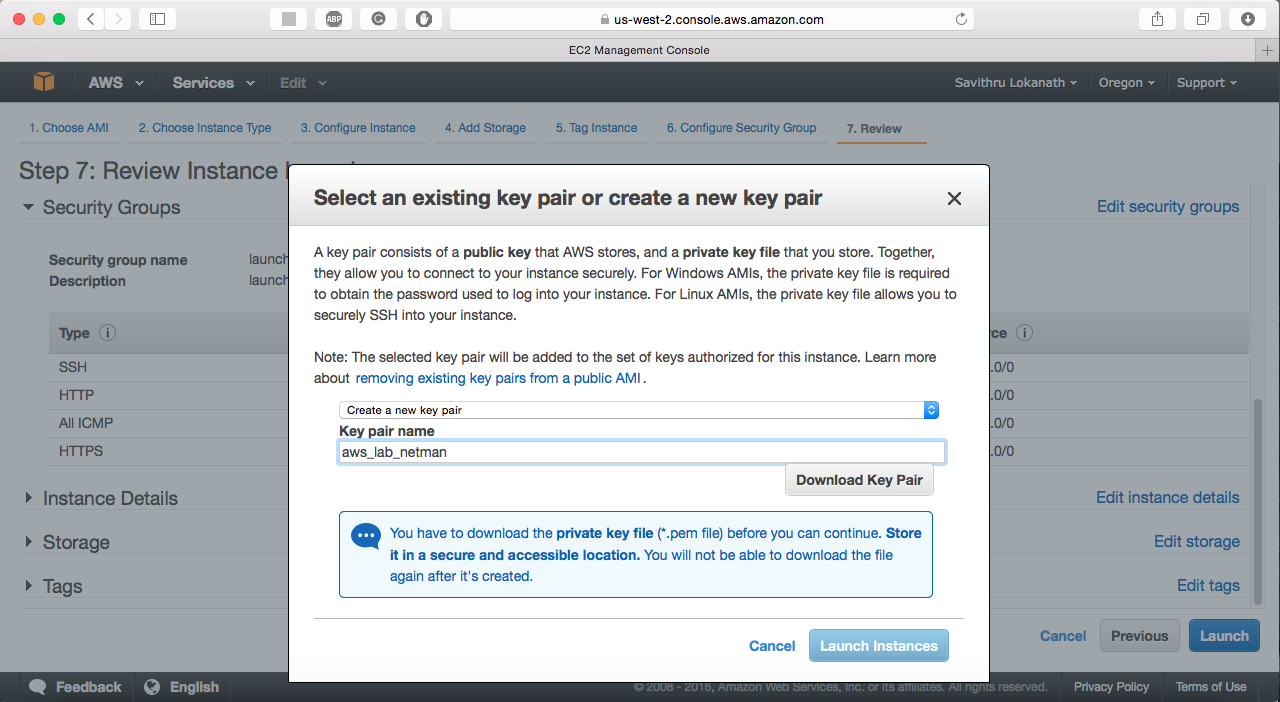
**Step 4:** Add storage (8 GB is fine, can add more later)

**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.

**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.

**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.



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

It is related to the instance because it is the software that is required to set up and boot an EC2.

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

Anyone could potentially access your instances on the internet.

There could be DOS attacks sent to your instances.

This is a security concern because the

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

North-South:

Firewalls and traffic encryption are best practice

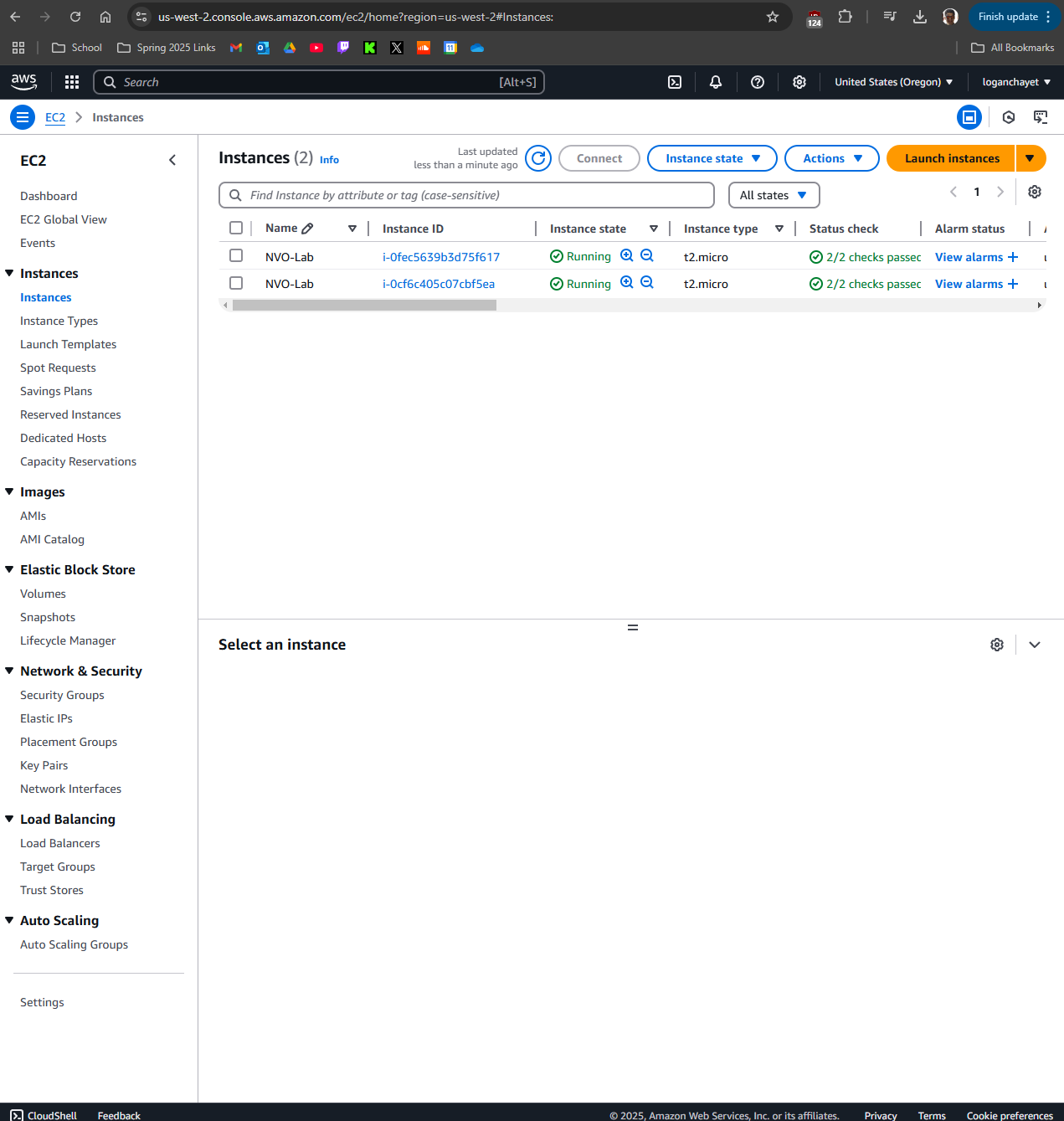
Maybe some DDoS protection and secure API gateways

East-West:

Monitor and log traffic

Network Access Controls

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



1. 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**]

A screenshot of a computer program

AI-generated content may be incorrect.

1. 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 SSH’ed to: 35.166.40.166. Interface enX0 is: 172.31.31.87. I believe there is some sort of NATing going on when SSHing to my instance. When I SSH, it leaves my network, destined for the public IP address. It reaches the AWS edge, and the network handles my request and sends it to the appropriate VPC. Then it goes through the security group I made, probably some ACLs and then arrives at the EC2 OS.

1. 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**]

sudo adduser logan

su logan

mkdir .ssh/

cd .ssh/

vim authorized\_keys

add in the public key created from the key pair in aws.

Then it works!

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

Stopping the instance turns off the instance but keeps storage and configs. Terminating the instance deletes the instance and its storage.

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

I went to my instance and then clicked:  
Actions → Instance Settings → Change Termination Protection

And checked the enable box.

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**]

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Put the DNS name or IP in the address bar.

Both

# 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**]

A storage service that allows for users to store or get information from anywhere.

1. 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.)]

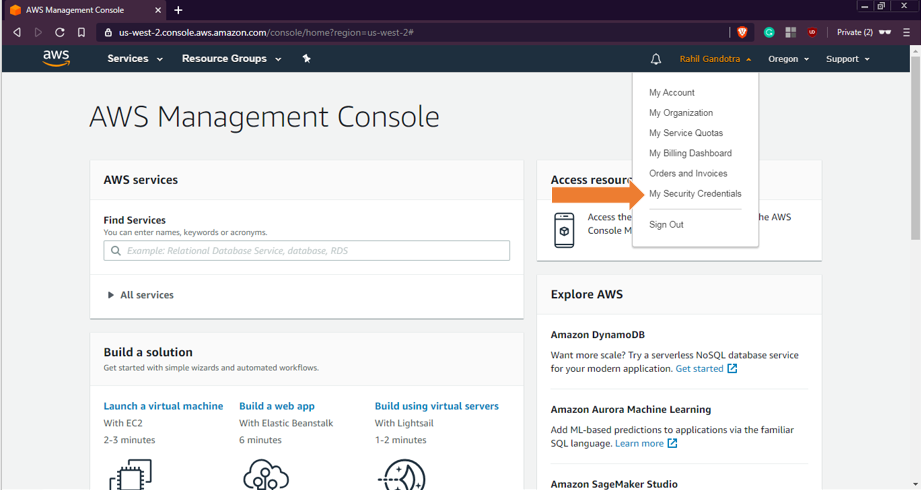
1. Update your VM and install S3 command-line utility

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

1. 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

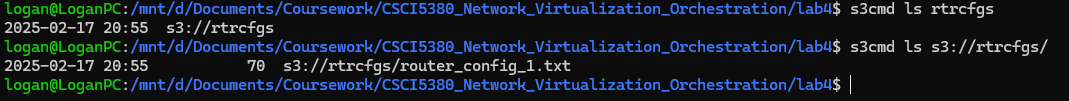


1. Create an S3 bucket

**(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.

1. 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**]



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

I could use the boto3 library to input files via code.

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

0 0 \* \* \* /usr/bin/aws s3 sync /home/ubuntu/my-folder/ s3://my-bucket/ >> /home/ubuntu/s3\_sync.log 2>&1

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

Glacier is a slightly slower access version of S3. It costs less, but getting access to materials is a bit slower. It is most commonly used for archival file storage.

# Part 4

# Objective 4.1 – Monitoring using CloudWatch

# 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]

A screenshot of a computer

AI-generated content may be incorrect.

Some of the key metrics I see are: CPU util, Network in/out, CPU credit balance, etc. This is important for an organization in many ways, some being for managing cost and need for instances/services. They need to know what’s going on so they can alter their network accordingly and efficiently.

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

Instead of monitoring every 5 minutes, it brings it down to 1 minute.

1. 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](https://www.tecmint.com/linux-cpu-load-stress-test-with-stress-ng-tool/) to generate artificial stress on your instance to increase its CPU utilization.]

A screenshot of a computer

AI-generated content may be incorrect.

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

It is the Simple Notification Service (SNS)

# 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>]

1. 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**]

# A computer screen with white text AI-generated content may be incorrect.

# 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>**

[Hint: <https://boto3.amazonaws.com/v1/documentation/api/latest/reference/services/cloudwatch.html#CloudWatch.Client.get_metric_statistics>]

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

A computer screen with blue and white text

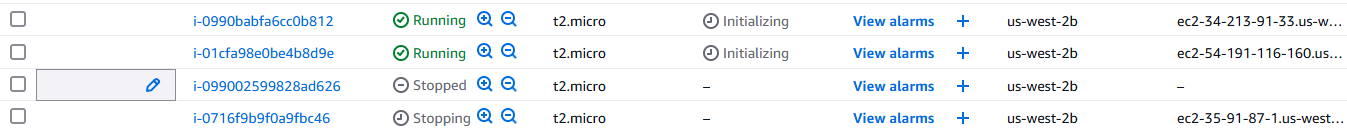
AI-generated content may be incorrect.

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**]

A computer screen with white and blue text

AI-generated content may be incorrect.



A screenshot of a computer

AI-generated content may be incorrect.

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**]
2. Suggest any other AWS modules that you would like to learn about. Why?
3. Suggest any other public cloud platforms that you would like to learn about. Why?

# Total Points \_\_\_\_\_\_\_\_\_\_\_\_ / 207