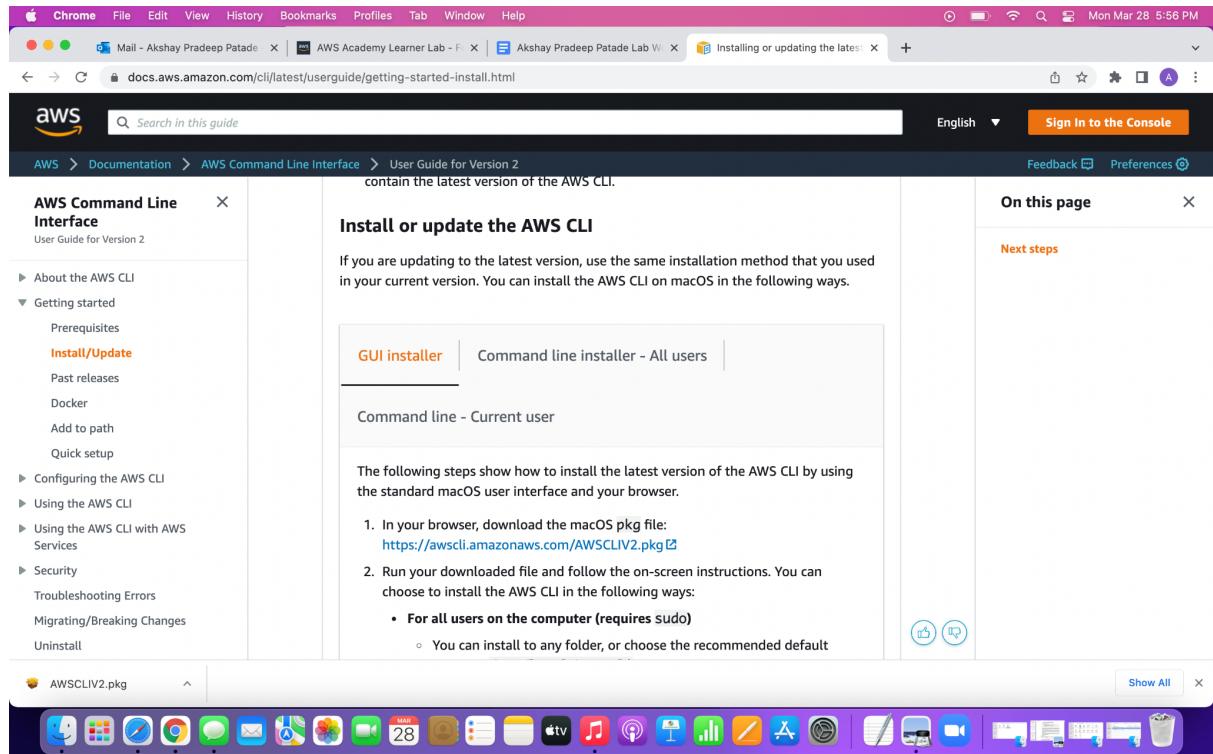


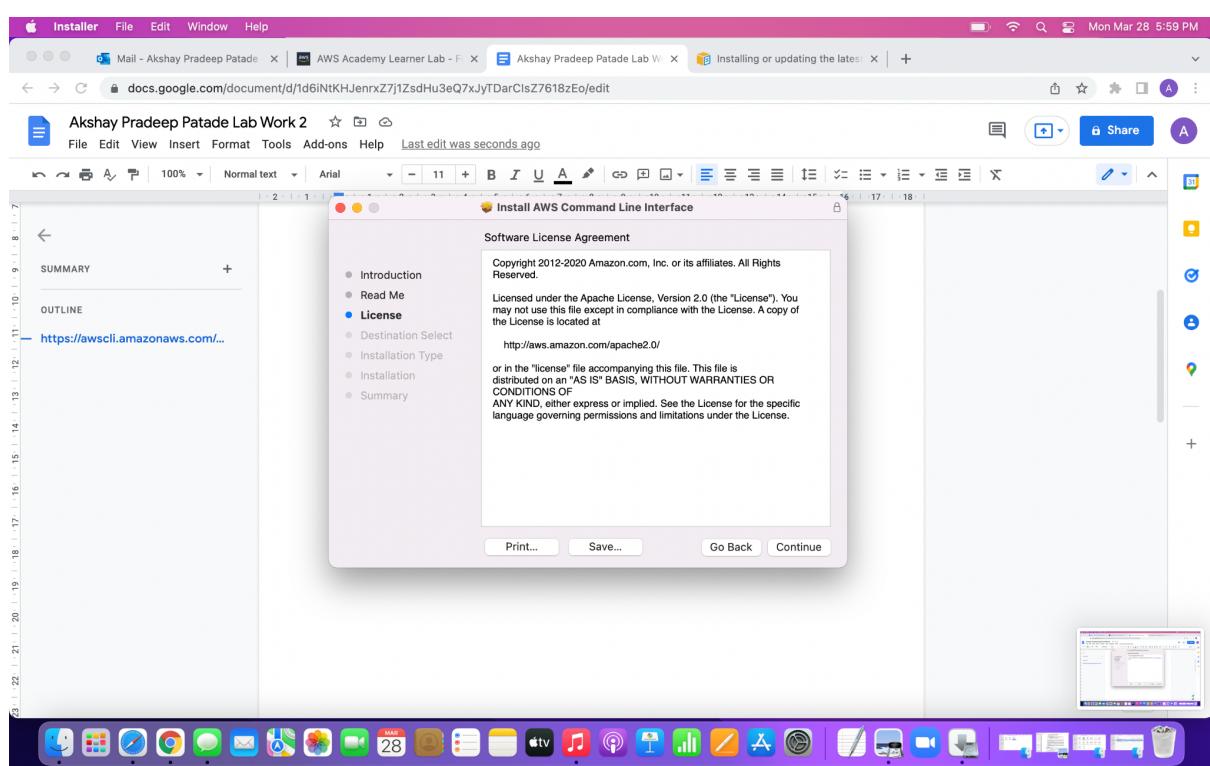
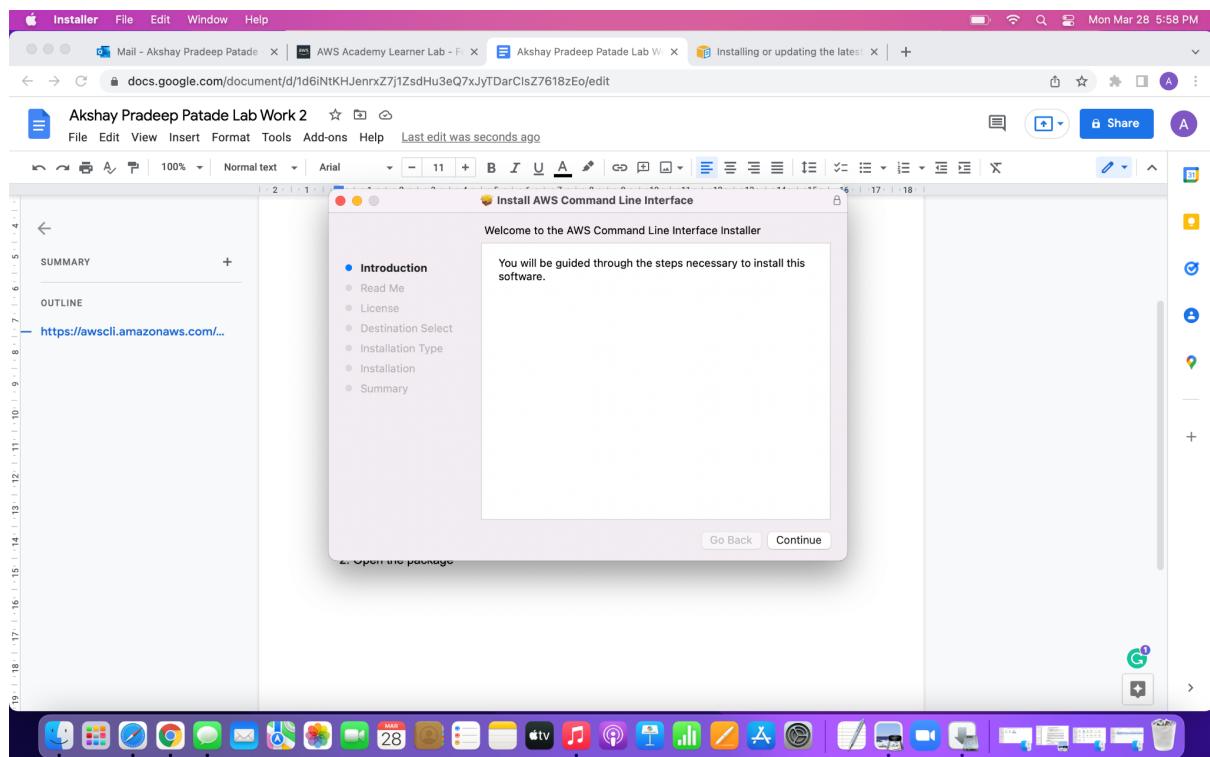
**In order to create and launch EC2 instances using the command line, we have to first install Amazon CLI.**

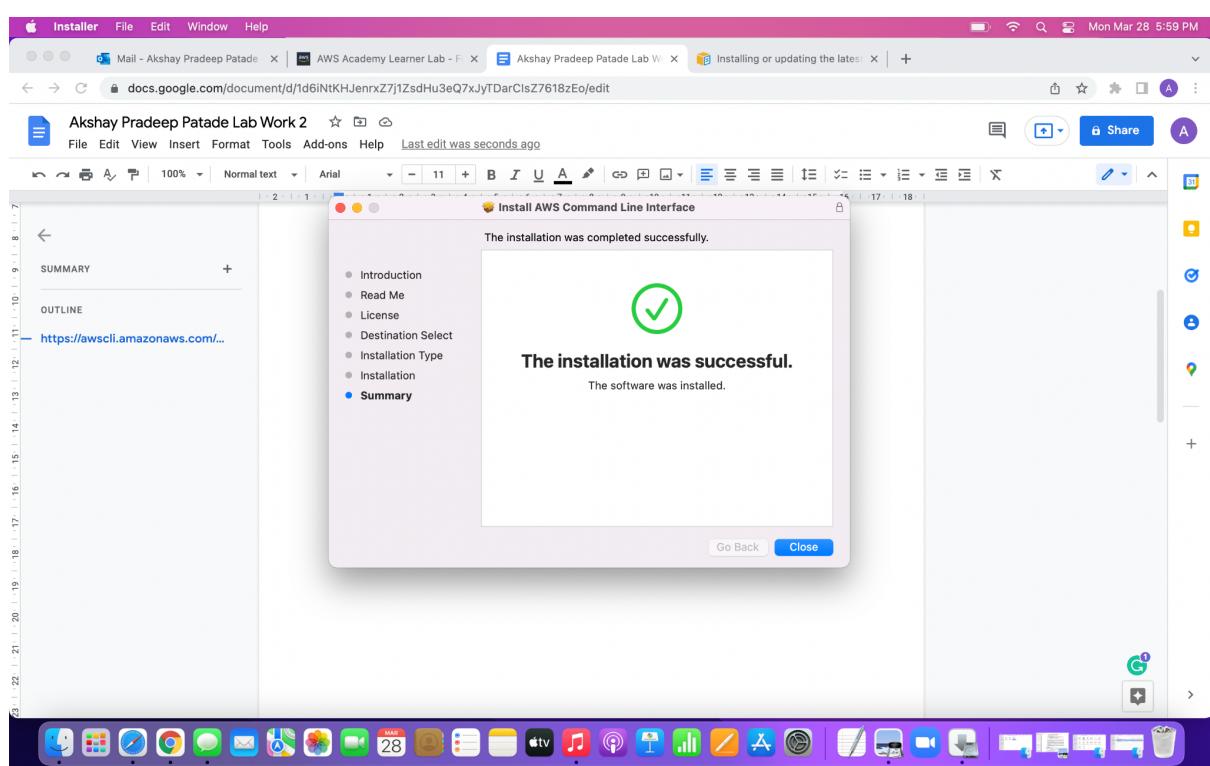
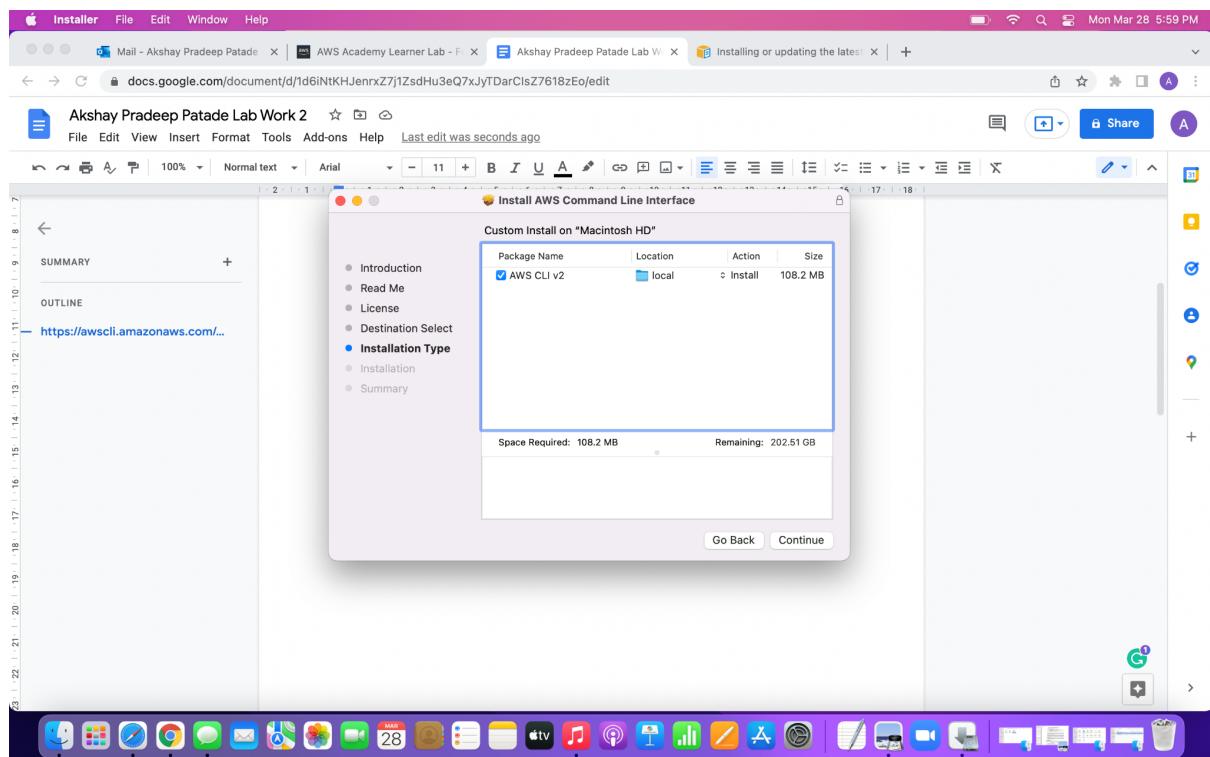
1. Go to this website and select the URL mentioned below.



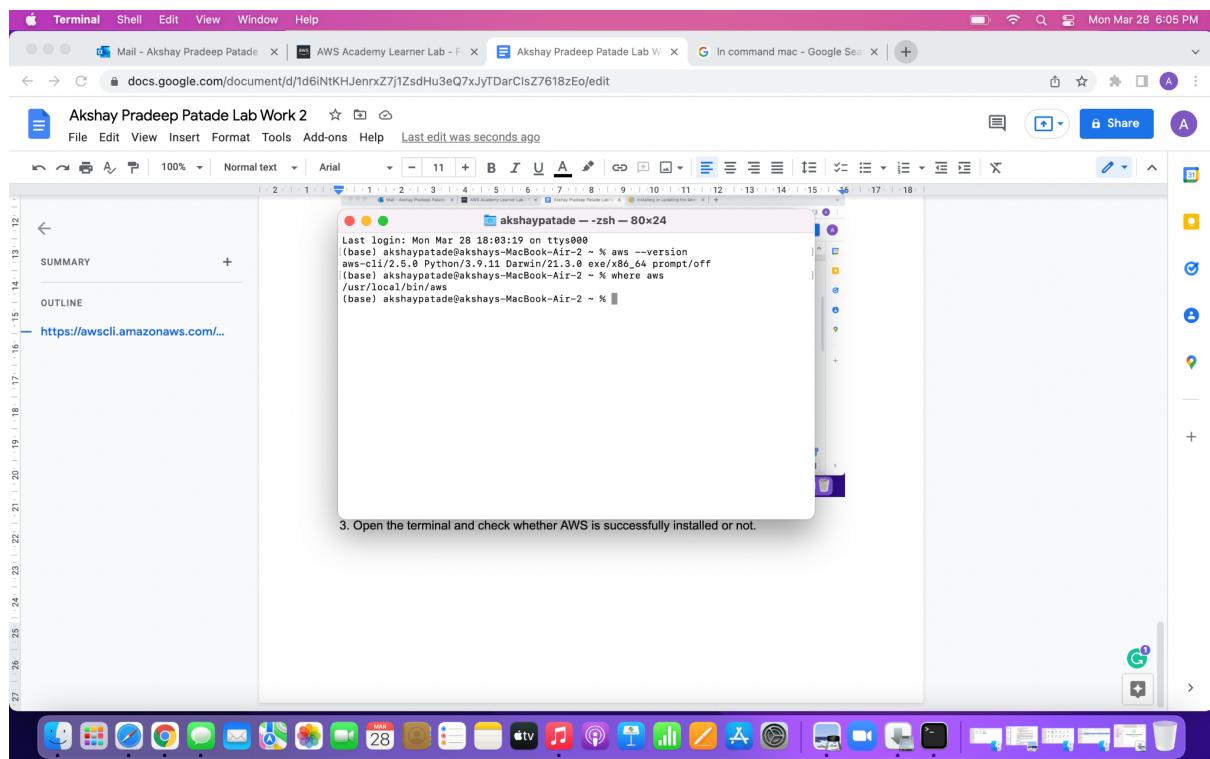
<https://awscli.amazonaws.com/AWSCLIV2.pkg>

2. Open the package and just click on continue.

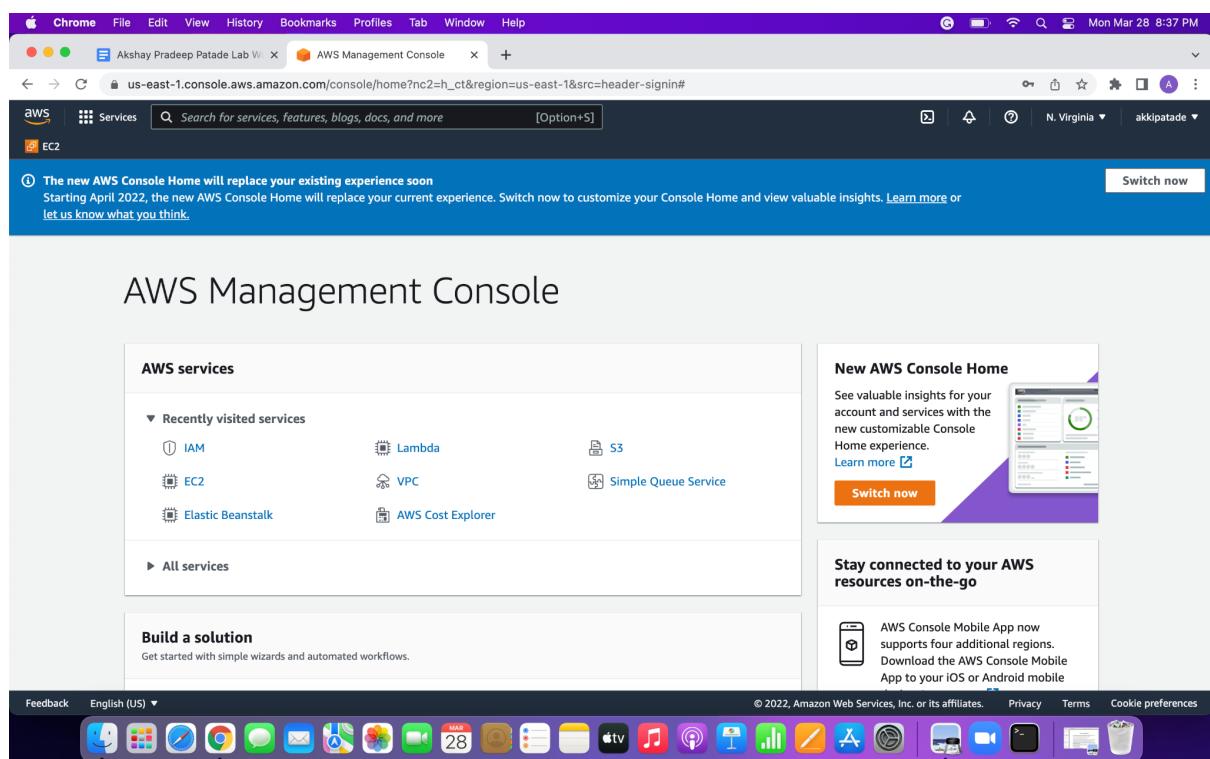




3. Open the terminal and check whether AWS is successfully installed or not.



#### 4. Login into AWS using your credentials and create an IAM role for your user.



The screenshot shows the AWS IAM Dashboard. On the left, there's a sidebar with navigation links like 'Identity and Access Management (IAM)', 'Access management', 'Access reports', and 'Feedback'. The main area has a blue header bar with the message: 'Introducing the new IAM dashboard experience. We've redesigned the IAM dashboard experience to make it easier to use. Let us know what you think.' Below this, the 'AWS Account' section displays the Account ID (484442202277), Account Alias (484442202277), and Sign-in URL (https://484442202277.signin.amazonaws.com/console). The 'Quick Links' section includes a link to 'My security credentials'. The 'Tools' section at the bottom right includes links for Privacy, Terms, and Cookie preferences.

## 5. Give a name to your user and select the Programmatic access option.

The screenshot shows the 'Add user' wizard. Step 1: Set user details. It shows a user name 'awsconsoleuser' and a link to 'Add another user'. Step 2: Select AWS access type. It shows two options: 'Access key - Programmatic access' (selected) and 'Password - AWS Management Console access'. A note says: 'Select how these users will primarily access AWS. If you choose only programmatic access, it does NOT prevent users from accessing the console using an assumed role. Access keys and autogenerated passwords are provided in the last step.' Step 3: A note at the bottom says: '\* Required'. Step 4: A 'Next: Permissions' button is visible at the bottom right. The footer includes links for Privacy, Terms, and Cookie preferences.

## 6. Click on permission and give the admin role to the user.

Add user

**Set permissions**

- Add user to group** (Selected)
- Copy permissions from existing user
- Attach existing policies directly

Add user to group

Create group Refresh

Group	Attached policies
awsconsoleuser	AdministratorAccess

Showing 1 result

**Set permissions boundary**

Cancel Previous Next: Tags

7. Click on create user button. Take the copy of the access id and key in the notepad.

Add user

**Success**

You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

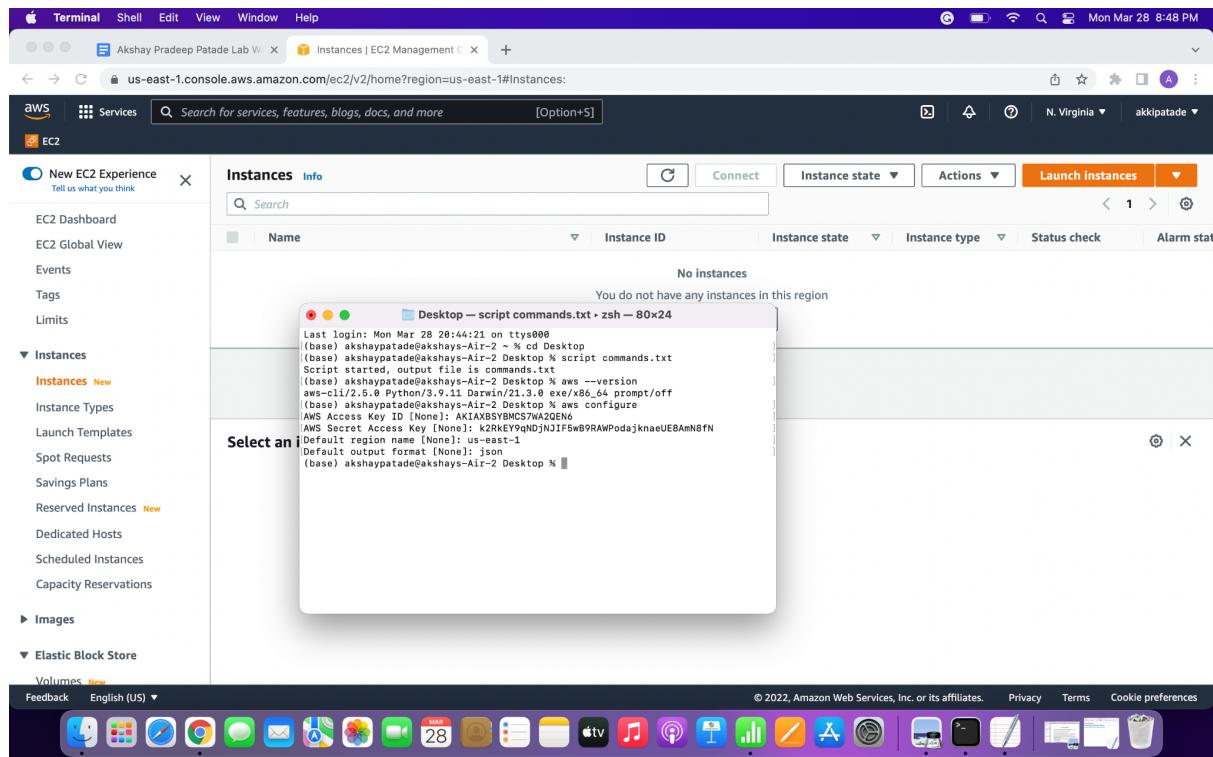
Users with AWS Management Console access can sign-in at: <https://484442202277.signin.aws.amazon.com/console>

Download .csv

User	Access key ID	Secret access key
awsconsoleuser	AKIAKBSYBMC57WA2QEN6	***** Show

Close

8. Configure the AWS CLI. Open the terminal and run the AWS configure command. This command is used to configure the AWS details in your terminal.



## Creating EC2 Instances using CLI

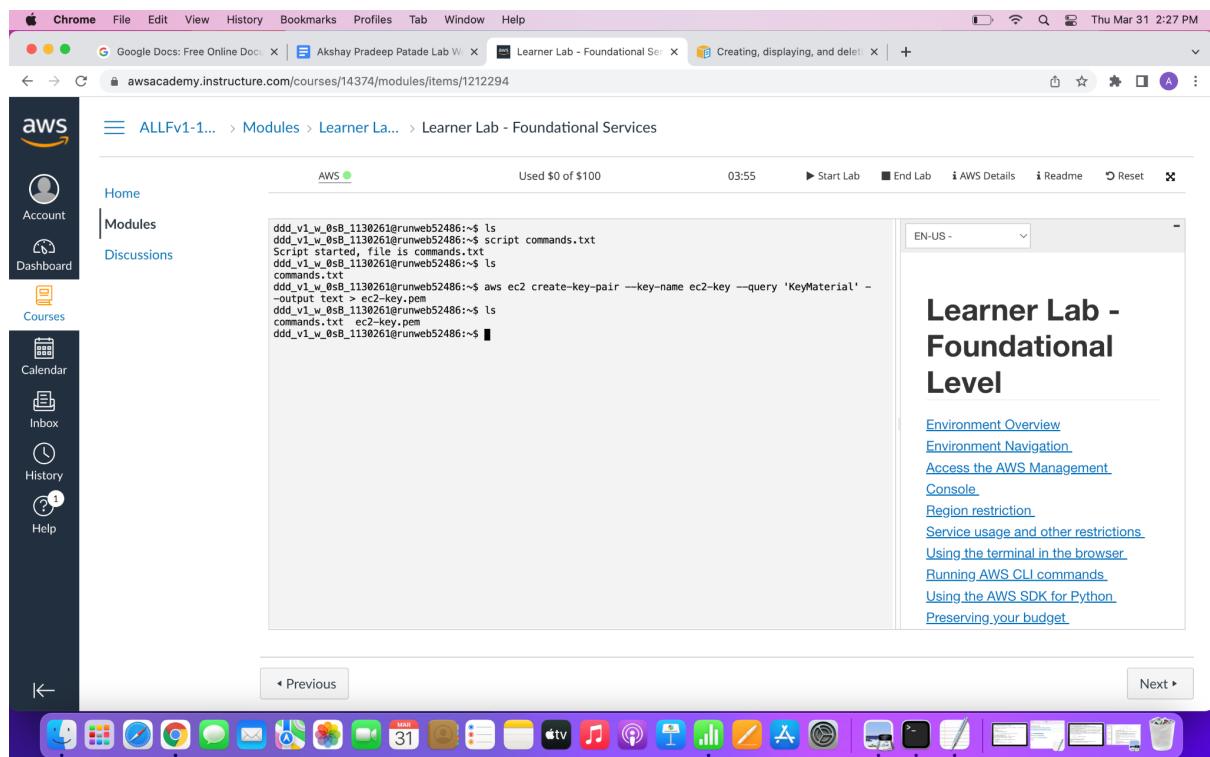
In order to create an instance using CLI, you need to have four things: Key pair, AMI (Amazon Machine Image) id, subnet id, and security group id.

You could see that currently we currently don't have any keypairs. So we will create one key pair and use that key pair for multiple instances.

### 1. Creating key Pair

In order to create a ec2 key pair we will use the below command.

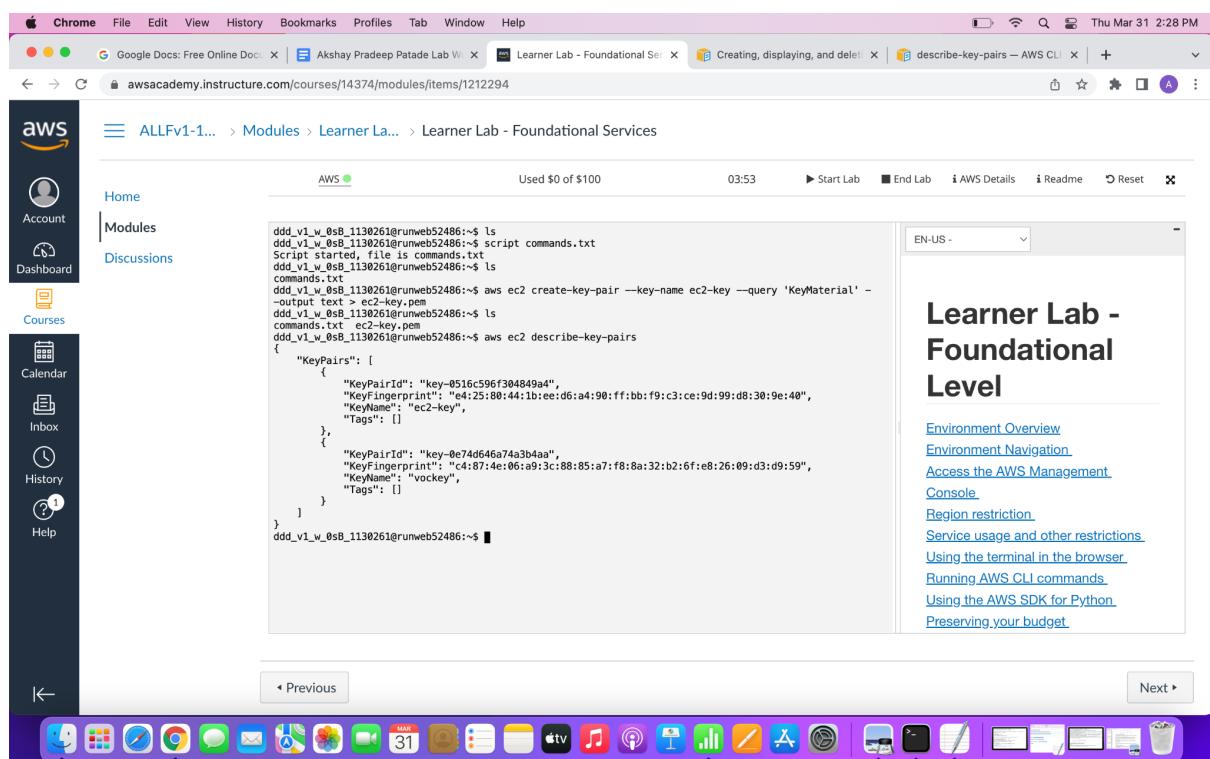
```
aws ec2 create-key-pair --key-name ec2-key --query 'KeyMaterial' --output text > ec2-key.pem
```



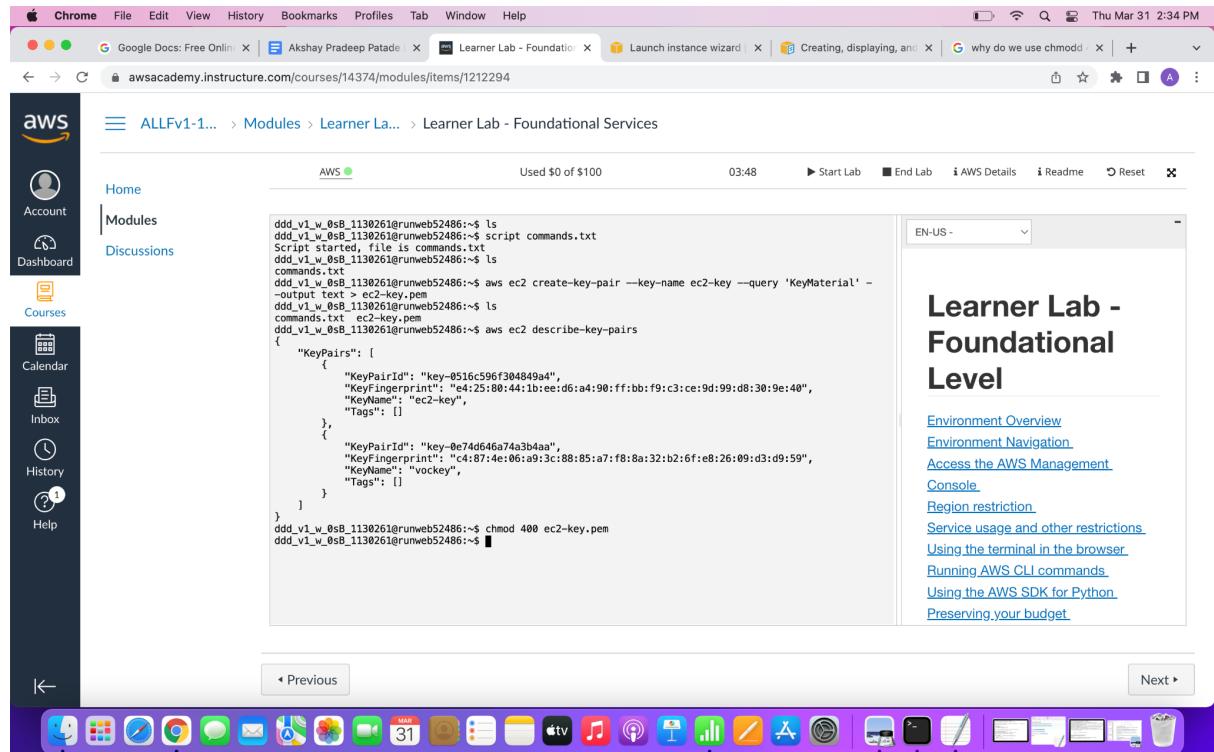
Check whether the key is successfully created or not using the below command.

`aws ec2 describe-key-pairs`

In the below screenshot we can see that the key is successfully created. We will use the same key to create the other four instances.



After creating a successful key pair we have to change the permission of the private key which is present with us to read-only. For that, we will use chmod 400 ec2-key.pem command.



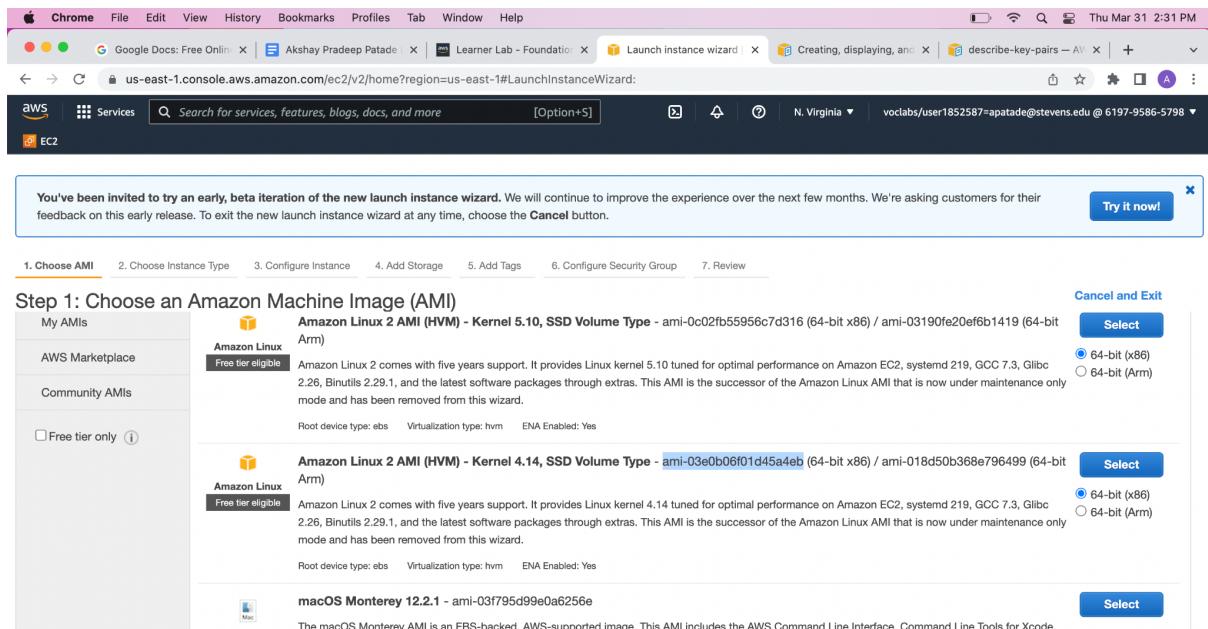
The screenshot shows a Mac desktop with a browser window open to [awsacademy.instructure.com/courses/14374/modules/items/1212294](https://awsacademy.instructure.com/courses/14374/modules/items/1212294). The browser has several tabs open, including Google Docs, Akshay Pradeep Patade, Learner Lab - Foundation, Launch instance wizard, Creating, displaying, and, why do we use chmod, and a search result for 'why do we use chmod'. The main content area displays a terminal session:

```
ddd_v1_w_0sB_1130261@runweb52486:~$ ls
ddd_v1_w_0sB_1130261@runweb52486:~$ script commands.txt
Script started. file is commands.txt
ddd_v1_w_0sB_1130261@runweb52486:~$ ls
commands.txt
ddd_v1_w_0sB_1130261@runweb52486:~$ aws ec2 create-key-pair --key-name ec2-key --query 'KeyMaterial' --output text > ec2-key.pem
ddd_v1_w_0sB_1130261@runweb52486:~$ ls
commands.txt ec2-key.pem
ddd_v1_w_0sB_1130261@runweb52486:~$ aws ec2 describe-key-pairs
{
  "KeyPairs": [
    {
      "KeyId": "key-8516c596f304849a4",
      "KeyFingerprint": "e4:25:80:44:1b:ee:d6:a4:90:ff:bb:f9:c3:ce:9d:99:d8:30:9e:40",
      "KeyName": "ec2-key",
      "Tags": []
    },
    {
      "KeyId": "key-8e74d646a74a3b4a9",
      "KeyFingerprint": "c4:87:4e:06:a9:3c:88:85:a7:f8:a8:32:b2:6f:e8:26:09:d3:d9:59",
      "KeyName": "vockey",
      "Tags": []
    }
  ]
}
ddd_v1_w_0sB_1130261@runweb52486:~$ chmod 400 ec2-key.pem
ddd_v1_w_0sB_1130261@runweb52486:~$
```

To the right of the terminal, there is a sidebar titled "Learner Lab - Foundational Level" with links to various AWS Academy resources.

## 2. Amazon Machine Image ID.

For the AMI Id we can directly go to AWS Management Console and copy the Machine ID. Copy the highlighted portion present in the screenshot.



### 3. Creating a Security group

To create a new security group, we need to have a VPC id. We can use the default security group which is present in AWS, but for safety purposes, we will create a new security group. For creating a new security group, we must have a VPC id. We will use AWS EC2. In the below screenshot you can see that there is already a VPC present. We will use the VPC id to create a new security group.

We will use the below command to create a security group.

```
aws ec2 create-security-group --group-name Ec2SecurityGroup --description "EC2 Security Group" --vpc-id vpc-0f19a97c7eaccce15
```

The screenshot shows the AWS Lambda console interface. On the left, there's a sidebar with icons for Account, Dashboard, Courses, Calendar, Inbox, History, and Help. The main area has tabs for Home, Modules (which is selected), and Discussions. A terminal window displays the following command-line session:

```

aws ec2 describe-vpcs
{
    "Vpcs": [
        {
            "CidrBlock": "172.31.0.0/16",
            "DhcpOptionsId": "dopt-82bca4596b7592e2c",
            "State": "available",
            "VpcId": "vpc-0f19a97c7eaccc15",
            "OwnerId": "613795865798",
            "InstanceTenancy": "default",
            "CidrBlockAssociationSet": [
                {
                    "AssociationId": "vpc-cidr-assoc-09f245cabfd738b8d",
                    "CidrBlock": "172.31.0.0/16",
                    "CidrBlockState": {
                        "State": "associated"
                    }
                }
            ],
            "IsDefault": true
        }
    ]
}
aws ec2 create-security-group --group-name Ec2SecurityGroup --description "EC2 Security Group" --vpc-id vpc-0f19a97c7eaccc15
{
    "GroupId": "sg-0264063663678a9ab"
}

```

To the right of the terminal, a sidebar titled "Learner Lab - Foundational Level" contains links to various AWS documentation and resources.

This screenshot is similar to the one above, showing the AWS Lambda console with the "Modules" tab selected. The terminal window now displays the output of the "aws ec2 describe-security-groups" command:

```

{
    "SecurityGroups": [
        {
            "Description": "EC2 Security Group",
            "GroupName": "Ec2SecurityGroup",
            "IpPermissions": [],
            "OwnerId": "613795865798",
            "GroupId": "sg-0264063663678a9ab",
            "IpPermissionsEgress": [
                {
                    "IpProtocol": "-1",
                    "IpRanges": [
                        {
                            "CidrIp": "0.0.0.0/0"
                        }
                    ],
                    "Ipv6Ranges": [],
                    "PrefixListIds": [],
                    "UserIdGroupPairs": []
                }
            ],
            "VpcId": "vpc-0f19a97c7eaccc15"
        },
        {
            "Description": "default VPC security group",
            "GroupName": "default",
            "IpPermissions": [
                {
                    "IpProtocol": "-1",
                    "IpRanges": [],
                    "Ipv6Ranges": [],
                    "PrefixListIds": []
                }
            ],
            "VpcId": "vpc-0f19a97c7eaccc15"
        }
    ]
}

```

The right sidebar "Learner Lab - Foundational Level" remains the same, providing links to AWS documentation.

After creating a security group we have to set some permissions so that only I can access the ec2 instances. We must know our public ip address in order to do the configurations.

We will use the below command.

```
dig +short myip.opendns.com @resolver1.opendns.com
```

The screenshot shows a web browser window with multiple tabs open. The active tab is 'awsacademy.instructure.com/courses/14374/modules/items/1212294'. The page displays the 'Learner Lab - Foundational Services' section of the AWS Academy. On the left, there's a sidebar with navigation links like Home, Modules, Discussions, Account, Dashboard, Courses, Calendar, Inbox, History, and Help. The main content area has a terminal window showing the command: `dig +short myip.opendns.com @resolver1.opendns.com`, which outputs the IP address `34.221.65.122`. To the right of the terminal is a large box titled 'Learner Lab - Foundational Level' containing various links related to AWS services and best practices. At the bottom of the page are 'Previous' and 'Next' navigation buttons.

Note down your public ip address somewhere.

We will use the below commands to add rules to our security group.

```
aws ec2 authorize-security-group-ingress --group-id sg-0264063663678a9ab --protocol tcp  
--port 22 --cidr 34.213.39.113/32
```

```
aws ec2 authorize-security-group-ingress --group-id sg-0264063663678a9ab --protocol tcp  
--port 80 --cidr 34.213.39.113/32
```

#### 4. Getting the subnet id.

Use the below command to get all the subnets. Among all the subnets, select any of the subnets which you desire.

```
aws ec2 describe-subnets.
```

The screenshot shows a browser window with multiple tabs open, including Google Docs, VPC Management, Mail, and Launching, listing. The main content area displays the AWS Learner Lab interface. On the left, there's a sidebar with navigation links like Home, Modules, Discussions, Account, Dashboard, Courses, Calendar, Inbox, History, and Help. The main panel shows a terminal window with the command `aws ec2 describe-subnets`. The output of this command is a JSON object containing details about two subnets in the us-east-1c availability zone. The subnets have CIDR blocks of 172.31.0.0/20 and 172.31.48.0/20. They both have 4091 available IPv4 addresses. The subnets are in an 'available' state and have their own public IP ranges. The subnet ARNs are `arn:aws:ec2:us-east-1:619795865798:subnet/subnet-02d6e208cf471bfd2` and `arn:aws:ec2:us-east-1:619795865798:subnet/subnet-0b7f166dbb15b1487`. To the right of the terminal, there's a sidebar titled "Learner Lab - Foundational Level" with links to various AWS resources and documentation.

```

ddd_v1_w_0sB_1130261@runweb52517:~$ aws ec2 describe-subnets
{
    "Subnets": [
        {
            "AvailabilityZone": "us-east-1c",
            "AvailabilityZoneId": "use1-az1",
            "AvailableIpAddressCount": 4091,
            "CidrBlock": "172.31.0.0/20",
            "DefaultForAz": true,
            "MapPublicIpOnLaunch": true,
            "MapCustomerOwnedIpOnLaunch": false,
            "State": "available",
            "SubnetId": "subnet-02d6e208cf471bfd2",
            "VpcId": "vpc-0f19a97c7eaccc15",
            "OwnerId": "619795865798",
            "AssignIpv6AddressOnCreation": false,
            "Ipv6CidrBlockAssociationSet": [],
            "SubnetArn": "arn:aws:ec2:us-east-1:619795865798:subnet/subnet-02d6e208cf471bfd2"
        },
        {
            "AvailabilityZone": "us-east-1c",
            "AvailabilityZoneId": "use1-az2",
            "AvailableIpAddressCount": 4091,
            "CidrBlock": "172.31.48.0/20",
            "DefaultForAz": true,
            "MapPublicIpOnLaunch": true,
            "MapCustomerOwnedIpOnLaunch": false,
            "State": "available",
            "SubnetId": "subnet-0b7f166dbb15b1487",
            "VpcId": "vpc-0f19a97c7eaccc15",
            "OwnerId": "619795865798",
            "AssignIpv6AddressOnCreation": false,
            "Ipv6CidrBlockAssociationSet": [],
            "SubnetArn": "arn:aws:ec2:us-east-1:619795865798:subnet/subnet-0b7f166dbb15b1487"
        }
    ]
}

```

Now we are all set and good to go. We will use the below command to create a new instance.

```
aws ec2 run-instances --image-id ami-0c02fb55956c7d316 --count 5 --instance-type t2.micro
--key-name ec2-key --security-group-ids sg-0264063663678a9ab --subnet-id
subnet-02d6e208cf471bfd2
```

Since we have to create five instances we made use of the count subcommand so that the instance of this configuration is created five times.

We will give the name of those servers.

The screenshot shows the AWS Management Console interface for the EC2 Management Console. The left sidebar has a tree view with 'Instances' selected, showing sub-options like 'Instances' (New), 'Instance Types', 'Launch Templates', etc. The main content area displays a table titled 'Instances (6) Info' with columns: Name, Instance ID, Instance state, Instance type, Status check, and Alarm stat. The table lists six instances: Server1, Server2, Server3, Server4, and a Load Balancer, all in the 'Running' state. Below the table is a modal window titled 'Select an instance'.

The screenshot shows the AWS Academy Learner Lab - Foundational Services page. The left sidebar includes 'Account', 'Dashboard', 'Calendar', 'Inbox', 'History', and 'Help'. The main content area shows a terminal session output and a summary card for the 'Learner Lab - Foundational Level'.

```

ddd_v1_w_0sB_1130261@runweb52542:~$ ls
commands.txt ec2-key.pem
ddd_v1_w_0sB_1130261@runweb52542:~$ aws ec2 run-instances --image-id ami-0c02fb55956c7d316
{
  "Groups": [],
  "Instances": [
    {
      "AmiLaunchIndex": 0,
      "ImageId": "ami-0c02fb55956c7d316",
      "InstanceId": "i-073529ec14febd0e99",
      "InstanceType": "t1.micro",
      "LaunchTime": "2022-03-31T19:37:57+00:00",
      "Monitoring": {
        "State": "disabled"
      },
      "Placement": {
        "AvailabilityZone": "us-east-1d",
        "GroupName": "",
        "Tenancy": "default"
      },
      "PrivateIpAddress": "ip-172-31-81-10.ec2.internal",
      "PrivateIpAddresses": "172.31.81.10",
      "ProductCodes": [],
      "PublicDnsName": "",
      "State": {
        "Code": 0,
        "Name": "pending"
      },
      "StateTransitionReason": "",
      "SubnetId": "subnet-0508013b8ae61df7d",
      "VpcId": "vpc-0f19a97c7eacccce15",
      "Architecture": "x86_64",
      "BlockDeviceMappings": [],
      "ClientToken": "6584da14-f797-497f-ab36-20b3987fbade"
    }
  ]
}
  
```

**Learner Lab - Foundational Level**

- Environment Overview
- Environment Navigation
- Access the AWS Management Console
- Region restriction
- Service usage and other restrictions
- Using the terminal in the browser
- Running AWS CLI commands
- Using the AWS SDK for Python
- Preserving your budget

We can go to AWS Management Console and check whether the instance is successfully created or not. We can see that it is created successfully.

The screenshot shows the AWS EC2 Management Console interface. On the left, there's a sidebar with navigation links like 'EC2 Dashboard', 'EC2 Global View', 'Events', 'Tags', 'Limits', 'Instances' (selected), 'Images', and 'AMIs'. The main area is titled 'Instances (6) Info' and displays a table of six running instances. The table columns include Name, Instance ID, Instance state, Instance type, Status check, and Alarm stat. The instances listed are Server1, Server2, Server3, Server4, and a Load Balancer. Below the table, a modal window titled 'Select an instance' is open. At the bottom of the screen, there's a Mac OS X dock with various application icons.

We will try to login into the server and check whether we are able to successfully connect or not.

ssh -i "ec2-key.pem" [ec2-user@ec2-100-27-39-254.compute-1.amazonaws.com](mailto:ec2-user@ec2-100-27-39-254.compute-1.amazonaws.com)

The screenshot shows the AWS Academy Learner Lab interface. On the left, there's a sidebar with 'Account' (selected), 'Dashboard', 'Courses', 'Calendar', 'Inbox', 'History', and 'Help'. The main area shows a terminal session titled 'Connect to instance | EC2 Main'. The terminal output shows a successful SSH connection to the EC2 instance. The right side of the screen displays the 'Learner Lab - Foundational Level' content, which includes a table of contents for 'Environment Overview', 'Environment Navigation', 'Access the AWS Management Console', 'Region restriction', 'Service usage and other restrictions', 'Using the terminal in the browser', 'Running AWS CLI commands', 'Using the AWS SDK for Python', and 'Preserving your budget'. At the bottom, there are 'Previous' and 'Next' navigation buttons.

As you can see in the above screenshot we are able to connect to the server.

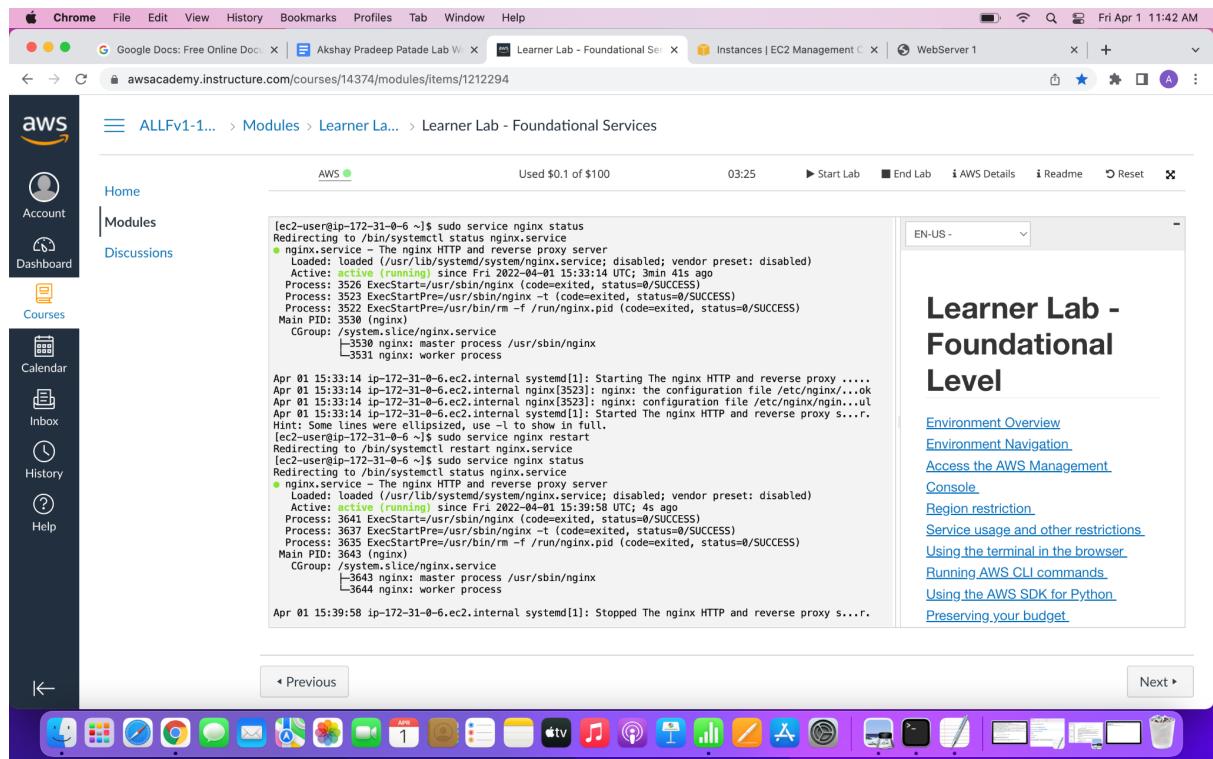
Now we will make use of amazon-linux-extras repository and install nginx on each server

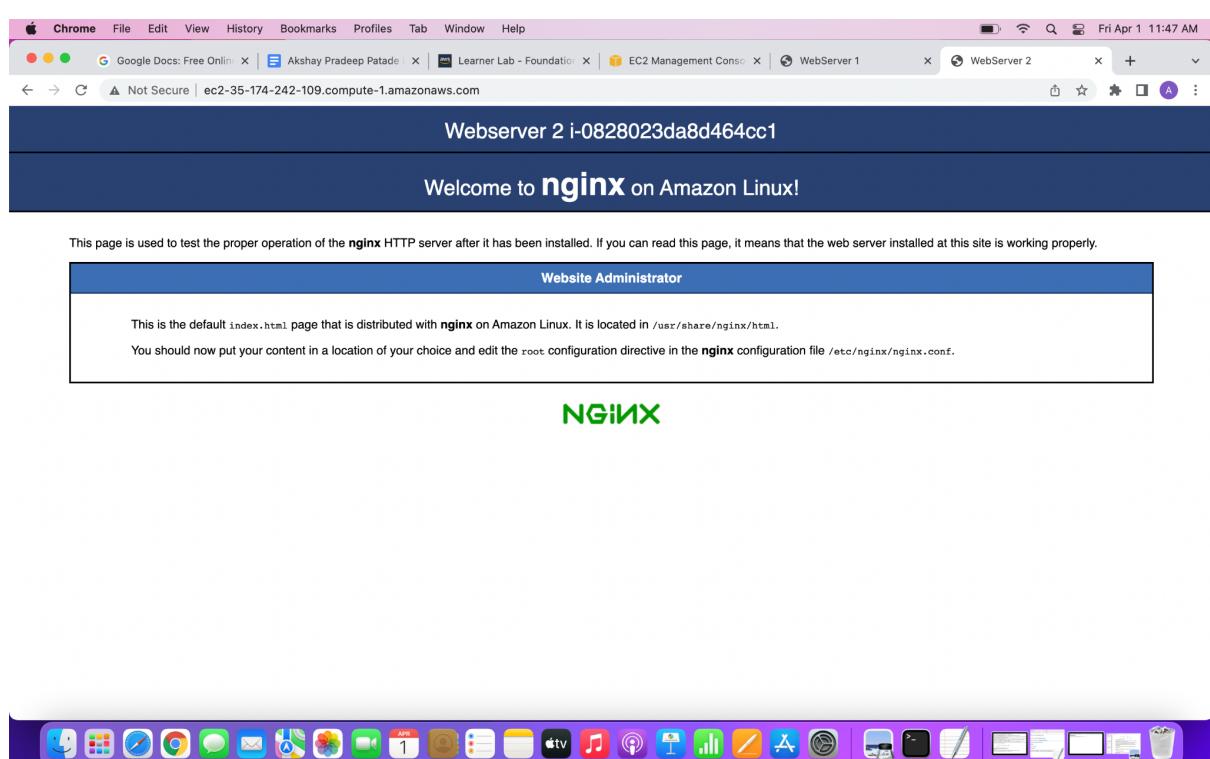
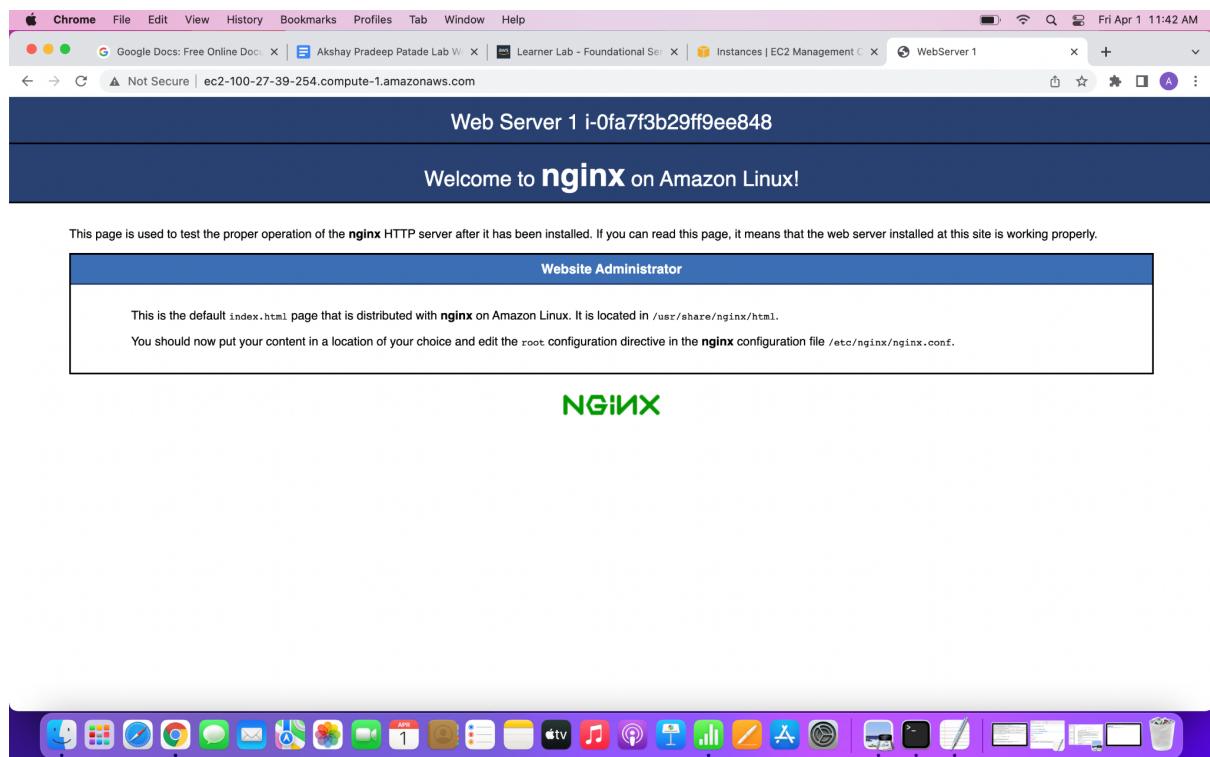
We will use the below-mentioned command.

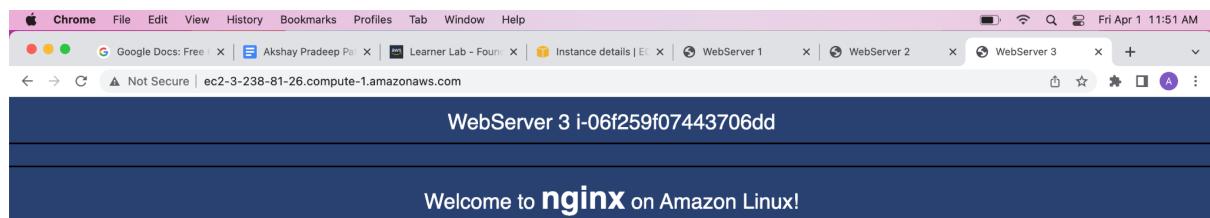
```
sudo amazon-linux-extras install nginx1.
```

Once it is done, Do some modifications to the index.html web page so that the web pages are distinguishable for each of the servers.

Once it is done, start the service and run it on the webserver. Do this activity for the rest of the servers.







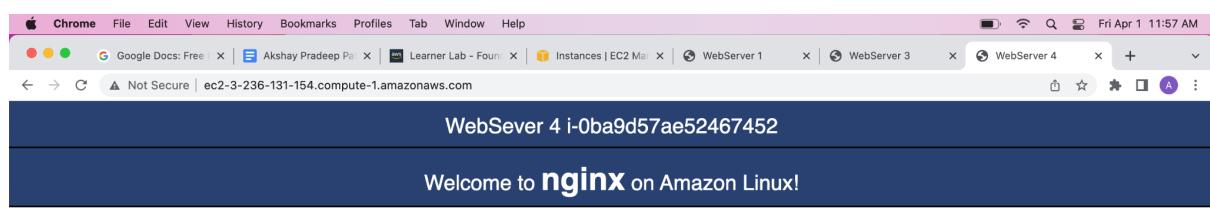
This page is used to test the proper operation of the **nginx** HTTP server after it has been installed. If you can read this page, it means that the web server installed at this site is working properly.

#### Website Administrator

This is the default `index.html` page that is distributed with **nginx** on Amazon Linux. It is located in `/usr/share/nginx/html`.

You should now put your content in a location of your choice and edit the `root` configuration directive in the **nginx** configuration file `/etc/nginx/nginx.conf`.

NGINX



This page is used to test the proper operation of the **nginx** HTTP server after it has been installed. If you can read this page, it means that the web server installed at this site is working properly.

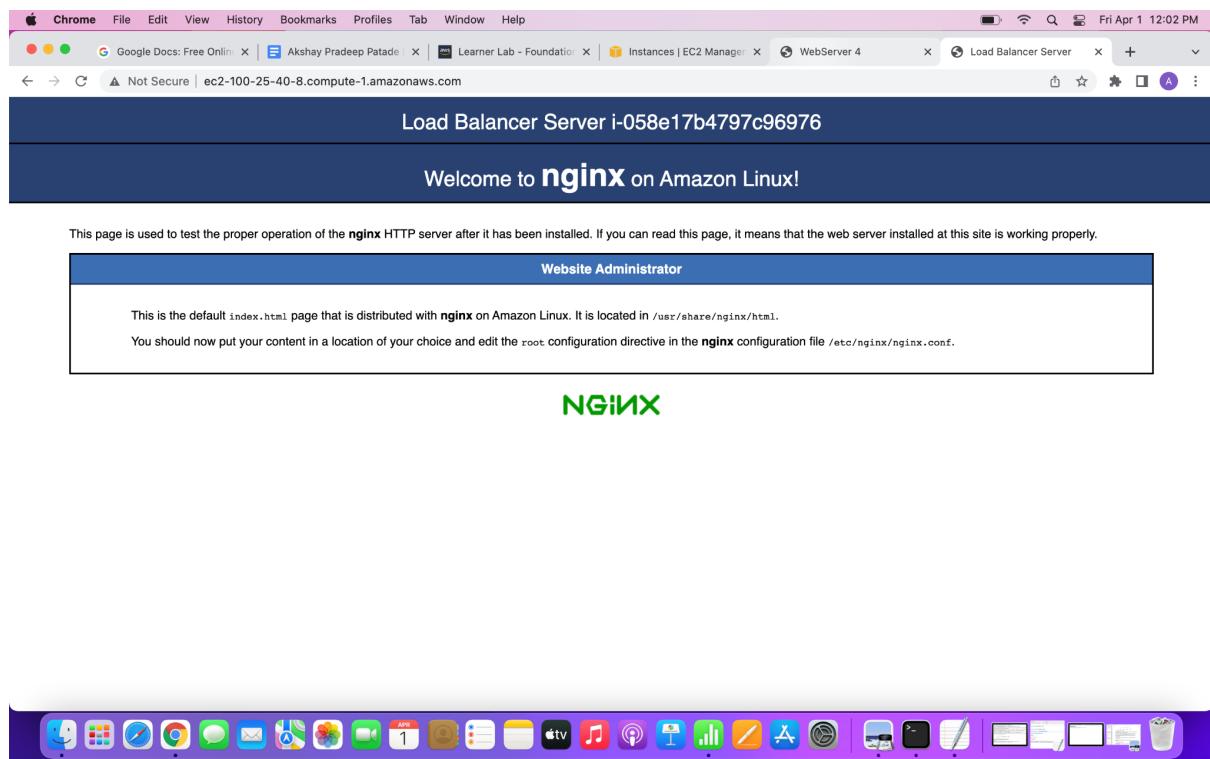
#### Website Administrator

This is the default `index.html` page that is distributed with **nginx** on Amazon Linux. It is located in `/usr/share/nginx/html`.

You should now put your content in a location of your choice and edit the `root` configuration directive in the **nginx** configuration file `/etc/nginx/nginx.conf`.

NGINX





The next step is to make the changes in the configuration file of the load balancer.

We will first log in to the load balancer server using the following command.

```
ssh -i "ec2-key.pem" ec2-user@ec2-100-25-40-8.compute-1.amazonaws.com
```

Then we will go to /etc/nginx/nginx.conf to change the configuration.

The screenshot shows a Mac desktop with a Chrome browser window open to the AWS Academy learner lab for Foundational Services. The browser has multiple tabs and icons in the toolbar. The main content area displays the AWS Lambda interface. On the left is a sidebar with options like Home, Modules, Discussions, Account, Courses, Calendar, Inbox, History, and Help. The central area shows a terminal window with the following Nginx configuration:

```
user nginx;
worker_processes auto;
error_log /var/log/nginx/error.log;
pid /run/nginx.pid;

# Load dynamic modules. See /usr/share/doc/nginx/README.dynamic.
include /usr/share/nginx/modules/*.conf;
events {
    worker_connections 768;
}
http {
    upstream myapp {
        ip_hash;
        server ec2-34-205-93-232.compute-1.amazonaws.com weight=1;
        server ec2-3-239-244-77.compute-1.amazonaws.com weight=1;
        server ec2-44-208-208-37.compute-1.amazonaws.com weight=1;
        server ec2-3-237-223-181.compute-1.amazonaws.com weight=1;
    }
    server {
        listen 80;
        server_name myapp.com;
        location / {
            proxy_pass http://myapp;
        }
    }
}
"/etc/nginx/nginx.conf" 31L, 637B
```

The right side of the interface shows a summary: Used \$0.1 of \$100, 01:41, Start Lab, End Lab, AWS Details, Readme, Reset, and a dropdown for EN-US. Below this is a section titled "Learner Lab - Foundational Level" with a list of links: Environment Overview, Environment Navigation, Access the AWS Management Console, Region restriction, Service usage and other restrictions, Using the terminal in the browser, Running AWS CLI commands, Using the AWS SDK for Python, and Preserving your budget.

Now we will use the curl command to visit the balancer, which will distribute traffic among the servers.

The screenshot shows a Mac desktop with a Chrome browser window open to the AWS Academy learner lab for Foundational Services. The browser has multiple tabs and icons in the toolbar. The main content area displays the AWS Lambda interface. On the left is a sidebar with options like Home, Modules, Discussions, Account, Courses, Calendar, Inbox, History, and Help. The central area shows a terminal window with the following HTML content:

```
Amazon Linux 2 AMI
https://aws.amazon.com/amazon-linux-2/
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/1999/xhtml1/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head>
    <title>WebServer 3</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <style type="text/css">
        /*!<!--[if !IE]>|
        body {
            background-color: #fff;
            color: #000;
            font-size: 0.9em;
            font-family: sans-serif, helvetica;
            margin: 0;
            padding: 0;
        }
        :link {
            color: #c00;
        }
        :visited {
            color: #c00;
        }
        :hover {
            color: #f50;
        }
        h1 {
            text-align: center;
            margin: 0;
            padding: 0.6em 2em 0.4em;
        }
        /*-->|<!--[if !IE]-->

The right side of the interface shows a summary: Used $0.1 of $100, 01:22, Start Lab, End Lab, AWS Details, Readme, Reset, and a dropdown for EN-US. Below this is a section titled "Learner Lab - Foundational Level" with a list of links: Environment Overview, Environment Navigation, Access the AWS Management Console, Region restriction, Service usage and other restrictions, Using the terminal in the browser, Running AWS CLI commands, Using the AWS SDK for Python, and Preserving your budget.


```

This screenshot shows a web browser window for the AWS Academy Learner Lab. The URL is [awsacademy.instructure.com/courses/14374/modules/items/1212294](https://awsacademy.instructure.com/courses/14374/modules/items/1212294). The page title is "ALLFv1-1... > Modules > Learner La... > Learner Lab - Foundational Services". On the left, there's a sidebar with navigation links: Home, Modules (selected), Discussions, Account, Dashboard, Courses, Calendar, Inbox, History, and Help. The main content area shows a terminal session output:

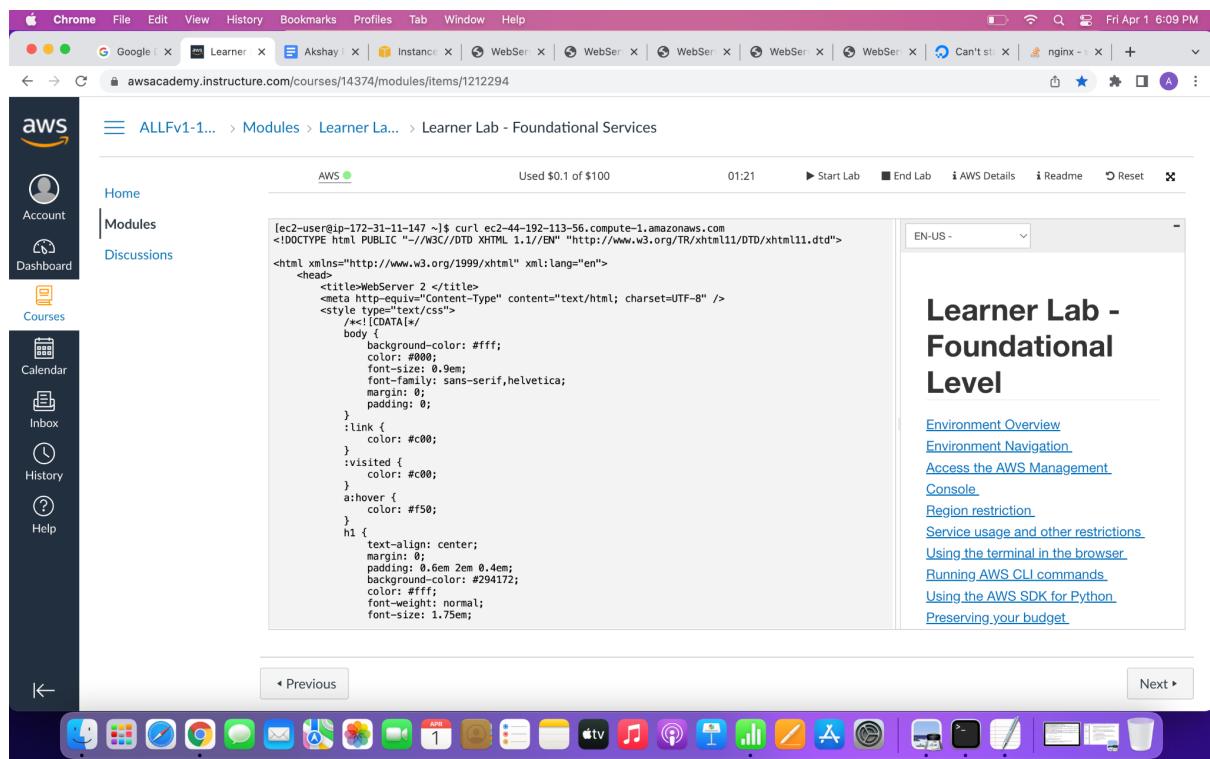
```
[ec2-user@ip-172-31-11-147 ~]$ curl ec2-44-192-113-56.compute-1.amazonaws.com
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>WebServer 4</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <style type="text/css">
      /*!CDATA[*/
      body {
        background-color: #ffff;
        color: #0000;
        font-size: 0.9em;
        font-family: sans-serif,helvetica;
        margin: 0;
        padding: 0;
      }
      :link {
        color: #c000;
      }
      :visited {
        color: #c000;
      }
      :hover {
        color: #ff00;
      }
      h1 {
        text-align: center;
        margin: 0;
        padding: 0.6em 2em 0.4em;
        background-color: #294172;
        color: #ffff;
        font-weight: normal;
        font-size: 1.75em;
      }
    
  
  <body>
```

The right side of the screen features a large title "Learner Lab - Foundational Level" and a sidebar with various links:

- Environment Overview
- Environment Navigation
- Access the AWS Management Console
- Region restriction
- Service usage and other restrictions
- Using the terminal in the browser
- Running AWS CLI commands
- Using the AWS SDK for Python
- Preserving your budget

At the bottom, there are "Previous" and "Next" navigation buttons.

This screenshot is identical to the one above, showing the same AWS Academy Learner Lab interface, terminal output, title, sidebar links, and navigation buttons. The only difference is the date in the top right corner, which has changed from "Fri Apr 1 6:08 PM" to "Fri Apr 1 6:09 PM".



Now we will write a script in ruby which will count the number of visits to the websites.

We will again use amazon-linux-extras to install ruby.

We will use the below command to install ruby

```
sudo amazon-linux-extras install ruby
```

awsacademy.instructure.com/courses/14374/modules/items/1212294

```
#!/usr/bin/env ruby
# This program is used for collecting web server visit information.
# Author: A. Genius
require 'optparse'
def print_usage
  puts "USAGE: visit_server -d DNS_NAME"
  exit
end
# add option switch and handler
options = {}
option_parser = OptionParser.new do |opts|
  # DNS_NAME argument
  options[:dns_name] = nil
  opts.on("-d", "--dns-name DNS_NAME", 'Specify a DNS NAME') { |dns_name|
    options[:dns_name] = dns_name
  }
  options[:help] = nil
  opts.on('-h', '--help', 'Display usage') { |help| options[:help] = help }
end
option_parser.parse!
# verify arguments
if options[:dns_name]
  dns_name = options[:dns_name]
else
  puts "Please set a balancer's DNS."
  exit
end
if options[:help]
  print_usage
  exit
end
# Keep STDOUT
#orig_stdout = $stdout
# redirect stdout to /dev/null
$stdout = File.new('/dev/null', 'w')
server1_visit_count = 0
server2_visit_count = 0
```

awsacademy.instructure.com/courses/14374/modules/items/1212294

```
server2_visit_count = 0
server3_visit_count = 0
server4_visit_count = 0
# starting to visit load balancing server
puts "Starting to visit load balancing server" 2000.times do
  # visit load balancer
  o = `curl #{$dns_name}`
  if o =~ /server1/
    server1_visit_count += 1
  elsif o =~ /server2/
    server2_visit_count += 1
  elsif o =~ /server3/
    server3_visit_count += 1
  else
    server4_visit_count += 1
  end
  print "."
end
# redirect output to stdout
#$stdout = orig_stdout
# print visit information
puts "-----"
puts 'Summary'
puts "-----"
puts "Server1 visit counts : " + server1_visit_count.to_s puts "Server2 visit counts : " + server2_visit_count.to_s
puts "Server3 visit counts : " + server3_visit_count.to_s
puts "Server4 visit counts : " + server4_visit_count.to_s
puts "Total visit counts : " + (server1_visit_count + server2_visit_count + server3_visit_count + server4_visit_count).to_s
```

Once the script is successfully written we will run that script using the below command.

```
sudo ruby visit_server.rb -d ec2-44-192-113-56.compute-1.amazonaws.com
```

Scenario 1 : When the weight for all the servers is set to 1

The screenshot shows a browser window for the AWS Academy Learner Lab. The URL is [awsacademy.instructure.com/courses/14374/modules/items/1212294](https://awsacademy.instructure.com/courses/14374/modules/items/1212294). The page title is "ALLFv1-1... > Modules > Learner La... > Learner Lab - Foundational Services". On the left, there's a sidebar with links for Account, Dashboard, Courses, Calendar, Inbox, History, and Help. The main content area shows a terminal session with the following text:

```
visit_server.rb:5: syntax error, unexpected `def', expecting end-of-input
require 'optparse'
  def print_usage
[...]
```

Below the terminal, a "Summary" section shows visit counts for four servers:

Server	Visit Counts
Server1	500
Server2	500
Server3	500
Server4	500
Total	2000

The right panel contains a sidebar titled "Learner Lab - Foundational Level" with links to various AWS resources.

Scenario 2 : When weight is distributed as Server 1 = 1, Server 2 = 2, Server 3 = 3, Server 4 = 4

The screenshot shows a browser window for the AWS Academy Learner Lab, identical to the previous one but with a different visit distribution. The terminal session now shows:

```
[ec2-user@ip-172-31-11-147 ~]$ sudo ruby visit_server.rb -d ec2-44-192-113-56.compute-1.amazonaws.com
Starting to visit load balancing server
[...]
```

The "Summary" section shows visit counts for four servers:

Server	Visit Counts
Server1	200
Server2	400
Server3	600
Server4	800
Total	2000

The right panel contains a sidebar titled "Learner Lab - Foundational Level" with links to various AWS resources.

Scenario 3 : When weight is distributed as Server 1 = 1, Server 2 = 2, Server 3 = 1, Server 4 = 2

The screenshot shows a Chrome browser window with the AWS Academy interface. The URL is [awsacademy.instructure.com/courses/14374/modules/items/1212294](https://awsacademy.instructure.com/courses/14374/modules/items/1212294). The page title is "ALLFv1-1... > Modules > Learner La... > Learner Lab - Foundational Services". On the left, there's a sidebar with links for Account, Dashboard, Courses, Calendar, Inbox, History, and Help. The main content area has tabs for Home, Modules (which is selected), and Discussions. A terminal window shows the following command and its output:

```
[ec2-user@ip-172-31-11-147 ~]$ sudo service nginx restart
Redirecting to /bin/systemctl restart nginx.service
[ec2-user@ip-172-31-11-147 ~]$ sudo ruby visit_server.rb -d ec2-44-192-113-56.compute-1.amazonaws.com
Starting to visit load balancer server...
```

Below the terminal, a "Summary" section shows visit counts for four servers:

Server	Visit Counts
Server1	333
Server2	667
Server3	333
Server4	667
Total	2000

The right sidebar is titled "Learner Lab - Foundational Level" and contains a list of links:

- Environment Overview
- Environment Navigation
- Access the AWS Management Console
- Region restriction
- Service usage and other restrictions
- Using the terminal in the browser
- Running AWS CLI commands
- Using the AWS SDK for Python
- Preserving your budget

## Analyzing the packets using the tcpdump command in linux.

First check whether tcpdump is installed in your laptop or not. It seems that it is installed. We checked it by using which tcpdump command.

The screenshot shows a Chrome browser window with the AWS Academy interface. The URL is [awsacademy.instructure.com/courses/14374/modules/items/1212294](https://awsacademy.instructure.com/courses/14374/modules/items/1212294). The page title is "ALLFv1-1... > Modules > Learner La... > Learner Lab - Foundational Services". On the left, there's a sidebar with links for Account, Dashboard, Courses, Calendar, Inbox, History, and Help. The main content area has tabs for Home, Modules (which is selected), and Discussions. A terminal window shows the following command and its output:

```
[ec2-user@ip-172-31-11-147 ~]$ which tcpdump
/usr/sbin/tcpdump
[ec2-user@ip-172-31-11-147 ~]$
```

The right sidebar is titled "Learner Lab - Foundational Level" and contains a list of links:

- Environment Overview
- Environment Navigation
- Access the AWS Management Console
- Region restriction
- Service usage and other restrictions
- Using the terminal in the browser
- Running AWS CLI commands
- Using the AWS SDK for Python
- Preserving your budget

Now we will dump all the packets using the below command

```
sudo tcpdump -i eth0 > data.txt
```

Below is the output of the tcpdump

```
2:48:41.948936 IP ip-172-31-11-147.ec2.internal.ssh > ec2-34-216-117-22.us-west-2.compute.amazonaws.com.32856: Flags [P..], seq 2156755228:1256755344, ack 1067016544, win 456, options [nop,nop,TS val 3062 290484 ecr 4164177], length 116
22:48:41.949036 IP ip-172-31-11-147.ec2.internal.ssh > ec2-34-216-117-22.us-west-2.compute.amazonaws.com.32856: Flags [P..], seq 1:1, win 456, options [nop,nop,Ts val 3062298950 ecr 4164177], length 116
22:48:41.949316 ARP, Request who-has ip-172-31-0-2.ec2.internal tell ip-172-31-11-147.ec2.internal, length 28
22:48:41.949461 ARP, Reply ip-172-31-0-2.ec2.internal is-at 02:66:c1:5d:9b:2d [oui Unknown], length 42
22:48:41.949467 IP ip-172-31-11-147.ec2.internal.58649 > ip-172-31-0-2.ec2.internal.domain: 1664+ PTR? 22.117.216.34.in-addr.arpa. (44)
22:48:41.949869 IP ip-172-31-0-2.ec2.internal.domain > ip-172-31-11-147.ec2.internal.58649: 1664 1/0/0 PTR ec2-34-216-117-22.us-west-2.compute.amazonaws.com. (107)
22:48:41.949935 IP ip-172-31-11-147.ec2.internal.56858 > ip-172-31-0-2.ec2.internal.domain: 57030+ PTR? 147.111.31.12.in-addr.arpa. (44)
22:48:41.950224 IP ip-172-31-0-2.ec2.internal.domain > ip-172-31-11-147.ec2.internal.56858: 57030 1/0/0 PTR ip-172-31-11-147.ec2.internal. (87)
22:48:41.950300 IP ip-172-31-11-147.ec2.internal.39713 > ip-172-31-0-2.ec2.internal.domain: 9100+ PTR? 2.0.31.172.in-addr.arpa. (41)
22:48:41.950699 IP ip-172-31-0-2.ec2.internal.domain > ip-172-31-11-147.ec2.internal.39713: 9100 1/0/0 PTR ip-172-31-0-2.ec2.internal.39713 > ip-172-31-11-147.ec2.internal.39713: 9100 1/0/0
22:48:41.951705 IP ip-172-31-0-2.ec2.internal.39713 > ip-172-31-11-147.ec2.internal.55382: 160 1/0/0 Flags [.], ack 0, win 854, options [nop,nop,Ts val 4164195 ecr 3062298926], length 0
22:48:42.018113 IP ec2-34-216-117-22.us-west-2.compute.amazonaws.com.32856 > ip-172-31-11-147.ec2.internal.ssh: Flags [.], ack 116, win 854, options [nop,nop,Ts val 4164201 ecr 3062298950], length 0
22:48:42.018182 IP ec2-34-216-117-22.us-west-2.compute.amazonaws.com.32856 > ip-172-31-11-147.ec2.internal.ssh: Flags [.], ack 123, win 854, options [nop,nop,Ts val 4164201 ecr 3062298950], length 0
22:48:42.018183 IP ec2-34-216-117-22.us-west-2.compute.amazonaws.com.32856 > ip-172-31-11-147.ec2.internal.ssh: Flags [.], ack 123, win 854, options [nop,nop,Ts val 4164201 ecr 3062298950], length 0
22:48:46.774279 IP ip-172-31-11-147.ec2.internal.52318 > ip-172-31-0-2.ec2.internal.domain: 1820+ PTR? 123.169.254.169.in-addr.arpa. (46)
22:48:46.774582 IP 169.254.169.123.ntp > ip-172-31-11-147.ec2.internal.55382: NTPv4, Server, length 48
22:48:46.774838 IP ip-172-31-0-2.ec2.internal.domain > ip-172-31-11-147.ec2.internal.52318: 1820 NOK domain 0/1/0 (105)
"data.txt" 868L, 170902B
```

Initially our load balancer is having exchange of the packets with my local computer. Then the load balancer server sends an ARP request for knowing its mac address. Since i have requested for a website , the load balancer sends the tcp request to any of the four web servers and hence there is an exchange of packet between the load balancer and the web server.

## EC2 Backup and Restore

1. Find the volume id attached to your instance. Go to Instances in AWS Management Console and check your instance id associated with the volume id.

The screenshot shows the AWS EC2 Instances page. On the left, a sidebar lists navigation options: Instances (with a New link), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances (with a New link), Dedicated Hosts, Scheduled Instances, Capacity Reservations, Images (with AMIs and AMI Catalog), Elastic Block Store (with Volumes and Snapshots, and Lifecycle Manager), and Network & Security (with Security Groups). The main content area displays a table titled "Instances (1/5) Info" with the following data:

Name	Instance ID	Instance state	Instance type	Status check	Alarm stat
Server1	i-0fa7f3b29ff9ee848	Running	t2.micro	2/2 checks passed	No alarms
Server2	i-0828023da8d464cc1	Running	t2.micro	2/2 checks passed	No alarms
Server3	i-06f259f07443706dd	Running	t2.micro	2/2 checks passed	No alarms
Server4	i-0ba9d57ae52467452	Running	t2.micro	2/2 checks passed	No alarms
<b>Load Balancer</b>	<b>i-058e17b4797c96976</b>	<b>Running</b>	<b>t2.micro</b>	<b>2/2 checks passed</b>	<b>No alarms</b>

Below the table, a detailed view for the Load Balancer instance (i-058e17b4797c96976) is shown. The "Details" tab is selected, displaying the following information:

- Instance ID: i-058e17b4797c96976 (Load Balancer)
- Public IPv4 address: 44.192.113.56 | [open address](#)
- Private IPv4 addresses: 172.31.11.147
- Public IPv4 DNS: ec2-44-192-113-56.compute-1.amazonaws.com | [open address](#)
- IPv6 address: -
- Instance state: Running

At the bottom of the page, there are links for Feedback, English (US), and a toolbar with various Mac OS X application icons.

The screenshot shows the AWS EC2 Volumes page. The sidebar is identical to the previous screenshot. The main content area displays a table titled "Volumes (5)" with the following data:

Name	Volume ID	Type	Size	IOPS	Throughput	Snapshot	Created
-	vol-0053b60f3e7d1804e	gp2	8 GiB	100	-	snap-0c1ac78...	2022/04/01 11:14 GMT-4
-	vol-0d80358be6e9f2440	gp2	8 GiB	100	-	snap-0c1ac78...	2022/04/01 11:14 GMT-4
-	vol-0fb96015395aa503c	gp2	8 GiB	100	-	snap-0c1ac78...	2022/04/01 11:14 GMT-4
-	vol-09f488f0959227e1f	gp2	8 GiB	100	-	snap-0c1ac78...	2022/04/01 11:14 GMT-4
-	vol-08b4c0112ccb74e33	gp2	8 GiB	100	-	snap-0c1ac78...	2022/04/01 11:14 GMT-4

Below the table, a message says "Select a volume above". At the bottom of the page, there are links for https://us-east-1.console.aws.amazon.com/ec2/v2/home?region=us-east-1#Volu..., 2022, Amazon Web Services, Inc. or its affiliates., Privacy, Terms, and Cookie preferences, along with a toolbar with various Mac OS X application icons.

## 2. Create a backup of the volume attached to the instance.

We will use the below command for backing the volume.

```
aws ec2 create-snapshot --volume-id vol-0053b60f3e7d1804e --description "Load Balancer Backup"
```

The screenshot shows a web browser window with multiple tabs open. The active tab displays a terminal session on the AWS Lambda console. The command run was:

```
aws ec2 create-snapshot --volume-id vol-0053b60f3e7d1804e --description "Load Balancer Backup"
```

The terminal output shows the creation of a new snapshot with the following details:

```
{ "Description": "Load Balancer Backup", "Encrypted": false, "OwnerId": "619795865798", "Progress": "", "SnapshotId": "snap-0b033d91bd7bcc57", "StartTime": "2022-04-01T23:44:44.835000+00:00", "Status": "pending", "VolumeId": "vol-0053b60f3e7d1804e", "VolumeSize": 8, "Tags": [] }
```

The browser interface includes a sidebar with navigation links like Home, Modules, Discussions, and Courses. On the right, there's a sidebar titled "Learner Lab - Foundational Level" with links to various AWS documentation pages.

The screenshot shows the AWS Management Console with the EC2 service selected. The left sidebar lists various EC2 resources: Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Scheduled Instances, Capacity Reservations, Images, AMIs, Elastic Block Store, Volumes, Snapshots, Network & Security, Security Groups, Elastic IPs, Placement Groups, Key Pairs, and Network Interfaces.

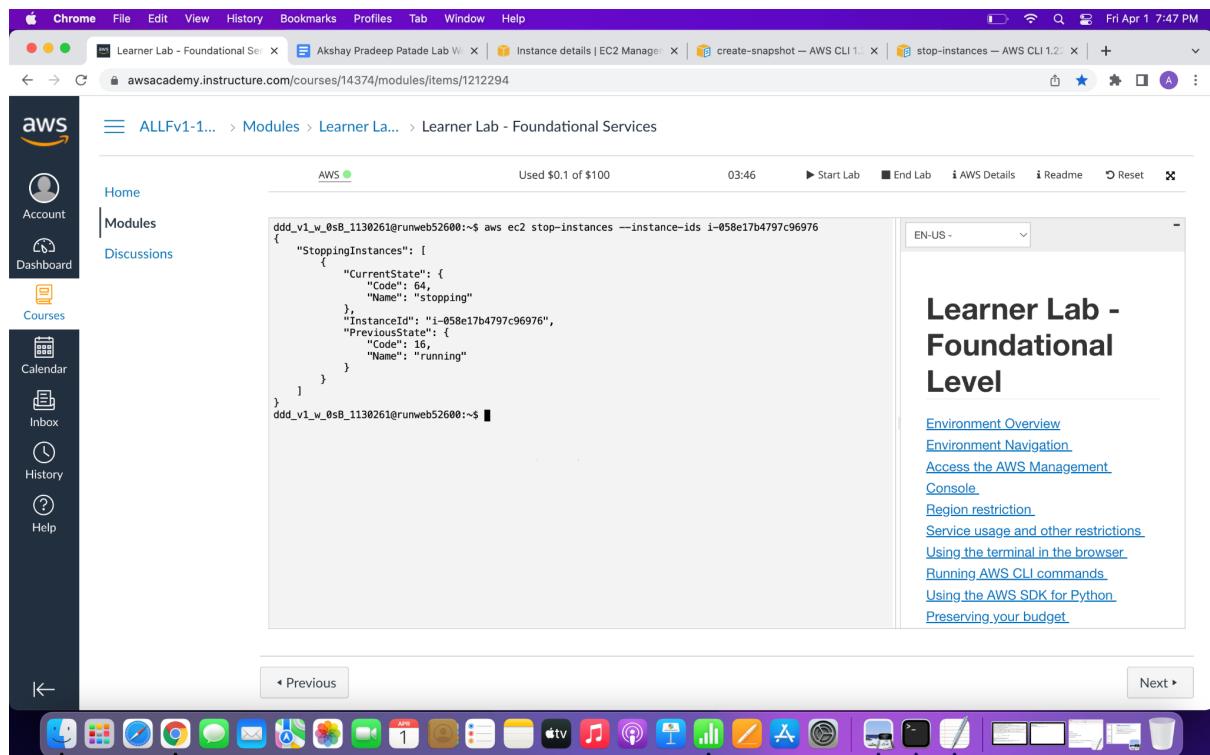
The main content area is titled "Snapshots (1)" and shows a table with one item:

Name	Snapshot ID	Size	Description	Storage...	Snapshot status	Started
-	snap-0b033d91bd7bcc57	8 GiB	Load Balancer Backup	Standard	Pending	2022/04/01 19:44 C

At the bottom of the page, there are links for "Select a snapshot above." and "Create snapshot". The browser address bar shows the URL: <https://us-east-1.console.aws.amazon.com/ec2/v2/home?region=us-east-1#Snapshots>.

### 3. Stopping the Load balancer instance so that we can detach its volume

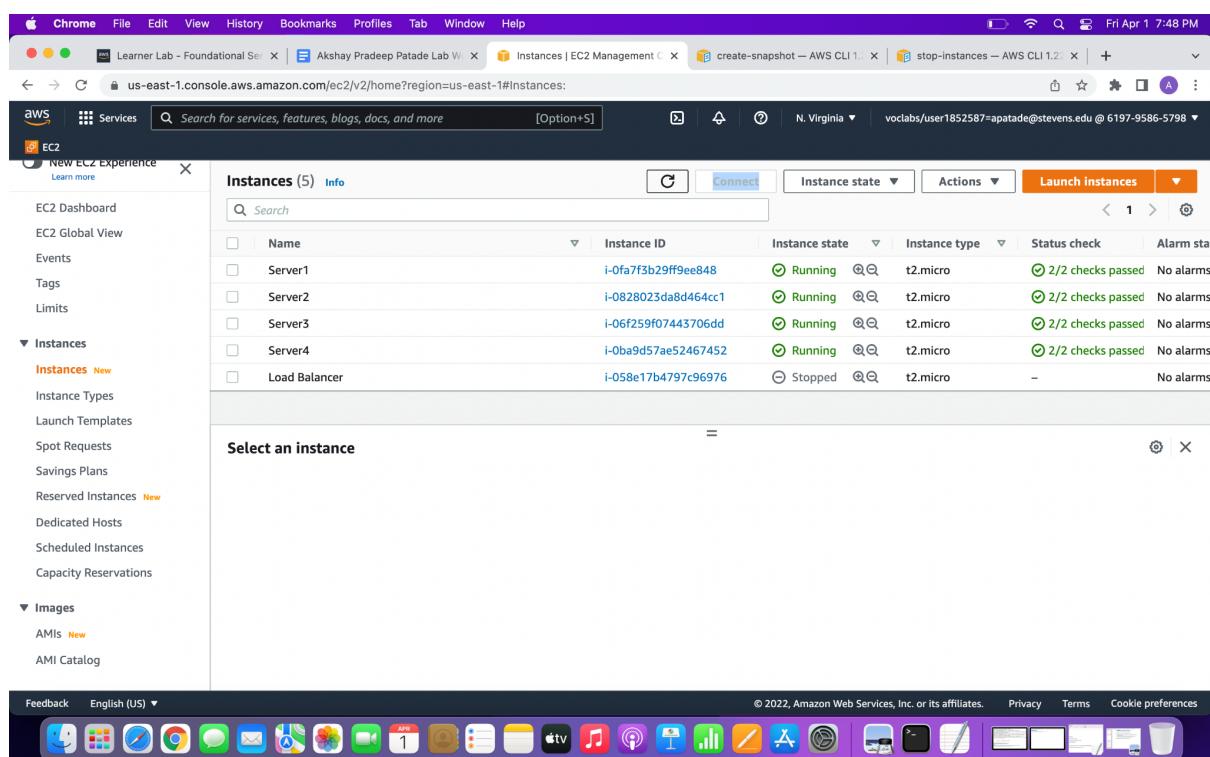
```
aws ec2 stop-instances --instance-ids i-058e17b4797c96976
```



The screenshot shows a web browser window with multiple tabs open. The main content area displays a terminal session with the following command and output:

```
ddd_v1_w_0sB_1130261@runweb52600:~$ aws ec2 stop-instances --instance-ids i-058e17b4797c96976
{
  "StoppingInstances": [
    {
      "CurrentState": {
        "Code": 64,
        "Name": "stopping"
      },
      "InstanceId": "i-058e17b4797c96976",
      "PreviousState": {
        "Code": 16,
        "Name": "running"
      }
    }
]
ddd_v1_w_0sB_1130261@runweb52600:~$
```

To the right of the terminal, there is a sidebar titled "Learner Lab - Foundational Level" containing links to various AWS resources and documentation.



The screenshot shows the AWS EC2 Instances page. The left sidebar navigation includes "EC2 Dashboard", "EC2 Global View", "Events", "Tags", "Limits", "Instances" (selected), "Instance Types", "Launch Templates", "Spot Requests", "Savings Plans", "Reserved Instances", "Dedicated Hosts", "Scheduled Instances", "Capacity Reservations", "Images", "AMIs", and "AMI Catalog".

The main content area displays a table titled "Instances (5)" with the following data:

Name	Instance ID	Instance state	Instance type	Status check	Alarm stat
Server1	i-0fa7f3b29ff9ee848	Running	t2.micro	2/2 checks passed	No alarms
Server2	i-0828023da8d464cc1	Running	t2.micro	2/2 checks passed	No alarms
Server3	i-06f259f07443706dd	Running	t2.micro	2/2 checks passed	No alarms
Server4	i-0ba9d57ae52467452	Running	t2.micro	2/2 checks passed	No alarms
Load Balancer	i-058e17b4797c96976	Stopped	t2.micro	-	No alarms

A modal dialog box titled "Select an instance" is open at the bottom of the page.

#### 4. Detach the volume from the Load Balancer.

```
aws ec2 detach-volume --volume-id vol-0053b60f3e7d1804e
```

#### 5. Creating a new instance.

```
aws ec2 run-instances --image-id ami-0c02fb55956c7d316 --count 1 --instance-type t2.micro  
--key-name ec2-key --security-group-ids sg-0264063663678a9ab --subnet-id  
subnet-02d6e208cf471bfd2
```

## 6. Stopping the instance.

```
aws ec2 stop-instances --instance-ids i-06eaac57677cd3de3
```

The screenshot shows a web browser window with multiple tabs open. The active tab is a Lambda function terminal window titled 'awsacademy.instructure.com/courses/14374/modules/items/1212294'. The terminal output shows an error message:

```
ddd_v1_w_0sB_1130261@runweb52610:~$ aws ec2 stop-instances --instance-ids i-06eac57677cd3de3
An error occurred (InvalidInstanceID.Malformed) when calling the StopInstances operation: Invalid id: "i-06eac57677cd3de3"
ddd_v1_w_0sB_1130261@runweb52610:~$ aws ec2 stop-instances --instance-ids i-06eac57677cd3de3
{
  "StoppingInstances": [
    {
      "CurrentState": {
        "Code": 64,
        "Name": "stopping"
      },
      "InstanceId": "i-06eac57677cd3de3",
      "PreviousState": {
        "Code": 16,
        "Name": "running"
      }
    }
}
ddd_v1_w_0sB_1130261@runweb52610:~$
```

To the right of the terminal, there is a sidebar with the title 'Learner Lab - Foundational Level' and several links:

- Environment Overview
- Environment Navigation
- Access the AWS Management Console
- Region restriction
- Service usage and other restrictions
- Using the terminal in the browser
- Running AWS CLI commands
- Using the AWS SDK for Python
- Preserving your budget

The screenshot shows the AWS EC2 Instances page at 'us-east-1.console.aws.amazon.com/ec2/v2/home?region=us-east-1#instances'. The left sidebar shows navigation options like Services, EC2, and various instance-related links. The main table lists six instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm stat
Server1	i-0fa7f3b29ff9ee848	Running	t2.micro	2/2 checks passed	No alarms
Server2	i-0828023da8d464cc1	Running	t2.micro	2/2 checks passed	No alarms
Server3	i-06f259f07443706dd	Running	t2.micro	2/2 checks passed	No alarms
Server4	i-0ba9d57ae52467452	Running	t2.micro	2/2 checks passed	No alarms
Load Balancer	i-058e17b4797c96976	Stopped	t2.micro	-	No alarms
Load Balancer Backup	i-06eac57677cd3de3	Stopped	t2.micro	-	No alarms

## 7. Detaching its volume.

`aws ec2 detach-volume --volume-id vol-046596d5131a763cf`

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with navigation links like Events, Tags, Limits, Instances (with sub-links for Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Scheduled Instances, Capacity Reservations), Images (AMIs, AMI Catalog), and Elastic Block Store (Volumes, Snapshots). The main area displays a table titled 'Instances (6) Info' with columns: Name, Instance ID, Instance state, Instance type, Status check, and Alarm stat. The table lists six instances: Server1, Server2, Server3, Server4, Load Balancer, and Load Balancer Backup. All instances are running t2.micro instances. Below the table is a section titled 'Select an instance'.

Name	Instance ID	Instance state	Instance type	Status check	Alarm stat
Server1	i-0fa7f3b29ff9ee848	Running	t2.micro	2/2 checks passed	No alarms
Server2	i-0828023da8d464cc1	Running	t2.micro	2/2 checks passed	No alarms
Server3	i-06f259f07443706dd	Running	t2.micro	2/2 checks passed	No alarms
Server4	i-0ba9d57ae52467452	Running	t2.micro	2/2 checks passed	No alarms
Load Balancer	i-058e17b4797c96976	Stopped	t2.micro	-	No alarms
Load Balancer Backup	i-06eaac57677cd3de3	Stopped	t2.micro	-	No alarms

## 8. Attaching a new instance with the existing volume.

```
aws ec2 attach-volume --volume-id vol-0053b60f3e7d1804e --instance-id i-06eaac57677cd3de3 --device /dev/xvda
```

Learner Lab - Foundational Services

Used \$0.1 of \$100      03:26      Start Lab      End Lab      AWS Details      Readme      Reset

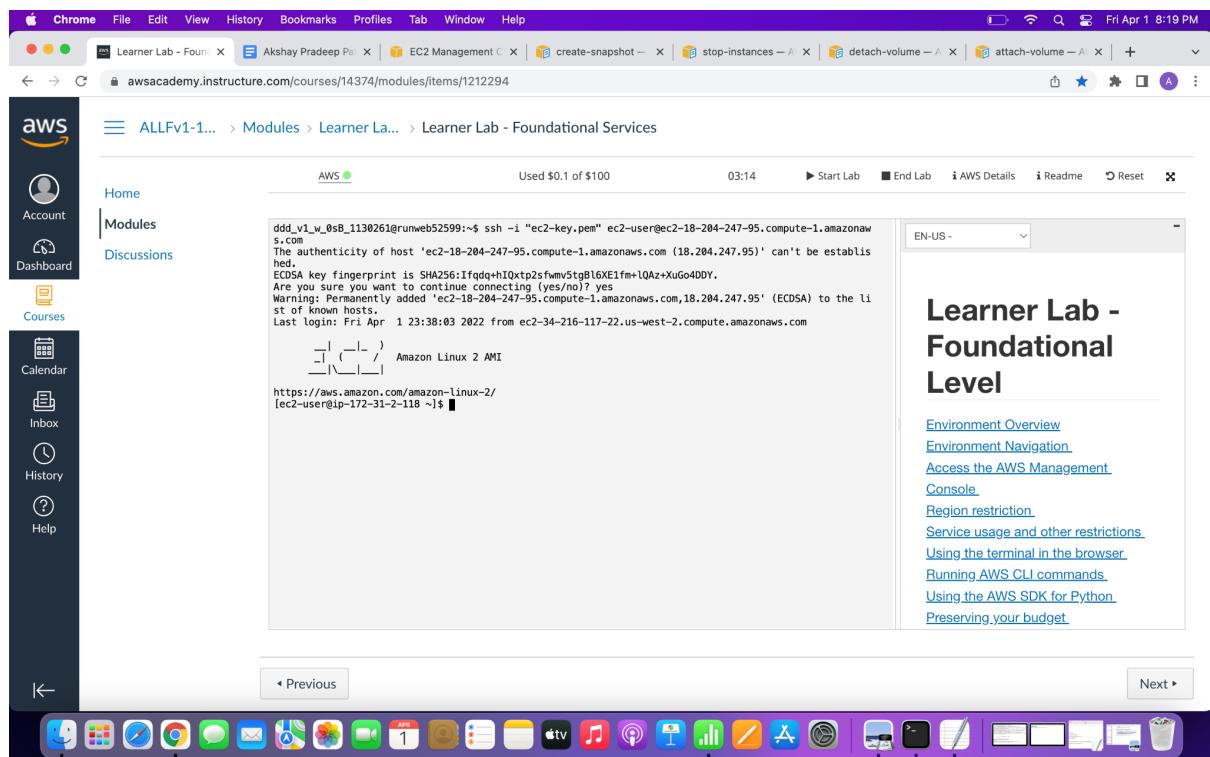
**Learner Lab - Foundational Level**

- Environment Overview
- Environment Navigation
- Access the AWS Management Console
- Region restriction
- Service usage and other restrictions
- Using the terminal in the browser
- Running AWS CLI commands
- Using the AWS SDK for Python
- Preserving your budget

## 9. Now again start the instance.

Name	Instance ID	Instance state	Instance type	Status check	Alarm stat
Server1	i-0fa7f3b29ff9ee848	Running	t2.micro	2/2 checks passed	No alarms
Server2	i-0828023da8d464cc1	Running	t2.micro	2/2 checks passed	No alarms
Server3	i-06f259f07443706dd	Running	t2.micro	2/2 checks passed	No alarms
Server4	i-0ba9d57ae52467452	Running	t2.micro	2/2 checks passed	No alarms
Load Balancer	i-058e17b4797c96976	Stopped	t2.micro	-	No alarms
Load Balancer Backup	i-06eaac57677cd3de3	Running	t2.micro	2/2 checks passed	No alarms

## 10. Check if you are able to connect to the server or not using ssh.



11. Finally, Start the nginx server and check whether you are able to access the website or not.

