# Lab 1 - AWS,EC2,EBS and S3

## Week 1 Tasks:

### a) AWS Management Console

### b) Understanding and creating AWS EC2(Elastic Compute) virtual machines

Deliverables:

i. 1b1.png showing the summary of instance before launching (*Step 7: Review Instance Launch*)
ii. 1b2.png showing the EC2 instance in running state
iii. 1b3.png should be a screenshot of the SSH terminal, showing compiling and running a sample C program

### c) Understanding and using AWS EBS(Elastic Block Store)

Deliverables:

i. 1c1.png showing the newly created EBS in an *available* state.
ii. 1c2.png showing the created EBS attached to an EC2 instance. The screenshot should be showing all the volumes of the EC2 instance on the EC2 management console.
iii. 1c3.png should show an output of the terminal, by running a linux command showing all the volumes, file systems and size attached to the EC2 instance.
iv. 1c4.png should the newly created *snapshot* on the management console
v. 1c5.png showing the newly created *volume* from the created snapshot

### d) Object Storage using S3 Buckets

Deliverables:

i. 1d1.png showing the uploaded image using the public URL(URL must be clearly shown)

ii. 1d2.png showing different versions of the text file in the bucket overview

iii. 1d3.png showing the modified text file using the public URL of the file(URL must be clearly shown)

## Task a : AWS Management Console

Before getting into the details of this lab (and further labs) it is important to be familiar with the central hub/point of AWS , the AWS Management Console. Take a read in the link to understand what you can do with the console.

What is AWS ?

Working with the AWS Management Console

All AWS services can be accessed using the AWS management console, take a look around the console and the services and familiarise yourself with the environment!

**Questions:**

1. The AWS management console is a web app to interact with AWS services. What other ways are there to interact with AWS services?

## Task b : Understanding and creating AWS EC2(Elastic Compute) virtual machines

One of the most important is the AWS EC2 (Elastic Compute) service. This provides the actual ***compute*** for your cloud applications and as any cloud service should be, scalable(hence the name elastic!). EC2 is a service that provides resizable compute capacity in the cloud. It is designed to make web-scale cloud computing easier for developers.In simpler words EC2 is none other than a virtual machine.

Understanding AWS EC2 - 4 minute video explaining AWS EC2

Every EC2 instance has the following properties/options/attributes that need to compulsorily be configured:

- Amazon Machine Image(AMI)
- Instance Type (The Instance Type usually depends on the use of the VM)
- Specific Instance Details such as network,subnets,start up scripts etc.
- Security Groups - These are essentially the firewalls to your instance, they control the access to your instance.

The following steps will help you in setting up a EC2 instance. Make sure you really understand how to create an EC2 instance as EC2 will be used throughout this course

1. Make sure you have your AWS Educate credentials ready, you would have received a mail from AWS Educate to register and sign up and then change your password.
2. Once mail is received, login to AWS Educate. Go to AWS Management Console page. Under ‘Services’ tab, click on EC2.
3. You’ll reach the EC2 console page. Click Launch Instance.

4. Here you'll be presented with a list of Amazon Machine Images (AMI). They're essentially the virtual OSes that will run your server. We recommend you pick Ubuntu as you're likely to be familiar with its command line tools and package manager.
5. Now you'll be presented with a list of Instance Types. Each one has different machine specs like number of vCPUs, size of memory, storage volume type, etc. Pick one that is appropriate to your needs. For this a t2.micro(free tier eligible) is good enough.You can leave the default settings in the Configure Instance, Add Storage, and Add Tags pages, or configure it according to your needs.
6. On the Configure Security Group page, make sure you allow traffic on SSH and HTTP. Make sure to also install an SSH client on your local machine. This is a critical step whenever setting up an EC2 instance as this defines the access to your instance from the internet!
7. Create a cryptographic key pair and download the private key to your system. Then launch the instance. You may need to change the permissions of the PEM key.
    a. PEM Key Permissions
8. Change the instance name to your SRN
9. Learn about how to remotely login to another machine using SSH.
    a. SSH Basics
10. Find the public DNS for your instance. Use that and the private key to SSH into your instance. You'll need to figure out what the default username of your VM is, and how to point your SSH client to the private key.
    a. SSH into AWS instances
    b. General prerequisites for connecting to your instance
11. To delete an instance select the instance, choose *Instance State* and in the dropdown select *Terminate Instance*.

Note: AWS Educate probably does not allow you to create all kinds of instances. Make sure it's EC2. And ensure your Region is "US-East"(Northern Virginia). AWS Educate does not work for any other region.

**Your task is to create an EC2 instance with the following configurations/settings:**

| | |
|---|---|
| **AMI** | Ubuntu Server 20.04 LTS (HVM), SSD Volume Type |
| **Instance Type** | t2.micro |
| **Instance Details** | Default values (But try to understand what the different options are) |
| **Storage** | Default values |

You may get a warning saying “Improve your instances' security. Your security group, launch-wizard-1, is open to the world”, you can ignore this for now.

1. After creating and getting into your EC2 instance, update the “apt” package, and install GCC compiler.
2. Write a simple C program and compile it.

**Questions to think about:**

1. What happens when you *stop* an instance? How is it different from *terminating* an instance?
2. How are EC2 instances charged? How does it come under the pay-per-use model of cloud computing?

## Task c : Understanding and using AWS EBS(Elastic Block Store)

Before getting into AWS EBS, it is important to understand what a block store is and how they are different from other types of cloud storage: What is Block Storage?

Think of EBS as the hard drive (SSD or HDD) that you have in your laptop.Amazon EBS offers persistent storage for Amazon Elastic Compute Cloud (Amazon EC2) instances. Amazon EBS volumes are network-attached and persist independently from the life of an instance. Amazon EBS volumes are highly available, highly reliable volumes that can be leveraged as an Amazon EC2 instance boot partition or attached to a running Amazon EC2 instance as a standard block device. Think of EBS as the hard drive (SSD or HDD) that you have in your laptop.

The following links will help you understand EBS in a good amount of detail:

Amazon Elastic Block Store (EBS) Overview

Amazon Elastic Block Store

- EBS Volume console can be accessed through the EC2 console. Go to the EC2 console , on the right hand side *Elastic Block Store -> Volumes.*
- To check the volumes attached to your instance, select the instance on the EC2 console, on the button you will see

Another important feature/concept of EBS is to create snapshots.You can back up the data on your Amazon EBS volumes to Amazon S3 by taking point-in-time snapshots. Snapshots are incremental backups, which means that only the blocks on the device that have changed after your most recent snapshot are saved.But note that although the EBS snapshots are stored in S3 buckets, you don't have access to this bucket as AWS abstracts those details from you as all you need are the snapshots.

Amazon EBS snapshots - Amazon Elastic Compute Cloud

**Your sub tasks are to:**

1. Create an EC2 instance with the below mentioned specs.The instance name should again be your SRN.
2. Create an EBS volume with the below mentioned specs. Check in the EBS console, the volume you have just created should have an "available" status, indicating it is ready to be attached to an EC2 instance. Name the EBS volume as SRN_ebs
3. Now, *attach* the created EBS volume to the created EC2 instance. Remember every EC2 instance already has a *boot volume EBS* , what you are *attaching* is an additional volume.

   **Attaching a volume to an EC2 instance is analogous to just connecting a bare hard drive to a system without any file system,information etc. It is only physically connected but the OS needs more than just a physical attachment , i.e it needs a (1) Mount Point and (2) File system information of the volume.**

4. SSH into the EC2 instance
5. Check to see the currently attached volume, can you see it?
6. Find the name of the block device from the shell. lsblk-command-in-linux.
7. Create a file system on the created EBS volume as per specifications. You need to create a file system on the new volume, don't modify any other file systems on any other volumes.
8. Create a directory on the host system(as per specs) and mount the file system-formatted volume to that directory.
9. List the current disks attached to the EC2 instance using the `df -h` command.
10. Snapshot the previously created volume:
    a. Give the description as: "Snapshot of my volume"
    b. Key: "Name"
    c. Value: "YOUR SRN_ebs"
11. Using the snapshot you just created, create another volume of 10 GiB. Name the newly created volume as "SNAPSHOT"

**The specifications for the EBS volume are:**

| | |
|---|---|
| **Volume Type** | General Purpose SSD (gp2) |
| **Size (GiB)** | 10 |
| **Availability Zone** | Same availability zone as the EC2 instance |
| **Encryption** | Default(No Encryption) |
| **File System** | ext3 |
| **Mount Directory on host** | /mnt/data-store |

**EC2 instance specs:**

| | |
|---|---|
| **AMI** | Ubuntu Server 18.04 LTS (HVM), SSD Volume Type |
| **Instance Type** | t2.micro |
| **Instance Details** | Default values |
| **Storage** | Default values |

**Note:** Ensure the created EBS and EC2 volume belong to the same availability zone otherwise you will not be able to attach the volume

Creating an Amazon EBS volume - Amazon Elastic Compute Cloud

Creating & Attaching additional EBS

Viewing volumes and filesystems in Linux

Mounting volumes to directories in Linux

**Questions:**

1. Is there any relationship between an Amazon AMI and the EBS ?

   *Hint: Think about the default EBS attached to the volume, what all should it have configured?*

2. What are the different types of EBS volumes available? What kind of volume is used as the *boot volume* ? What should the different types be used for?
3. How many volumes can you attach to a single EC2 instance? What factors does this depend on?
4. What is the lifetime of an EBS volume? Is/Can an EBS volumes lifetime be bound to the lifetime the EC2 instance it is attached to?
5. Can a single volume be bound to different EC2 instances?

## Task d : Object Storage using S3 Buckets

Before getting into AWS S3, it is important to understand what an object store is and how they are different from other types of cloud storage: What is object storage?

Amazon Simple Storage Service (Amazon S3) is an object storage service that offers industry-leading scalability, data availability, security, and performance. This means customers of all sizes and industries can use it to store and protect any amount of data for a range of use cases, such as websites, mobile applications, backup and restore, archive, enterprise applications, Internet of Things (IoT) devices, and big data analytics. Amazon S3 provides easy-to-use management features so you can organize your data and configure finely-tuned access controls to meet your specific business, organizational, and compliance requirements. Amazon S3 is designed for 99.999999999% (11 9's) of durability and stores data for millions of applications for companies all around the world.

Introduction to Amazon S3

AWS Cloud Object Storage

This lab on AWS S3 will focus on two parts:

1. Creating an S3 bucket,adding a file,modifying access
2. Enable and use versioning for objects in S3.

**Your sub-tasks for 1 are to:**

1. Create an S3 bucket with a unique name as per specs.
    a. How do I create an S3 Bucket? - Amazon Simple Storage Service
2. Upload an image file(mentioned in specs) to the S3 bucket.
    a. Uploading an object to a bucket - Amazon Simple Storage Service
3. Get the public URL for the uploaded image and check access by pasting the URL into a browser.
    a. How do I see an overview of an object? - Amazon Simple Storage Service
4. Change access of the bucket to allow objects to have public access.
    a. Change bucket public access
5. Make the object *public* so as to allow public access. Check access again in a similar manner as sub-task 3.
    a. Grant public read access to some objects in my Amazon S3 bucket

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures.

Object Versioning - Amazon Simple Storage Service

Using versioning - Amazon Simple Storage Service

**Your sub-tasks for 2 are to:**

1. Use the same S3 bucket in the previous step.
2. Enable bucket versioning for the previously created bucket.
    a. How do I enable or suspend versioning for an S3 bucket? - Amazon Simple Storage Service
3. Download the sample text file(mentioned in specs)

4. Upload the sample text file to the bucket, enable public access for the object and check public access.
5. Now modify the sample text file as follows:
    a. Add a new line to the file mentioning "This is version 2 of the file"
    b. Add a new line saying "My SRN is " <Your SRN here>.
    **c. Do not change the file name, if you change the file name the bucket will treat it as a different file**
6. Upload the modified text file to the S3 bucket.
7. Enter the same URL of the text file as you did before uploading the modified text file - you should see the updated text.
8. Go back to the bucket overview page and ensure that the *List Versions* toggle is on.
9. Under the sample text file uploaded, a new line appears showing the older version of the text file.
10. Try to access the older version of the text file.

**S3 bucket specs:**

| | |
|---|---|
| **Bucket Name** | Your SRN, letters of your SRN must be in small case |
| **Image file** | new-report.png |
| **Text file** | sample-file.txt |

**Few points to note:**

1. Download the image and text file to your system before uploading to S3, don't change the file names while downloading.
2. Ensure the screenshots you upload clearly show the S3 object URL.

**Questions:**

1. This lab dealt with image and text files. What kind of objects can be uploaded to S3? Are there any types of objects you can't upload to S3?
2. Is there any size limit for the S3 bucket? What is the largest size a single object can have in a S3 object?
3. Think of an application that uses S3 buckets to store user images. This application is used across continents(Americas,Europe,Asia). Should you use the same bucket across regions or multiple buckets across regions ? How would you do the latter?