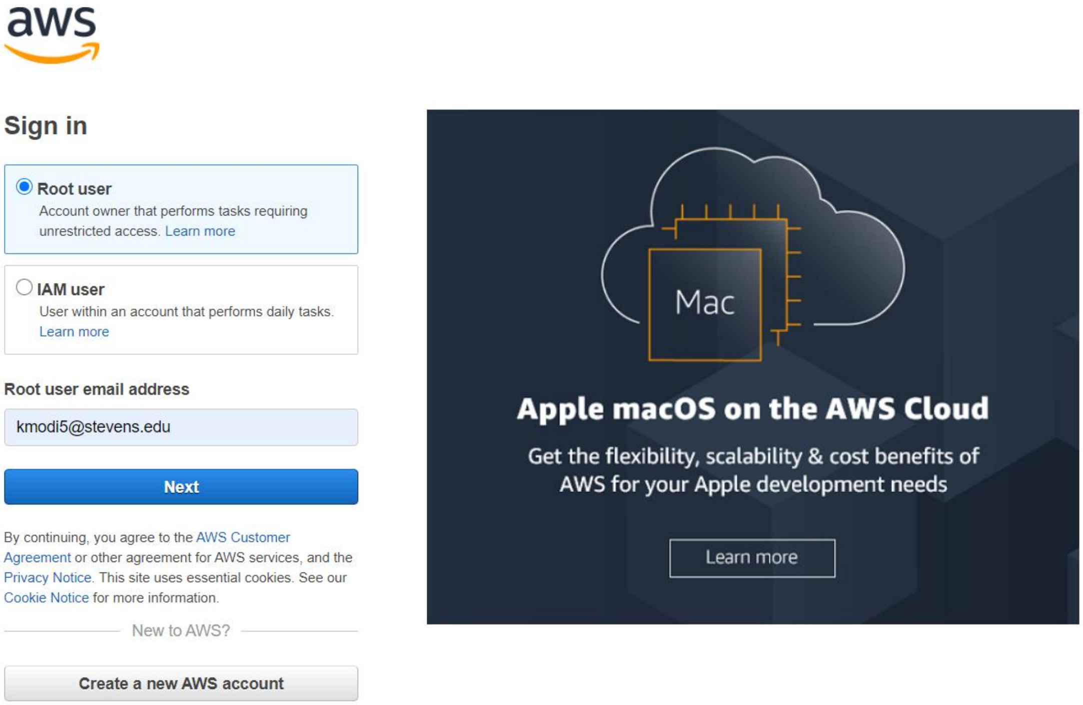
Lab 2: CS 524

In this assignment, you will learn to develop and load-balance your own infrastructure (a server farm) while applying your knowledge of DNS and other application entities and protocols. You will also learn to use Cloud Storage for back-up and recovery of your data. In addition, you will perform rudimentary analysis of the IP traffic. You will use Nginx (a popular http server available at http://nginx.org/) to host a simple website on four Amazon EC2 instances; you will also configure an Nginx server on another instance, which will act as a load balancer. You will learn how to distribute networking workload across multiple servers.

ANS.

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



Sign in

● Root user
Account owner that performs tasks requiring unrestricted access. Learn more

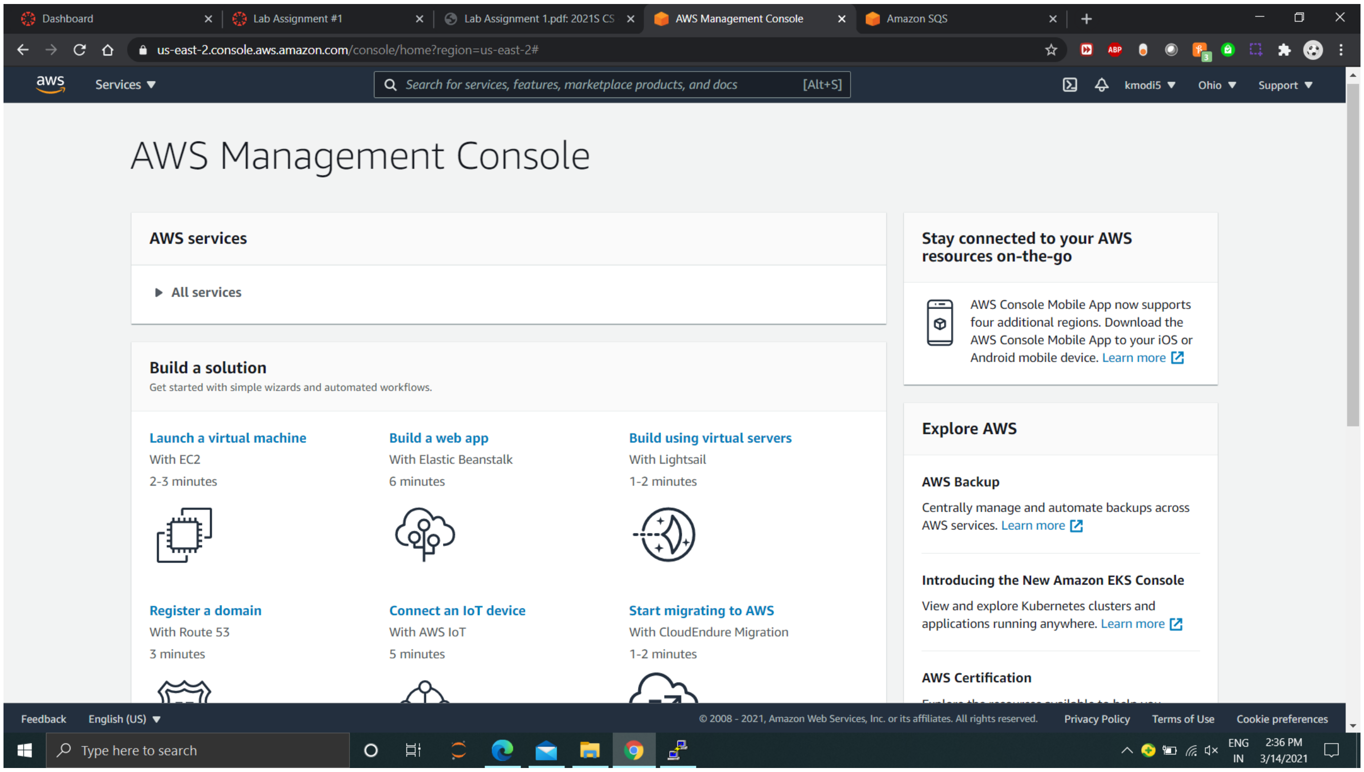
○ IAM user
User within an account that performs daily tasks. Learn more

Root user email address: kmodi5@stevens.edu

By continuing, you agree to the AWS Customer Agreement or other agreement for AWS services, and the Privacy Notice. This site uses essential cookies. See our Cookie Notice for more information.

aws
Next
New to AWS?
Create a new AWS account
Mac
Apple macOS on the AWS Cloud
Get the flexibility, scalability & cost benefits of AWS for your Apple development needs
Learn more

b. Go into AWS Console and Launch a virtual machine



AWS Management Console

AWS services

▸ All services

Build a solution

Get started with simple wizards and automated workflows.

Launch a virtual machine
With EC2
2-3 minutes

Build a web app
With Elastic Beanstalk
6 minutes

Build using virtual servers
With Lightsail
1-2 minutes

Register a domain
With Route 53
3 minutes

Connect an IoT device
With AWS IoT
5 minutes

Start migrating to AWS
With CloudEndure Migration
1-2 minutes

Stay connected to your AWS resources on-the-go

AWS Console Mobile App now supports four additional regions. Download the AWS Console Mobile App to your iOS or Android mobile device. Learn more

Explore AWS

AWS Backup

Centrally manage and automate backups across AWS services. Learn more

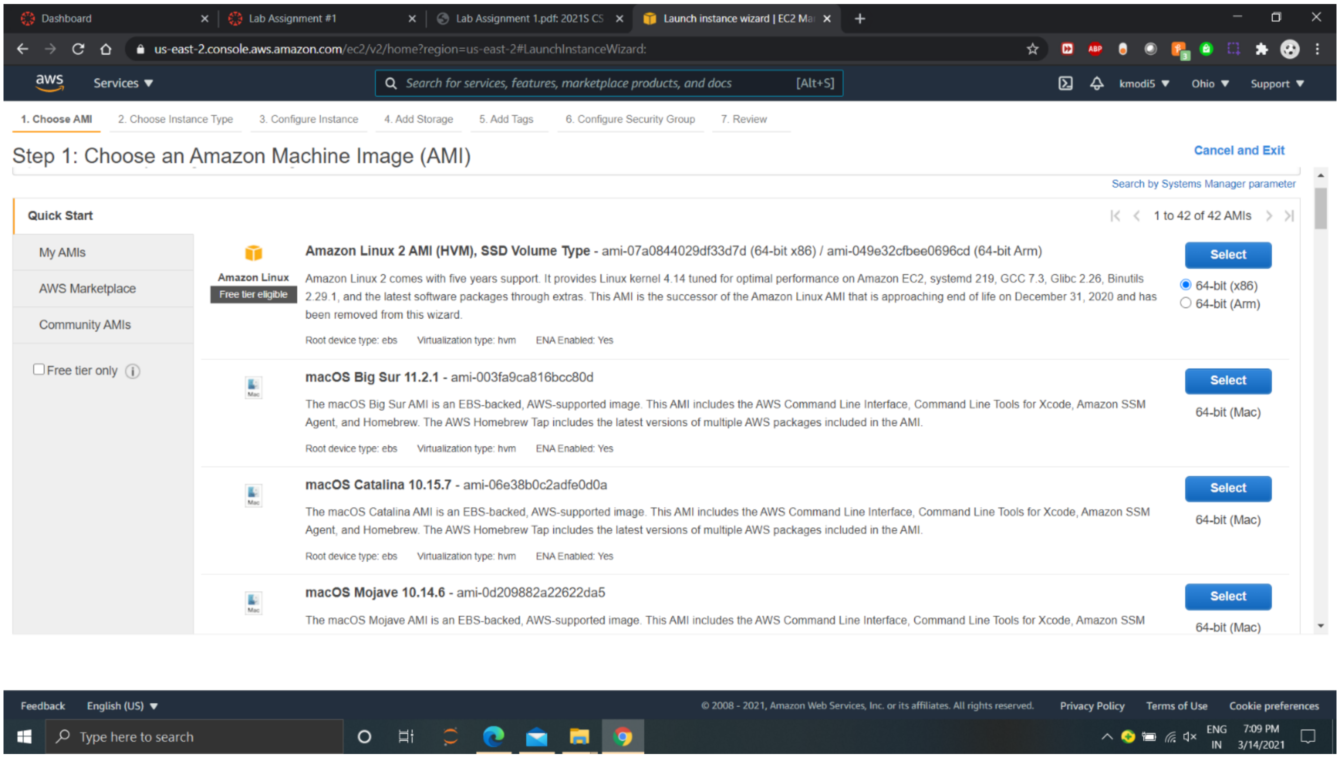
Introducing the New Amazon EKS Console

View and explore Kubernetes clusters and applications running anywhere. Learn more

AWS Certification

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

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



Step 1: Choose an Amazon Machine Image (AMI)

Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-07a0844029df33d7d (64-bit x86) / ami-049e32cfbee0696cd (64-bit Arm)

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 approaching end of life on December 31, 2020 and has been removed from this wizard.

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

● 64-bit (x86)
○ 64-bit (Arm)

macOS Big Sur 11.2.1 - ami-003fa9ca816bcc80d

The macOS Big Sur AMI is an EBS-backed, AWS-supported image. This AMI includes the AWS Command Line Interface, Command Line Tools for Xcode, Amazon SSM Agent, and Homebrew. The AWS Homebrew Tap includes the latest versions of multiple AWS packages included in the AMI.

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

64-bit (Mac)

macOS Catalina 10.15.7 - ami-06e38b0c2adfe0d0a

The macOS Catalina AMI is an EBS-backed, AWS-supported image. This AMI includes the AWS Command Line Interface, Command Line Tools for Xcode, Amazon SSM Agent, and Homebrew. The AWS Homebrew Tap includes the latest versions of multiple AWS packages included in the AMI.

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

64-bit (Mac)

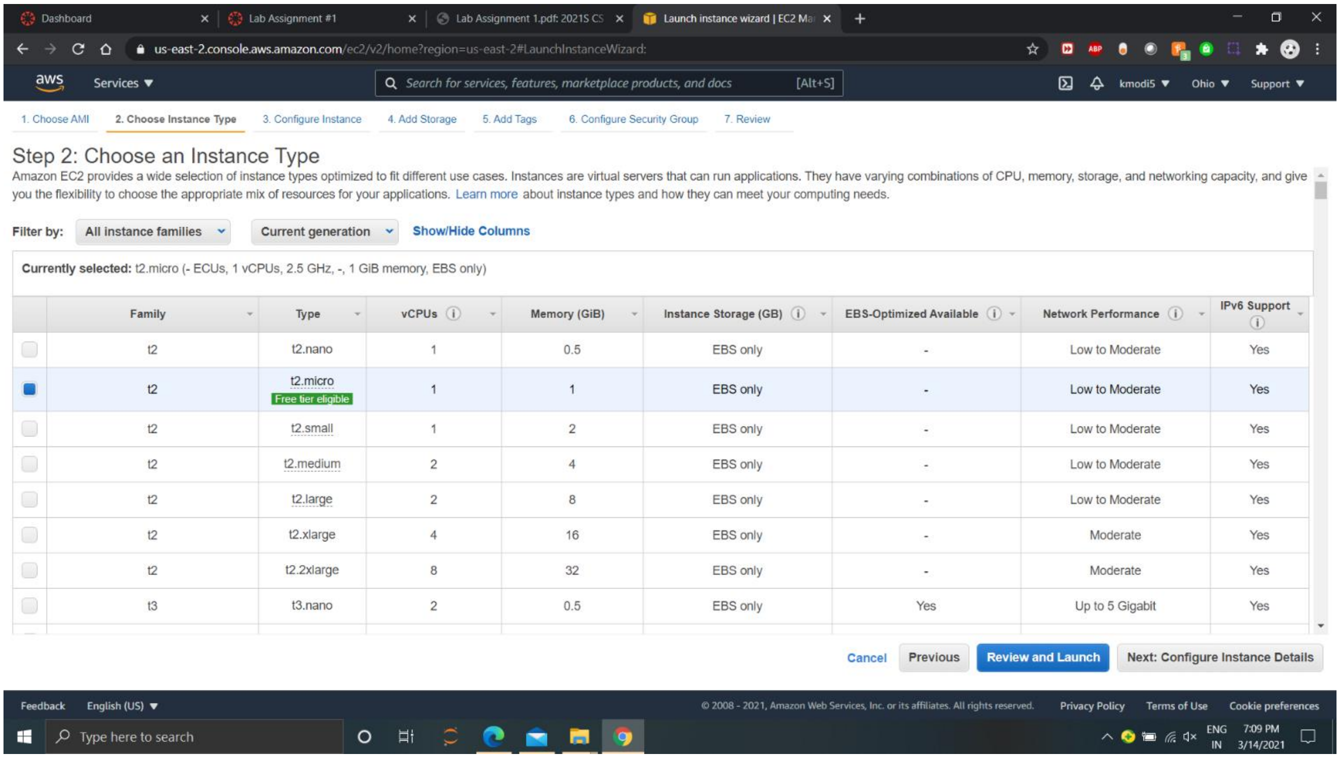
macOS Mojave 10.14.6 - ami-0d209882a22622da5

The macOS Mojave AMI is an EBS-backed, AWS-supported image. This AMI includes the AWS Command Line Interface, Command Line Tools for Xcode, Amazon SSM

64-bit (Mac)

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

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


Dashboard
Lab Assignment #1
Lab Assignment 1.pdf: 2021S CS
Launch instance wizard | EC2 Ma
us-east-2.console.aws.amazon.com/ec2/v2/home?region=us-east-2#LaunchInstanceWizard:
aws
Services ▼
Search for services, features, marketplace products, and docs [Alt+S]
kmodi5 ▼
Ohio ▼
Support ▼
1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review

Step 2: Choose an Instance Type

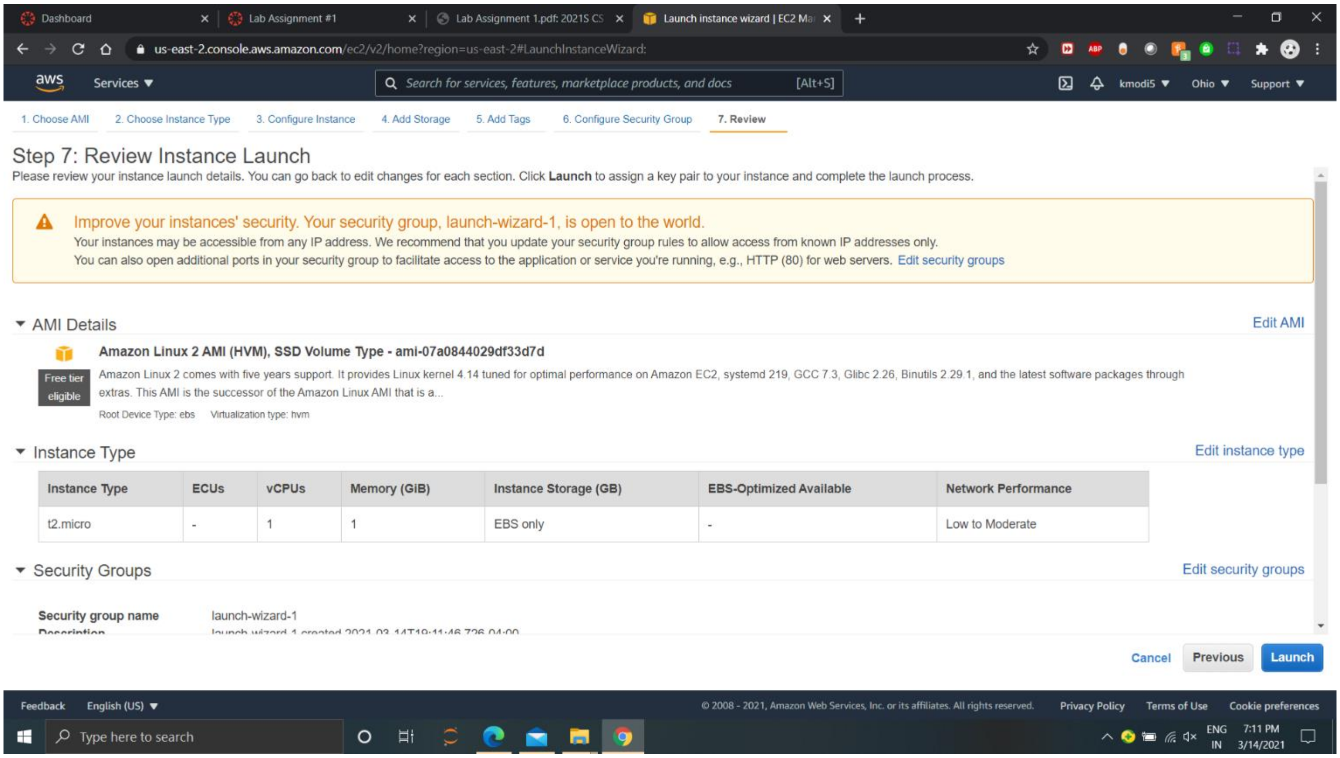
Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources 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
□	t2	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
■	t2	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
□	t2	t2.small	1	2	EBS only	-	Low to Moderate	Yes
□	t2	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
□	t2	t2.large	2	8	EBS only	-	Low to Moderate	Yes
□	t2	t2.xlarge	4	16	EBS only	-	Moderate	Yes
□	t2	t2.2xlarge	8	32	EBS only	-	Moderate	Yes
□	t3	t3.nano	2	0.5	EBS only	Yes	Up to 5 Gigabit	Yes

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


Dashboard
Lab Assignment #1
Lab Assignment 1.pdf: 2021S CS
Launch instance wizard | EC2 Ma
us-east-2.console.aws.amazon.com/ec2/v2/home?region=us-east-2#LaunchInstanceWizard:
aws
Services ▼
Search for services, features, marketplace products, and docs [Alt+S]
kmodi5 ▼
Ohio ▼
Support ▼
1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review

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

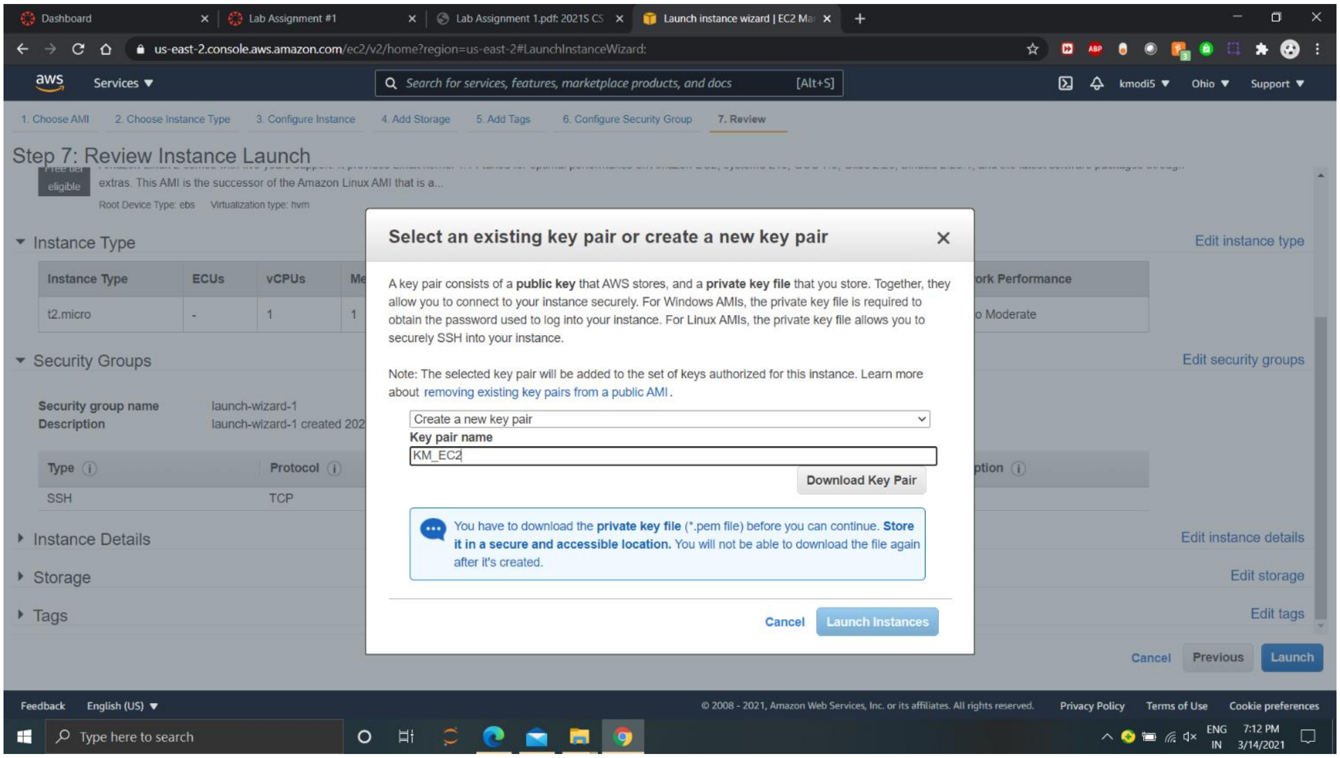
▼ Instance Type — Edit instance type

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	launch-wizard-1
Description	launch-wizard-1 created 2021-03-14T19:11:46.726-04:00

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



Step 7: Review Instance Launch

Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI.

Create a new key pair

Key pair name

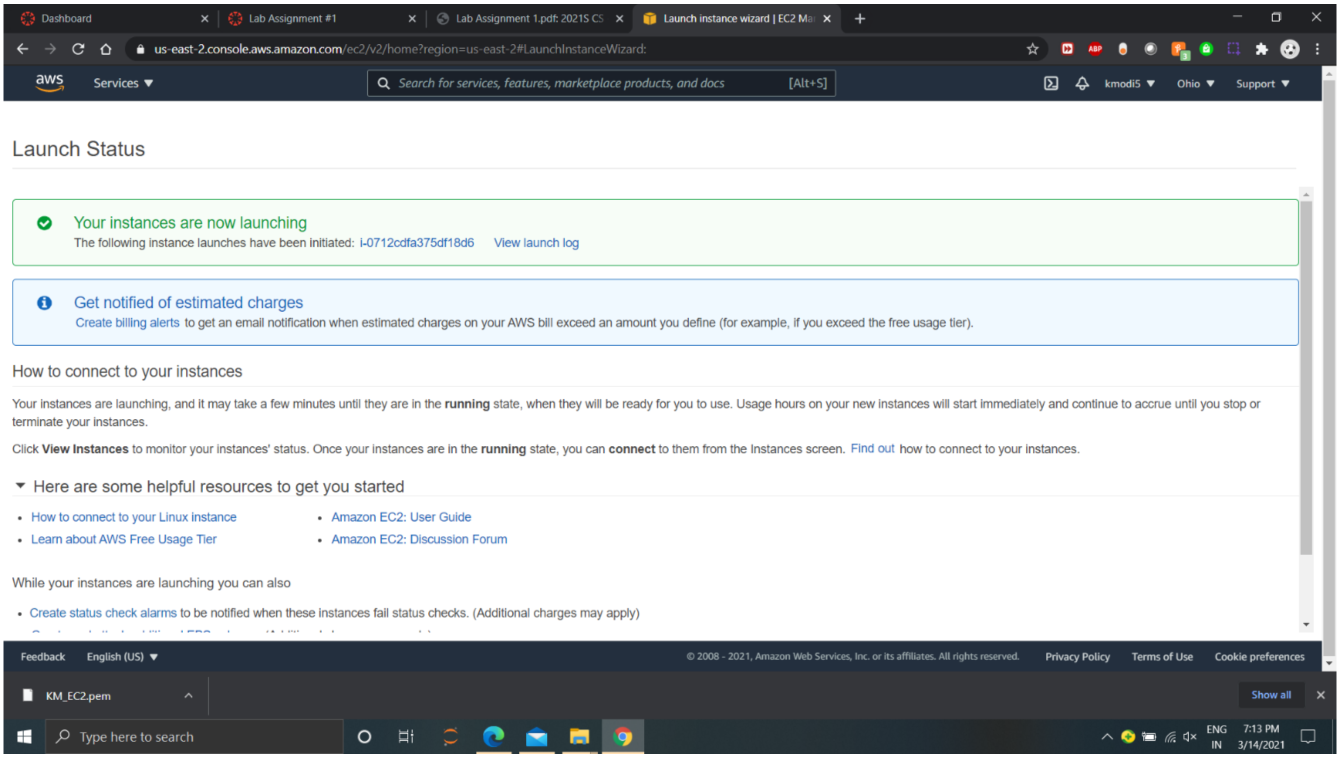
KM_EC2

Download Key Pair

You have to download the **private key file** (*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.

Cancel Launch Instances

g. Launch the instance and it should be initiated.



Launch Status

Your instances are now launching

The following instance launches have been initiated: i-0712cdfa375df18d6 View launch log

Get notified of estimated charges

Create billing alerts to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier).

How to connect to your instances

Your instances are launching, and it may take a few minutes until they are in the **running** state, when they will be ready for you to use. Usage hours on your new instances will start immediately and continue to accrue until you stop or terminate your instances.

Click **View Instances** to monitor your instances' status. Once your instances are in the **running** state, you can **connect** to them from the Instances screen. Find out how to connect to your instances.

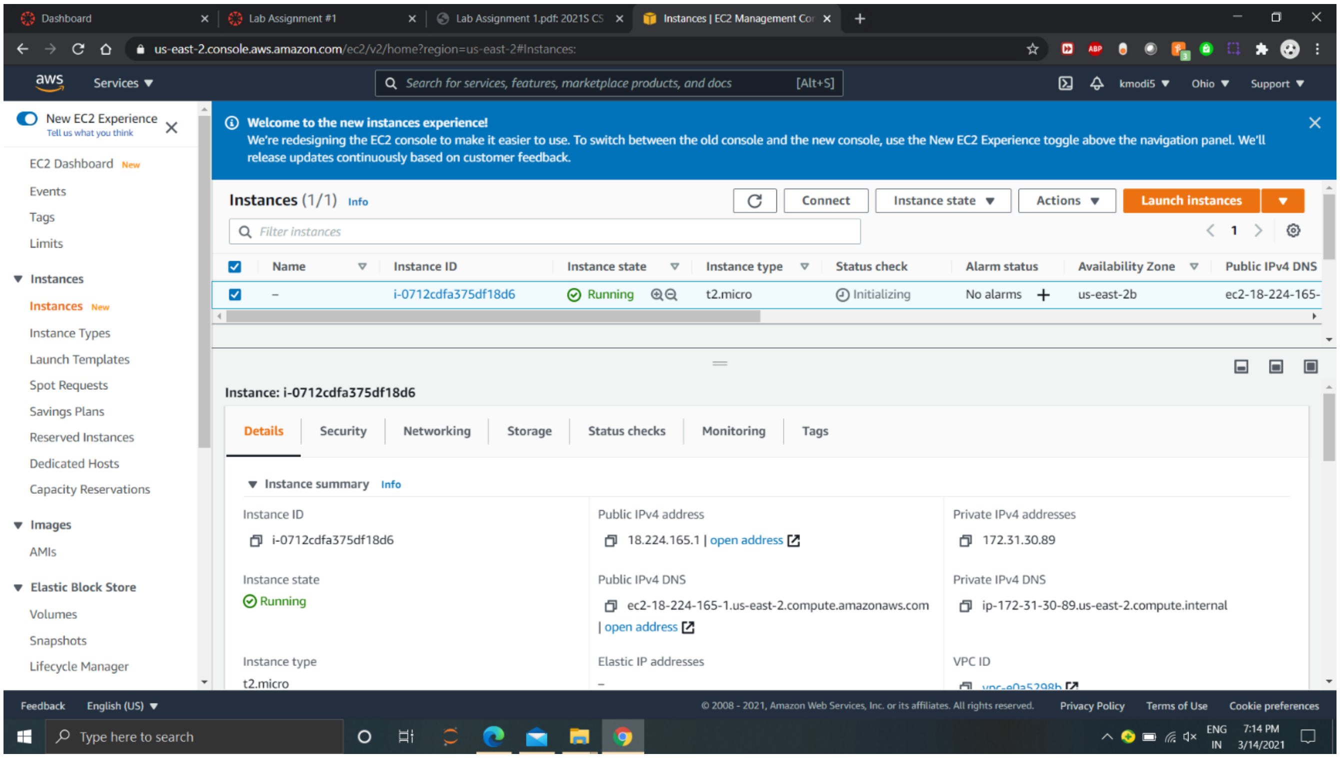
Here are some helpful resources to get you started

- How to connect to your Linux instance
- Learn about AWS Free Usage Tier
- Amazon EC2: User Guide
- Amazon EC2: Discussion Forum

While your instances are launching you can also

- Create status check alarms to be notified when these instances fail status checks. (Additional charges may apply)

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



Welcome to the new instances experience!
We're redesigning the EC2 console to make it easier to use. To switch between the old console and the new console, use the New EC2 Experience toggle above the navigation panel. We'll release updates continuously based on customer feedback.

Instances (1/1) Info

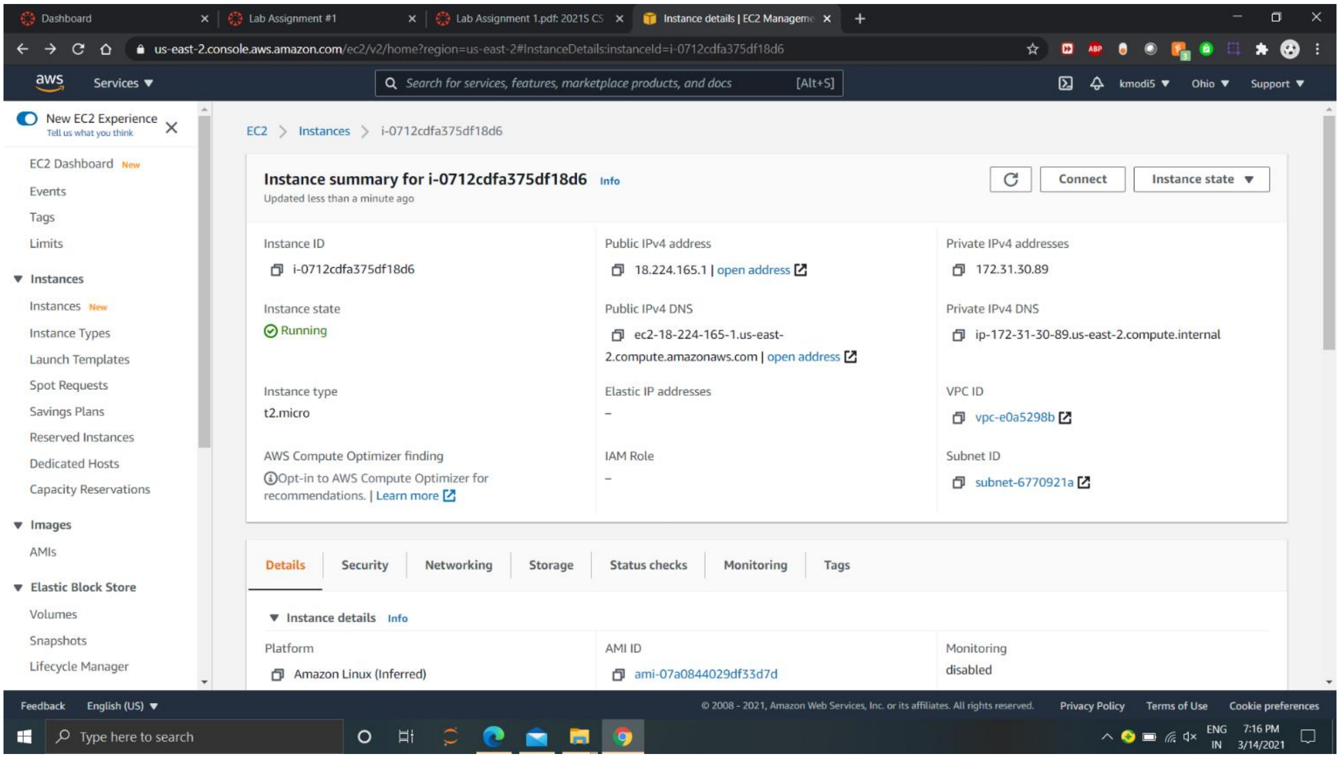
☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
☑	–	i-0712cdfa375df18d6	Running	t2.micro	Initializing	No alarms +	us-east-2b	ec2-18-224-165-

Instance: i-0712cdfa375df18d6

▼ Instance summary Info

Instance ID: i-0712cdfa375df18d6
Public IPv4 address: 18.224.165.1 | open address
Private IPv4 addresses: 172.31.30.89
Instance state: Running
Public IPv4 DNS: ec2-18-224-165-1.us-east-2.compute.amazonaws.com | open address
Private IPv4 DNS: ip-172-31-30-89.us-east-2.compute.internal
Instance type: t2.micro
Elastic IP addresses: –
VPC ID: vpc-e0a5298b

i. Connect to the instance and select a medium



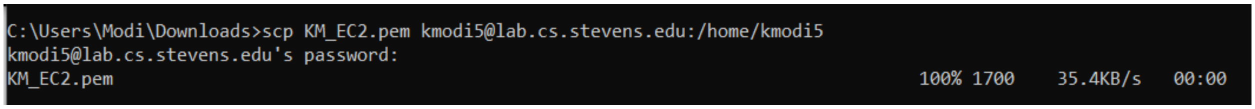
Instance summary for i-0712cdfa375df18d6 Info
Updated less than a minute ago

Instance ID: i-0712cdfa375df18d6
Public IPv4 address: 18.224.165.1 | open address
Private IPv4 addresses: 172.31.30.89
Instance state: Running
Public IPv4 DNS: ec2-18-224-165-1.us-east-2.compute.amazonaws.com | open address
Private IPv4 DNS: ip-172-31-30-89.us-east-2.compute.internal
Instance type: t2.micro
Elastic IP addresses: –
VPC ID: vpc-e0a5298b
AWS Compute Optimizer finding: Opt-in to AWS Compute Optimizer for recommendations. | Learn more
IAM Role: –
Subnet ID: subnet-6770921a

▼ Instance details Info

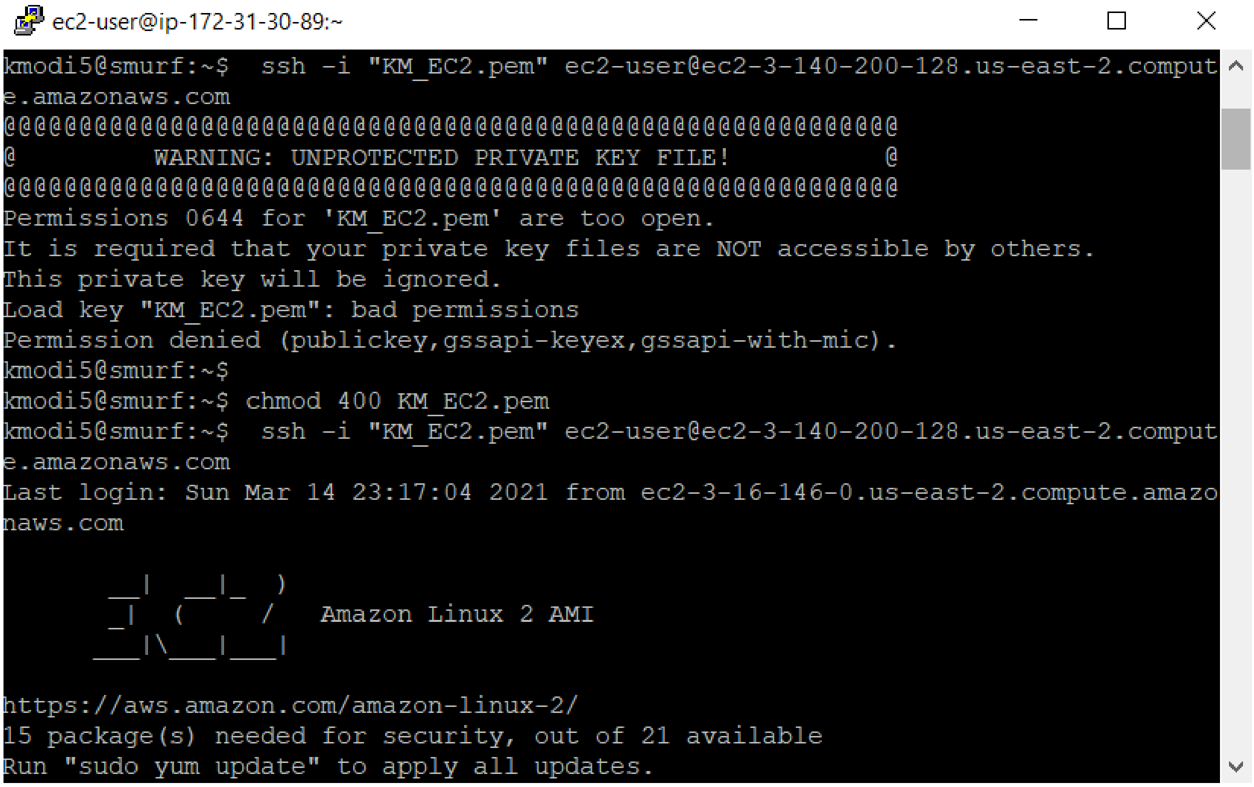
Platform: Amazon Linux (Inferred)
AMI ID: ami-07a0844029df33d7d
Monitoring: disabled

j. Using SCP to send pem private key to stevens server from your desktop

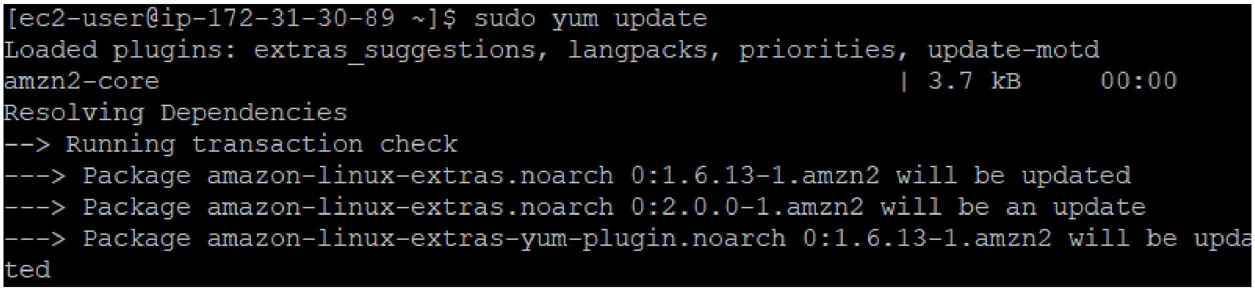


```
C:\Users\Modi\Downloads>scp KM_EC2.pem kmodi5@lab.cs.stevens.edu:/home/kmodi5
kmodi5@lab.cs.stevens.edu's password:
KM_EC2.pem                                                                100% 1700    35.4KB/s   00:00
```

k. Connecting to remote stevens server and giving right permissions

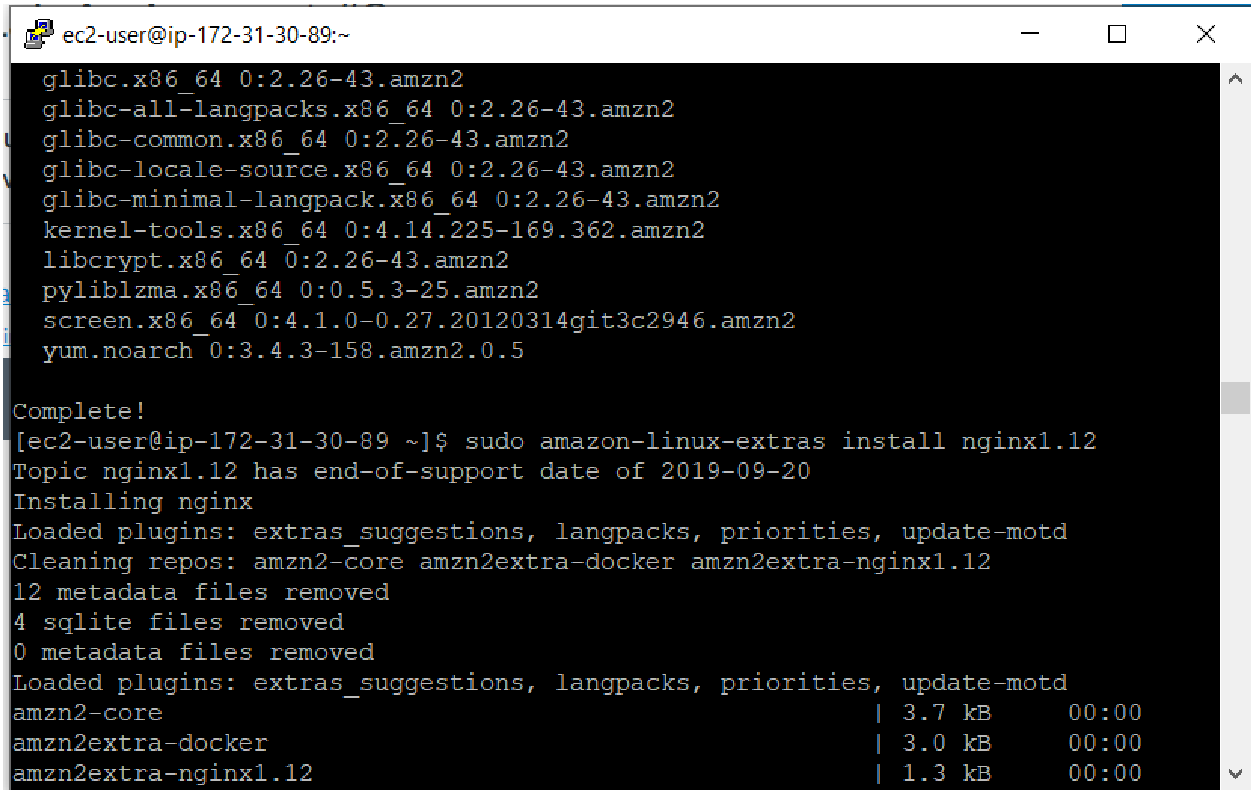


```
kmodi5@smurf:~$  ssh -i "KM_EC2.pem" ec2-user@ec2-3-140-200-128.us-east-2.comput
e.amazonaws.com
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@         WARNING: UNPROTECTED PRIVATE KEY FILE!          @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0644 for 'KM_EC2.pem' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "KM_EC2.pem": bad permissions
Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
kmodi5@smurf:~$
kmodi5@smurf:~$ chmod 400 KM_EC2.pem
kmodi5@smurf:~$  ssh -i "KM_EC2.pem" ec2-user@ec2-3-140-200-128.us-east-2.comput
e.amazonaws.com
Last login: Sun Mar 14 23:17:04 2021 from ec2-3-16-146-0.us-east-2.compute.amazo
naws.com

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
15 package(s) needed for security, out of 21 available
Run "sudo yum update" to apply all updates.
```



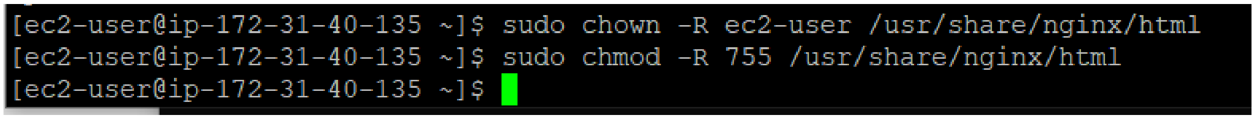
```
[ec2-user@ip-172-31-30-89 ~]$ sudo yum update
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core                                                    | 3.7 kB     00:00
Resolving Dependencies
--> Running transaction check
---> Package amazon-linux-extras.noarch 0:1.6.13-1.amzn2 will be updated
---> Package amazon-linux-extras.noarch 0:2.0.0-1.amzn2 will be an update
---> Package amazon-linux-extras-yum-plugin.noarch 0:1.6.13-1.amzn2 will be upda
ted
```

l. Install nginx on your aws instance



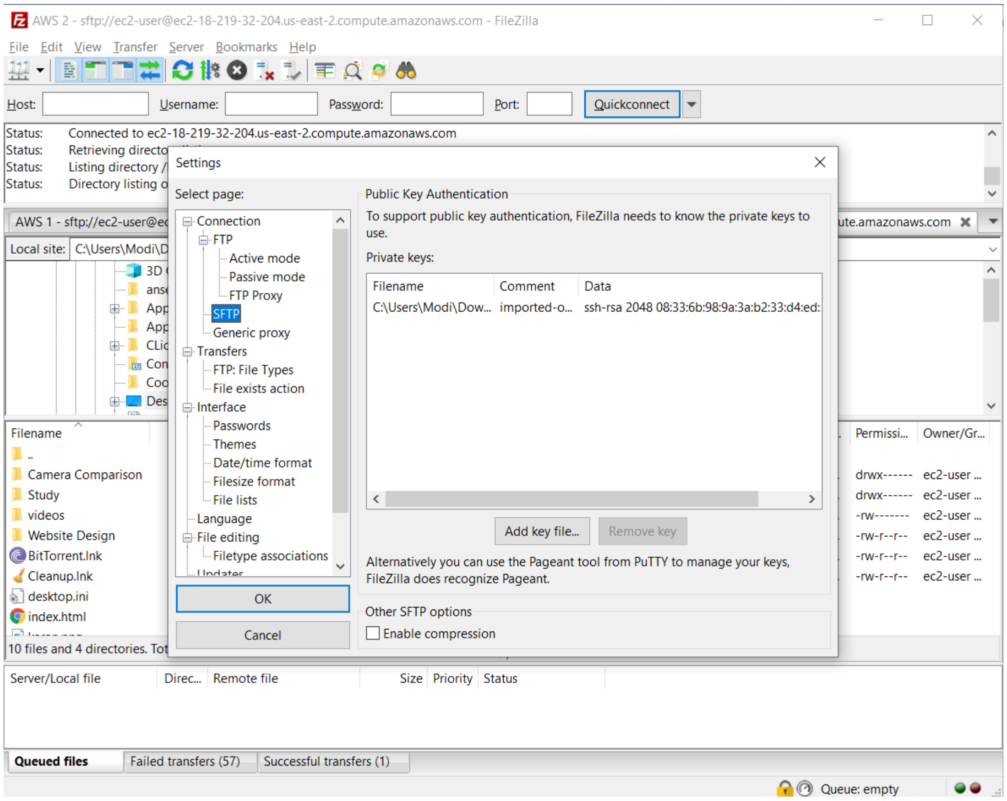
```
  glibc.x86_64 0:2.26-43.amzn2
  glibc-all-langpacks.x86_64 0:2.26-43.amzn2
  glibc-common.x86_64 0:2.26-43.amzn2
  glibc-locale-source.x86_64 0:2.26-43.amzn2
  glibc-minimal-langpack.x86_64 0:2.26-43.amzn2
  kernel-tools.x86_64 0:4.14.225-169.362.amzn2
  libcrypt.x86_64 0:2.26-43.amzn2
  pyliblzma.x86_64 0:0.5.3-25.amzn2
  screen.x86_64 0:4.1.0-0.27.20120314git3c2946.amzn2
  yum.noarch 0:3.4.3-158.amzn2.0.5

Complete!
[ec2-user@ip-172-31-30-89 ~]$ sudo amazon-linux-extras install nginx1.12
Topic nginx1.12 has end-of-support date of 2019-09-20
Installing nginx
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Cleaning repos: amzn2-core amzn2extra-docker amzn2extra-nginx1.12
12 metadata files removed
4 sqlite files removed
0 metadata files removed
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core                                                 | 3.7 kB     00:00
amzn2extra-docker                                          | 3.0 kB     00:00
amzn2extra-nginx1.12                                       | 1.3 kB     00:00
```

m. Press Y to install all dependencies
n. For easier handling we will use Filezilla to connect to our AWS Servers using SFTP: SSH FTP
o. First let’s give the right permissions to communicate and write to our servers.



```
[ec2-user@ip-172-31-40-135 ~]$ sudo chown -R ec2-user /usr/share/nginx/html
[ec2-user@ip-172-31-40-135 ~]$ sudo chmod -R 755 /usr/share/nginx/html
[ec2-user@ip-172-31-40-135 ~]$ 
```

p. Now we will import our Key <NAME>.pem to Filezilla for AWS authentication.



AWS 2 - sftp://ec2-user@ec2-18-219-32-204.us-east-2.compute.amazonaws.com - FileZilla

Status: Connected to ec2-18-219-32-204.us-east-2.compute.amazonaws.com
Status: Retrieving directo
Status: Listing directory /
Status: Directory listing o

Settings

Select page:

- Connection
  - FTP
    - Active mode
    - Passive mode
    - FTP Proxy
  - SFTP
  - Generic proxy
- Transfers
  - FTP: File Types
  - File exists action
- Interface
  - Passwords
  - Themes
  - Date/time format
  - Filesize format
  - File lists
- Language
- File editing
  - Filetype associations

Public Key Authentication

To support public key authentication, FileZilla needs to know the private keys to use.

Private keys:

Filename	Comment	Data
C:\Users\Modi\Dow...	imported-o...	ssh-rsa 2048 08:33:6b:98:9a:3a:b2:33:d4:ed:

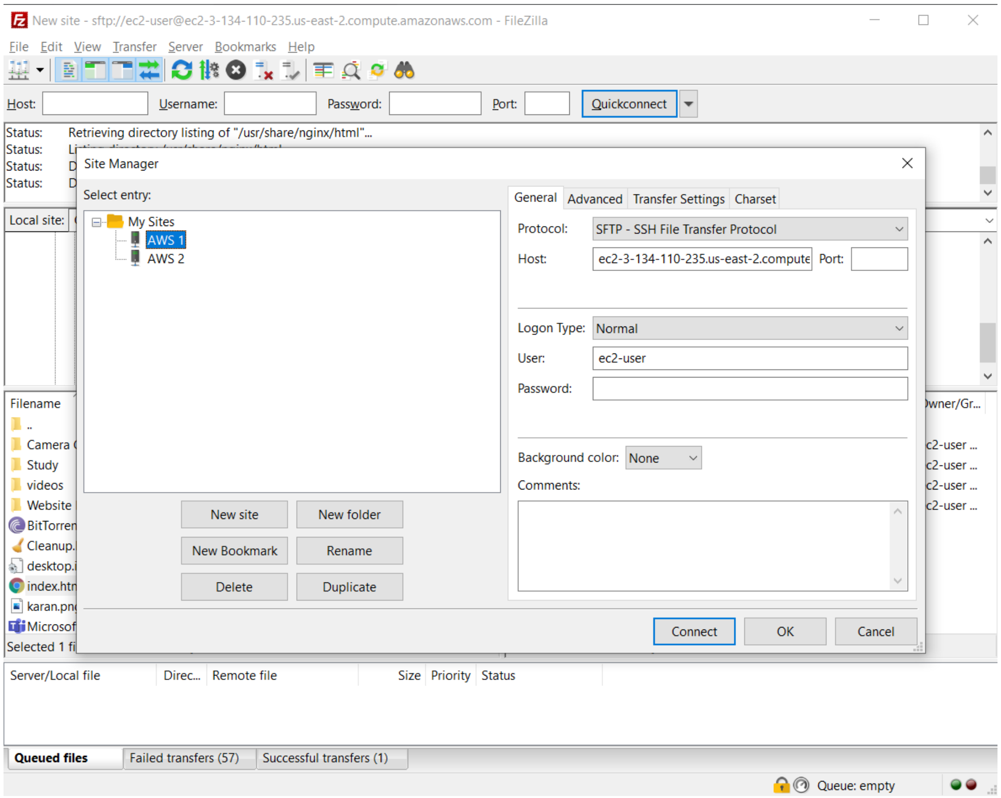
Alternatively you can use the Pageant tool from PuTTY to manage your keys, FileZilla does recognize Pageant.

Other SFTP options

□ Enable compression

File Edit View Transfer Server Bookmarks Help
Host: Username: Password: Port: Quickconnect
AWS 1 - sftp://ec2-user@ec
Local site: C:\Users\Modi\D
Filename
Camera Comparison
Study
videos
Website Design
BitTorrent.lnk
Cleanup.lnk
desktop.ini
index.html
10 files and 4 directories. Tot
Permissi... Owner/Gr...
drwx------ ec2-user ...
drwx------ ec2-user ...
-rw------- ec2-user ...
-rw-r--r-- ec2-user ...
-rw-r--r-- ec2-user ...
-rw-r--r-- ec2-user ...
Add key file... Remove key
OK
Cancel
Server/Local file Direc... Remote file Size Priority Status
Queued files Failed transfers (57) Successful transfers (1)
Queue: empty

q. Now we will enter the credentials for connecting to our servers



New site - sftp://ec2-user@ec2-3-134-110-235.us-east-2.compute.amazonaws.com - FileZilla

Status: Retrieving directory listing of "/usr/share/nginx/html"...

Site Manager

Select entry:

- My Sites
  - AWS 1
  - AWS 2

General | Advanced | Transfer Settings | Charset

Protocol: SFTP - SSH File Transfer Protocol

Host: ec2-3-134-110-235.us-east-2.compute Port:

Logon Type: Normal

User: ec2-user

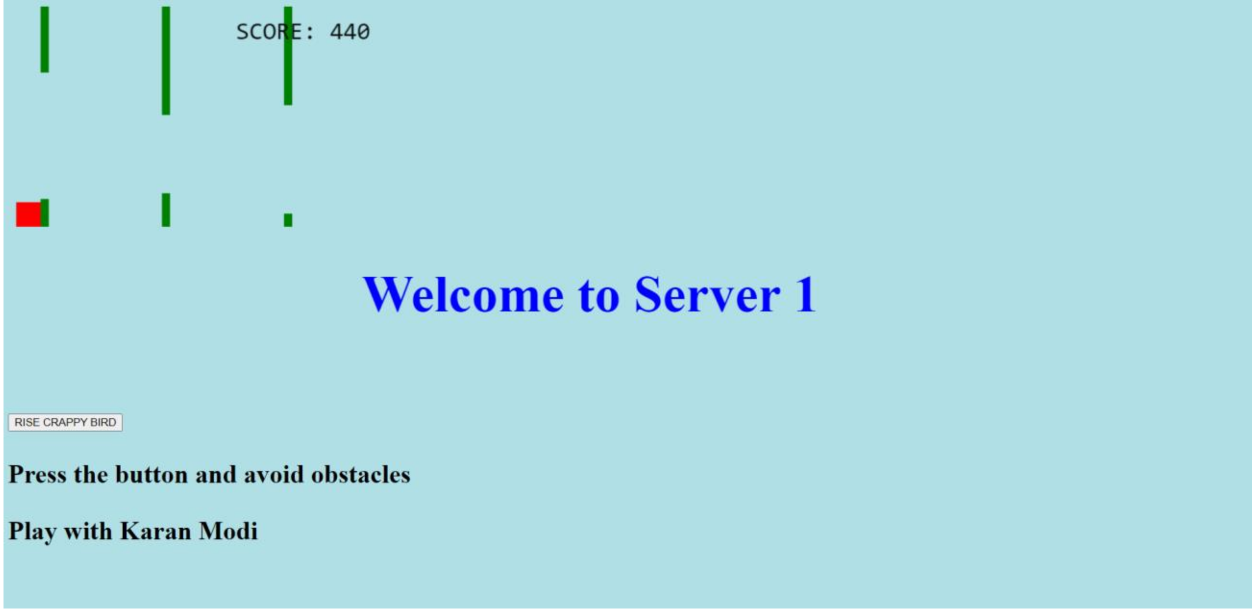
Password:

Background color: None

Comments:

File Edit View Transfer Server Bookmarks Help
Host: Username: Password: Port: Quickconnect
Local site:
Filename
Camera
Study
videos
Website
BitTorren
Cleanup.
desktop.i
index.htr
karan.pn
Microsof
Selected 1 fi
New site New folder
New Bookmark Rename
Delete Duplicate
Connect OK Cancel
Server/Local file Direc... Remote file Size Priority Status
Queued files Failed transfers (57) Successful transfers (1)
Queue: empty

r. Once our files are sent using Filezilla , start Nginx again

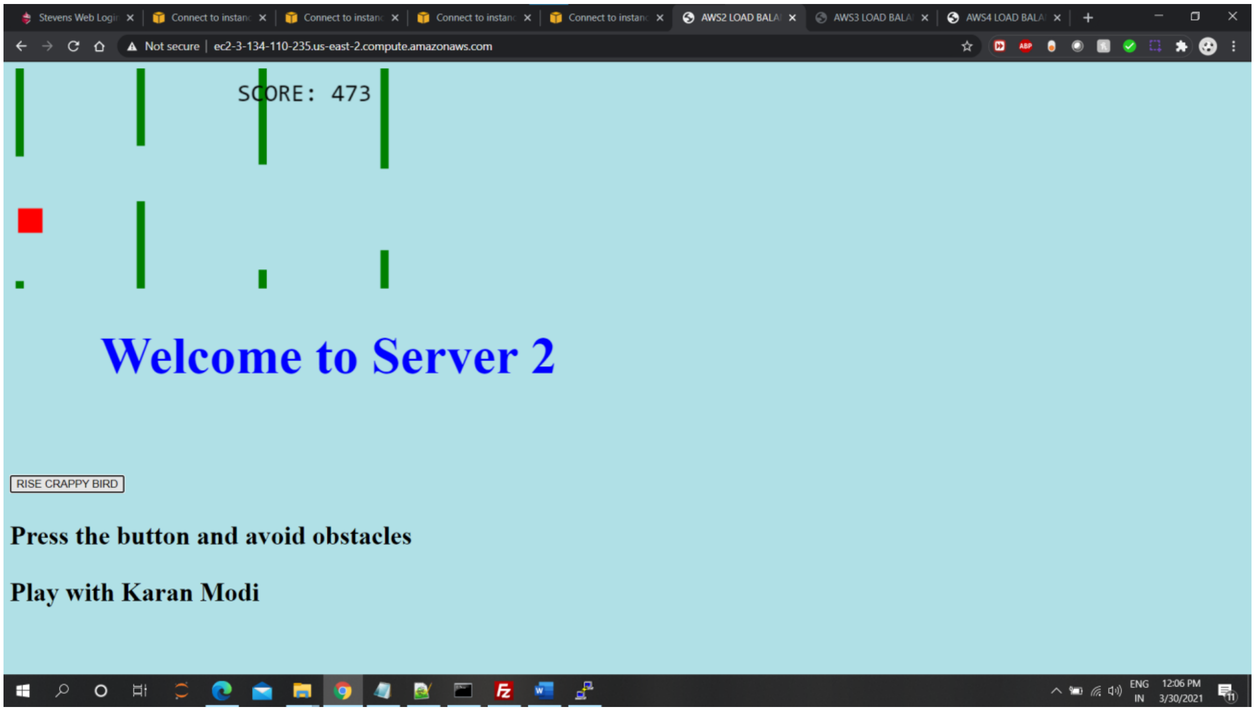


Welcome to Server 1

Press the button and avoid obstacles

Play with Karan Modi

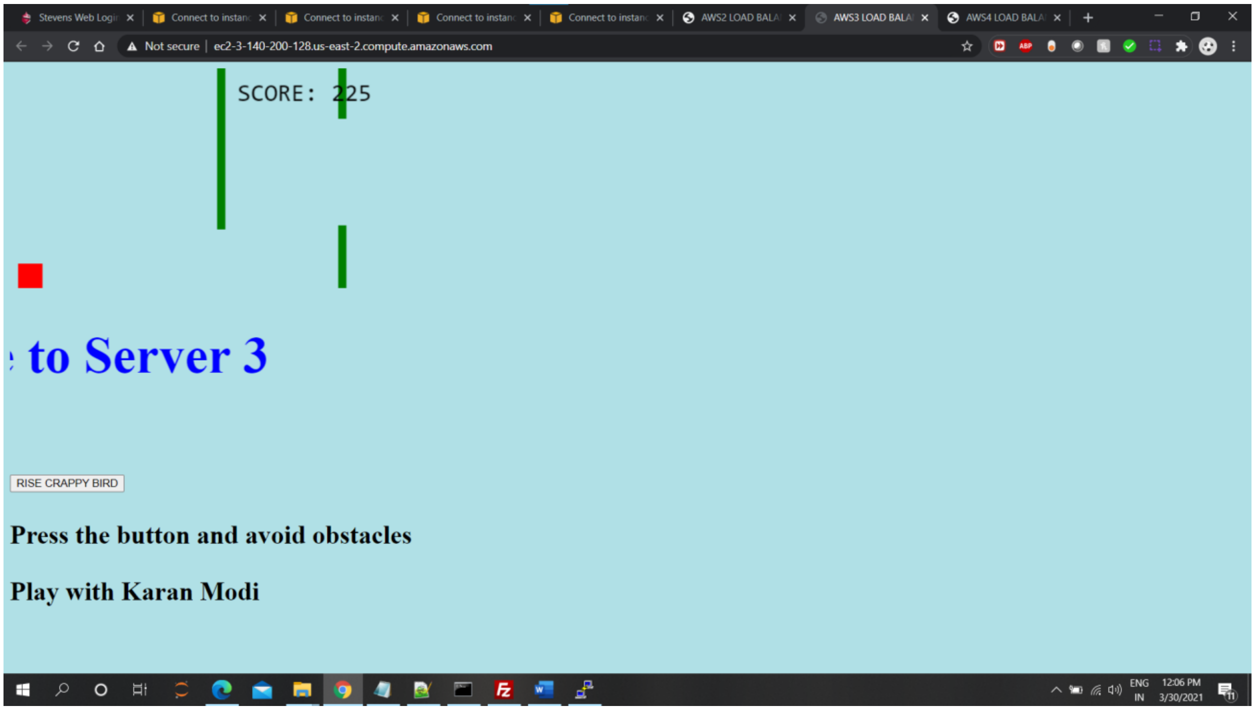
s. Similarly for Server 2, 3, 4 and Load Balancer



Welcome to Server 2

Press the button and avoid obstacles

Play with Karan Modi



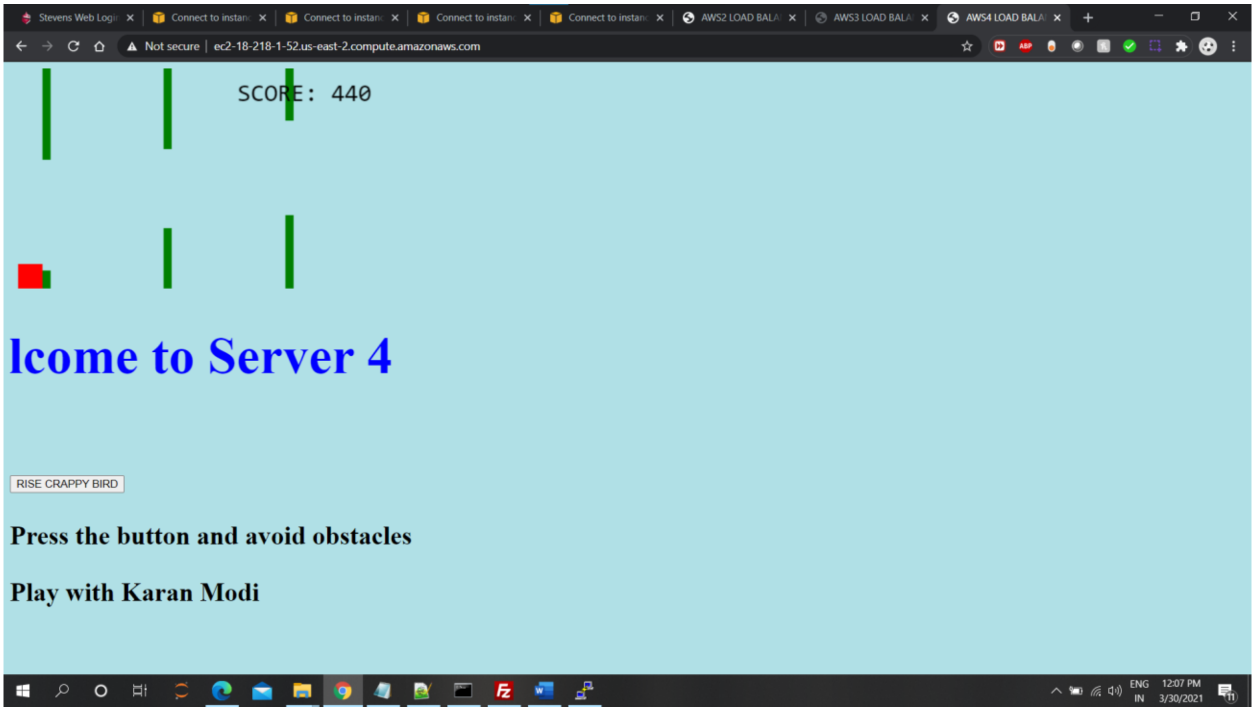
ec2-3-140-200-128.us-east-2.compute.amazonaws.com

SCORE: 225

to Server 3

Press the button and avoid obstacles

Play with Karan Modi



ec2-18-218-1-52.us-east-2.compute.amazonaws.com

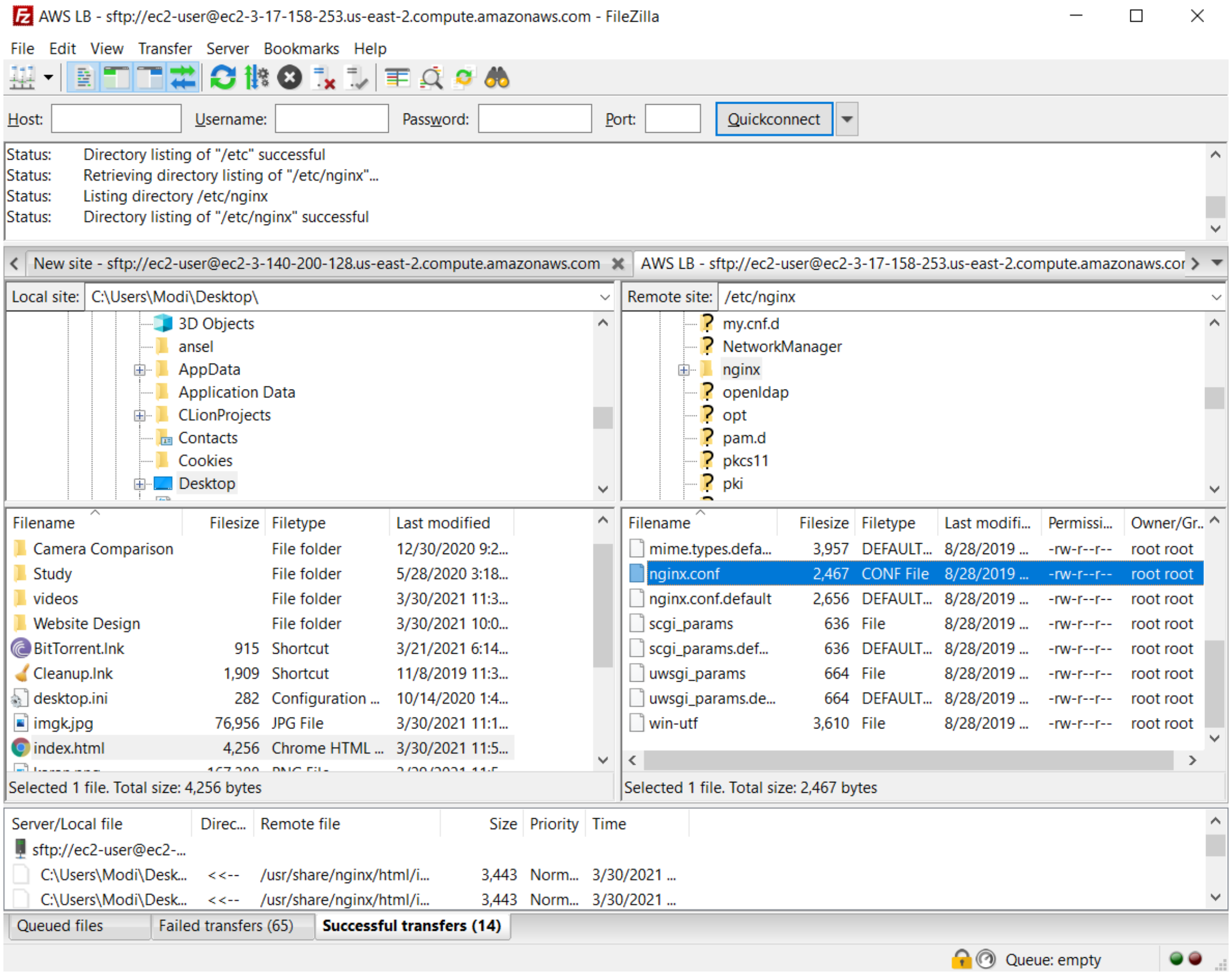
SCORE: 440

lcome to Server 4

Press the button and avoid obstacles

Play with Karan Modi

t. Go to etc/nginx and open nginx.conf file



AWS LB - sftp://ec2-user@ec2-3-17-158-253.us-east-2.compute.amazonaws.com - FileZilla

```
Status:     Directory listing of "/etc" successful
Status:     Retrieving directory listing of "/etc/nginx"...
Status:     Listing directory /etc/nginx
Status:     Directory listing of "/etc/nginx" successful
```

New site - sftp://ec2-user@ec2-3-140-200-128.us-east-2.compute.amazonaws.com

AWS LB - sftp://ec2-user@ec2-3-17-158-253.us-east-2.compute.amazonaws.coı

Local site: C:\Users\Modi\Desktop\

Remote site: /etc/nginx

Filename	Filesize	Filetype	Last modified
Camera Comparison		File folder	12/30/2020 9:2...
Study		File folder	5/28/2020 3:18...
videos		File folder	3/30/2021 11:3...
Website Design		File folder	3/30/2021 10:0...
BitTorrent.lnk	915	Shortcut	3/21/2021 6:14...
Cleanup.lnk	1,909	Shortcut	11/8/2019 11:3...
desktop.ini	282	Configuration ...	10/14/2020 1:4...
imgk.jpg	76,956	JPG File	3/30/2021 11:1...
index.html	4,256	Chrome HTML ...	3/30/2021 11:5...

Selected 1 file. Total size: 4,256 bytes

Filename	Filesize	Filetype	Last modifi...	Permissi...	Owner/Gr...
mime.types.defa...	3,957	DEFAULT...	8/28/2019 ...	-rw-r--r--	root root
nginx.conf	2,467	CONF File	8/28/2019 ...	-rw-r--r--	root root
nginx.conf.default	2,656	DEFAULT...	8/28/2019 ...	-rw-r--r--	root root
scgi_params	636	File	8/28/2019 ...	-rw-r--r--	root root
scgi_params.def...	636	DEFAULT...	8/28/2019 ...	-rw-r--r--	root root
uwsgi_params	664	File	8/28/2019 ...	-rw-r--r--	root root
uwsgi_params.de...	664	DEFAULT...	8/28/2019 ...	-rw-r--r--	root root
win-utf	3,610	File	8/28/2019 ...	-rw-r--r--	root root

Selected 1 file. Total size: 2,467 bytes

Server/Local file	Direc...	Remote file	Size	Priority	Time
sftp://ec2-user@ec2-...					
C:\Users\Modi\Desk...	<<--	/usr/share/nginx/html/i...	3,443	Norm...	3/30/2021 ...
C:\Users\Modi\Desk...	<<--	/usr/share/nginx/html/i...	3,443	Norm...	3/30/2021 ...

SERVER1_PUBLIC_DNS_NAME = ec2-52-15-177-236.us-east-2.compute.amazonaws.com

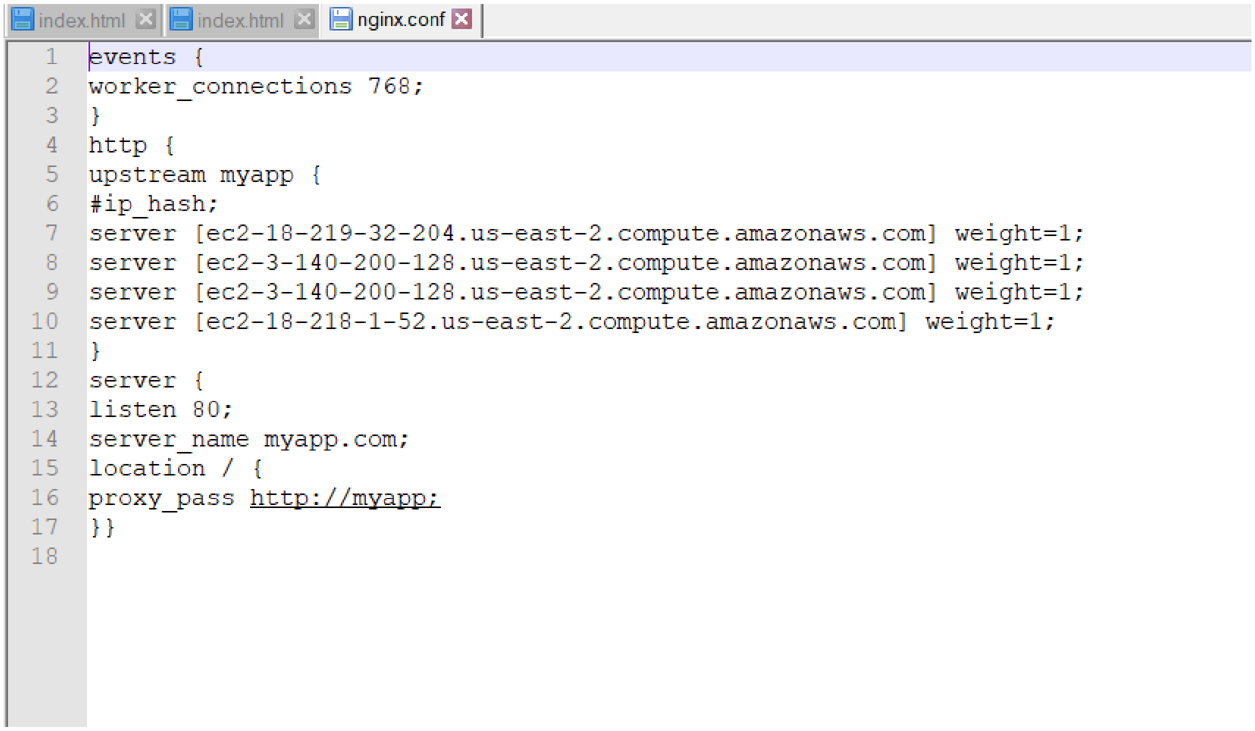
SERVER2_PUBLIC_DNS_NAME = ec2-3-141-19-134.us-east-2.compute.amazonaws.com

SERVER3_PUBLIC_DNS_NAME = ec2-18-188-106-12.us-east-2.compute.amazonaws.com

SERVER4_PUBLIC_DNS_NAME = ec2-3-140-239-6.us-east-2.compute.amazonaws.com

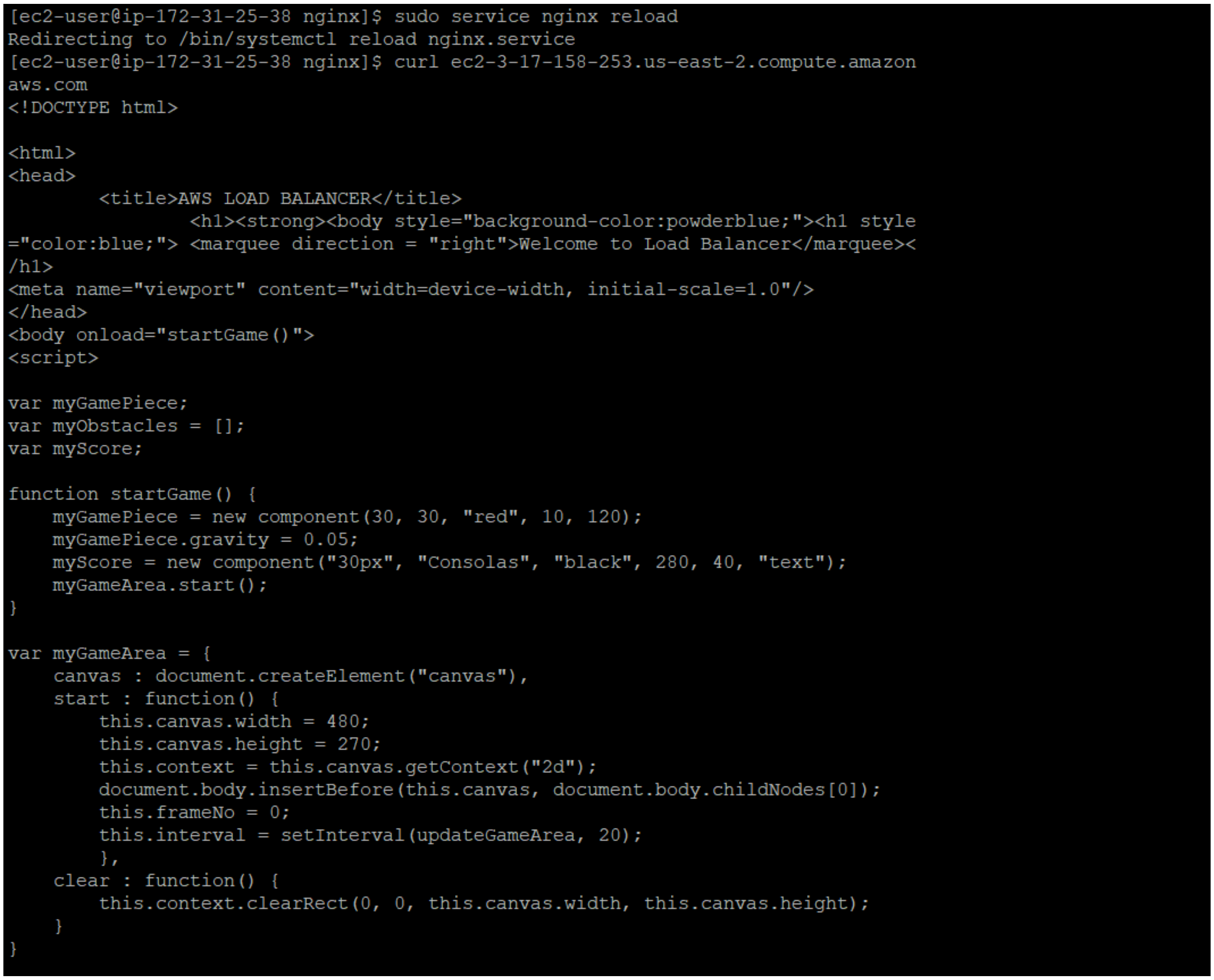
SERVERLB_PUBLIC_DNS_NAME = ec2-3-138-142-56.us-east-2.compute.amazonaws.com

u. Replace the nginx.conf file with below code



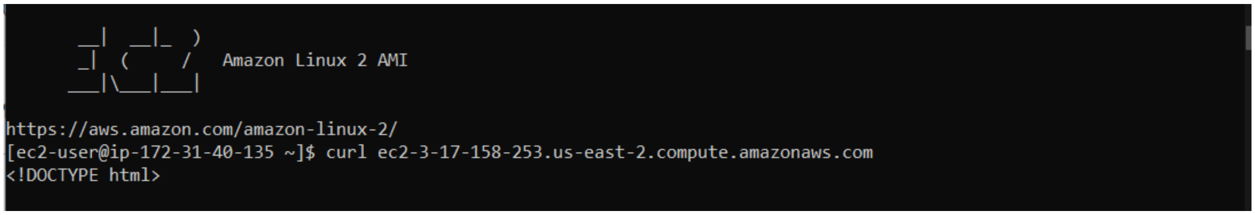
```
events {
worker_connections 768;
}
http {
upstream myapp {
#ip_hash;
server [ec2-18-219-32-204.us-east-2.compute.amazonaws.com] weight=1;
server [ec2-3-140-200-128.us-east-2.compute.amazonaws.com] weight=1;
server [ec2-3-140-200-128.us-east-2.compute.amazonaws.com] weight=1;
server [ec2-18-218-1-52.us-east-2.compute.amazonaws.com] weight=1;
}
server {
listen 80;
server_name myapp.com;
location / {
proxy_pass http://myapp;
}}
```

v. Reload nginx service and curl to Load Balancer



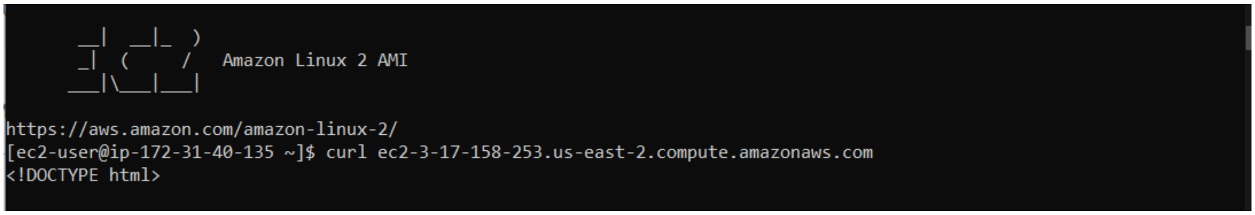
```
[ec2-user@ip-172-31-25-38 nginx]$ sudo service nginx reload
Redirecting to /bin/systemctl reload nginx.service
[ec2-user@ip-172-31-25-38 nginx]$ curl ec2-3-17-158-253.us-east-2.compute.amazon
aws.com
<!DOCTYPE html>

<html>
<head>
        <title>AWS LOAD BALANCER</title>
                <h1><strong><body style="background-color:powderblue;"><h1 style
="color:blue;"> <marquee direction = "right">Welcome to Load Balancer</marquee><
/h1>
<meta name="viewport" content="width=device-width, initial-scale=1.0"/>
</head>
<body onload="startGame()">
<script>

var myGamePiece;
var myObstacles = [];
var myScore;

function startGame() {
    myGamePiece = new component(30, 30, "red", 10, 120);
    myGamePiece.gravity = 0.05;
    myScore = new component("30px", "Consolas", "black", 280, 40, "text");
    myGameArea.start();
}

var myGameArea = {
    canvas : document.createElement("canvas"),
    start : function() {
        this.canvas.width = 480;
        this.canvas.height = 270;
        this.context = this.canvas.getContext("2d");
        document.body.insertBefore(this.canvas, document.body.childNodes[0]);
        this.frameNo = 0;
        this.interval = setInterval(updateGameArea, 20);
        },
    clear : function() {
        this.context.clearRect(0, 0, this.canvas.width, this.canvas.height);
    }
}
```

Server 1



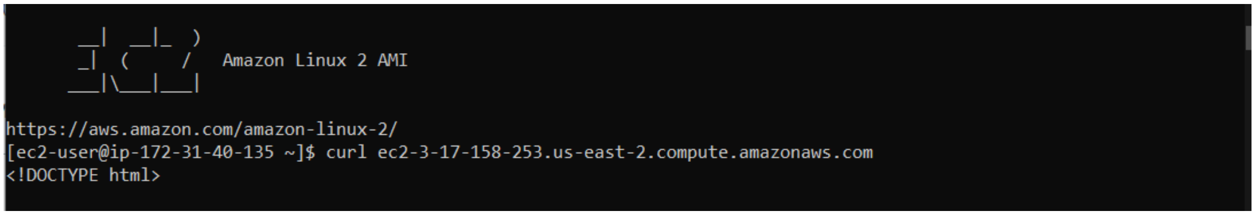
```
      __|  __|_  )
      _|  (     /   Amazon Linux 2 AMI
     ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-40-135 ~]$ curl ec2-3-17-158-253.us-east-2.compute.amazonaws.com
<!DOCTYPE html>
```

Server 2



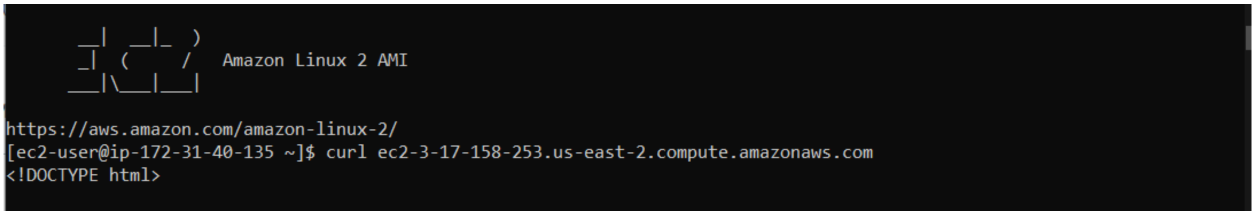
```
      __|  __|_  )
      _|  (     /   Amazon Linux 2 AMI
     ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-40-135 ~]$ curl ec2-3-17-158-253.us-east-2.compute.amazonaws.com
<!DOCTYPE html>
```

Server 3



```
       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-40-135 ~]$ curl ec2-3-17-158-253.us-east-2.compute.amazonaws.com
<!DOCTYPE html>
```

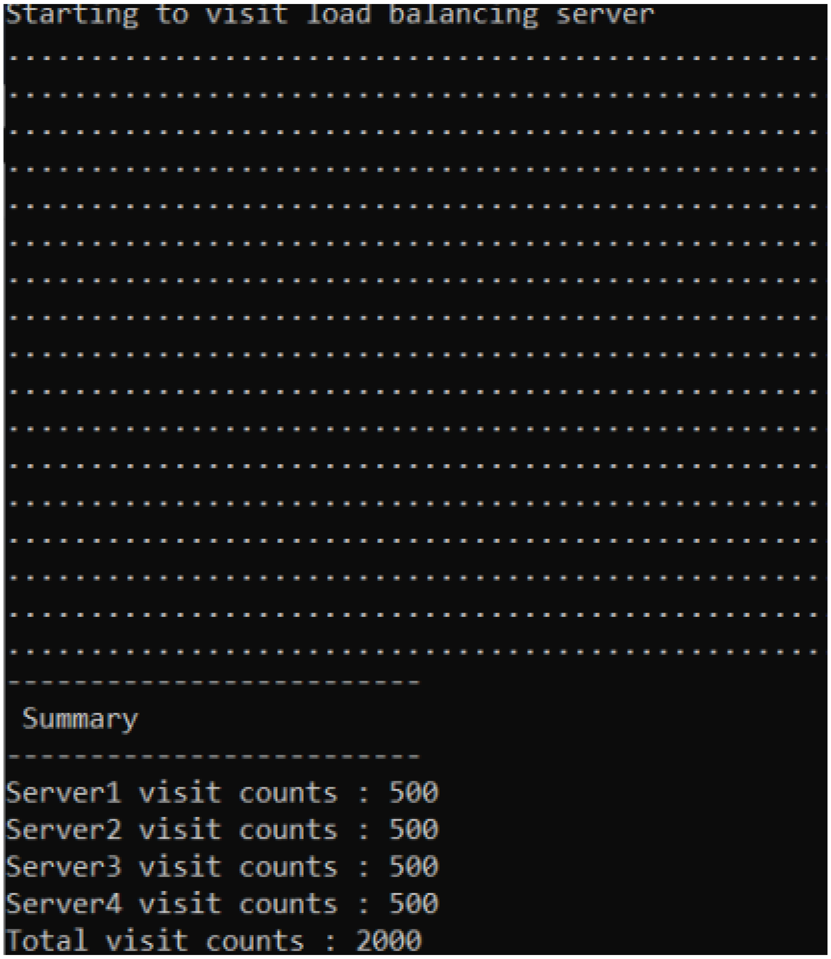
Server 4



```
       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-40-135 ~]$ curl ec2-3-17-158-253.us-east-2.compute.amazonaws.com
<!DOCTYPE html>
```

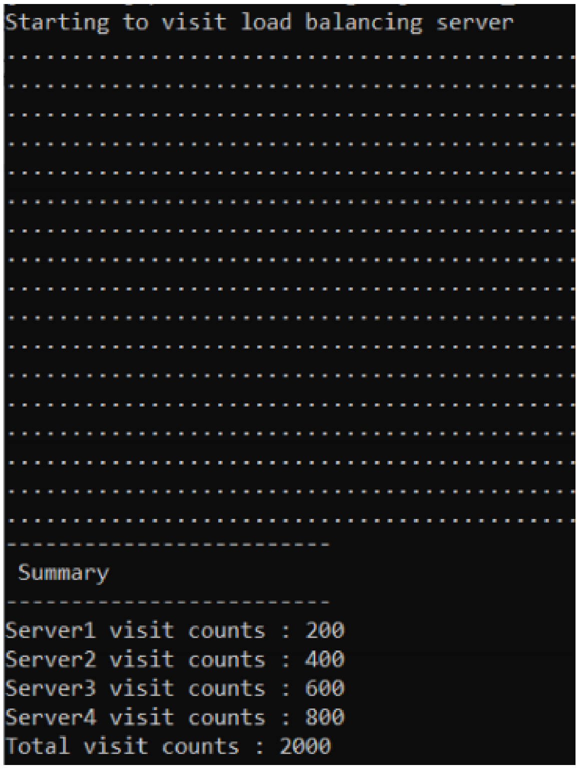
w. Now install ruby on test server and run the visit_server script

For Server 1=1, Server 2=1, Server 3=1, Server 4=1.



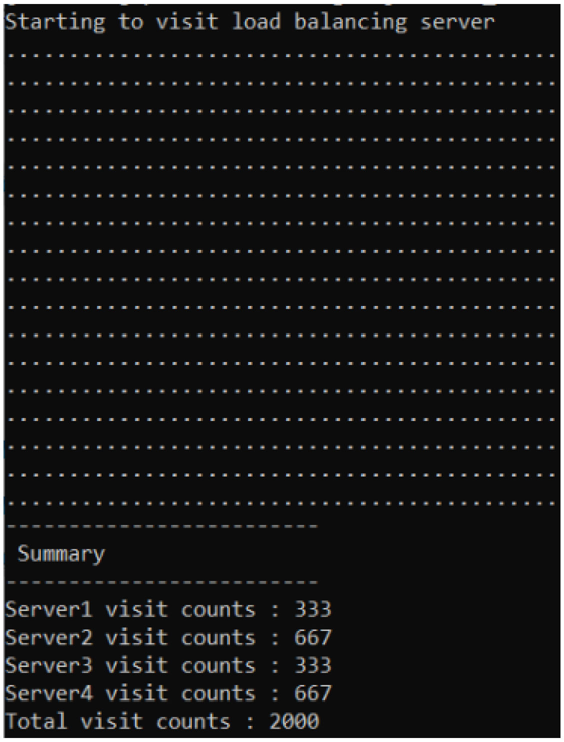
```
Starting to visit load balancing server
..................................................
..................................................
..................................................
..................................................
..................................................
..................................................
..................................................
..................................................
..................................................
..................................................
..................................................
..................................................
..................................................
..................................................
..................................................
..................................................
..................................................
-------------------------
 Summary
-------------------------
Server1 visit counts : 500
Server2 visit counts : 500
Server3 visit counts : 500
Server4 visit counts : 500
Total visit counts : 2000
```

For Server 1=1, Server 2=2, Server 3=3, Server 4=4.



```
Starting to visit load balancing server
............................................
............................................
............................................
............................................
............................................
............................................
............................................
............................................
............................................
............................................
............................................
............................................
............................................
............................................
............................................
............................................
............................................
-------------------------
 Summary
-------------------------
Server1 visit counts : 200
Server2 visit counts : 400
Server3 visit counts : 600
Server4 visit counts : 800
Total visit counts : 2000
```

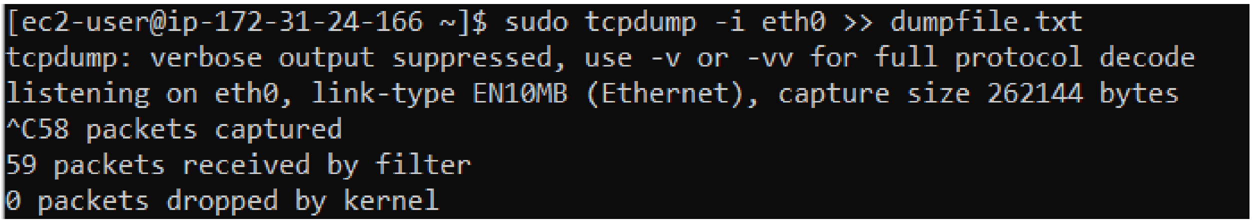
For Server 1=1, Server 2=2, Server 3=1, Server 4=2.



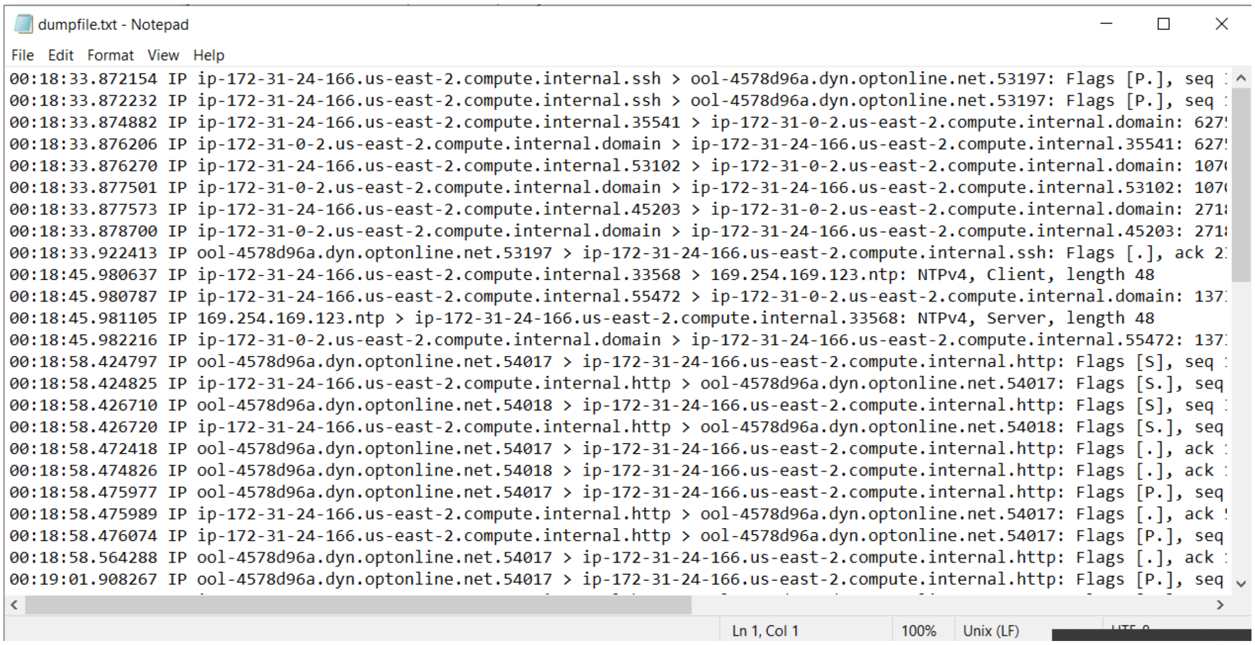
```
Starting to visit load balancing server
............................................
............................................
............................................
............................................
............................................
............................................
............................................
............................................
............................................
............................................
............................................
............................................
............................................
............................................
............................................
............................................
............................................
-------------------------
 Summary
-------------------------
Server1 visit counts : 333
Server2 visit counts : 667
Server3 visit counts : 333
Server4 visit counts : 667
Total visit counts : 2000
```

Q. After having deployed the balancer 1. Use the `tcpdump` command to collect all the packets that had been exchanged. 2. Analyze the packets and report your observations.

Tcmdumpfile in a txt file



```
[ec2-user@ip-172-31-24-166 ~]$ sudo tcpdump -i eth0 >> dumpfile.txt
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
^C58 packets captured
59 packets received by filter
0 packets dropped by kernel
```

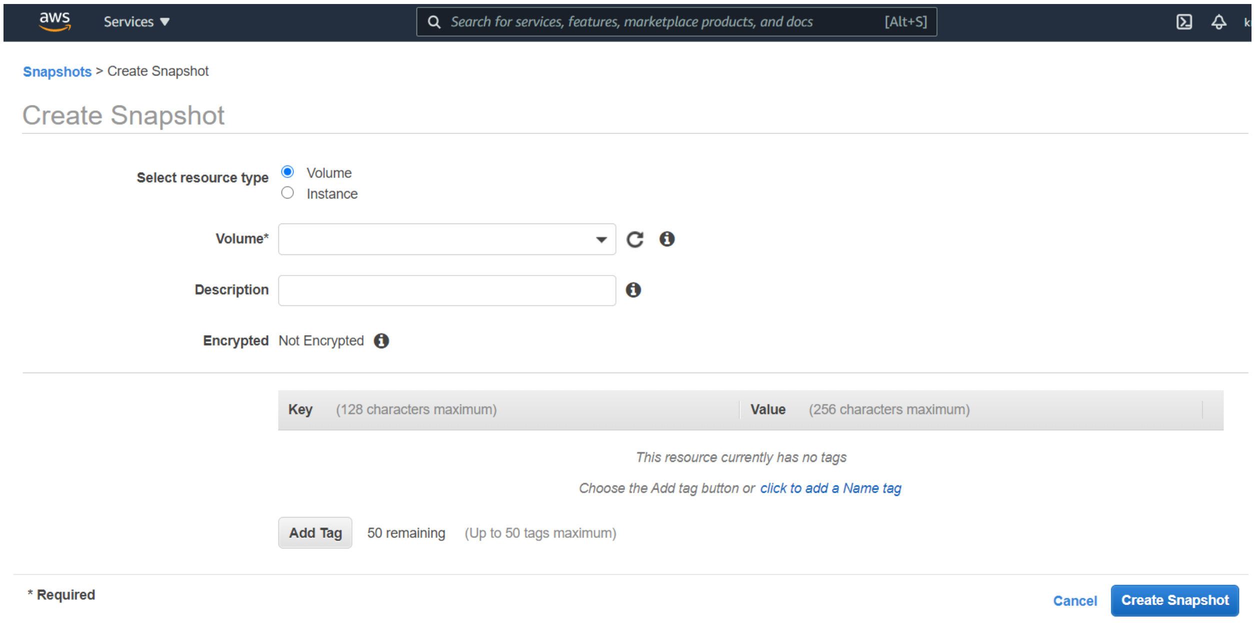


```
00:18:33.872154 IP ip-172-31-24-166.us-east-2.compute.internal.ssh > ool-4578d96a.dyn.optonline.net.53197: Flags [P.], seq
00:18:33.872232 IP ip-172-31-24-166.us-east-2.compute.internal.ssh > ool-4578d96a.dyn.optonline.net.53197: Flags [P.], seq
00:18:33.874882 IP ip-172-31-24-166.us-east-2.compute.internal.35541 > ip-172-31-0-2.us-east-2.compute.internal.domain: 627
00:18:33.876206 IP ip-172-31-0-2.us-east-2.compute.internal.domain > ip-172-31-24-166.us-east-2.compute.internal.35541: 627
00:18:33.876270 IP ip-172-31-24-166.us-east-2.compute.internal.53102 > ip-172-31-0-2.us-east-2.compute.internal.domain: 107
00:18:33.877501 IP ip-172-31-0-2.us-east-2.compute.internal.domain > ip-172-31-24-166.us-east-2.compute.internal.53102: 107
00:18:33.877573 IP ip-172-31-24-166.us-east-2.compute.internal.45203 > ip-172-31-0-2.us-east-2.compute.internal.domain: 271
00:18:33.878700 IP ip-172-31-0-2.us-east-2.compute.internal.domain > ip-172-31-24-166.us-east-2.compute.internal.45203: 271
00:18:33.922413 IP ool-4578d96a.dyn.optonline.net.53197 > ip-172-31-24-166.us-east-2.compute.internal.ssh: Flags [.], ack 2
00:18:45.980637 IP ip-172-31-24-166.us-east-2.compute.internal.33568 > 169.254.169.123.ntp: NTPv4, Client, length 48
00:18:45.980787 IP ip-172-31-24-166.us-east-2.compute.internal.55472 > ip-172-31-0-2.us-east-2.compute.internal.domain: 137
00:18:45.981105 IP 169.254.169.123.ntp > ip-172-31-24-166.us-east-2.compute.internal.33568: NTPv4, Server, length 48
00:18:45.982216 IP ip-172-31-0-2.us-east-2.compute.internal.domain > ip-172-31-24-166.us-east-2.compute.internal.55472: 137
00:18:58.424797 IP ool-4578d96a.dyn.optonline.net.54017 > ip-172-31-24-166.us-east-2.compute.internal.http: Flags [S], seq
00:18:58.424825 IP ip-172-31-24-166.us-east-2.compute.internal.http > ool-4578d96a.dyn.optonline.net.54017: Flags [S.], seq
00:18:58.426710 IP ool-4578d96a.dyn.optonline.net.54018 > ip-172-31-24-166.us-east-2.compute.internal.http: Flags [S], seq
00:18:58.426720 IP ip-172-31-24-166.us-east-2.compute.internal.http > ool-4578d96a.dyn.optonline.net.54018: Flags [S.], seq
00:18:58.472418 IP ool-4578d96a.dyn.optonline.net.54017 > ip-172-31-24-166.us-east-2.compute.internal.http: Flags [.], ack
00:18:58.474826 IP ool-4578d96a.dyn.optonline.net.54018 > ip-172-31-24-166.us-east-2.compute.internal.http: Flags [.], ack
00:18:58.475977 IP ool-4578d96a.dyn.optonline.net.54017 > ip-172-31-24-166.us-east-2.compute.internal.http: Flags [P.], seq
00:18:58.475989 IP ip-172-31-24-166.us-east-2.compute.internal.http > ool-4578d96a.dyn.optonline.net.54017: Flags [.], ack
00:18:58.476074 IP ip-172-31-24-166.us-east-2.compute.internal.http > ool-4578d96a.dyn.optonline.net.54017: Flags [P.], seq
00:18:58.564288 IP ool-4578d96a.dyn.optonline.net.54017 > ip-172-31-24-166.us-east-2.compute.internal.http: Flags [.], ack
00:19:01.908267 IP ool-4578d96a.dyn.optonline.net.54017 > ip-172-31-24-166.us-east-2.compute.internal.http: Flags [P.], seq
```

I can see that there are various interactions from remote aws instances and my computer. Also I can see stevens linux machine interacting to it by analysing the dump file.

Backup and Restore a Snapshot

- Create snapshot
- Select Volume
- Create snapchot


aws
Services ▼
Search for services, features, marketplace products, and docs [Alt+S]

Snapshots > Create Snapshot

Create Snapshot

Select resource type ● Volume ○ Instance

Volume*

Description

Encrypted Not Encrypted

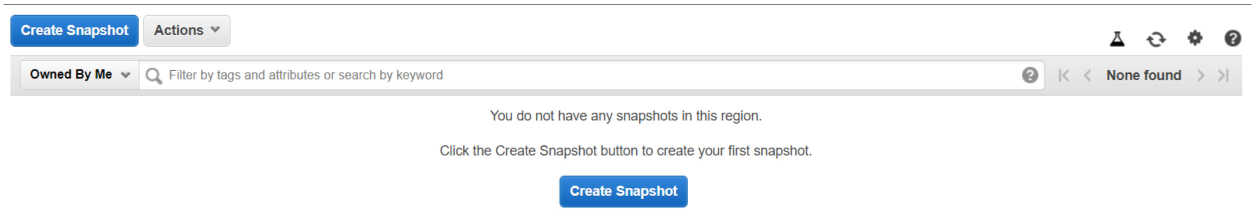
Key (128 characters maximum)	Value (256 characters maximum)

This resource currently has no tags

Choose the Add tag button or click to add a Name tag

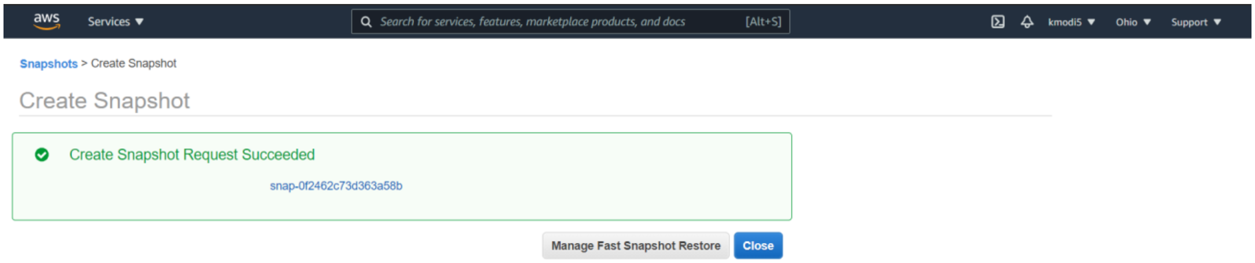
50 remaining (Up to 50 tags maximum)

* Required



You do not have any snapshots in this region.

Click the Create Snapshot button to create your first snapshot.


aws
Services ▼
Search for services, features, marketplace products, and docs [Alt+S]
kmodi5 ▼
Ohio ▼
Support ▼

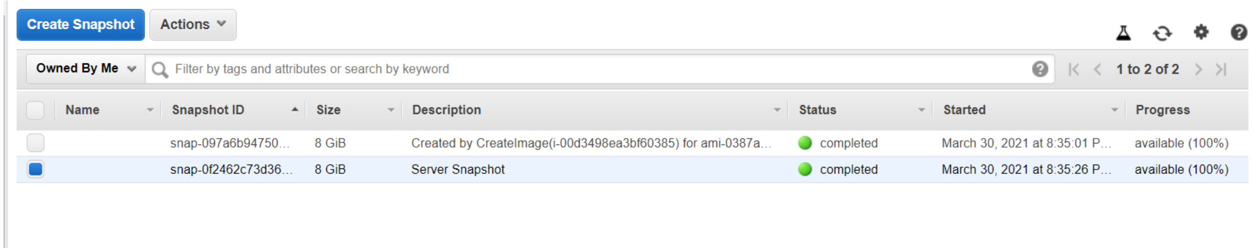
Snapshots > Create Snapshot

Create Snapshot

Create Snapshot Request Succeeded

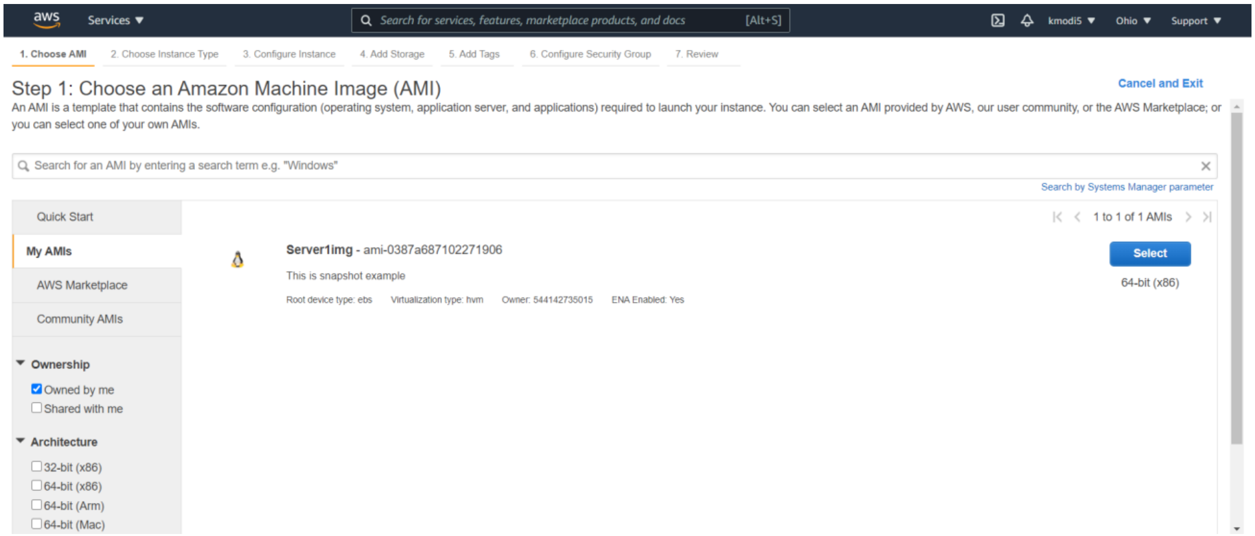
snap-0f2462c73d363a58b

Manage Fast Snapshot Restore
Close



	Name	Snapshot ID	Size	Description	Status	Started	Progress
□		snap-097a6b94750...	8 GiB	Created by CreateImage(i-00d3498ea3bf60385) for ami-0387a...	completed	March 30, 2021 at 8:35:01 P...	available (100%)
■		snap-0f2462c73d36...	8 GiB	Server Snapshot	completed	March 30, 2021 at 8:35:26 P...	available (100%)

Now create new instance by selecting my AMI and selecting your backup



Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

Search for an AMI by entering a search term e.g. "Windows"

Server1img - ami-0387a687102271906

This is snapshot example

Root device type: ebs Virtualization type: hvm Owner: 544142735015 ENA Enabled: Yes

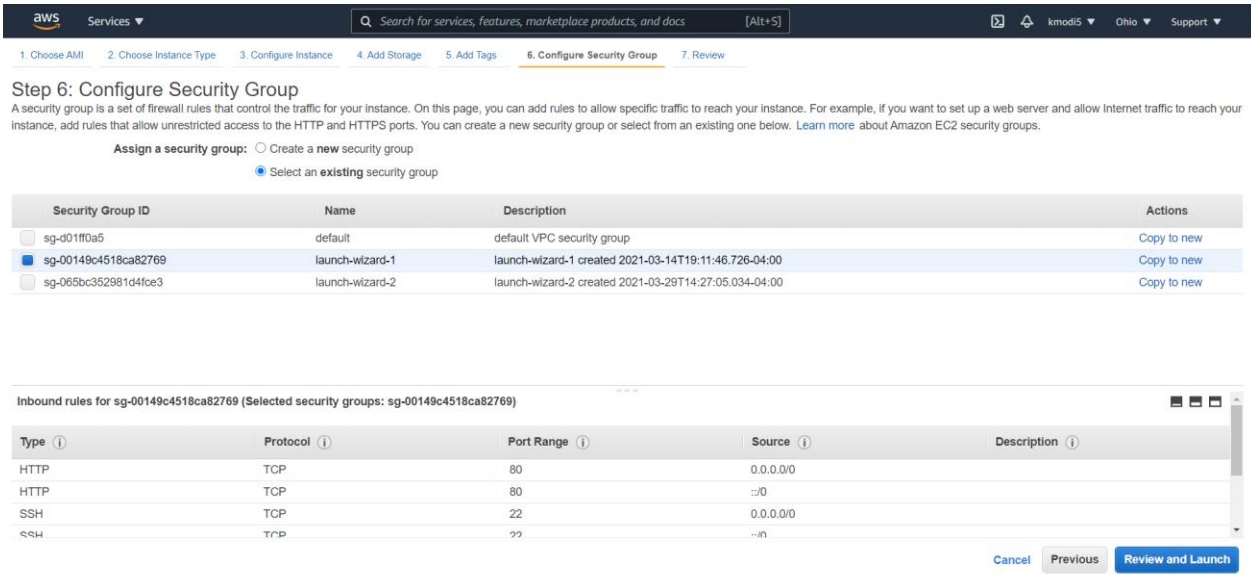
64-bit (x86)

Ownership

☑ Owned by me
☐ Shared with me

Architecture

☐ 32-bit (x86)
☐ 64-bit (x86)
☐ 64-bit (Arm)
☐ 64-bit (Mac)



Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.

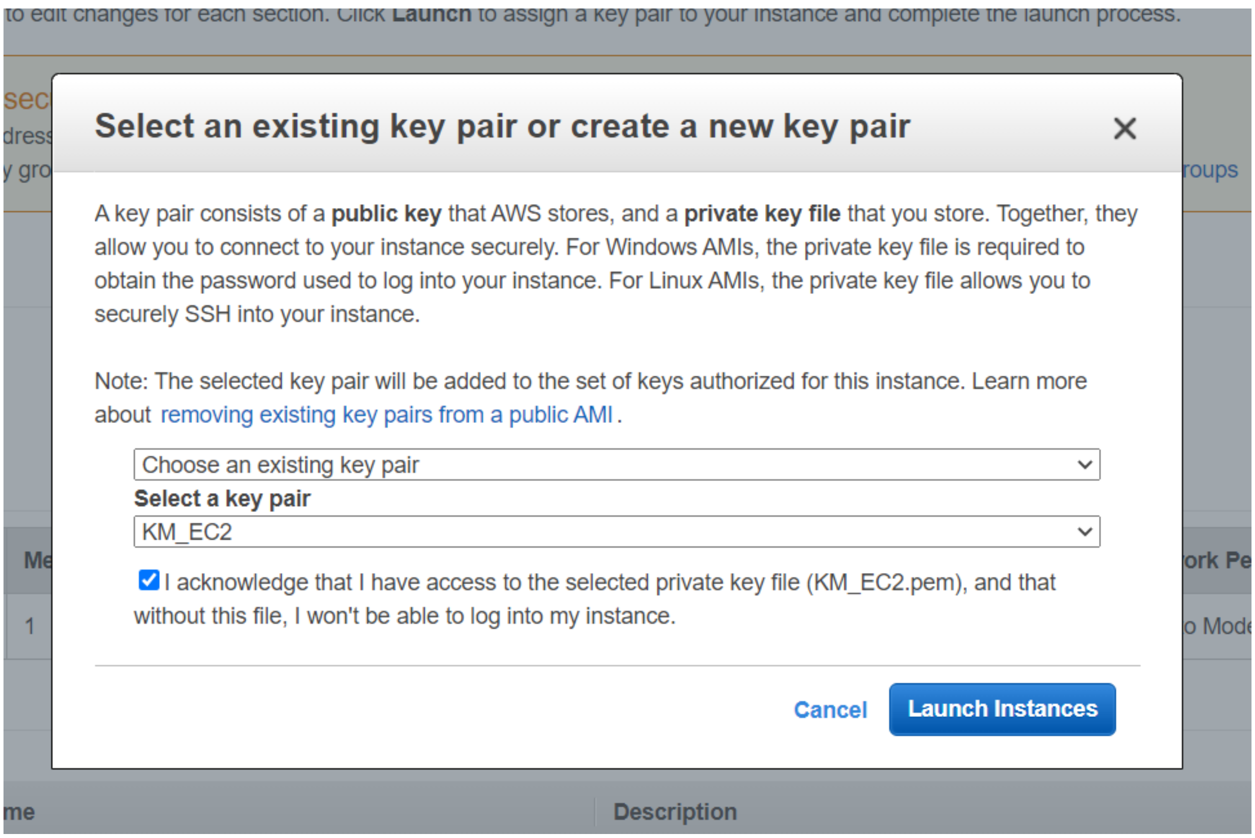
Assign a security group: ○ Create a **new** security group
● Select an **existing** security group

	Security Group ID	Name	Description	Actions
☐	sg-d01ff0a5	default	default VPC security group	Copy to new
☑	sg-00149c4518ca82769	launch-wizard-1	launch-wizard-1 created 2021-03-14T19:11:46.726-04:00	Copy to new
☐	sg-065bc352981d4fce3	launch-wizard-2	launch-wizard-2 created 2021-03-29T14:27:05.034-04:00	Copy to new

Inbound rules for sg-00149c4518ca82769 (Selected security groups: sg-00149c4518ca82769)

Type	Protocol	Port Range	Source	Description
HTTP	TCP	80	0.0.0.0/0	
HTTP	TCP	80	::/0	
SSH	TCP	22	0.0.0.0/0	
SSH	TCP	22	::/0	

Use your existing key or create a new key



to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI .

Choose an existing key pair

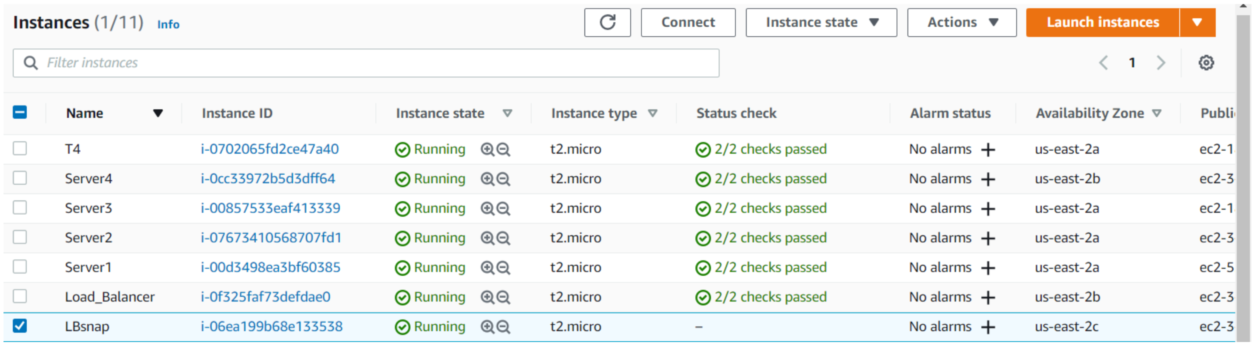
Select a key pair

KM_EC2

☑ I acknowledge that I have access to the selected private key file (KM_EC2.pem), and that without this file, I won't be able to log into my instance.

Now you have your restored image just like your load balancer and you can use it anytime.

This way it allows to restore your work in case things go south!



Instances (1/11) Info

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Publi
□	T4	i-0702065fd2ce47a40	✓ Running	t2.micro	✓ 2/2 checks passed	No alarms +	us-east-2a	ec2-1
□	Server4	i-0cc33972b5d3dff64	✓ Running	t2.micro	✓ 2/2 checks passed	No alarms +	us-east-2b	ec2-3
□	Server3	i-00857533eaf413339	✓ Running	t2.micro	✓ 2/2 checks passed	No alarms +	us-east-2a	ec2-1
□	Server2	i-07673410568707fd1	✓ Running	t2.micro	✓ 2/2 checks passed	No alarms +	us-east-2a	ec2-3
□	Server1	i-00d3498ea3bf60385	✓ Running	t2.micro	✓ 2/2 checks passed	No alarms +	us-east-2a	ec2-5
□	Load_Balancer	i-0f325faf73defdae0	✓ Running	t2.micro	✓ 2/2 checks passed	No alarms +	us-east-2b	ec2-3
☑	LBsnap	i-06ea199b68e133538	✓ Running	t2.micro	–	No alarms +	us-east-2c	ec2-3