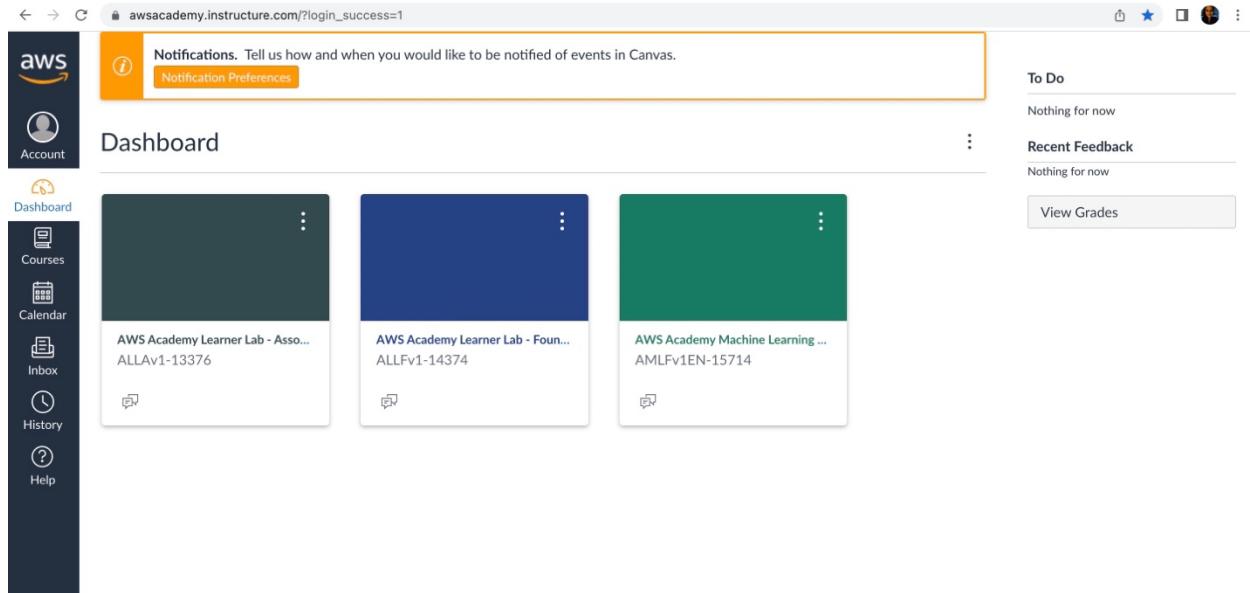


# Lab Assignment 2

Submitted by: Savleen Kaur  
10476867

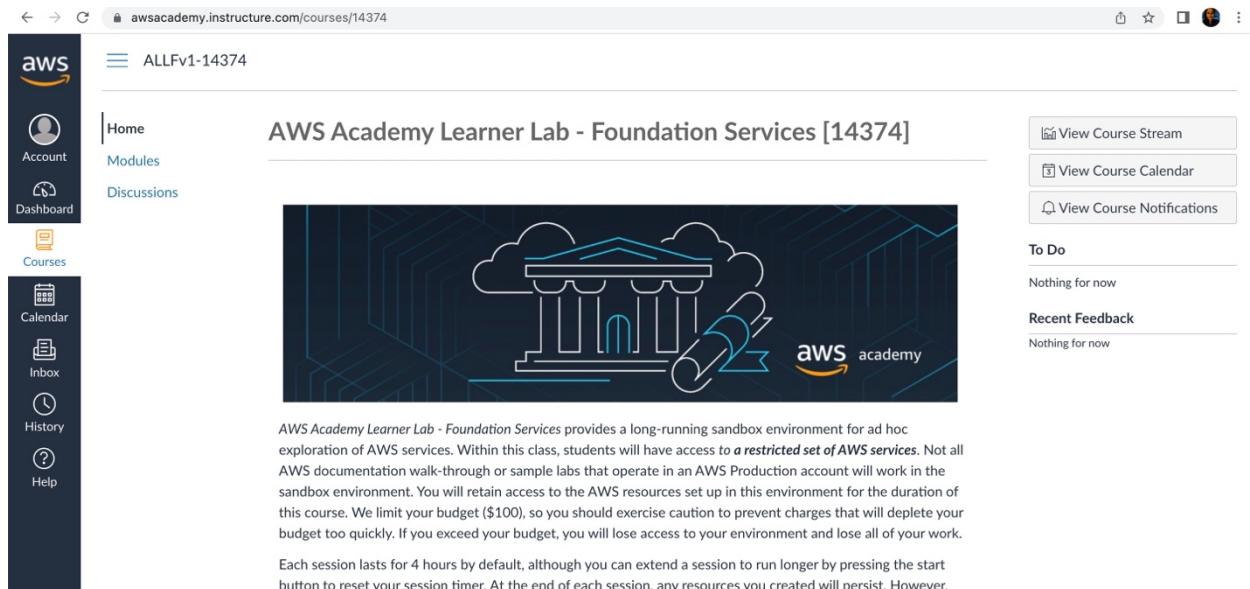
Open the learner lab and launch the lab



The screenshot shows the AWS Academy Dashboard. At the top, there is a notification bar with the message "Notifications. Tell us how and when you would like to be notified of events in Canvas." and a "Notification Preferences" button. Below the dashboard, there are three cards representing learner lab sessions:

- AWS Academy Learner Lab - Assisted ALLAv1-13376
- AWS Academy Learner Lab - Foundation Services ALLFv1-14374
- AWS Academy Machine Learning ... AMLFv1EN-15714

The sidebar on the left includes links for Account, Courses, Calendar, Inbox, History, and Help.



The screenshot shows the AWS Academy Learner Lab - Foundation Services [14374] course page. The top navigation bar includes Home, Modules, and Discussions. The main content area features a large graphic of a classical building with columns and clouds above it, with the "aws academy" logo at the bottom right. Below the graphic, there is descriptive text about the learner lab and session details.

**AWS Academy Learner Lab - Foundation Services [14374]**

AWS Academy Learner Lab - Foundation Services provides a long-running sandbox environment for ad hoc exploration of AWS services. Within this class, students will have access to a **restricted set of AWS services**. Not all AWS documentation walk-through or sample labs that operate in an AWS Production account will work in the sandbox environment. You will retain access to the AWS resources set up in this environment for the duration of this course. We limit your budget (\$100), so you should exercise caution to prevent charges that will deplete your budget too quickly. If you exceed your budget, you will lose access to your environment and lose all of your work.

Each session lasts for 4 hours by default, although you can extend a session to run longer by pressing the start button to reset your session timer. At the end of each session, any resources you created will persist. However,

The sidebar on the left includes links for Courses, Calendar, Inbox, History, and Help.

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

ALLFv1-1... > Modules > Learner La... > Learner Lab - Foundational Services

Home Modules Discussions

AWS ● Start Lab End Lab AWS Details Readme Reset

ddd\_v1\_w\_zhl\_1135907@runweb52102:~\$

EN-US -

## Learner Lab - Foundational Level

[Environment Overview](#)  
[Environment Navigation](#)  
[Access the AWS Management Console](#)  
[Region restriction](#)  
[Service usage and other restrictions](#)  
[Using the terminal in the browser](#)  
[Running AWS CLI commands](#)  
[Using the AWS SDK for Python](#)

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

ALLFv1-1... > Modules > Learner La... > Learner Lab - Foundational Services

Home Modules Discussions

AWS ● Start Lab End Lab AWS Details Readme Reset

EN-US -

## Learner Lab - Foundational Level

[Environment Overview](#)  
[Environment Navigation](#)  
[Access the AWS Management Console](#)  
[Region restriction](#)  
[Service usage and other restrictions](#)  
[Using the terminal in the browser](#)  
[Running AWS CLI commands](#)



The screenshot shows the AWS Academy Learner Lab interface. At the top, the URL is [awsacademy.instructure.com/courses/14374/modules/items/1212294](https://awsacademy.instructure.com/courses/14374/modules/items/1212294). The main navigation bar includes 'Home', 'Modules' (which is selected), and 'Discussions'. On the left, a sidebar lists 'Account', 'Dashboard', 'Courses', 'Calendar', 'Inbox', 'History', and 'Help'. The central workspace shows a terminal window with the command 'ddd\_v1\_w\_zhl\_1135907@runweb52102:~\$'. To the right, a large title box displays 'Learner Lab - Foundational Level' with a list of links: 'Environment Overview', 'Environment Navigation', 'Access the AWS Management Console', 'Region restriction', 'Service usage and other restrictions', and 'Using the terminal in the browser'.

After successful Login, we will be migrated to AWS Management Console. At the top left we will have Services. From the services dropdown select EC2 Instance.

The screenshot shows the AWS Management Console homepage. A banner at the top informs users that 'The new AWS Console Home will replace your existing experience soon' and provides a 'Switch now' button. The main content area features a large heading 'AWS Management Console'. Below it, there's a section for 'AWS services' showing 'Recently visited services' like EC2 and Simple Queue Service, and a link to 'All services'. To the right, there's a promotional box for the 'New AWS Console Home' with a preview image and a 'Switch now' button. Another section encourages users to 'Stay connected to your AWS resources on-the-go' by using the 'AWS Console Mobile App'.

On Clicking EC2 service, EC2 Dashboard will open. Now we have to click on “Launch Instance” under create instance tab

Start creating instances- Load Balancer and 4 servers named Server1, Server2, Server3 and Server4

You are using the following Amazon EC2 resources in the US East (N. Virginia) Region:

Instances (running)	0	Dedicated Hosts	0
Elastic IPs	0	Instances	0
Key pairs	1	Load balancers	0
Placement groups	0	Security groups	1
Snapshots	0	Volumes	0

**Launch instance**

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

**Service health**

Region: US East (N. Virginia) Status: This service is operating normally

**Explore AWS**

- Enable Best Price-Performance with AWS Graviton2
- AWS Graviton2 powered EC2 instances enable up to 40% better price performance for a broad spectrum of cloud workloads. [Learn more](#)
- Save up to 90% on EC2 with Spot Instances

Choose an Amazon Machine Image (AMI) page displays a list of basic configurations, called Amazon Machine Images (AMIs), that serve as templates for your instance. Select an Amazon Linux 2 AMI(HVM) – Kernel 4.14, SSD Volume Type. Notice that these AMIs are marked "Free tier eligible."

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.

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

**Quick Start**

- My AMIs
- AWS Marketplace
- Community AMIs
- Free tier only

**Amazon Linux** Free tier eligible

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

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 now under maintenance only mode and has been removed from this wizard.

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

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

Select  64-bit (x86)  64-bit (Arm)

Choose an Instance Type page, you can select the hardware configuration of your instance. Select the t2.micro instance type, which is selected by default. The t2.micro instance type is eligible for the free tier. In Regions where t2.micro is unavailable, you can use a t3.micro instance under the free tier. After selecting AMI, we have an option to choose an Instance Type. Select t2.micro (free tier eligibility) Click “Review and Launch”

**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
<input type="checkbox"/>	t2	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	t2	t2.micro <small>Free tier eligible</small>	1	1	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.small	1	2	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.large	2	8	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.xlarge	4	16	EBS only	-	Moderate	Yes
<input type="checkbox"/>	t2	t2.2xlarge	8	32	EBS only	-	Moderate	Yes
<input type="checkbox"/>	t2	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes

**Cancel** **Previous** **Review and Launch** **Next: Configure Instance Details**

**Step 7: Review Instance Launch**

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

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

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

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

**AMI Details** Edit AMI

**Amazon Linux 2 AMI (HVM) - 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

**Create a new Key pair**

**Cancel** **Previous** **Launch**

← → C us-east-1.console.aws.amazon.com/ec2/v2/home?region=us-east-1#LaunchInstanceWizard:  
Click to go back, hold to see history

Services Search for services, features, blogs, docs, and more [Option+S] N. Virginia vocabs/user1824796=skaur7@stevens.edu @ 7293-8314-3665 ▾

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

**Step 7: Review Instance Launch**

eligible 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**

Instance Type	ECUs	vCPUs
t2.micro	-	1

**Security Groups**

Security group name	Description
launch-wizard-1	launch-wizard-1 created

Type: SSH Protocol: TCP

**Instance Details**

**Storage**

**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](#).

Create a new key pair Key pair type RSA  ED25519 Key pair name Load Balancer Download Key Pair

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

Cancel Launch Instances

← → C us-east-1.console.aws.amazon.com/ec2/v2/home?region=us-east-1#LaunchInstanceWizard:  
Click to go back, hold to see history

Services Search for services, features, blogs, docs, and more [Option+S] N. Virginia vocabs/user1824796=skaur7@stevens.edu @ 7293-8314-3665 ▾

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

**Step 7: Review Instance Launch**

eligible 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**

Instance Type	ECUs	vCPUs
t2.micro	-	1

**Security Groups**

Security group name	Description
launch-wizard-1	launch-wizard-1 created

Type: SSH Protocol: TCP

**Instance Details**

**Storage**

**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](#).

Create a new key pair Key pair type RSA  ED25519 Key pair name Load Balancer Download Key Pair

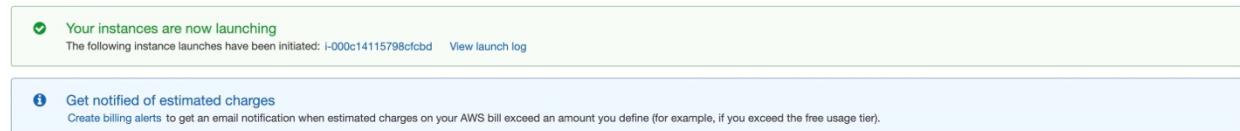
You have to download the **private key file** (\*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.

Cancel Launch Instances

A confirmation page lets you know that your instance is launching. Choose View Instances to close the confirmation page and return to the console.



## Launch Status



Your instances are now launching  
The following instance launches have been initiated: i-000c14115798cfcbd [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](#)

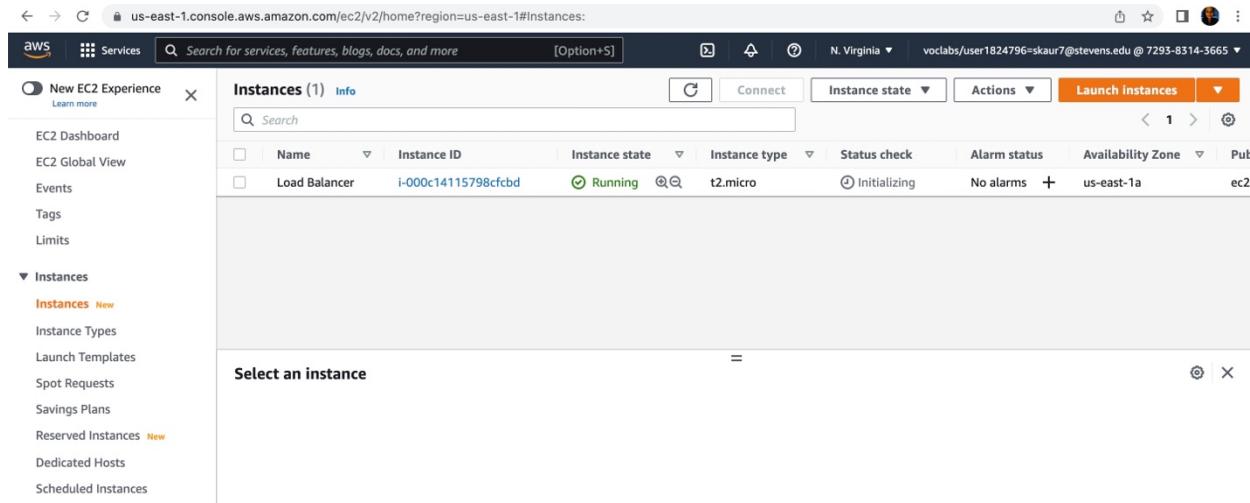
While your instances are launching you can also

- [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](#)

On the Instances screen, you can view the status of the launch. It takes a short time for an instance to launch. When you launch an instance, its initial state is pending. After the instance starts, its state changes to running and it receives a public DNS name.

Load balancer instance created and running.



Instances (1) [Info](#)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Publ
Load Balancer	i-000c14115798cfcbd	Running	t2.micro	Initializing	No alarms	us-east-1a	ec2-

Select an instance

Create another instances using the same steps

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.

**Try it now!**

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

**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"

Quick Start

My AMIs	Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type - ami-0c02fb55956c7d316 (64-bit x86) / ami-03190fe20ef6b1419 (64-bit Arm)
AWS Marketplace	Amazon Linux Free tier eligible
Community AMIs	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, Gilbc 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 now under maintenance only mode and has been removed from this wizard.
<input type="checkbox"/> Free tier only	Root device type: ebs   Virtualization type: hvm   ENA Enabled: Yes

Search by Systems Manager parameter

Cancel and Exit

Select

64-bit (x86)  
 64-bit (Arm)

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

**Step 2: Choose an Instance Type**

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

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

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

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

Cancel Previous Review and Launch Next: Configure Instance Details

**Step 7: Review Instance Launch**

**⚠ Improve your instances' security. Your security group, launch-wizard-2, is open to the world.**  
 Your instances may be accessible from any IP address. We recommend that you update your security group rules to allow access from known IP addresses only.  
 You can also open additional ports in your security group to facilitate access to the application or service you're running, e.g., HTTP (80) for web servers. [Edit security groups](#)

**AMI Details** [Edit AMI](#)

	<b>Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type - ami-0c02fb55956c7d316</b>
<small>Free tier eligible</small>	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	launch-wizard-2
---------------------	-----------------

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

**Step 7: Review Instance Launch**

**⚠ Improve your instances' security. Your instances may be accessible from any IP address. You can also open additional ports in your security group.**

**AMI Details**

	<b>Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type - ami-0c02fb55956c7d316</b>
<small>Free tier eligible</small>	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**

Instance Type	ECUs	vCPUs
t2.micro	-	1

**Security Groups**

**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.

[Cancel](#) [Launch Instances](#)

The screenshot shows the AWS Launch Status page. At the top, there's a header with the AWS logo, a search bar, and navigation links. Below the header, a green box displays a success message: "Your instances are now launching" followed by a link to the launch log. A blue box below it contains a notification about estimated charges. The main content area is titled "How to connect to your instances" and includes a note about instance launching and a link to view instances. It also lists helpful resources like the EC2 User Guide and Discussion Forum. A section for managing security groups is present, along with a "View Instances" button.

## Launch Status

**Your instances are now launching**  
The following instance launches have been initiated: i-01718885a7cba5f9e [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 screenshot shows the AWS EC2 Instances page. The left sidebar has a "New EC2 Experience" section with a "Learn more" link, followed by a list of services: EC2 Dashboard, EC2 Global View, Events, Tags, Limits, Instances (selected), Instances (New), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances (New), Dedicated Hosts, Scheduled Instances, and Capacity Reservations. The main content area is titled "Instances (5) Info". It shows a table with columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IPv4 DNS. The table lists five instances: Server 4, Server 1, Server 3, Load Balancer, and Server 2, all in the "Running" state. Below the table, a "Select an instance" section is visible.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
Server 4	i-0c803e6a3b17f3f81	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-1b	ec2-50-19-162-255.co...
Server 1	i-01718885a7cba5f9e	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-1b	ec2-54-234-136-250.co...
Server 3	i-0d176e2952239be5f	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-1b	ec2-3-87-167-229.com...
Load Balancer	i-000c14115798cfbd	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-1a	ec2-18-207-217-202.co...
Server 2	i-0df7c196d643f0d08	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-1a	ec2-3-87-51-35.comput...

Select an instance

Screenshot of the AWS EC2 Instances page showing the details for a Load Balancer instance.

**Instance summary for i-000c14115798cfcbd (Load Balancer)**

Instance ID	i-000c14115798cfcbd (Load Balancer)	Public IPv4 address	18.207.217.202   open address
IPv6 address	-	Instance state	Running
Hostname type	IP name: ip-172-31-83-112.ec2.internal	Private IP DNS name (IPv4 only)	ip-172-31-83-112.ec2.internal
Instance type	t2.micro	Elastic IP addresses	-
AWS Compute Optimizer finding	Opt-in to AWS Compute Optimizer for recommendations.   Learn more	IAM Role	-

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

**Instance details** | Info

## Add the inbound and outbound traffic security group for Load Balancer

Screenshot of the AWS Security Groups page showing the configuration of an inbound rule for a load balancer's security group.

**Edit inbound rules** | Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-012628e94229eb8c2	SSH	TCP	22	My IP	100.8.11.199/32
-	HTTPS	TCP	443	Custom	
-	HTTP	TCP	80	Custom	

**Add rule**

Cancel | Preview changes | Save rules

us-east-1.console.aws.amazon.com/ec2/v2/home?region=us-east-1#SecurityGroup:securityGroupId=sg-0e3f07767a15ead92

New EC2 Experience Learn more

Services Search for services, features, blogs, docs, and more [Option+S]

N. Virginia vocabs/user1824796=skaur7@stevens.edu @ 7293-8314-3665

EC2 Dashboard EC2 Global View Events Tags Limits Instances Instances New Instance Types Launch Templates Spot Requests Savings Plans Reserved Instances New Dedicated Hosts Scheduled Instances Capacity Reservations Images AMIs New AMI Catalog

**Details**

Security group name	Security group ID	Description	VPC ID
launch-wizard-1	sg-0e3f07767a15ead92	launch-wizard-1 created 2022-03-25T23:03:34.065-04:00	vpc-027c10a7d002e25e9
Owner	Inbound rules count	Outbound rules count	
729383143665	1 Permission entry	3 Permission entries	

Inbound rules Outbound rules Tags

You can now check network connectivity with Reachability Analyzer Run Reachability Analyzer

**Inbound rules (1/1)**

Name	Security group rule...	IP version	Type	Protocol	Port range
-	sgr-003213fbedd5e8aff	IPv4	All traffic	All	All

us-east-1.console.aws.amazon.com/ec2/v2/home?region=us-east-1#SecurityGroup:securityGroupId=sg-0e3f07767a15ead92

New EC2 Experience Learn more

Services Search for services, features, blogs, docs, and more [Option+S]

N. Virginia vocabs/user1824796=skaur7@stevens.edu @ 7293-8314-3665

EC2 Dashboard EC2 Global View Events Tags Limits Instances Instances New Instance Types Launch Templates Spot Requests Savings Plans Reserved Instances New Dedicated Hosts Scheduled Instances

**Inbound security group rules successfully modified on security group (sg-0e3f07767a15ead92 | launch-wizard-1)**

Details

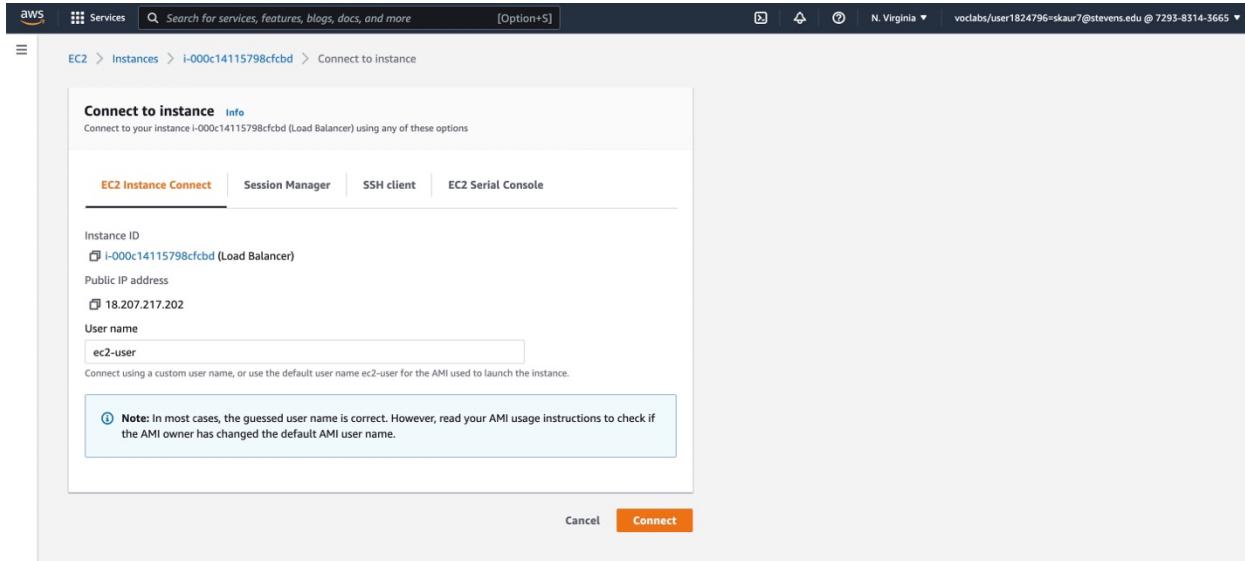
EC2 > Security Groups > sg-0e3f07767a15ead92 - launch-wizard-1 Actions ▾

**sg-0e3f07767a15ead92 - launch-wizard-1**

**Details**

Security group name	Security group ID	Description	VPC ID
launch-wizard-1	sg-0e3f07767a15ead92	launch-wizard-1 created 2022-03-25T23:03:34.065-04:00	vpc-027c10a7d002e25e9
Owner	Inbound rules count	Outbound rules count	
729383143665	3 Permission entries	1 Permission entry	

Inbound rules Outbound rules Tags



To verify that it is working, we need to connect it after the running status and then do nginx start by using following commands:

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

```
[ec2-user@ip-172-31-83-112 ~]$ 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-kernel-5.10/2/x86_64/primary_db
(4/9): amzn2extra-docker/2/x86_64/updateinfo
(5/9): amzn2extra-docker/2/x86_64/primary_db
(6/9): amzn2extra-kernel-5.10/2/x86_64/updateinfo
(7/9): amzn2extra-nginx1/2/x86_64/updateinfo
(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: libcrypto.so.1.1() (64bit) for package: 1:nginx-1.20.0-2.amzn2.0.4.x86_64
--> Processing Dependency: libcrypto.so.1.1() (64bit) for package: 1:nginx-1.20.0-2.amzn2.0.4.x86_64
--> Running transaction check
--> Package gperf-tools-libs.x86_64 0:2.6.1-1.amzn2 will be installed
--> Package nginx-filesystem.noarch 1:1.20.0-2.amzn2.0.4 will be installed
--> Package openssl11-libs.x86_64 1:1.1.1g-12.amzn2.0.7 will be installed
--> Processing Dependency: openssl11-pkcs11 for package: 1:openssl11-libs-1.1.1g-12.amzn2.0.7.x86_64
--> Running transaction check
--> Package openssl11-pkcs11.x86_64 0:0.4.10-6.amzn2.0.1 will be installed
```

Apple Chrome File Edit View History Bookmarks Profiles Tab Window Help Fri Mar 25 11:25:13 PM

Meet - n | Learner Lab | Connect to | i-000c14115798cfcbd | AWS-Labs/L | Lab Assignm | Lab Assignm | Stevens-Coll | Stevens-Coll | + |

us-east-1.console.aws.amazon.com/ec2/v2/connect/ec2-user/i-000c14115798cfcbd

```
--> 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: libcryptopp.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
--> Processing Dependency: libcryptopp.so.1.1() for package: 1:nginx-1.20.0-2.amzn2.0.4.x86_64
--> Running transaction check
--> Package gperftools-libs.x86_64 0:2.6.1-1.amzn2 will be installed
--> Package nginx-filesystem.noarch 1:1.20.0-2.amzn2.0.4 will be installed
--> Package openssl11-libs.x86_64 1:1.1.1g-12.amzn2.0.7 will be installed
--> Processing Dependency: openssl11-pkcs11 for package: 1:openssl11-libs-1.1.1g-12.amzn2.0.7.x86_64
--> Running transaction check
--> Package openssl11-pkcs11.x86_64 0:0.4.10-6.amzn2.0.1 will be installed
--> Finished Dependency Resolution
```

**Dependencies Resolved**

Package	Arch	Version	Repository	Size
Installing:				
nginx	x86_64	1:1.20.0-2.amzn2.0.4	amzn2extra-nginx1	579 k
Installing for dependencies:				
gperftools-libs	x86_64	2.6.1-1.amzn2	amzn2-core	274 k
nginx-filesystem	noarch	1:1.20.0-2.amzn2.0.4	amzn2extra-nginx1	23 k
openssl11-libs	x86_64	1:1.1.1g-12.amzn2.0.7	amzn2-core	1.4 M
openssl11-pkcs11	x86_64	0.4.10-6.amzn2.0.1	amzn2-core	61 k

**Transaction Summary**

```
Install 1 Package (+4 Dependent packages)
```

Total download size: 2.3 M  
Installed size: 6.6 M  
Is this ok [y/d/N]:

i-000c14115798cfcbd (Load Balancer)

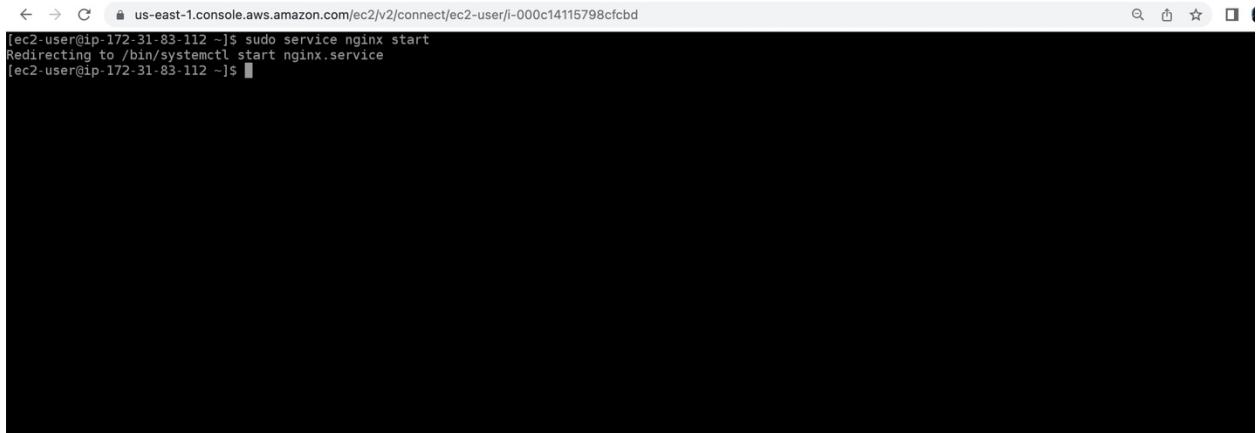
Public IPs: 18.207.217.202 Private IPs: 172.31.83.112



us-east-1.console.aws.amazon.com/ec2/v2/connect/ec2-user/i-000c14115798cfcbd

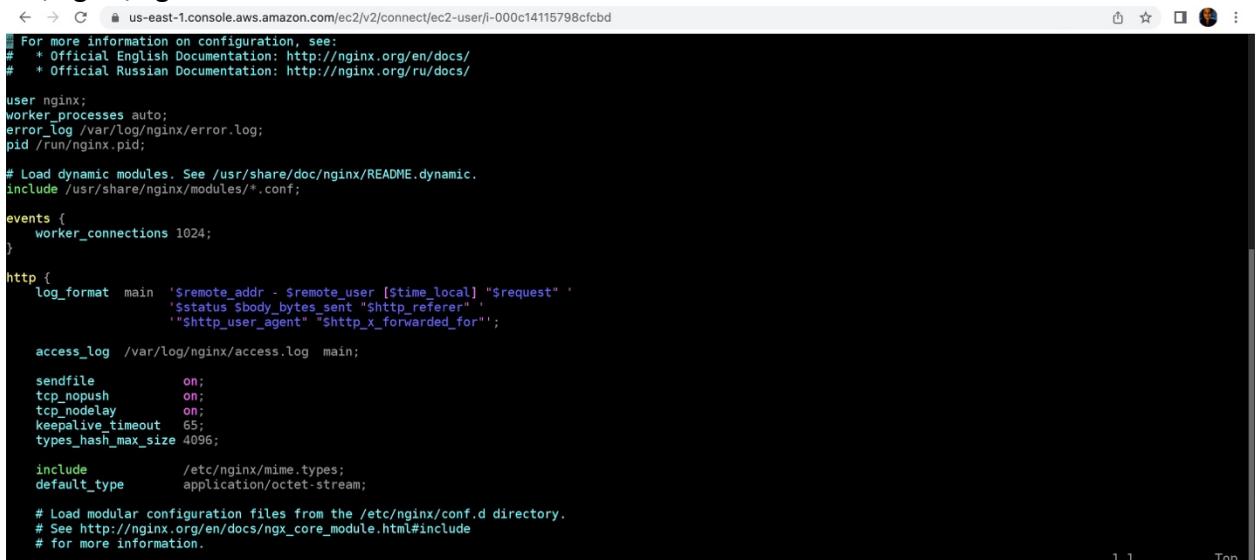
```
26 ecs available [ =stable ]
27 corretto8 available \
   [ =1.8.0_192 =1.8.0_202 =1.8.0_212 =1.8.0_222 =1.8.0_232
   =1.8.0_242 =stable ]
28 firecracker available [ =0.11 =stable ]
29 golang11 available \
   [ =1.11.3 =1.11.11 =1.11.13 =stable ]
30 squid4 available [ =4 =stable ]
32 lustre2.10 available \
   [ =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 ]
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-172-31-83-112 ~]$
```

Start nginx service for Load Balancer



```
[ec2-user@ip-172-31-83-112 ~]$ sudo service nginx start
Redirecting to /bin/systemctl start nginx.service
[ec2-user@ip-172-31-83-112 ~]$
```

To configure the load balancer, you need to edit the load balancer's configuration file `etc/nginx/nginx.conf`.



```
# 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 1024;
}

http {
    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.
}
```



**NGINX**

This is the DNS page without editing for Load Balancer

We need to validate other four server Other 4 ec2 instances are created using the same steps  
 Setting security group for server 1 Since server will receive traffic from load balancer so its source is load balancer security group id  
 Setup inbound and outbound traffic for all other servers

Inbound security group rules successfully modified on security group (sg-08a0ade3e7c564e4d | launch-wizard-2)

**sg-08a0ade3e7c564e4d - launch-wizard-2**

Details			
Security group name launch-wizard-2	Security group ID sg-08a0ade3e7c564e4d	Description launch-wizard-2 created 2022-03-25T23:06:56.882-04:00	VPC ID vpc-027c10a7d002e25e9
Owner 729383143665	Inbound rules count 3 Permission entries	Outbound rules count 1 Permission entry	

**Inbound rules**    **Outbound rules**    **Tags**

You can now check network connectivity with Reachability Analyzer    Run Reachability Analyzer

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

**Security details**

IAM Role -	Owner ID 729383143665	Launch time Sat Mar 26 2022 16:22:44 GMT-0400 (Eastern Daylight Time)
---------------	--------------------------	--

**Security groups**  
sg-0702f47b7c87561e5 (launch-wizard-3)

**Inbound rules**

Security group rule ID	Port range	Protocol	Source	Security groups
sgr-0272b0a07d28497de	22	TCP	0.0.0.0/0	launch-wizard-3

**Outbound rules**

Security group rule ID	Port range	Protocol	Destination	Security groups
sgr-01e49790e8e7cf635	All	All	0.0.0.0/0	launch-wizard-3

us-east-1.console.aws.amazon.com/ec2/v2/home?region=us-east-1#SecurityGroup:securityGroupId=sg-0702f47b7c87561e5

New EC2 Experience Learn more

Services Search for services, features, blogs, docs, and more [Option+S]

N. Virginia v vocabs/user1824796=skaur7@stevens.edu @ 7293-8314-3665 ▾

EC2 Dashboard

EC2 Global View

Events

Tags

Limits

Instances New

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

Images New

- AMIs New
- AMI Catalog

Inbound security group rules successfully modified on security group (sg-08a0ade3e7c564e4d | launch-wizard-2) ▾ Details

EC2 > Security Groups > sg-0702f47b7c87561e5 - launch-wizard-3

## sg-0702f47b7c87561e5 - launch-wizard-3

Actions ▾

Details

Security group name launch-wizard-3	Security group ID sg-0702f47b7c87561e5	Description launch-wizard-3 created 2022-03-25T23:10:17.000-04:00	VPC ID vpc-027c10a7d002e25e9
Owner 729383143665	Inbound rules count 1 Permission entry	Outbound rules count 1 Permission entry	

Inbound rules Outbound rules Tags

You can now check network connectivity with Reachability Analyzer Run Reachability Analyzer

Inbound rules (1/1) Manage tags Edit inbound rules

us-east-1.console.aws.amazon.com/ec2/v2/home?region=us-east-1#SecurityGroup:group-id=sg-0702f47b7c87561e5

New EC2 Experience Learn more

Services Search for services, features, blogs, docs, and more [Option+S]

N. Virginia v vocabs/user1824796=skaur7@stevens.edu @ 7293-8314-3665 ▾

EC2 Dashboard

EC2 Global View

Events

Tags

Limits

Instances New

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

Images New

- AMIs New

Inbound security group rules successfully modified on security group (sg-0702f47b7c87561e5 | launch-wizard-3) ▾ Details

EC2 > Security Groups > sg-0702f47b7c87561e5 - launch-wizard-3

## sg-0702f47b7c87561e5 - launch-wizard-3

Actions ▾

Details

Security group name launch-wizard-3	Security group ID sg-0702f47b7c87561e5	Description launch-wizard-3 created 2022-03-25T23:10:17.000-04:00	VPC ID vpc-027c10a7d002e25e9
Owner 729383143665	Inbound rules count 3 Permission entries	Outbound rules count 1 Permission entry	

Inbound rules Outbound rules Tags

You can now check network connectivity with Reachability Analyzer Run Reachability Analyzer

us-east-1.console.aws.amazon.com/ec2/v2/home?region=us-east-1#SecurityGroup:group-id=sg-053d3a67c6df13a3b

New EC2 Experience Learn more Services Search for services, features, blogs, docs, and more [Option+S] N. Virginia v vocabs/user1824796=skaur7@stevens.edu @ 7293-8314-3665 ▾

EC2 Dashboard EC2 Global View Events Tags Limits Instances Instances New Instance Types Launch Templates Spot Requests Savings Plans Reserved Instances New Dedicated Hosts Scheduled Instances Capacity Reservations Images AMIs New AMI Catalog

Inbound security group rules successfully modified on security group (sg-053d3a67c6df13a3b | launch-wizard-4) Details

EC2 > Security Groups > sg-053d3a67c6df13a3b - launch-wizard-4

## sg-053d3a67c6df13a3b - launch-wizard-4

Actions ▾

Details			
Security group name	Security group ID	Description	VPC ID
launch-wizard-4	sg-053d3a67c6df13a3b	launch-wizard-4 created 2022-03-25T23:11:22.618-04:00	vpc-027c10a7d002e25e9
Owner	Inbound rules count	Outbound rules count	
729383143665	3 Permission entries	1 Permission entry	

Inbound rules Outbound rules Tags

You can now check network connectivity with Reachability Analyzer Run Reachability Analyzer

Inbound rules (3) Manage tags Edit inbound rules

us-east-1.console.aws.amazon.com/ec2/v2/home?region=us-east-1#SecurityGroup:group-id=sg-048a49522660f7b19

New EC2 Experience Learn more Services Search for services, features, blogs, docs, and more [Option+S] N. Virginia v vocabs/user1824796=skaur7@stevens.edu @ 7293-8314-3665 ▾

EC2 Dashboard EC2 Global View Events Tags Limits Instances Instances New Instance Types Launch Templates Spot Requests Savings Plans Reserved Instances New Dedicated Hosts Scheduled Instances Capacity Reservations Images AMIs New AMI Catalog

Inbound security group rules successfully modified on security group (sg-048a49522660f7b19 | launch-wizard-5) Details

EC2 > Security Groups > sg-048a49522660f7b19 - launch-wizard-5

## sg-048a49522660f7b19 - launch-wizard-5

Actions ▾

Details			
Security group name	Security group ID	Description	VPC ID
launch-wizard-5	sg-048a49522660f7b19	launch-wizard-5 created 2022-03-25T23:11:58.682-04:00	vpc-027c10a7d002e25e9
Owner	Inbound rules count	Outbound rules count	
729383143665	3 Permission entries	1 Permission entry	

Inbound rules Outbound rules Tags

You can now check network connectivity with Reachability Analyzer Run Reachability Analyzer

Inbound rules (3) Manage tags Edit inbound rules

us-east-1.console.aws.amazon.com/ec2/v2/home?region=us-east-1#SecurityGroup:securityGroupId=sg-0e3f07767a15ead92

New EC2 Experience Learn more

Services Search for services, features, blogs, docs, and more [Option+S]

N. Virginia v vocabs/user1824796=skaur7@stevens.edu @ 7293-8314-3665 ▾

EC2 Dashboard

EC2 Global View

Events

Tags

Limits

Instances Instances New

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances New

Dedicated Hosts

Scheduled Instances

Capacity Reservations

Images AMIs New

AMI Catalog

Inbound rules Outbound rules Tags

You can now check network connectivity with Reachability Analyzer Run Reachability Analyzer

Outbound rules (3)

Name	Security group rule...	IP version	Type	Protocol	Port range
-	sgr-0a13d70aca0257e83	-	HTTPS	TCP	443
-	sgr-00177d058779f6d5c	-	HTTP	TCP	80
-	sgr-00e2407a473df9aa5	IPv4	All traffic	All	All

us-east-1.console.aws.amazon.com/ec2/v2/home?region=us-east-1#SecurityGroup:group-id=sg-08a0ade3e7c564e4d

New EC2 Experience Learn more

Services Search for services, features, blogs, docs, and more [Option+S]

N. Virginia v vocabs/user1824796=skaur7@stevens.edu @ 7293-8314-3665 ▾

EC2 > Security Groups > sg-08a0ade3e7c564e4d - launch-wizard-2 Actions ▾

Details

Security group name launch-wizard-2 Security group ID sg-08a0ade3e7c564e4d Description launch-wizard-2 created 2022-03-25T23:06:56.882-04:00 VPC ID vpc-027c10a7d002e25e9

Owner 729383143665 Inbound rules count 1 Permission entry Outbound rules count 1 Permission entry

Inbound rules Outbound rules Tags

You can now check network connectivity with Reachability Analyzer Run Reachability Analyzer

Inbound rules (1/1)

Name	Security group rule...	IP version	Type	Protocol	Port range
-	sgr-0c6aba2b8f0127032	IPv4	All traffic	All	All

After launching the instances, we need to install amazon-linux-extras (the Amazon Linux native software manager) to install Nginx on every instance and start the Nginx service and will see the Welcome message. By using following commands:

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

```
sudo service nginx start
```

```
=1.8.0_242 =stable ]  
28 firecracker available [ =0.11 =stable ]  
29 golang1.11 available \  
[ =1.11.3 =1.11.11 =1.11.13 =stable ]  
30 squid4 available [ =4 =stable ]  
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 dnsmasq2.85 available [ =stable ]  
[ec2-user@ip-172-31-16-227 ~]$
```

```
=1.8.0_242 =stable ]  
28 firecracker available [ =0.11 =stable ]  
29 golang1.11 available \  
[ =1.11.3 =1.11.11 =1.11.13 =stable ]  
30 squid4 available [ =4 =stable ]  
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 dnsmasq2.85 available [ =stable ]  
[ec2-user@ip-172-31-16-227 ~]$
```

i-01718885a7cba5f9e ( Server 1 )

Public IPs: 18.234.45.92 Private IPs: 172.31.16.227

meet.google.com is sharing your screen. Stop sharing Hide

```

← → C us-east-1.console.aws.amazon.com/ec2/v2/connect/ec2-user/i-01718885a7cba5f9e
=1.8.0_242 =stable ]
28 firecracker available [ =0.11 =stable ]
29 golang1.11 available \
[ =1.11.3 =1.11.11 =1.11.13 =stable ]
30 squid4 available [ =4 =stable ]
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 dnsmasq2.85 available [ =stable ]
[ec2-user@ip-172-31-16-227 ~]$ sudo service nginx start
Redirecting to /bin/systemctl start nginx.service
[ec2-user@ip-172-31-16-227 ~]$
[ec2-user@ip-172-31-16-227 ~]$
[ec2-user@ip-172-31-16-227 ~]$ █

```

101718885a7cba5f9e (Server 1)

The screenshot shows the AWS Services console with the search bar set to "Search for services, features, blogs, docs, and more". The user is signed in as "voclabs/user1824796=skaur7@stevens.edu @ 7293-8314-3665". The main navigation menu on the left includes "New EC2 Experience", "EC2 Dashboard", "EC2 Global View", "Events", "Tags", "Limits", "Instances", "Images", and "AMIs".

The "Details" section for the security group "launch-wizard-3" displays the following information:

- Security group name:** launch-wizard-3
- Security group ID:** sg-0702f47b7c87561e5
- Description:** launch-wizard-3 created 2022-03-25T23:10:17.000-04:00
- VPC ID:** vpc-027c10a7d002e25e9
- Owner:** 729383143665
- Inbound rules count:** 1 Permission entry
- Outbound rules count:** 1 Permission entry

The "Inbound rules" tab is selected, showing one rule:

Name	Security group rule...	IP version	Type	Protocol	Port range
-	sgr-0a4f63750f262316d	-	All traffic	All	All

A message at the bottom of the page says: "You can now check network connectivity with Reachability Analyzer" with a "Run Reachability Analyzer" button.

AWS Services Search for services, features, blogs, docs, and more [Option+S] N. Virginia v vocabs/user1824796=skaur7@stevens.edu @ 7293-8314-3665 ▾

**New EC2 Experience** Learn more

**Details**

Security group name	Security group ID	Description	VPC ID
launch-wizard-3	sg-0702f47b7c87561e5	launch-wizard-3 created 2022-03-25T23:10:17.000-04:00	vpc-027c10a7d002e25e9
Owner	Inbound rules count	Outbound rules count	
729383143665	1 Permission entry	1 Permission entry	

**Instances**

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

**Images**

- AMIs New

**Inbound rules** **Outbound rules** Tags

You can now check network connectivity with Reachability Analyzer Run Reachability Analyzer

**Outbound rules (1/1)**

<input checked="" type="checkbox"/>	Name	Security group rule...	IP version	Type	Protocol	Port range
<input checked="" type="checkbox"/>	-	sgr-01e49790e8e7cf635	IPv4	All traffic	All	All

Chrome File Edit View History Bookmarks Tab Window Help 84% Mon Mar 28 11:41:48 AM

us-east-1.console.aws.amazon.com/ec2/v2/connect/ec2-user/i-000c14115798cfcb

```
[ =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 ]
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 ]
```

i-000c14115798cfcb (Load Balancer)  
Public IPs: 54.236.213.113 Private IPs: 172.31.83.112

```
← → C 🔒 us-east-1.console.aws.amazon.com/ec2/v2/connect/ec2-user/i-0d176e2952239be5f
--> Processing Dependency: libprofiler.so.0()(64bit) for package: 1:nginx-1.20.0-2.amzn2.0.4.x86_64
--> Processing Dependency: libcrypto.so.1.1()(64bit) for package: 1:nginx-1.20.0-2.amzn2.0.4.x86_64
--> Running transaction check
--> Package gperftools-libs.x86_64 0:2.6.1-1.amzn2 will be installed
--> Package nginx-filesystem.noarch 1:1.20.0-2.amzn2.0.4 will be installed
--> Package openssl11-libs.x86_64 1:1.1.1g-12.amzn2.0.7 will be installed
--> Processing Dependency: openssl11-pkcs11 for package: 1:openssl11-libs-1.1.1g-12.amzn2.0.7.x86_64
--> Package openssl11-pkcs11.x86_64 0:0.4.10-6.amzn2.0.1 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package          Arch      Version       Repository   Size
=====
Installing:
nginx            x86_64    1:1.20.0-2.amzn2.0.4      amzn2extra-nginx1  579 k
Installing for dependencies:
gperftools-libs  x86_64    2.6.1-1.amzn2          amzn2-core           274 k
nginx-filesystem noarch   1:1.20.0-2.amzn2.0.4      amzn2extra-nginx1  23 k
openssl11-libs   x86_64    1:1.1.1g-12.amzn2.0.7  amzn2-core           1.4 M
openssl11-pkcs11 x86_64    0.4.10-6.amzn2.0.1     amzn2-core           61 k
=====
Transaction Summary
=====
Install 1 Package (+4 Dependent packages)

Total download size: 2.3 M
Installed size: 6.6 M
Is this ok [y/d/N]:
```

i-0d176e2952239be5f (Server 3)

Public IPs: 54.87.220.156 Private IPs: 172.31.26.34

```
← → C us-east-1.console.aws.amazon.com/ec2/v2/connect/ec2-user/i-0d176e2952239be5f
[ =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 nginx=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 dnsmasq2.85 available [ =stable ]
[ec2-user@ip-172-31-26-34 ~]$
```

i-0d176e2952239be5f (Server 3)

Public IPs: 54.87.220.156 Private IPs: 172.31.26.34

```
← → C us-east-1.console.aws.amazon.com/ec2/v2/connect/ec2-user/i-0df7c196d643f0d08
[ =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 nginx=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 dnsmasq2.85 available [ =stable ]
[ec2-user@ip-172-31-92-248 ~]$
```

i-0df7c196d643f0d08 (Server 2)

Public IPs: 52.91.194.61 Private IPs: 172.31.92.248

```
[ =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 ]
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-172-31-24-120 ~]$
```

i-0c803e6a3b17f3f81 (Server 4)

Public IPs: 54.226.3.80 Private IPs: 172.31.24.120

```
[ =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 ]
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-172-31-16-227 ~]$
```

i-01718885a7cba5f9e ( Server 1 )

Public IPs: 18.234.45.92 Private IPs: 172.31.16.227

```

← → C us-east-1.console.aws.amazon.com/ec2/v2/connect/ec2-user/i-000c14115798cfcb
34 lynis available [ =stable ]
35 kernel-ng available [ =stable ]
36 BCC available [ =0.x =stable ]
37 mono available [ =5.x =stable ]
38 nginx1=latest enabled [ =stable ]
39 ruby2.6 available [ =2.6 =stable ]
40 mock available [ =stable ]
41 postgresql11 available [ =11 =stable ]
42 php7.4 available [ =stable ]
43 livepatch available [ =stable ]
44 python3.8 available [ =stable ]
45 haproxy2 available [ =stable ]
46 collectd available [ =stable ]
47 aws-nitro-enclaves-cli available [ =stable ]
48 R4 available [ =stable ]
49 kernel-5.4 available [ =stable ]
50 selinux-ng available [ =stable ]
51 php8.0 available [ =stable ]
52 tomcat9 available [ =stable ]
53 unbound1.13 available [ =stable ]
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-172-31-83-112 ~]$ sudo service nginx start
Redirecting to /bin/systemctl start nginx.service
[ec2-user@ip-172-31-83-112 ~]$ 

```

i-000c14115798cfcb (Load Balancer)

Public IPs: 54.236.213.113 Private IPs: 172.31.83.112

After that we need to go to cd /usr/share/nginx/html and by command sudo vim index.html we need to Use a text editor (such as vi) to Add the header with server ID in the index.html file like following: <h1>Server 1</h1>, <h1>Server 2</h1>, <h1>Server 3</h1>, <h1>Server 4</h1>

Edit the script for Server1

```

← → C us-east-1.console.aws.amazon.com/ec2/v2/connect/ec2-user/i-01718885a7cba5f9e
}
a:hover img {
    border: 2px solid #294172;
}
.logos {
    margin: 1em;
    text-align: center;
}
/*]]>*/
</style>
</head>

<body>
    <h1>Welcome to <strong>server 1</strong> on Amazon Linux!</h1>
    <div class="content">
        <p>This page is used to test the proper operation of the
        <strong>nginx</strong> HTTP server after it has been
        installed. If you can read this page, it means that the
        web server installed at this site is working
        properly.</p>
        <div class="alert">
            <h2>Website Administrator</h2>
            <div class="content">
                <p>This is the default <tt>index.html</tt> page that
                is distributed with <strong>nginx</strong> on
                Amazon Linux. It is located in
                <tt>/usr/share/nginx/html</tt>.</p>

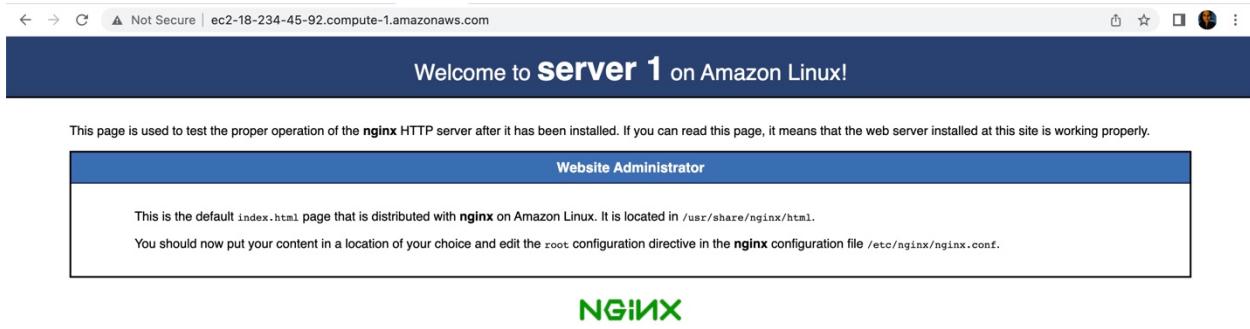
```

"index.html" 111L, 3523B written

[ec2-user@ip-172-31-16-227 html]\$

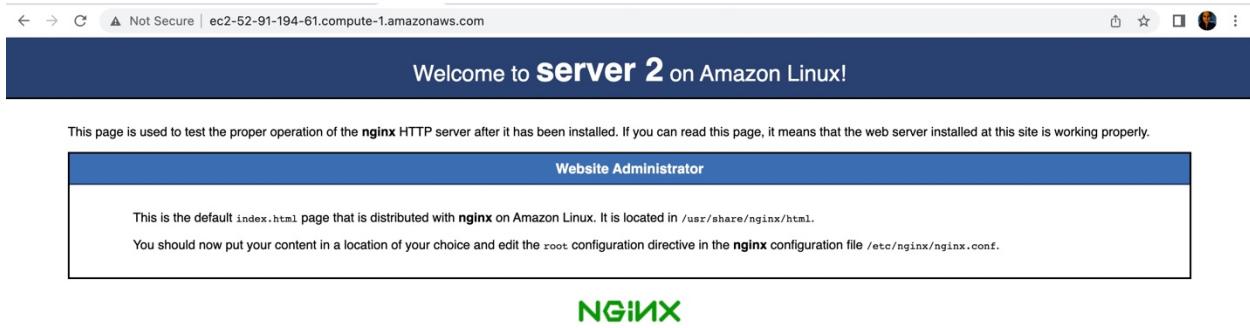
i-01718885a7cba5f9e ( Server 1)

Public IPs: 18.234.45.92 Private IPs: 172.31.16.227



## Server 1 from DNS

Edit the same script for Server 2 and load DNS Page



## Server 3 script edited and Server 3 from DNS

The screenshot shows a web browser window with the URL "Not Secure | ec2-54-87-220-156.compute-1.amazonaws.com". The main content area displays the text "Welcome to **server 3** on Amazon Linux!". Below this, a message states: "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." A "Website Administrator" section contains instructions: "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`". At the bottom right, the Nginx logo is visible.

## Server 4 script edited and Server 4 from DNS

The screenshot shows a web browser window with the URL "Not Secure | ec2-54-226-3-80.compute-1.amazonaws.com". The main content area displays the text "Welcome to **server 4** on Amazon Linux!". Below this, a message states: "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." A "Website Administrator" section contains instructions: "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`". At the bottom right, the Nginx logo is visible.

## Editing the Load Balancer script

The screenshot shows a terminal window with the command "us-east-1.console.aws.amazon.com/ec2/v2/connect/ec2-user|i-000c14115798cfcbd". The terminal content displays the Nginx configuration script:

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

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

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

    # Load configuration files for the default server block.
    include /etc/nginx/default.d/*.conf;
    location = / {
        proxy_pass http://myapp;
    }

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

    error_page 500 502 503 504 /50x.html;
    location = /50x.html {
    }
}
"nginx.conf" 102L, 2758B written
[ec2-user@ip-172-31-83-112 nginx]$
```

i-000c14115798cfcbd (Load Balancer)

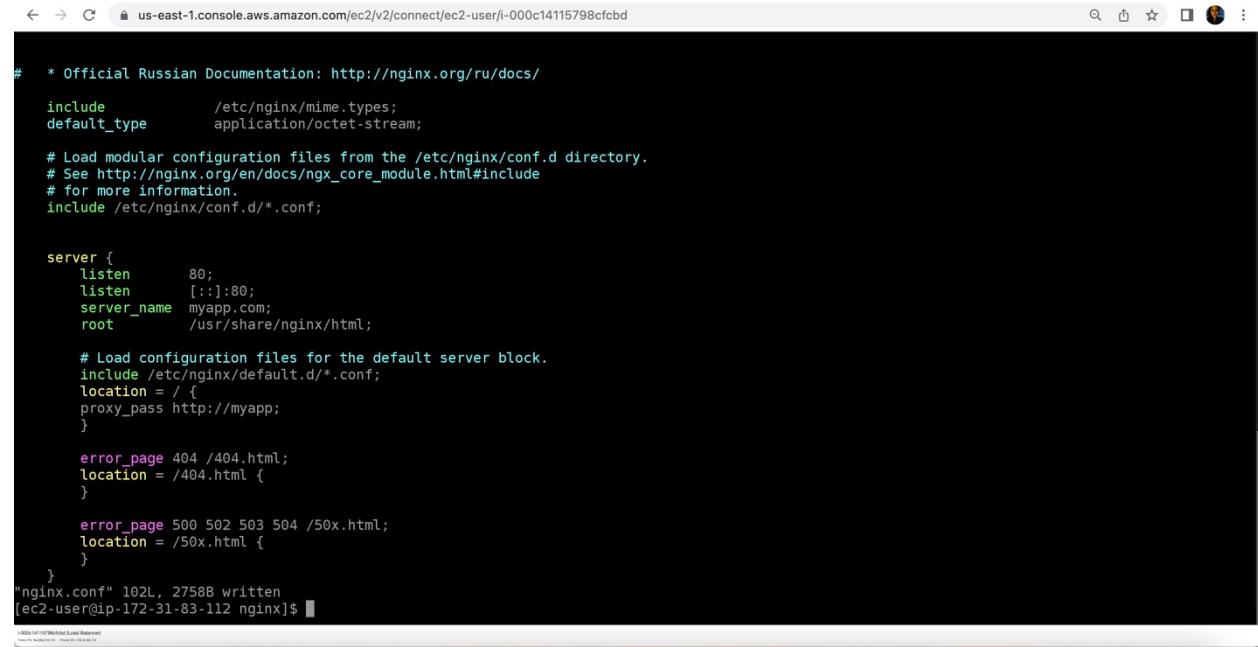
Public IPs: 54.236.213.113 Private IPs: 172.31.83.112

Replacing the text provided in the lab instruction

```
events {
    worker_connections 768;
}

http {
    upstream myapp {
        #ip_hash;
        server ec2-3-84-126-109.compute-1.amazonaws.com weight=1;
        server ec2-3-93-81-244.compute-1.amazonaws.com weight=1;
        server ec2-54-226-84-198.compute-1.amazonaws.com weight=1;
        server ec2-54-145-113-195.compute-1.amazonaws.com weight=1;
    }

    server {
        listen 80;
        server_name myapp.com;
        location / {
            proxy_pass http://myapp;
        }
    }
}
```



The screenshot shows a terminal window with the URL `us-east-1.console.aws.amazon.com/ec2/v2/connect/ec2-user/i-000c14115798cfcbd` in the address bar. The terminal displays the Nginx configuration file (`/etc/nginx/nginx.conf`) with several changes made:

```
# * Official Russian Documentation: http://nginx.org/ru/docs/
include          /etc/nginx/mime.types;
default_type     application/octet-stream;
# Load modular configuration files from the /etc/nginx/conf.d directory.
# See http://nginx.org/en/docs/ngx_core_module.html#include
# for more information.
include /etc/nginx/conf.d/*.conf;

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

    # Load configuration files for the default server block.
    include /etc/nginx/default.d/*.conf;
    location = / {
        proxy_pass http://myapp;
    }

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

    error_page 500 502 503 504 /50x.html;
    location = /50x.html {
    }
}
"nginx.conf" 102L, 2758B written
[ec2-user@ip-172-31-83-112 nginx]$
```

A small footer at the bottom left of the terminal window reads: "1000c14115798cfcbd Load Balancer".

```

events {
    worker_connections 768;
}

http {
    upstream myapp {
        #ip_hash;

        server ec2-54-175-42-137.compute-1.amazonaws.com weight=1;
        server ec2-54-175-42-137.compute-1.amazonaws.com weight=1;
        server ec2-44-203-179-87.compute-1.amazonaws.com weight=1;
        server ec2-18-233-224-22.compute-1.amazonaws.com weight=1;
    }

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

-- INSERT --

```

```

#       ssl_certificate "/etc/pki/nginx/server.crt";
#       ssl_certificate_key "/etc/pki/nginx/private/server.key";
#       ssl_session_cache shared:SSL:1m;
#       ssl_session_timeout 10m;
#       ssl_ciphers PROFILE=SYSTEM;
#       ssl_prefer_server_ciphers on;
#
#       # Load configuration files for the default server block.
#       include /etc/nginx/default.d/*.conf;
#
#       error_page 404 /404.html;
#           location = /40x.html {
#       }
#
#       error_page 500 502 503 504 /50x.html;
#           location = /50x.html {
#       }
#   }

}

"nginx.conf" 108L, 2729B written
[ec2-user@ip-172-31-94-252 nginx]$ 

```

i-0a97c08c3f28a1a47 (Load Balancer)

Public IPs: 52.91.14.6 Private IPs: 172.31.94.252

Reload nginx as conf file is changed so this step is required otherwise those changes will not work. Load balancer console through public dns name

The screenshot shows a web browser window with the following details:

- Address Bar:** Not Secure | ec2-18-234-45-92.compute-1.amazonaws.com
- Title Bar:** Welcome to nginx on Amazon Linux!
- Content Area:**
  - A message: "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."
  - A "Website Administrator" button.
  - A note: "This is the default index.html page that is distributed with nginx on Amazon Linux. It is located in /usr/share/nginx/html."
  - A note: "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

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

```
[ec2-user@ip-172-31-93-169 nginx]$ sudo systemctl restart nginx
[ec2-user@ip-172-31-93-169 nginx]$ curl ec2-54-164-96-229.compute-1.amazonaws.com
```

[India Intranet Home...](#) [Xzone – Deals & Co...](#) [Xzone – Deals & Co...](#) [React JS Calculator...](#) [Microsoft's Kuber...](#) [How to create Votin...](#) [Dashboard – Unna...](#) [\(571\) React Tutorial...](#) [»](#)

```
#      server_name _;
[ec2-user@ip-172-31-93-169 nginx]$ sudo systemctl restart nginx
[ec2-user@ip-172-31-93-169 nginx]$ curl ec2-54-164-96-229.compute-1.amazonaws.com
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>Test Page for the Nginx HTTP Server on Amazon Linux</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <style type="text/css">
      /*<![CDATA[*/
      body {
        background-color: #fff;
        color: #000;
        font-size: 0.9em;
        font-family: sans-serif, helvetica;
        margin: 0;
        padding: 0;
      }
      :link {
        color: #c00;
      }
      :visited {
        color: #c00;
      }
    <![CDATA]]>
```

```
        }
        :visited {
            color: #c00;
        }
        a:hover {
            color: #f50;
        }
        h1 {
            text-align: center;
            margin: 0;
            padding: 0.6em 2em 0.4em;
            background-color: #294172;
            color: #fff;
            font-weight: normal;
            font-size: 1.75em;
            border-bottom: 2px solid #000;
        }
        h1 strong {
            font-weight: bold;
            font-size: 1.5em;
        }
        h2 {
            text-align: center;
            background-color: #3C6EB4;
            font-size: 1.1em;
        }
```

```
}
```

---

```
.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>Server 1</strong> on Amazon Linux!</h1>

<div class="content">
    <p>This page is used to test the proper operation of the
    <strong>nginx</strong> HTTP server after it has been
    installed. If you can read this page, it means that the
    web server installed at this site is working
    properly.</p>
<div class="alert">
    <h2>Website Administrator</h2>
    <div class="content">
        <p>This is the default <tt>index.html</tt> page that
        is distributed with <strong>nginx</strong> on
        Amazon Linux. It is located in
        <tt>/usr/share/nginx/html</tt>.</p>
        <p>You should now put your content in a location of
        your choice and edit the <tt>root</tt> configuration
        directive in the <strong>nginx</strong>
        configuration file</p>
    </div>
</div>
```

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

```
</style>
</head>

<body>
    <h1>Welcome to <strong>Server 3</strong> on Amazon Linux!</h1>
    <div class="content">
        <p>This page is used to test the proper operation of the
        <strong>nginx</strong> HTTP server after it has been
        installed. If you can read this page, it means that the
        web server installed at this site is working
        properly.</p>

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

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

i-0232bcc0cde5c2e4f (Load Balancer)

```
<body>
    <h1>Welcome to <strong>Server 4</strong> on Amazon Linux!</h1>
    <div class="content">
        <p>This page is used to test the proper operation of the
        <strong>nginx</strong> HTTP server after it has been
        installed. If you can read this page, it means that the
        web server installed at this site is working
        properly.</p>

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

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

i-0232bcc0cde5c2e4f (Load Balancer)

Public IPs: 54.205.11.237 Private IPs: 172.31.91.115

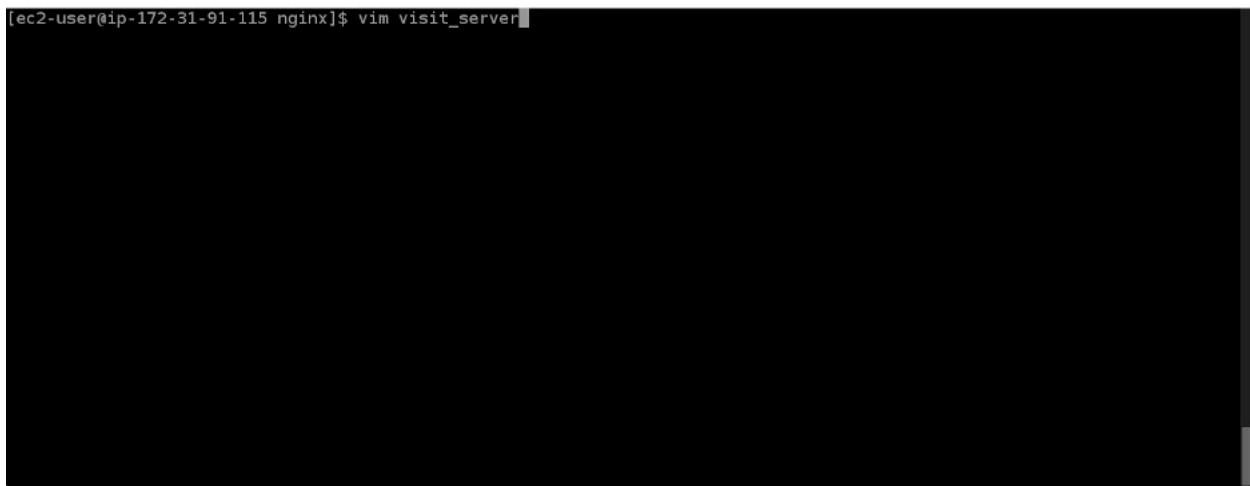
After making changes to load balancer, and after reloading it we get servers as output as per its traffic

Steps to collect information on visits to your website using Amazon EC2 instance

② Setting up Visit Server tool to track the distribution of the load. This tool visits the cluster 2000 times and returns the visit count on each server

Command: \$ vim visit\_server

```
[ec2-user@ip-172-31-91-115 nginx]$ vim visit_server
```

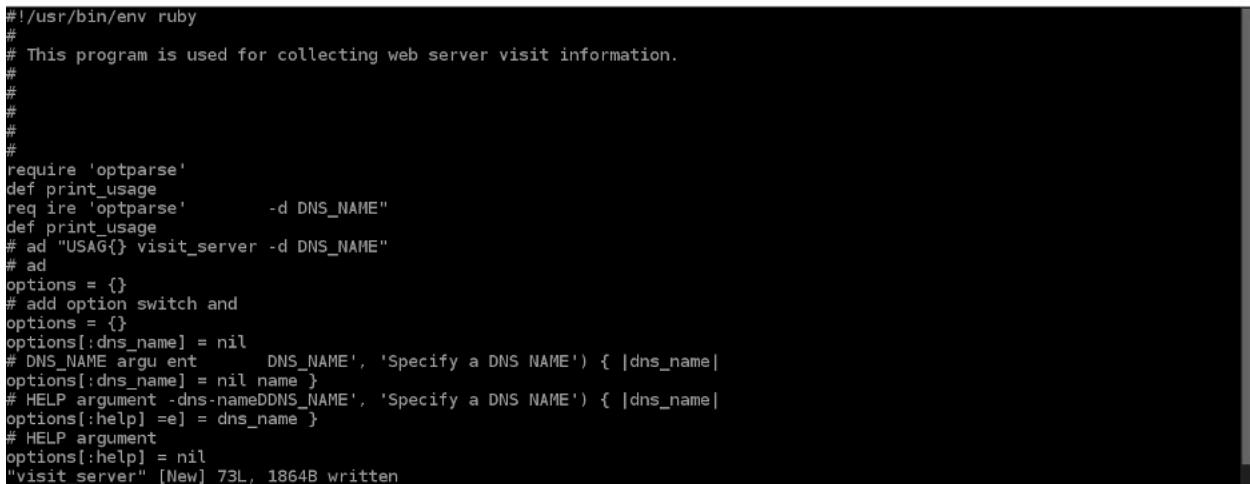


i-0232bcc0cde5c2e4f (Load Balancer)

Public IPs: 54.205.11.237 Private IPs: 172.31.91.115

We need to write the script in the visit server tool (Ex: ruby). After connecting to load balancer, we need to write script which is in ruby. After writing and saving it, we need to run command “Sudo yum install ruby”

```
#!/usr/bin/env ruby
#
# This program is used for collecting web server visit information.
#
#
#
#
require 'optparse'
def print_usage
req ire 'optparse'      -d DNS_NAME"
def print_usage
# ad "USAG{} visit_server -d DNS_NAME"
# ad
options = {}
# add option switch and
options = {}
options[:dns_name] = nil
# DNS_NAME argu ent      DNS_NAME', 'Specify a DNS NAME') { |dns_name|
options[:dns_name] = nil name }
# HELP argument -dns-nameDNS_NAME', 'Specify a DNS NAME') { |dns_name|
options[:help] = e} = dns_name }
# HELP argument
options[:help] = nil
"visit server" [New] 73L, 1864B written
```



i-0232bcc0cde5c2e4f (Load Balancer)

Public IPs: 54.205.11.237 Private IPs: 172.31.91.115

```
#!/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
#visit_server" 73L, 1864B
```

5,1 Top

i-0232bcc0cde5c2e4f (Load Balancer)

Public IPs: 54.205.11.237 Private IPs: 172.31.91.115

```
# HELP argument
options[:help] = nil
opts.on('-h', '--help', 'Display usage') { |help| options[:help] = help }
end
option_parser.parse!
# verify arguments
if options[:dns_name] then
dns_name = options[:dns_name]
else
puts "Please set a balancer's DNS."
print_usage
exit
end
if options[:help] then
print_usage
exit
end
# Keep STDOUT
$orig_stdout = $stdout
# redirect stdout to /dev/null
#$stdout = File.new('/dev/null', 'w')
server1_visit_count = 0
server2_visit_count = 0
server3_visit_count = 0
```

37,1 36%

i-0232bcc0cde5c2e4f (Load Balancer)

Public IPs: 54.205.11.237 Private IPs: 172.31.91.115

```

#orig_stdout = $stdout
# redirect stdout to /dev/null
#$stdout = File.new('/dev/null', 'w')
server1_visit_count = 0
server2_visit_count = 0
server2_visit_count = 0
server2_visit_count = 0
server3_visit_count = 0
p starting to visit load balancing server
s starting to visit load balancing server
puts "Starting to visit load balancing server"
# visit load balancer
p = `curl -s #{dns_name}`
if o =~ /server\s*1/i
server1_visit_count += 1
elsif o =~ /server\s*2/i
server2_visit_count += 1
elsif o =~ /server\s*3/i
server3_visit_count += 1
elsif o =~ /server\s*4/i
server4_visit_count += 1
end
print "."
end

```

42,1 73%

i-0232bcc0cde5c2e4f (Load Balancer)

Public IPs: 54.205.11.237 Private IPs: 172.31.91.115

```

if o =~ /server\s*1/i
server1_visit_count += 1
elsif o =~ /server\s*2/i
server2_visit_count += 1
elsif o =~ /server\s*3/i
server3_visit_count += 1
elsif o =~ /server\s*4/i
server4_visit_count += 1
end
print "."
end
# redirect output to stdout
#$stdout = orig_stdout
# print visit information
puts
puts '-----'
puts ' Summary'
puts '-----'
puts "Server1 visit counts : " + server1_visit_count.to_s
puts "Server2 visit counts : " + server2_visit_count.to_s
puts "Server3 visit counts : " + server3_visit_count.to_s
puts "Server4 visit counts : " + server4_visit_count.to_s
puts "Total visit counts : " + (server1_visit_count + server2_visit_count + server3_visit_count +
server4_visit_count).to_s

```

73,1 Bot

i-0232bcc0cde5c2e4f (Load Balancer)

Public IPs: 54.205.11.237 Private IPs: 172.31.91.115

After installing ruby and writing the script, we further need to run command visit\_server.rb -d LOAD\_BALANCER\_DNS\_NAME

After Ruby installation run the visit\_server file by providing loadbalance public dns name

```
[ec2-user@ip-172-31-91-115 ~]$ ruby visit_server.rb -d ec2-54.205.11.237.compute-1.amazonaws.com
```

i-0232bcc0cde5c2e4f (Load Balancer)

Public IPs: 54.205.11.237 Private IPs: 172.31.91.115

```
sudo: apt: command not found
[ec2-user@ip-172-31-91-115 ~]$ 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
```

i-0232bcc0cde5c2e4f (Load Balancer)

Public IPs: 54.205.11.237 Private IPs: 172.31.91.115

Now, we will trace the load balancing server load distribution for 3 scenarios by changing the server weight in the nginx.conf file (which we edited in previous step).

- o Scenario #1 – Server1 weight=1, Server2 weight=1, Server3 weight=1, Server4 weight=1
  - o Scenario #2 – Server1 weight=1, Server2 weight=2, Server3 weight=3, Server4 weight=1
  - o Scenario #3 – Server1 weight=1, Server2 weight=2, Server3 weight=1, Server4 weight=2
- ¶ First our weights to the server is 1 so visit counts will be same for all servers.
- ¶ Tracing load balancer for Scenario #1 – Server1 weight=1, Server2 weight=1, Server3 weight=1, Server4 weight=1

```
[ec2-user@ip-172-31-91-115 ~]$ ruby visit_server.rb -d ec2-54-205-11-237.compute-1.amazonaws.com
```

i-0232bcc0cde5c2e4f (Load Balancer)

Public IPs: 54.205.11.237 Private IPs: 172.31.91.115

```
events {
    worker_connections 768;
}

http {
    upstream myapp {
        upstream myapp {

            server ec2-3-84-126-109.compute-1.amazonaws.com weight=1;
            server ec2-3-84-126-109.compute-1.amazonaws.com weight=1;
            server ec2-3-93-81-244.compute-1.amazonaws.com weight=1;
            server ec2-54-145-113-195.compute-1.amazonaws.com weight=1;

        }
        log_format main '$remote_addr - $remote_user [$time_local] "$request" '
        log_format main '$remote_addr - $remote_user [$time_local] "$request" '
                    '$status $body_bytes_sent "$http_referer" '

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

17,1 14%

i-0232bcc0cde5c2e4f (Load Balancer)

Public IPs: 54.205.11.237 Private IPs: 172.31.91.115

```
Summary
Server1 visit counts : 500
Server2 visit counts : 500
Server3 visit counts : 500
Server4 visit counts : 500
Total visit counts : 2000
[ec2-user@ip-172-31-91-115 ~]$
```

i-0232bcc0cde5c2e4f (Load Balancer)

Public IPs: 54.206.11.237 Private IPs: 172.31.91.115

The visit counts for each server in the load balancer (like Server 1, Server 2, Server 3, and Server 4) are 500. As the weight for each server is 1. As the weight are 1, 1, 1, and 1.

Tracing load balancer for Scenario #2 – Server1 weight=1, Server2 weight=2, Server3 weight=3, Server4 weight=4

```
# 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 ec2-3-84-126-109.compute-1.amazonaws.com weight=1;
        server ec2-3-93-81-244.compute-1.amazonaws.com weight=2;
        server ec2-54-226-84-198.compute-1.amazonaws.com weight=3;
        server ec2-54-145-113-195.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"';
}

-- INSERT --
```

i-0232bcc0cde5c2e4f (Load Balancer)

Public IPs: 54.206.11.237 Private IPs: 172.31.91.115

24, 60 11%

```
Summary
-----
Server1 visit counts : 200
Server2 visit counts : 400
Server3 visit counts : 600
Server4 visit counts : 800
Total visit counts : 2000
[ec2-user@ip-172-31-91-115 ~]$
```

i-0232bcc0cde5c2e4f (Load Balancer)

Public IPs: 54.205.11.237 Private IPs: 172.31.91.115

The visit counts for each server in the load balancer (like Server 1, Server 2, Server 3, and Server 4) is 200, 400, 600, and 800 respectively. As the weight are 1, 2, 3, and 4.

🔗 Tracing load balancer for Scenario #3 – Server1 weight=1, Server2 weight=2, Server3 weight=1, Server4 weight=2

```
# Load dynamic modules. See /usr/share/doc/nginx/README.dynamic.
include /usr/share/nginx/modules/*.conf;

events {
    worker_connections 768;
}

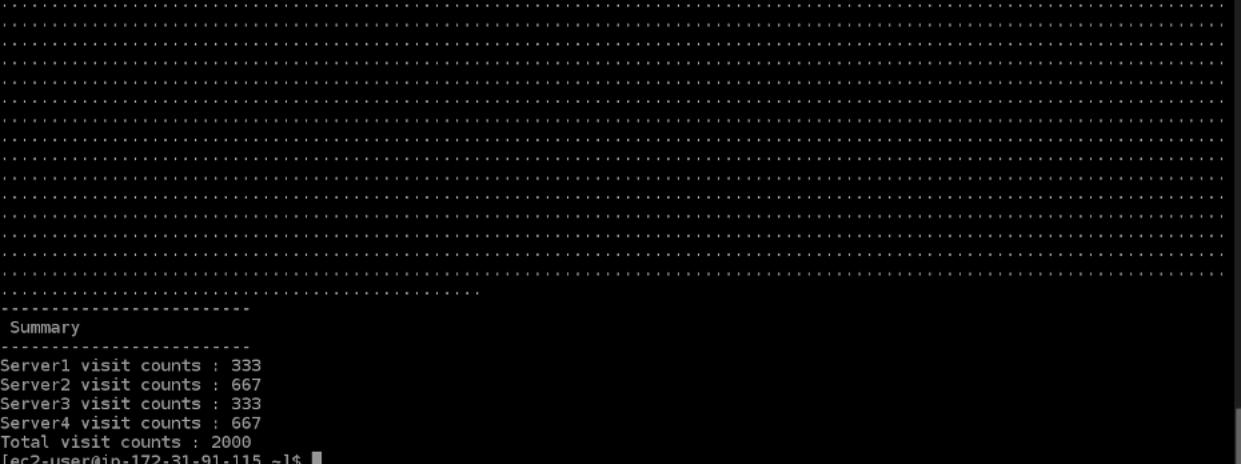
http {
    upstream myapp {
        #ip_hash;

        server ec2-3-84-126-109.compute-1.amazonaws.com weight=1;
        server ec2-3-93-81-244.compute-1.amazonaws.com weight=2;
        server ec2-54-226-84-198.compute-1.amazonaws.com weight=1;
        server ec2-54-145-113-195.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"';
}
"nginx.conf" 100L, 2681B written
[ec2-user@ip-172-31-91-115 nginx]$
```

i-0232bcc0cde5c2e4f (Load Balancer)

Public IPs: 54.205.11.237 Private IPs: 172.31.91.115



```
India Intranet Hom... Xzone - Deals & Co... Xzone - Deals & Co... React JS Calculator... Microsoft's Kuber... How to create Votin... Dashboard - Unna... (571) React Tutorial...
Summary
-----
Server1 visit counts : 333
Server2 visit counts : 667
Server3 visit counts : 333
Server4 visit counts : 667
Total visit counts : 2000
[ec2-user@ip-172-31-91-115 ~]$
```

i-0232bcc0cde5c2e4f (Load Balancer)  
Public IPs: 54.206.11.237 Private IPs: 172.31.91.115

The visit counts for each server in the load balancer (like Server 1, Server 2, Server 3, and Server 4) is 333, 667, 333, and 667 respectively. As the weight are 1, 2, 1, and 2.

After having deployed the balancer

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

Steps to tcpdump Analysis using Amazon EC2 instance

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

Running tcpdump at the command prompt on the terminal would be of no help. Since tcpdump analyzes tcp/ip packets to and from the host, running the command on a remote terminal would go on indefinitely and would be full of the packet information mostly pertaining to packets exchanged while running the command itself. Hence the command is run with its output redirected to a remote file. When we analyze the file contents, we see the first few lines being sent by the remote host to my desktop. Then the remote host issues an ARP request to get its own mac address. Since I had made a http request to the load balancer while tcpdump was running (on the load balancer), there are packet information from my local desktop to the load balancer, then from the load balancer to one of the servers, and finally back all the way to my local desktop.

```
[ec2-user@ip-172-31-91-115 ~]$ 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-172-31-91-115 ~]$ sudo tcpdump --interface any
```

i-0232bcc0cde5c2e4f (Load Balancer)

Public IPs: 54.205.11.237 Private IPs: 172.31.91.115

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

It will start collecting packets

```
17:12:49.410163 IP ip-172-31-91-115.ec2.internal.ssh > ec2-18-206-107-26.compute-1.amazonaws.com.53385: Flags [P.], seq 52
525584:52525856, ack 5313, win 246, options [nop,nop,TS val 3139285418 ecr 2326485074], length 272
17:12:49.410247 IP ip-172-31-91-115.ec2.internal.ssh > ec2-18-206-107-26.compute-1.amazonaws.com.53385: Flags [P.], seq 52
525856:52526128, ack 5313, win 246, options [nop,nop,TS val 3139285418 ecr 2326485074], length 272
17:12:49.410328 IP ip-172-31-91-115.ec2.internal.ssh > ec2-18-206-107-26.compute-1.amazonaws.com.53385: Flags [P.], seq 52
526128:52526400, ack 5313, win 246, options [nop,nop,TS val 3139285418 ecr 2326485074], length 272
17:12:49.410359 IP ec2-18-206-107-26.compute-1.amazonaws.com.53385 > ip-172-31-91-115.ec2.internal.ssh: Flags [.,], ack 525
24560, win 4353, options [nop,nop,TS val 2326485074 ecr 3139285417], length 0
17:12:49.410409 IP ip-172-31-91-115.ec2.internal.ssh > ec2-18-206-107-26.compute-1.amazonaws.com.53385: Flags [P.], seq 52
526400:52526880, ack 5313, win 246, options [nop,nop,TS val 3139285418 ecr 2326485074], length 480
17:12:49.410475 IP ip-172-31-91-115.ec2.internal.ssh > ec2-18-206-107-26.compute-1.amazonaws.com.53385: Flags [P.], seq 52
526880:52527152, ack 5313, win 246, options [nop,nop,TS val 3139285418 ecr 2326485074], length 272
17:12:49.410542 IP ip-172-31-91-115.ec2.internal.ssh > ec2-18-206-107-26.compute-1.amazonaws.com.53385: Flags [P.], seq 52
527152:52527424, ack 5313, win 246, options [nop,nop,TS val 3139285418 ecr 2326485074], length 272
17:12:49.410608 IP ip-172-31-91-115.ec2.internal.ssh > ec2-18-206-107-26.compute-1.amazonaws.com.53385: Flags [P.], seq 52
527424:52527696, ack 5313, win 246, options [nop,nop,TS val 3139285418 ecr 2326485074], length 272
17:12:49.410681 IP ip-172-31-91-115.ec2.internal.ssh > ec2-18-206-107-26.compute-1.amazonaws.com.53385: Flags [P.], seq 52
527696:52527968, ack 5313, win 246, options [nop,nop,TS val 3139285418 ecr 2326485074], length 272
17:12:49.410730 IP ec2-18-206-107-26.compute-1.amazonaws.com.53385 > ip-172-31-91-115.ec2.internal.ssh: Flags [.,], ack 525
26128, win 4353, options [nop,nop,TS val 2326485074 ecr 3139285418], length 0
17:12:49.410750 IP ip-172-31-91-115.ec2.internal.ssh > ec2-18-206-107-26.compute-1.amazonaws.com.53385: Flags [P.], seq 52
527968:52528240, ack 5313, win 246, options [nop,nop,TS val 3139285419 ecr 2326485074], length 272
17:12:49.410814 IP ip-172-31-91-115.ec2.internal.ssh > ec2-18-206-107-26.compute-1.amazonaws.com.53385: Flags [P.], seq 52
528240:52528720, ack 5313, win 246, options [nop,nop,TS val 3139285419 ecr 2326485074], length 480
17:12:49.410879 IP ip-172-31-
```

i-0232bcc0cde5c2e4f (Load Balancer)

Public IPs: 54.205.11.237 Private IPs: 172.31.91.115

My system's domain name – ip-172-31-91-115.ec2.internal.ssh

What I have observed from the packets from my machine's domain, they are going to ec2 server through seq and then sending ack. But if we want to capture packets in a file we should use sudo tcpdump -w filename.pcap

```
Last login: Mon Mar 28 17:00:17 2022 from ec2-18-206-107-26.compute-1.amazonaws.com
[ec2-user@ip-172-31-91-115 ~]$ sudo tcpdump -w lab2packets.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
```

```
https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-91-115 ~]$ sudo tcpdump -w lab2packets.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
```

i-0232bcc0cde5c2e4f (Load Balancer)

Public IPs: 54.205.11.237 Private IPs: 172.31.91.115

```
Last login: Mon Mar 28 17:00:17 2022 from ec2-18-206-107-26.compute-1.amazonaws.com
[ec2-user@ip-172-31-91-115 ~]$ sudo tcpdump -w lab2packets.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
```

```
https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-91-115 ~]$ sudo tcpdump -w lab2packets.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
^C1051748 packets captured
1052319 packets received by filter
562 packets dropped by kernel
[ec2-user@ip-172-31-91-115 ~]$
```

i-0232bcc0cde5c2e4f (Load Balancer)

Public IPs: 54.205.11.237 Private IPs: 172.31.91.115

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

```

17:40:14.728365 IP ec2-18-206-107-26.compute-1.amazonaws.com.53385 > ip-172-31-91-115.ec2.internal.ssh: Flags [.], ack 861
74128, win 4353, options [nop,nop,TS val 2328130384 ecr 3140930718], length 0
17:40:14.728426 IP ip-172-31-91-115.ec2.internal.ssh > ec2-18-206-107-26.compute-1.amazonaws.com.53385: Flags [P.], seq 86
176528:86177024, ack 8833, win 246, options [nop,nop,TS val 3140930719 ecr 2328130384], length 496
17:40:14.728511 IP ip-172-31-91-115.ec2.internal.ssh > ec2-18-206-107-26.compute-1.amazonaws.com.53385: Flags [P.], seq 86
177024:86177312, ack 8833, win 246, options [nop,nop,TS val 3140930719 ecr 2328130384], length 288
17:40:14.728540 IP ec2-18-206-107-26.compute-1.amazonaws.com.53385 > ip-172-31-91-115.ec2.internal.ssh: Flags [.], ack 861
74912, win 4353, options [nop,nop,TS val 2328130384 ecr 3140930718], length 0
17:40:14.728601 IP ip-172-31-91-115.ec2.internal.ssh > ec2-18-206-107-26.compute-1.amazonaws.com.53385: Flags [P.], seq 86
177312:86177808, ack 8833, win 246, options [nop,nop,TS val 3140930719 ecr 2328130384], length 496
17:40:14.728684 IP ip-172-31-91-115.ec2.internal.ssh > ec2-18-206-107-26.compute-1.amazonaws.com.53385: Flags [P.], seq 86
177808:86178096, ack 8833, win 246, options [nop,nop,TS val 3140930719 ecr 2328130384], length 288
17:40:14.728715 IP ec2-18-206-107-26.compute-1.amazonaws.com.53385 > ip-172-31-91-115.ec2.internal.ssh: Flags [.], ack 861
75696, win 4353, options [nop,nop,TS val 2328130384 ecr 3140930718], length 0
17:40:14.728726 IP ip-172-31-91-115.ec2.internal.ssh > ec2-18-206-107-26.compute-1.amazonaws.com.53385: Flags [P.], seq 86
178096:86178592, ack 8833, win 246, options [nop,nop,TS val 3140930719 ecr 2328130384], length 496
17:40:14.728855 IP ip-172-31-91-115.ec2.internal.ssh > ec2-18-206-107-26.compute-1.amazonaws.com.53385: Flags [P.], seq 86
178592:86178880, ack 8833, win 246, options [nop,nop,TS val 3140930719 ecr 2328130384], length 288
17:40:14.728881 IP ec2-18-206-107-26.compute-1.amazonaws.com.53385 > ip-172-31-91-115.ec2.internal.ssh: Flags [.], ack 861
76528, win 4353, options [nop,nop,TS val 2328130384 ecr 3140930718], length 0
17:40:14.728949 IP ip-172-31-91-115.ec2.internal.ssh > ec2-18-206-107-26.compute-1.amazonaws.com.53385: Flags [P.], seq 86
178880:86179376, ack 8833, win 246, options [nop,nop,TS val 3140930719 ecr 2328130384], length 496
17:40:14.729033 IP ip-172-31-91-115.ec2.internal.ssh > ec2-18-206-107-26.compute-1.amazonaws.com.53385: Flags [P.], seq 86
179376:86179664, ack 8833, win 246, options [nop,nop,TS val 3140930719 ecr 2328130384], length 288
17:40:14.729047 IP ec2-18-206-107-26.compute-1.amazonaws.com.53385 > ip-172-31-91-115.ec2.intern

```

i-0232bcc0cd5c2e4f (Load Balancer)

Public IPs: 54.205.11.237 Private IPs: 172.31.91.115

As an additional step, perform the EC2 backup and restore:

1. Register an AMI with your Load Balancer instance image and launch a new EC2 instance.
2. Verify that the new instance has all the files created in the load balancer instance. List and explain all the steps in achieving this goal.

Perform the EC2 backup and restore To take the backup of instance first we need to take the snapshot of the existing load balancer instance from EBS-Volumes

Savings Plans																																																																
Reserved Instances <small>New</small>																																																																
Dedicated Hosts																																																																
Scheduled Instances																																																																
Capacity Reservations																																																																
<b>Images</b>																																																																
AMIs <small>New</small>																																																																
AMI Catalog																																																																
<b>Elastic Block Store</b>																																																																
Volumes <small>New</small>																																																																
Snapshots <small>New</small>																																																																
Lifecycle Manager <small>New</small>																																																																
<b>Network &amp; Security</b>																																																																
Security Groups																																																																
Elastic IPs																																																																
Placement Groups																																																																
Key Pairs																																																																
Volumes (5)																																																																
<input type="text"/> Filter volumes																																																																
<table border="1"> <thead> <tr> <th><input type="checkbox"/></th> <th>Name</th> <th>Volume ID</th> <th>Type</th> <th>Size</th> <th>IOPS</th> <th>Throughput</th> <th>Snapshot</th> <th>Created</th> </tr> </thead> <tbody> <tr> <td><input type="checkbox"/></td> <td>-</td> <td>vol-0aa160d0fb17a1ae1</td> <td>gp2</td> <td>8 GiB</td> <td>100</td> <td>-</td> <td>snap-0ef4350...</td> <td>2022/03/28 11</td> </tr> <tr> <td><input type="checkbox"/></td> <td>-</td> <td>vol-0bac953546aacb0d4</td> <td>gp2</td> <td>8 GiB</td> <td>100</td> <td>-</td> <td>snap-0ef4350...</td> <td>2022/03/28 11</td> </tr> <tr> <td><input type="checkbox"/></td> <td>-</td> <td>vol-03b5a2735498f01d6</td> <td>gp2</td> <td>8 GiB</td> <td>100</td> <td>-</td> <td>snap-0ef4350...</td> <td>2022/03/28 10</td> </tr> <tr> <td><input type="checkbox"/></td> <td>-</td> <td>vol-09478a6e2e5d70395</td> <td>gp2</td> <td>8 GiB</td> <td>100</td> <td>-</td> <td>snap-0ef4350...</td> <td>2022/03/28 10</td> </tr> <tr> <td><input type="checkbox"/></td> <td>-</td> <td>vol-01a752bf291dd9907</td> <td>gp2</td> <td>8 GiB</td> <td>100</td> <td>-</td> <td>snap-0ef4350...</td> <td>2022/03/28 11</td> </tr> </tbody> </table>											<input type="checkbox"/>	Name	Volume ID	Type	Size	IOPS	Throughput	Snapshot	Created	<input type="checkbox"/>	-	vol-0aa160d0fb17a1ae1	gp2	8 GiB	100	-	snap-0ef4350...	2022/03/28 11	<input type="checkbox"/>	-	vol-0bac953546aacb0d4	gp2	8 GiB	100	-	snap-0ef4350...	2022/03/28 11	<input type="checkbox"/>	-	vol-03b5a2735498f01d6	gp2	8 GiB	100	-	snap-0ef4350...	2022/03/28 10	<input type="checkbox"/>	-	vol-09478a6e2e5d70395	gp2	8 GiB	100	-	snap-0ef4350...	2022/03/28 10	<input type="checkbox"/>	-	vol-01a752bf291dd9907	gp2	8 GiB	100	-	snap-0ef4350...	2022/03/28 11
<input type="checkbox"/>	Name	Volume ID	Type	Size	IOPS	Throughput	Snapshot	Created																																																								
<input type="checkbox"/>	-	vol-0aa160d0fb17a1ae1	gp2	8 GiB	100	-	snap-0ef4350...	2022/03/28 11																																																								
<input type="checkbox"/>	-	vol-0bac953546aacb0d4	gp2	8 GiB	100	-	snap-0ef4350...	2022/03/28 11																																																								
<input type="checkbox"/>	-	vol-03b5a2735498f01d6	gp2	8 GiB	100	-	snap-0ef4350...	2022/03/28 10																																																								
<input type="checkbox"/>	-	vol-09478a6e2e5d70395	gp2	8 GiB	100	-	snap-0ef4350...	2022/03/28 10																																																								
<input type="checkbox"/>	-	vol-01a752bf291dd9907	gp2	8 GiB	100	-	snap-0ef4350...	2022/03/28 11																																																								
Select a volume above																																																																

<https://us-east-1.console.aws.amazon.com/ec2/v2/home?region=us-east-1#Volu...>

© 2022, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

	Availability Zone	Volume state	Alarm status	Attached Instances	Volume state...	Encryption
11:10 GMT-4	us-east-1d	✓ In-use	No alarms	+ i-0df38a0237449f23d (Ser...	✓ Okay	Not encrypted
11:14 GMT-4	us-east-1d	✓ In-use	No alarms	+ i-0297318cc08cedf21 (Ser...	✓ Okay	Not encrypted
10:57 GMT-4	us-east-1c	✓ In-use	No alarms	+ i-0232bcc0cd5e5c2e4f (Lo...	✓ Okay	Not encrypted
11:14 GMT-4	us-east-1c	✓ In-use	No alarms	+ i-011e9e2a704360870 (Se...	✓ Okay	Not encrypted
11:15 GMT-4	us-east-1c	✓ In-use	No alarms	+ i-04daa510e1b4fc4a3 (Ser...	✓ Okay	Not encrypted

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

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
	Load Balancer	i-0232bcc0cd5e5c2e4f	✓ Running	t2.micro	✓ 2/2 checks passed	No alarms	+ us-east-1c

To create snapshot from action select create snapshot

The screenshot shows the AWS Management Console with the navigation bar on the left. Under the 'Elastic Block Store' section, 'Volumes' is selected. A context menu is open over the second volume in the list, showing options like 'Modify volume', 'Create snapshot', 'Create snapshot lifecycle policy', 'Delete volume', etc.

Name	Volume ID	Type	Size	IOPS
-	vol-0aa160d0fb17a1ae1	gp2	8 GiB	100
-	vol-0bac953546aacb0d4	gp2	8 GiB	100
<input checked="" type="checkbox"/>	vol-03b5a2735498f01d6	gp2	8 GiB	100
-	vol-09478a6e2e5d70395	gp2	8 GiB	100
-	vol-01a752bf291dd9907	gp2	8 GiB	100

Volume ID: vol-03b5a2735498f01d6

Details | Status checks | Monitoring | Tags

## Provide the description for snapshot

The screenshot shows the 'Create snapshot' wizard. The current step is 'Details'. The volume ID is set to 'vol-03b5a2735498f01d6'. The description field contains 'Snapshot for Load Balance'. The encryption setting is 'Not encrypted'.

EC2 > Volumes > vol-03b5a2735498f01d6 > Create snapshot

Create snapshot [Info](#)

Create a point-in-time snapshot to back up the data on an Amazon EBS volume to Amazon S3.

**Details**

Volume ID  
vol-03b5a2735498f01d6

Description  
Add a description for your snapshot  
 255 characters maximum.

Encryption [Info](#)  
Not encrypted

**Tags** [Info](#)

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter.

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

The screenshot shows the AWS EC2 Volumes page. A green banner at the top indicates a successful snapshot creation: "Successfully created snapshot snap-0f4ed45d2321d7451 from volume vol-03b5a2735498f01d6. If you need your snapshot to be immediately available consider using Fast Snapshot Restore." Below the banner is a table titled "Volumes (5)". The table has columns for Name, Volume ID, Type, Size, IOPS, Throughput, Snapshot, and Created. The data includes:

Name	Volume ID	Type	Size	IOPS	Throughput	Snapshot	Created
-	vol-Uaa1b0d0tb17a1ae1	gp2	8 GiB	100	-	snap-0ef4350...	2022/03/28
-	vol-0bac953546aacb0d4	gp2	8 GiB	100	-	snap-0ef4350...	2022/03/28
-	vol-03b5a2735498f01d6	gp2	8 GiB	100	-	snap-0ef4350...	2022/03/28
-	vol-09478a6e2e5d70395	gp2	8 GiB	100	-	snap-0ef4350...	2022/03/28
-	vol-01a752bf291dd9907	gp2	8 GiB	100	-	snap-0ef4350...	2022/03/28

Below the table, a message says "Select a volume above".

Click on the snapshots, there will be many snapshots provided. Under snapshots we can see the newly created snapshots

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

The screenshot shows the AWS EC2 Instances page. The left sidebar shows the navigation menu. The main table is titled "Instances (1/5) Info". It lists five instances: "Load Balancer", "Server 4", "Server 2", "Server 1", and "Server 3". The "Load Balancer" instance is selected, indicated by a checked checkbox. The table columns include Name, Instance ID, Instance state, Instance type, Status check, Alarm status, and Availability Zone. The "Load Balancer" row shows "Running", "t2.micro", "2/2 checks passed", "No alarms", and "us-east-1c". Below the table, a modal window is open for the "Load Balancer" instance, titled "Instance: i-0232bcc0cde5c2e4f (Load Balancer)". The modal shows the "Details" tab selected, displaying the instance ID, public IPv4 address (54.205.11.237), private IPv4 address (172.31.91.115), and other details like IPv6 address and instance state.

New EC2 Experience [Learn more](#)

EC2 Dashboard

EC2 Global View

Events

Tags

Limits

Instances

Instances [New](#)

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances [New](#)

Dedicated Hosts

Scheduled Instances

Capacity Reservations

Instances (1/5) Info

C Connect Instance state Actions Launch Instances

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
Load Balancer	i-0232bcc0cde5c2e4f	Running	t2.micro	2/2 checks passed	No alarms	us-east-1c
Server 4	i-04daa510e1b4fc4a3	Running	t2.micro	2/2 checks passed	No alarms	us-east-1c
Server 2	i-011e9e2a704360870	Running	t2.micro	2/2 checks passed	No alarms	us-east-1c
Server 1	i-0df3ba0237449f23d	Running	t2.micro	2/2 checks passed	No alarms	us-east-1d
Server 3	i-0297518cc08ccff21	Running	t2.micro	2/2 checks passed	No alarms	us-east-1d

Instance: i-0232bcc0cde5c2e4f (Load Balancer)

Select an instance above

Details Security Networking Storage Status checks Monitoring Tags

Instance summary Info

Instance ID i-0232bcc0cde5c2e4f (Load Balancer)	Public IPv4 address 54.205.11.237   <a href="#">open address</a>	Private IPv4 addresses 172.31.91.115
IPv6 address	Instance state	Public IPv4 DNS

New EC2 Experience [Learn more](#)

EC2 Dashboard

EC2 Global View

Events

Tags

Limits

Instances

Instances [New](#)

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances [New](#)

Dedicated Hosts

Scheduled Instances

Capacity Reservations

Instances (1/5) Info

C Connect Instance state Actions Launch Instances

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
Load Balancer	i-0232bcc0cde5c2e4f	Stopping	t2.micro	2/2 checks passed	No alarms	us-east-1c
Server 4	i-04daa510e1b4fc4a3	Running	t2.micro	2/2 checks passed	No alarms	us-east-1c
Server 2	i-011e9e2a704360870	Running	t2.micro	2/2 checks passed	No alarms	us-east-1c

Instance: i-0232bcc0cde5c2e4f (Load Balancer)

Select an instance above

Details Security Networking Storage Status checks Monitoring Tags

Instance summary Info

Instance ID i-0232bcc0cde5c2e4f (Load Balancer)	Public IPv4 address 54.205.11.237   <a href="#">open address</a>	Private IPv4 addresses 172.31.91.115
IPv6 address	Instance state	Public IPv4 DNS

Successfully stopped i-0232bcc0cde5c2e4f

© 2022, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

Instance was stopped successfully.

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

Snapshots (1/1)

Name	Snapshot ID	Size	Description	Storage
-	snap-0f4ed45d2321d7451	8 GiB	Snapshot for Load Balancer	Standard

Actions ▾ Create snapshot

Create volume from snapshot

Create image from snapshot Started

Copy snapshot

Modify permissions

2022/03

Manage fast snapshot restore

Archive snapshot

Restore snapshot from archive

Change restore period

Delete snapshot

Manage tags

Owned by me Filter snapshots by attributes and tags

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

Create the copy of the load balancer so no we will not change any fields

Image settings

Snapshot ID: snap-0f4ed45d2321d7451

Image name: AMI-from-LoadBalancer

Description: My image description

Architecture: x86\_64

Root device name: /dev/sda1

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

New EC2 Experience [Learn more](#)

EC2 Dashboard  
EC2 Global View  
Events  
Tags  
Limits

Instances

- Instances [New](#)
- Instance Types
- Launch Templates
- Spot Requests
- Savings Plans
- Reserved Instances [New](#)
- Dedicated Hosts
- Scheduled Instances
- Capacity Reservations

Feedback English (US) ▾

Successfully requested new image ami-0ch55a56464041517.  
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	Starte...
-	snap-0f4ed45d2321d7451	8 GiB	Snapshot for Load Balancer	Standard	Completed	2022/

Select a snapshot above.

© 2022, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

Instance Types  
Launch Templates  
Spot Requests  
Savings Plans  
Reserved Instances [New](#)  
Dedicated Hosts  
Scheduled Instances  
Capacity Reservations

Images

- AMIs [New](#)
- AMI Catalog

Elastic Block Store

- Volumes [New](#)
- Snapshots [New](#)
- Lifecycle Manager

Network & Security

- Security Groups

Feedback English (US) ▾

Amazon Machine Images (AMIs) (1) [Info](#)

AMI name	Source	Owner	Visibility	Status	
1517	AMI-from-LoadBalancer	465434389122/AMI-from-LoadBalancer	465434389122	Private	Available <a href="#">Details</a>

Select an AMI

© 2022, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

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

The screenshot shows the AWS EC2 console with the 'Amazon Machine Images (AMIs)' section selected. A single AMI entry is listed:

Name	AMI ID	AMI name	Source	Owner
-	ami-0cb55a56464041517	AMI-from-LoadBalancer	465434389122/AMI-from-LoadBalancer	465434389122

Below the table, a modal window titled 'AMI ID: ami-0cb55a56464041517' displays detailed information about the AMI:

Details	Permissions	Storage	Tags
AMI ID ami-0cb55a56464041517	Image type machine	Platform details Linux/UNIX	Root device type EBS
AMI name AMI-from-LoadBalancer	Owner account ID 465434389122	Architecture x86_64	Usage operation RunInstances
Root device name	Status	Source	Virtualization type

Select free tier instance and click review and launch

The screenshot shows the 'Step 2: Choose an Instance Type' screen. The user has selected the 't2.micro' instance type, which is highlighted with a green background and labeled 'Free tier eligible'.

Below the table, there are navigation buttons: 'Cancel', 'Previous', 'Review and Launch' (which is highlighted in blue), and 'Next: Configure Instance Details'.

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
<input type="checkbox"/>	t2	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	t2	t2.micro <span style="background-color: green; color: white;">Free tier eligible</span>	1	1	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.small	1	2	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.large	2	8	EBS only	-	Low to Moderate	Yes

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

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

### Step 6: Configure Security Group

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

Assign a security group:  Create a new security group  Select an existing security group

Security Group ID	Name	Description	Actions
<input type="checkbox"/> sg-072e9c479cd3f2057	default	default VPC security group	Copy to new
<input checked="" type="checkbox"/> sg-0c79efc95e2a5ceb8	Load Balancer Security Group	launch-wizard-1 created 2022-03-26T10:55:56.413-04:00	Copy to new

Inbound rules for sg-0c79efc95e2a5ceb8 (Selected security groups: sg-0c79efc95e2a5ceb8)

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

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

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

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

### Step 7: Review Instance Launch

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

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

**AMI Details** [Edit AMI](#)

AMI-from-LoadBalancer - ami-0cb55a56464041517  
 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

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

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

**AMI Details** [Edit AMI](#)

AMI-from-LoadBalancer - ami-0cb55a56464041517  
 Root Device Type: ebs Virtualization type: hvm

**Instance Type** [Edit instance type](#)

Instance Type	ECUs	vCPUs
---------------	------	-------

**Key Pair**

Create a new key pair  
 RSA  ED25519  
 Key pair name: Load Balancer Security Group [Download Key Pair](#)

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

[Cancel](#) [Launch Instances](#)

Feedback English (US) © 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences [Show all](#) [X](#)

The screenshot shows the AWS Launch Status page. At the top, there's a green box with a checkmark stating "Your instances are now launching". Below it, a message says "The following instance launches have been initiated: i-0fe63bd0f41f5965d" with a link to "View launch log". There's also a blue box with an info icon and the text "Get notified of estimated charges" and "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)".

## Launch Status

The screenshot shows the AWS Instances page. It displays a table of instances with the following columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, and Availability Zone. The table contains four rows: "Server 2" (Running, t2.micro, 2/2 checks passed, No alarms, us-east-1c), "Server 1" (Running, t2.micro, 2/2 checks passed, No alarms, us-east-1d), "Server 3" (Running, t2.micro, 2/2 checks passed, No alarms, us-east-1d), and "Load Balancer ..." (Running, t2.micro, Initializing, No alarms, us-east-1d). A sidebar on the left shows navigation links for EC2 Dashboard, EC2 Global View, Events, Tags, Limits, and Instances (with sub-links for Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Scheduled Instances, and Capacity Reservations).

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

The screenshot shows the AWS Instances page with a modal window open for the instance "i-0fe63bd0f41f5965d". The modal has tabs for Details, Security, Networking, Storage, Status checks, Monitoring, and Tags. Under the Details tab, the "Instance summary" section shows the Instance ID (i-0fe63bd0f41f5965d), Public IPv4 address (54.227.163.213), Private IPv4 addresses (172.31.21.177), and Instance state (Running). Other sections like Security, Networking, Storage, Status checks, Monitoring, and Tags are also visible.

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

EC2 instance is in running status.

Connect to the instance

The screenshot shows the 'Connect to instance' page for an EC2 instance. The instance ID is i-0fe63bd0f41f5965d (Load Balancer New). The public IP address is 54.227.163.213. The user name is set to 'Load Balancer New'. A note indicates that the guessed user name is correct but to check if the AMI owner has changed it. Navigation tabs include EC2 Instance Connect, Session Manager, SSH client, and EC2 Serial Console.

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

```
Last login: Mon Mar 28 17:38:51 2022 from ec2-18-206-107-25.compute-1.amazonaws.com
[ec2-user@ip-172-31-21-177 ~]$ cd /etc/nginx/[REDACTED]
```

i-0fe63bd0f41f5965d (Load Balancer New)

Public IPs: 54.227.163.213 Private IPs: 172.31.21.177

```
Last login: Mon Mar 28 17:38:51 2022 from ec2-18-206-107-25.compute-1.amazonaws.com
[ec2-user@ip-172-31-21-177 ~]$ cd /etc/nginx/
[ec2-user@ip-172-31-21-177 nginx]$ sudo vim nginx.conf
```

i-0fe63bd0f41f5965d (Load Balancer New)

Public IPs: 54.227.163.213 Private IPs: 172.31.21.177

```
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 ec2-3-84-126-109.compute-1.amazonaws.com weight=1;
        server ec2-3-93-81-244.compute-1.amazonaws.com weight=2;
        server ec2-54-226-84-198.compute-1.amazonaws.com weight=1;
        server ec2-54-145-113-195.compute-1.amazonaws.com weight=2;
    }

    log_format main '$remote_addr - $remote_user [$time_local] "$request" '
"nginx.conf" 100L, 2681B
```

25,61 7%

i-0fe63bd0f41f5965d (Load Balancer New)

Public IPs: 54.227.163.213 Private IPs: 172.31.21.177

```
[ec2-user@ip-172-31-21-177 ~]$ ruby visit_server -d ec2-54-205-11-237.compute-1.amazonaws.com
Starting to visit load balancing server
^Cvisit_server:49:in `': Interrupt
    from visit_server:49:in `block in <main>'
    from visit_server:46:in `times'
    from visit_server:46:in `<main>'

[ec2-user@ip-172-31-21-177 ~]$ ruby visit_server -d ec2-54-227-163-213.compute-1.amazonaws.com
Starting to visit load balancing server
```

i-0fe63bd0f41f5965d (Load Balancer New)

Public IPs: 54.227.163.213 Private IPs: 172.31.21.177

Terminated all the instances

The screenshot shows the AWS EC2 Instances page with the following details:

Name	Instance ID	Instance state	Instance type	Health	Alarm status	Availability Zone
Load Balancer	i-0232bcc0cde5c2e4f	Stopped	t2.n	<span style="color: red;">XX</span>	No alarms	us-east-1c
Server 4	i-04daa510e1b4fc4a3	Running	t2.n	<span style="color: green;">GG</span>	No alarms	us-east-1c
Server 2	i-011e9e2a704360870	Running	t2.micro	<span style="color: green;">GG</span>	No alarms	us-east-1c
Server 1	i-0df38a0237449f23d	Running	t2.micro	<span style="color: green;">GG</span>	2/2 checks passed	us-east-1d
Server 3	i-0297318cc08cedf21	Running	t2.micro	<span style="color: green;">GG</span>	2/2 checks passed	us-east-1d

Instances: i-0232bcc0cde5c2e4f (Load Balancer), i-04daa510e1b4fc4a3 (Server 4), i-011e9e2a704360870 (Server 2), i-0df38a0237449f23d (Server 1), i-0297318cc08cedf21 (Server 3), i-0fe63bd0f41f5965d (Load Balancer New)

Select an instance above

Monitoring

1h 3h 12h 1d 3d 1w Custom Add to dashboard

## All instances are terminated

The screenshot shows the AWS EC2 Instances page. The left sidebar has sections for EC2 Dashboard, EC2 Global View, Events, Tags, Limits, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances (New), Dedicated Hosts, Scheduled Instances, and Capacity Reservations. The main content area shows a table titled 'Instances (6)'. The columns are Name, Instance ID, Instance state, Instance type, Status check, Alarm status, and Availability Zone. All instances listed are in the 'terminated' state. The table includes rows for 'Load Balancer' (i-0232bcc0cde5c2e4f), 'Server 4' (i-04daa510e1b4fc4a3), 'Server 2' (i-011e9e2a704360870), 'Server 1' (i-0df3ba0257449f23d), and 'Server 3' (i-0297318cc08ccff21). The last column shows 'us-east-1c' or 'us-east-1d' for each instance.

To delete the snapshot first we need to deregister the ami

The screenshot shows the AWS AMI Management page. The left sidebar has sections for Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances (New), Dedicated Hosts, Scheduled Instances, and Capacity Reservations. The main content area shows a table titled 'Amazon Machine Images (AMIs) (1/1)'. The table has columns for Actions, Name, AMI ID, Source, and Owner. One row is selected with the AMI ID 'ami-0cb55a56464041517'. A context menu is open over this row, with 'Deregister AMI' highlighted. The table also lists other actions like Copy AMI, Edit AMI permissions, Request Spot Instances, Manage tags, Change description, Manage image optimization, Manage AMI Deprecation, and Register instance store-backed AMI. Below the table, there is a detailed view of the AMI with tabs for Details, Permissions, Storage, and Tags. The Details tab shows fields like AMI ID (ami-0cb55a56464041517), Image type (machine), Platform details (Linux/UNIX), Root device type (EBS), AMI name (AMI-from-LoadBalancer), Owner account ID (465434389122), Architecture (x86\_64), Usage operation (RunInstances), Root device name, Status, Source, and Virtualization type.

The screenshot shows the AWS Management Console interface for managing Amazon Machine Images (AMIs). On the left, a navigation sidebar lists various services under 'Instances', 'Images', and 'Elastic Block Store'. The main area displays a table of AMIs, with one row selected. A modal dialog titled 'Deregister AMI' is open over the table. The dialog contains a confirmation message: 'After you deregister an AMI, you can't use it to launch new instances.' Below this, it asks 'Are you sure that you want to deregister these AMIs?' with a list item 'ami-0cb55a56464041517'. At the bottom of the dialog are 'Cancel' and 'Deregister AMI' buttons. The table in the background shows the following data:

Name	AMI ID	AMI name	Source	Owner
ami-0cb55a56464041517	ami-0cb55a56464041517	AMI-from-LoadBalancer	Amazon Machine Image	465434389122

Successfully deregistered the ami

The screenshot shows the same AWS AMI management interface after the deregistration process. A green success notification bar at the top states 'Successfully deregistered ami-0cb55a56464041517.' The main table now shows the message 'You do not have any images in this Region.' A modal dialog titled 'Select an AMI' is open, indicating that no AMIs are available to select. The navigation sidebar and other UI elements remain consistent with the first screenshot.

## Snapshot can be deleted.

The screenshot shows the AWS Management Console interface for the Elastic Block Store (EBS) service. On the left, there is a navigation sidebar with sections for Instances, Images, and Elastic Block Store. Under 'Elastic Block Store', 'Volumes' and 'Snapshots' are listed. The 'Snapshots' section is currently selected, showing a table with one item: 'snap-0f4ed45d2321d7451'. This snapshot is 8 GiB in size, has a description 'Snapshot for Load Balancer', and is in the 'Standard' storage tier. It was created on '2022/03'. In the top right corner of the main content area, there is a context menu with various actions. The 'Delete snapshot' option is highlighted with a red box. Below the main table, there is a detailed view for the selected snapshot, showing its ID, name, size, description, storage tier, and creation date. At the bottom of this view, there is a confirmation dialog box asking 'Delete snap-0f4ed45d2321d7451?'. The 'Delete' button in this dialog is also highlighted with a red box.

**Snapshots (1/1)**

Name	Snapshot ID	Size	Description	Storage tier
-	snap-0f4ed45d2321d7451	8 GiB	Snapshot for Load Balancer	Standard

**Actions**

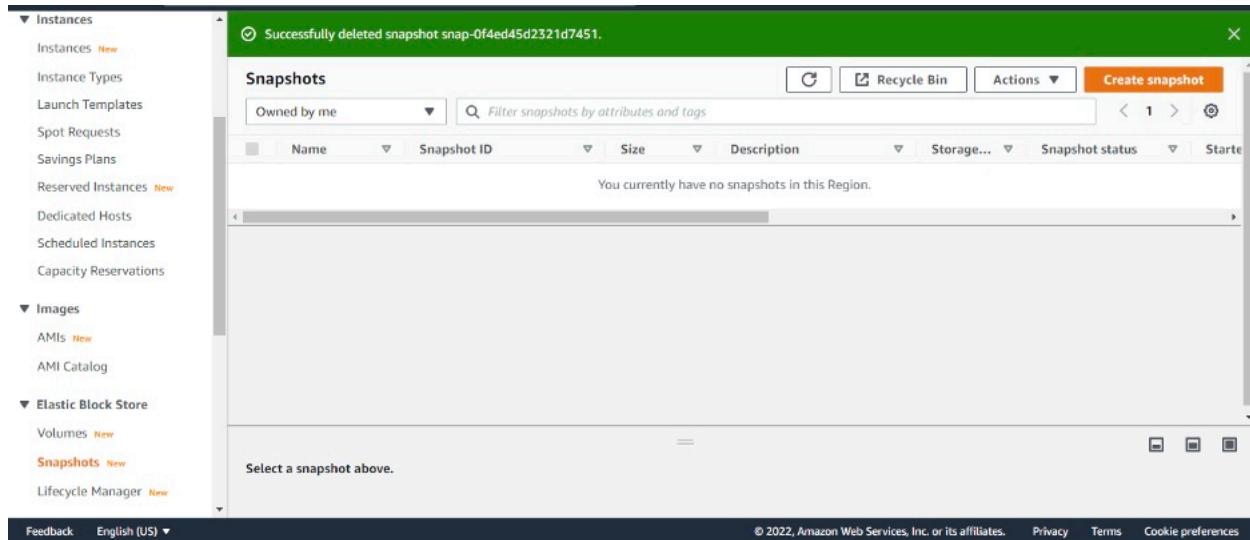
- Create volume from snapshot
- Create image from snapshot
- Copy snapshot
- Modify permissions
- Manage fast snapshot restore
- Archive snapshot
- Restore snapshot from archive
- Change restore period
- Delete snapshot**
- Manage tags

**Snapshot ID: snap-0f4ed45d2321d7451**

**Details** | Permissions | Storage tier | Tags

Are you sure that you want to delete snap-0f4ed45d2321d7451?

Cancel **Delete**



References:

- <https://www.linuxtechi.com/capture-analyze-packets-tcpdump-command-linux/>
- <https://opensource.com/article/18/10/introduction-tcpdump>
- [https://docs.amazonaws.cn/en\\_us/cli/latest/userguide/cli-services-ec2-instances.html](https://docs.amazonaws.cn/en_us/cli/latest/userguide/cli-services-ec2-instances.html)
- <https://stackoverflow.com/questions/49227466/aws-instance-only-allow-traffic-from-load-balancer>
- <https://docs.aws.amazon.com/elasticloadbalancing/latest/application/load-balancer-update-securitygroups.html>
- <https://rvm.io/>
- <https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/ruby-development-environment.html#rubydevelopment-environment-ruby> <https://www.cyberciti.biz/faq/nginx-linux-restart/>
- <https://docs.aws.amazon.com/prescriptive-guidance/latest/backup-recovery/ec2-backup.html>
- <https://docs.aws.amazon.com/prescriptive-guidance/latest/backup-recovery/new-ebs-volume-backups.html>
- <https://docs.aws.amazon.com/cli/latest/userguide/cli-services-ec2-instances.html>