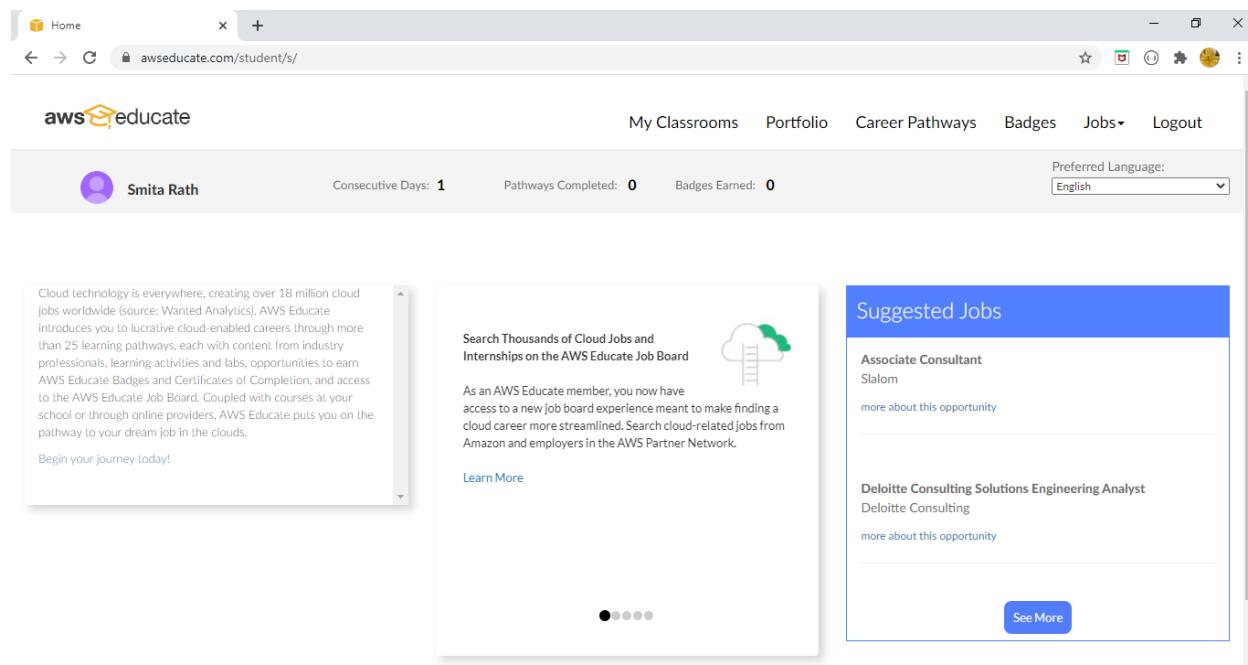
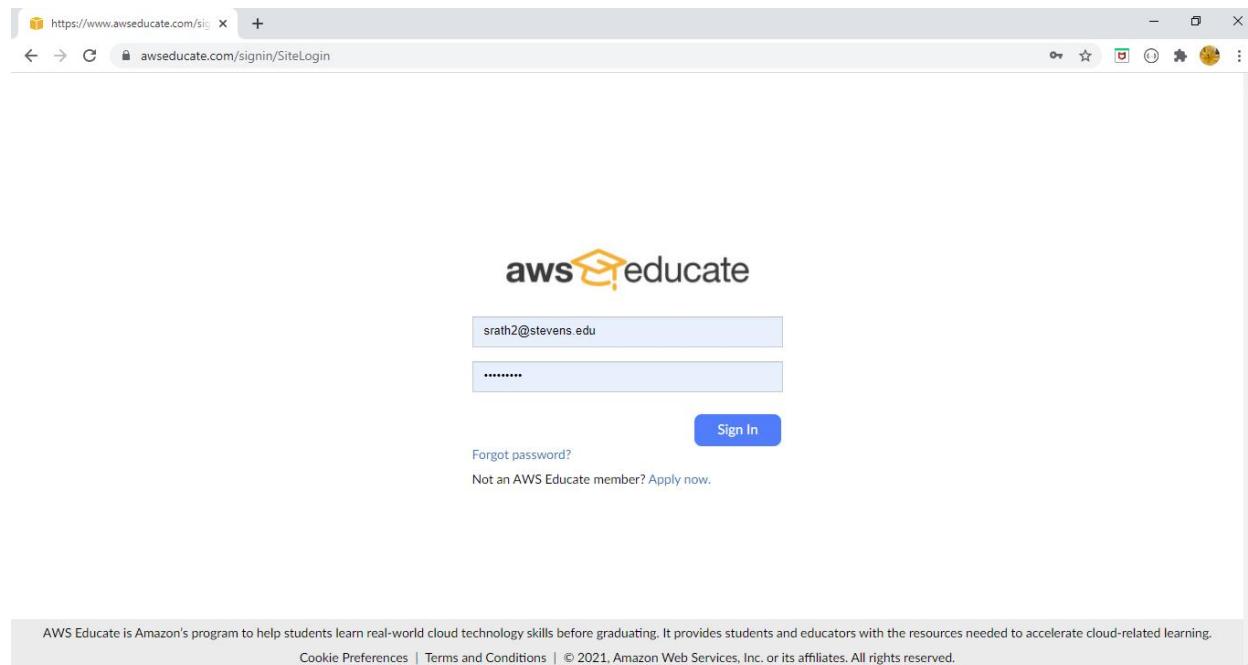


Steps to sign in to aws console



The screenshot shows the AWS Educate 'My Classrooms' page. At the top, there's a header with the AWS Educate logo, user profile (Smita Rath), and navigation links: My Classrooms, Portfolio, Career Pathways, Badges, Jobs, and Logout. A dropdown menu for Preferred Language is set to English. Below the header, a section titled 'My Classrooms' displays a single invitation:

Course Name	Description	Educator	Course End Date	Credit Allocated Per Student	Status
Introduction to Cloud Computing	Full introduction to Cloud technologies and economics with the lab work on AWS.	Igor Faynberg	05/19/2021	\$100	Accepted Go to classroom

At the bottom of the page, there are links for FAQs, AWS Support Forum, Contact Us, Cookie Preferences, and Terms and Conditions, along with a copyright notice for 2021, Amazon Web Services, Inc.

The screenshot shows the Vocabareum Workbench page. The top navigation bar includes 'My Classrooms', 'Workbench', and other tabs. The main content area has two main sections: 'Welcome to your AWS Educate Account' and 'Your AWS Account Status'.

Welcome to your AWS Educate Account

AWS Educate provides you with access to a wide variety of AWS Services for you to get your hands on and build on AWS! To get started, click on the AWS Console button to log in to your AWS console.

Please read the FAQ below to help you get started on your Starter Account.

- What are the list of services supported?
- What regions are supported with Starter Accounts or Classroom Accounts?
- I can't start any resources. What happened?
- Can I create users within my Starter or Classroom Account for others to access?
- Can I create my own IAM policy within Starter Account or Classroom?

Your AWS Account Status

	Active full access ()
	\$99.44 remaining credits (estimated)
	2:60 session time

Buttons for 'Account Details' and 'AWS Console' are visible at the bottom of the status box.

Please use AWS Educate Account responsibly. Remember to shut down your instances when not in use to make the best use of your credits. And, don't forget to logout once you are done with your work!

Create the Amazon EC2 instances

Creating load balancer ec2 instance

The screenshot shows the AWS Management Console homepage. The top navigation bar includes tabs for 'My Classrooms', 'Workbench', and 'AWS Management Console'. The search bar contains the URL 'console.aws.amazon.com/console/home?region=us-east-1#'. The main content area is titled 'AWS Management Console' and features a sidebar titled 'AWS services' with sections for 'Recently visited services' (EC2, Security Hub, Simple Queue Service) and 'All services' (Compute, Quantum Technologies, Security, Identity, & Compliance, Management & Governance, AWS Organizations, CloudWatch, AWS Auto Scaling, CloudFormation). To the right, there's a box for 'Stay connected to your AWS resources on-the-go' about the AWS Console Mobile App, and another box for 'Explore AWS' featuring Amazon SageMaker Resources and a promotion for Graviton2 Based EC2 T4g Instances.

The screenshot shows the 'Dashboard | EC2 Management C' page of the new EC2 console. The top navigation bar includes tabs for 'My Classrooms', 'Workbench', and 'Dashboard | EC2 Management C'. The search bar contains the URL 'console.aws.amazon.com/ec2/v2/home?region=us-east-1#Home'. A blue banner at the top says 'Welcome to the new EC2 console! We're redesigning the EC2 console to make it easier to use and improve performance. We'll release new screens periodically. We encourage you to try them and let us know where we can make improvements. To switch between the old console and the new console, use the New EC2 Experience toggle.' The main content area is titled 'Resources' and displays a table of Amazon EC2 resources in the US East (N. Virginia) Region. The table includes columns for 'Instances (running)', 'Dedicated Hosts', 'Elastic IPs', 'Instances', 'Key pairs', 'Load balancers', 'Placement groups', 'Security groups', 'Snapshots', and 'Volumes'. On the left, a sidebar lists 'New EC2 Experience' (with a 'Learn more' link), 'EC2 Dashboard' (marked as 'New'), and categories 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' (with sub-links for 'AMIs'), and 'Feedback'. On the right, there's a 'Account attributes' section with a 'Supported platforms' list (including 'VPC' and 'Default VPC vpc-2e883e53'), and 'Explore AWS' sections for 'EBS encryption', 'Zones', 'Default credit specification', and 'Console experiments'.

This screenshot shows the AWS EC2 Management Console. The left sidebar is collapsed, showing the 'Instances' section with 'Instances New' selected. The main area displays the 'Instances Info' page with a search bar and filter options. A large message states 'No matching instances found'. Below it, a note says 'Select an instance above'. The top navigation bar includes tabs for 'My Classrooms', 'Workbench', and 'Instances | EC2 Management Con...', along with user information and a support link.

This screenshot shows the 'Launch instance wizard | EC2 Ma...' page at step 1. The title is 'Step 1: Choose an Amazon Machine Image (AMI)'. It explains that an AMI is a template containing software configuration required to launch an instance. A search bar is available to find specific AMIs. The 'Amazon Linux' section is highlighted, showing its details: 'Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-0742b4e673072066f (64-bit x86) / ami-015f1226b535bd02d (64-bit Arm)'. It notes that Amazon Linux 2 is five years old and approaching end of life. Root device type is set to ebs, virtualization type to hvm, and ENA Enabled to Yes. There are two radio buttons for architecture: '64-bit (x86)' (selected) and '64-bit (Arm)'. The 'macOS Big Sur 11.2.3' section is also visible, with a 'Select' button and '64-bit (Mac)' option. The bottom of the page includes standard AWS footer links.

My Classrooms | Workbench | Launch instance wizard | EC2 Manager

console.aws.amazon.com/ec2/v2/home?region=us-east-1#LaunchInstanceWizard

Services ▾ Search for services, features, marketplace products, and documentation [Alt+S]

vocstartsoft/user1314845=srath2@stevens.edu @ 1512-8321-6846 N. Virginia 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
<input type="checkbox"/>	t2	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	t2	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.small	1	2	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.large	2	8	EBS only	-	Low to Moderate	Yes

Cancel Previous Review and Launch Next: Configure Instance Details

Feedback English (US) ▾ © 2008 – 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

Setting security group for load balancer

My Classrooms | Workbench | Launch instance wizard | EC2 Manager

console.aws.amazon.com/ec2/v2/home?region=us-east-1#LaunchInstanceWizard

Services ▾ Search for services, features, marketplace products, and documentation [Alt+S]

vocstartsoft/user1314845=srath2@stevens.edu @ 1512-8321-6846 N. Virginia Support

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

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 name: LB-SG

Description: launch-wizard-2 created 2021-03-29T19:55:37.627-04:00

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	My IP	173.20.217.190/32
HTTPS	TCP	443	Custom	0.0.0.0/0
HTTP	TCP	80	Custom	0.0.0.0/0

Add Rule Cancel Previous Review and Launch

Feedback English (US) ▾ © 2008 – 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

My Classrooms | Workbench | Launch instance wizard | EC2 Management

console.aws.amazon.com/ec2/v2/home?region=us-east-1#LaunchInstanceWizard

Services ▾ Search for services, features, marketplace products, and documentation [Alt+S]

N. Virginia ▾ 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

AMI Details

Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-0742b4e67307206f

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 available.

Root Device Type: ebs Virtualization type: hvm

Edit AMI

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

Edit instance type

Security Groups

Edit security groups

Security group name LB-SG
Description launch-wizard-2 created 2021-03-29T19:55:37Z 627-04:00

Type (i) Protocol (i) Port Range (i) Source (i) Description (i)

Cancel Previous Launch

Feedback English (US) ▾ © 2008 – 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

Creating keypair for load balancer

My Classrooms | Workbench | Launch instance wizard | EC2 Management

console.aws.amazon.com/ec2/v2/home?region=us-east-1#LaunchInstanceWizard

Services ▾ Search for services, features, marketplace products, and documentation [Alt+S]

N. Virginia ▾ 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

AMI Details

Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-0742b4e67307206f

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 available.

Root Device Type: ebs Virtualization type: hvm

Edit AMI

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

Edit instance type

Security Groups

Edit security groups

Security group name LB-SG
Description launch-wizard-2 created 2021-03-29T19:55:37Z 627-04:00

Type (i) Protocol (i) Port Range (i) Source (i) Description (i)

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 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

Feedback English (US) ▾ © 2008 – 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

My Classrooms | Workbench | Launch instance wizard | EC2 Management

We're scanning your download just to be safe. » Scanning...

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: LB_lab2_keypair
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

Feedback English (US) ▾ © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

L8_lab2_keypair.pem Show all X

My Classrooms | Workbench | Launch instance wizard | EC2 Management

Search for services, features, marketplace products, and documentation [Alt+S]

aws Services v ocstartsoft/user1314845=srath2@stevens.edu @ 1512-8321-6846 N. Virginia Support

Launch Status

Your instances are now launching

The following instance launches have been initiated: i-07c920a1be63923db 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

Feedback English (US) ▾ © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

L8_lab2_keypair.pem Show all X

The screenshot shows the AWS EC2 Instances page. The left sidebar includes 'New EC2 Experience' (selected), 'EC2 Dashboard', 'Events', 'Tags', 'Limits', 'Instances' (selected), 'Instance Types', 'Launch Templates', 'Spot Requests', 'Savings Plans', 'Reserved Instances' (New), 'Dedicated Hosts', 'Scheduled Instances', and 'Capacity Reservations'. The main content area displays 'Instances (1) Info' with a table header: Instance ID, Name, Instance state, Instance type, Status check, Alarm status, Availability Zone. A single row is listed: Instance ID i-07c920a1be63923db, Name -, Instance state Running, Instance type t2.micro, Status check -, Alarm status No alarms, Availability Zone us-east-1b. Below the table, a message says 'Select an instance above'.

Load balancer instance created and running.

The screenshot shows the AWS EC2 Instances page. The left sidebar is identical to the previous one. The main content area displays 'Instances (1/1) Info' with the same table structure. The instance row now has a checked checkbox in the first column and the name 'Load Balancer'. The 'Actions' dropdown menu is open, showing options: 'Stop', 'Start', 'Reboot', 'Terminate', 'Launch instances', and 'Edit'. Below the table, a 'Instance summary' section is expanded, showing details: Instance ID i-07c920a1be63923db (Load Balancer), Instance state Running, Instance type t2.micro. Public IPv4 address 34.238.125.182 | open address, Public IPv4 DNS ec2-34-238-125-182.compute-1.amazonaws.com | open address. Private IPv4 addresses 172.31.85.245, Private IPv4 DNS ip-172-31-85-245.ec2.internal. VPC ID vpc-2e883e53.

Other 4 ec2 instances are created using the same steps

Setting security group for server 1

Since server will receive traffic from load balancer so its source is load balancer security group id

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 name:

Description:

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	My IP 173.70.217.190/32	e.g. SSH for Admin Desktop
HTTP	TCP	80	Custom sg-02c76cc3b4deca216	e.g. SSH for Admin Desktop
HTTPS	TCP	443	Custom sg-02c76cc3b4deca216	e.g. SSH for Admin Desktop

Add Rule

Cancel Previous Review and Launch

Step 7: Review Instance Launch

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

Security group name: launch-wizard-2
Description: launch-wizard-2 created 2021-03-29T20:08:00.876-04:00

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	173.70.217.190/32	
HTTP	TCP	80	sg-02c76cc3b4deca216	
HTTPS	TCP	443	sg-02c76cc3b4deca216	

Instance Details

Edit instance details

Cancel Previous Launch

Server 1 created

The screenshot shows the AWS EC2 Instances page. The left sidebar is collapsed. The main area displays a table of instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
Load Balancer	i-07c920a1be63923db	Running	t2.micro	2/2 checks ...	1 alarm...	us-east-1b
Server1	i-02e30c4a30629cc5c	Running	t2.micro	-	No alarms	us-east-1c

Below the table, there is an "Instance summary" section with details for the selected instance (Server1):

Instance ID	Public IPv4 address	Private IPv4 addresses
i-02e30c4a30629cc5c (Server1)	54.159.25.179 open address	172.31.19.6
Instance state	Public IPv4 DNS	Private IPv4 DNS
Running	ec2-54-159-25-179.compute-1.amazonaws.com open address	ip-172-31-19-6.ec2.internal
Instance type	Elastic IP addresses	VPC ID
t2.micro	-	vpc-2e883e53

Server 2 security group

The screenshot shows the AWS Launch Instance Wizard Step 7: Review Instance Launch. The top navigation bar includes tabs for 1. Choose AMI, 2. Choose Instance Type, 3. Configure Instance, 4. Add Storage, 5. Add Tags, 6. Configure Security Group, and 7. Review.

The "Step 7: Review Instance Launch" section shows the following configuration:

t2.micro	-	1	1	EBS only	-	Low to Moderate
----------	---	---	---	----------	---	-----------------

Under "Security Groups", there is a table of rules:

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	173.70.217.190/32	
HTTP	TCP	80	sg-02c76cc3b4deca216	
HTTPS	TCP	443	sg-02c76cc3b4deca216	

Below the security group table, there are sections for "Instance Details", "Storage", and "Tags". At the bottom right are "Cancel", "Previous", and "Launch" buttons.

Server 2 created

The screenshot shows the AWS EC2 Management Console. The left sidebar is collapsed. The main area displays the 'Instances' table with the following data:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
Load Balancer	i-07c920a1be63923db	Running	t2.micro	2/2 checks ...	1 alarm...	us-east-1b
Server1	i-02e30c4a30629cc5c	Running	t2.micro	2/2 checks ...	1/1 h...	us-east-1c
Server2	i-05f384696b14e49cd	Pending	t2.micro	-	No alarms	us-east-1c

Below the table, the 'Instance summary' section provides detailed information for Server2:

Instance ID	Public IPv4 address	Private IPv4 addresses
i-05f384696b14e49cd (Server2)	18.208.189.103 open address	172.31.22.204
Instance state	Public IPv4 DNS	Private IPv4 DNS
Pending	ec2-18-208-189-103.compute-1.amazonaws.com open address	ip-172-31-22-204.ec2.internal
Instance type	Elastic IP addresses	VPC ID
t2.micro	-	vpc-2e883e53

Server 3 security group

The screenshot shows the 'Launch instance wizard' in progress, specifically Step 7: Review Instance Launch. The top navigation bar shows the URL as 'console.aws.amazon.com/ec2/v2/home?region=us-east-1#LaunchInstanceWizard'. The steps are numbered 1 through 7.

Step 7: Review Instance Launch

Instance Type: t2.micro

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: launch-wizard-4

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	173.70.217.190/32	
HTTP	TCP	80	sg-02c76cc3b4deca216	
HTTPS	TCP	443	sg-02c76cc3b4deca216	

Instance Details:

Cancel Previous Launch

Server 3 created

The screenshot shows the AWS EC2 Management Console. The left sidebar is collapsed, and the main area displays the 'Instances' section. A table lists four instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
Load Balancer	i-07c920a1be63923db	Running	t2.micro	2/2 checks ...	1 alarm...	us-east-1b
Server1	i-02e30c4a30629cc5c	Running	t2.micro	2/2 checks ...	1 alarm...	us-east-1c
Server2	i-05f384696b14e49cd	Running	t2.micro	Initializing	1/1 h...	us-east-1c
Server3	i-0f0645e046a02f45e	Running	t2.micro	-	No alarms	us-east-1e

Below the table, an 'Instance summary' panel provides detailed information for Server3:

Instance ID	Public IPv4 address	Private IPv4 addresses
i-0f0645e046a02f45e (Server3)	54.210.79.200 open address	172.31.62.76
Instance state	Public IPv4 DNS	Private IPv4 DNS
Running	ec2-54-210-79-200.compute-1.amazonaws.com open address	ip-172-31-62-76.ec2.internal
Instance type	Elastic IP addresses	VPC ID
t2.micro	-	vpc-2e883e53

Server 4 security group

The screenshot shows the 'Launch instance wizard' in progress, specifically Step 7: Review Instance Launch. The top navigation bar shows the steps: 1. Choose AMI, 2. Choose Instance Type, 3. Configure Instance, 4. Add Storage, 5. Add Tags, 6. Configure Security Group, and 7. Review.

Step 7: Review Instance Launch

t2.micro	-	1	1	EBS only	-	Low to Moderate
----------	---	---	---	----------	---	-----------------

Security Groups

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	173.70.217.190/32	
HTTP	TCP	80	sg-02c76cc3b4deca216	
HTTPS	TCP	443	sg-02c76cc3b4deca216	

Instance Details

Storage

Tags

Buttons at the bottom: Cancel, Previous, Launch.

Server 4 created

The screenshot shows the AWS EC2 Management Console. The left sidebar is collapsed. The main area displays a table of instances:

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
Load Balancer	i-07c920a1be63923db	Running	t2.micro	2/2 checks ...	1 alarm...	us-east-1b	
Server1	i-02e30c4a30629cc5c	Running	t2.micro	2/2 checks ...	1 alarm...	us-east-1c	
Server2	i-05f384696b14e49cd	Running	t2.micro	2/2 checks ...	1 alarm...	us-east-1c	
Server3	i-0f0645e046a02f45e	Running	t2.micro	1/1 h...	No alarms	us-east-1e	
Server4	i-0210a7b472c9051ab	Running	t2.micro	-	+ No alarms	us-east-1b	

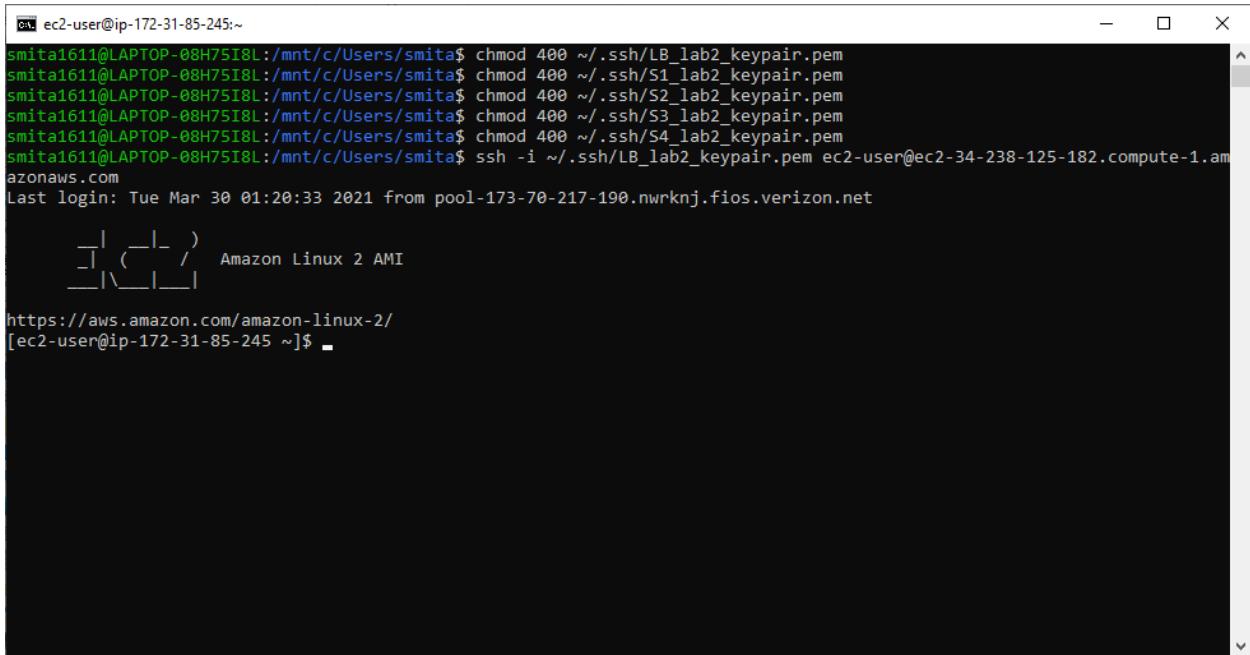
Below the table, there is a detailed view for Server4:

Instance summary	
Instance ID	i-0210a7b472c9051ab (Server4)
Instance state	Running
Instance type	
Public IPv4 address	3.95.6.130 open address
Public IPv4 DNS	ec2-3-95-6-130.compute-1.amazonaws.com open address
Elastic IP addresses	
Private IPv4 addresses	172.31.84.207
Private IPv4 DNS	ip-172.31.84-207.ec2.internal
VPC ID	

Setting permission for the keypair file for all the instances

```
smita1611@LAPTOP-08H75I8L:/mnt/c/Users/smita$ chmod 400 ~/.ssh/LB_lab2_keypair.pem
smita1611@LAPTOP-08H75I8L:/mnt/c/Users/smita$ chmod 400 ~/.ssh/S1_lab2_keypair.pem
smita1611@LAPTOP-08H75I8L:/mnt/c/Users/smita$ chmod 400 ~/.ssh/S2_lab2_keypair.pem
smita1611@LAPTOP-08H75I8L:/mnt/c/Users/smita$ chmod 400 ~/.ssh/S3_lab2_keypair.pem
smita1611@LAPTOP-08H75I8L:/mnt/c/Users/smita$ chmod 400 ~/.ssh/S4_lab2_keypair.pem
```

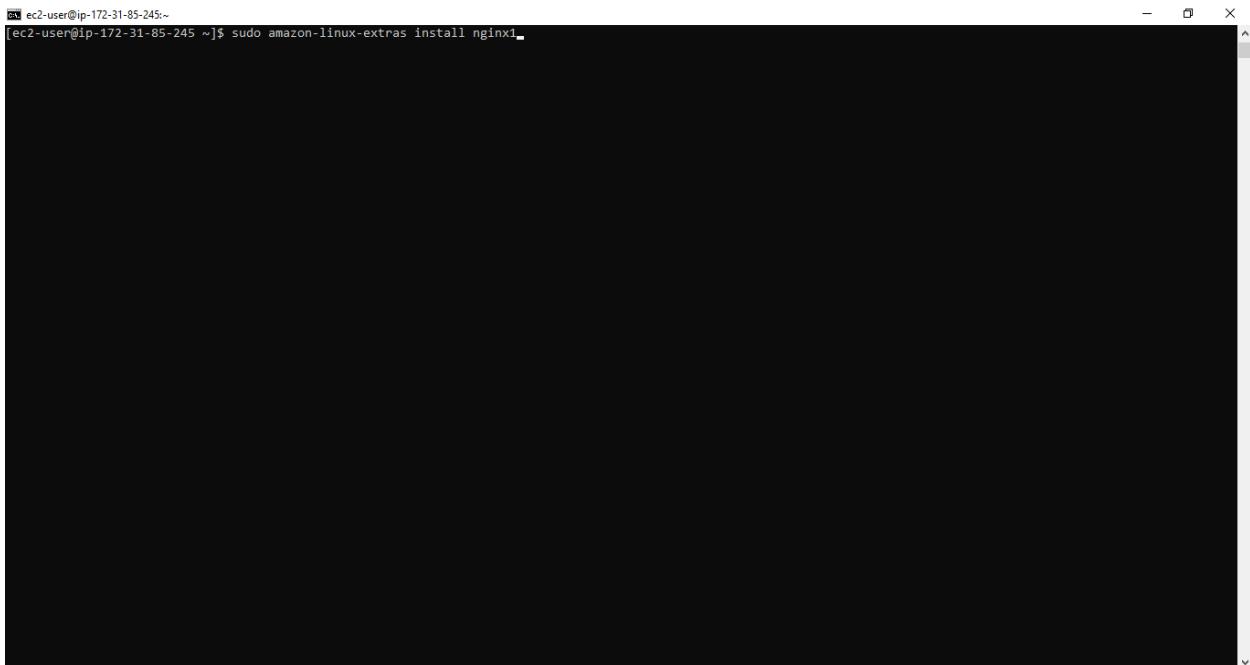
Connected to load balancer instance



```
ec2-user@ip-172-31-85-245:~  
smita1611@LAPTOP-08H75I8L:/mnt/c/Users/smita$ chmod 400 ~/ssh/LB_lab2_keypair.pem  
smita1611@LAPTOP-08H75I8L:/mnt/c/Users/smita$ chmod 400 ~/ssh/S1_lab2_keypair.pem  
smita1611@LAPTOP-08H75I8L:/mnt/c/Users/smita$ chmod 400 ~/ssh/S2_lab2_keypair.pem  
smita1611@LAPTOP-08H75I8L:/mnt/c/Users/smita$ chmod 400 ~/ssh/S3_lab2_keypair.pem  
smita1611@LAPTOP-08H75I8L:/mnt/c/Users/smita$ chmod 400 ~/ssh/S4_lab2_keypair.pem  
smita1611@LAPTOP-08H75I8L:/mnt/c/Users/smita$ ssh -i ~/ssh/LB_lab2_keypair.pem ec2-user@ec2-34-238-125-182.compute-1.amazonaws.com  
Last login: Tue Mar 30 01:20:33 2021 from pool-173-70-217-190.nwrknj.fios.verizon.net  
[ec2-user@ip-172-31-85-245 ~]$
```

The terminal window shows the user has changed file permissions for several SSH key files. Then, they attempt to log in to an EC2 instance using their private key. The system identifies them as 'Amazon Linux 2 AMI'. Finally, they run a command to install nginx.

Installing nginx



```
ec2-user@ip-172-31-85-245:~  
[ec2-user@ip-172-31-85-245 ~]$ sudo amazon-linux-extras install nginx1
```

The terminal window shows the user running a command to install the nginx package via the Amazon Linux Extras repository.

```
ec2-user@ip-172-31-85-245:~
```

```
21 mate-desktop1.x      available  \
22   [ ~1.19.0 =1.20.0 =stable ]
23 GraphicsMagick1.3    available  \
24   [ ~1.3.29 =1.3.32 =1.3.34 =stable ]
25 tomcat8.5            available  \
26   [ ~8.5.31 =8.5.32 =8.5.38 =8.5.40 =8.5.42 =8.5.50
27     =stable ]
28 epel                  available  [ ~7.11 =stable ]
29 testing               available  [ ~1.0 =stable ]
30 ecs                  available  [ =stable ]
31 corretto8             available  \
32   [ ~1.8.0_192 =1.8.0_202 =1.8.0_212 =1.8.0_222 =1.8.0_232
33     =1.8.0_242 =stable ]
34 firecracker           available  [ ~0.11 =stable ]
35 golang1.11            available  \
36   [ ~1.11.3 =1.11.11 =1.11.13 =stable ]
37 squid4               available  [ ~4 =stable ]
38 php7.3                available  \
39   [ ~7.3.2 =7.3.3 =7.3.4 =7.3.6 =7.3.8 =7.3.9 =7.3.10
40     =7.3.11 =7.3.13 =stable ]
41 lustre2.10             available  \
42   [ ~2.10.5 =2.10.8 =stable ]
43 java-openjdk11         available  [ ~11 =stable ]
44 lynis                 available  [ =stable ]
45 kernel-ng              available  [ =stable ]
46 BCC                   available  [ ~0.x =stable ]
47 mono                  available  [ ~5.x =stable ]
48 nginx1.latest          enabled   [ =stable ]
49 ruby2.6                available  [ ~2.6 =stable ]
50 mock                  available  [ =stable ]
51 postgresql11           available  [ ~11 =stable ]
52 php7.4                available  [ =stable ]
53 livepatch              available  [ =stable ]
54 python3.8              available  [ =stable ]
55 haproxy2               available  [ =stable ]
56 collectd               available  [ =stable ]
57 aws-nitro-enclaves-cli available  [ =stable ]
58 R4                     available  [ =stable ]
59 kernel-5.4              available  [ =stable ]
60 selinux-ng              available  [ =stable ]
61 php8.0                available  [ =stable ]
62 tomcat9               available  [ =stable ]
63 unbound1.13             available  [ =stable ]
[ec2-user@ip-172-31-85-245 ~]$
```

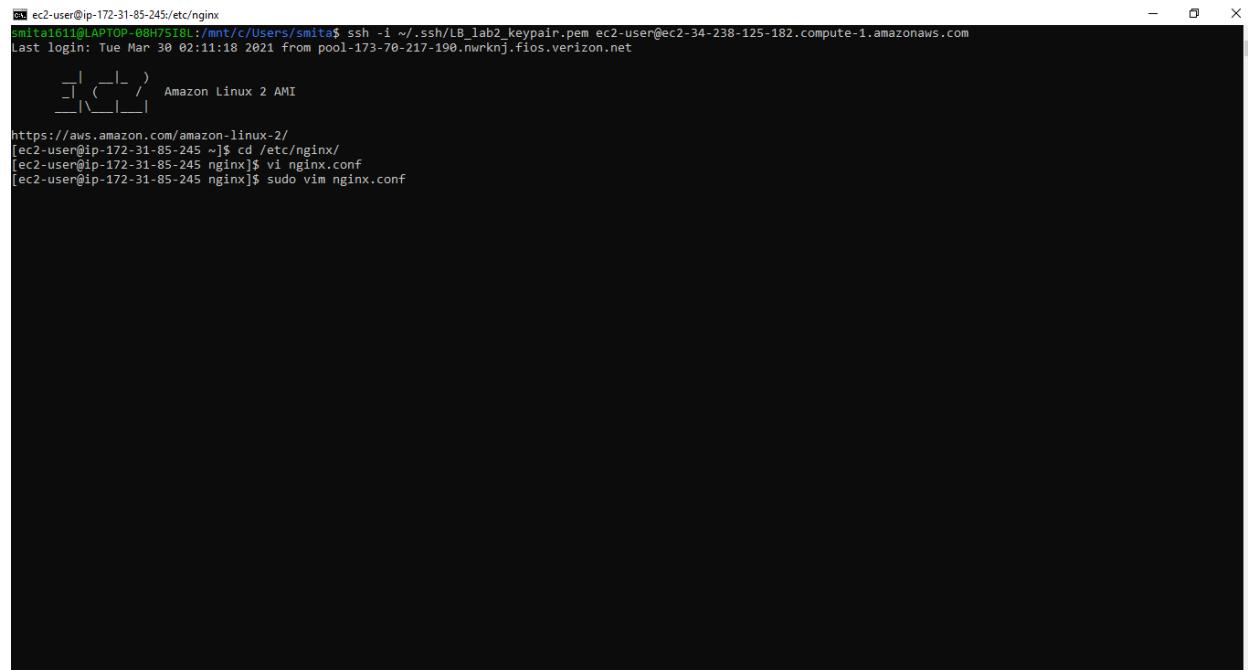
Start nginx

```
ec2-user@ip-172-31-85-245:~
```

```
33 java-openjdk11         available  [ ~11 =stable ]
34 lynis                  available  [ =stable ]
35 kernel-ng               available  [ =stable ]
36 BCC                    available  [ ~0.x =stable ]
37 mono                   available  [ ~5.x =stable ]
38 nginx1.latest           enabled   [ =stable ]
39 ruby2.6                 available  [ ~2.6 =stable ]
40 mock                   available  [ =stable ]
41 postgresql11            available  [ ~11 =stable ]
42 php7.4                 available  [ =stable ]
43 livepatch               available  [ =stable ]
44 python3.8               available  [ =stable ]
45 haproxy2                available  [ =stable ]
46 collectd                available  [ =stable ]
47 aws-nitro-enclaves-cli available  [ =stable ]
48 R4                     available  [ =stable ]
49 kernel-5.4              available  [ =stable ]
50 selinux-ng              available  [ =stable ]
51 php8.0                 available  [ =stable ]
52 tomcat9                available  [ =stable ]
53 unbound1.13             available  [ =stable ]
[ec2-user@ip-172-31-85-245 ~]$ sudo service nginx start
Redirecting to /bin/systemctl start nginx.service
[ec2-user@ip-172-31-85-245 ~]$
```



Editing nginx.conf in load balancer file which is inside /etc/nginx



```
ec2-user@ip-172-31-85-245:/etc/nginx
[smita1611@LAPTOP-08H75IBL:/mnt/c/Users/smita$ ssh -i ~/ssh/LB_lab2.keypair.pem ec2-user@ec2-34-238-125-182.compute-1.amazonaws.com
Last login: Tue Mar 30 02:11:18 2021 from pool-173-78-217-190.nwrknj.fios.verizon.net
[smita1611@LAPTOP-08H75IBL:/mnt/c/Users/smita$ cd /etc/nginx/
[smita1611@LAPTOP-08H75IBL:/etc/nginx$ vi nginx.conf
[smita1611@LAPTOP-08H75IBL:/etc/nginx$ sudo vim nginx.conf
```

Replacing the text provided in the lab instruction

```
events {
    worker_connections 768;
}

http {
    upstream myapp {
        #ip_hash;

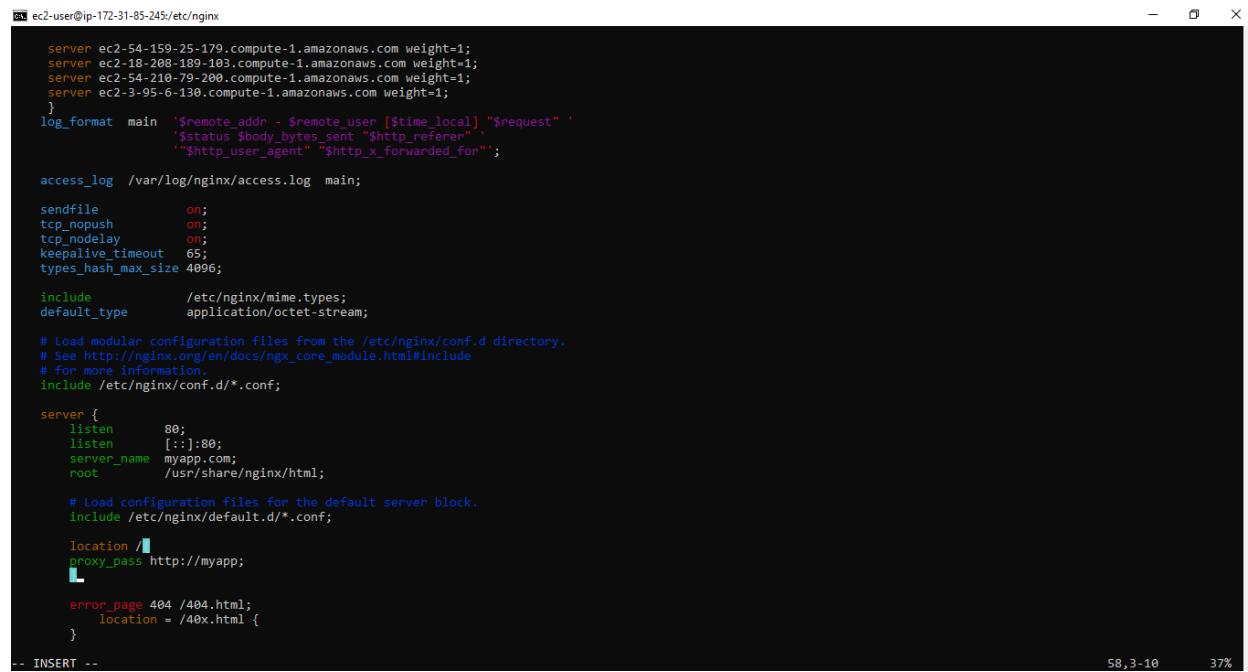
        server [SERVER_PUBLIC_DNS_NAME] weight=1;
        server [SERVER_PUBLIC_DNS_NAME] weight=1;
        server [SERVER_PUBLIC_DNS_NAME] weight=1;
        server [SERVER_PUBLIC_DNS_NAME] weight=1;
    }

    server {
        listen 80;
        server_name myapp.com;
```

```

location / {
    proxy_pass http://myapp;
}

```

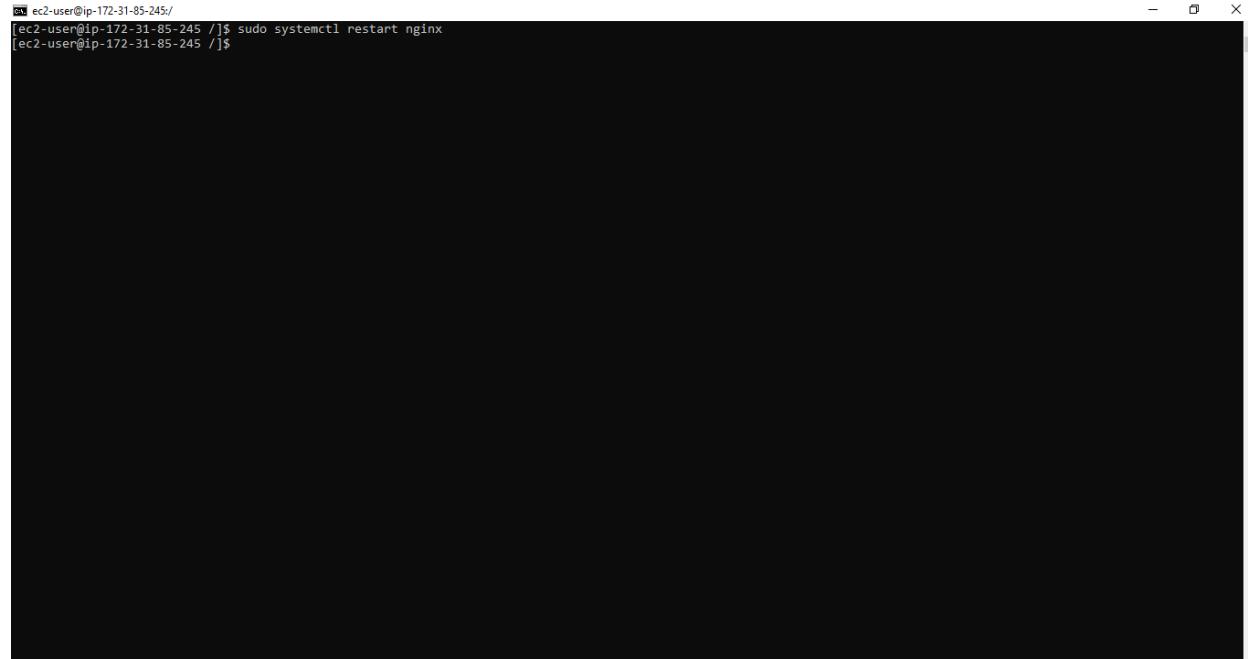


```

ec2-user@ip-172-31-85-245:/etc/nginx
server ec2-54-159-25-179.compute-1.amazonaws.com weight=1;
server ec2-18-208-189-103.compute-1.amazonaws.com weight=1;
server ec2-54-210-79-200.compute-1.amazonaws.com weight=1;
server ec2-3-95-6-130.compute-1.amazonaws.com weight=1;
}
log_format main '$remote_addr - $remote_user [$time_local] "$request" '
                  '$status $body_bytes_sent "'.$http_referer"
                  '"$http_user_agent" "$http_x_forwarded_for"';
access_log /var/log/nginx/access.log main;
sendfile      on;
tcp_nopush    on;
tcp_nodelay   on;
keepalive_timeout 65;
types_hash_max_size 4096;
include        /etc/nginx/mime.types;
default_type   application/octet-stream;
# Load modular configuration files from the /etc/nginx/conf.d directory.
# See http://nginx.org/en/docs/ngx_core_module.html#include
# for more information.
include /etc/nginx/conf.d/*.conf;
server {
    listen      80;
    listen      [::]:80;
    server_name myapp.com;
    root       /usr/share/nginx/html;
    # Load configuration files for the default server block.
    include /etc/nginx/default.d/*.conf;
    location / {
        proxy_pass http://myapp;
    }
    error_page 404 /404.html;
    location = /40x.html {
    }
}
-- INSERT --

```

Reload nginx as conf file is changed so this step is required otherwise those changes will not work.

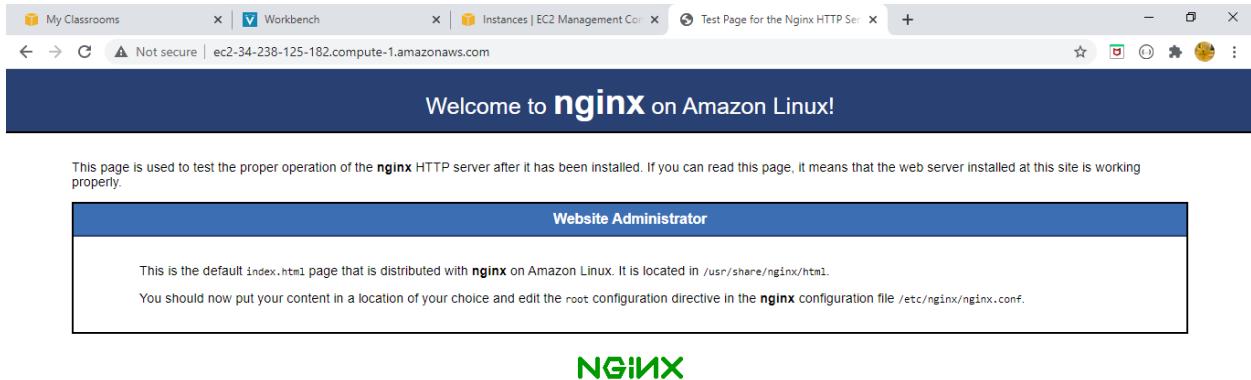


```

ec2-user@ip-172-31-85-245:/
[ec2-user@ip-172-31-85-245 /]$ sudo systemctl restart nginx
[ec2-user@ip-172-31-85-245 /]$ 

```

Load balancer console through public dns name



Installing nginx on others servers

A screenshot of a Windows terminal window titled 'Windows PowerShell'. The command entered is 'ssh -i ~/ssh/51.lab2.keypair.pem ec2-user@ec2-54-159-25-179.compute-1.amazonaws.com'. The response shows the user is connected to an Amazon Linux 2 AMI. The command 'sudo amazon-linux-extras install nginx1' is run, and the output shows the package is already installed. The terminal window is set against a dark background with a light-colored text area. The taskbar at the bottom shows various icons for Microsoft Office applications like Word, Excel, and Powerpoint.

```
ec2-user@ip-172-31-19-6:~
```

```
  -stable ]  
24 epel available [ ~7.11 ~stable ]  
25 testing available [ ~1.0 ~stable ]  
26 ecs available [ ~stable ]  
27 corretto8 available \  
[ ~1.8.0_192 ~1.8.0_202 ~1.8.0_212 ~1.8.0_222 ~1.8.0_232  
~1.8.0_242 ~stable ]  
28 firecracker available [ ~0.11 ~stable ]  
29 golang1.11 available \  
[ ~1.11.3 ~1.11.11 ~1.11.13 ~stable ]  
30 squid4 available [ ~4 ~stable ]  
31 php7.3 available \  
[ ~7.3.2 ~7.3.3 ~7.3.4 ~7.3.6 ~7.3.8 ~7.3.9 ~7.3.10  
~7.3.11 ~7.3.13 ~stable ]  
32 lustre2.10 available \  
[ ~2.10.5 ~2.10.8 ~stable ]  
33 java-openjdk11 available [ ~11 ~stable ]  
34 lynx available [ ~stable ]  
35 kernel-ng available [ ~stable ]  
36 BCC available [ ~0.x ~stable ]  
37 mono available [ ~5.x ~stable ]  
38 nginx1.latest enabled [ ~stable ]  
39 ruby2.6 available [ ~2.6 ~stable ]  
40 mock available [ ~stable ]  
41 postgresql11 available [ ~11 ~stable ]  
42 php7.4 available [ ~stable ]  
43 livepatch available [ ~stable ]  
44 python3.8 available [ ~stable ]  
45 haproxy2 available [ ~stable ]  
46 collectd available [ ~stable ]  
47 aws-nitro-enclaves-cli available [ ~stable ]  
48 R4 available [ ~stable ]  
49 kernel-5.4 available [ ~stable ]  
50 selinux-ng available [ ~stable ]  
51 php8.0 available [ ~stable ]  
52 tomcat9 available [ ~stable ]  
53 unbound1.13 available [ ~stable ]  
[ec2-user@ip-172-31-19-6 ~]$ sudo service nginx start  
Redirecting to /bin/systemctl start nginx.service  
[ec2-user@ip-172-31-19-6 ~]$
```

Editing index.html which is placed inside /usr/share/nginx/html

```
ec2-user@ip-172-31-22-204:/usr/share/nginx/html
```

```
smita1611@LAPTOP-08H751BL:/mnt/c/Users/smita$ ssh -i ~/.ssh/s2_lab2_keypair.pem ec2-user@ec2-18-208-189-103.compute-1.amazonaws.com  
Last login: Tue Mar 30 01:59:11 2021 from pool-173-70-217-190.nwrknj.fios.verizon.net  
_ _| _ _| _ )  
_ _| ( _ _| / _ Amazon Linux 2 AMI  
_ _| \_ _| _ |  
https://aws.amazon.com/amazon-linux-2/  
[ec2-user@ip-172-31-22-204 ~]$ cd /usr/share/nginx/html  
[ec2-user@ip-172-31-22-204 html]$
```

```
ec2-user@ip-172-31-84-207:/usr/share/nginx/html
[mitali011@LAPTOP-08H7518U:~/mint/c/Users/smito]$ ssh -i ~/ssh/S4_lab2_keypair.pem ec2-user@ec2-3-95-6-130.compute-1.amazonaws.com
Last login: Tue Mar 30 02:10:19 2021 from pool-173-70-217-190.nwrknj.fios.verizon.net
[Amazon Linux 2 AMI]
https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-84-207 ~]$ cd /usr/share/nginx/html
[ec2-user@ip-172-31-84-207 html]$ sudo vim index.html
```

Changed the `<h1>` tag to Server 4

```
ec2-user@ip-172-31-84-207:/usr/share/nginx/html
.alert {
    border: 2px solid #000;
}

img {
    border: 2px solid #ffff;
    padding: 2px;
    margin: 2px;
}
a:hover img {
    border: 2px solid #294172;
}
.logos {
    margin: 1em;
    text-align: center;
}
/*]]>*/
```

`</style>`

```
</head>

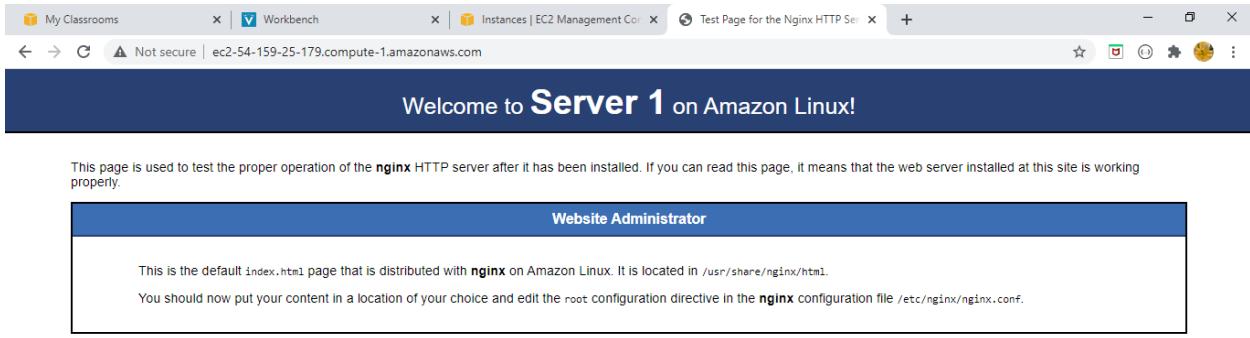
<body>
<h1>Welcome to <strong>Server 4</strong> on Amazon Linux!</h1>

<div class="content">
<p>This page is used to test the proper operation of the
<strong>nginx</strong> 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.</p>
<div class="alert">
<h2>Website Administrator</h2>
<div class="content">
<p>This is the default <tt>index.html</tt> page that
is distributed with <strong>nginx</strong> on
Amazon Linux. It is located in
<tt>/usr/share/nginx/html</tt>.</p>
<p>You should now put your content in a location of
your choice and edit the <tt>root</tt> configuration
directive in the <strong>nginx</strong>
configuration file
<tt>/etc/nginx/nginx.conf</tt>.</p>

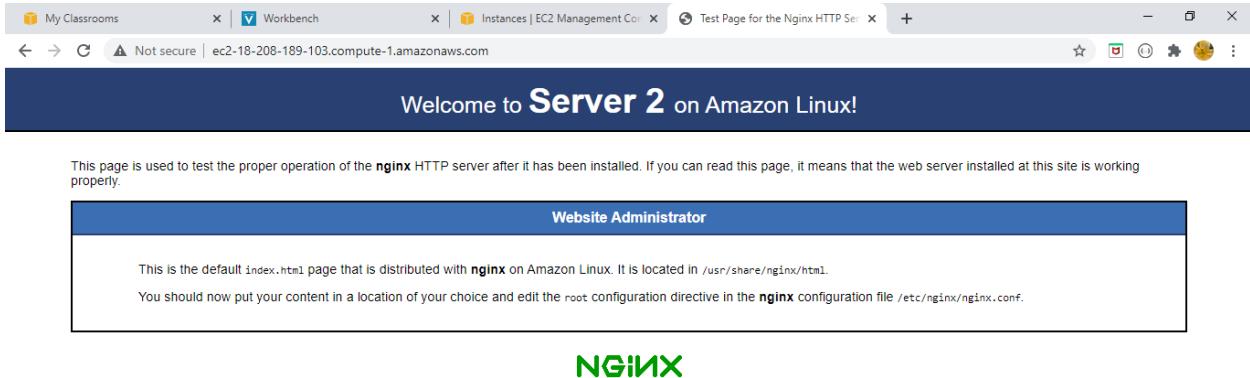
```

"index.html" 111L, 3523C

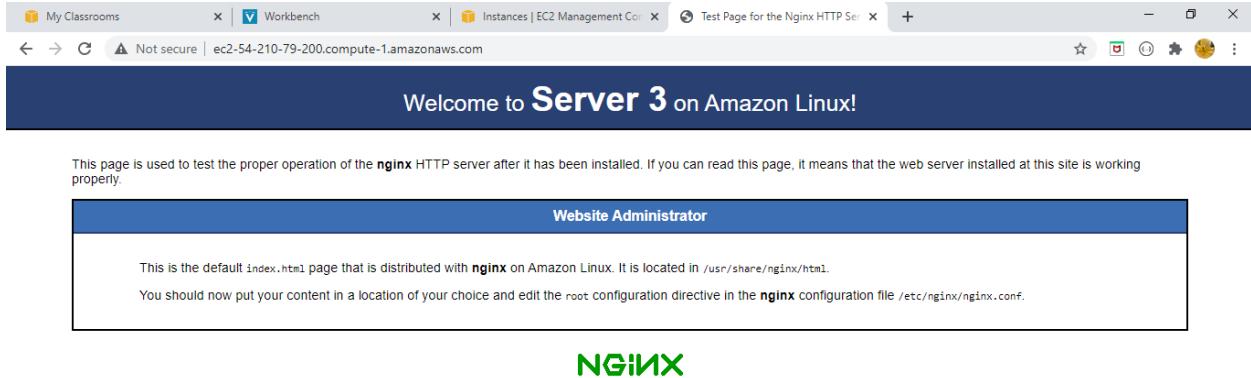
Visited default webpage through public dns name of server1



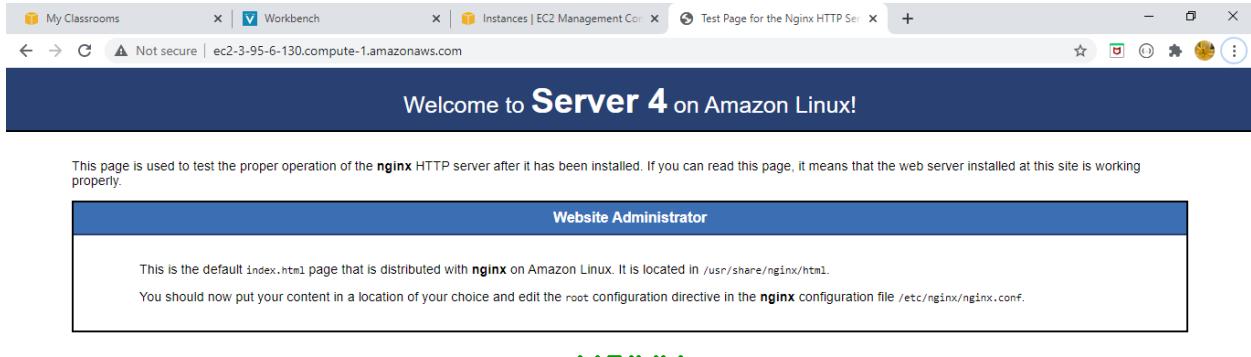
Visited default webpage through public dns name of server2



Visited default webpage through public dns name of server3



Visited default webpage through public dns name of server4



Use the `curl` command in the shell to visit the balancer, which will distribute traffic among the servers.
`$ curl [LOAD_BALANCER_DNS_NAME]`, while running this command it redirected traffic to different servers one by one because `weight=1` for all the servers

```
ec2-user@ip-172-31-62-76:/usr/share/nginx/html
[ec2-user@ip-172-31-62-76 html]$ curl ec2-34-238-125-182.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>Test Page for the Nginx HTTP Server on Amazon Linux</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <style type="text/css">
        /*<![CDATA[*/
        body {
            background-color: #ffff;
            color: #000;
            font-size: 0.9em;
            font-family: sans-serif,helvetica;
            margin: 0;
            padding: 0;
        }
        :link {
            color: #c00;
        }
        :visited {
            color: #c00;
        }
        a:hover {
            color: #f50;
        }
        h1 {
            text-align: center;
            margin: 0;
            padding: 0.6em 2em 0.4em;
            background-color: #294172;
            color: #fff;
            font-weight: normal;
            font-size: 1.75em;
            border-bottom: 2px solid #000;
        }
        h1 strong {
            font-weight: bold;
            font-size: 1.5em;
        }
        h2 {
            text-align: center;
            background-color: #3C6EB4;
            font-size: 1.1em;
        }
    &lt;style type="text/css"&gt;</pre>
```

```
ec2-user@ip-172-31-62-76:/usr/share/nginx/html
<body>
    <h1>Welcome to <strong>Server 4</strong> on Amazon Linux!</h1>
    <div class="content">
        <p>This page is used to test the proper operation of the
        <strong>nginx</strong> 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.</p>
        <div class="alert">
            <h2>Website Administrator</h2>
            <div class="content">
                <p>This is the default <tt>index.html</tt> page that
                is distributed with <strong>nginx</strong> on
                Amazon Linux. It is located in
                <tt>/usr/share/nginx/html</tt>.</p>
                <p>You should now put your content in a location of
                your choice and edit the <tt>root</tt> configuration
                directive in the <strong>nginx</strong>
                configuration file
                <tt>/etc/nginx/nginx.conf</tt>.</p>
            </div>
        </div>
        <div class="logos">
            <a href="http://nginx.net/"></a>
        </div>
    </div>
</body>
</html>
[ec2-user@ip-172-31-62-76 html]$
```

```
ec2-user@ip-172-31-62-76:/usr/share/nginx/html
    margin: 1em;
    text-align: center;
}
/*]]>/*
</style>
</head>

<body>
<h1>Welcome to <strong>Server 1</strong> on Amazon Linux!</h1>

<div class="content">
<p>This page is used to test the proper operation of the
<strong>nginx</strong> 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.</p>
<div class="alert">
<h2>Website Administrator</h2>
<div class="content">
<p>This is the default <tt>index.html</tt> page that
is distributed with <strong>nginx</strong> on
Amazon Linux. It is located in
<tt>/usr/share/nginx/html</tt>.</p>
<p>You should now put your content in a location of
your choice and edit the <tt>root</tt> configuration
directive in the <strong>nginx</strong>
configuration file
<tt>/etc/nginx/nginx.conf</tt>.</p>
</div>
</div>
<div class="logos">
<a href="http://nginx.net/">
</div>
</div>
</body>
</html>
[ec2-user@ip-172-31-62-76 html]$
```

```
ec2-user@ip-172-31-62-76:/usr/share/nginx/html
    margin: 1em;
    text-align: center;
}
/*]]>/*
</style>
</head>

<body>
<h1>Welcome to <strong>Server 2 </strong> on Amazon Linux!</h1>

<div class="content">
<p>This page is used to test the proper operation of the
<strong>nginx</strong> 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.</p>
<div class="alert">
<h2>Website Administrator</h2>
<div class="content">
<p>This is the default <tt>index.html</tt> page that
is distributed with <strong>nginx</strong> on
Amazon Linux. It is located in
<tt>/usr/share/nginx/html</tt>.</p>
<p>You should now put your content in a location of
your choice and edit the <tt>root</tt> configuration
directive in the <strong>nginx</strong>
configuration file
<tt>/etc/nginx/nginx.conf</tt>.</p>
</div>
</div>
<div class="logos">
<a href="http://nginx.net/">
</div>
</div>
</body>
</html>
[ec2-user@ip-172-31-62-76 html]$
```

```
ec2-user@ip-172-31-62-76:/usr/share/nginx/html
    margin: 1em;
    text-align: center;
}
/*]]>/*
</style>
</head>

<body>
<h1>Welcome to <strong>Server 3</strong> on Amazon Linux!</h1>

<div class="content">
<p>This page is used to test the proper operation of the
<strong>nginx</strong> 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.</p>

<div class="alert">
<h2>Website Administrator</h2>
<div class="content">
<p>This is the default <tt>index.html</tt> page that
is distributed with <strong>nginx</strong> on
Amazon Linux. It is located in
<tt>/usr/share/nginx/html</tt>.</p>

<p>You should now put your content in a location of
your choice and edit the <tt>root</tt> configuration
directive in the <strong>nginx</strong>
configuration file
<tt>/etc/nginx/nginx.conf</tt>.</p>
</div>
</div>

<div class="logos">
<a href="http://nginx.net/"></a>
</div>
</div>
</body>
</html>
[ec2-user@ip-172-31-62-76 html]$
```

Collect the information on visits

Created visit_server.rb file to collect the information, since we are going to connect this to load balancer public dns name so we can create this file in any of the other four servers.

```
ec2-user@ip-172-31-62-76:/usr/share/nginx/html
[ec2-user@ip-172-31-62-76 html]$ vim visit_server.rb
```

```
ec2-user@ip-172-31-62-76:/usr/share/nginx/html
$!/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 }
  # HELP argument
  options[:help] = nil
  opts.on('-h', '--help', 'Display usage') { |help| options[:help] = help }
end
option_parser.parse!
# verify arguments
if options[:dns_name] then
  dns_name = options[:dns_name]
else
  puts "Please set a balancer's DNS."
  print_usage
  exit
end
if options[:help] then
  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
server3.visit_count = 0
server4.visit_count = 0
-- INSERT --
```

```
ec2-user@ip-172-31-62-76:/usr/share/nginx/html
end
if options[:help] then
  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
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 -s #{dns_name}`
  o = `curl -s #{dns_name}`
  if o =~ /server1/i
    server1.visit_count += 1
  elsif o =~ /server2/i
    server2.visit_count += 1
  elsif o =~ /server3/i
    server3.visit_count += 1
  elsif o =~ /server4/i
    server4.visit_count += 1
  end
  print "."
end
# redirect output to stdout
#$stdout = orig_stdout
# print visit information
puts
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
-- INSERT --
```

Installation of ruby on ec2 instance

```
gpg2 --recv-keys 409B6B1796C275462A1703113804BB82D39DC0E3  
7D2BAF1CF37B13E2069D6956105BD0E739499BDB
```

```
\curl -sSL https://get.rvm.io | bash -s stable --rails
```

```
ec2-user@ip-172-31-19-6:~  
Downloading https://github.com/rvm/rvm/archive/1.29.12.tar.gz  
Downloading https://github.com/rvm/rvm/releases/download/1.29.12/1.29.12.tar.gz.asc  
GPG: Signature made Fri Jan 15 18:46:22 2021 UTC using RSA key ID 39499BDB  
GPG: Good signature from "Piotr Kuczynski <piotr.kuczynski@mail.com>"  
GPG: WARNING: This key is not certified with a trusted signature!  
GPG:         There is no indication that the signature belongs to the owner.  
Primary key fingerprint: 7D2B AF1C F37B 13E2 0690  6956 105B D0E7 3949 9BDB  
GPG verified '/home/ec2-user/.rvm/archives/rvm-1.29.12.tgz'  
Installing RVM to /home/ec2-user/.rvm/  
  Adding rvm PATH line to /home/ec2-user/.profile /home/ec2-user/.mkshrc /home/ec2-user/.bashrc /home/ec2-user/.zshrc.  
  Adding rvm loading line to /home/ec2-user/.profile /home/ec2-user/.bash_profile /home/ec2-user/.zlogin.  
Installation of RVM in /home/ec2-user/.rvm/ is almost complete:  
  * To start using RVM you need to run `source /home/ec2-user/.rvm/scripts/rvm`  
    in all your open shell windows, in rare cases you need to reopen all shell windows.  
Thanks for installing RVM!  
Please consider donating to our open collective to help us maintain RVM.  
  Donate: https://opencollective.com/rvm/donate  
  
Ruby enVironment Manager 1.29.12 (latest) (c) 2009-2020 Michal Papis, Piotr Kuczynski, Wayne E. Seguin  
  
Searching for binary rubies, this might take some time.  
No binary rubies available for: amazon/2/x86_64/ruby-3.0.0.  
continuing with compilation. Please read 'rvm help mount' to get more information on binary rubies.  
Checking requirements for amazon.  
Installing requirements for amazon.  
installing required packages: patch, autoconf, automake, bison, gcc-c++, libffi-devel, libtool, patch, readline-devel, ruby, sqlite-devel, zlib-devel, glibc-headers, glibc-devel, libyaml-devel, openssl-devel.....  
Requirements installation successful.  
Installing Ruby from source to: /home/ec2-user/.rvm/rubies/ruby-3.0.0, this may take a while depending on your cpu(s)...  
ruby-3.0.0 - #downloading ruby-3.0.0, this may take a while depending on your connection...  
% Total    % Received % Xferd  Average Speed   Time   Time  Current  
          Dload Upload Total Spent   Left  Speed  
100 18.6M  100 18.6M    0  51.0M  0  --::--  --::--  --::-- 51.6M  
ruby-3.0.0 - #extracting ruby-3.0.0 to /home/ec2-user/.rvm/src/ruby-3.0.0.....  
ruby-3.0.0 - #configuring.....  
ruby-3.0.0 - #post-configuration..  
ruby-3.0.0 - #compiling...
```

```
ec2-user@ip-172-31-19-6:~$ 
Installing ri documentation for mini_mime-1.0.3
Parsing documentation for marcel-1.0.0
Installing ri documentation for marcel-1.0.0
Parsing documentation for activemodel-6.1.3.1
Installing ri documentation for activemodel-6.1.3.1
Parsing documentation for activerecord-6.1.3.1
Installing ri documentation for activerecord-6.1.3.1
Parsing documentation for globalid-0.4.2
Installing ri documentation for globalid-0.4.2
Parsing documentation for activejob-6.1.3.1
Installing ri documentation for activejob-6.1.3.1
Parsing documentation for activestorage-6.1.3.1
Installing ri documentation for activestorage-6.1.3.1
Parsing documentation for actiontext-6.1.3.1
Installing ri documentation for actiontext-6.1.3.1
Parsing documentation for mail-2.7.1
Installing ri documentation for mail-2.7.1
Parsing documentation for actionmailer-6.1.3.1
Installing ri documentation for actionmailer-6.1.3.1
Parsing documentation for actionmailbox-6.1.3.1
Installing ri documentation for actionmailbox-6.1.3.1
Parsing documentation for websocket-extensions-0.1.5
Installing ri documentation for websocket-extensions-0.1.5
Parsing documentation for websocket-driver-0.7.3
Installing ri documentation for websocket-driver-0.7.3
Parsing documentation fornio4r-2.5.7
Installing ri documentation fornio4r-2.5.7
Parsing documentation for actionable-6.1.3.1
Installing ri documentation for actionable-6.1.3.1
Parsing documentation for rails-6.1.3.1
Installing ri documentation for rails-6.1.3.1
Done installing documentation for rack, concurrent-ruby, sprockets, zeitwerk, tzinfo, i18n, activesupport, nokogiri, crass, loofah, rails-html-sanitizer, rails-dom-testing, rack-test, erubi, builder, actionview, actionpack, sprockets-rails, thor, method_source, railties, mini_mime, marcel, activemodel, activerecord, globalid, activejob, activestorage, actiontext, mail, actionmailer, actionmailbox, websocket-extensions, websocket-driver, nio4r, actionable, rails after 49 seconds
37 gems installed

* To start using RVM you need to run `source /home/ec2-user/.rvm/scripts/rvm`
in all your open shell windows, in rare cases you need to reopen all shell windows.

* To start using rails you need to run `rails new <project_dir>` .
[ec2-user@ip-172-31-19-6 ~]$ source /home/ec2-user/.rvm/scripts/rvm
[ec2-user@ip-172-31-19-6 ~]$ ruby -v
ruby 3.0.0p0 (2020-12-25 revision 95aff21468) [x86_64-linux]
[ec2-user@ip-172-31-19-6 ~]$
```

After Ruby installation run the visit_server.rb file by providing loadbalance public dns name

```
ec2-user@ip-172-31-19-6:~$ vim visit_server.rb
[ec2-user@ip-172-31-19-6 ~]$ visit_server.rb -d ec2-34-238-125-182.compute-1.amazonaws.com
-bash: visit_server.rb: command not found
[ec2-user@ip-172-31-19-6 ~]$ ruby visit_server.rb -d ec2-34-238-125-182.compute-1.amazonaws.com
Starting to visit load balancing server
.....^[[2;3~
```

First our weights to the server is 1 so visit counts will be same for all servers.

```
ec2-user@ip-172-31-85-245:/etc/nginx
```

```
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 ec2-54-159-25-179.compute-1.amazonaws.com weight=1;
```

```
    server ec2-18-208-189-103.compute-1.amazonaws.com weight=1;
```

```
    server ec2-54-210-79-200.compute-1.amazonaws.com weight=1;
```

```
    server ec2-3-95-6-130.compute-1.amazonaws.com weight=1;
```

```
}
```

```
log_format main '$remote_addr - $remote_user [$time_local] "$request" '
```

```
    '$status $body_bytes_sent "$http_referer"'
```

```
    '"$http_user_agent" "$http_x_forwarded_for"';
```

```
access_log /var/log/nginx/access.log main;
```

```
sendfile on;
```

```
tcp_nopush on;
```

```
tcp_nodelay on;
```

```
keepalive_timeout 65;
```

```
types_hash_max_size 4096;
```

```
include /etc/nginx/mime.types;
```

```
default_type application/octet-stream;
```

```
# Load modular configuration files from the /etc/nginx/conf.d directory.
```

```
# See http://nginx.org/en/docs/ngx_core_module.html#include
```

```
# For more information,
```

```
include /etc/nginx/conf.d/*.conf;
```

```
server {
```

```
    listen 80;
```

```
    listen [::]:80;
```

```
    server_name myapp.com;
```

Changed weight according to given in lab specification

Weight \ Server		Server 1	Server 2	Server 3	Server 4
Scenario					
1		1	1	1	1
2		1	2	3	4
3		1	2	1	2

```
ec2-user@ip-172-31-85-245:/etc/nginx
smita1611@LAPTOP-0BH751BL:/mnt/c/Users/smita$ ssh -i ~/.ssh/LB_lab2_keypair.pem ec2-user@ec2-34-238-125-182.compute-1.amazonaws.com
Last login: Tue Mar 30 02:15:32 2021 from pool-173-70-217-190.nwrknj.fios.verizon.net
[| ( | / Amazon Linux 2 AMI
[| \ | |]

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-85-245 ~]$ cd /etc/nginx/
[ec2-user@ip-172-31-85-245 nginx]$ vim nginx.conf
[ec2-user@ip-172-31-85-245 nginx]$ sudo vim nginx.conf
```

```
ec2-user@ip-172-31-85-245:/etc/nginx
```

```
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 ec2-54-159-25-179.compute-1.amazonaws.com weight=1;
```

```
    server ec2-18-208-189-103.compute-1.amazonaws.com weight=2;
```

```
    server ec2-54-210-79-200.compute-1.amazonaws.com weight=3;
```

```
    server ec2-3-95-6-130.compute-1.amazonaws.com weight=4;
```

```
}
```

```
log_format main '$remote_addr - $remote_user [$time_local] "$request" '
```

```
    '$status $body_bytes_sent "$http_referer"'
```

```
    '"$http_user_agent" "$http_x_forwarded_for"';
```

```
access_log /var/log/nginx/access.log main;
```

```
sendfile      on;
```

```
tcp_nopush   on;
```

```
tcp_nodelay  on;
```

```
keepalive_timeout 65;
```

```
types_hash_max_size 4096;
```

```
include      /etc/nginx/mime.types;
```

```
default_type application/octet-stream;
```

```
# Load modular configuration files from the /etc/nginx/conf.d directory.
```

```
# See http://nginx.org/en/docs/ngx_core_module.html#include
```

```
# For more information,
```

```
include /etc/nginx/conf.d/*.conf;
```

```
server {
```

```
    listen      80;
```

```
    listen      [::]:80;
```

```
    server_name myapp.com;
```

```
-- INSERT --
```

```
ec2-user@ip-172-31-19-6:~  
C:\Users\smita>bash  
smita1611@LAPTOP-08H75IBL:/mnt/c/Users/smita$ ssh -i ~/.ssh/s1_lab2_keypair.pem ec2-user@ec2-54-159-25-179.compute-1.amazonaws.com  
Last login: Tue Mar 30 04:17:36 2021 from pool-173-70-217-190.nwrknj.fios.verizon.net  
  
[|_(-_-)| / Amazon Linux 2 AMI  
  
https://aws.amazon.com/amazon-linux-2/  
[ec2-user@ip-172-31-19-6 ~]$ ruby visit_server.rb -d ec2-34-238-125-182.compute-1.amazonaws.com  
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  
[ec2-user@ip-172-31-19-6 ~]$
```

Weight changed again for third scenario and we can see the visit counts also changed. And after every change restart of nginx is required to reflect the changes of conf file.

```
ec2-user@ip-172-31-85-245:/etc/nginx
# For more information on configuration, see:
#   * Official English Documentation: http://nginx.org/en/docs/
#   * Official Russian Documentation: http://nginx.org/ru/docs/

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 ec2-54-159-25-179.compute-1.amazonaws.com weight=1;
        server ec2-18-208-189-103.compute-1.amazonaws.com weight=2;
        server ec2-54-210-79-200.compute-1.amazonaws.com weight=1;
        server ec2-3-95-6-130.compute-1.amazonaws.com weight=2;
    }
    log_format main "$remote_addr - $remote_user [$time_local] \"$request\" "
                  "$status $body_bytes_sent \"$http_referer\""
                  "\"$http_user_agent\" \"$http_x_forwarded_for\"";

    access_log /var/log/nginx/access.log main;

    sendfile      on;
    tcp_nopush    on;
    tcp_nodelay   on;
    keepalive_timeout 65;
    types_hash_max_size 4096;

    include       /etc/nginx/mime.types;
    default_type  application/octet-stream;

    # Load modular configuration files from the /etc/nginx/conf.d directory.
    # See http://nginx.org/en/docs/ngx_core_module.html#include
"nginx.conf" 97L, 2698C
```

```
ec2-user@ip-172-31-85-245:/etc/nginx
smita@LAPTOP-08H751BL:/mnt/c/Users/smita$ ssh -i ~/.ssh/LB_lab2.keypair.pem ec2-user@ec2-34-238-125-182.compute-1.amazonaws.com
Last login: Tue Mar 30 04:19:36 2021 from pool-173-70-217-190.nwrknj.fios.verizon.net
[Amazon Linux 2 AMI]
https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-85-245 ~]$ cd /etc/nginx/
[ec2-user@ip-172-31-85-245 nginx]$ sudo vim nginx.conf
[ec2-user@ip-172-31-85-245 nginx]$ sudo vim nginx.conf
[ec2-user@ip-172-31-85-245 nginx]$ sudo systemctl restart nginx
[ec2-user@ip-172-31-85-245 nginx]$
```

```
ec2-user@ip-172-31-19-6~  
[unit111@LAPTOP-08H751BL:/mnt/c/Users/smita]$ ssh -i ~/ssh/S1_lab2_keypair.pem ec2-user@ec2-54-25-179.compute-1.amazonaws.com  
Last login: Tue Mar 30 04:20:50 2021 from pool-173-70-217-190.nwrknj.fios.verizon.net  
_ _ | _ / ) Amazon Linux 2 AMI  
_ _ \_\_|_ |  
https://aws.amazon.com/amazon-linux-2/  
[ec2-user@ip-172-31-19-6 ~]$ ruby visit_server.rb -d ec2-34-238-125-182.compute-1.amazonaws.com  
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  
[ec2-user@ip-172-31-19-6 ~]$
```

Use the `tcpdump` command to collect all the packets that had been exchanged.
Install tcpdump to collect the packet information

```
ec2-user@ip-172-31-19-6~  
[ec2-user@ip-172-31-19-6 ~]$ sudo yum install tcpdump -y  
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd  
amzn2-core  
Package 14:tcpdump-4.9.2-4.amzn2.1.x86_64 already installed and latest version  
Nothing to do  
[ec2-user@ip-172-31-19-6 ~]$
```

To Collect packets from all the interfaces through tcpdump use –interface any

```
[ec2-user@ip-172-31-19-6:~]$ sudo tcpdump --interface any -
```

It will start collecting packets.

```

ec2-user@ip-172-31-19-6:~#
04:38:09.494847 IP ip-172-31-19-6.ec2.internal.ssh > pool-173-70-217-198.nwrknj.fios.verizon.net.49758: Flags [P.], seq 15916008:15916008, ack 8425, win 279, length 356
04:38:09.494930 IP ip-172-31-19-6.ec2.internal.ssh > pool-173-70-217-198.nwrknj.fios.verizon.net.49758: Flags [P.], seq 15916008:15916212, ack 8425, win 279, length 204
04:38:09.495000 IP ip-172-31-19-6.ec2.internal.ssh > pool-173-70-217-198.nwrknj.fios.verizon.net.49758: Flags [P.], seq 15916212:15916416, ack 8425, win 279, length 204
04:38:09.495097 IP ip-172-31-19-6.ec2.internal.ssh > pool-173-70-217-198.nwrknj.fios.verizon.net.49758: Flags [P.], seq 15916416:15916620, ack 8425, win 279, length 204
04:38:09.495183 IP ip-172-31-19-6.ec2.internal.ssh > pool-173-70-217-198.nwrknj.fios.verizon.net.49758: Flags [P.], seq 15916620:15916824, ack 8425, win 279, length 204
04:38:09.495281 IP ip-172-31-19-6.ec2.internal.ssh > pool-173-70-217-198.nwrknj.fios.verizon.net.49758: Flags [P.], seq 15916824:15917028, ack 8425, win 279, length 204
04:38:09.495396 IP ip-172-31-19-6.ec2.internal.ssh > pool-173-70-217-198.nwrknj.fios.verizon.net.49758: Flags [P.], seq 15917028:15917232, ack 8425, win 279, length 204
04:38:09.495411 IP pool-173-70-217-198.nwrknj.fios.verizon.net.49758 > ip-172-31-19-6.ec2.internal.ssh: Flags [.], ack 15887572, win 124, length 0
04:38:09.495464 IP ip-172-31-19-6.ec2.internal.ssh > pool-173-70-217-198.nwrknj.fios.verizon.net.49758: Flags [P.], seq 15917232:15917588, ack 8425, win 279, length 356
04:38:09.495575 IP ip-172-31-19-6.ec2.internal.ssh > pool-173-70-217-198.nwrknj.fios.verizon.net.49758: Flags [P.], seq 15917588:15917792, ack 8425, win 279, length 204
04:38:09.495678 IP ip-172-31-19-6.ec2.internal.ssh > pool-173-70-217-198.nwrknj.fios.verizon.net.49758: Flags [P.], seq 15917792:15917996, ack 8425, win 279, length 204
04:38:09.495760 IP ip-172-31-19-6.ec2.internal.ssh > pool-173-70-217-198.nwrknj.fios.verizon.net.49758: Flags [P.], seq 15917996:15918200, ack 8425, win 279, length 204
04:38:09.495839 IP ip-172-31-19-6.ec2.internal.ssh > pool-173-70-217-198.nwrknj.fios.verizon.net.49758: Flags [P.], seq 15918200:15918404, ack 8425, win 279, length 204
04:38:09.495916 IP ip-172-31-19-6.ec2.internal.ssh > pool-173-70-217-198.nwrknj.fios.verizon.net.49758: Flags [P.], seq 15918404:15918608, ack 8425, win 279, length 204
04:38:09.495992 IP pool-173-70-217-198.nwrknj.fios.verizon.net.49758 > ip-172-31-19-6.ec2.internal.ssh: Flags [.], ack 15889356, win 117, length 0
04:38:09.496031 IP ip-172-31-19-6.ec2.internal.ssh > pool-173-70-217-198.nwrknj.fios.verizon.net.49758: Flags [P.], seq 15918608:15918812, ack 8425, win 279, length 204
04:38:09.496094 IP ip-172-31-19-6.ec2.internal.ssh > pool-173-70-217-198.nwrknj.fios.verizon.net.49758: Flags [P.], seq 15918812:15919168, ack 8425, win 279, length 356
04:38:09.496547 IP pool-173-70-217-198.nwrknj.fios.verizon.net.49758 > ip-172-31-19-6.ec2.internal.ssh: Flags [.], ack 15890936, win 111, length 0
04:38:09.496547 IP pool-173-70-217-198.nwrknj.fios.verizon.net.49758 > ip-172-31-19-6.ec2.internal.ssh: Flags [.], ack 15892556, win 105, length 0
04:38:09.497195 IP pool-173-70-217-198.nwrknj.fios.verizon.net.49758 > ip-172-31-19-6.ec2.internal.ssh: Flags [.], ack 15894076, win 99, length 0
04:38:09.497915 IP pool-173-70-217-198.nwrknj.fios.verizon.net.49758 > ip-172-31-19-6.ec2.internal.ssh: Flags [.], ack 15895452, win 93, length 0
04:38:09.498561 IP pool-173-70-217-198.nwrknj.fios.verizon.net.49758 > ip-172-31-19-6.ec2.internal.ssh: Flags [.], ack 15897032, win 87, length 0
04:38:09.499172 IP pool-173-70-217-198.nwrknj.fios.verizon.net.49758 > ip-172-31-19-6.ec2.internal.ssh: Flags [.], ack 15898256, win 82, length 0
04:38:09.499675 IP pool-173-70-217-198.nwrknj.fios.verizon.net.49758 > ip-172-31-19-6.ec2.internal.ssh: Flags [.], ack 15900336, win 74, length 0
04:38:09.411385 IP pool-173-70-217-198.nwrknj.fios.verizon.net.49758 > ip-172-31-19-6.ec2.internal.ssh: Flags [.], ack 15900336, win 74, length 0
04:38:09.411812 IP pool-173-70-217-198.nwrknj.fios.verizon.net.49758 > ip-172-31-19-6.ec2.internal.ssh: Flags [.], ack 15901508, win 70, length 0
04:38:09.411863 IP pool-173-70-217-198.nwrknj.fios.verizon.net.49758 > ip-172-31-19-6.ec2.internal.ssh: Flags [.], ack 15902732, win 65, length 0
04:38:09.412549 IP pool-173-70-217-198.nwrknj.fios.verizon.net.49758 > ip-172-31-19-6.ec2.internal.ssh: Flags [.], ack 15903956, win 68, length 0
04:38:09.413603 IP pool-173-70-217-198.nwrknj.fios.verizon.net.49758 > ip-172-31-19-6.ec2.internal.ssh: Flags [.], ack 15903956, win 76, length 0
04:38:09.413609 IP ip-172-31-19-6.ec2.internal.ssh > pool-173-70-217-198.nwrknj.fios.verizon.net.49758: Flags [P.], seq 15919168:15921208, ack 8425, win 279, length 204
a
81772 packets captured
81775 packets received by filter
3 packets dropped by kernel
[ec2-user@ip-172-31-19-6 ~]#

```

My system's domain name - pool-173-70-217-190.nwrknj.fios.verizon.net

What I have observed from the packets from my machine's domain, they are going to ec2 server through seq and then sending ack.

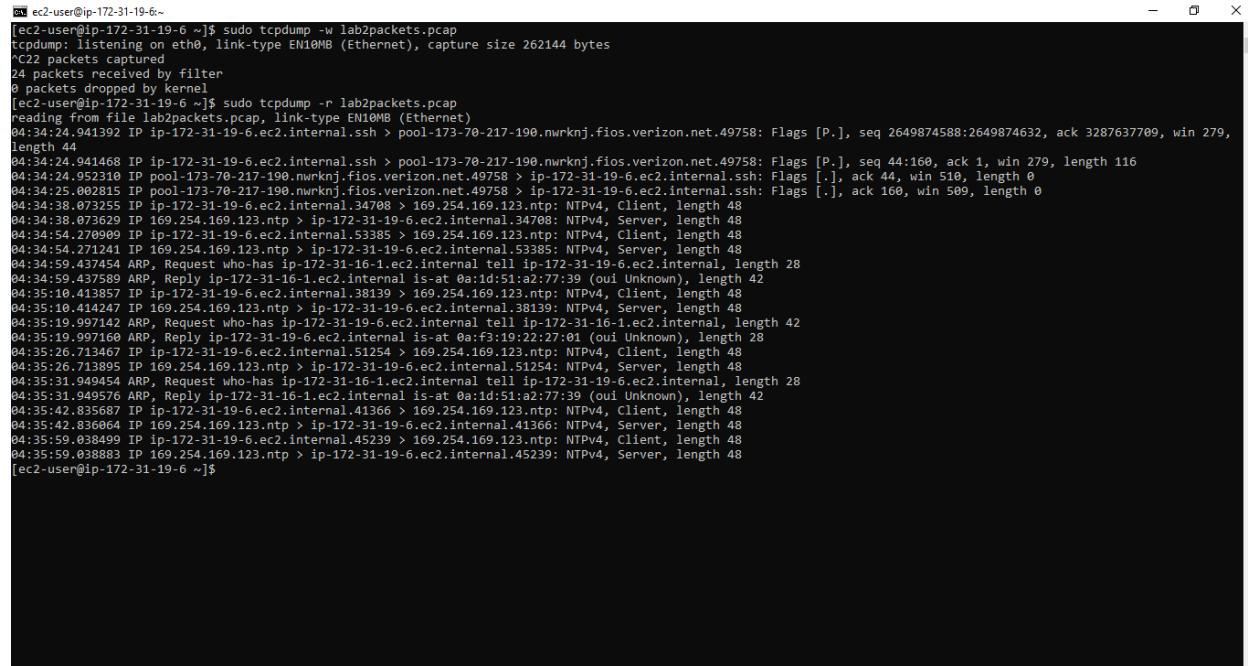
But if we want to capture packets in a file we should use sudo tcpdump -w filename.pcap

```

ec2-user@ip-172-31-19-6:~#
[ec2-user@ip-172-31-19-6 ~]$ sudo tcpdump -w lab2packets.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
^C24 packets captured
0 packets received by filter
0 packets dropped by kernel
[ec2-user@ip-172-31-19-6 ~]$

```

To read the packets from the file use command sudo tcpdump -w lab2packets.pcap



```
[ec2-user@ip-172-31-19-6 ~]$ sudo tcpdump -w lab2packets.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
^C22 packets received by filter
0 packets dropped by kernel
[ec2-user@ip-172-31-19-6 ~]$ sudo tcpdump -r lab2packets.pcap
reading from file lab2packets.pcap, link-type EN10MB (Ethernet)
04:34:24.941392 IP ip-172-31-19-6.ec2.internal.ssh > pool-173-70-217-190.nwrknj.fios.verizon.net.49758: Flags [P.], seq 2649874588:2649874632, ack 3287637709, win 279, length 44
04:34:24.941468 IP ip-172-31-19-6.ec2.internal.ssh > pool-173-70-217-190.nwrknj.fios.verizon.net.49758: Flags [P.], seq 44:160, ack 1, win 279, length 116
04:34:24.952310 IP pool-173-70-217-190.nwrknj.fios.verizon.net.49758 > ip-172-31-19-6.ec2.internal.ssh: Flags [.], ack 44, win 510, length 0
04:34:25.002815 IP pool-173-70-217-190.nwrknj.fios.verizon.net.49758 > ip-172-31-19-6.ec2.internal.ssh: Flags [.], ack 160, win 509, length 0
04:34:38.073255 IP ip-172-31-19-6.ec2.internal.34788 > 169.254.169.123.ntp: NTPv4, Client, length 48
04:34:38.073626 IP 169.254.169.123.ntp > ip-172-31-19-6.ec2.internal.34788: NTPv4, Server, length 48
04:34:54.270900 IP ip-172-31-19-6.ec2.internal.53385 > 169.254.169.123.ntp: NTPv4, Client, length 48
04:34:54.271241 IP 169.254.169.123.ntp > ip-172-31-19-6.ec2.internal.53385: NTPv4, Server, length 48
04:34:59.437454 ARP, Request who-has ip-172-31-16-1.ec2.internal tell ip-172-31-19-6.ec2.internal, length 28
04:34:59.437589 ARP, Reply ip-172-31-16-1.ec2.internal is-at 0:a:1d:51:a2:77:39 (oui Unknown), length 42
04:35:10.413857 IP ip-172-31-19-6.ec2.internal.38139 > 169.254.169.123.ntp: NTPv4, Client, length 48
04:35:10.414247 IP 169.254.169.123.ntp > ip-172-31-19-6.ec2.internal.38139: NTPv4, Server, length 48
04:35:19.997142 ARP, Request who-has ip-172-31-19-6.ec2.internal tell ip-172-31-16-1.ec2.internal, length 42
04:35:19.997160 ARP, Reply ip-172-31-19-6.ec2.internal is-at 0:a:f3:19:22:27:01 (oui Unknown), length 28
04:35:26.713467 IP ip-172-31-19-6.ec2.internal.51254 > 169.254.169.123.ntp: NTPv4, Client, length 48
04:35:26.713895 IP 169.254.169.123.ntp > ip-172-31-19-6.ec2.internal.51254: NTPv4, Server, length 48
04:35:31.949454 ARP, Request who-has ip-172-31-16-1.ec2.internal tell ip-172-31-19-6.ec2.internal, length 28
04:35:31.949576 ARP, Reply ip-172-31-16-1.ec2.internal is-at 0:a:1d:51:a2:77:39 (oui Unknown), length 42
04:35:42.835687 IP ip-172-31-19-6.ec2.internal.41366 > 169.254.169.123.ntp: NTPv4, Client, length 48
04:35:42.836064 IP 169.254.169.123.ntp > ip-172-31-19-6.ec2.internal.41366: NTPv4, Server, length 48
04:35:59.038490 IP ip-172-31-19-6.ec2.internal.45239 > 169.254.169.123.ntp: NTPv4, Client, length 48
04:35:59.038883 IP 169.254.169.123.ntp > ip-172-31-19-6.ec2.internal.45239: NTPv4, Server, length 48
[ec2-user@ip-172-31-19-6 ~]$
```

Perform the EC2 backup and restore

To take the backup of instance first we need to take the snapshot of the existing load balancer instance from EBS-Volumes

My Classrooms Workbench Volumes | EC2 Management Con...

Search for services, features, marketplace products, and doc [Alt+S]

aws Services

Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances New
Dedicated Hosts
Scheduled Instances
Capacity Reservations

Images AMIs

Elastic Block Store Volumes Snapshots Lifecycle Manager

Network & Security Security Groups New Elastic IPs New Placement Groups

Create Volume Actions

Filter by tags and attributes or search by keyword

	Name	Volume ID	Size	Volume Type	IOPS	Throughput	Snapshot	Created
1	vol-05476cd...	8 GiB	gp2	100	-	-	snap-097c45e6d3c6e1d1b	March 29, 2021 at 8:20:02 PM UTC-4
2	vol-0bb34a7...	8 GiB	gp2	100	-	-	snap-097c45e6d3c6e1d1b	March 29, 2021 at 8:17:51 PM UTC-4
3	vol-0475344...	8 GiB	gp2	100	-	-	snap-097c45e6d3c6e1d1b	March 29, 2021 at 8:14:31 PM UTC-4
4	vol-0a9c109...	8 GiB	gp2	100	-	-	snap-097c45e6d3c6e1d1b	March 29, 2021 at 8:10:07 PM UTC-4
5	vol-0b130d6...	8 GiB	gp2	100	-	-	snap-097c45e6d3c6e1d1b	March 29, 2021 at 8:01:41 PM UTC-4

Select a volume above

Feedback English (US) ▾

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

So last volume is the volume for the load balance so we will use that volume to create the snapshot

My Classrooms Workbench Volumes | EC2 Management Con...

Search for services, features, marketplace products, and doc [Alt+S]

aws Services

Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances New
Dedicated Hosts
Scheduled Instances
Capacity Reservations

Images AMIs

Elastic Block Store Volumes Snapshots Lifecycle Manager

Network & Security Security Groups New Elastic IPs New Placement Groups

Create Volume Actions

Filter by tags and attributes or search by keyword

	Created	Availability Zone	State	Alarm Status	Attachment Information	Monitoring	Volume St
1	March 29, 2021 at 8:20:02 PM UTC-4	us-east-1b	in-use	None	i-0210a7b472c9051a...	Okay	Okay
2	March 29, 2021 at 8:17:51 PM UTC-4	us-east-1e	in-use	None	i-0f0645e046a02f45e...	Okay	Okay
3	March 29, 2021 at 8:14:31 PM UTC-4	us-east-1c	in-use	None	i-05f384696b14e49c...	Okay	Okay
4	March 29, 2021 at 8:10:07 PM UTC-4	us-east-1c	in-use	None	i-02e30c4a30629cc5...	Okay	Okay
5	March 29, 2021 at 8:01:41 PM UTC-4	us-east-1b	in-use	None	i-07c920a1be63923d...	Okay	Okay

Select a volume above

https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#Instances:search=i-07c920a1be63923db;sort=instanceId

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

When we clicked to the last volume it redirected to us the load balancer instance so in that way, we can verify whether we are creating snapshot for the correct volume.

The screenshot shows the AWS EC2 Management Console interface. On the left, there's a navigation sidebar with various services like Instance Types, Launch Templates, and Network & Security. The main area is titled 'Instances (1)' and shows a single instance named 'Load Balancer' with the ID 'i-07c920a1be63923db'. The instance is listed as 'Running' with an 't2.micro' instance type, a '2/2 checks ...' status check, and an '1 alarm...' alarm status. The instance is located in the 'us-east-1b' availability zone. At the top right, there are buttons for 'Launch instances' and other actions.

To create snapshot from action select create snapshot

The screenshot shows the AWS Volumes page. In the center, there's a table of volumes. A context menu is open over one of the volumes, with the 'Create Snapshot' option highlighted. The table below shows several volumes, each with a unique ID, IOPS, Throughput, and a corresponding snapshot ID and creation date. The bottom part of the screen shows detailed information for a selected volume, including its Volume ID, Snapshot ID, and creation date.

Volume Type	IOPS	Throughput	Snapshot	Created
gp2	100	-	snap-097c45e6d3c6e1d1b	March 29, 2021 at 8:20:02 PM UTC-4
gp2	100	-	snap-097c45e6d3c6e1d1b	March 29, 2021 at 8:17:51 PM UTC-4
gp2	100	-	snap-097c45e6d3c6e1d1b	March 29, 2021 at 8:14:31 PM UTC-4
gp2	100	-	snap-097c45e6d3c6e1d1b	March 29, 2021 at 8:10:07 PM UTC-4
gp2	100	-	snap-097c45e6d3c6e1d1b	March 29, 2021 at 8:01:41 PM UTC-4

Provide the description for snapshot

The screenshot shows the AWS EC2 Management Console with the 'Create Snapshot' wizard open. The top navigation bar includes tabs for 'My Classrooms', 'Workbench', and 'Create Snapshot | EC2 Management'. The main content area is titled 'Create Snapshot' and shows a volume named 'vol-0b130d66178bac7d2'. A description field contains 'Snapshot for load balancer'. The 'Encrypted' option is set to 'Not Encrypted'. Below these fields is a section for tags, which currently has no tags. A note says 'Choose the Add tag button or click to add a Name tag'. An 'Add Tag' button is available, along with a note about the remaining 50 tags. At the bottom right are 'Cancel' and 'Create Snapshot' buttons. The footer of the page includes links for Feedback, English (US), Privacy Policy, Terms of Use, and Cookie preferences.

Snapshot created successfully.

The screenshot shows the same AWS EC2 Management Console interface as the previous one, but now with a green message box at the top indicating a successful operation. The message reads 'Create Snapshot Request Succeeded' and shows the snapshot ID 'snap-06e980bbae2ddc35b'. Below the message box are two buttons: 'Manage Fast Snapshot Restore' and 'Close'. The footer of the page includes links for Feedback, English (US), Privacy Policy, Terms of Use, and Cookie preferences.

Under snapshots we can see the newly created snapshots

The screenshot shows the AWS EC2 Management Console with the 'Schemas' tab selected. On the left, the navigation menu includes 'Instance Types', 'Launch Templates', 'Spot Requests', 'Savings Plans', 'Reserved Instances', 'Dedicated Hosts', 'Scheduled Instances', 'Capacity Reservations', 'Images', 'AMIs', 'Elastic Block Store', 'Volumes', 'Schemas', and 'Lifecycle Manager'. Under 'Network & Security', there are 'Security Groups', 'Elastic IPs', and 'Placement Groups'. The main pane displays a table of snapshots, with one row selected: 'snap-06e980bbae2ddc35b'. The details for this snapshot are shown in a modal: 'Description' tab, 'Permissions' tab (disabled), 'Tags' tab (disabled). The snapshot details are: Snapshot ID: snap-06e980bbae2ddc35b, Status: completed, Volume: vol-0b130d66178bac7d2, Progress: 100%, Capacity: 8 GiB, Encryption: Not Encrypted.

To register the AMI with the newly created snapshot, first we need to stop the load balancer server, so that it will not create conflicts.

The screenshot shows the AWS EC2 Management Console with the 'Instances' tab selected. The left navigation menu includes 'EC2 Dashboard', 'Events', 'Tags', 'Limits', 'Instances', 'Images', and 'Elastic Block Store'. The 'Instances' section is expanded, showing 'Instances (1/5)'. A table lists five instances: Server3, Load Balancer, Server4, Server1, and Server2. The 'Load Balancer' instance is selected. The 'Actions' dropdown menu is open over this instance, showing options: 'Stop instance', 'Start instance', 'Reboot instance', 'Hibernate instance', and 'Terminate instance'. Below the table, a detailed view for the selected 'Load Balancer' instance (i-07c920a1be63923db) is shown. The 'Details' tab is selected, followed by 'Security', 'Networking', 'Storage', 'Status checks', 'Monitoring', and 'Tags'. At the bottom, there are tabs for 'Instance summary' and 'Info'.

Successfully stopped i-07c920a1be63923db

Instances (1/5) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
Server3	i-0f0645e046a02f45e	Running	t2.micro	2/2 checks ...	1 alarm...	us-east-1e
Load Balancer	i-07c920a1be63923db	Stopping	t2.micro	2/2 checks ...	1 alarm...	us-east-1b
Server4	i-0210a7b472c9051ab	Running	t2.micro	2/2 checks ...	1 alarm...	us-east-1b
Server1	i-02e30c4a30629cc5c	Running	t2.micro	2/2 checks ...	1 alarm...	us-east-1c
Server2	i-05f384696b14e49cd	Running	t2.micro	2/2 checks ...	1 alarm...	us-east-1c

Instance: i-07c920a1be63923db (Load Balancer)

Details **Security** **Networking** **Storage** **Status checks** **Monitoring** **Tags**

Instance summary **Info**

To create the AMI with the newly created snapshot we can go to the snapshot and select create image

Create Snapshot Actions

- Delete
- Create Volume
- Manage Fast Snapshot Restore
- Create Image**
- Copy
- Modify Permissions
- Add/Edit Tags

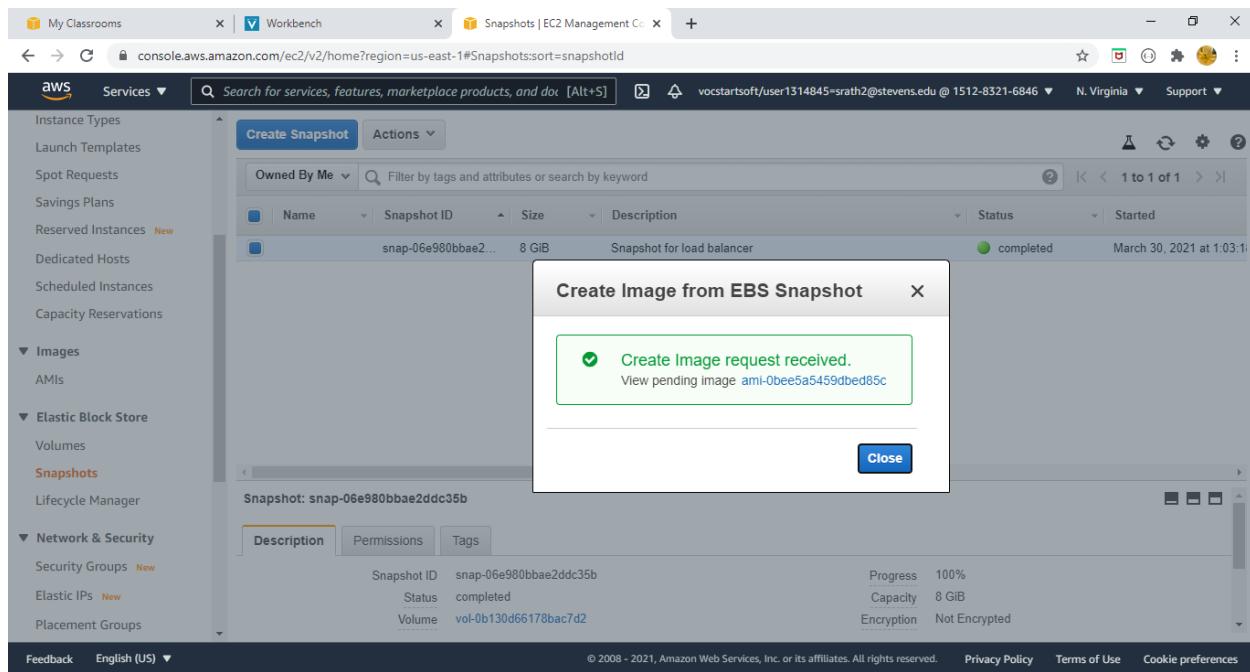
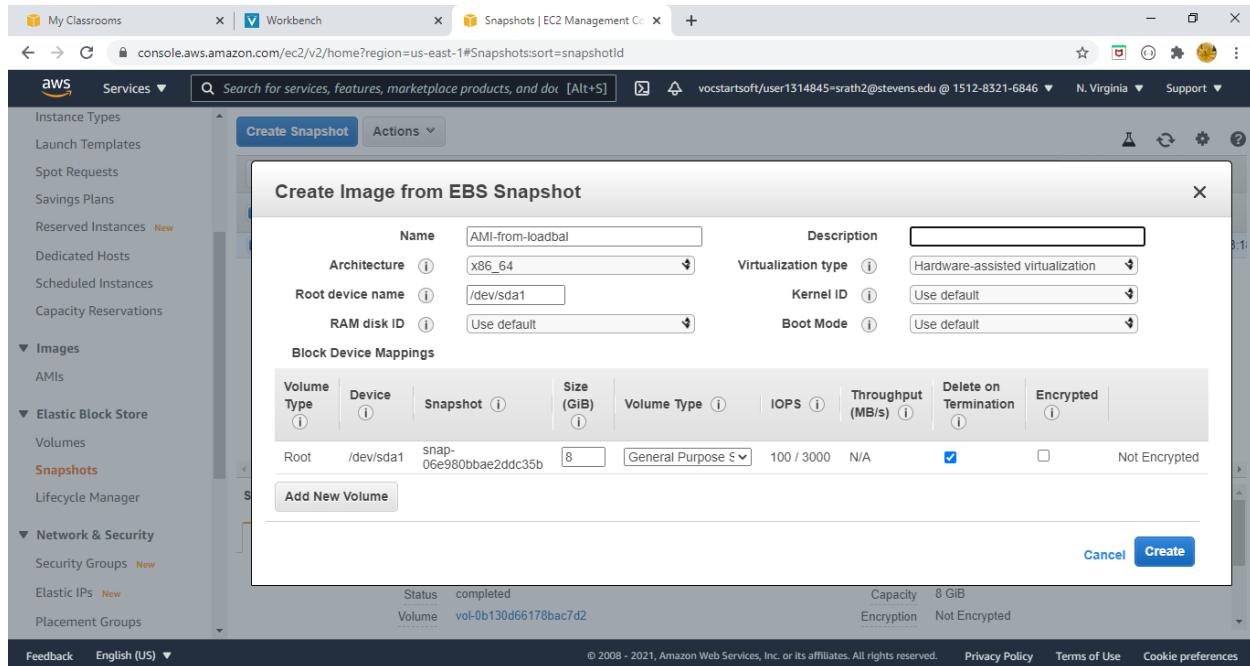
Snapshots | EC2 Management Con...

Snapshot: snap-06e980bbbe2ddc35b

Description **Permissions** **Tags**

Snapshot ID	snap-06e980bbbe2ddc35b	Progress	100%
Status	completed	Capacity	8 GiB
Volume	vol-0b130d66178bac7d2	Encryption	Not Encrypted

We want to create the copy of the load balancer so no we will not change any fields



AMI Image created successfully

The screenshot shows the AWS EC2 Management Console interface. The left sidebar navigation includes 'Instances', 'Images' (selected), 'AMIs' (selected), 'Elastic Block Store', and 'Network & Security'. The main content area displays a table of AMIs under the heading 'Owned by me'. One entry is visible: 'AMI-from-loadbal' with AMI ID 'ami-0bee5a5459dbed85c', Source '151283216846...', Owner '151283216846', Visibility 'Private', Status 'available', and Creation Date 'March 30, 2021 at 1:10:06 A...'. Below the table, there's a 'Details' tab and an 'Edit' button. The 'Actions' dropdown menu is open, listing: Launch, Spot Request, Deregister, Register New AMI, Copy AMI, Modify Image Permissions, Add/Edit Tags, Modify Boot Volume Setting, and EC2 Image Builder.

To launch the AMIs, under AMIs we can see the newly created AMI and through actions it can be launched.

This screenshot is similar to the one above, showing the AWS EC2 Management Console. The 'Actions' dropdown menu is now specifically focused on the 'Launch' option, which is highlighted with a dark background. The rest of the interface, including the sidebar navigation, the table of AMIs, and the 'Details' tab below, remains the same as in the first screenshot.

Select free tier instance and click review and launch

The screenshot shows the AWS Launch Instance Wizard Step 2: Choose an Instance Type. The t2.micro instance is selected, highlighted with a blue border and labeled "Free tier eligible". Other instances listed include t2.nano, t2.small, t2.medium, t2.large, t2.xlarge, t2.2xlarge, and t3.nano. The interface includes a table with columns for Family, Type, vCPUs, Memory (GiB), Instance Storage (GB), EBS-Optimized Available, Network Performance, and IPv6 Support. Buttons at the bottom include Cancel, Previous, Review and Launch (highlighted in blue), and Next: Configure Instance Details.

We can use the existing security group for load balancer or can create the new one.

The screenshot shows the AWS Launch Instance Wizard Step 6: Configure Security Group. A new security group is being created, named "launch-wizard-6". The "Assign a security group:" section has "Create a new security group" selected. The "Security group name:" field contains "launch-wizard-6". The "Description:" field contains "launch-wizard-6 created 2021-03-30T01:12:15.336-04:00". Below this, a table lists three inbound rules: SSH (Protocol TCP, Port Range 22, Source My IP, Description e.g. SSH for Admin Desktop), HTTP (Protocol TCP, Port Range 80, Source Anywhere, Description e.g. SSH for Admin Desktop), and HTTPS (Protocol TCP, Port Range 443, Source Anywhere, Description e.g. SSH for Admin Desktop). An "Add Rule" button is available. Buttons at the bottom include Cancel, Previous, Review and Launch (highlighted in blue), and Next: Configure Instance Details.

My Classrooms Workbench Launch instance wizard | EC2 Ma... +

console.aws.amazon.com/ec2/v2/home?region=us-east-1#LaunchInstanceWizard:ami=ami-0bee5a5459dbed85c

Services Search for services, features, marketplace products, and doc [Alt+S] vocstartsoft/user1314845@sra... N. Virginia 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.

AMI Details

AMI-from-loadbal - ami-0bee5a5459dbed85c
Root Device Type: ebs Virtualization type: hvm

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

Security group name: launch-wizard-6
Description: launch-wizard-6 created 2021-03-30T01:12:15.336-04:00

Type (i) Protocol (i) Port Range (i) Source (i) Description (i) Cancel Previous Launch

Feedback English (US) © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

My Classrooms Workbench Launch instance wizard | EC2 Ma... +

console.aws.amazon.com/ec2/v2/home?region=us-east-1#LaunchInstanceWizard:ami=ami-0bee5a5459dbed85c

McAfee™ WebAdvisor We're scanning your download just to be safe. Scanning...

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

Instance Type

Instance Type	ECUs	vCPUs
t2.micro	-	1

Security Groups

Security group name: launch-wizard-6
Description: launch-wizard-6 created 2021-03-30T01:12:15.336-04:00

Type (i) Protocol (i) Port Range (i) Source (i) Description (i) Cancel Previous 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: LBBKP_lab2_keypair
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.

Feedback English (US) © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences Show all

The screenshot shows the AWS EC2 Launch instance wizard with the 'Launch Status' step selected. A green box at the top indicates that instances are now launching, with a link to view the launch log. Below it, a blue box provides information about estimated charges and billing alerts. A section titled 'How to connect to your instances' explains the instance's current state and how to monitor it. A 'Helpful resources' section lists links to the User Guide and Discussion Forum. The bottom of the page includes standard AWS footer links like Feedback, English (US), Privacy Policy, Terms of Use, and Cookie preferences.

After successful launch ec2 instance is in running status.

The screenshot shows the AWS EC2 Management Console on the 'Instances' page. A green box at the top indicates that an instance has been successfully stopped. The main table displays one instance named 'load-restored' with the ID 'i-00d7e82ea2000fd7e'. The instance is shown as 'Running' with a green status indicator. The 'Instance summary' section provides detailed information about the instance's network settings, including its Public IPv4 address (18.205.157.65), Private IPv4 address (172.31.88.151), Public IPv4 DNS (ec2-18-205-157-65.compute-1.amazonaws.com), Private IPv4 DNS (ip-172-31-88-151.ec2.internal), and VPC ID (vpc-2e883e53). The left sidebar shows navigation links for Instances, Images, Elastic Block Store, and Network & Security.

Connecting to the restored instance using ssh

```
smita1611@LAPTOP-08H75I8L:/mnt/c/Users/smita
smita1611@LAPTOP-08H75I8L:/mnt/c/Users/smita$ chmod 400 ~/.ssh/LBBKP_lab2_keypair.pem
smita1611@LAPTOP-08H75I8L:/mnt/c/Users/smita$ ssh -i ~/.ssh/LBBKP_lab2_keypair.pem ec2-user@ec2-18-205-157-65.compute-1.amazonaws.com
```

```
ec2-user@ip-172-31-88-151:~ 
smita1611@LAPTOP-08H75I8L:/mnt/c/Users/smita$ chmod 400 ~/.ssh/LBBKP_lab2_keypair.pem
smita1611@LAPTOP-08H75I8L:/mnt/c/Users/smita$ ssh -i ~/.ssh/LBBKP_lab2_keypair.pem ec2-user@ec2-18-205-157-65.compute-1.amazonaws.com
The authenticity of host 'ec2-18-205-157-65.compute-1.amazonaws.com (18.205.157.65)' can't be established.
ECDSA key fingerprint is SHA256:bBPlmsuh+8fpjpuT45C48n0UpyIQTH9tDpg2WkxnRBk.
Are you sure you want to continue connecting (yes/no/[fingerprint])?
Host key verification failed.

ec2-user@ip-172-31-88-151:~ 
smita1611@LAPTOP-08H75I8L:/mnt/c/Users/smita$ ssh -i ~/.ssh/LBBKP_lab2_keypair.pem ec2-user@ec2-18-205-157-65.compute-1.amazonaws.com
The authenticity of host 'ec2-18-205-157-65.compute-1.amazonaws.com (18.205.157.65)' can't be established.
ECDSA key fingerprint is SHA256:bBPlmsuh+8fpjpuT45C48n0UpyIQTH9tDpg2WkxnRBk.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-18-205-157-65.compute-1.amazonaws.com,18.205.157.65' (ECDSA) to the list of known hosts.
Last login: Tue Mar 30 04:22:00 2021 from pool-173-70-217-190.nvrknj.fios.verizon.net
[ec2-user@ip-172-31-88-151 ~]$
```

Now if we go and check the nginx.conf file, it has the same changes which we did for load balancer.

```
ec2-user@ip-172-31-88-151:/etc/nginx
[mita1611@LAPTOP-08H751BL:~/c/Users/smita]$ chmod 400 ~/ssh/LBBKP_lab2_keypair.pem
[mita1611@LAPTOP-08H751BL:~/c/Users/smita]$ ssh -i ~/ssh/LBBKP_lab2_keypair.pem ec2-user@ec2-18-205-157-65.compute-1.amazonaws.com
The authenticity of host 'ec2-18-205-157-65.compute-1.amazonaws.com (18.205.157.65)' can't be established.
ECDSA key fingerprint is SHA256:bbP1msuh+8fppjputT45C48n0UpyIQTH9tbpz2WKnRBk.
Are you sure you want to continue connecting (yes/no/[fingerprint])?
Host key verification failed.
[mita1611@LAPTOP-08H751BL:~/c/Users/smita]$ ssh -i ~/ssh/LBBKP_lab2_keypair.pem ec2-user@ec2-18-205-157-65.compute-1.amazonaws.com
The authenticity of host 'ec2-18-205-157-65.compute-1.amazonaws.com (18.205.157.65)' can't be established.
ECDSA key fingerprint is SHA256:bbP1msuh+8fppjputT45C48n0UpyIQTH9tbpz2WKnRBk.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-18-205-157-65.compute-1.amazonaws.com,18.205.157.65' (ECDSA) to the list of known hosts.
Last login: Tue Mar 30 04:22:00 2021 from pool-173-70-217-190.mwrkj.fios.verizon.net
[mita1611@ip-172-31-88-151 ~]$ cd /etc/nginx/
[mita1611@ip-172-31-88-151 nginx]$ sudo vim nginx.conf
[mita1611@ip-172-31-88-151 nginx]$ sudo vim nginx.conf
```

```
ec2-user@ip-172-31-88-151:/etc/nginx
# For more information on configuration, see:
#   * Official English Documentation: http://nginx.org/en/docs/
#   * Official Russian Documentation: http://nginx.org/ru/docs/

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 ec2-54-159-25-179.compute-1.amazonaws.com weight=1;
        server ec2-18-208-189-103.compute-1.amazonaws.com weight=2;
        server ec2-54-210-79-200.compute-1.amazonaws.com weight=1;
        server ec2-3-95-6-138.compute-1.amazonaws.com weight=2;
    )
    log_format main '$remote_addr - $remote_user [$time_local] "$request" '
                    '$status $body_bytes_sent "$http_referer" '
                    '"$http_user_agent" "$http_x_forwarded_for"';

    access_log /var/log/nginx/access.log main;

    sendfile      on;
    tcp_nopush    on;
    tcp_nodelay   on;
    keepalive_timeout 65;
    types_hash_max_size 4096;

    include       /etc/nginx/mime.types;
    default_type  application/octet-stream;

    # Load modular configuration files from the /etc/nginx/conf.d directory.
    # See http://nginx.org/en/docs/ngx_core_module.html#include
"nginx.conf" 97L, 2698C
```

```
ec2-user@ip-172-31-88-151:/etc/nginx
tcp_nopush      on;
tcp_nodelay     on;
keepalive_timeout 65;
types_hash_max_size 4096;

include          /etc/nginx/mime.types;
default_type    application/octet-stream;

# Load modular configuration files from the /etc/nginx/conf.d directory.
# See http://nginx.org/en/docs/ngx_core_module.html#include
# For more information.
include /etc/nginx/conf.d/*.conf;

server {
    listen      80;
    listen      [::]:80;
    server_name myapp.com;
    root       /usr/share/nginx/html;

    # Load configuration files for the default server block.
    include /etc/nginx/default.d/*.conf;

    location /{
        proxy_pass http://myapp;
    }

    error_page 404 /404.html;
        location = /40x.html {

    }

    error_page 500 502 503 504 /50x.html;
        location = /50x.html {
    }
}

# Settings for a TLS enabled server.

server {
    listen      443 ssl http2;
    listen      [::]:443 ssl http2;
    server_name _;
    root       /usr/share/nginx/html;
}

-- INSERT --
```

Now if I try to connect to one of the servers and then run the visit-server tool, it should behave exactly same when these servers are linked to load balancer, now they are linked to newly launched ec2 instance.

```
ec2-user@ip-172-31-19-6:~  
smita1611@LAPTOP-08H751BL:/mnt/c/Users/smita$ ssh -i ~/ssh/s1_lab2_keypair.pem ec2-user@ec2-54-25-179.compute-1.amazonaws.com  
Last login: Tue Mar 30 04:23:26 2021 from pool-173-70-217-190.nwrknj.fios.verizon.net  
_ _| |_) / Amazon Linux 2 AMI  
_ _| \_\_|__|  
https://aws.amazon.com/amazon-linux-2/  
[ec2-user@ip-172-31-19-6 ~]$
```

Public dns name which is provided is the dns name of the newly created server.

```
ec2-user@ip-172-31-19-6:~  
smita1611@LAPTOP-08H751BL:/mnt/c/Users/smita$ ssh -i ~/ssh/s1_lab2_keypair.pem ec2-user@ec2-54-25-179.compute-1.amazonaws.com  
Last login: Tue Mar 30 05:22:24 2021 from pool-173-70-217-190.nwrknj.fios.verizon.net  
_ _| |_) / Amazon Linux 2 AMI  
_ _| \_\_|__|  
https://aws.amazon.com/amazon-linux-2/  
[ec2-user@ip-172-31-19-6 ~]$ ruby visit_server.rb -d ec2-18-205-157-65.compute-1.amazonaws.com
```

```
ec2-user@ip-172-31-19-6:~  
smita1611@LAPTOP-08H751BL:/mnt/c/Users/smita$ ssh -i ~/ssh/S1_lab2_keypair.pem ec2-user@ec2-54-159-25-179.compute-1.amazonaws.com  
Last login: Tue Mar 30 05:22:24 2021 from pool-173-70-217-190.nwrknj.fios.verizon.net  
_ _|_ / Amazon Linux 2 AMI  
_ \_\_|_  
https://aws.amazon.com/amazon-linux-2/  
[ec2-user@ip-172-31-19-6 ~]$ ruby visit_server.rb -d ec2-18-205-157-65.compute-1.amazonaws.com  
Starting to visit load balancing server  
.....
```

```
ec2-user@ip-172-31-19-6:~  
smita1611@LAPTOP-08H751BL:/mnt/c/Users/smita$ ssh -i ~/ssh/S1_lab2_keypair.pem ec2-user@ec2-54-159-25-179.compute-1.amazonaws.com  
Last login: Tue Mar 30 05:22:24 2021 from pool-173-70-217-190.nwrknj.fios.verizon.net  
_ _|_ / Amazon Linux 2 AMI  
_ \_\_|_  
https://aws.amazon.com/amazon-linux-2/  
[ec2-user@ip-172-31-19-6 ~]$ ruby visit_server.rb -d ec2-18-205-157-65.compute-1.amazonaws.com  
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  
[ec2-user@ip-172-31-19-6 ~]$
```

Terminated all the instances

The screenshot shows the AWS EC2 Management Console interface. On the left, a navigation sidebar lists various services: Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Images (AMIs), Elastic Block Store (Volumes, Snapshots, Lifecycle Manager), Network & Security (Security Groups, Elastic IPs, Placement Groups). The main content area is titled "Instances (6) Info". It displays a table with columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, and Availability Zone. All six instances listed are in a terminated state (indicated by a red circle with a slash icon). The table includes a search bar at the top and pagination controls at the bottom.

To delete the snapshot first we need to deregister the ami

The screenshot shows the AWS AMI Management Console interface. The left sidebar includes: Instances (Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances), Images (AMIs), and Network & Security (Elastic Block Store, Network & Security). The main area has tabs for "Launch" and "EC2 Image Builder". A dropdown menu under "Actions" is open, showing options: Launch, Spot Request, Deregister, Register New AMI, Copy AMI, Modify Image Permissions, Add/Edit Tags, Modify Boot Volume Setting, and EC2 Image Builder. The "Deregister" option is highlighted. Below this, a table lists an AMI entry: "Image: ami-0bee5a5459dbed85c". The table columns are: Details, Permissions, and Tags. The "Details" tab is selected, showing fields: AMI ID (ami-0bee5a5459dbed85c), Owner (151283216846), AMI Name (AMI-from-loadbal), and Source (151283216846/AMI-from-loadbal). An "Edit" button is located in the top right corner of the table area.

Screenshot of the AWS EC2 Management Console showing the process of deregistering an AMI.

The left sidebar shows navigation options: Instances, Images (selected), AMIs, Elastic Block Store, and Network & Security.

The main area displays a table of registered AMIs:

Name	AMI Name	AMI ID	Source	Owner	Visibility	Status	Creation Date
AMI-from-loadbal	ami-0bee5a5459dbed85c	151283216846...	151283216846	Private	available	available	March 30, 2021 at 1:10:06 A...

A modal dialog titled "Deregister" is open, asking "Are you sure you want to deregister these images? ami-0bee5a5459dbed85c - AMI-from-loadbal". It has "Cancel" and "Continue" buttons.

Below the table, details for the selected AMI (ami-0bee5a5459dbed85c) are shown, including AMI ID, Owner, AMI Name, and Source.

Screenshot of the AWS EC2 Management Console showing the results of deregistering an AMI.

The left sidebar shows navigation options: Instances, Images (selected), AMIs, Elastic Block Store, and Network & Security.

The main area displays a message: "None found" under the heading "Owned by me". It also includes a note: "Use the filter controls above to view Images owned by Amazon and others." Below this is an advertisement for the AWS Marketplace.

The advertisement features the AWS Marketplace logo and the text: "Find more than 500 AMIs of popular Open Source and commercial software from the AWS Marketplace. Build your own custom AMI using EC2 Image Builder".

Below the advertisement, there is a placeholder text: "Select an AMI above".

Then Snapshot can be deleted.

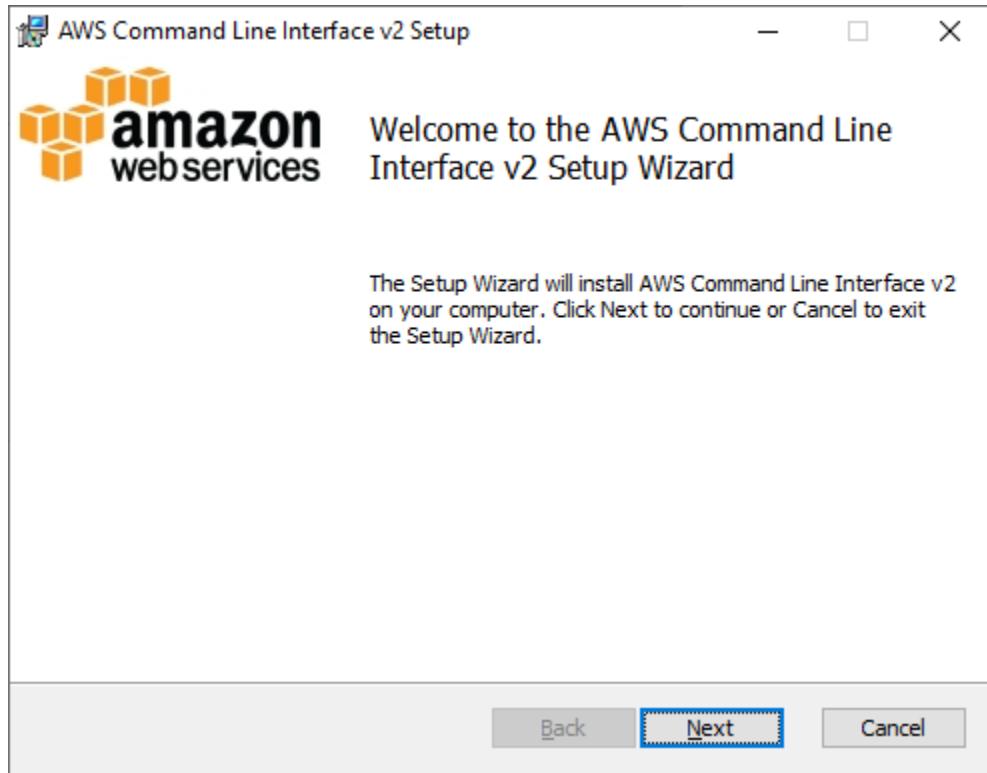
The screenshot shows the AWS EC2 Management Console. On the left, the navigation pane includes sections like Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Scheduled Instances, Capacity Reservations, Images, AMIs, and Elastic Block Store. Under Elastic Block Store, Snapshots is selected. The main content area displays a table of snapshots. A context menu is open over a specific row, with 'Delete' highlighted. The table row shows a snapshot named 'Snapshot: snap-06e980bb...'. The details panel below the table shows the snapshot ID as 'snap-06e980bbae2ddc35b', status as 'completed', volume as 'vol-0b130d66178bac7d2', and other metadata.

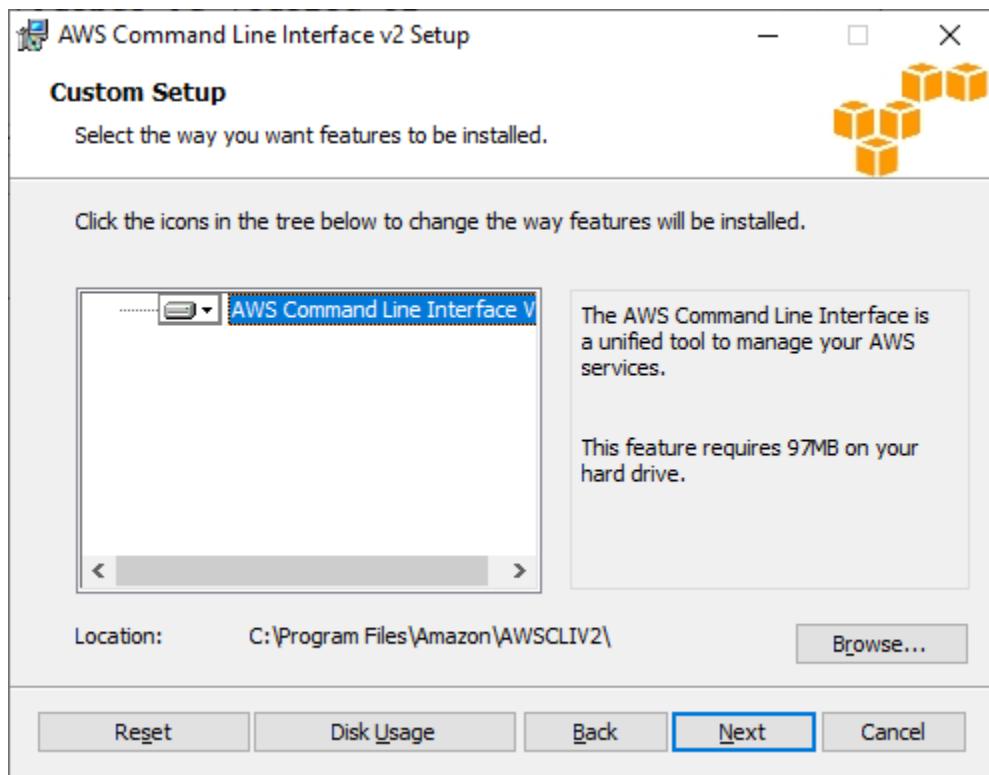
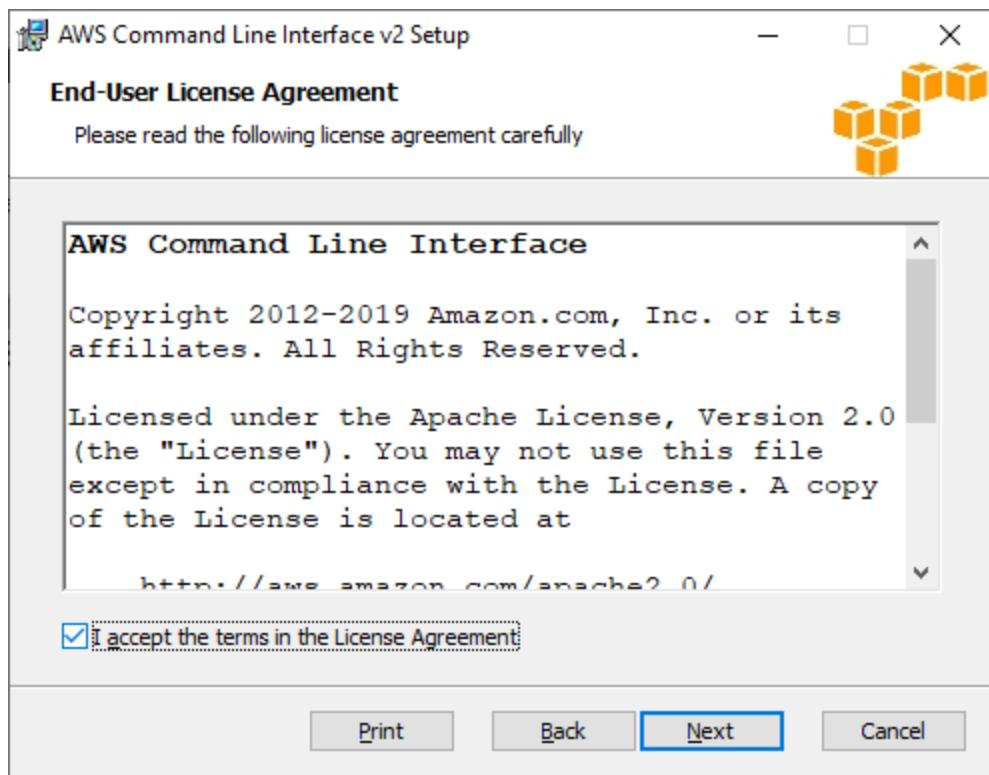
This screenshot shows the same AWS EC2 Management Console interface as the previous one. A 'Delete Snapshot' dialog box is centered on the screen, asking 'Are you sure you want to delete this snapshot?' with a list item 'snap-06e980bbae2ddc35b'. At the bottom of the dialog are 'Cancel' and 'Yes, Delete' buttons. The background table and details panel are visible but dimmed.

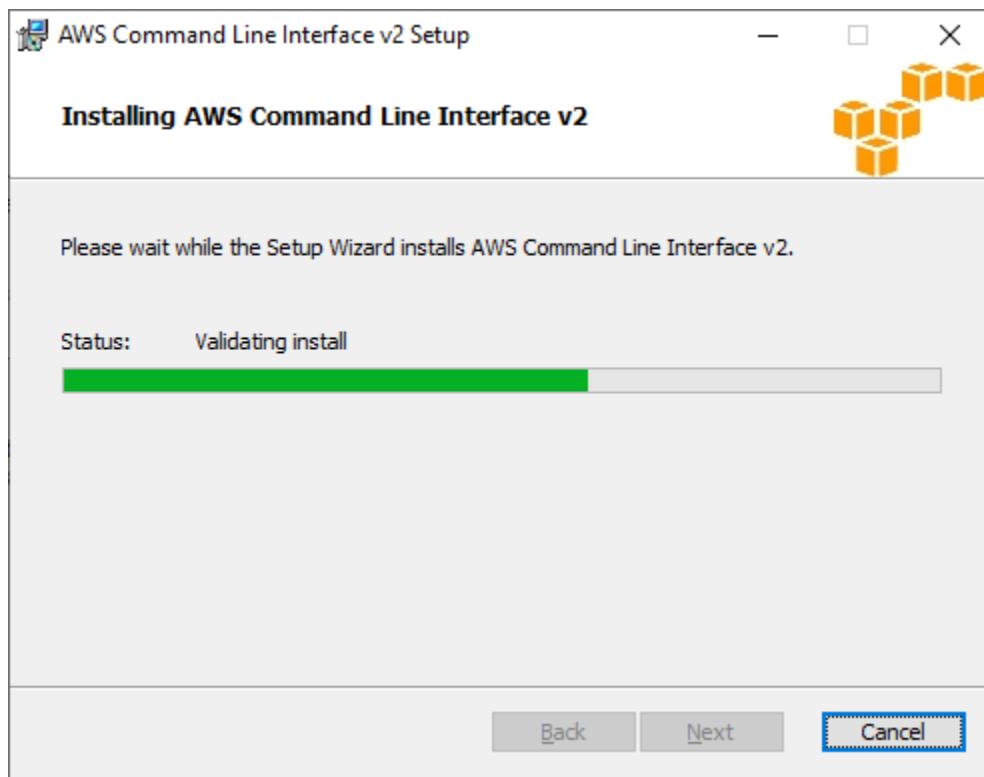
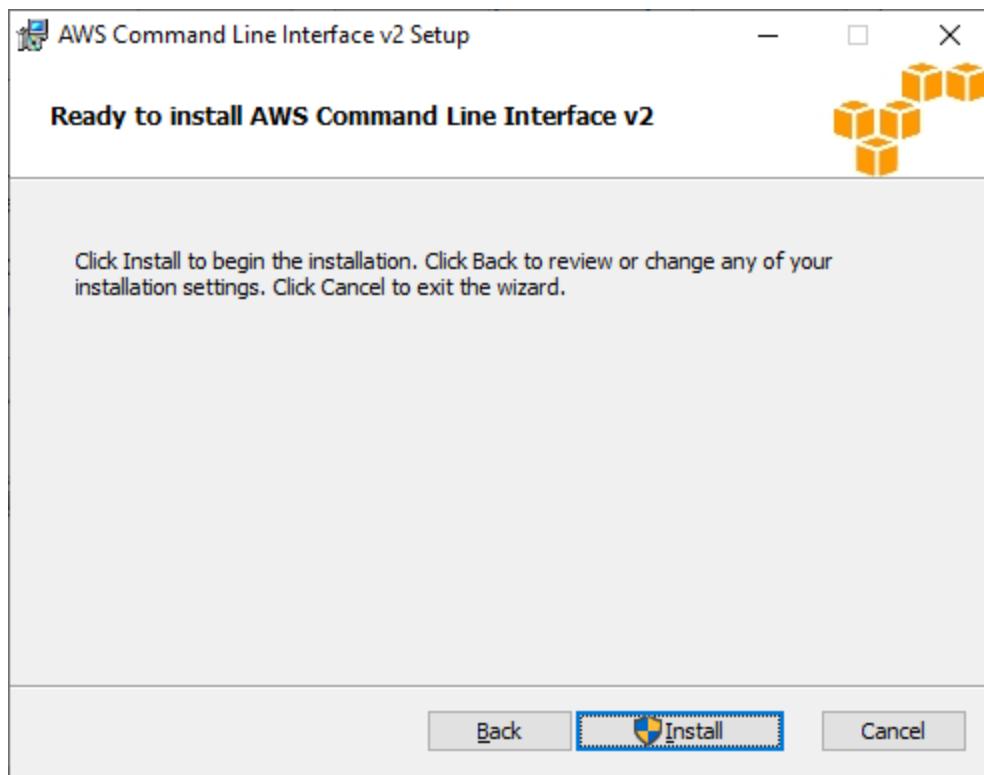
The screenshot shows the AWS EC2 Management Console with the 'Schemas' tab selected. The left sidebar lists various services: Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Scheduled Instances, Capacity Reservations, Images (AMIs), Elastic Block Store (Volumes, Snapshots), Network & Security (Security Groups, Elastic IPs, Placement Groups), and Feedback. The main content area displays a message: 'You do not have any snapshots in this region.' It includes a search bar and a 'Create Snapshot' button.

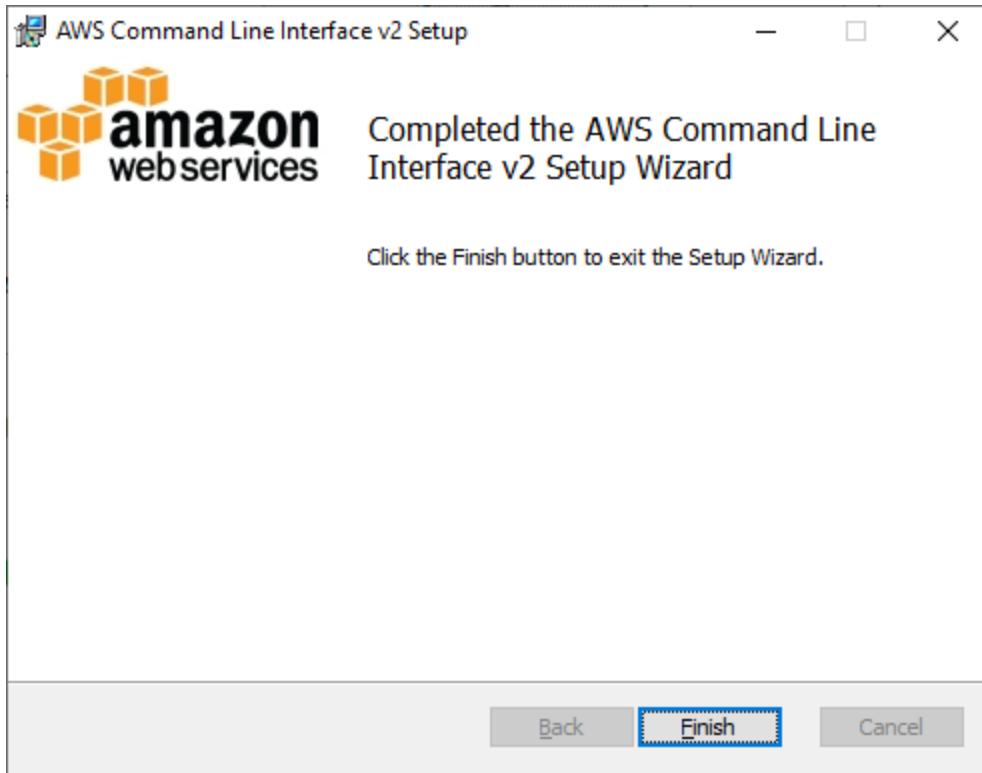
Create EC2 by using command line

First we have to install CLI interface



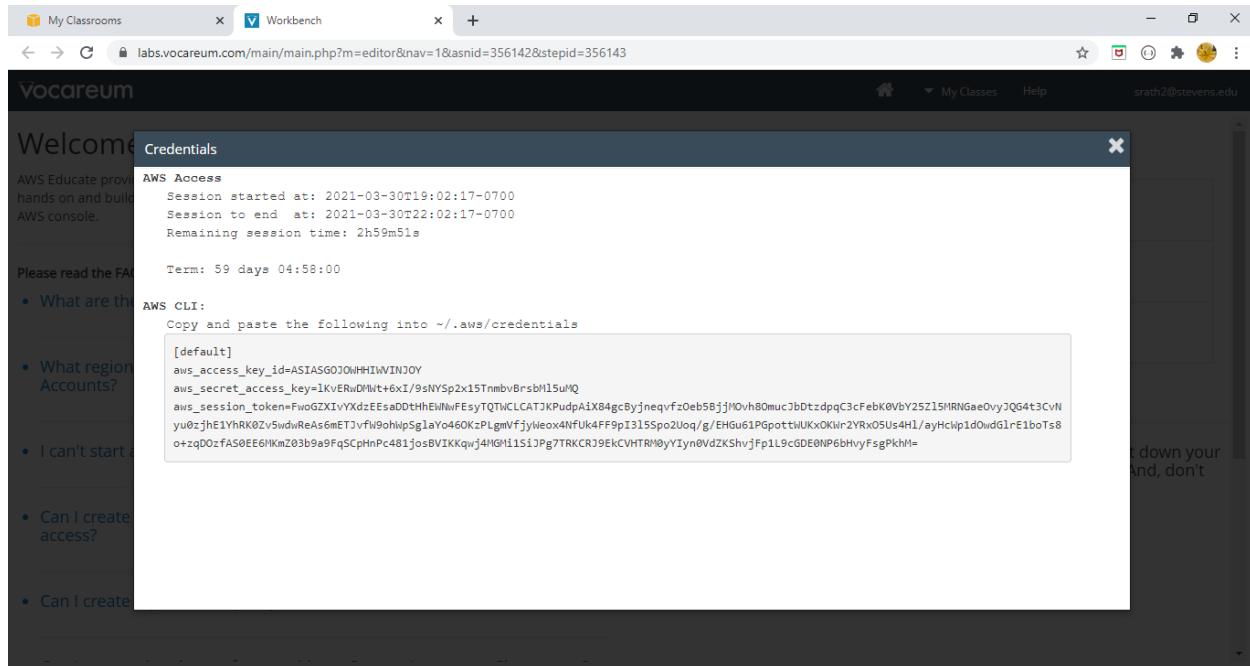






After the installation from the workbench we have to take the access key from account details

A screenshot of a web browser window titled "Vocabareum" showing the "Workbench" tab. The URL in the address bar is "labs.vocabareum.com/main/main.php?m=editor&nav=1&asnid=356142&stepid=356143". The page content includes a "Welcome to your AWS Educate Account" section with a FAQ and a list of frequently asked questions. To the right is a "Your AWS Account Status" section displaying account information: "Active" (full access), "\$99.59" (remaining credits estimated), and "2:47" (session time). Below these are "Account Details" and "AWS Console" buttons. A note at the bottom right encourages responsible usage of the account.



To create ec2 from cli first we need to configure credentials and config file and provide the required access key and secret key from workbench.

```
smita1611@LAPTOP-08H751BL:~$ aws configure
smita1611@LAPTOP-08H751BL:~$ aws configure
AWS Access Key ID [*****SRNZ]: ASIASGOJ0WHHIWVINJOY
AWS Secret Access Key [*****SerR]: lKvERwDMWt+6xI/9sNYSp2x15TrmbvBrsbM15uMQ
Default region name [us-east-1]:
Default output format [None]:
smita1611@LAPTOP-08H751BL:~$
```

Once it is done, go to the `~/.aws/` folder and edit the credentials and provide the session token.

```
smita1611@LAPTOP-08H75I8L: ~/.aws
smita1611@LAPTOP-08H75I8L:~$ cd ~/.aws/
smita1611@LAPTOP-08H75I8L:~/\.aws$ sudo vim credentials
```

```
[default]
aws_access_key_id = ASIASGOJ0WHHIWVINJOY
aws_secret_access_key = lkvERwDMlt+6xI/9sNYSp2x15TrmbvBrsbM15uMQ
aws_session_token=FwoGZXIVYXdzEEsaDDtHhEWnFEsyTQTwCLCATJKPudpAix84gcByjneqvFz0eb5BjjMovh80mucJbdTzdpqC3cFebK0VbY25Z15MRNGaeOvyJQG4t3CvNyU0zjhE1yHRK0Zv5wdwReAs6mETJvfW9ohWpSglaYo460K2PlgmVfjyloex4NFUK4FF9pI315Spo2Uoq/g/EHGu61PGpottWUKxOKWn2YRx05Us4H1/ayHcWlp1d0wdGlrE1boTs8o+zqD0zfas0EE6MKmZ03b9a9FqScpHnPc481josVIKKqvj4MGMi1S1Pg7TRKCRJ9EkCVHTRM0yYIyn0VdzKShvJFp1L9cgDE0NP6bHvyFsgPkhM=
```

Checking the config file.

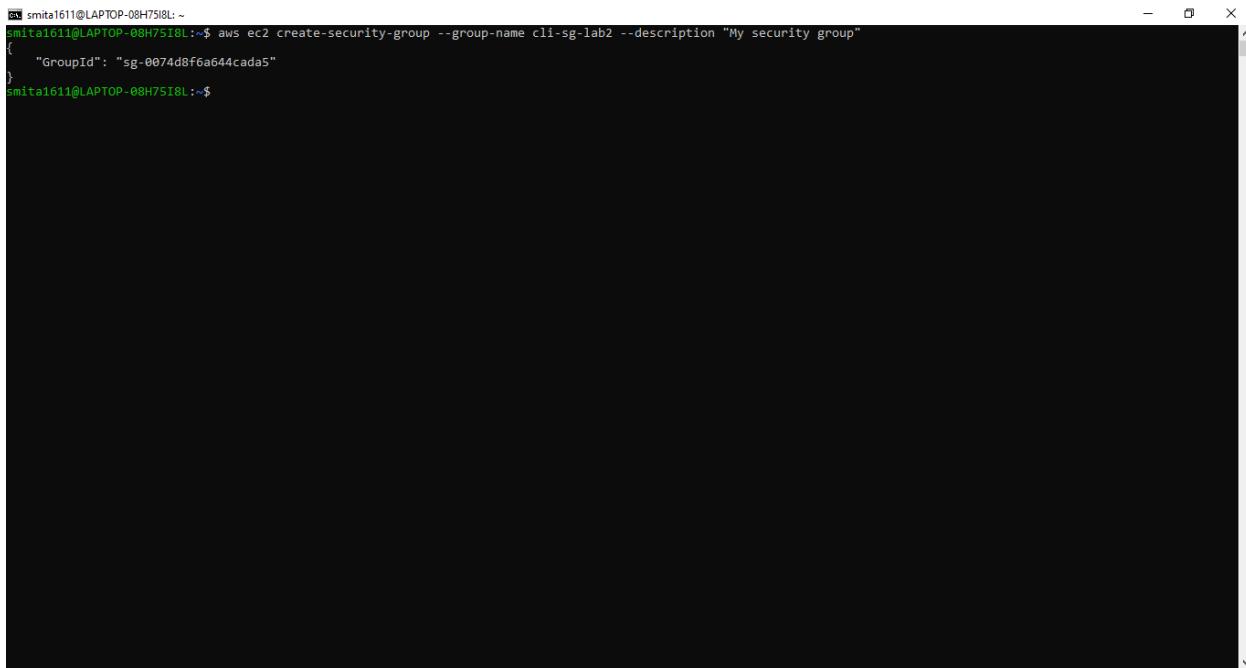
```
smita1611@LAPTOP-08H751BL:~/aws  
smita1611@LAPTOP-08H751BL:~/aws$ cd ~/aws/  
smita1611@LAPTOP-08H751BL:~/aws$ sudo vim credentials  
[sudo] password for smita1611:  
smita1611@LAPTOP-08H751BL:~/aws$ sudo vim credentials  
smita1611@LAPTOP-08H751BL:~/aws$ vim config  
smita1611@LAPTOP-08H751BL:~/aws$ vim config
```

```
smita1611@LAPTOP-08H751BL:~/aws  
default  
region = us-east-1
```

"config" 2L, 29C 1,1 All

After configuration is done, we will create a new security group to use in our ec2 instance.

It will generate the security group id.

A screenshot of a terminal window titled "smita1611@LAPTOP-08H75I8L:~". The window contains a single line of command output:

```
smita1611@LAPTOP-08H75I8L:~$ aws ec2 create-security-group --group-name cli-sg-lab2 --description "My security group"
{
  "GroupId": "sg-0074d8f6a644cada5"
}smita1611@LAPTOP-08H75I8L:~$
```

The terminal has a dark background and light-colored text. The command and its output are visible at the top left, while the rest of the window is mostly black.

We can set the different rules for different ports, suppose if we want this instance to accept incoming traffic from some other instance we can provide in the source group.

```
aws ec2 authorize-security-group-ingress \
  --group-id sg-1234567890abcdef0 \
  --protocol tcp \
  --port 80 \
  --source-group sg-1a2b3c4d
```

I am setting the port 22 with my IP address

```
smita1611@LAPTOP-08H75IBL:~$ aws ec2 create-security-group --group-name cli-sg-lab2 --description "My security group"
{
  "GroupId": "sg-0074d8f6a644cada5"
}
smita1611@LAPTOP-08H75IBL:~$ aws ec2 authorize-security-group-ingress --group-name my-sg --protocol tcp --port 22 --cidr 173.70.217.190/32
An error occurred (InvalidGroup.NotFound) when calling the AuthorizeSecurityGroupIngress operation: The security group 'my-sg' does not exist in default VPC 'vpc-2e883e53'
smita1611@LAPTOP-08H75IBL:~$ aws ec2 authorize-security-group-ingress --group-name cli-sg-lab2 --protocol tcp --port 22 --cidr 173.70.217.190/32
smita1611@LAPTOP-08H75IBL:~$ aws ec2 describe-security-groups --group-names my-sg
```

To view the security group created we can use the below command

```
smita1611@LAPTOP-08H75IBL:~$ aws ec2 describe-security-groups --group-names cli-sg-lab2
{
  "SecurityGroups": [
    {
      "Description": "My security group",
      "GroupName": "cli-sg-lab2",
      "IpPermissions": [
        {
          "FromPort": 22,
          "IpProtocol": "tcp",
          "IpRanges": [
            {
              "CidrIp": "173.70.217.190/32"
            }
          ],
          "Ipv6Ranges": [],
          "PrefixListIds": [],
          "ToPort": 22,
          "UserIdGroupPairs": []
        }
      ],
      "OwnerId": "151283216846",
      "GroupId": "sg-0074d8f6a644cada5",
      "IpPermissionsEgress": [
        {
          "IpProtocol": "-1",
          "IpRanges": [
            {
              "CidrIp": "0.0.0.0/0"
            }
          ],
          "Ipv6Ranges": [],
          "PrefixListIds": [],
          "UserIdGroupPairs": []
        }
      ],
      "VpcId": "vpc-2e883e53"
    }
  ]
}
smita1611@LAPTOP-08H75IBL:~$
```

Checking the same security group in amazon management console

The screenshot shows the AWS Management Console interface for the EC2 service, specifically the Security Groups section. The left sidebar lists various AWS services like Launch Templates, Spot Requests, Savings Plans, etc. The main content area displays the details of a security group named 'sg-0074d8f6a644cada5 - cli-sg-lab2'. The 'Details' tab is selected, showing the following information:

Security group name	Security group ID	Description	VPC ID
cli-sg-lab2	sg-0074d8f6a644cada5	My security group	vpc-2e883e53

Below this, the 'Owner' is listed as '151283216846'. The 'Inbound rules count' is '1 Permission entry' and the 'Outbound rules count' is '1 Permission entry'. There are three tabs at the bottom: 'Inbound rules' (selected), 'Outbound rules', and 'Tags'. The 'Inbound rules' tab shows one rule:

Type	Protocol	Port range	Source	Description - optional
SSH	TCP	22	173.70.217.190/32	-

Creating the keypair through CLI

```
smita1611@LAPTOP-08H75I8L:~$ aws ec2 create-key-pair --key-name cli_lab2_keypair --query 'KeyMaterial' --output text > cli_lab2_keypair.pem
```

Keypair created

The screenshot shows the AWS EC2 Management Console with the 'Key pairs' page open. The left sidebar lists various services: Images, AMIs, Elastic Block Store (Volumes, Snapshots, Lifecycle Manager), Network & Security (Security Groups, Elastic IPs, Placement Groups, Key Pairs), Load Balancing (Load Balancers, Target Groups), and Auto Scaling (Launch Configurations). The 'Key Pairs' section is currently selected. The main content area displays a table titled 'Key pairs (1)'. The table has columns for 'Name', 'Fingerprint', and 'ID'. One row is shown, corresponding to the 'cli_lab2_keypair' created earlier. The 'Actions' button is visible at the top right of the table.

Name	Fingerprint	ID
cli_lab2_keypair	47:ed:03:3e:15:ace2:5a:35:ae:23:81:1...	key-0fac019d5fd7bd5ca

To create the ec2 instance we have to use the following command

Where we have to following details

--image-id - has to be taken from amazon management console

--count – how many instances we want to create

--instance-type – we will create the free tier one t2.micro

--key-name – provide the keypair name

--security-group-ids – provide the security group id which is created

```
smita1611@LAPTOP-08H751BL:~$ aws ec2 create-key-pair --key-name cli_lab2_keypair --query 'KeyMaterial' --output text > cli_lab2_keypair.pem
smita1611@LAPTOP-08H751BL:~$ aws ec2 run-instances --image-id ami-0742b4e673072066f --count 1 --instance-type t2.micro --key-name cli_lab2_keypair --security-group-ids sg-0074d8f6a644cada5
```

Instance is created.

```
smita1611@LAPTOP-08H751BL:~$ aws ec2 create-key-pair --key-name cli_lab2_keypair --query 'KeyMaterial' --output text > cli_lab2_keypair.pem
smita1611@LAPTOP-08H751BL:~$ aws ec2 run-instances --image-id ami-0742b4e673072066f --count 1 --instance-type t2.micro --key-name cli_lab2_keypair --security-group-ids sg-0074d8f6a644cada5
{
    "Groups": [],
    "Instances": [
        {
            "AmiLaunchIndex": 0,
            "ImageId": "ami-0742b4e673072066f",
            "InstanceId": "i-0aeec1bddb9617df",
            "InstanceType": "t2.micro",
            "KeyName": "cli_lab2_keypair",
            "LaunchTime": "2021-03-30T21:42:15.000Z",
            "Monitoring": {
                "State": "disabled"
            },
            "Placement": {
                "AvailabilityZone": "us-east-1b",
                "GroupName": "",
                "Tenancy": "default"
            },
            "PrivateDnsName": "ip-172-31-89-88.ec2.internal",
            "PrivateIpAddress": "172.31.89.88",
            "ProductCodes": [],
            "PublicDnsName": "",
            "State": {
                "Code": 0,
                "Name": "pending"
            },
            "StateTransitionReason": "",
            "SubnetId": "subnet-c56000e4",
            "VpcId": "vpc-2e883e53",
            "Architecture": "x86_64",
            "BlockDeviceMappings": [],
            "ClientToken": "549adcaf-159c-47d7-9bf2-9d9e6d3d7cd1",
            "EbsOptimized": false,
            "EnaSupport": true,
            "Hypervisor": "xen",
            "NetworkInterfaces": [
                {
                    "Attachment": {
                        "AttachTime": "2021-03-30T21:42:15.000Z",
                        "AttachmentId": "eni-attach-07b63381cbc2c9715",
                        "DeleteOnTermination": true
                    }
                }
            ]
        }
    ]
}
```

```
smita1611@LAPTOP-08H75IBL: ~
{
    "Attachment": {
        "AttachmentTime": "2021-03-30T21:42:15.000Z",
        "AttachmentId": "eni-attach-07b63381cbc2c9715",
        "DeleteOnTermination": true,
        "DeviceIndex": 0,
        "Status": "attaching"
    },
    "Description": "",
    "Groups": [
        {
            "GroupName": "cli-sg-lab2",
            "GroupId": "sg-0074d8f6a644cada5"
        }
    ],
    "Ipv6Addresses": [],
    "MacAddress": "12:48:f5:fa:04:13",
    "NetworkInterfaceId": "eni-0144bc2b81255ddcb",
    "OwnerId": "151283216846",
    "PrivateDnsName": "ip-172-31-89-88.ec2.internal",
    "PrivateIpAddress": "172.31.89.88",
    "PrivateIpAddresses": [
        {
            "Primary": true,
            "PrivateDnsName": "ip-172-31-89-88.ec2.internal",
            "PrivateIpAddress": "172.31.89.88"
        }
    ],
    "SourceDestCheck": true,
    "Status": "in-use",
    "SubnetId": "subnet-c56000e4",
    "VpcId": "vpc-2e883e53",
    "InterfaceType": "interface"
},
"RootDeviceName": "/dev/xvda",
"RootDeviceType": "ebs",
"SecurityGroups": [
    {
        "GroupName": "cli-sg-lab2",
        "GroupId": "sg-0074d8f6a644cada5"
    }
],
"SourceDestCheck": true,
```

```
smita1611@LAPTOP-08H75IBL: ~
    },
    "SourceDestCheck": true,
    "Status": "in-use",
    "SubnetId": "subnet-c56000e4",
    "VpcId": "vpc-2e883e53",
    "InterfaceType": "interface"
},
"RootDeviceName": "/dev/xvda",
"RootDeviceType": "ebs",
"SecurityGroups": [
    {
        "GroupName": "cli-sg-lab2",
        "GroupId": "sg-0074d8f6a644cada5"
    }
],
"SourceDestCheck": true,
"StateReason": {
    "Code": "pending",
    "Message": "pending"
},
"VirtualizationType": "hvm",
"CpuOptions": {
    "CoreCount": 1,
    "ThreadsPerCore": 1
},
"CapacityReservationSpecification": {
    "CapacityReservationPreference": "open"
},
"MetadataOptions": [
    {
        "State": "pending",
        "HttpTokens": "optional",
        "HttpPutResponseHopLimit": 1,
        "HttpEndpoint": "enabled"
    }
],
"OwnerId": "151283216846",
"ReservationId": "r-04b0940af447b7707"
}smita1611@LAPTOP-08H75IBL:~$
```

Under instances we can see its running.

The screenshot shows the AWS EC2 Instances page. The left sidebar is collapsed. The main area displays a table titled "Instances (1) Info". The table has columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, and Availability Zone. One row is present, showing "i-0aee1cbdddb9617df" as the Instance ID, "Running" as the Instance state, "t2.micro" as the Instance type, "2/2 checks ..." as the Status check, "1 alarm..." as the Alarm status, and "us-east-1b" as the Availability Zone. A "Clear filters" button is visible above the table.

The screenshot shows the AWS EC2 Instance details page for instance "i-0aee1cbdddb9617df". The left sidebar is collapsed. The main area shows an "Instance summary for i-0aee1cbdddb9617df" table. The table includes fields like Instance ID, Instance state, Instance type, Public IPv4 address, Private IPv4 addresses, Public IPv4 DNS, Private IPv4 DNS, VPC ID, and Subnet ID. The "Instance state" row shows "Running". The "AWS Compute Optimizer finding" row contains a red error message: "User: arn:aws:sts::151283216846:assumed-role/vocstartsoft/user1314845=srath2@stevens.edu is not authorized to perform: compute-optimizer:GetEnrollmentStatus on resource: * with an explicit deny". A "Retry" link is also present. Buttons for "Connect" and "Instance state" are at the top right of the summary table.

Security group and key pair which we provided through cli.

The screenshot shows the AWS EC2 Instance Details page for an instance with ID i-0aee1cbdddb9617df. The instance was launched on Tuesday, March 30, 2021, at 17:42:15 GMT-0400 (Eastern Daylight Time). It is associated with the IAM Role ' - ' and the Owner ID 151283216846. The security group assigned is sg-0074d8f6a644cada5 (cli-sg-lab2). The inbound rules table shows one rule allowing port 22 (TCP) from source 173.70.217.190/32 to security group cli-sg-lab2. The outbound rules table shows one rule allowing all traffic (All) to destination 0.0.0.0/0 from security group cli-sg-lab2.

Port range	Protocol	Source	Security groups
22	TCP	173.70.217.190/32	cli-sg-lab2

Port range	Protocol	Destination	Security groups
All	All	0.0.0.0/0	cli-sg-lab2

References

<https://www.linuxtech.com/capture-analyze-packets-tcpdump-command-linux/>

<https://opensource.com/article/18/10/introduction-tcpdump>

https://docs.amazonaws.cn/en_us/cli/latest/userguide/cli-services-ec2-instances.html

<https://stackoverflow.com/questions/49227466/aws-instance-only-allow-traffic-from-load-balancer>

<https://docs.aws.amazon.com/elasticloadbalancing/latest/application/load-balancer-update-security-groups.html>

<https://rvm.io/>

<https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/ruby-development-environment.html#ruby-development-environment-ruby>

<https://www.cyberciti.biz/faq/nginx-linux-restart/>

<https://docs.aws.amazon.com/prescriptive-guidance/latest/backup-recovery/ec2-backup.html>

<https://docs.aws.amazon.com/prescriptive-guidance/latest/backup-recovery/new-ebs-volume-backups.html>

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/device_naming.html

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/creating-an-ami-ebs.html>

<https://www.cyberciti.biz/faq/nginx-linux-restart/>

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/deregister-ami.html>

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-deleting-snapshot.html>

<https://docs.aws.amazon.com/cli/latest/userguide/cli-chap-install.html>

<https://docs.aws.amazon.com/cli/latest/userguide/install-cliv2-windows.html#cliv2-windows-install>

<https://docs.aws.amazon.com/cli/latest/userguide/cli-services-ec2-keypairs.html#creating-a-key-pair>

<https://docs.aws.amazon.com/cli/latest/userguide/cli-services-ec2-instances.html>