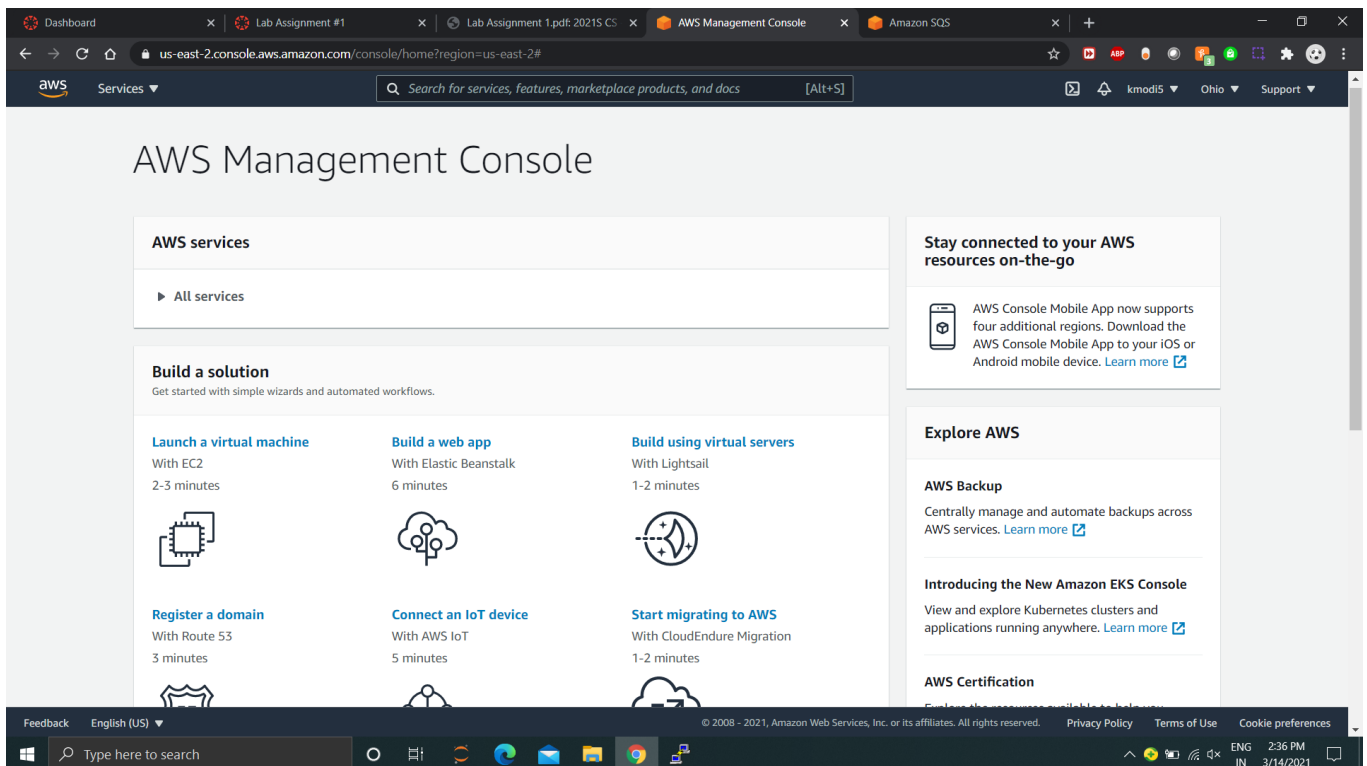


Lab 1: CS 524

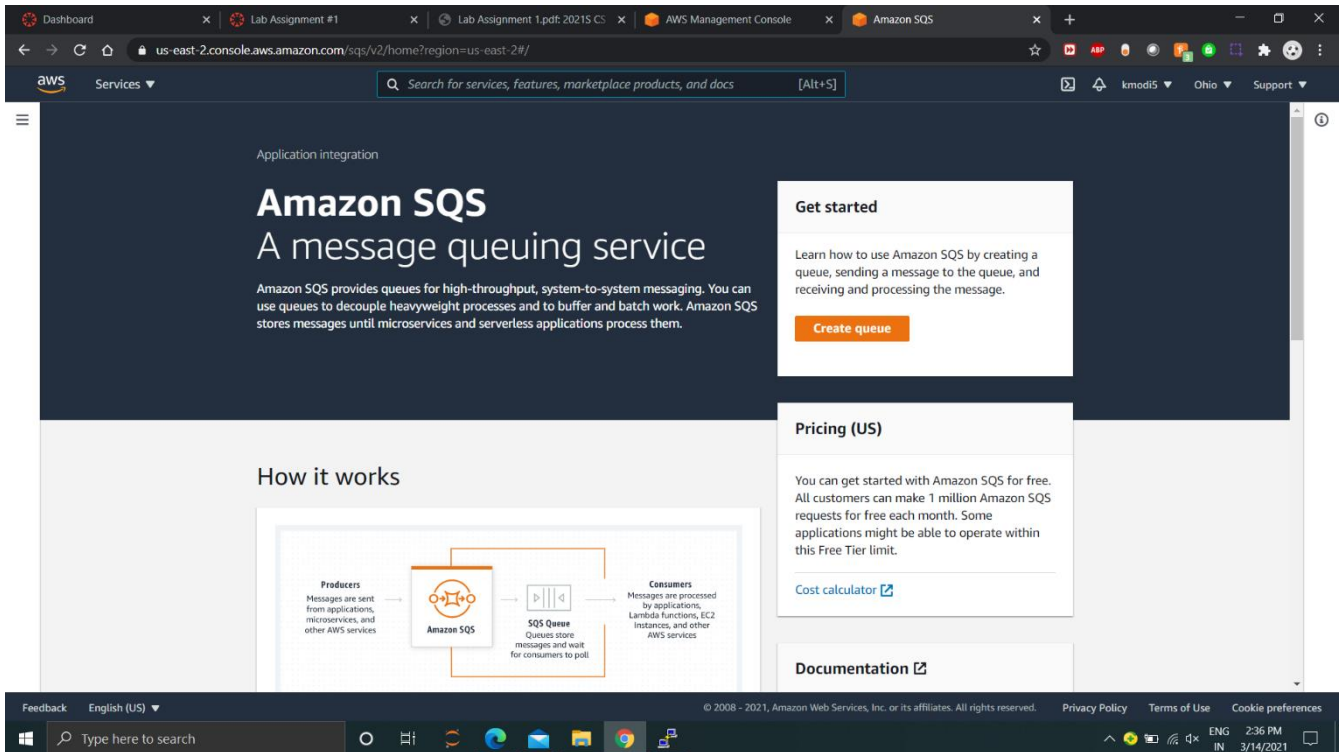
1. Part 1: Before proceeding with part 1, please review the link: <https://aws.amazon.com/getting-started/hands-on/send-messages-distributedapplications/> In the first part, you are going to learn about Amazon Simple Queue Service (SQS), which is an asynchronous messaging service that allows application components to communicate in the cloud. You will be creating a standard SQS queue named. Make sure to name the queue as your first name. Once the queue is created, you will be sending a simple message to the queue. The message body can have any one “fun” fact about you. Once you have successfully created and sent the message, you will view your message by polling the messages in the queue (in this case, it’s just one message). After you view the message, you will be deleting that message. [Screenshots of all of the above steps should be attached]. After deleting the message, make sure you delete the queue that you had created.

ANS.

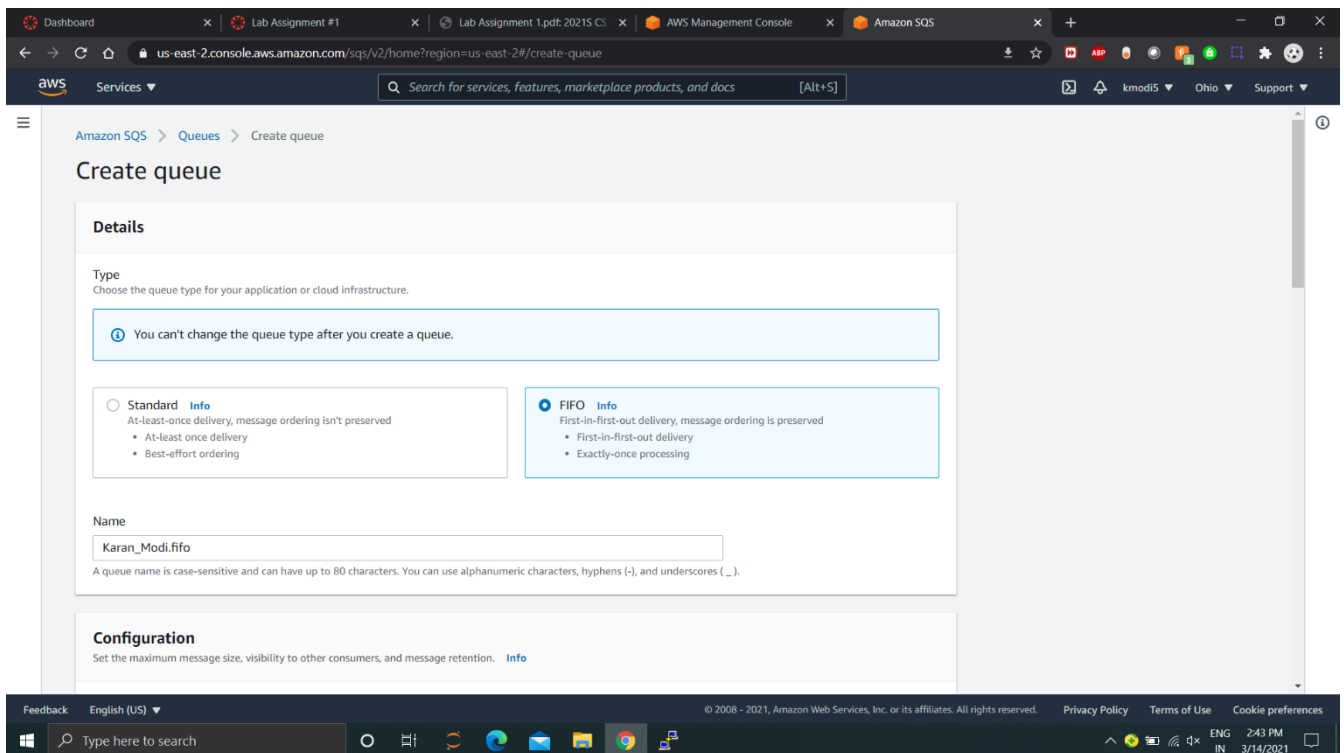
- a. Once registered for AWS, now go into AWS Console



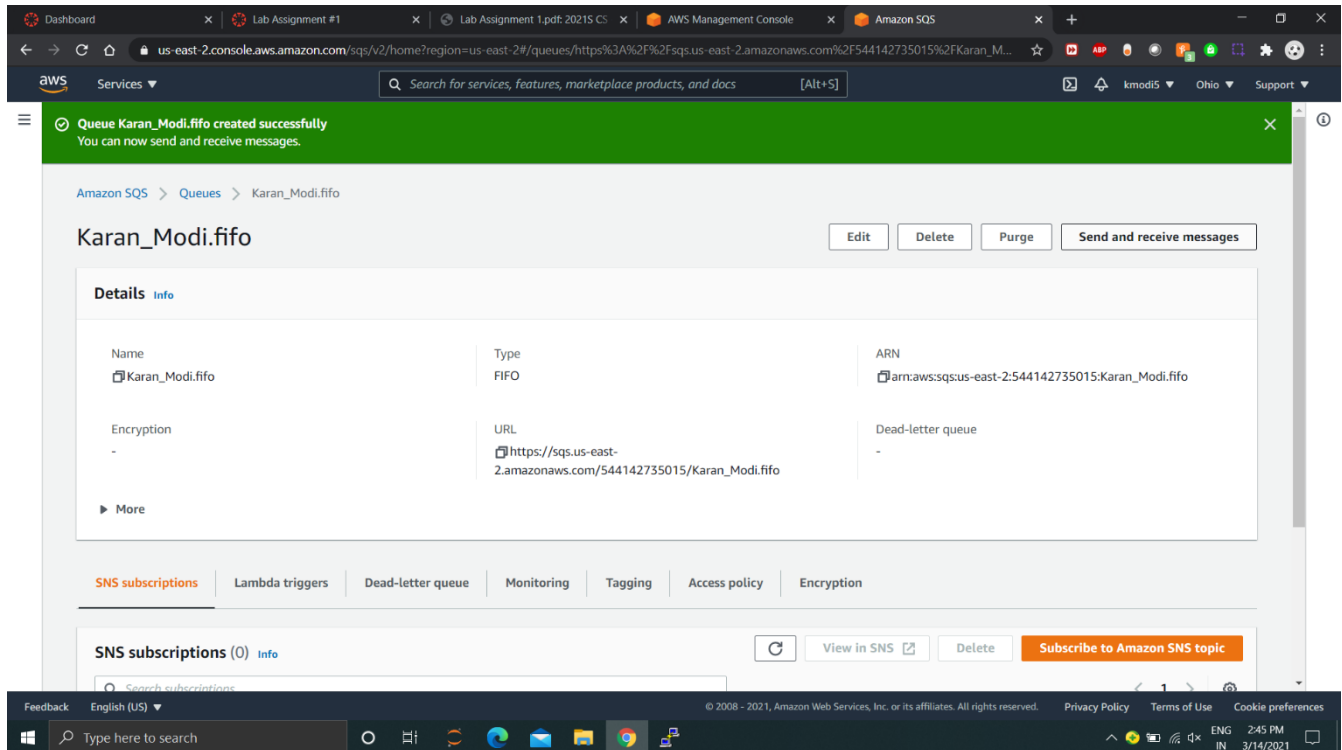
b. Go to Amazon SQS using the search bar



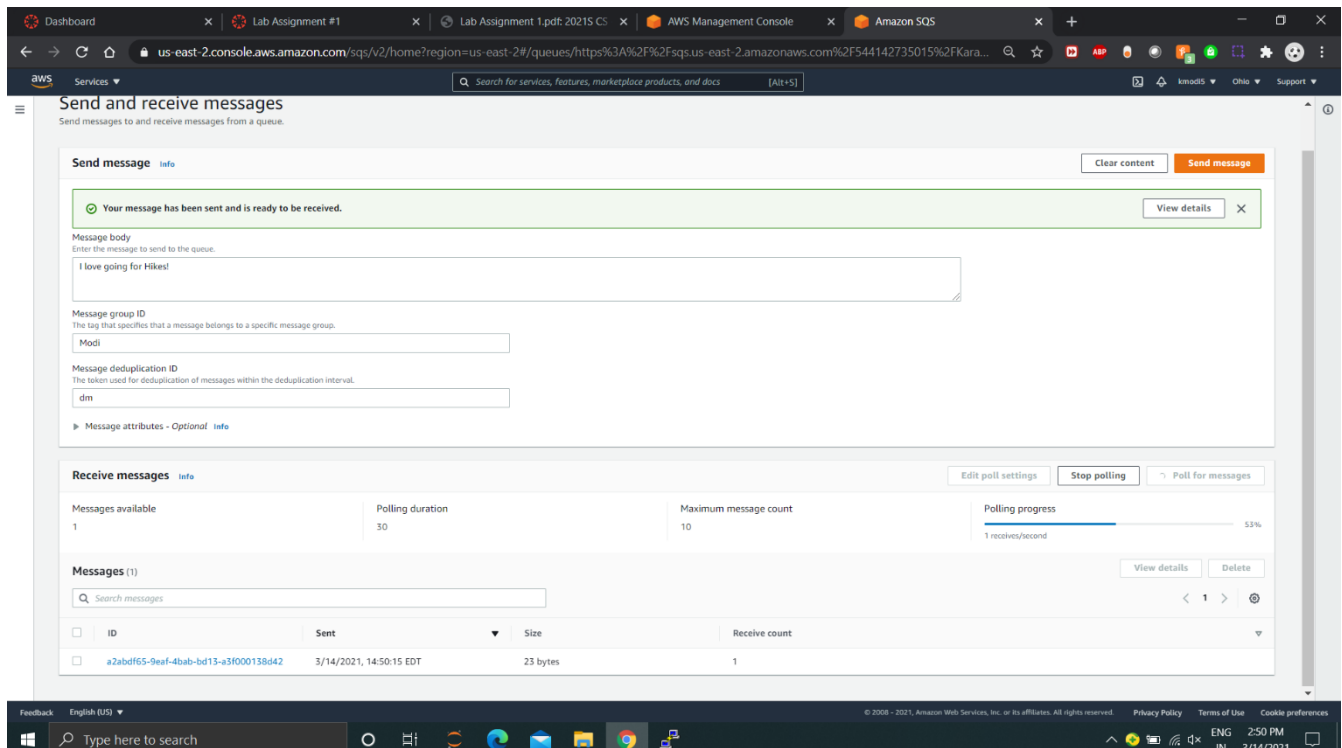
c. Create queue and select the parameters and name it as <NAME>.fifo



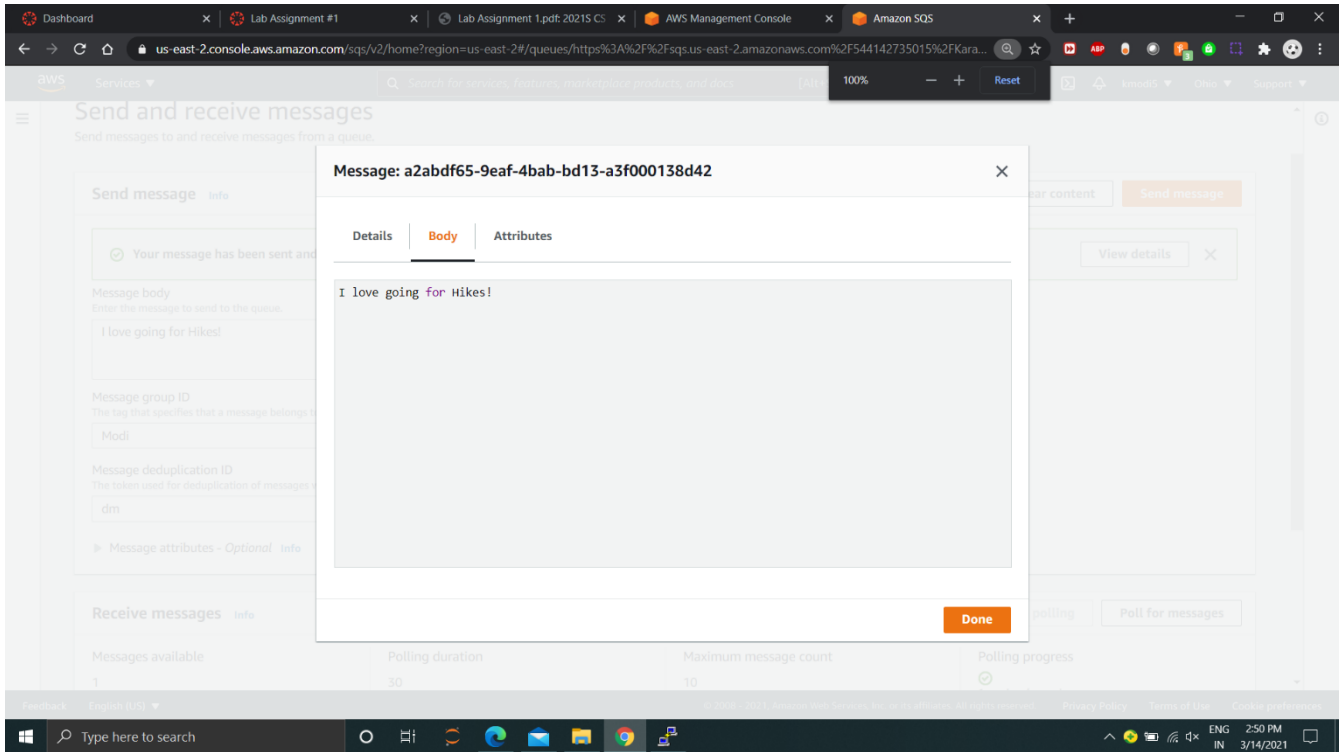
- d. Click on Create to finish creating a queue(in our case a FIFO queue) and click on send and receive messages to communicate messages.



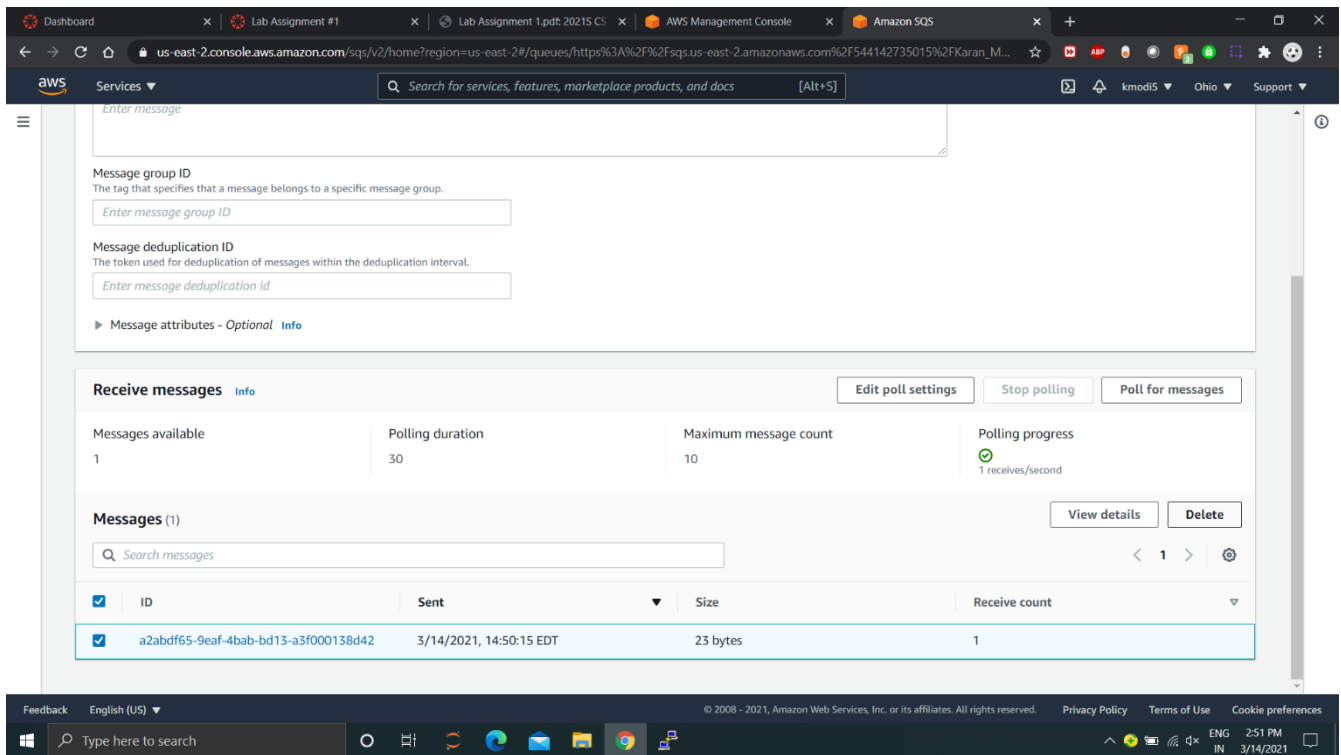
- e. Fill in the message body and other input boxes.



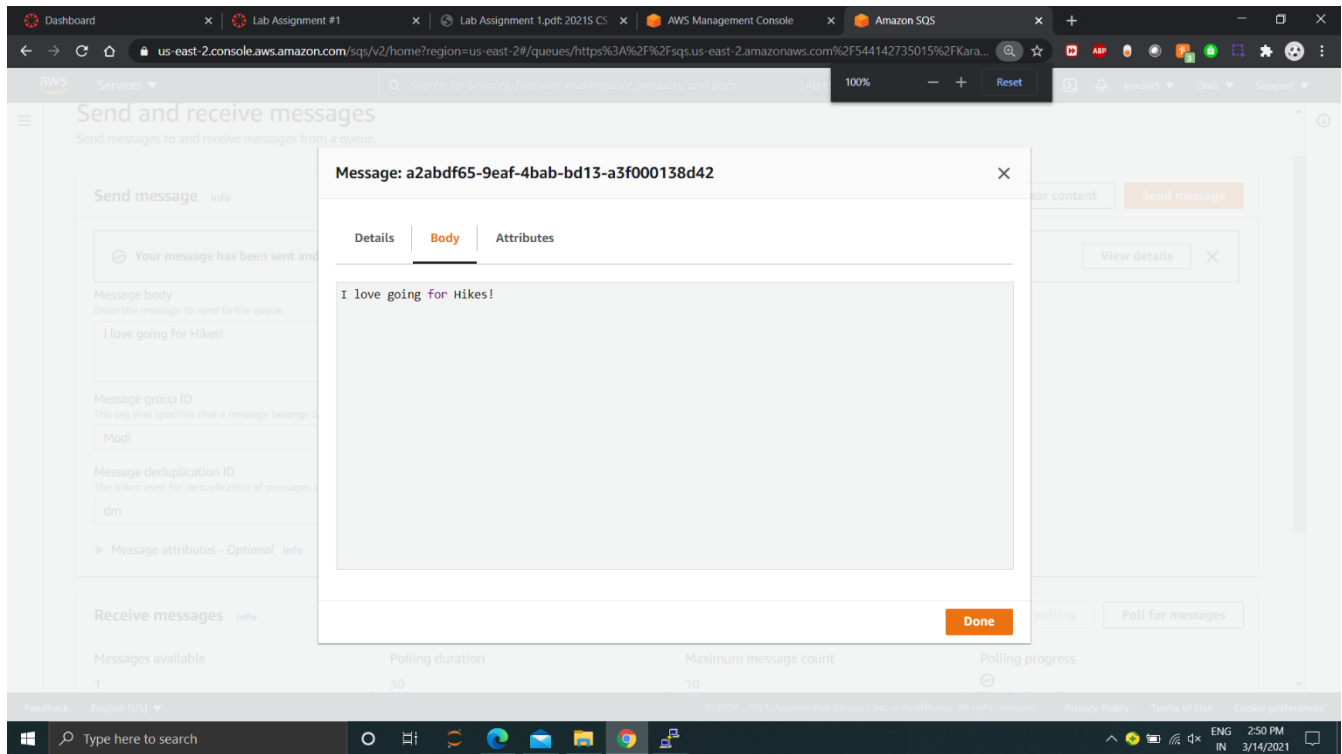
f. Fill the information and hit send message.



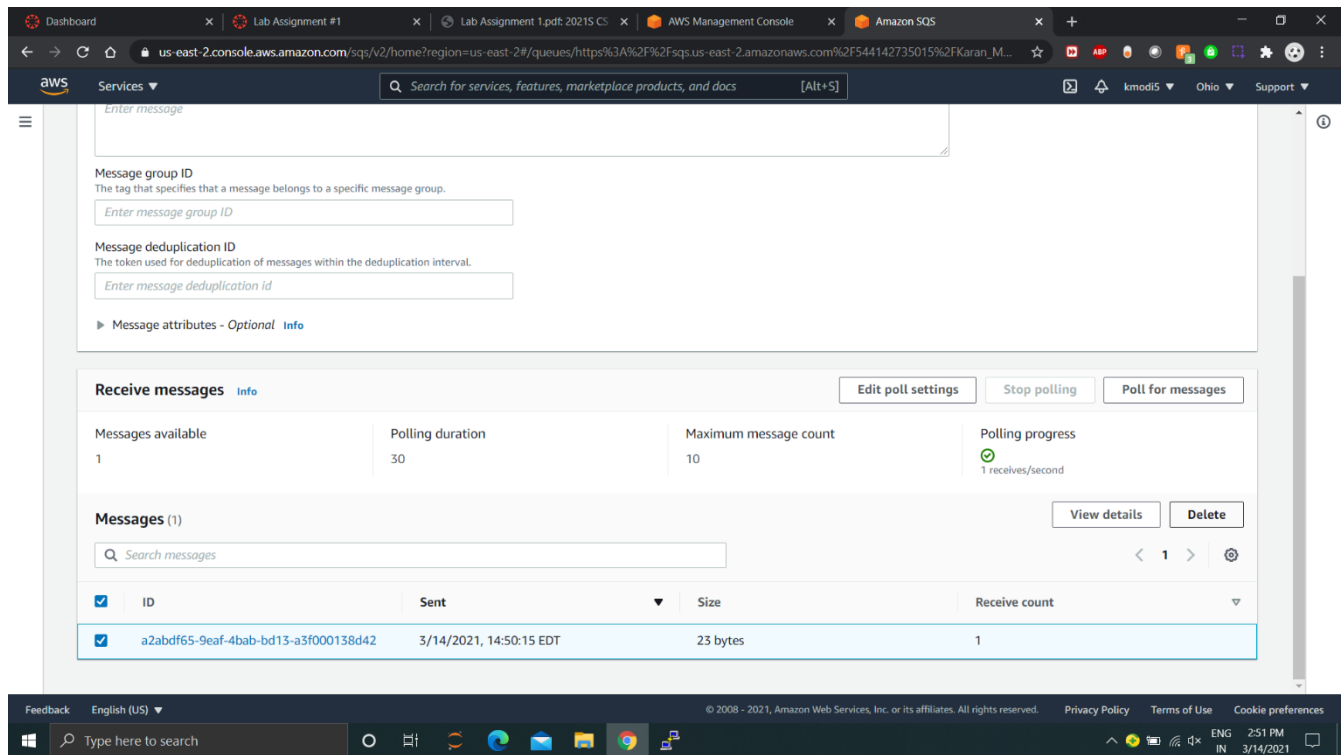
g. Poll for the messages in the queue



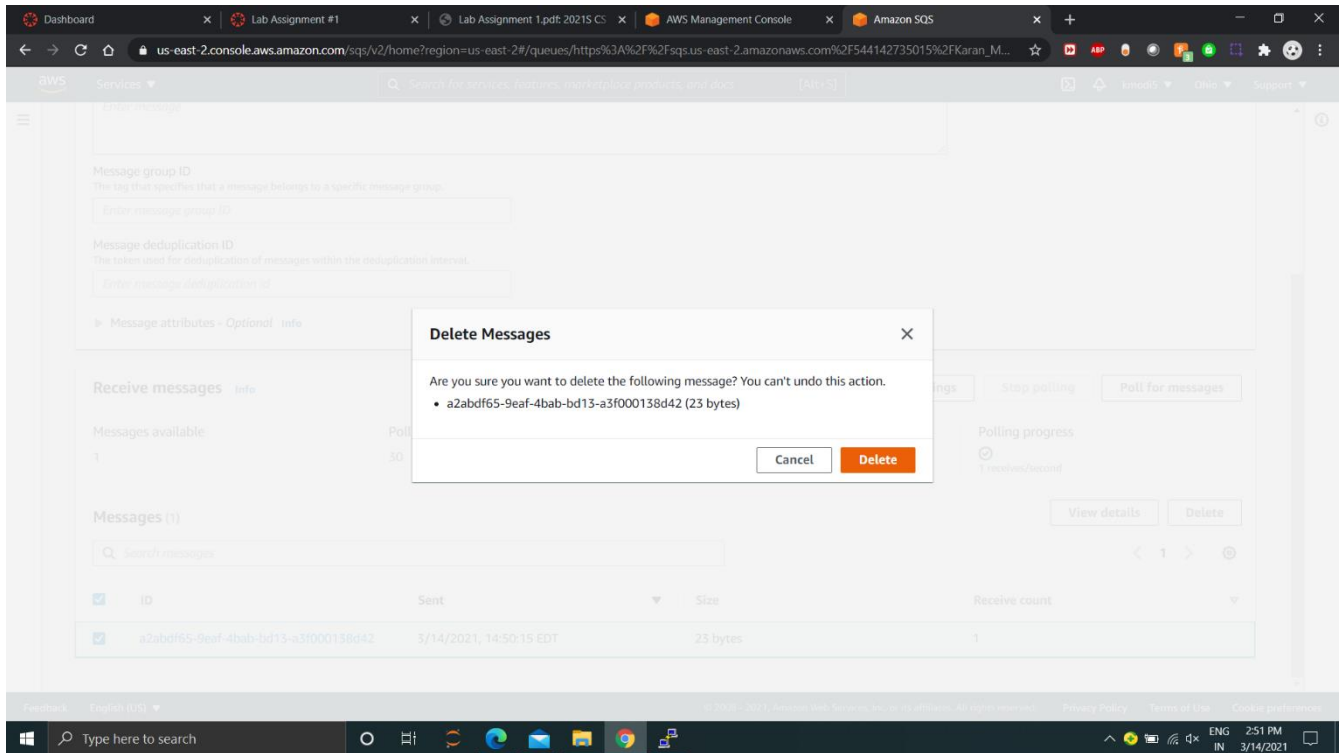
i. Poll for the messages in the queue and click to open it



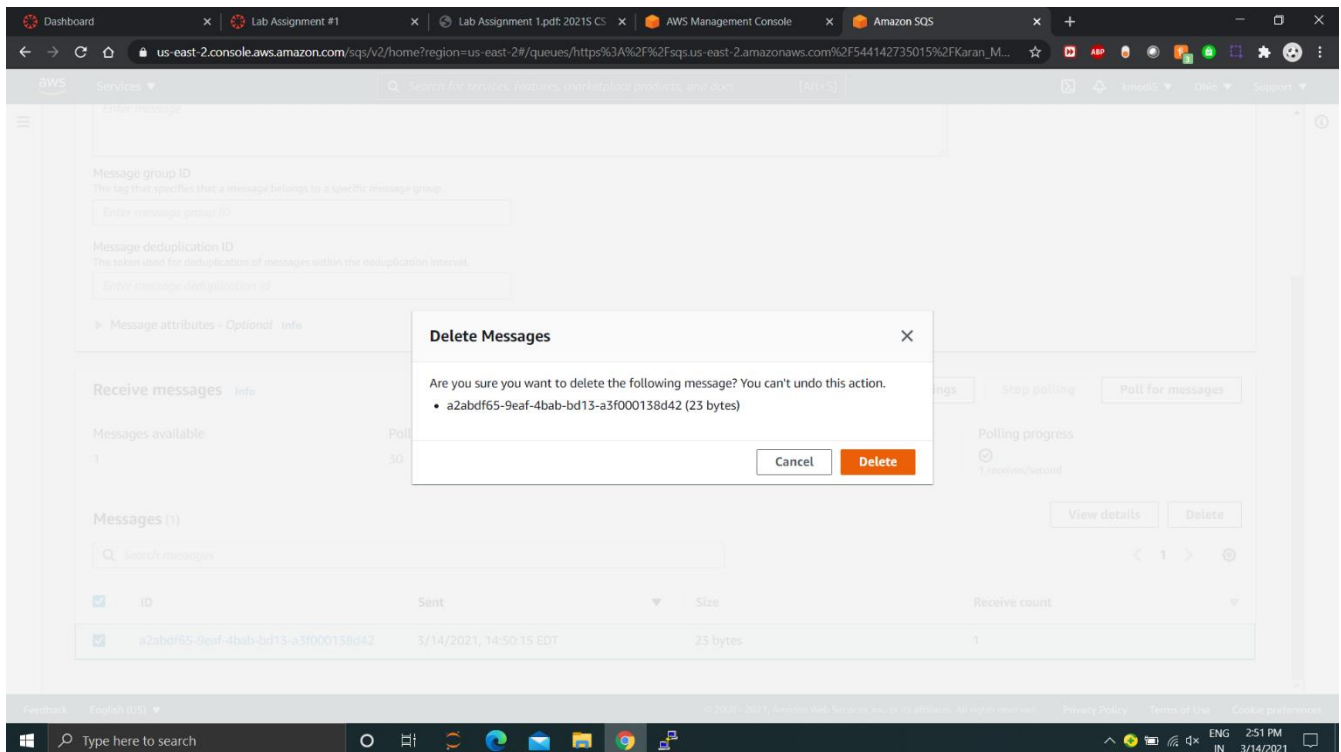
j. Check the message to be deleted



k. Delete the message



l. The message has been deleted from the queue



2. This section involves setting up a free Amazon EC2 instance (i.e., a virtual machine) and understanding some of its key networking properties. Although this is seemingly simple and straight-forward, there is much to read and learn here, so make sure you start working at once. The next lab assignment will build on what you will have achieved in this one.

The first part of the assignment is understanding the respective SLA, which Homework #2 had prepared you for. The second part is purely technical (and it will involve an independent learning as a follow-up to Lecture 4): After having created an EC2 instance, you will execute several systems commands, which will give you information on the networking set-up. In order to understand the results, you will need to learn the output resulted from invoking the commands.

Before your start, please read carefully the posting from the Course Assistant on how to create an AWS account so that you get the credit. If in doubt, please follow-up with the Course Assistant during her office hours.

Please make sure you have activated your Stevens Linux account. To request this use this link:

<https://sit.teamdynamix.com/TDClient/1865/Portal/Requests/ServiceCatalog?CategoryID=2880>

[Note: You may actually be able to use your own PC; however, you will most likely need to install additional software (e.g., SSH), and the effect of some commands may be different.]

Now, you need to review the following documents:

<http://aws.amazon.com/ec2/>

<http://docs.amazonwebservices.com/AWSEC2/2009-11-30/GettingStartedGuide/>

Then visit <http://aws.amazon.com/ec2/> and click “Sign Up Now” button to setup an account. Again, make sure that you understand what you need to do to keep this experiment free of charge.

At this point, please proceed to creating an EC2 instance with this Amazon Machine Image (AMI): Basic 64-bit Amazon Linux AMI. Once it is running, log into it and execute the following five commands:

1. `uname -a`
2. `whoami`

3. df -h
4. ifconfig -a
5. netstat.

Now you need to understand what these commands do with the parameters chosen (by reading the respective part of the system manual—obtainable by executing `man .`). To get the meaning of the output, you will use your knowledge of IP networking:

1. You have learned about both the class-based IP addressing scheme and the Classless Inter-Domain Routing (CIDR) [for the detail see RFC4632 (<http://tools.ietf.org/html/rfc4632>)];
2. To understand how the IP addresses are mapped into Layer 2 addresses, please read RFC 826 <http://tools.ietf.org/html/rfc826>; and
3. To understand the parameters related to the dynamic host configuration, please read <http://tools.ietf.org/html/rfc2131>.

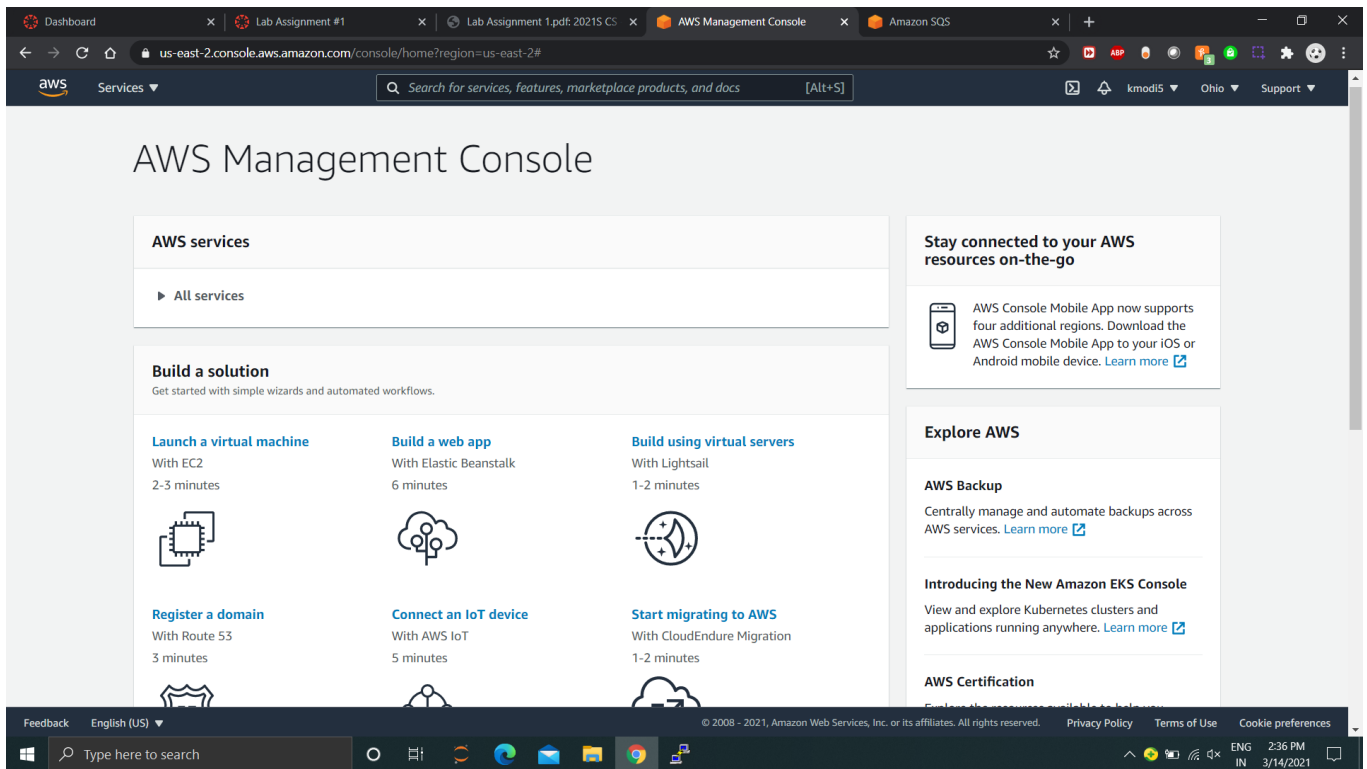
You must submit a report documenting

- 1) all the steps that you have executed in setting up your account and
- 2) the results of the command execution along with the explanation of the parameters obtained in the process.

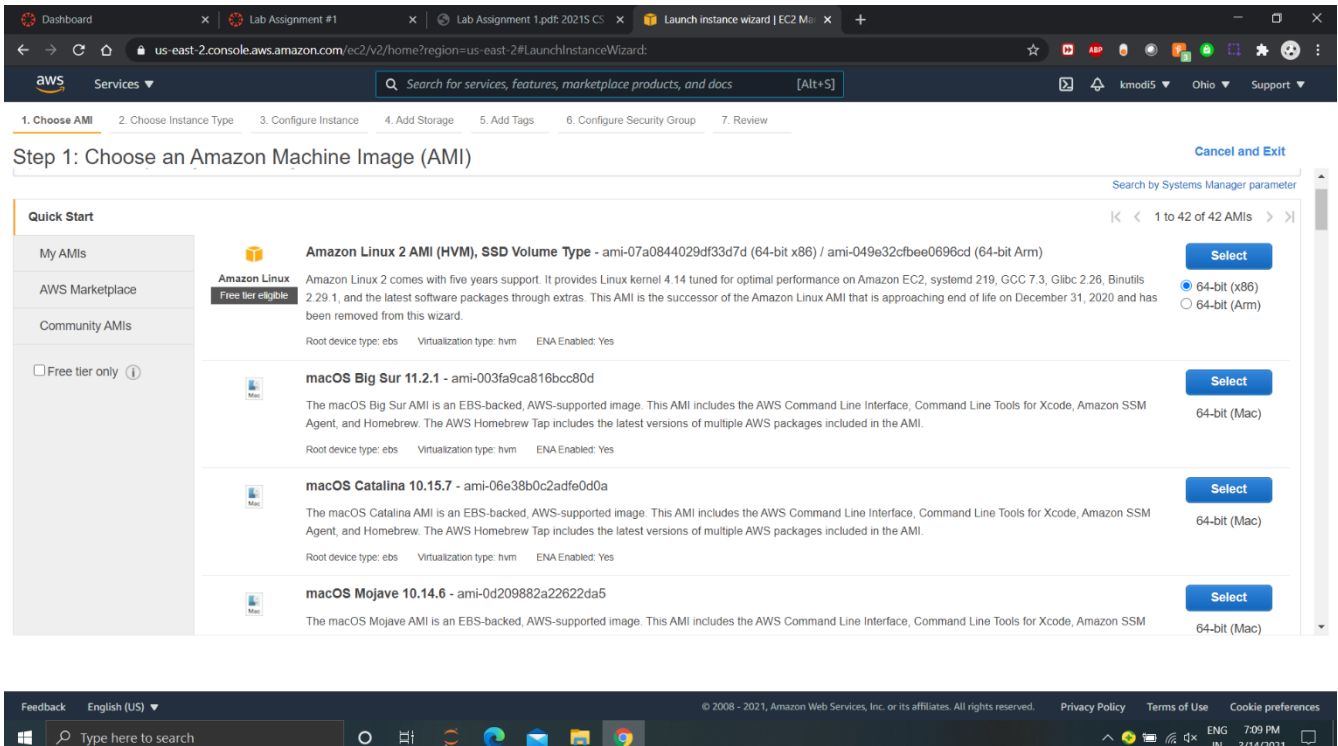
After you finish your assignment, make sure you shut down the instance you have created

ANS.

a. Go into AWS Console and Launch a virtual machine



b. Select Amazon Linux 2 AMI as Amazon Machine Image which eligible for Free Tier



c. Choose an Instance Type and select t2.micro which is eligible for free tier

Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn more](#) about instance types and how they can meet your computing needs.

Filter by: All instance families Current generation Show/Hide Columns

Currently selected: t2.micro (- ECUs, 1 vCPUs, 2.5 GHz, -, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
<input type="checkbox"/>	t2	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	t2	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.small	1	2	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.large	2	8	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.xlarge	4	16	EBS only	-	Moderate	Yes
<input type="checkbox"/>	t2	t2.2xlarge	8	32	EBS only	-	Moderate	Yes
<input type="checkbox"/>	t3	t3.nano	2	0.5	EBS only	Yes	Up to 5 Gigabit	Yes

Cancel Previous Review and Launch Next: Configure Instance Details

d. Review your Instance and launch it to start the instant

Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

Improve your instances' security. Your security group, launch-wizard-1, is open to the world.

Your instances may be accessible from any IP address. We recommend that you update your security group rules to allow access from known IP addresses only. You can also open additional ports in your security group to facilitate access to the application or service you're running, e.g., HTTP (80) for web servers. [Edit security groups](#)

AMI Details [Edit AMI](#)

Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-07a0844029df33d7d

Free tier eligible

Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras. This AMI is the successor of the Amazon Linux AMI that is a...

Root Device Type: ebs Virtualization type: hvm

Instance Type [Edit instance type](#)

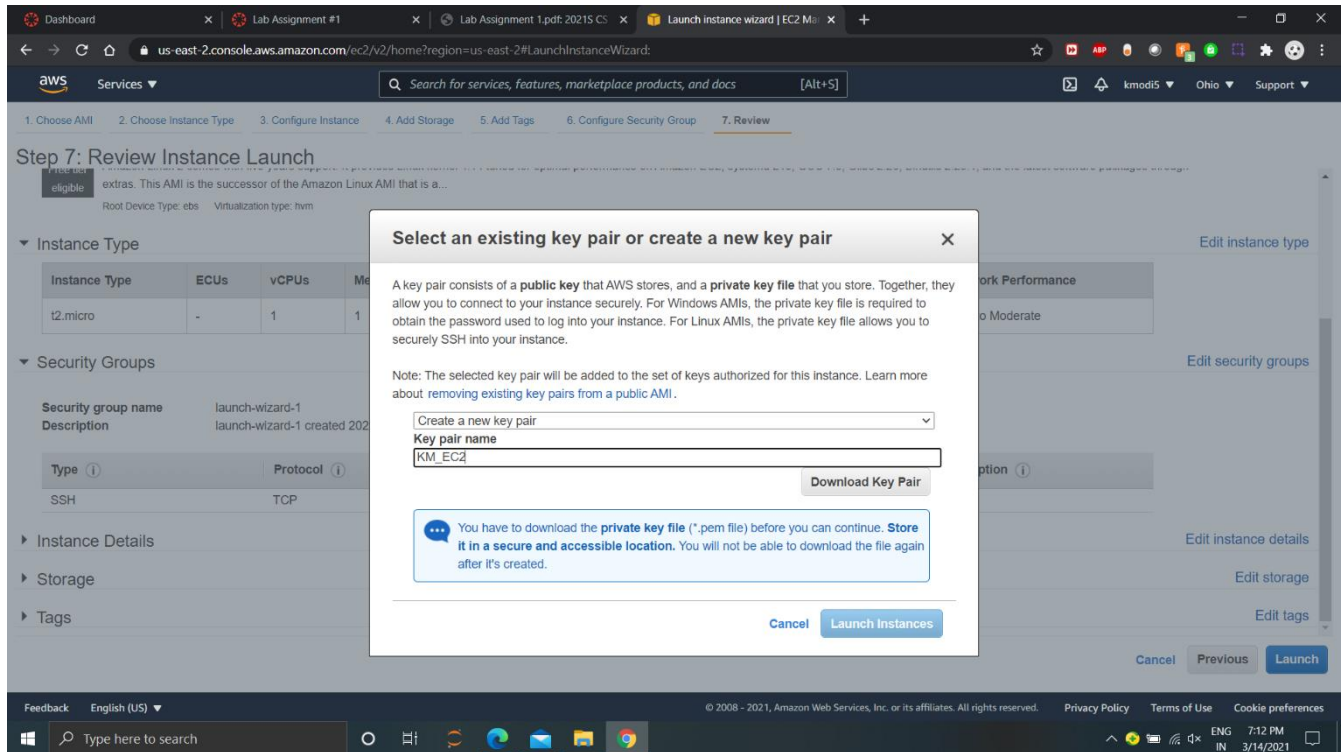
Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	-	1	1	EBS only	-	Low to Moderate

Security Groups [Edit security groups](#)

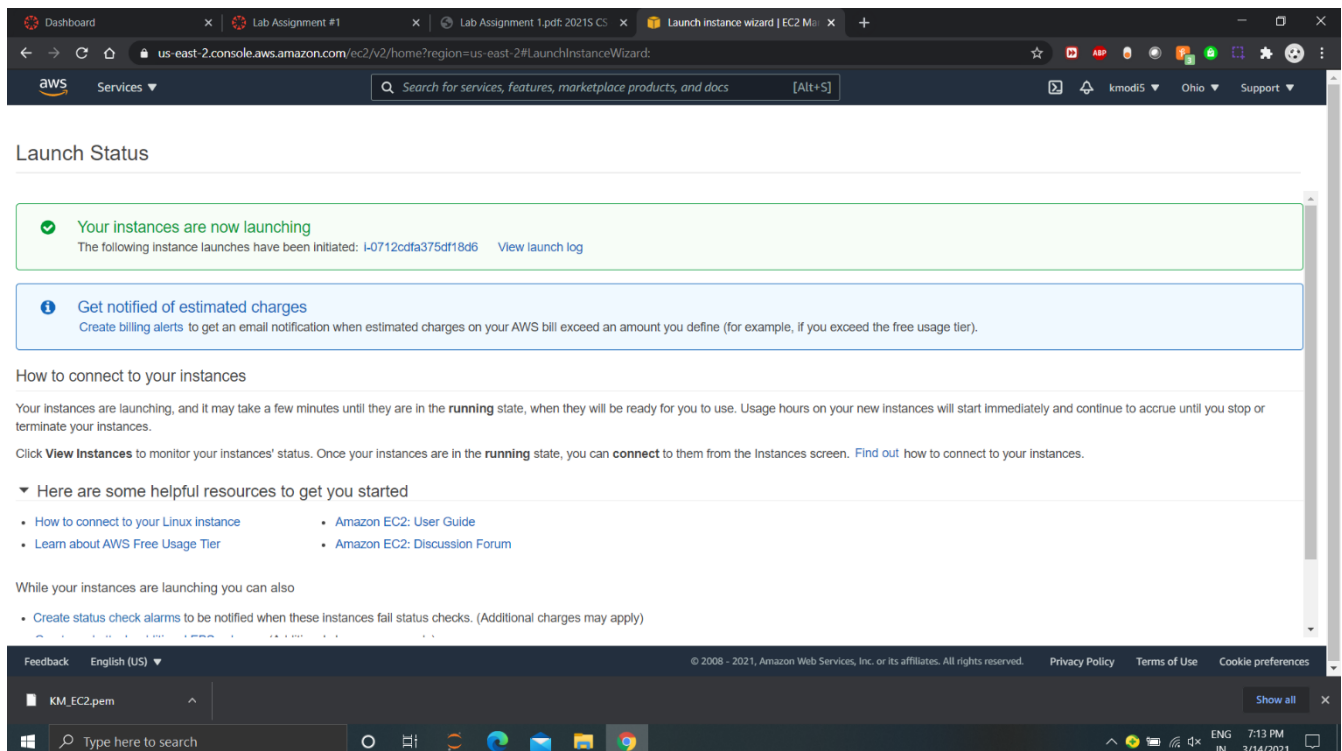
Security group name	Description
launch-wizard-1	launch-wizard-1 created 2021-03-14T10:44:48-04:00

Cancel Previous Launch

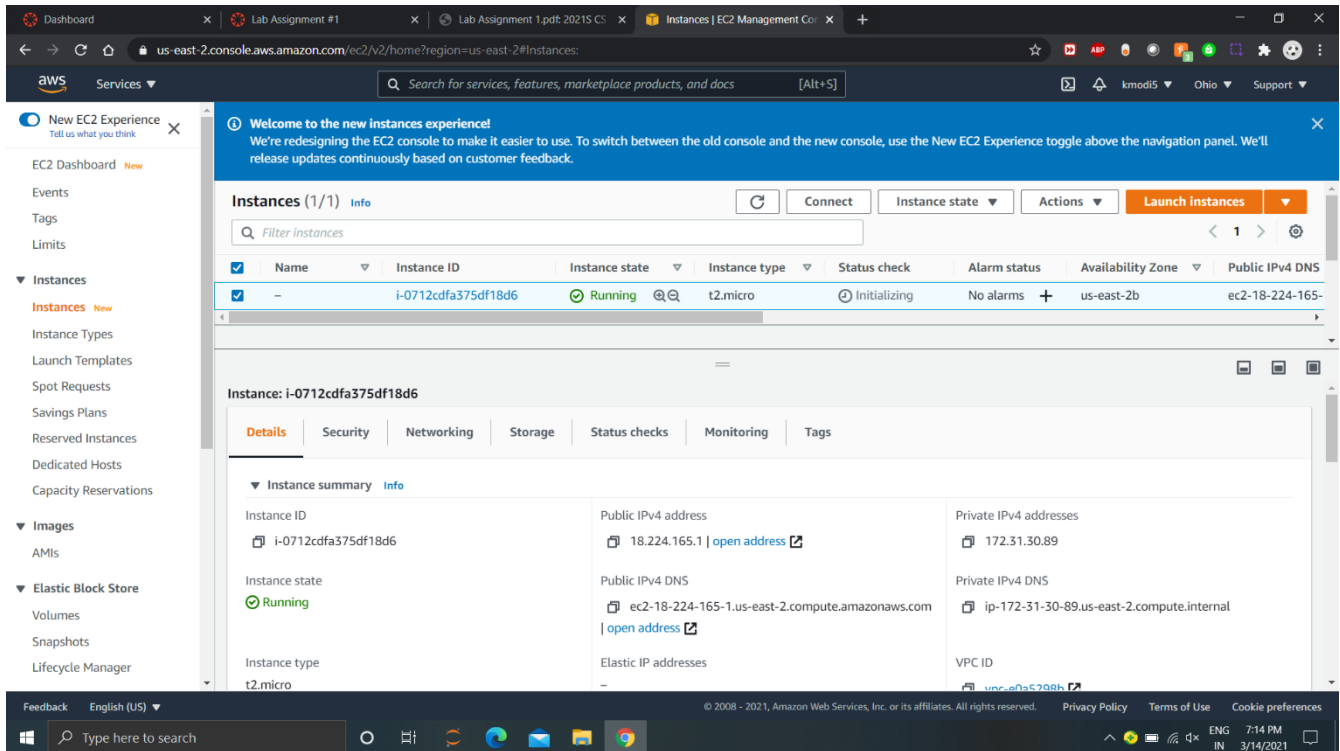
- e. Select Create a new key pair(or use an existing key pair) and name it. Download the <NAME>.pem.



- f. Launch the instance and it should be initiated.



- g. The instance will start and will be in running state in 1-2 mins. Review the details.



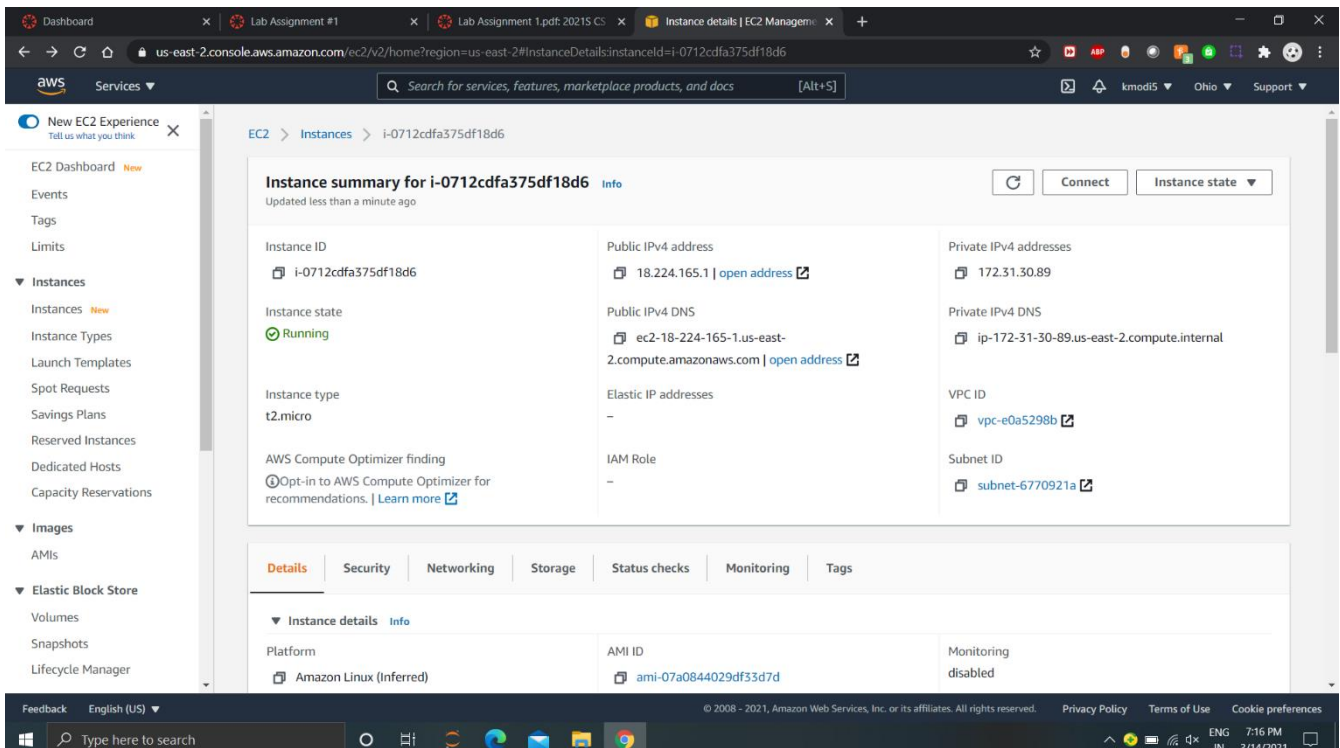
The screenshot shows the AWS Management Console for the 'us-east-2' region. The 'Instances' page is active, displaying a list of instances. The instance 'i-0712cdfa375df18d6' is highlighted, showing its state as 'Running'. The console also displays a detailed view of the instance, including its ID, state, type, and various addresses.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
-	i-0712cdfa375df18d6	Running	t2.micro	Initializing	No alarms	us-east-2b	ec2-18-224-165-

Instance: i-0712cdfa375df18d6

Instance summary	
Instance ID	Public IPv4 address
i-0712cdfa375df18d6	18.224.165.1 open address
Instance state	Public IPv4 DNS
Running	ec2-18-224-165-1.us-east-2.compute.amazonaws.com open address
Instance type	Elastic IP addresses
t2.micro	-

- h. Connect to the instance and select a medium



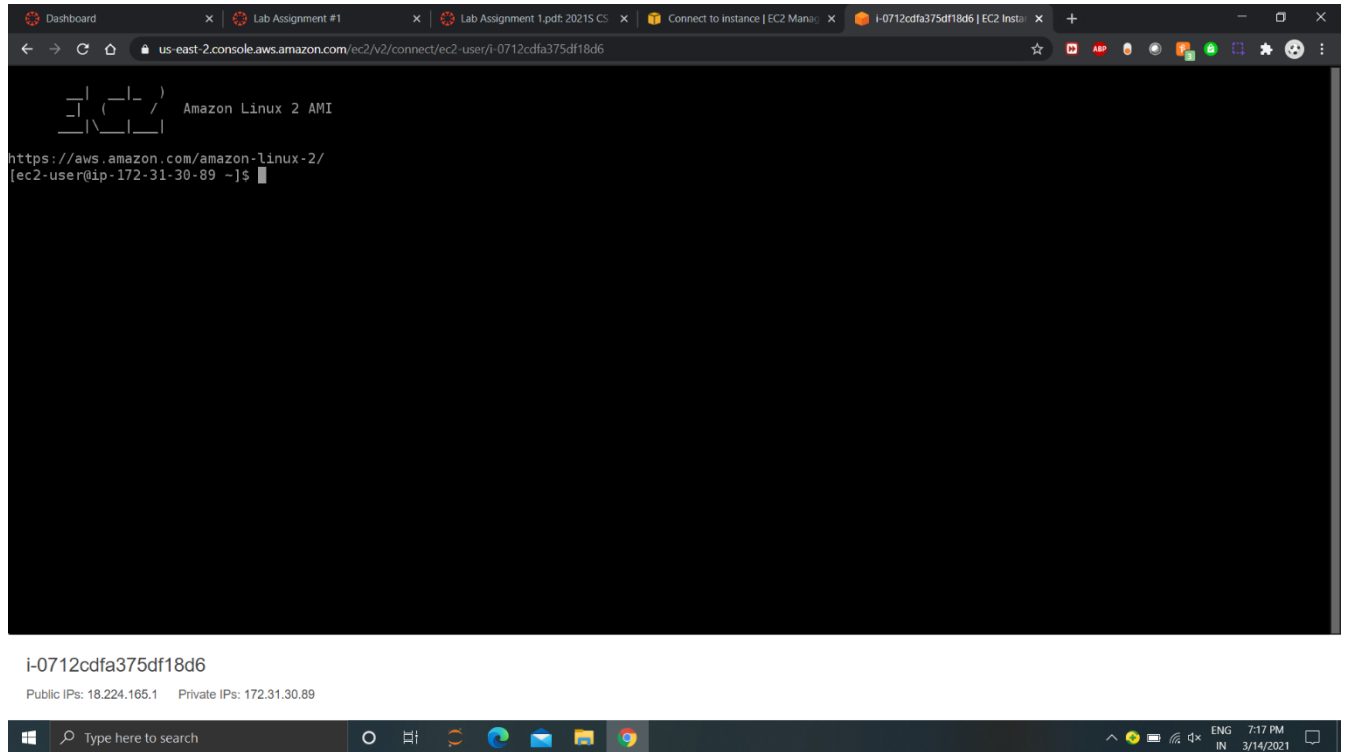
The screenshot shows the AWS Management Console for the 'us-east-2' region, specifically the 'Instance details' page for the instance 'i-0712cdfa375df18d6'. The page displays the instance summary, including its ID, state, type, and various addresses.

Instance summary for i-0712cdfa375df18d6	
Instance ID	Public IPv4 address
i-0712cdfa375df18d6	18.224.165.1 open address
Instance state	Public IPv4 DNS
Running	ec2-18-224-165-1.us-east-2.compute.amazonaws.com open address
Instance type	Elastic IP addresses
t2.micro	-

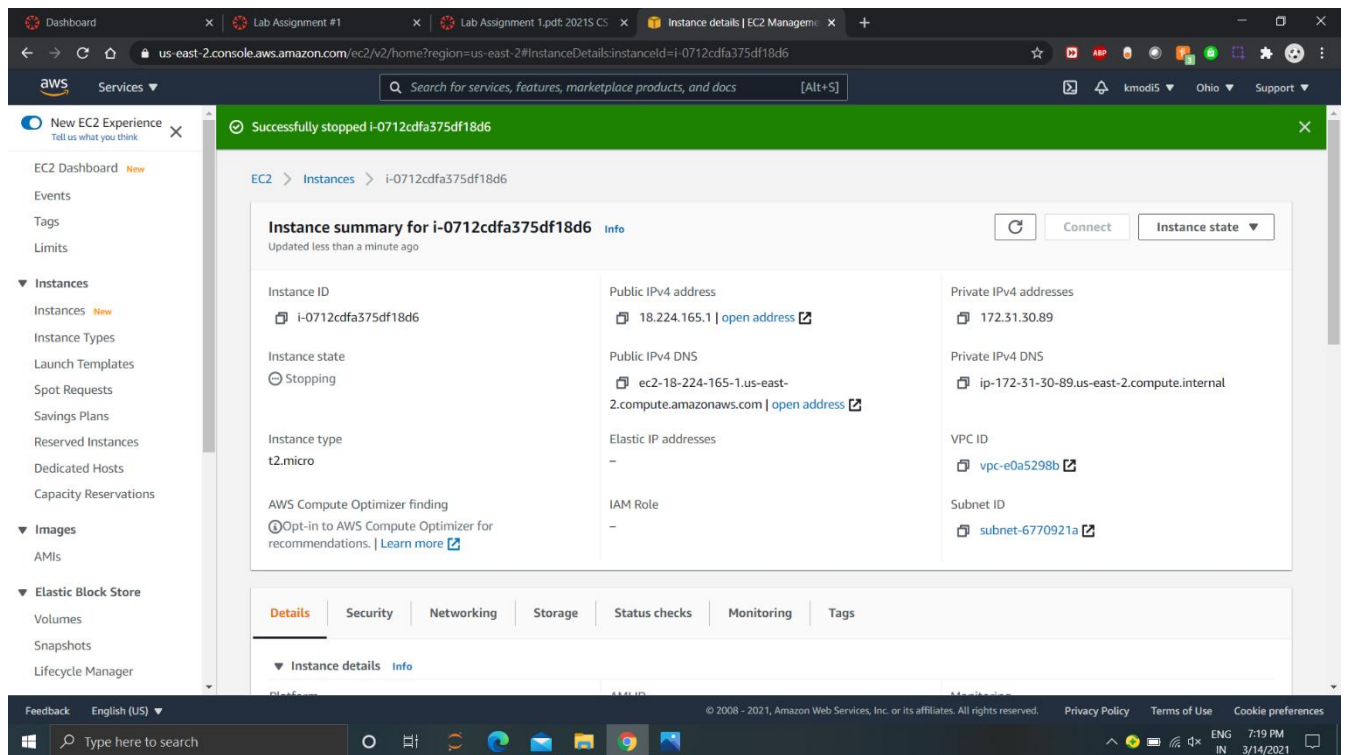
Instance details

Platform	AMI ID	Monitoring
Amazon Linux (Inferred)	ami-07a0844029df33d7d	disabled

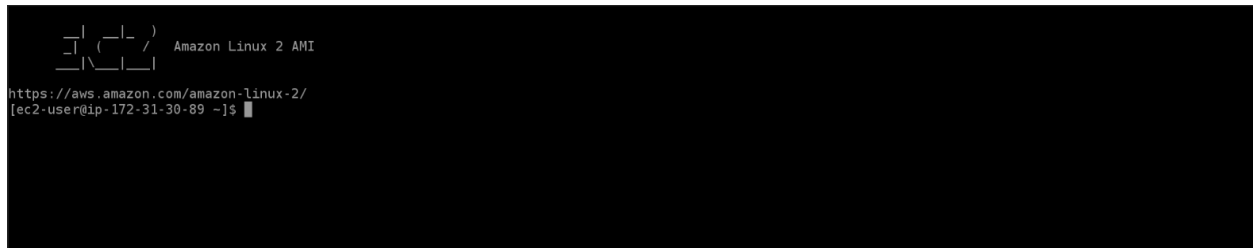
- i. The terminal should launch. Now run the commands.



- j. Stop the instance when the tasks are completed.



AWS Terminal:



```

  _|_  _|_  )
 _|_  _|_  |
 _|_  _|_  |
Amazon Linux 2 AMI
https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-30-89 ~]$
```

Commands:

1. uname:

uname is a command-line utility that prints basic information about the operating system name and system hardware.

Usually, the uname command is used with the -a option to print all available information.

The other options are as follows:

- -s, (--kernel-name) - Prints the kernel name.
- -n, (--nodename) - Prints the system's node name (hostname). This is the name the system uses when communicating over the network. When used with the -n option, uname produces the same output as the hostname command.
- -r, (--kernel-release) - Prints the kernel release.
- -v, (--kernel-version) - Prints the kernel version.
- -m, (--machine) - Prints the name of the machine's hardware name.
- -p, (--processor) - Prints the architecture of the processor.
- -i, (--hardware-platform) - Prints the hardware platform.
- -o, (--operating-system) - Print the name of the operating system. On Linux systems that is "GNU/Linux"
- -a, (--all) - When the -a option is used, uname behaves the same as if the -snrvmo options have been given.

When invoked without any options, uname prints the kernel name, as if the -s option had been specified.

```

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-30-89 ~]$ uname -a
Linux ip-172-31-30-89.us-east-2.compute.internal 4.14.219-164.354.amzn2.x86_64 #1 SMP Mon Feb 22 21:18:39 UTC 2021 x86_64 x86_64 x86_64 GNU/Linux
```

(references: [Uname Command in Linux](#) | [Linuxize](#))

2. whoami:

whoami command is used both in Unix Operating System and as well as in Windows Operating System.

It is basically the concatenation of the strings "who","am","i" as whoami.

It displays the username of the current user when this command is invoked.

It is similar as running the id command with the options -un.

```
[ec2-user@ip-172-31-30-89 ~]$ whoami
ec2-user
```

(references: [whoami command in Linux with example - GeeksforGeeks](#))

3. df -h:

On Linux and Unix operating systems, the df command is used to get a detailed report on the system's disk space usage.

By default, the df command shows the disk space in 1-kilobyte blocks and the size of used and available disk space in kilobytes.

To display information about disk drives in human-readable format (kilobytes, megabytes, gigabytes and so on), invoke the df command with the -h option.

Each line includes the following columns:

- "Filesystem" - The name of the filesystem.
- "Size" - The size of the filesystem blocks.
- "Used" - The used space in 1K blocks.
- "Available" - The available space in 1K blocks.
- "Use%" - The percentage of used space.
- "Mounted on" the directory on which the filesystem is mounted.

```
[ec2-user@ip-172-31-30-89 ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        482M   0    482M   0% /dev
tmpfs           492M   0    492M   0% /dev/shm
tmpfs           492M 400K   492M   1% /run
tmpfs           492M   0    492M   0% /sys/fs/cgroup
/dev/xvda1      8.0G  1.4G   6.7G  18% /
tmpfs           99M   0     99M   0% /run/user/1000
```

(references: [Df Command in Linux \(Check Disk Space\) | Linuxize](#))

4. Ifconfig -a:

ifconfig stands for "interface configuration." It is used to view and change the configuration of the network interfaces on your system.

Running the ifconfig command with no arguments, displays information about all network interfaces currently in operation.

If we would like to view the configuration of all network interfaces on the system (not just the ones that are currently active), you can specify the -a option. This produces output similar to running ifconfig, but if there are any inactive interfaces on the system, their configuration is also shown.

Here, eth0, lo and wlan0 are the names of the active network interfaces on the system.

- eth0 is the first Ethernet interface. (Additional Ethernet interfaces would be named eth1, eth2, etc.) This type of interface is usually a NIC connected to the network by a category 5 cable.
- lo is the loopback interface. This is a special network interface that the system uses to communicate with itself.
- wlan0 is the name of the first wireless network interface on the system. Additional wireless interfaces would be named wlan1, wlan2, etc.

```
[ec2-user@ip-172-31-30-89 ~]$ ifconfig -a
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 9001
    inet 172.31.30.89 netmask 255.255.240.0 broadcast 172.31.31.255
    inet6 fe80::4d7:54ff:feaf:df3c prefixlen 64 scopeid 0x20<link>
    ether 06:d7:54:af:df:3c txqueuelen 1000 (Ethernet)
    RX packets 39350 bytes 54520311 (51.9 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 7201 bytes 442862 (432.4 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 8 bytes 648 (648.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 8 bytes 648 (648.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

(references: [Linux ifconfig command help and examples \(computerhope.com\)](https://www.computerhope.com/linux/ifconfig-command-help-and-examples))

5. netstat:

netstat command is a useful command to reveal a network status of your system. It allows a system administrator to keep track of any network connections by querying their status, origin and destination.

Furthermore, netstat is the all-in-one networking monitoring tool as it can also be used to display route tables, interface statistics, masquerade connections, as well as multicast memberships.

- # netstat -st: Show a TCP protocol specific summary
- # netstat -r: Display routing table
- # netstat -ie: The equivalent to default ifconfig command to list all active network interfaces
- # netstat -antc: Continuously monitor all TCP connection. use -u for UDP.
- # netstat -ltp: Display all open TCP ports along with PID and program.
- # netstat -atp | grep ssh: Show all SSH connection along with user name, program and corresponding PID.
- # netstat -atnp | grep 443: Show all connection on port 443 along with user ID, program and corresponding PID.
- # netstat -s -w: Display raw network statistics

```

unix 2      [ ]          DGRAM          20623
unix 3      [ ]          STREAM         CONNECTED      19199          /run/systemd/journal/stdout
unix 3      [ ]          STREAM         CONNECTED      17468
unix 3      [ ]          STREAM         CONNECTED      15257          /run/systemd/journal/stdout
unix 3      [ ]          STREAM         CONNECTED      17470
unix 3      [ ]          STREAM         CONNECTED      17494
unix 3      [ ]          STREAM         CONNECTED      15369          /run/dbus/system_bus_socket
unix 3      [ ]          STREAM         CONNECTED      15210
unix 3      [ ]          STREAM         CONNECTED      17465
unix 3      [ ]          STREAM         CONNECTED      15256
unix 3      [ ]          STREAM         CONNECTED      17464
unix 3      [ ]          STREAM         CONNECTED      17497
unix 3      [ ]          STREAM         CONNECTED      17467
unix 3      [ ]          STREAM         CONNECTED      17498
unix 3      [ ]          STREAM         CONNECTED      17488
unix 2      [ ]          DGRAM          15330
unix 3      [ ]          STREAM         CONNECTED      17461
unix 3      [ ]          STREAM         CONNECTED      17489
unix 3      [ ]          STREAM         CONNECTED      20630
unix 3      [ ]          STREAM         CONNECTED      15366
unix 2      [ ]          DGRAM          18734
unix 2      [ ]          DGRAM          17394
unix 3      [ ]          STREAM         CONNECTED      17462
unix 2      [ ]          DGRAM          17555
unix 3      [ ]          STREAM         CONNECTED      17500
unix 3      [ ]          STREAM         CONNECTED      18665          /run/systemd/journal/stdout
unix 3      [ ]          STREAM         CONNECTED      19198
unix 3      [ ]          STREAM         CONNECTED      18664
unix 3      [ ]          STREAM         CONNECTED      17456
unix 3      [ ]          STREAM         CONNECTED      19540          /var/lib/amazon/ssm/ipc/health
unix 3      [ ]          STREAM         CONNECTED      17459
unix 2      [ ]          DGRAM          17548
unix 3      [ ]          STREAM         CONNECTED      17458

```

(references: [Learning Linux Commands: netstat - LinuxConfig.org](https://www.linuxconfig.org/en/2018/05/20/netstat.html))