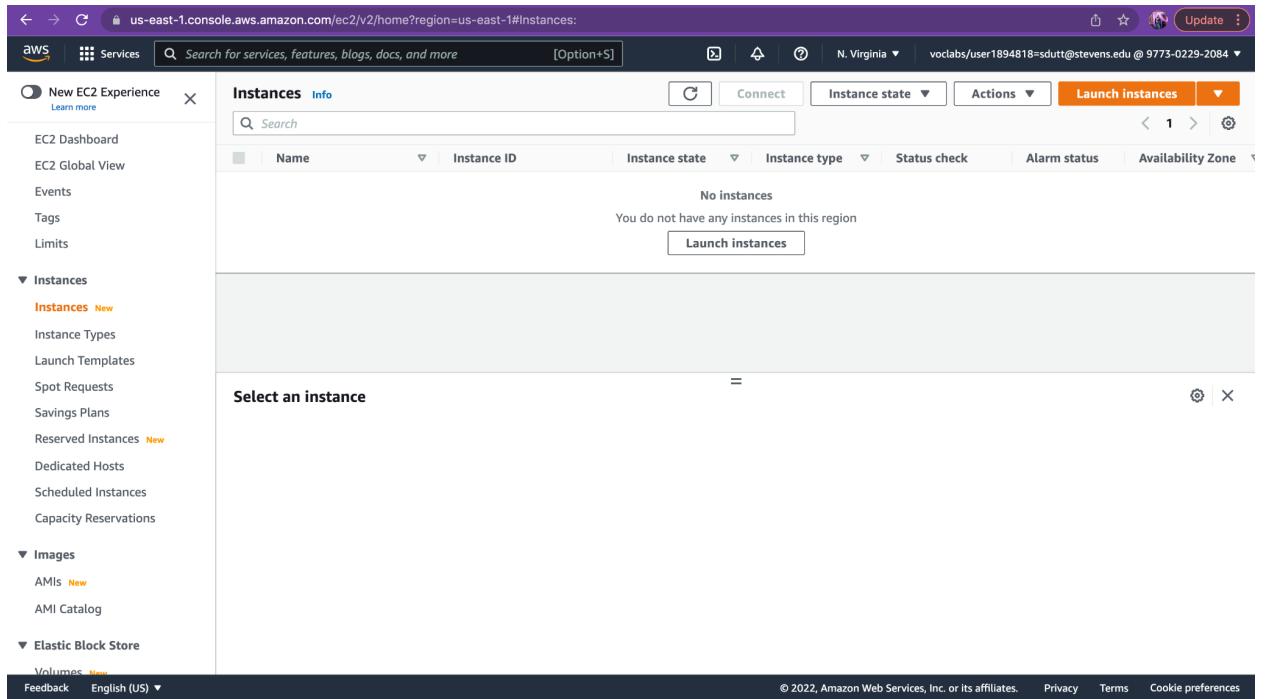


CS- 524 Introduction to Cloud Computing

Lab Assignment-2

Creating Load balancer and servers.

1. Created an instance of Amazon Linux 2 AMI with security group configurations by selecting Launch Instances option.



I selected the Amazon Linux 2 AMI.

You've been invited to try an early, beta iteration of the new launch instance wizard. We will continue to improve the experience over the next few months. We're asking customers for their feedback on this early release. To exit the new launch instance wizard at any time, choose the **Cancel** button.

Step 1: Choose an Amazon Machine Image (AMI)

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

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

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type - ami-0c02fb55956c7d316 (64-bit x86) / ami-03190fe20ef6b1419 (64-bit Arm)
 64-bit (x86)
 64-bit (Arm)

Amazon Linux 2 AMI (HVM) - Kernel 4.14, SSD Volume Type - ami-03e0b06f01d45a4eb (64-bit x86) / ami-018d50b368e796499 (64-bit Arm)
 64-bit (x86)
 64-bit (Arm)

I went on to configure the security group by selecting 6. Configure Security Group.

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

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

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

Cancel Previous Review and Launch Next: Configure Instance Details

I created a new security group and also selected rules ([https](https://), [https](https://)) for the security group load_balancer1.

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	Custom	0.0.0.0/0 e.g. SSH for Admin Desktop
HTTP	TCP	80	Custom	0.0.0.0/0, ::/0 e.g. SSH for Admin Desktop
HTTPS	TCP	443	Custom	0.0.0.0/0, ::/0 e.g. SSH for Admin Desktop

[Add Rule](#)

Warning
Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

[Cancel](#) [Previous](#) [Review and Launch](#)

I selected the launch option.

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 [Edit AMI](#)

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type - ami-0c02fb55956c7d316

Free tier eligible Amazon Linux 2 comes with five years support. It provides Linux kernel 5.10 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 n...

Root Device Type: ebs Virtualization type: hvm

Instance Type [Edit instance type](#)

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

Security Groups [Edit security groups](#)

Security group name	Description
load_balancer1	launch-wizard-1 created 2022-03-29T19:38:23.020-04:00

[Cancel](#) [Previous](#) [Launch](#)

I saved the security RSA and selected Launch Instance.

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

Instance Type

Instance Type	ECUs	vCPUs
t2.micro	-	1

Security Groups

Security group name	Description
load_balancer1	launch-wizard-1 created 2022-03-29T19:38:23.020-04:00

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. Amazon EC2 supports ED25519 and RSA key pair types.

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](#).

I acknowledge that I have access to the corresponding private key file, and that without this file, I won't be able to log into my instance.

Launch Status

Your instances are now launching

The following instance launches have been initiated: i-0d48818283e042de3 [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](#)
- [Amazon EC2: User Guide](#)
- [Learn about AWS Free Usage Tier](#)
- [Amazon EC2: Discussion Forum](#)

While your instances are launching you can also

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

Successfully created load_balancer1 and other servers: server1, server2, server3 and server4 similarly.
Upon connecting to the instance of server 1, I used the following commands :-

- sudo amazon-linux-extras install nginx1
- sudo service nginx start
- cd /etc/nginx/
- sudo vim nginx.conf

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

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Publ
load_balancer	i-09525b7b98a5bbe32	Running	t2.micro	2/2 checks passed	No alarms	us-east-1d	ec2-~
server2	i-0fcf6625bf1313ea	Running	t2.micro	2/2 checks passed	No alarms	us-east-1d	ec2-~
server1	i-05087fc0a6b5ff947	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-~
server4	i-0b4b728013de1f557	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-~
server3	i-0b15d40200dc092bb	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-~

Instance: i-05087fc0a6b5ff947 (server1)

Select an instance above

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

Instance summary

Instance ID	Public IPv4 address	Private IPv4 addresses
i-05087fc0a6b5ff947 (server1)	54.234.132.238 open address	172.31.30.166
IPv6 address	Instance state	Public IPv4 DNS
-	Running	ec2-54-234-132-238.compute-1.amazonaws.com open address
Hostname type	Private IP DNS name (IPv4 only)	
IP name: ip-172-31-30-166.ec2.internal	ip-172-31-30-166.ec2.internal	

Last login: Tue Mar 29 18:00:15 2022 from ec2-18-206-107-26.compute-1.amazonaws.com

```

Amazon Linux 2 AMI

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-30-166 ~]$ sudo amazon-linux-extras install nginx1
Installing nginx
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Cleaning repos: amzn2-core amzn2extra-docker amzn2extra-kernel-5.10 amzn2extra-nginx1
22 metadata files removed
8 sqlite files removed
0 metadata files removed
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
amzn2extra-docker
amzn2extra-kernel-5.10
amzn2extra-nginx1
(1/9): amzn2-core/x86_64/group.gz
(2/9): amzn2-core/x86_64/updateinfo
(3/9): amzn2extra-docker/x86_64/updateinfo
(4/9): amzn2extra-kernel-5.10/x86_64/updateinfo
(5/9): amzn2extra-nginx1/x86_64/updateinfo
(6/9): amzn2extra-docker/x86_64-primary_db
(7/9): amzn2extra-nginx1/x86_64-primary_db
(8/9): amzn2extra-kernel-5.10/x86_64-primary_db
(9/9): amzn2-core/x86_64-primary_db
Package 1:nginx-1.20.0-2.amzn2.0.4.x86_64 already installed and latest version
Nothing to do
  0 ansible2           available  \
    [ =2.4.2  =2.4.6  =2.8  =stable ] 
  2 httpd_modules       available  [ =1.0  =stable ]
  3 memcached1.5        available  \
    [ =1.5.1  =1.5.16  =1.5.17 ] 
  5 postgresql9.6        available  \
    [ =9.6.6  =9.6.8  =stable ] 

```

i-05087fc0a6b5ff947 (server1)

Public IPs: 54.234.132.238 Private IPs: 172.31.30.166

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with various services like EC2 Dashboard, EC2 Global View, Events, Tags, Limits, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Scheduled Instances, Capacity Reservations, Images, AMIs, and Elastic Block Store. The main area displays a table of instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Publ
load_balancer	i-09525b7b98a5bbe32	Running	t2.micro	2/2 checks passed	No alarms	us-east-1d	ec2-~
server2	i-02fc6625bf1313ea	Running	t2.micro	2/2 checks passed	No alarms	us-east-1d	ec2-~
server1	i-05087fc0a6b5ff947	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-~
server4	i-0b4b728013de1f557	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-~
server3	i-0b15d40200dc092bb	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-~

Below the table, a modal window is open for the selected instance 'server1'. It shows the instance summary with fields for Instance ID (i-05087fc0a6b5ff947), Public IPv4 address (54.234.132.238), Private IPv4 addresses (172.31.30.166), Instance state (Running), Public IPv4 DNS (ec2-54-234-132-238.compute-1.amazonaws.com), and Private IP DNS name (IPv4 only) (ip-172-31-30-166.ec2.internal).

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

The screenshot shows the default index.html page served by the nginx web server. The page title is "Website Administrator". The content area contains the text: "This is the default index.html page that is distributed with nginx on Amazon Linux. It is located in /usr/share/nginx/html. You should now put your content in a location of your choice and edit the root configuration directive in the nginx configuration file /etc/nginx/nginx.conf." Below the content, the word "NGINX" is displayed in the green nginx logo font.

I performed the same steps for servers 2,3 and 4 as shown below.

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

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Publ
load_balancer	i-09525b7b98a5bbe32	Running	t2.micro	2/2 checks passed	No alarms	us-east-1d	ec2-~
<input checked="" type="checkbox"/> server2	i-02fc6625bf1313ea	Running	t2.micro	2/2 checks passed	No alarms	us-east-1d	ec2-~
server1	i-05087fc0a6b5ff947	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-~
server4	i-0b4b728013de1f557	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-~
server3	i-0b15d40200dc092bb	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-~

Instance: i-02fc6625bf1313ea (server2)

Instance ID i-02fc6625bf1313ea (server2)	Public IPv4 address 18.206.61.121 open address	Private IPv4 addresses 172.31.93.26
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-18-206-61-121.compute-1.amazonaws.com open address
Hostname type IP name: ip-172-31-93-26.ec2.internal	Private IP DNS name (IPv4 only) ip-172-31-93-26.ec2.internal	Elastic IP addresses -
Answer private resource DNS name IPv4 (A)	Instance type t2.micro	

Feedback English (US) ▾

us-east-1.console.aws.amazon.com/ec2/v2/connect/ec2-user/i-02fc6625bf1313ea

```

Amazon Linux 2 AMI

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-93-26 ~]$ sudo amazon-linux-extras install nginx1
Installing nginx
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Cleaning repos: amzn2-core amzn2extra-docker amzn2extra-kernel-5.10 amzn2extra-nginx1
17 metadata files removed
6 sqlite files removed
0 metadata files removed
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
amzn2extra-docker
amzn2extra-kernel-5.10
amzn2extra-nginx1
(1/9): amzn2-core/2/x86_64/group.gz
(2/9): amzn2-core/2/x86_64/updateinfo
(3/9): amzn2extra-docker/2/x86_64/updateinfo
(4/9): amzn2extra-kernel-5.10/2/x86_64/updateinfo
(5/9): amzn2extra-nginx1/2/x86_64/updateinfo
(6/9): amzn2extra-docker/2/x86_64-primary_db
(7/9): amzn2extra-kernel-5.10/2/x86_64-primary_db
(8/9): amzn2extra-nginx1/2/x86_64-primary_db
(9/9): amzn2-core/2/x86_64-primary_db
Resolving Dependencies
--> Running transaction check
--> Package nginx.x86_64 1:1.20.0-2.amzn2.0.4 will be installed
--> Processing Dependency: nginx-filesystem = 1:1.20.0-2.amzn2.0.4 for package: 1:nginx-1.20.0-2.amzn2.0.4.x86_64
--> Processing Dependency: nginx-filesystem for package: 1:nginx-1.20.0-2.amzn2.0.4.x86_64
--> Processing Dependency: libssl.so.1.1(OPENSSL_1_1_1)(64bit) for package: 1:nginx-1.20.0-2.amzn2.0.4.x86_64
--> Processing Dependency: libssl.so.1.1(OPENSSL_1_1_0)(64bit) for package: 1:nginx-1.20.0-2.amzn2.0.4.x86_64
--> Processing Dependency: libcrypto.so.1.1(OPENSSL_1_1_0)(64bit) for package: 1:nginx-1.20.0-2.amzn2.0.4.x86_64
--> Processing Dependency: libssl.so.1.1()(64bit) for package: 1:nginx-1.20.0-2.amzn2.0.4.x86_64
--> Processing Dependency: libprofiler.so.0()(64bit) for package: 1:nginx-1.20.0-2.amzn2.0.4.x86_64

```

i-02fc6625bf1313ea (server2)

Public IPs: 18.206.61.121 Private IPs: 172.31.93.26

```

29 golang1.11      available  \
[ =1.11.3  =1.11.11  =1.11.13  =stable ]
30 squid4         available  [ =4  =stable ]
32 lustre2.10     available  \
[ =2.10.5  =2.10.8  =stable ]
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 ]
kernel-5.4        available  [ =stable ]
50 selinux-ng      available  [ =stable ]
51 php8.0          available  [ =stable ]
52 tomcat9         available  [ =stable ]
53 unbound1.13     available  [ =stable ]
54 mariadb10.5    available  [ =stable ]
55 kernel-5.10=latest enabled   [ =stable ]
56 redis6          available  [ =stable ]
57 ruby3.0         available  [ =stable ]
58 postgresql12    available  [ =stable ]
59 postgresql13    available  [ =stable ]
60 mock2           available  [ =stable ]
61 dnsmasq         available  [ =stable ]
[ec2-user@ip-172-31-93-26 ~]$ sudo service nginx start
Redirecting to /bin/systemctl start nginx.service
[ec2-user@ip-172-31-93-26 ~]$ 

```

i-02fc6625bf1313ea (server2)

Public IPs: 18.206.61.121 Private IPs: 172.31.93.26

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Publ
load_balancer	i-09525b7b98a5bbe32	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-1d	ec2-.
server2	i-02fc6625bf1313ea	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-1d	ec2-.
server1	i-05087fc0a6b5ff947	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-1a	ec2-.
server4	i-0b4b728013de1f557	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-1a	ec2-.
server3	i-0b15d40200dc092bb	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-1a	ec2-.

Instance: i-02fc6625bf1313ea (server2)

Instance ID	Public IPv4 address	Private IPv4 addresses
i-02fc6625bf1313ea (server2)	18.206.61.121 open address	172.31.93.26
IPv6 address	Instance state	Public IPv4 DNS
-	Running	ec2-18-206-61-121.compute-1.amazonaws.com open address
Hostname type	Private IP DNS name (IPv4 only)	Elastic IP addresses
IP name: ip-172-31-93-26.ec2.internal	ip-172-31-93-26.ec2.internal	-
Answer private resource DNS name	Instance type	
IPv4 (A)	t2.micro	



NGINX

Instances (1/5) [Info](#)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Publ
load_balancer	i-09525b7b98a5bbe32	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-1d	ec2-.
server2	i-02fc6625bf1313ea	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-1d	ec2-.
server1	i-05087fc0a6b5ff947	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-1a	ec2-.
server4	i-0b4b728013de1f557	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-1a	ec2-.
server3	i-0b15d40200dc092bb	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-1a	ec2-.

Instance: i-0b15d40200dc092bb (server3)

[Instance summary](#) [Info](#)

Instance ID i-0b15d40200dc092bb (server3)	Public IPv4 address 54.242.139.37 open address	Private IPv4 addresses 172.31.26.55
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-54-242-139-37.compute-1.amazonaws.com open address
Hostname type IP name: ip-172-31-26-55.ec2.internal	Private IP DNS name (IPv4 only) ip-172-31-26-55.ec2.internal	Elastic IP addresses -
Answer private resource DNS name -	Instance type t2.micro	

```

Amazon Linux 2 AMI

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-26-55 ~]$ sudo amazon-linux-extras install nginx1
Installing nginx
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Cleaning repos: amzn2-core amzn2extra-docker amzn2extra-kernel-5.10 amzn2extra-nginx1
17 metadata files removed
6 sqlite files removed
0 metadata files removed
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
amzn2extra-docker
amzn2extra-kernel-5.10
amzn2extra-nginx1
(1/9): amzn2-core/2/x86_64/group.gz
(2/9): amzn2-core/2/x86_64/updateinfo
(3/9): amzn2extra-docker/2/x86_64/updateinfo
(4/9): amzn2extra-nginx1/2/x86_64/updateinfo
(5/9): amzn2extra-kernel-5.10/2/x86_64/updateinfo
(6/9): amzn2extra-docker/2/x86_64/primary_db
(7/9): amzn2extra-nginx1/2/x86_64/primary_db
(8/9): amzn2extra-kernel-5.10/2/x86_64/primary_db
(9/9): amzn2-core/2/x86_64/primary_db
| 3.7 KB 00:00:00
| 3.0 KB 00:00:00
| 3.0 KB 00:00:00
| 3.0 KB 00:00:00
| 2.5 KB 00:00:00
| 452 KB 00:00:00
| 5.9 KB 00:00:00
| 76 B 00:00:00
| 12 KB 00:00:00
| 86 KB 00:00:00
| 42 KB 00:00:00
| 7.4 MB 00:00:00
| 60 MB 00:00:01

```

i-0b15d40200dc092bb (server3)

Public IPs: 54.242.139.37 Private IPs: 172.31.26.55

```

29 golang1.11      available \
[ =1.11.3  =1.11.11 =1.11.13 =stable ]
30 squid4          available [ =4 =stable ]
32 lustre2.10      available \
[ =2.10.5  =2.10.8 =stable ]
33 java-openjdk11  available [ =11 =stable ]
34 tynis            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 ]
54 mariadb10.5      available [ =stable ]
55 kernel-5.10=latest  enabled [ =stable ]
56 redis6            available [ =stable ]
57 ruby3.0           available [ =stable ]
58 postgresql12     available [ =stable ]
59 postgresql13     available [ =stable ]
60 mock2             available [ =stable ]
61 dnsmasq2.85      available [ =stable ]
[ec2-user@ip-172-31-26-55 ~]$ sudo service nginx start
Redirecting to /bin/systemctl start nginx.service
[ec2-user@ip-172-31-26-55 ~]$ 
```

i-0b15d40200dc092bb (server3)

Public IPs: 54.242.139.37 Private IPs: 172.31.26.55



Screenshot of the AWS EC2 Instances page showing five running t2.micro instances. The instance **server3** (i-0b15d40200dc092bb) is selected.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Publ
load_balancer	i-09525b7b98a5bbe32	Running	t2.micro	2/2 checks passed	No alarms	us-east-1d	ec2-
server2	i-02fcbb6625bf1313ea	Running	t2.micro	2/2 checks passed	No alarms	us-east-1d	ec2-
server1	i-05087fc0a6b5ff947	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2:-
server4	i-0b4b728013de1f557	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2:-
server3	i-0b15d40200dc092bb	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2:-

Instance: i-0b15d40200dc092bb (server3)

Instance summary: **i-0b15d40200dc092bb (server3)**

Instance ID	Public IPv4 address	Private IPv4 addresses
i-0b15d40200dc092bb (server3)	54.242.139.37 open address	172.31.26.55
IPv6 address	Instance state	Public IPv4 DNS
-	Running	ec2-54-242-139-37.compute-1.amazonaws.com open address
Hostname type	Private IP DNS name (IPv4 only)	Elastic IP addresses
IP name: ip-172-31-26-55.ec2.internal	ip-172-31-26-55.ec2.internal	-
Answer private resource DNS name	Instance type	
-	t2.micro	

Feedback English (US) © 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Not Secure ec2-54-242-139-37.compute-1.amazonaws.com

Welcome to nginx on Amazon Linux!

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

Website Administrator

This is the default `index.html` page that is distributed with **nginx** on Amazon Linux. It is located in `/user/share/nginx/html`. You should now put your content in a location of your choice and edit the `root` configuration directive in the **nginx** configuration file `/etc/nginx/nginx.conf`.

NGINX

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

Instances (1/5) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Publ
load_balancer	i-09525b7b98a5bbe32	Running	t2.micro	2/2 checks passed	No alarms	us-east-1d	ec2-.
server2	i-02fc6625bf1313ea	Running	t2.micro	2/2 checks passed	No alarms	us-east-1d	ec2-.
server1	i-05087fc0a6b5ff947	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-.
server4	i-0b4b728013de1f557	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-.
server3	i-0b15d40200dc092bb	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-.

Instance: i-0b4b728013de1f557 (server4)

Select an instance above

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

Instance summary

Instance ID	Public IPv4 address	Private IPv4 addresses
i-0b4b728013de1f557 (server4)	34.228.57.192 open address	172.31.28.190
IPv6 address	Instance state	Public IPv4 DNS
-	Running	ec2-34-228-57-192.compute-1.amazonaws.com open address
Hostname type	Private IP DNS name (IPv4 only)	
IP name: ip-172-31-28-190.ec2.internal	ip-172-31-28-190.ec2.internal	

Feedback English (US) ▾ © 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Stevens-Comput X | Lab Assignment X | W9 Medium Acc X | Learner Lab - Foli X | Untitled document X | Connect to insta X | i-0b4b728013de1f557 X | WhatsApp X | + Update

```
Amazon Linux 2 AMI
https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-28-190 ~]$ sudo amazon-linux-extras install nginx
Installing nginx
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Cleaning repos: amzn2-core amzn2extra-docker amzn2extra-kernel-5.10 amzn2extra-nginx1
17 metadata files removed
6 sqlite files removed
0 metadata files removed
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
amzn2extra-docker
amzn2extra-kernel-5.10
amzn2extra-nginx1
(1/9): amzn2-core/2/x86_64/group_gz
(2/9): amzn2-core/2/x86_64/updateinfo
(3/9): amzn2extra-docker/2/x86_64/updateinfo
(4/9): amzn2extra-kernel-5.10/2/x86_64/primary_db
(5/9): amzn2extra-nginx1/2/x86_64/updateinfo
(6/9): amzn2extra-kernel-5.10/2/x86_64/updateinfo
(7/9): amzn2extra-docker/2/x86_64/primary_db
(8/9): amzn2extra-nginx1/2/x86_64/primary_db
(9/9): amzn2-core/2/x86_64/primary_db
```

i-0b4b728013de1f557 (server4)

Public IPs: 34.228.57.192 Private IPs: 172.31.28.190

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

Instances (1/5) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Publ
load_balancer	i-09525b7b98a5bbe32	Running	t2.micro	2/2 checks passed	No alarms	us-east-1d	ec2-.
server2	i-02fc6625bf1313ea	Running	t2.micro	2/2 checks passed	No alarms	us-east-1d	ec2-.
server1	i-05087fc0a6b5ff947	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-.
server4	i-0b4b728013de1f557	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-.
server3	i-0b15d40200dc092bb	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-.

Instance: i-0b4b728013de1f557 (server4)

Select an instance above

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

Instance summary

Instance ID	Public IPv4 address	Private IPv4 addresses
i-0b4b728013de1f557 (server4)	34.228.57.192 open address	172.31.28.190
IPv6 address	Instance state	Public IPv4 DNS
-	Running	ec2-34-228-57-192.compute-1.amazonaws.com open address
Hostname type	Private IP DNS name (IPv4 only)	
IP name: ip-172-31-28-190.ec2.internal	ip-172-31-28-190.ec2.internal	

Feedback English (US) © 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Not Secure ec2-34-228-57-192.compute-1.amazonaws.com

Welcome to nginx on Amazon Linux!

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

Website Administrator

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

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

nginx

For load balancer also I used the following commands :-

- sudo amazon-linux-extras install nginx1
- sudo service nginx start

Then I used the Public IPv4 DNS and landed on the welcome page.

```
← → C us-east-1.console.aws.amazon.com/ec2/v2/connect/ec2-user/i-09525b7b98a5bbe32
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 ]
54 mariadb10.5        available [ =stable ]
55 kernel-5.10=latest enabled   [ =stable ]
56 redis6             available [ =stable ]
57 ruby3.0             available [ =stable ]
58 postgresql12       available [ =stable ]
59 postgresql13       available [ =stable ]
60 mock2              available [ =stable ]
61 dnsmasq2.85        available [ =stable ]
[ec2-user@ip-172-31-88-138 ~]$
[ec2-user@ip-172-31-88-138 ~]$
[ec2-user@ip-172-31-88-138 ~]$ sudo service nginx start
Redirecting to /bin/systemctl start nginx.service
[ec2-user@ip-172-31-88-138 ~]$
[ec2-user@ip-172-31-88-138 ~]$
[ec2-user@ip-172-31-88-138 ~]$ start nginx.service
-bash: start: command not found
[ec2-user@ip-172-31-88-138 ~]$
[ec2-user@ip-172-31-88-138 ~]$
[ec2-user@ip-172-31-88-138 ~]$

i-09525b7b98a5bbe32 (load_balancer)
Public IP: 44.203.122.133 Private IP: 172.31.88.138
```

After starting the nginx server, I opened the DNS for load_balancer1 from the Instances page and landed on this welcome page.



Configuring Nginx.conf in load balancer.

In order to configure the nginx.conf file I used the following commands :-

- cd /etc/nginx/
- sudo vim nginx.conf

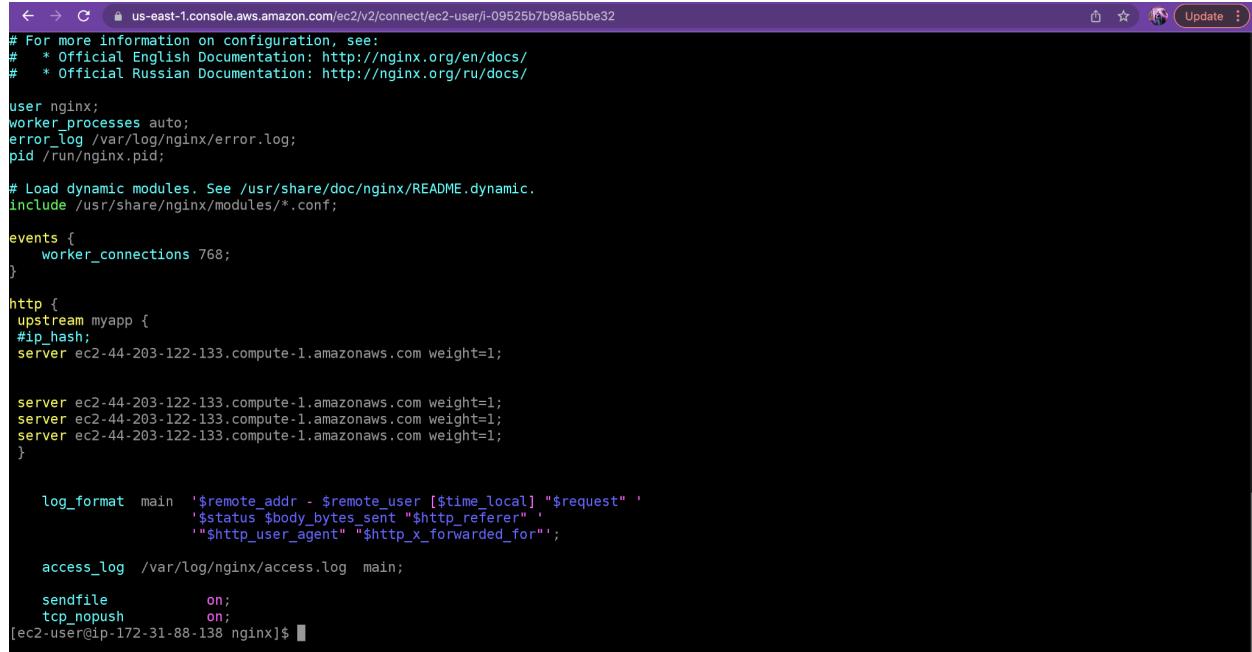
The screenshot shows a terminal window with a purple header bar. The URL in the header is "us-east-1.console.aws.amazon.com/ec2/v2/connect/ec2-user/i-09525b7b98a5bbe32". The terminal content is as follows:

```
Last login: Tue Mar 29 19:11:06 2022 from ec2-18-206-107-26.compute-1.amazonaws.com
[ec2-user@ip-172-31-88-138 ~]$ start nginx1
-bash: start: command not found
[ec2-user@ip-172-31-88-138 ~]$ sudo service nginx1 start
Redirecting to /bin/systemctl start nginx1.service
Failed to start nginx1.service: Unit not found.
[ec2-user@ip-172-31-88-138 ~]$ sudo service nginx start
Redirecting to /bin/systemctl start nginx.service
[ec2-user@ip-172-31-88-138 ~]$ start nginx.service
-bash: start: command not found
[ec2-user@ip-172-31-88-138 ~]$ sudo service nginx start
Redirecting to /bin/systemctl start nginx.service
[ec2-user@ip-172-31-88-138 ~]$ cd /etc/nginx/
[ec2-user@ip-172-31-88-138 nginx]$ sudo vim nginx.conf
```

i-09525b7b98a5bbe32 (load_balancer)

Public IPs: 44.203.122.133 Private IPs: 172.31.88.138

I edited the DNS names and locations of the server and saved the file.



```

# 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 ec2-44-203-122-133.compute-1.amazonaws.com weight=1;
    }

    server ec2-44-203-122-133.compute-1.amazonaws.com weight=1;
    server ec2-44-203-122-133.compute-1.amazonaws.com weight=1;
    server ec2-44-203-122-133.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"';

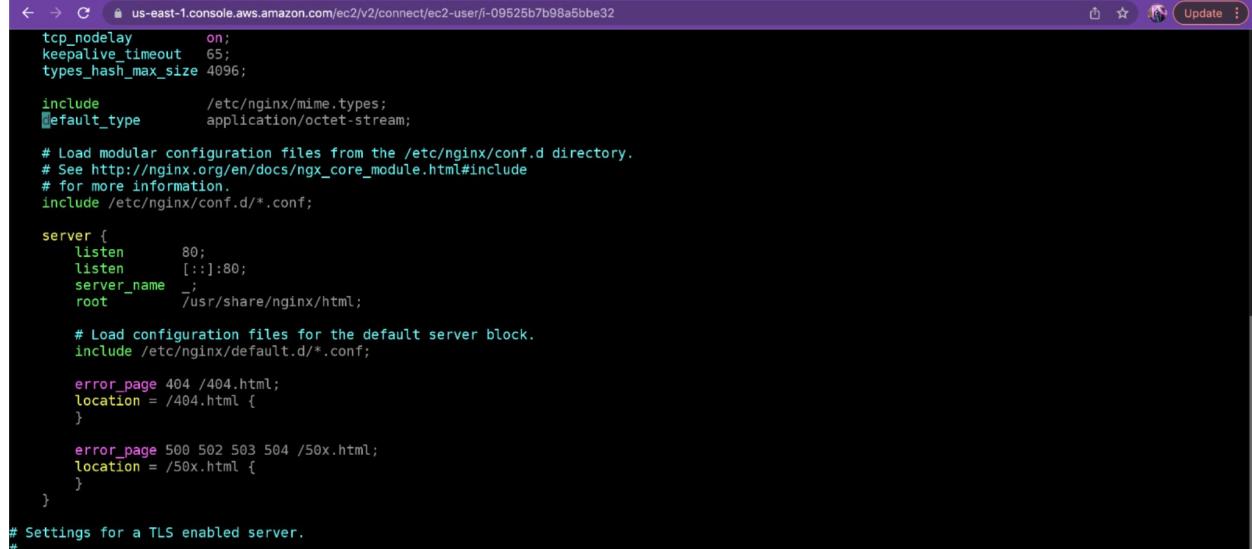
    access_log /var/log/nginx/access.log main;

    sendfile on;
    tcp_nopush on;
}
[ec2-user@ip-172-31-88-138 nginx]$ 

```

i-09525b7b98a5bbe32 (load_balancer)

Public IPs: 44.203.122.133 Private IPs: 172.31.88.138



```

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 ;
    root /usr/share/nginx/html;

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

    error_page 404 /404.html;
    location = /404.html {
    }

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

# Settings for a TLS enabled server.
#

```

i-09525b7b98a5bbe32 (load_balancer)

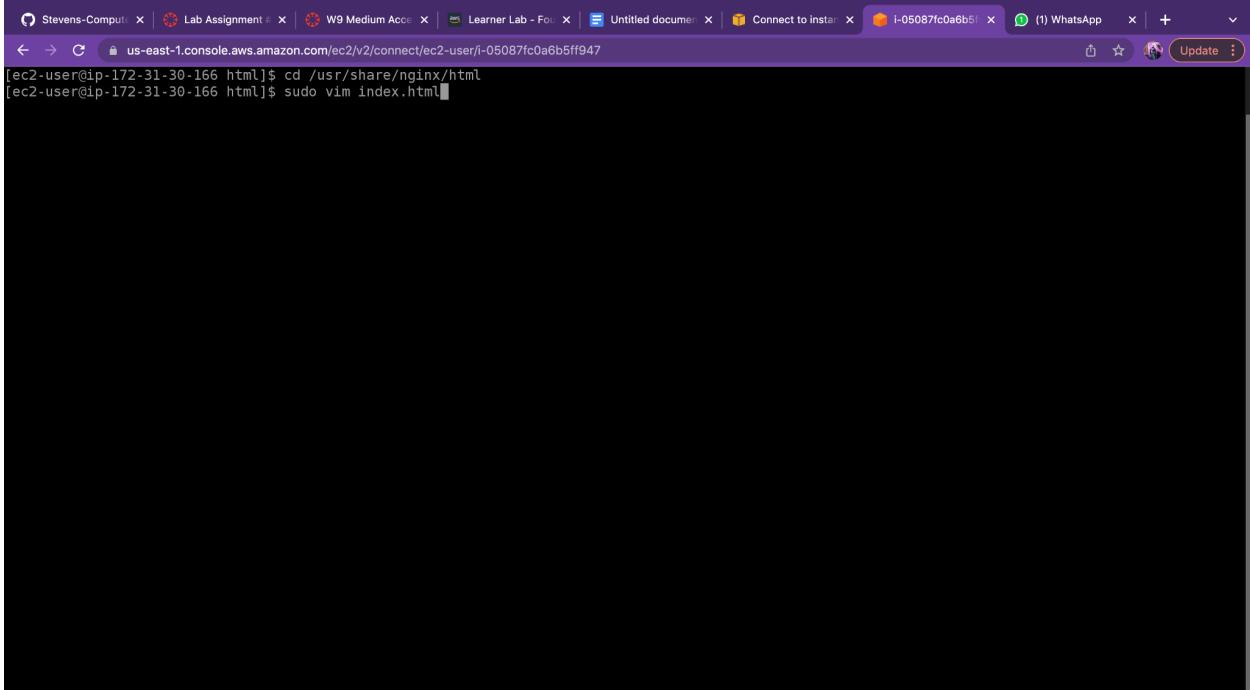
Public IPs: 44.203.122.133 Private IPs: 172.31.88.138

Editing Index.html files of servers

I performed the following commands in order to edit index.html files of servers

- sudo vim index.html
Then I edited the value of the header to server_name.
- sudo systemctl restart nginx

After editing the files I reopened the DNS and could see Server names on Welcome pages.



```
Stevens-Compute | Lab Assignment | W9 Medium Acc... | Learner Lab - Fo... | Untitled document | Connect to insta... | i-05087fc0a6b5 | (1) WhatsApp | + | Update | ← → C | us-east-1.console.aws.amazon.com/ec2/v2/connect/ec2-user/i-05087fc0a6b5ff947 | [ec2-user@ip-172-31-30-166 html]$ cd /usr/share/nginx/html | [ec2-user@ip-172-31-30-166 html]$ sudo vim index.html
```

i-05087fc0a6b5ff947 (server1)

Public IPs: 54.234.132.238 Private IPs: 172.31.30.166

Stevens-Compute | Lab Assignment | W9 Medium Acc | Learner Lab - Fo | Untitled document | Connect to insta | i-05087fc0a6b5ff947 | (1) WhatsApp | + | Update

[us-east-1.console.aws.amazon.com/ec2/v2/connect/ec2-user/i-05087fc0a6b5ff947](#)

```

        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>Server1</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>
            </div>
        </div>
        <div class="logos">
            <a href="http://nginx.net/"></a>
        </div>
    </div>
</body>

```

-- INSERT --

77,7 80%

i-05087fc0a6b5ff947 (server1)

Public IPs: 54.234.132.238 Private IPs: 172.31.30.166

Stevens-Compute | Lab Assignment | W9 Medium Acc | Learner Lab - Fo | Untitled document | Connect to insta | i-05087fc0a6b5ff947 | (1) WhatsApp | + | Update

[us-east-1.console.aws.amazon.com/ec2/v2/connect/ec2-user/i-05087fc0a6b5ff947](#)

```

<body>
    <h1>Welcome to <strong>Server1</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>

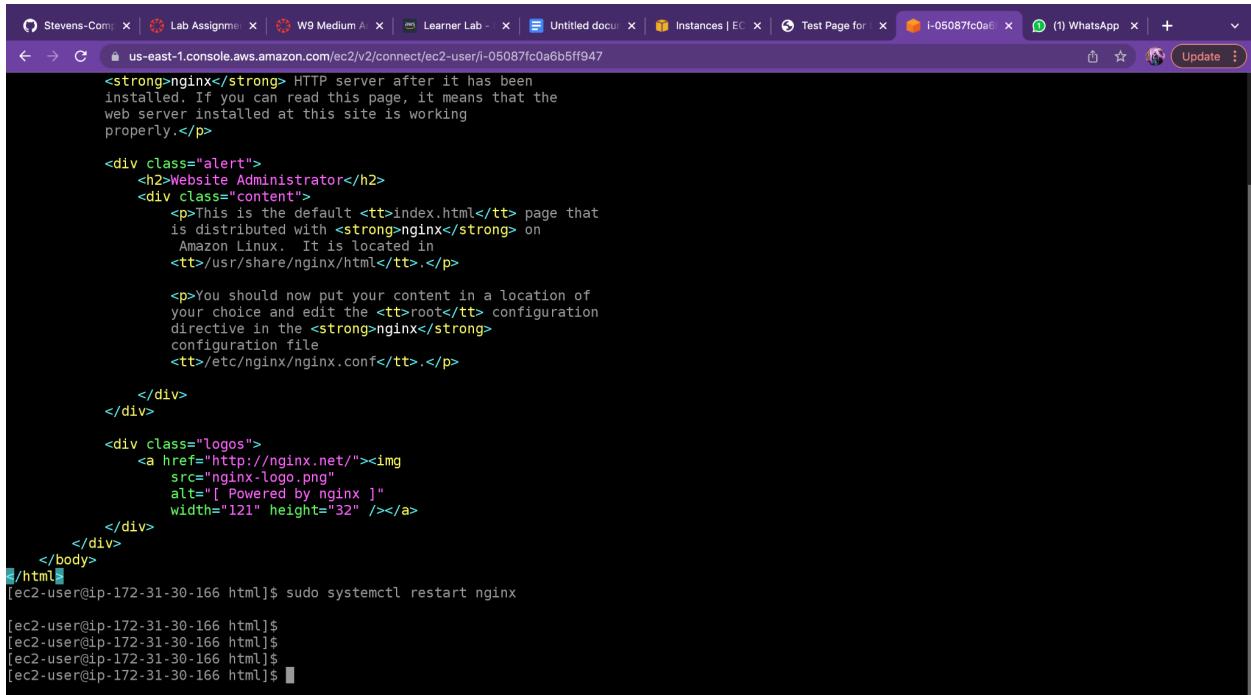
```

"index.html" 111L, 3522B

111,7 Bot

i-05087fc0a6b5ff947 (server1)

Public IPs: 54.234.132.238 Private IPs: 172.31.30.166



The screenshot shows a browser window with multiple tabs open. The active tab is titled "us-east-1.console.aws.amazon.com/ec2/v2/connect/ec2-user/i-05087fc0a6b5ff947". The page content is the standard Nginx welcome message:

```

<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>

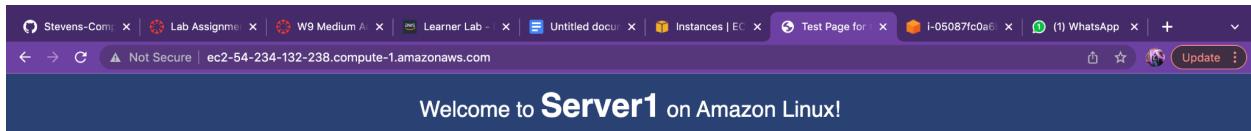
<p>You should now put your content in a location of
your choice and edit the <tt>root</tt> configuration
directive in the <strong>nginx</strong>
configuration file
<tt>/etc/nginx/nginx.conf</tt>.</p>

```

The terminal below shows the command `sudo systemctl restart nginx` being run.

i-05087fc0a6b5ff947 (server1)

Public IPs: 54.234.132.238 Private IPs: 172.31.30.166



Website Administrator

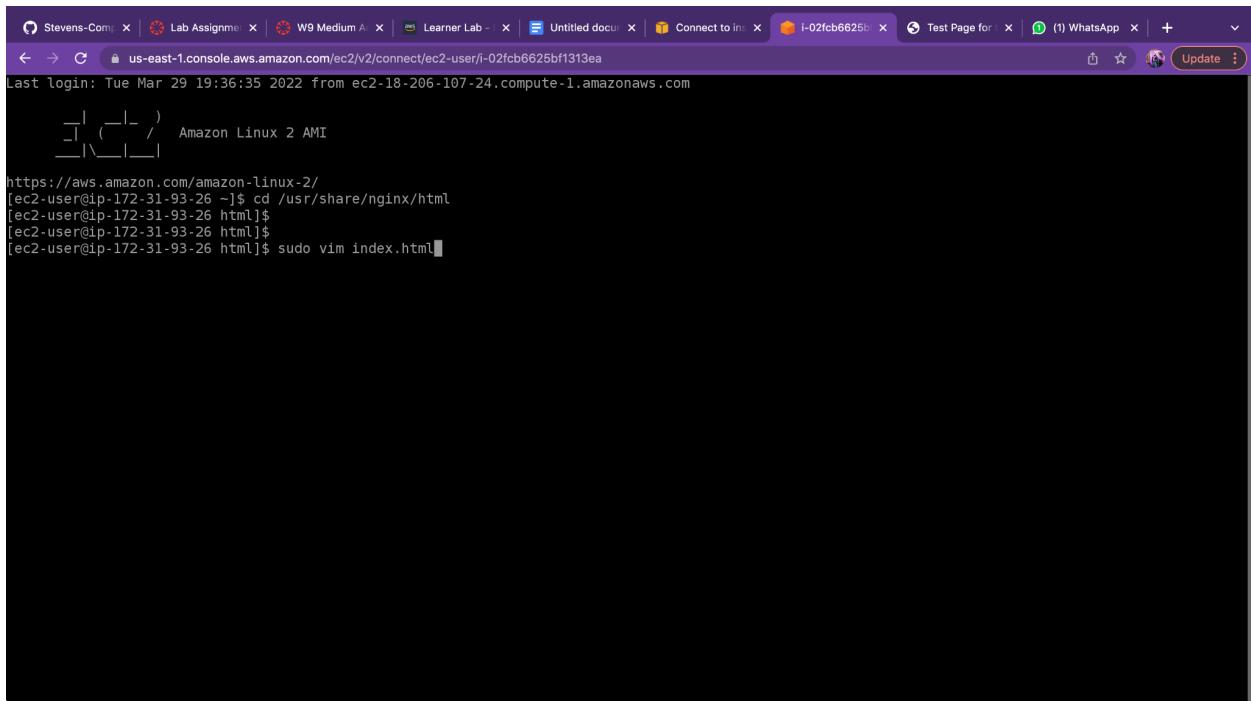
This is the default `index.html` page that is distributed with **nginx** on Amazon Linux. It is located in `/usr/share/nginx/html`.

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

NGINX



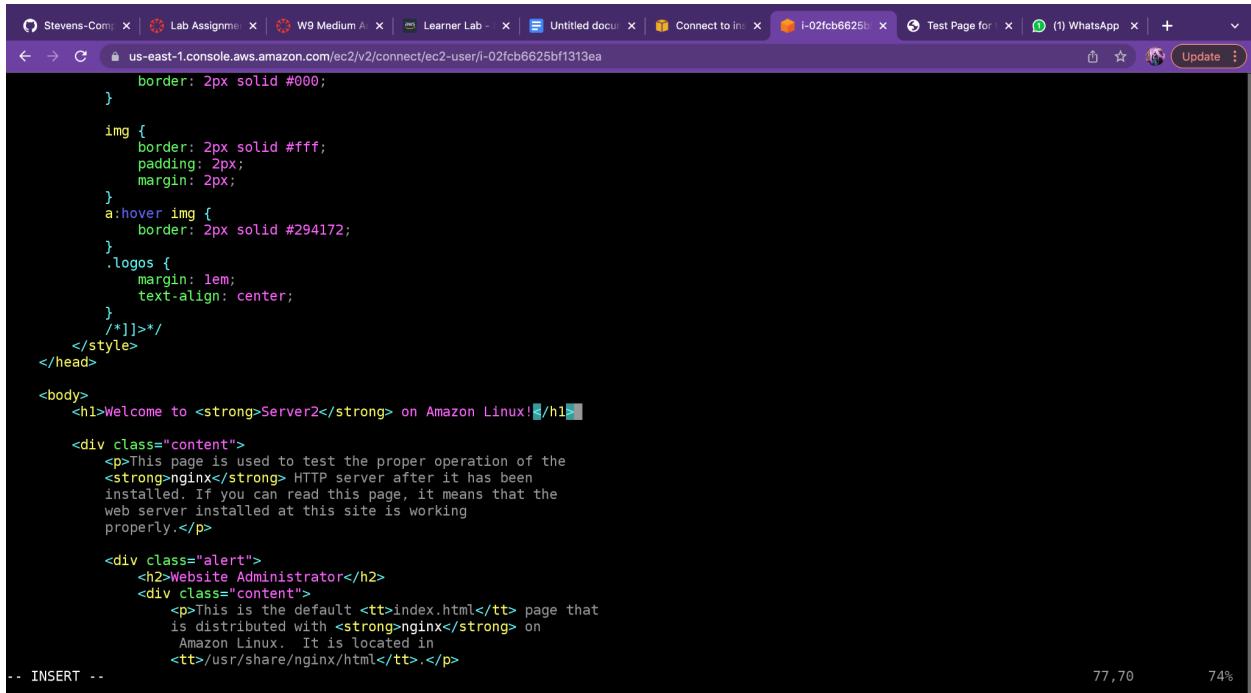
I repeated the steps for Server 2,3 and4.



```
Last login: Tue Mar 29 19:36:35 2022 from ec2-18-206-107-24.compute-1.amazonaws.com
[ec2-user@ip-172-31-93-26 ~]$ cd /usr/share/nginx/html
[ec2-user@ip-172-31-93-26 html]$ 
[ec2-user@ip-172-31-93-26 html]$ sudo vim index.html
```

i-02fcb6625bf1313ea (server2)

Public IPs: 18.206.61.121 Private IPs: 172.31.93.26



```
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>Server2</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>
-- INSERT --
```

i-02fcb6625bf1313ea (server2)

Public IPs: 18.206.61.121 Private IPs: 172.31.93.26

```

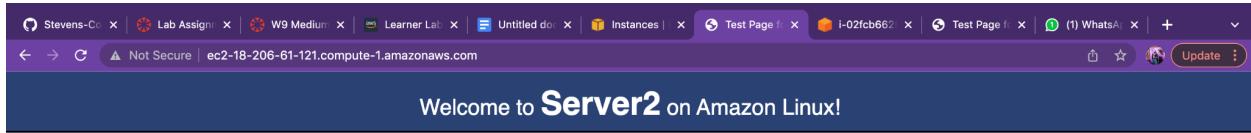
    border: 2px solid #fff;
    padding: 2px;
    margin: 2px;
}
a:hover img {
    border: 2px solid #294172;
}
.logos {
    margin: 1em;
    text-align: center;
}
/*]]> */
</style>
</head>

<body>
    <h1>Welcome to <strong>Server2</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>
        "index.html" 11L, 3522B written
[ec2-user@ip-172-31-93-26 html]$ sudo systemctl restart nginx
[ec2-user@ip-172-31-93-26 html]$ 
[ec2-user@ip-172-31-93-26 html]$ 
[ec2-user@ip-172-31-93-26 html]$ 

```

i-02fc6625bf1313ea (server2)

Public IPs: 18.206.61.121 Private IPs: 172.31.93.26



NGINX

Last login: Tue Mar 29 19:39:49 2022 from ec2-18-206-107-24.compute-1.amazonaws.com

```
[ec2-user@ip-172-31-26-55 ~]$ cd /usr/share/nginx/html
[ec2-user@ip-172-31-26-55 html]$ 
[ec2-user@ip-172-31-26-55 html]$ 
[ec2-user@ip-172-31-26-55 html]$ sudo vim index.html
```

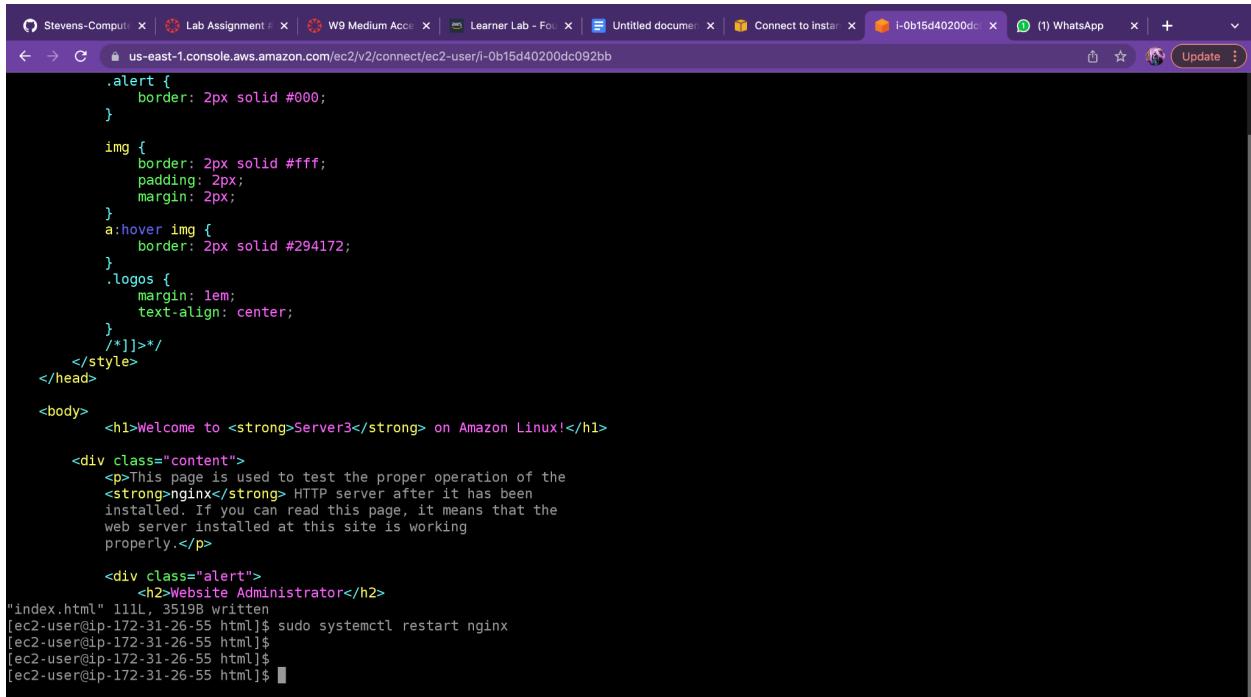
i-0b15d40200dc092bb (server3)

Public IPs: 54.242.139.37 Private IPs: 172.31.26.55

```
<head>
  <style>
    .content {
      padding: 1em 5em;
    }
    .alert {
      border: 2px solid #000;
    }
    img {
      border: 2px solid #fff;
      padding: 2px;
      margin: 2px;
    }
    a:hover img {
      border: 2px solid #294172;
    }
    .logos {
      margin: 1em;
      text-align: center;
    }
  /*]]>
</style>
</head>
<body>
  <h1>Welcome to <strong>Server3</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>
  ... INSERT ...
  77, 36-43   68%
```

i-0b15d40200dc092bb (server3)

Public IPs: 54.242.139.37 Private IPs: 172.31.26.55



The screenshot shows a browser window with the URL `us-east-1.console.aws.amazon.com/ec2/v2/connect/ec2-user/i-0b15d40200dc092bb`. The page displays the source code of the `index.html` file. The code includes CSS styles for an alert box, an img tag, and a hover effect. It also contains HTML for a header and a content section. At the bottom, there is a terminal session showing the command `sudo systemctl restart nginx` being run.

```

.alert {
    border: 2px solid #000;
}

img {
    border: 2px solid #fff;
    padding: 2px;
    margin: 2px;
}

a:hover img {
    border: 2px solid #294172;
}

.logos {
    margin: 1em;
    text-align: center;
}

/*]]>*/

</style>
</head>

<body>
    <h1>Welcome to <strong>Server3</strong> on Amazon Linux!</h1>

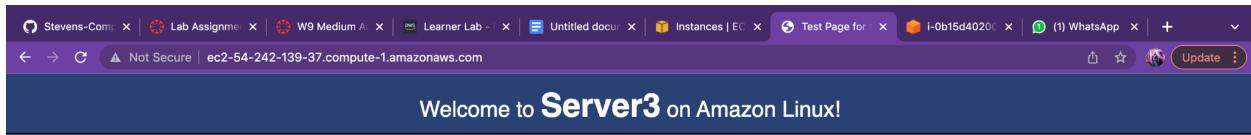
    <div class="content">
        <p>This page is used to test the proper operation of the
        <strong>nginx</strong> HTTP server after it has been
        installed. If you can read this page, it means that the
        web server installed at this site is working
        properly.</p>
    </div>
    <div class="alert">
        <h2>Website Administrator</h2>

```

"index.html" 111L, 3519B written
[ec2-user@ip-172-31-26-55 html]\$ sudo systemctl restart nginx
[ec2-user@ip-172-31-26-55 html]\$
[ec2-user@ip-172-31-26-55 html]\$
[ec2-user@ip-172-31-26-55 html]\$

i-0b15d40200dc092bb (server3)

Public IPs: 54.242.139.37 Private IPs: 172.31.26.55



NGINX

```

!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: #fff;
    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 {
}
<![CDATA]]>
</style>
</head>
<body>
<h1>index.html 111L 3520B</h1>
</body>

```

1,1 Top

i-0b4b728013de1f557 (server4)

Public IPs: 34.228.57.192 Private IPs: 172.31.28.190

```

<head>
<style type="text/css">
    display: none;
}
.content {
    padding: 1em 5em;
}
.alert {
    border: 2px solid #000;
}

img {
    border: 2px solid #fff;
    padding: 2px;
    margin: 2px;
}
a:hover img {
    border: 2px solid #294172;
}
.logos {
    margin: 1em;
    text-align: center;
}
/*]]>/*
</style>
</head>
<body>
<h1>Welcome to <strong>Server4</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">
-- INSERT --

```

77,39 66%

i-0b4b728013de1f557 (server4)

Public IPs: 34.228.57.192 Private IPs: 172.31.28.190

```

        }
        .alert {
            border: 2px solid #000;
        }

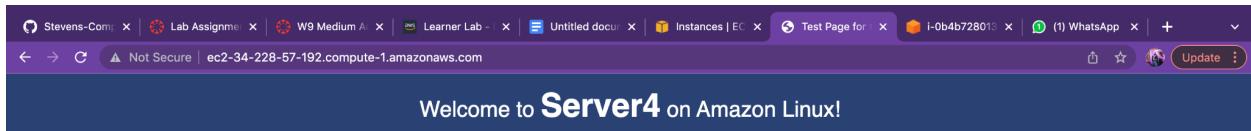
        img {
            border: 2px solid #fff;
            padding: 2px;
            margin: 2px;
        }
        a:hover img {
            border: 2px solid #294172;
        }
        .logos {
            margin: 1em;
            text-align: center;
        }
        /*]]> */
    </style>
</head>

<body>
    <h1>Welcome to <strong>Server4</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">
        "index.html" 111L, 3522B written
[ec2-user@ip-172-31-28-190 html]$ sudo systemctl restart nginx
[ec2-user@ip-172-31-28-190 html]$
[ec2-user@ip-172-31-28-190 html]$
[ec2-user@ip-172-31-28-190 html]$ 

```

i-0b4b728013de1f557 (server4)

Public IPs: 34.228.57.192 Private IPs: 172.31.28.190



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

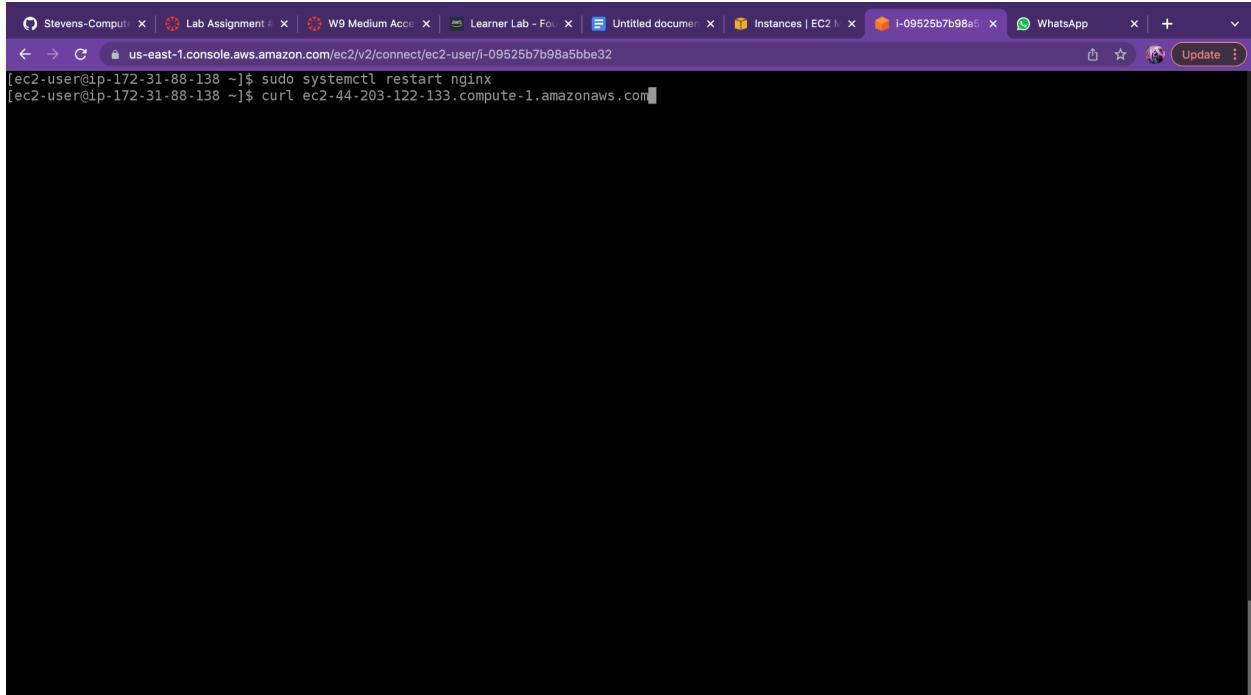
Website Administrator
<p>This is the default <code>index.html</code> page that is distributed with nginx on Amazon Linux. It is located in <code>/usr/share/nginx/html</code>. You should now put your content in a location of your choice and edit the <code>root</code> configuration directive in the nginx configuration file <code>/etc/nginx/nginx.conf</code>.</p>

NGINX

Curl Commands

I used the following Curl command

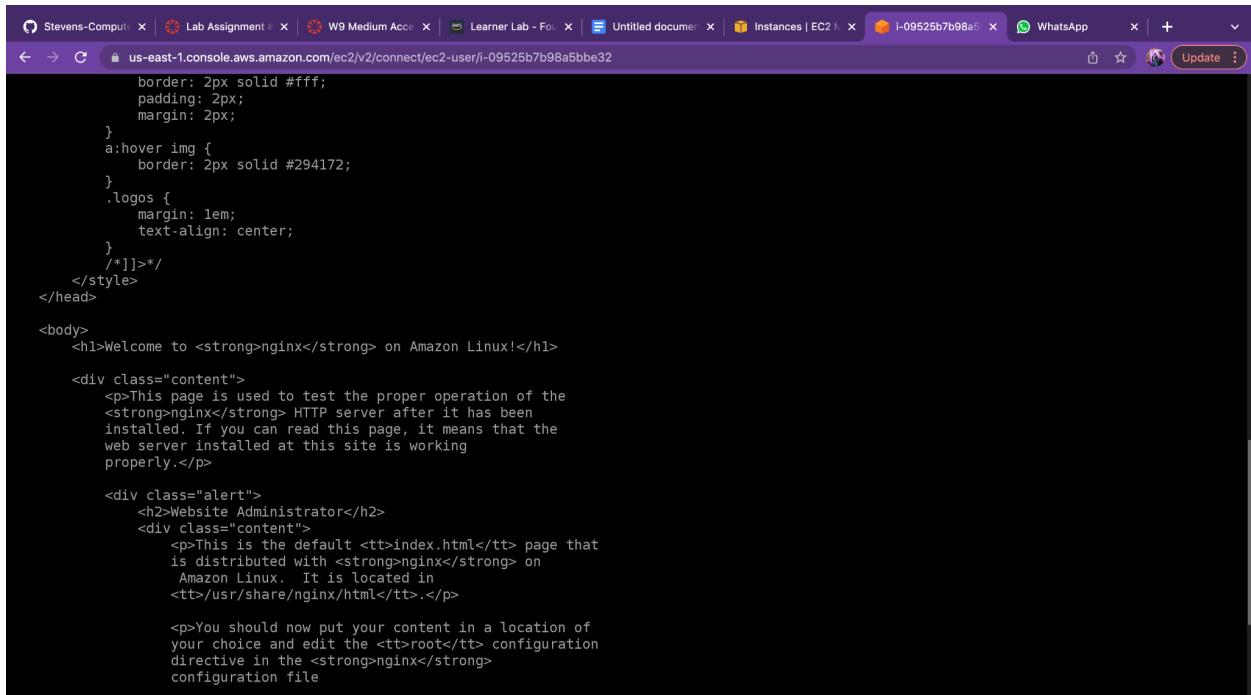
Curl DNS-name for load balancer and all the servers.



```
[ec2-user@ip-172-31-88-138 ~]$ sudo systemctl restart nginx
[ec2-user@ip-172-31-88-138 ~]$ curl ec2-44-203-122-133.compute-1.amazonaws.com
```

i-09525b7b98a5bbe32 (load_balancer)

Public IPs: 44.203.122.133 Private IPs: 172.31.88.138



```
<html>
    <head>
        <style>
            a {
                border: 2px solid #fff;
                padding: 2px;
                margin: 2px;
            }
            a:hover img {
                border: 2px solid #294172;
            }
            .logos {
                margin: 1em;
                text-align: center;
            }
        /*]]>/*
    </style>
</head>
<body>
    <h1>Welcome to <strong>nginx</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</p>
            </div>
        </div>
    </div>
</body>
</html>
```

i-09525b7b98a5bbe32 (load_balancer)

Public IPs: 44.203.122.133 Private IPs: 172.31.88.138

```
↳ Stevens-Comp X | Lab Assignment X | W9 Medium A X | Learner Lab - X | Untitled document X | Instances | EC X | Test Page for i-09525b7b98a5bbe32 X | i-09525b7b98a5bbe32 X | WhatsApp X | + | Update :  
← → C | us-east-1.console.aws.amazon.com/ec2/v2/connect/ec2-user/i-09525b7b98a5bbe32  
<body>  
  <h1>Welcome to <strong>Server1</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-172-31-88-138 nginx]$
```

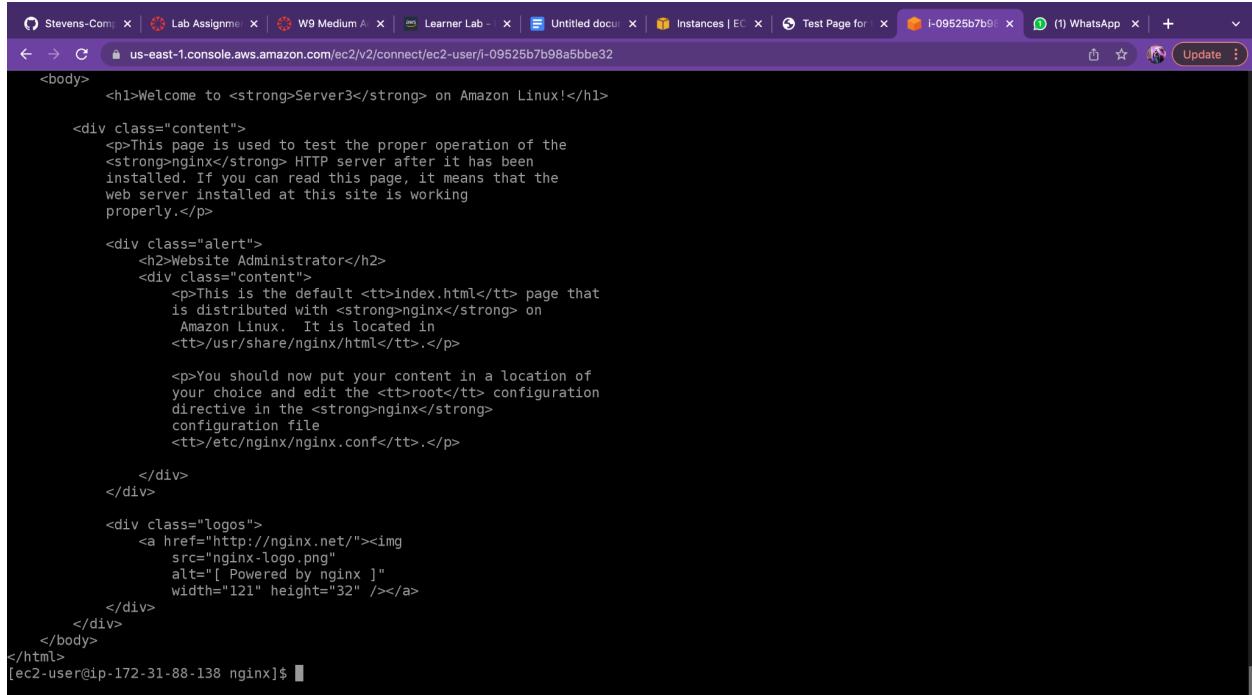
i-09525b7b98a5bbe32 (load_balancer)

Public IPs: 44.203.122.133 Private IPs: 172.31.88.138

```
↳ Stevens-Comp X | Lab Assignment X | W9 Medium A X | Learner Lab - X | Untitled document X | Instances | EC X | Test Page for i-09525b7b98a5bbe32 X | i-09525b7b98a5bbe32 X | WhatsApp X | + | Update :  
← → C | us-east-1.console.aws.amazon.com/ec2/v2/connect/ec2-user/i-09525b7b98a5bbe32  
<body>  
  <h1>Welcome to <strong>Server2</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-172-31-88-138 nginx]$
```

i-09525b7b98a5bbe32 (load_balancer)

Public IPs: 44.203.122.133 Private IPs: 172.31.88.138



The screenshot shows a web browser window with multiple tabs open. The active tab displays the default Nginx configuration page for an EC2 instance. The content of the page is as follows:

```

<body>
    <h1>Welcome to <strong>Server3</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>

    <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>

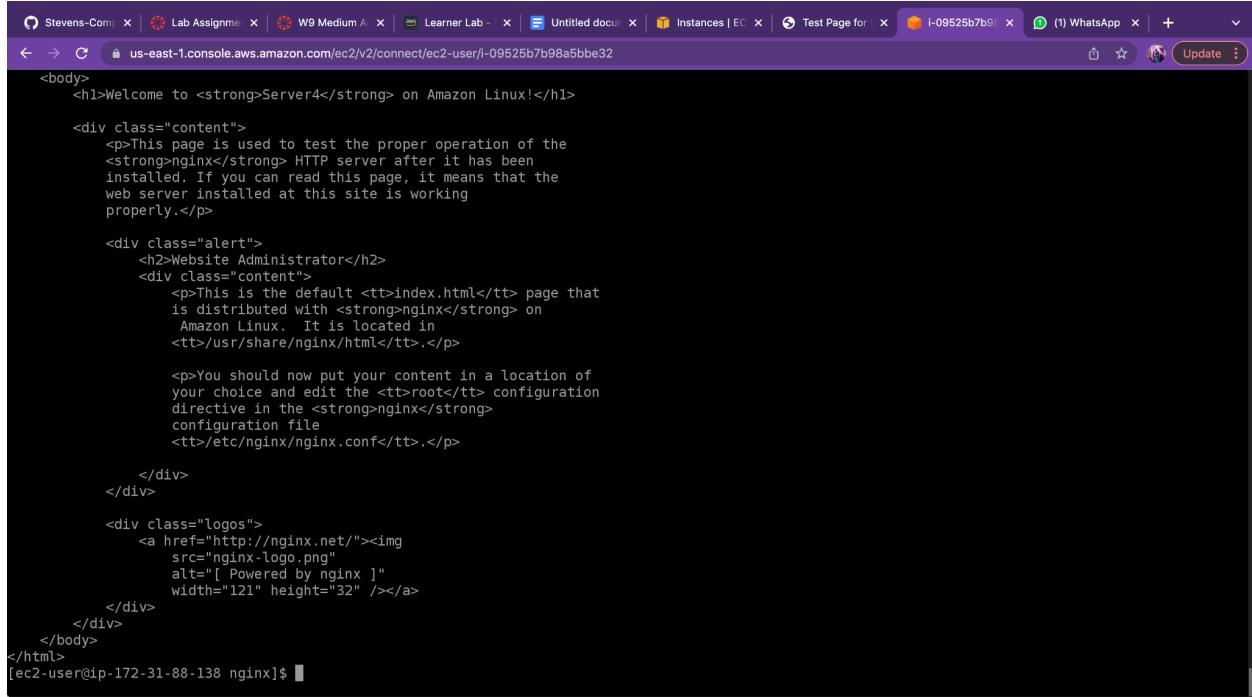
    <div class="logos">
        <a href="http://nginx.net/"></a>
    </div>
</body>

```

[ec2-user@ip-172-31-88-138 nginx]\$

i-09525b7b98a5bbe32 (load_balancer)

Public IPs: 44.203.122.133 Private IPs: 172.31.88.138



The screenshot shows a web browser window with multiple tabs open. The active tab displays the default Nginx configuration page for an EC2 instance. The content of the page is as follows:

```

<body>
    <h1>Welcome to <strong>Server4</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>

    <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>
</body>

```

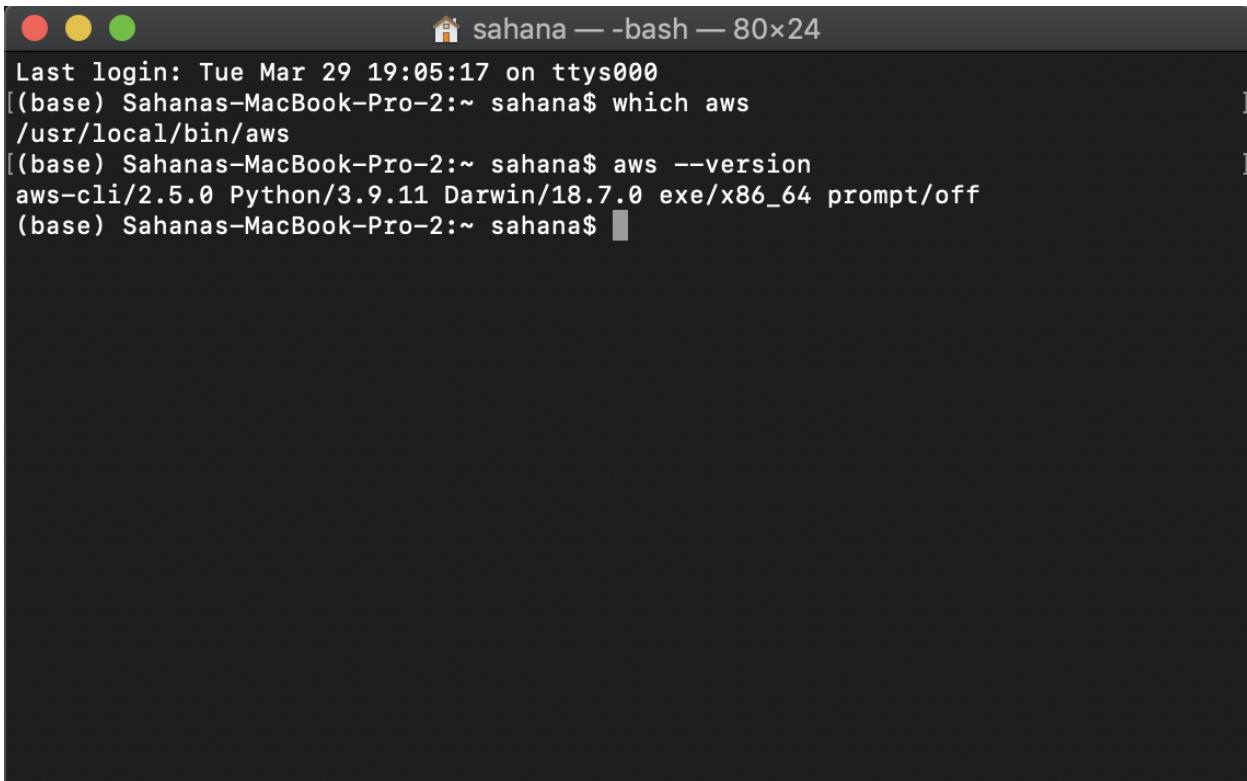
[ec2-user@ip-172-31-88-138 nginx]\$

i-09525b7b98a5bbe32 (load_balancer)

Public IPs: 44.203.122.133 Private IPs: 172.31.88.138

Creating an instance using AWS CLI

I downloaded aws cli and installed it.



```
Last login: Tue Mar 29 19:05:17 on ttys000
[(base) Sahanas-MacBook-Pro-2:~ sahana$ which aws
/usr/local/bin/aws
[(base) Sahanas-MacBook-Pro-2:~ sahana$ aws --version
aws-cli/2.5.0 Python/3.9.11 Darwin/18.7.0 exe/x86_64 prompt/off
(base) Sahanas-MacBook-Pro-2:~ sahana$
```

I configured the aws credential file



```
Last login: Tue Mar 29 19:02:51 on ttys000
(base) Sahanas-MacBook-Pro-2:~ sahana$ curl "https://awscli.amazonaws.com/AWSCLIV2.pkg" -o "AWSCLIV2.pkg"
(base) Sahanas-MacBook-Pro-2:~ sahana$ sudo installer -pkg AWSCLIV2.pkg -target /
Password:
installer: Package name is AWS Command Line Interface
installer: Upgrading at base path
installer: The upgrade was successful.
(base) Sahanas-MacBook-Pro-2:~ sahana$ curl "https://awscli.amazonaws.com/AWSCLIV2.pkg" -o "AWSCLIV2.pkg"
(base) Sahanas-MacBook-Pro-2:~ sahana$ sudo installer -pkg ./AWSCLIV2.pkg -target /
installer: Package name is AWS Command Line Interface
installer: Upgrading at base path
installer: The upgrade was successful.
(base) Sahanas-MacBook-Pro-2:~ sahana$ which aws
/usr/local/bin/aws
(base) Sahanas-MacBook-Pro-2:~ sahana$ aws --version
aws-cli/2.5.0 Python/3.9.11 Darwin/18.7.0 exe/x86_64 prompt/off
(base) Sahanas-MacBook-Pro-2:~ sahana$ aws configure
AWS Access Key ID [None]: ASIA4H6C6QZBHK5D5T7
AWS Secret Access Key [None]: ZWz23YNKwrg8sC9Qj/Mtc/ebq5CpZlSN4ynJlyit
aws_session_token=[None]: Vw202XtVYkdxEgcaOke2uT2dBXSRuSyg+SLCAQr7y0FT7R0iWzgwPTZPIV+qv5ao+XoUtmtTwjAGiBVPhipSHL4NxywlEBaj18ML75bh6v1aqhrpifElmBwMmoxe10ww4oFnjrgL4BKQnJ0ti75D6Wtf/6gCsqkFnSZPu/80Leruw5DgmHcbPLvsP
Ra6hZBY5U9ta057k4/WTJct2ejz90OpWPMDiR15/JN7hf4ykhEmwgBQ04wsIC+6a8XwJ3qkS1EVxZjegNBee7ncXc6G4EHM11CwAn0mibKpZ/jZlOM1tnjdFTJds4Vwxr0GGSnEgJ/dm7Umnz9nktdt6YjE3e94fn2ra8t0EuUcs=Default region name [a
ws_session_token=w02XtVYkdxEgcaOke2uT2dBXSRuSyg+SLCAQr7y0FT7R0iWzgwPTZPIV+qv5ao+XoUtmtTwjAGiBVPhipSHL4NxywlEBaj18ML75bh6v1aqhrpifElmBwMmoxe10ww4oFnjrgL4BKQnJ0ti75D6Wtf/6gCsqkFnSZPu/80Leruw5DgmHcbPLvsP
adzhZBY5U9ta057k4/WTJct2ejz90OpWPMDiR15/JN7hf4ykhEmwgBQ04wsIC+6a8XwJ3qkS1EVxZjegNBee7ncXc6G4EHM11CwAn0mibKpZ/jZlOM1tnjdFTJds4Vwxr0GGSnEgJ/dm7Umnz9nktdt6YjE3e94fn2ra8t0EuUcs=us-east-1]
Default output format [None]: json
(base) Sahanas-MacBook-Pro-2:~ sahana$
```

After which I created the instance.

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with various navigation options like New EC2 Experience, EC2 Dashboard, Events, Tags, Limits, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Scheduled Instances, Capacity Reservations, Images, AMIs, AMI Catalog, and Elastic Block Store. The main content area has a header 'Instances (1/1) Info' with a search bar and buttons for Connect, Instance state, Actions, and Launch instances. Below this is a table with one row for 'Instance_cli'. The table columns include Name, Instance ID, Instance state, Instance type, Status check, Alarm status, and Availability Zone. The instance details page below shows the instance summary for 'i-00031ad7060ec5d7d (Instance_cli)' with tabs for Details, Security, Networking, Storage, Status checks, Monitoring, and Tags. Under Details, it shows Instance ID (i-00031ad7060ec5d7d), Public IPv4 address (50.16.98.110), Private IPv4 addresses (172.31.81.202), Instance state (Running), Public IPv4 DNS (ec2-50-16-98-110.compute-1.amazonaws.com), and Hostname type (IP name: ip-172-31-81-202.ec2.internal).

Ruby Installation and visit_server.rb creation

Next I installed ruby using sudo yum install ruby command.

I created the visit_server.rb file using vim visit_Server.rb

Then I ran the following command :-

```
ruby visit_server.rb -d ec2-44-203-122-133.compute-1.amazonaws.com
```

The screenshot shows a terminal window with the title "us-east-1.console.aws.amazon.com/ec2/v2/connect/ec2-user/i-09525b7b98a5bbe32". The terminal output is as follows:

```
Amazon Linux 2 AMI

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-88-138 ~]$ ruby visit_server.rb ec2-54-234-132-238.compute-1.amazonaws.com
-bash: ruby: command not found
[ec2-user@ip-172-31-88-138 ~]$ sudo yum install ruby
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
Resolving Dependencies
--> Running transaction check
--> Package ruby.x86_64 0:2.0.0.648-36.amzn2.0.2 will be installed
--> Processing Dependency: ruby-libs(x86-64) = 2.0.0.648-36.amzn2.0.2 for package: ruby-2.0.0.648-36.amzn2.0.2.x86_64
--> Processing Dependency: rubygem/bigdecimal >= 1.2.0 for package: ruby-2.0.0.648-36.amzn2.0.2.x86_64
--> Processing Dependency: ruby(rubygems) >= 2.0.14.1 for package: ruby-2.0.0.648-36.amzn2.0.2.x86_64
--> Processing Dependency: libruby.so.2.0()(64bit) for package: ruby-2.0.0.648-36.amzn2.0.2.x86_64
--> Running transaction check
--> Package ruby-libs.x86_64 0:2.0.0.648-36.amzn2.0.2 will be installed
--> Package rubygem-bigdecimal.x86_64 0:1.2.0-36.amzn2.0.2 will be installed
--> Package rubygems.noarch 0:2.0.14.1-36.amzn2.0.2 will be installed
--> Processing Dependency: rubygem(rdoc) >= 4.0.0 for package: rubygems-2.0.14.1-36.amzn2.0.2.noarch
--> Processing Dependency: rubygem(psych) >= 2.0.0 for package: rubygems-2.0.14.1-36.amzn2.0.2.noarch
--> Processing Dependency: rubygem(io-console) >= 0.4.2 for package: rubygems-2.0.14.1-36.amzn2.0.2.noarch
--> Running transaction check
--> Package rubygem-io-console.x86_64 0:0.4.2-36.amzn2.0.2 will be installed
--> Package rubygem-psych.x86_64 0:2.0.0-36.amzn2.0.2 will be installed
--> Package rubygem-rdoc.noarch 0:4.0.0-36.amzn2.0.2 will be installed
--> Processing Dependency: ruby(irb) = 2.0.0.648 for package: rubygem-rdoc-4.0.0-36.amzn2.0.2.noarch
--> Processing Dependency: rubygem(json) >= 1.7.7 for package: rubygem-rdoc-4.0.0-36.amzn2.0.2.noarch
--> Running transaction check
--> Package ruby-irb.noarch 0:2.0.0.648-36.amzn2.0.2 will be installed
--> Package rubygem-json.x86_64 0:1.7.7-36.amzn2.0.2 will be installed
--> Finished Dependency Resolution

Dependencies Resolved
```

i-09525b7b98a5bbe32 (load_balancer)

Public IPs: 44.203.122.133 Private IPs: 172.31.88.138

```

#!/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
"visit_server.rb" 85L, 2047B
21,2 Top

```

i-09525b7b98a5bbe32 (load_balancer)

Public IPs: 44.203.122.133 Private IPs: 172.31.88.138

The weights for the different runs were as per the given table.

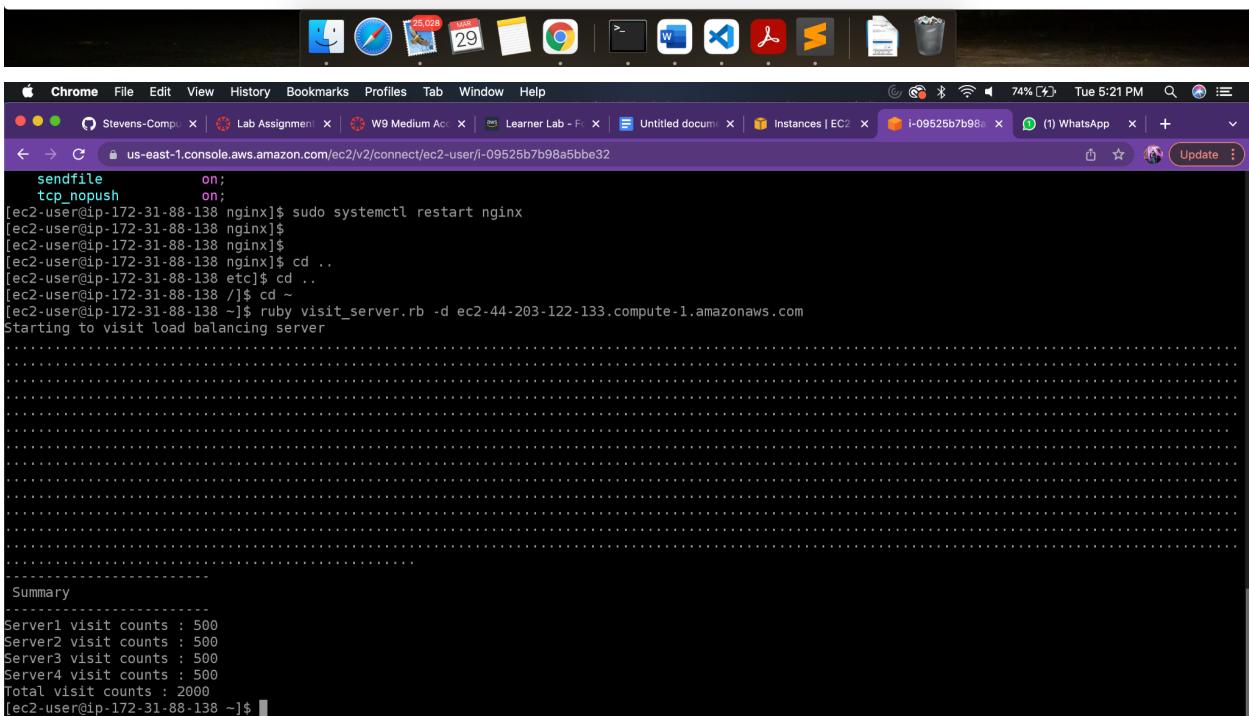
Weight \ Server Scenario	Server 1	Server 2	Server 3	Server 4
1	1	1	1	1
2	1	2	3	4
3	1	2	1	2

The weights can be seen in the nginx.conf file and the result can be seen in the console.

```
# 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 ec2-54-234-132-238.compute-1.amazonaws.com weight=1;  
    server ec2-18-206-61-121.compute-1.amazonaws.com weight=1;  
    server ec2-54-242-139-37.compute-1.amazonaws.com weight=1;  
    server ec2-34-228-57-192.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;  
fec2-user@ip-172-31-88-138:~$ sudo systemctl restart nginx
```

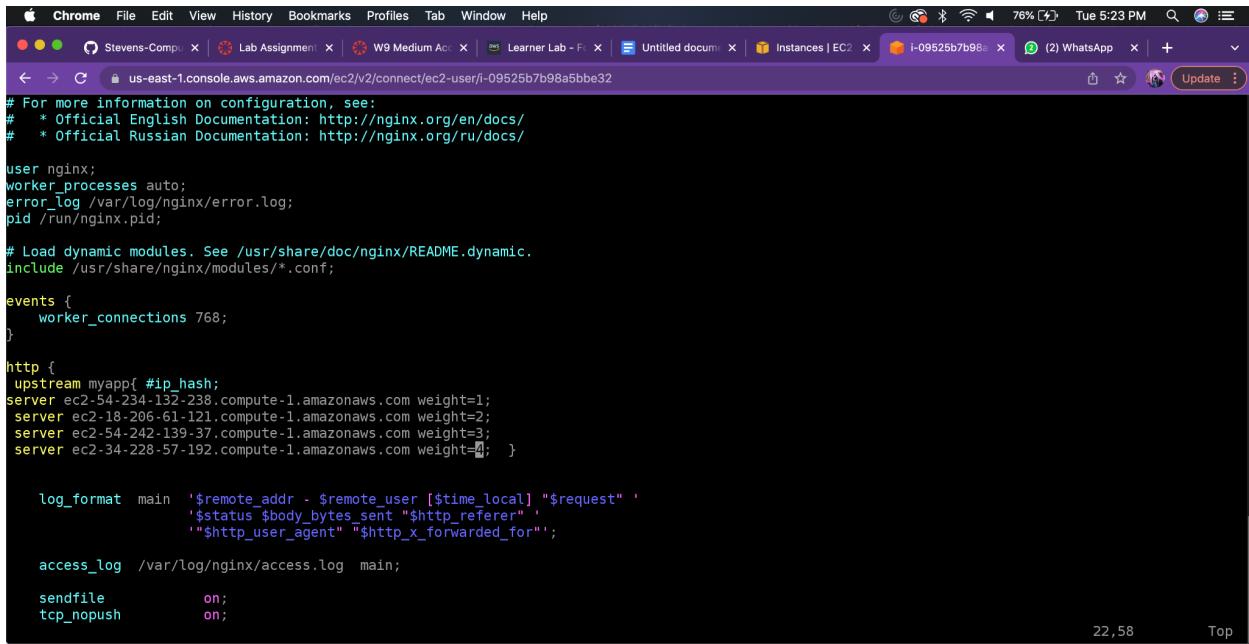
i-09525b7b98a5bbe32 (load_balancer)

Public IPs: 44.203.122.133 Private IPs: 172.31.88.138



i-09525b7b98a5bbe32 (load_balancer)

Public IPs: 44.203.122.133 Private IPs: 172.31.88.138



```
# 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 ec2-54-234-132-238.compute-1.amazonaws.com weight=1;
server ec2-18-206-61-121.compute-1.amazonaws.com weight=2;
server ec2-54-242-139-37.compute-1.amazonaws.com weight=3;
server ec2-34-228-57-192.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;

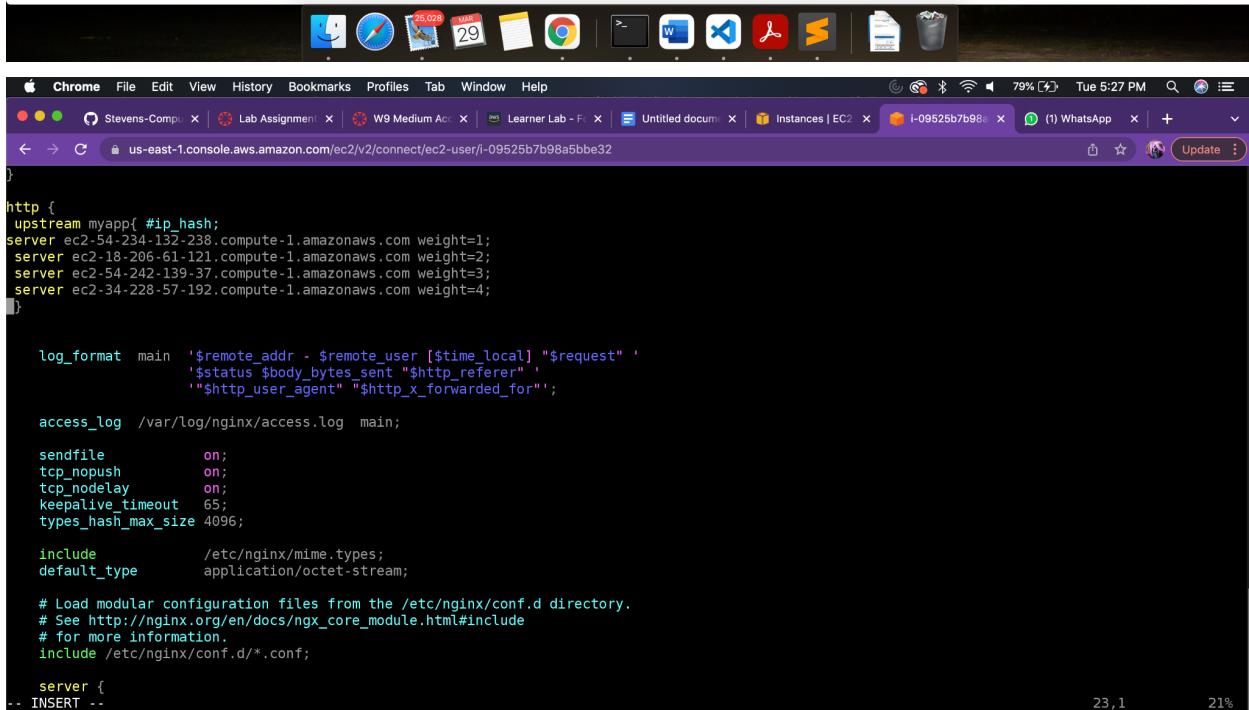
    sendfile      on;
    tcp_nopush    on;
}

```

22,58 Top

i-09525b7b98a5bbe32 (load_balancer)

Public IPs: 44.203.122.133 Private IPs: 172.31.88.138



```
}
```

```
http {
    upstream myapp{ #ip_hash;
server ec2-54-234-132-238.compute-1.amazonaws.com weight=1;
server ec2-18-206-61-121.compute-1.amazonaws.com weight=2;
server ec2-54-242-139-37.compute-1.amazonaws.com weight=3;
server ec2-34-228-57-192.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 {
-- INSERT --

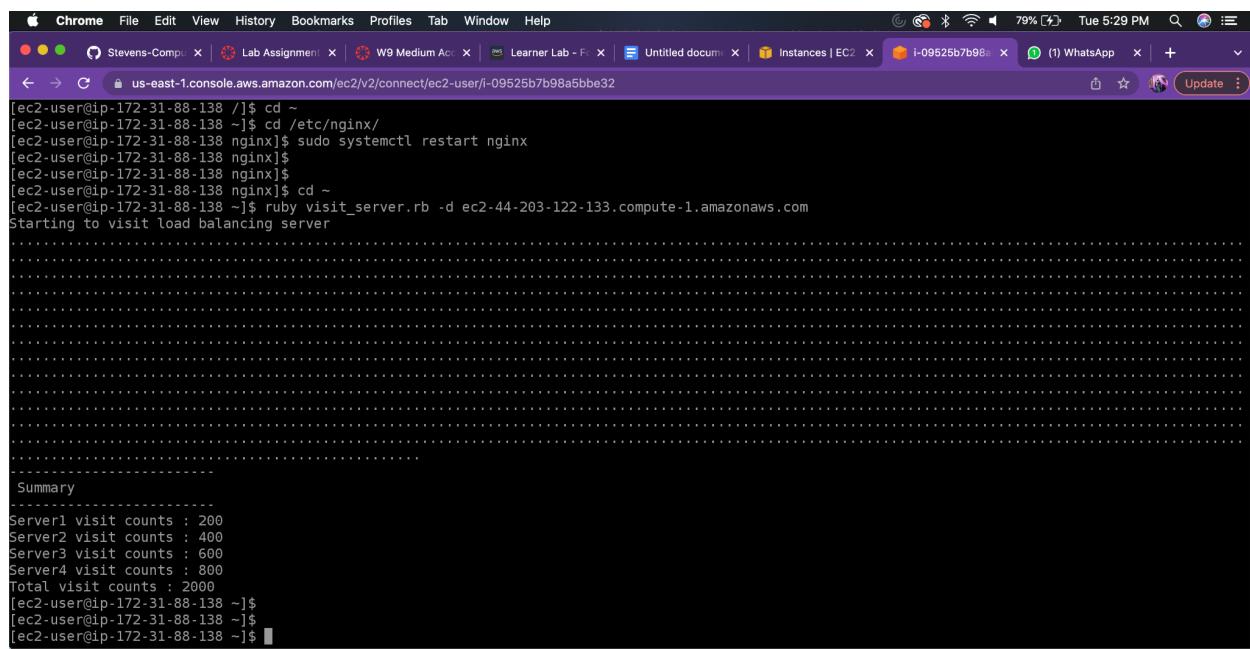
```

23,1 21%

i-09525b7b98a5bbe32 (load_balancer)

Public IPs: 44.203.122.133 Private IPs: 172.31.88.138





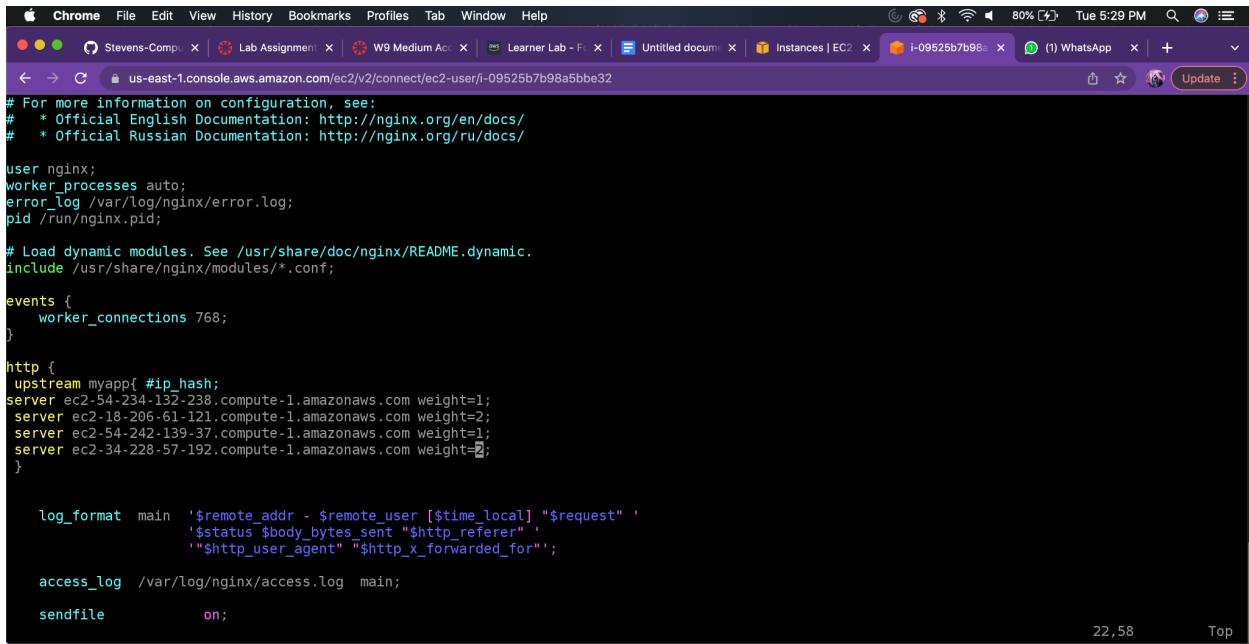
A screenshot of a Mac OS X desktop environment. At the top is the Dock, which includes icons for Finder, Mail, Safari, Calendar, Reminders, Notes, Chrome, Microsoft Word, Microsoft Excel, Microsoft PowerPoint, Visual Studio Code, and a trash can. Above the Dock is the system menu bar with options like Apple, Chrome, File, Edit, View, History, Bookmarks, Profiles, Tab, Window, and Help. The main window is a terminal session titled 'us-east-1.console.aws.amazon.com/ec2/v2/connect/ec2-user/i-09525b7b98a5bbe32'. The terminal output shows a series of commands being run on an EC2 instance:

```
[ec2-user@ip-172-31-88-138 ~]$ cd ~  
[ec2-user@ip-172-31-88-138 ~]$ cd /etc/nginx/  
[ec2-user@ip-172-31-88-138 nginx]$ sudo systemctl restart nginx  
[ec2-user@ip-172-31-88-138 nginx]$  
[ec2-user@ip-172-31-88-138 nginx]$  
[ec2-user@ip-172-31-88-138 nginx]$ cd ~  
[ec2-user@ip-172-31-88-138 ~]$ ruby visit_server.rb -d ec2-44-203-122-133.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-172-31-88-138 ~]$  
[ec2-user@ip-172-31-88-138 ~]$  
[ec2-user@ip-172-31-88-138 ~]$
```

i-09525b7b98a5bbe32 (load_balancer)

Public IPs: 44.203.122.133 Private IPs: 172.31.88.138





```
# 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 ec2-54-234-132-238.compute-1.amazonaws.com weight=1;
server ec2-18-206-61-121.compute-1.amazonaws.com weight=2;
server ec2-54-242-139-37.compute-1.amazonaws.com weight=1;
server ec2-34-228-57-192.compute-1.amazonaws.com weight=2;
}

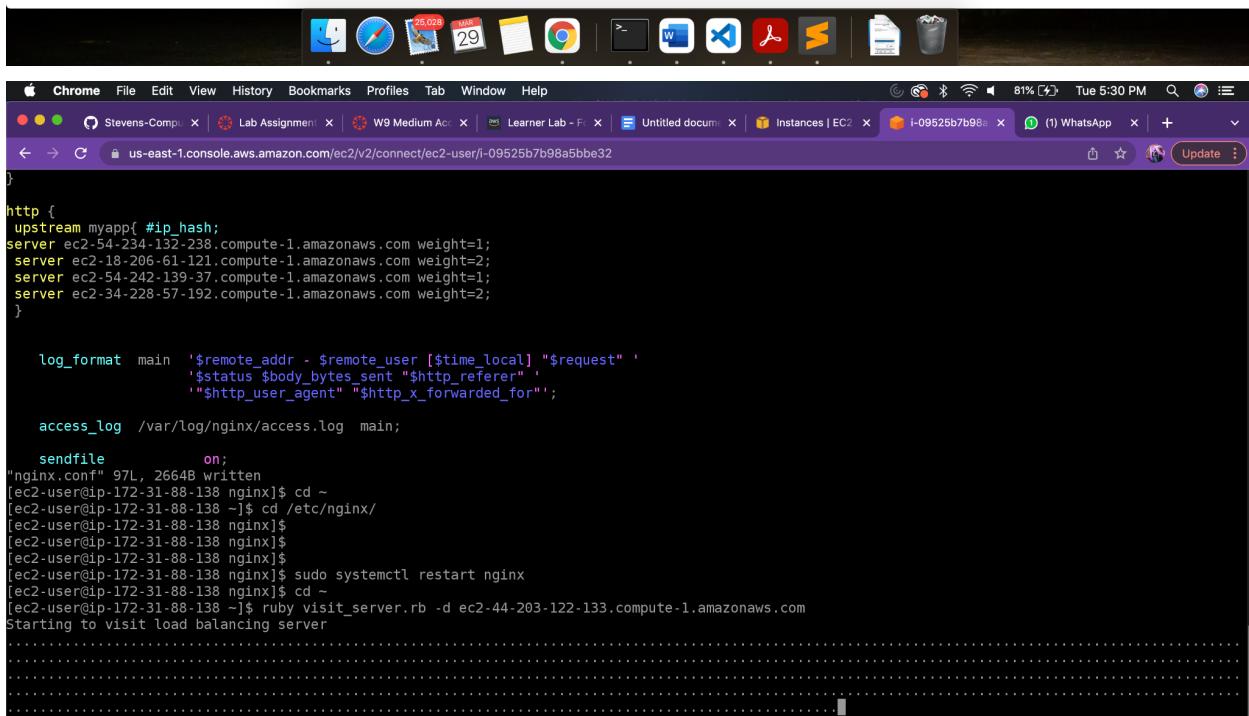
    log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
                    '$status $body_bytes_sent "$http_referer" '
                    '"$http_user_agent" "$http_x_forwarded_for"';

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

    sendfile      on;
}

```

i-09525b7b98a5bbe32 (load_balancer)
 Public IPs: 44.203.122.133 Private IPs: 172.31.88.138



```
}

http {
    upstream myapp{ #ip_hash;
server ec2-54-234-132-238.compute-1.amazonaws.com weight=1;
server ec2-18-206-61-121.compute-1.amazonaws.com weight=2;
server ec2-54-242-139-37.compute-1.amazonaws.com weight=1;
server ec2-34-228-57-192.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"';

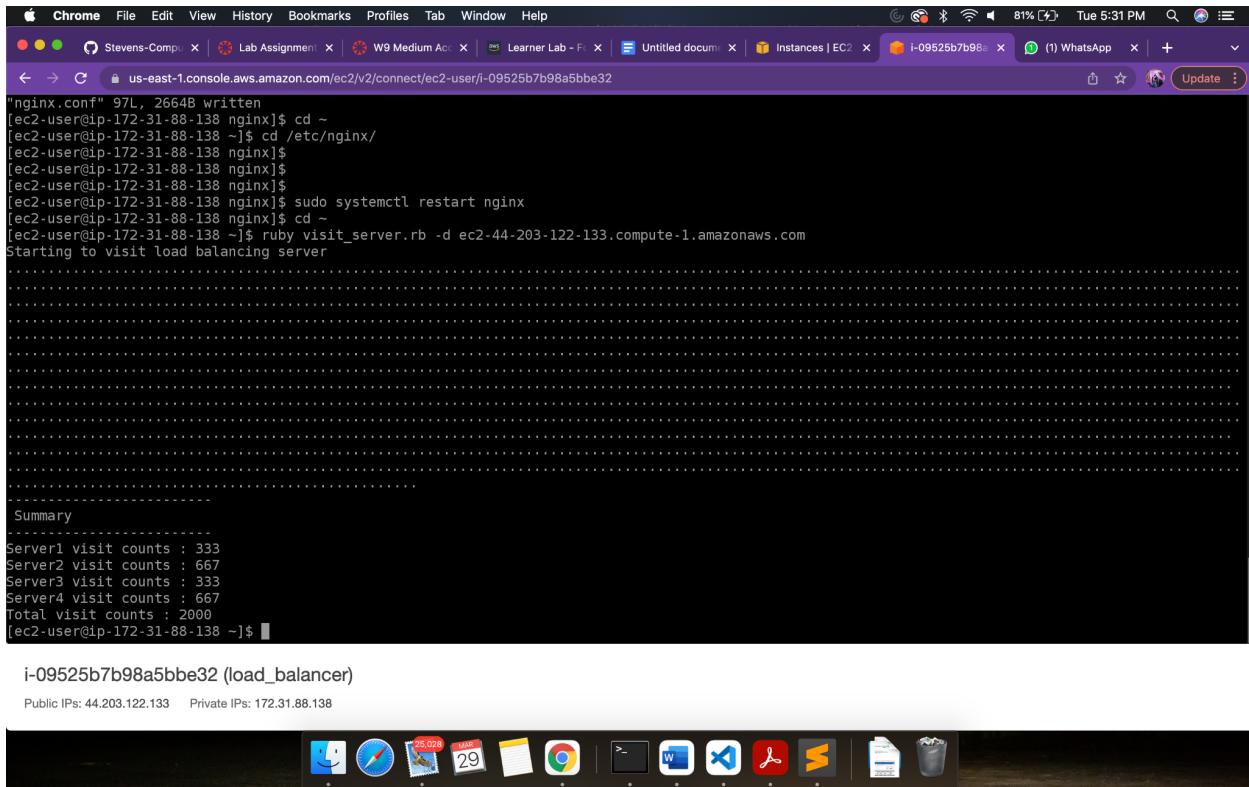
    access_log  /var/log/nginx/access.log  main;

    sendfile      on;
"nginx.conf" 97L, 2664B written
[ec2-user@ip-172-31-88-138 nginx]$ cd ~
[ec2-user@ip-172-31-88-138 ~]$ cd /etc/nginx/
[ec2-user@ip-172-31-88-138 nginx]$ 
[ec2-user@ip-172-31-88-138 nginx]$ 
[ec2-user@ip-172-31-88-138 nginx]$ 
[ec2-user@ip-172-31-88-138 nginx]$ sudo systemctl restart nginx
[ec2-user@ip-172-31-88-138 nginx]$ cd ~
[ec2-user@ip-172-31-88-138 ~]$ ruby visit_server.rb -d ec2-44-203-122-133.compute-1.amazonaws.com
Starting to visit load balancing server
.....
```

i-09525b7b98a5bbe32 (load_balancer)

Public IPs: 44.203.122.133 Private IPs: 172.31.88.138





The screenshot shows a Chrome browser window with multiple tabs open. The active tab displays terminal command-line output. The commands run include navigating to the nginx configuration directory, restarting the nginx service, and executing a Ruby script named `visit_server.rb` which performs a distributed load test across four servers. The output shows visit counts for each server and a total visit count.

```
"nginx.conf" 97L, 2664B written
[ec2-user@ip-172-31-88-138 nginx]$ cd ~
[ec2-user@ip-172-31-88-138 ~]$ cd /etc/nginx/
[ec2-user@ip-172-31-88-138 nginx]$
[ec2-user@ip-172-31-88-138 nginx]$
[ec2-user@ip-172-31-88-138 nginx]$
[ec2-user@ip-172-31-88-138 nginx]$
[ec2-user@ip-172-31-88-138 nginx]$ sudo systemctl restart nginx
[ec2-user@ip-172-31-88-138 nginx]$ cd ~
[ec2-user@ip-172-31-88-138 nginx]$
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-172-31-88-138 ~]$ 
```

i-09525b7b98a5bbe32 (load_balancer)
 Public IPs: 44.203.122.133 Private IPs: 172.31.88.138



The following results were achieved :-

Run no.	Server 1 Weight	Visit Count	Server 2 Weight	Visit Count	Server 3 Weight	Visit Count	Server 4 Weight	Visit Count
1	1	500	1	500	1	500	1	500
2	1	200	2	400	3	600	4	800
3	1	333	2	667	1	333	2	667

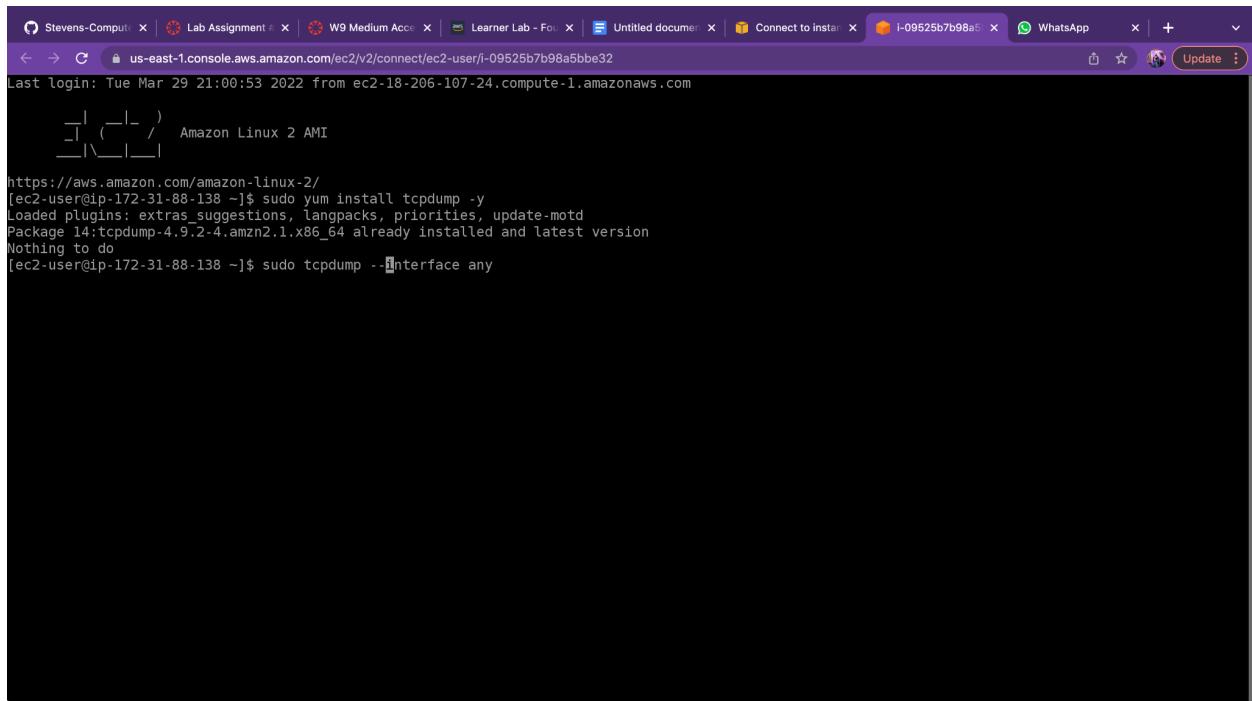
TCPDump

I installed tcp dump using the command

Sudo yum install tcp -y.

Then I used the following command to collect packets :-

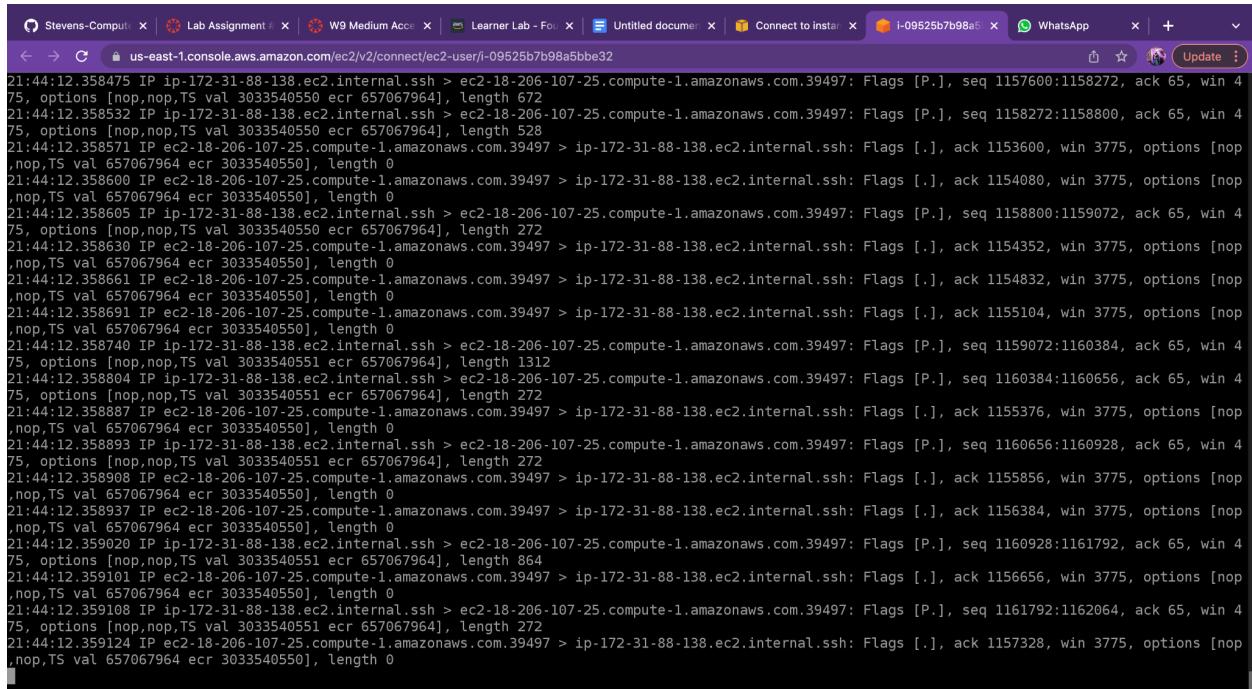
command sudo tcpdump -interface any



```
Stevens-Comput X | Lab Assignment X | W9 Medium Acc X | Learner Lab - Fo X | Untitled document X | Connect to insta X | i-09525b7b98a5bbe32 X | WhatsApp X | + Update : 
us-east-1.console.aws.amazon.com/ec2/v2/connect/ec2-user/i-09525b7b98a5bbe32
Last login: Tue Mar 29 21:00:53 2022 from ec2-18-206-107-24.compute-1.amazonaws.com
[ec2-user@ip-172-31-88-138 ~]$ sudo yum install tcpdump -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Package 14:tcpdump-4.9.2-4.amzn2.1.x86_64 already installed and latest version
Nothing to do
[ec2-user@ip-172-31-88-138 ~]$ sudo tcpdump -i interface any
```

i-09525b7b98a5bbe32 (load_balancer)

Public IPs: 44.203.122.133 Private IPs: 172.31.88.138



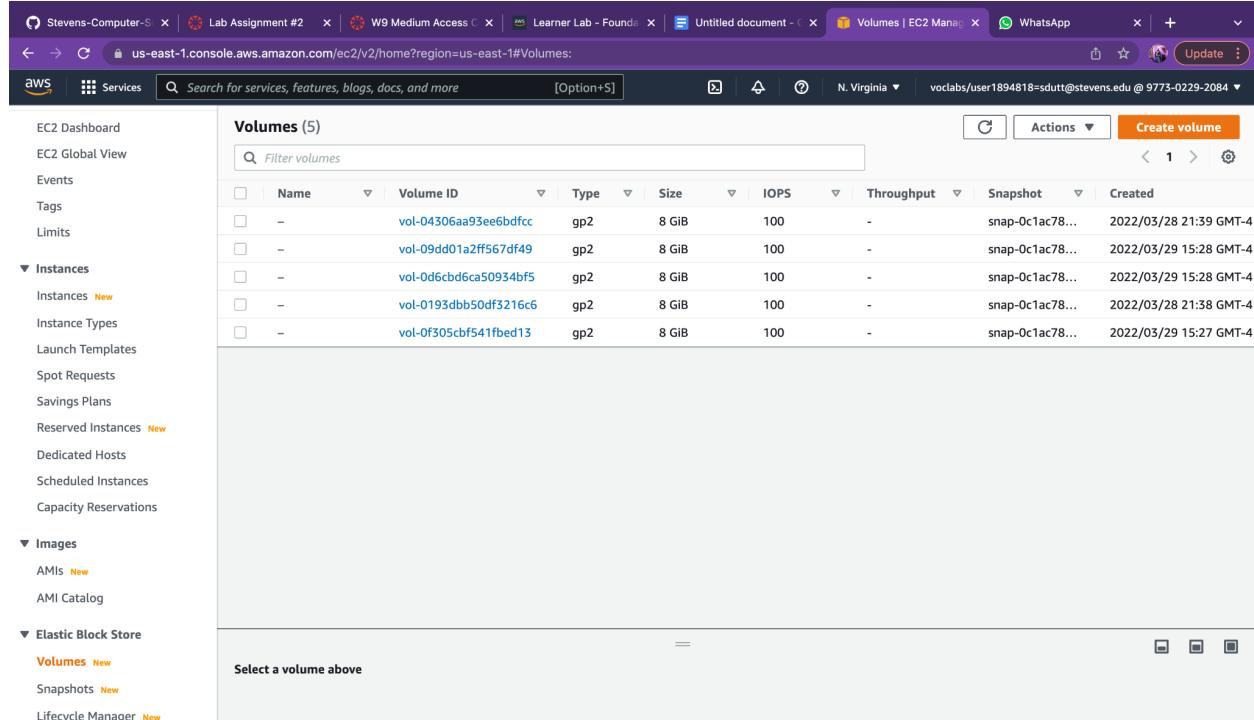
```
21:44:12.358475 IP ip-172-31-88-138.ec2.internal.ssh > ec2-18-206-107-25.compute-1.amazonaws.com.39497: Flags [P.], seq 1157600:1158272, ack 65, win 475, options [nop,nop,TS val 3033540550 ecr 657067964], length 672
21:44:12.358532 IP ip-172-31-88-138.ec2.internal.ssh > ec2-18-206-107-25.compute-1.amazonaws.com.39497: Flags [P.], seq 1158272:1158800, ack 65, win 475, options [nop,nop,TS val 657067964 ecr 3033540550], length 528
21:44:12.358571 IP ec2-18-206-107-25.compute-1.amazonaws.com.39497 > ip-172-31-88-138.ec2.internal.ssh: Flags [.], ack 1153600, win 3775, options [nop,nop,TS val 657067964 ecr 3033540550], length 0
21:44:12.358600 IP ec2-18-206-107-25.compute-1.amazonaws.com.39497 > ip-172-31-88-138.ec2.internal.ssh: Flags [.], ack 1154080, win 3775, options [nop,nop,TS val 657067964 ecr 3033540550], length 0
21:44:12.358605 IP ip-172-31-88-138.ec2.internal.ssh > ec2-18-206-107-25.compute-1.amazonaws.com.39497: Flags [P.], seq 1158800:1159072, ack 65, win 475, options [nop,nop,TS val 3033540550 ecr 657067964], length 272
21:44:12.358630 IP ec2-18-206-107-25.compute-1.amazonaws.com.39497 > ip-172-31-88-138.ec2.internal.ssh: Flags [.], ack 1154352, win 3775, options [nop,nop,TS val 657067964 ecr 3033540550], length 0
21:44:12.358661 IP ec2-18-206-107-25.compute-1.amazonaws.com.39497 > ip-172-31-88-138.ec2.internal.ssh: Flags [.], ack 1154832, win 3775, options [nop,nop,TS val 657067964 ecr 3033540550], length 0
21:44:12.358691 IP ec2-18-206-107-25.compute-1.amazonaws.com.39497 > ip-172-31-88-138.ec2.internal.ssh: Flags [.], ack 1155104, win 3775, options [nop,nop,TS val 657067964 ecr 3033540550], length 0
21:44:12.358740 IP ip-172-31-88-138.ec2.internal.ssh > ec2-18-206-107-25.compute-1.amazonaws.com.39497: Flags [P.], seq 1159072:1160384, ack 65, win 475, options [nop,nop,TS val 3033540551 ecr 657067964], length 1312
21:44:12.358804 IP ip-172-31-88-138.ec2.internal.ssh > ec2-18-206-107-25.compute-1.amazonaws.com.39497: Flags [P.], seq 1160384:1160656, ack 65, win 475, options [nop,nop,TS val 3033540551 ecr 657067964], length 272
21:44:12.358887 IP ec2-18-206-107-25.compute-1.amazonaws.com.39497 > ip-172-31-88-138.ec2.internal.ssh: Flags [.], ack 1155376, win 3775, options [nop,nop,TS val 657067964 ecr 3033540550], length 0
21:44:12.358893 IP ip-172-31-88-138.ec2.internal.ssh > ec2-18-206-107-25.compute-1.amazonaws.com.39497: Flags [P.], seq 1160656:1160928, ack 65, win 475, options [nop,nop,TS val 3033540551 ecr 657067964], length 272
21:44:12.358908 IP ec2-18-206-107-25.compute-1.amazonaws.com.39497 > ip-172-31-88-138.ec2.internal.ssh: Flags [.], ack 1155856, win 3775, options [nop,nop,TS val 657067964 ecr 3033540550], length 0
21:44:12.358937 IP ec2-18-206-107-25.compute-1.amazonaws.com.39497 > ip-172-31-88-138.ec2.internal.ssh: Flags [.], ack 1156384, win 3775, options [nop,nop,TS val 657067964 ecr 3033540550], length 0
21:44:12.359020 IP ip-172-31-88-138.ec2.internal.ssh > ec2-18-206-107-25.compute-1.amazonaws.com.39497: Flags [P.], seq 1160928:1161792, ack 65, win 475, options [nop,nop,TS val 3033540551 ecr 657067964], length 864
21:44:12.359101 IP ec2-18-206-107-25.compute-1.amazonaws.com.39497 > ip-172-31-88-138.ec2.internal.ssh: Flags [.], ack 1156656, win 3775, options [nop,nop,TS val 657067964 ecr 3033540550], length 0
21:44:12.359108 IP ip-172-31-88-138.ec2.internal.ssh > ec2-18-206-107-25.compute-1.amazonaws.com.39497: Flags [P.], seq 1161792:1162064, ack 65, win 475, options [nop,nop,TS val 3033540551 ecr 657067964], length 272
21:44:12.359124 IP ec2-18-206-107-25.compute-1.amazonaws.com.39497 > ip-172-31-88-138.ec2.internal.ssh: Flags [.], ack 1157328, win 3775, options [nop,nop,TS val 657067964 ecr 3033540550], length 0
```

i-09525b7b98a5bbe32 (load_balancer)

Public IPs: 44.203.122.133 Private IPs: 172.31.88.138

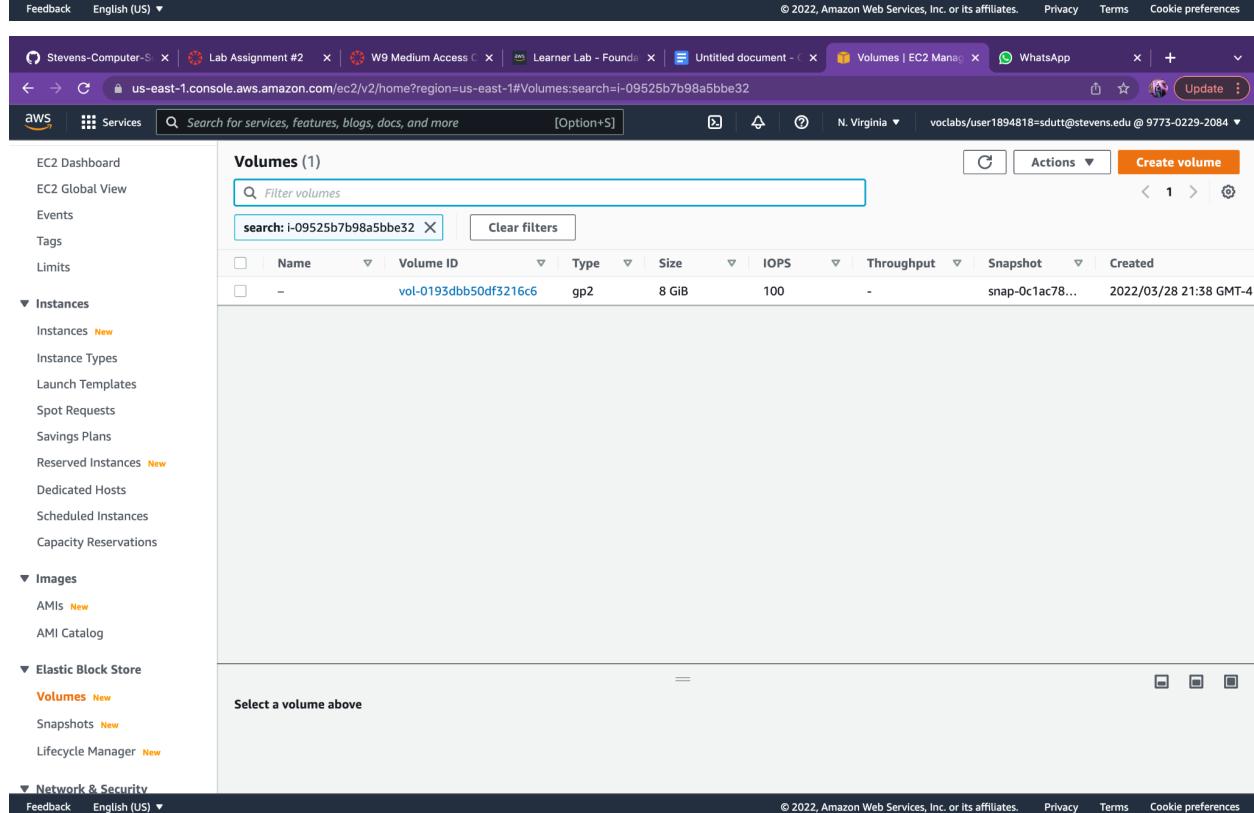
Creating an instance from AMI

I went to the volumes tab and selected the load balancer volume.



The screenshot shows the AWS EC2 Dashboard with the 'Volumes' section selected. There are five gp2 volumes listed:

Name	Volume ID	Type	Size	IOPS	Throughput	Snapshot	Created
-	vol-04306aa93ee6bdfcc	gp2	8 GiB	100	-	snap-0c1ac78...	2022/03/28 21:39 GMT-4
-	vol-09dd01a2ff567df49	gp2	8 GiB	100	-	snap-0c1ac78...	2022/03/29 15:28 GMT-4
-	vol-0d6cbd6ca50934bf5	gp2	8 GiB	100	-	snap-0c1ac78...	2022/03/29 15:28 GMT-4
-	vol-0193dbb50df3216c6	gp2	8 GiB	100	-	snap-0c1ac78...	2022/03/28 21:38 GMT-4
-	vol-0f305cbf541fbcd13	gp2	8 GiB	100	-	snap-0c1ac78...	2022/03/29 15:27 GMT-4



The screenshot shows the same AWS EC2 Dashboard, but with a search filter applied: 'search: i-09525b7b98a5bbe32'. The results show one gp2 volume:

Name	Volume ID	Type	Size	IOPS	Throughput	Snapshot	Created
-	vol-0193dbb50df3216c6	gp2	8 GiB	100	-	snap-0c1ac78...	2022/03/28 21:38 GMT-4

Selected the option to create snapshot from 'Actions' dropdown.

The screenshot shows the AWS EC2 Volumes page. On the left, there's a sidebar with options like EC2 Dashboard, EC2 Global View, Events, Tags, Limits, Instances, Images, and Elastic Block Store. Under Instances, 'Instances' is selected. In the main area, a table titled 'Volumes (1/1)' shows one volume: vol-0193dbb50df3216c6, which is gp2, 8 GiB, and 100 IOPS. To the right of the table is an 'Actions' dropdown menu with options: Modify volume, Create snapshot, Create snapshot lifecycle policy, Delete volume, Attach volume, Detach volume, Force detach volume, Manage auto-enabled I/O, and Manage tags. The 'Create snapshot' option is highlighted. At the bottom of the page, there's a 'Volume ID: vol-0193dbb50df3216c6' section with tabs for Details, Status checks, Monitoring, and Tags.

Entered the description of snapshot and created snapshot.

The screenshot shows the 'Create snapshot' wizard. The first step, 'Details', is active. It has fields for 'Volume ID' (vol-0193dbb50df3216c6), 'Description' (Snapshot of Load Balancer), and 'Encryption' (Not encrypted). Below this is a 'Tags' section with a note about tags being optional. The 'Tags' section also includes an 'Add tag' button and a note that you can add up to 50 more tags. At the bottom of the wizard are 'Cancel' and 'Create snapshot' buttons.

The screenshot shows the AWS EC2 Volumes page. At the top, a green success message states: "Successfully created snapshot snap-00af83511018bb4ea from volume vol-0193dbb50df3216c6. If you need your snapshot to be immediately available consider using Fast Snapshot Restore." Below this, there is a table titled "Volumes (5)" with columns: Name, Volume ID, Type, Size, IOPS, Throughput, Snapshot, and Created. The table lists five volumes, each with a unique ID and creation date. The last volume listed is "vol-0193dbb50df3216c6".

Upon creating a snapshot I went to the Images tab and selected AMIs.

The screenshot shows the AWS Snapshots page. At the top, a green success message states: "Successfully created snapshot snap-00af83511018bb4ea from volume vol-0193dbb50df3216c6. If you need your snapshot to be immediately available consider using Fast Snapshot Restore." Below this, there is a table titled "Snapshots (1)" with columns: Name, Snapshot ID, Size, Description, Storage..., Snapshot status, and Started. The table lists one snapshot, "snap-00af83511018bb4ea", which is a "Snapshot of Load Balancer" in Standard storage and is currently Pending.

I selected the Snapshot from the list and selected create image from snapshot option in Actions drop down.

The screenshot shows the AWS EC2 Snapshots page. On the left, there's a sidebar with navigation links for EC2 Dashboard, EC2 Global View, Events, Tags, Limits, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Scheduled Instances, Capacity Reservations, Images, AMIs, and Elastic Block Store. The main area displays a table titled 'Snapshots (1/1)' with one item: 'snap-00af83511018bb4ea'. The table columns include Name, Snapshot ID, Size, Description, and Storage type. To the right of the table is a context menu with options like Create volume from snapshot, Create image from snapshot (which is highlighted), Copy snapshot, Modify permissions, Manage fast snapshot restore, Archive snapshot, Restore snapshot from archive, Change restore period, Delete snapshot, and Manage tags. At the bottom of the main area, there's a detailed view for the selected snapshot.

The screenshot shows the 'Create image from snapshot' wizard. The current step is 'Image settings'. It has fields for Snapshot ID (set to 'snap-00af83511018bb4ea'), Image name (set to 'AMI from Snapshot'), Description (set to 'My image description'), Architecture (set to 'x86_64'), Root device name (set to '/dev/sda1'), and Virtualization type (set to 'The virtualization type to be used by instances launched from this image'). The status bar at the bottom indicates the task is 0% complete.

Successfully created image.

Screenshot of the AWS EC2 Snapshots page showing a successful snapshot creation:

Successfully requested new image ami-0c35fc34c042709d2.
The image is being created. The image-creation process can take several minutes to complete.

Snapshots (1)

Name	Snapshot ID	Size	Description	Storage...	Snapshot status	Started
-	snap-00af83511018bb4ea	8 GiB	Snapshot of Load Balancer	Standard	Completed	2022/03/29 17:55 G

Select a snapshot above.

Feedback English (US) ▾ © 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Screenshot of the AWS EC2 Snapshots page showing the same successful snapshot creation, but with a different snapshot ID:

Successfully requested new image ami-0c35fc34c042709d2.
The image is being created. The image-creation process can take several minutes to complete.

Snapshots (1/1)

Name	Snapshot ID	Size	Description	Storage...	Snapshot status	Started
-	snap-00af83511018bb4ea	8 GiB	Snapshot of Load Balancer	Standard	Completed	2022/03/29 17:55 G

Snapshot ID: snap-00af83511018bb4ea

Details	Permissions	Storage tier	Tags
Snapshot ID snap-00af83511018bb4ea	Size 8 GiB	Progress Available (100%)	Snapshot status Completed
Owner 977302292084	Volume ID vol-0193ddb50df3216c6	Started Tue Mar 29 2022 17:55:42 GMT-0400 (Eastern Daylight Time)	Product codes -
Encryption Not encrypted	KMS key ID -	KMS key alias -	KMS key ARN -
Fast snapshot restore -	Description Snapshot of Load Balancer		

Feedback English (US) ▾ © 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

I went to the AMI tab again and selected the Image. I selected the option to Launch instance from AMI.

Amazon Machine Images (AMIs) (1/1)

Name	AMI ID	AMI name	Source	Owner
–	ami-0c35fc34c042709d2	AMI from Snapshot	977302292084/AMI from Snapshot	977302292084

AMI ID: ami-0c35fc34c042709d2

Details	Permissions	Storage	Tags
AMI ID ami-0c35fc34c042709d2	Image type machine	Platform details Linux/UNIX	Root device type EBS
AMI name AMI from Snapshot	Owner account ID 977302292084	Architecture x86_64	Usage operation RunInstances
Root device name /dev/sda1	Status Available	Source 977302292084/AMI from Snapshot	Virtualization type hvm
Boot mode –	State reason –	Creation date Tue Mar 29 2022 17:59:53 GMT-0400	Kernel ID –

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

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

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.

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

Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
t2	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes

Review and Launch

The screenshot shows the AWS Lambda console with the URL <https://us-east-1.console.aws.amazon.com/ec2/v2/home?region=us-east-1#LaunchInstanceWizard:ami=ami-0c35fc34c042709d2>. The tab bar includes 'Launch instance wizard' and 'Configure Security Group'. The main content area shows a table of security groups with one selected:

Security Group ID	Name	Description	Actions
sg-0d9ba1b79526bc364	default	default VPC security group	Copy to new
sg-0ffa0081fad3943a9	load_balancer1	launch-wizard-1 created 2022-03-28T21:34:18.831-04:00	Copy to new

Step 6: Configure Security Group

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

Assign a security group: Create a new security group

Select an existing security group

Security Group ID	Name	Description	Actions
sg-0d9ba1b79526bc364	default	default VPC security group	Copy to new
sg-0ffa0081fad3943a9	load_balancer1	launch-wizard-1 created 2022-03-28T21:34:18.831-04:00	Copy to new

Inbound rules for sg-0ffa0081fad3943a9 (Selected security groups: sg-0ffa0081fad3943a9)

Type	Protocol	Port Range	Source	Description
HTTP	TCP	80	0.0.0.0/0	
HTTP	TCP	80	::/0	
SSH	TCP	22	0.0.0.0/0	
HTTPS	TCP	443	0.0.0.0/0	

[Cancel](#) [Previous](#) [Review and Launch](#)

The screenshot shows the AWS Lambda console with the URL <https://us-east-1.console.aws.amazon.com/ec2/v2/home?region=us-east-1#LaunchInstanceWizard:ami=ami-0c35fc34c042709d2>. The tab bar includes 'Launch instance wizard' and 'Configure Security Group'. The main content area shows a summary of the launch details:

AMI	Instance Type	Security Groups	Key Pair	Block Device Mapping
AMI from Snapshot - ami-0c35fc34c042709d2	t2.micro	load_balancer1	sdutt	Root Device Type: ebs Virtualization type: hvm

Step 7: Review Instance Launch

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

⚠ Improve your instances' security. Your security group, load_balancer1, is open to the world.

Your instances may be accessible from any IP address. We recommend that you update your security group rules to allow access from known IP addresses only.

You can also open additional ports in your security group to facilitate access to the application or service you're running, e.g., HTTP (80) for web servers. [Edit security groups](#)

The screenshot shows the AWS Lambda console with the URL <https://us-east-1.console.aws.amazon.com/ec2/v2/home?region=us-east-1#LaunchInstanceWizard:ami=ami-0c35fc34c042709d2>. The tab bar includes 'Launch instance wizard' and 'Configure Security Group'. The main content area shows detailed launch settings:

- AMI Details**: AMI from Snapshot - ami-0c35fc34c042709d2, Root Device Type: ebs, Virtualization type: hvm.
- Instance Type**: t2.micro
- Security Groups**: load_balancer1
- Launch Options**: Define key pair and launch

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 Snapshot - ami-0c35fc34c
Root Device Type: ebs Virtualization type: hvm

Instance Type
t2.micro ECU: 1 vCPU: 1

Security Groups
sg-0ffa0081fad3943a9 load_balancer1

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. Amazon EC2 supports ED25519 and RSA key pair types.

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

Choose an existing key pair
Select a key pair
load_balancer_Security | RSA
 I acknowledge that I have access to the corresponding private key file, and that without this file, I won't be able to log into my instance.

Cancel Launch Instances

launch-wizard-1 created 2022-03-28T21:34:18.831-04:00

Launch Status

Your instances are now launching.

The following instance launches have been initiated: i-0e1ab066b4b5ee716 [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](#)
- [Amazon EC2: User Guide](#)
- [Learn about AWS Free Usage Tier](#)
- [Amazon EC2: Discussion Forum](#)

While your instances are launching you can also

- [Create status check alarms](#) to be notified when these instances fail status checks. (Additional charges may apply)
- [Create and attach additional EBS volumes](#) (Additional charges may apply)
- [Manage security groups](#)

[View Instances](#)

The instance was successfully created using AMI.

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with navigation links like EC2 Dashboard, EC2 Global View, Events, Tags, Limits, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Scheduled Instances, Capacity Reservations, Images, AMIs, and Elastic Block Store. The main area displays a table titled 'Instances (1/6) Info' with columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, and Availability Zone. There are six entries: load_balancer (stopped, t2.micro, us-east-1d), server2 (stopped, t2.micro, us-east-1d), server1 (stopped, t2.micro, us-east-1a), server4 (stopped, t2.micro, us-east-1a), server3 (stopped, t2.micro, us-east-1a), and a new entry 'Instance from AMI' (pending, t2.micro, us-east-1d). Below the table, a modal window for 'Instance: i-0e1ab066b4b5ee716 (Instance from AMI)' is open, showing details like Public IPv4 address (54.163.188.201), Instance state (Pending), and Private IP DNS name (ec2-54-163-188-201.compute-1.amazonaws.com).

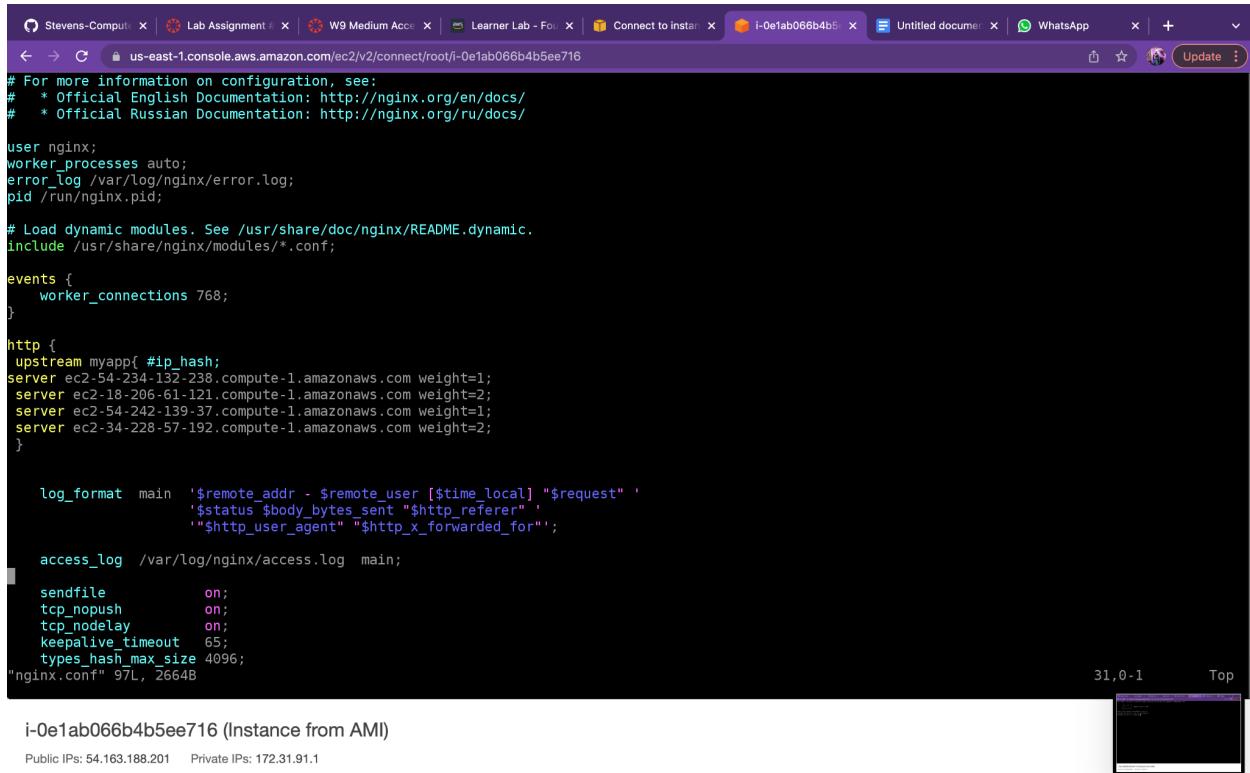
I connected to the console and was able to see the files originally created in `load_balancer`.

The screenshot shows a CloudShell terminal window. The URL in the address bar is `us-east-1.console.aws.amazon.com/ec2/v2/connect/root/i-0e1ab066b4b5ee716`. The terminal output shows:

```
Last login: Tue Mar 29 22:04:01 2022 from ec2-18-206-107-24.compute-1.amazonaws.com
[root@ip-172-31-91-1 ~]# cd /etc/nginx/
[root@ip-172-31-91-1 nginx]# ls
```

`i-0e1ab066b4b5ee716 (Instance from AMI)`

Public IPs: 54.163.188.201 Private IPs: 172.31.91.1



```
# 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 ec2-54-234-132-238.compute-1.amazonaws.com weight=1;
        server ec2-18-206-61-121.compute-1.amazonaws.com weight=2;
        server ec2-54-242-139-37.compute-1.amazonaws.com weight=1;
        server ec2-34-228-57-192.compute-1.amazonaws.com weight=2;
    }

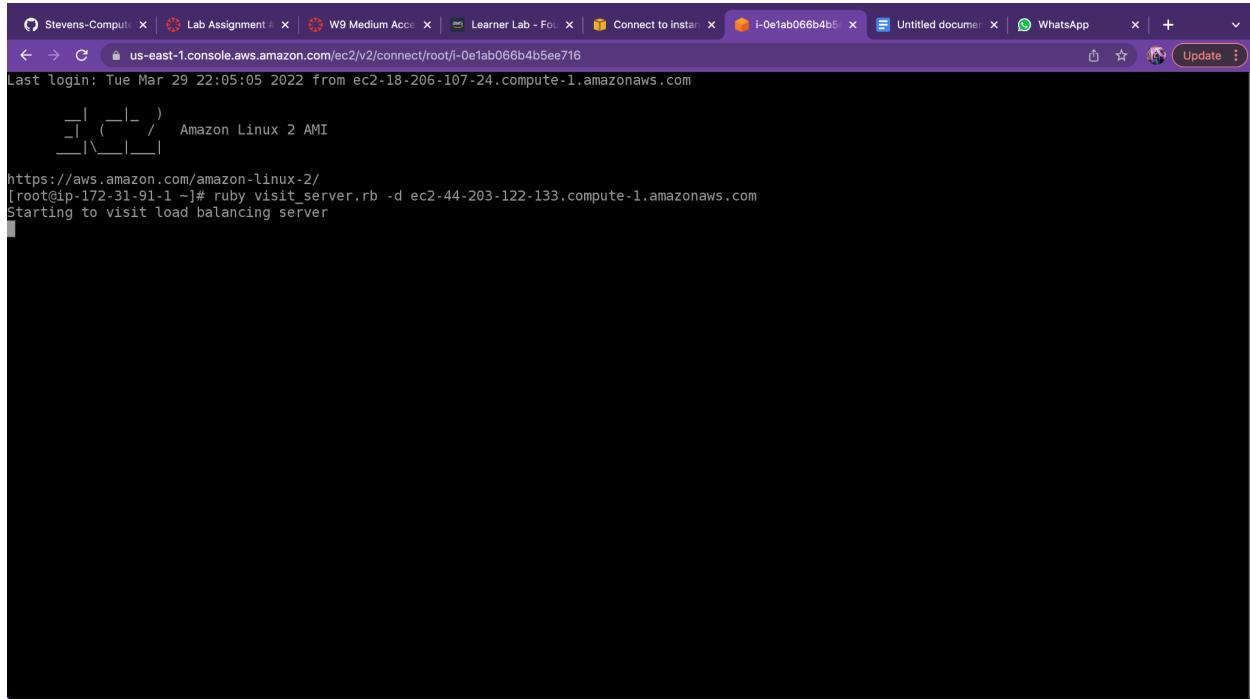
    log_format main '$remote_addr - $remote_user [$time_local] "$request" '
                   '$status $body_bytes_sent "$http_referer" '
                   '"$http_user_agent" "$http_x_forwarded_for"';

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

    sendfile      on;
    tcp_nopush    on;
    tcp_nodelay   on;
    keepalive_timeout 65;
    types_hash_max_size 4096;
}
"nginx.conf" 97L, 2664B
```

i-0e1ab066b4b5ee716 (Instance from AMI)

Public IPs: 54.163.188.201 Private IPs: 172.31.91.1



```
Last login: Tue Mar 29 22:05:05 2022 from ec2-18-206-107-24.compute-1.amazonaws.com
 _ _|_(_|_) / Amazon Linux 2 AMI
 __|_\_|__|_

https://aws.amazon.com/amazon-linux-2/
[root@ip-172-31-91-1 ~]# ruby visit_server.rb -d ec2-44-203-122-133.compute-1.amazonaws.com
Starting to visit load balancing server
```

i-0e1ab066b4b5ee716 (Instance from AMI)

Public IPs: 54.163.188.201 Private IPs: 172.31.91.1

Terminate Instances

I selected the instances and went to the instance state dropdown list. I selected terminate instance option.

Instances (6/6) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
load_balancer	i-09525b7b98a5bbe32	Running	t2.micro	2/2 checks passed	No alarms	us-east-1d
server2	i-02fc6625bf1313ea	Running	t2.micro	2/2 checks passed	No alarms	us-east-1d
Instance from AMI	i-0e1ab066b4b5ee716	Running	t2.micro	2/2 checks passed	No alarms	us-east-1d
server1	i-05087fc0a6b5ff947	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a
server4	i-0b4b728013de1f557	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a
server3	i-0b15d40200dc092bb	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a

Instances: i-09525b7b98a5bbe32 (load_balancer), i-02fc6625bf1313ea (server2), i-0e1ab066b4b5ee716 (Instance from AMI), i-05087fc0a6b5ff947 (server1), i-0b4b728013de1f557 (server4), i-0b15d40200dc092bb (server3)

Terminate instances?

⚠ On an EBS-backed instance, the default action is for the root EBS volume to be deleted when the instance is terminated. Storage on any local drives will be lost.

Are you sure you want to terminate these instances?

i-09525b7b98a5bbe32 (load_balancer)
 i-02fc6625bf1313ea (server2)
 i-0e1ab066b4b5ee716 (Instance from AMI)
 i-05087fc0a6b5ff947 (server1)
 i-0b4b728013de1f557 (server4)
+ 1 more

To confirm that you want to terminate the instances, choose the terminate button below. Terminating the instance cannot be undone.

Cancel **Terminate**

The instances were successfully terminated.

The screenshot shows the AWS EC2 Instances page with a success message: "Successfully terminated i-09525b7b98a5bbe32, i-02fc6625bf1313ea, i-0e1ab066b4b5ee716, i-05087fc0a6b5ff947, i-0b4b728013de1f557, i-0b15d40200dc092bb". The table lists six terminated t2.micro instances with their respective names and instance IDs. The status check column shows a minus sign, and the alarm status column shows "No alarms". The availability zone for all instances is "us-east-1d".

Next I deregistered AMI by going to the AMI tab and selecting the image from the list then going to Actions dropdown list and selecting the option to Deregister AMI.

The screenshot shows the AWS AMI Management page for the "Amazon Machine Images (AMIs) (1/1)" section. It displays a single AMI entry with the ID "ami-0c35fc34c042709d2", which is an "AMI from Snapshot". The "Actions" dropdown menu is open, showing options like "Copy AMI", "Edit AMI permissions", "Request Spot Instances", "Manage tags", "Deregister AMI", "Change description", "Manage image optimization", "Manage AMI Deprecation", and "Register instance store-backed AMI". The "Deregister AMI" option is highlighted.

The screenshot shows the AWS EC2 console interface. On the left, a sidebar navigation menu includes options like EC2 Dashboard, Instances, Images, and Elastic Block Store. The main content area displays a table titled "Amazon Machine Images (AMIs) (1/1)" with one item: "ami-0c35fc34c042709d2". A modal window titled "Deregister AMI" is open, asking for confirmation to deregister the AMI. The modal also contains a link to "Delete associated snapshots". Below the table, detailed information about the AMI is shown, including its ID, name, owner account, architecture, root device name, status, and creation date.

The AMI was successfully deregistered.

The screenshot shows the AWS EC2 console after the AMI has been deregistered. A green success message at the top states "Successfully deregistered ami-0c35fc34c042709d2." The main content area now shows a message: "You do not have any images in this Region." The sidebar and table structure remain the same as in the previous screenshot.

Finally I deleted the snapshot by selecting the snapshot and going to Actions drop down list then selecting delete snapshot option.

Successfully requested new image ami-0c35fc34c042709d2.
The image is being created. The image-creation process can take several minutes to complete.

Snapshots (1/1)						
Owned by me		Filter snapshots by attributes and tags				
<input checked="" type="checkbox"/>	Name	Snapshot ID	Size	Description	Storage...	SN
<input checked="" type="checkbox"/>	-	snap-00af83511018bb4ea	8 GiB	Snapshot of Load Balancer	Standard	

Snapshot ID: snap-00af83511018bb4ea						
Details		Permissions	Storage tier	Tags		
Snapshot ID	snap-00af83511018bb4ea	Size	8 GiB	Progress	Available (100%)	
Owner	977302292084	Volume ID	vol-0193dbb50df3216c6	Started	Tue Mar 29 2022 17:55:42 GMT-0400 (Eastern Daylight Time)	
Encryption	Not encrypted	KMS key ID	-	KMS key alias	-	
Fast snapshot restore	-	Description	Snapshot of Load Balancer	KMS key ARN	-	

Successfully requested new image ami-0c35fc34c042709d2.
The image is being created. The image-creation process can take several minutes to complete.

Snapshots (1/1)						
Owned by me		Filter snapshots by attributes and tags				
<input checked="" type="checkbox"/>	Name	Snapshot ID	Size	Description	Storage...	Snapshot status
<input checked="" type="checkbox"/>	-	snap-00af83511018bb4ea	8 GiB	Snapshot of Load Balancer	Standard	

Delete snap-00af83511018bb4ea?

Are you sure that you want to delete snap-00af83511018bb4ea?

Snapshot ID: snap-00af83511018bb4ea						
Details		Permissions	Storage	Tags		
Snapshot ID	snap-00af83511018bb4ea	Volume ID	vol-0193dbb50df3216c6	Started	Product codes	
Owner	977302292084	KMS key ID	-	KMS key alias	snapshot status	
Encryption	Not encrypted	Description	Snapshot of Load Balancer	KMS key ARN		

Successfully deleted snapshot.

The screenshot shows the AWS Management Console interface for the EC2 service. The left sidebar contains navigation links for Tags, Limits, Instances (with sub-links for Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Scheduled Instances, Capacity Reservations), Images (AMIs, AMI Catalog), Elastic Block Store (Volumes, Snapshots, Lifecycle Manager), Network & Security (Security Groups, Elastic IPs, Placement Groups), and a Feedback link. The main content area is titled "Snapshots" and displays a message: "Successfully deleted snapshot snap-00af83511018bb4ea." Below this, there is a table header for "Snapshots" with columns: Name, Snapshot ID, Size, Description, Storage..., Snapshot status, and Started. A message at the bottom of the table says, "You currently have no snapshots in this Region." At the bottom right of the main area, there are three small icons: a square with a minus sign, a square with a plus sign, and a square with a circular arrow. The footer of the page includes links for "Feedback", "English (US)", "© 2022, Amazon Web Services, Inc. or its affiliates.", "Privacy", "Terms", and "Cookie preferences".