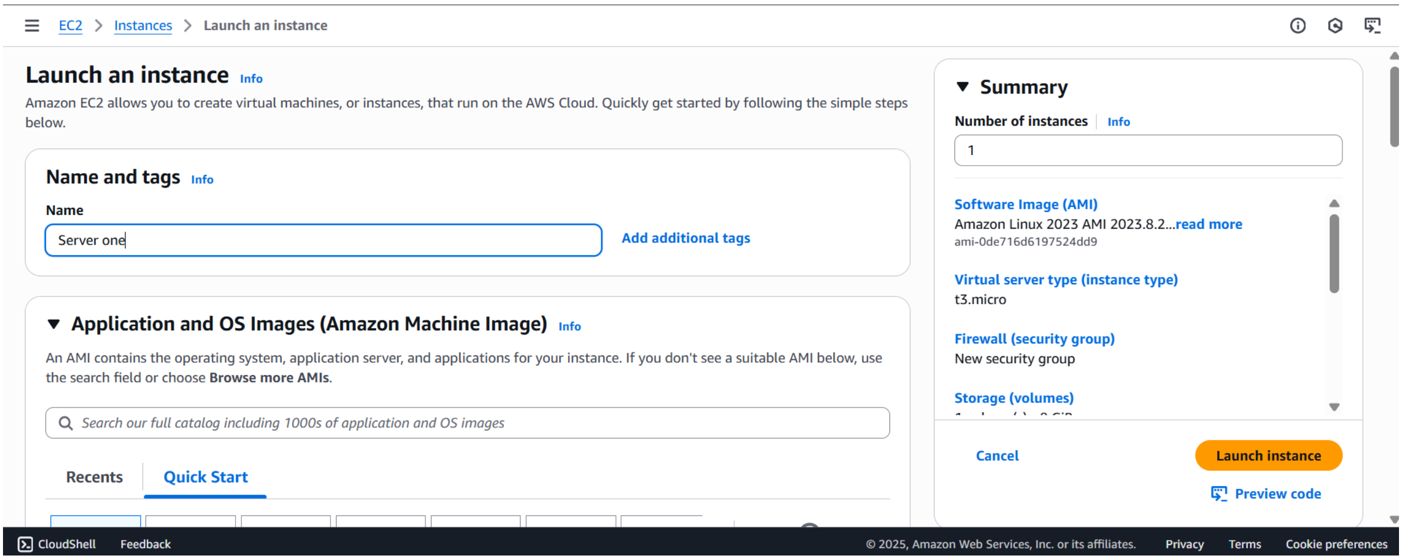
Configure an AWS cross zone loadbalancer. And where my web-app is going to expose on port number 80.

Created two ec2 instances in different zones and allowed port 80 in both.



EC2 > Instances > Launch an instance

Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name: Server one

Add additional tags

Application and OS Images (Amazon Machine Image) Info

An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose **Browse more AMIs**.

Search our full catalog including 1000s of application and OS images

Recents | Quick Start

Summary

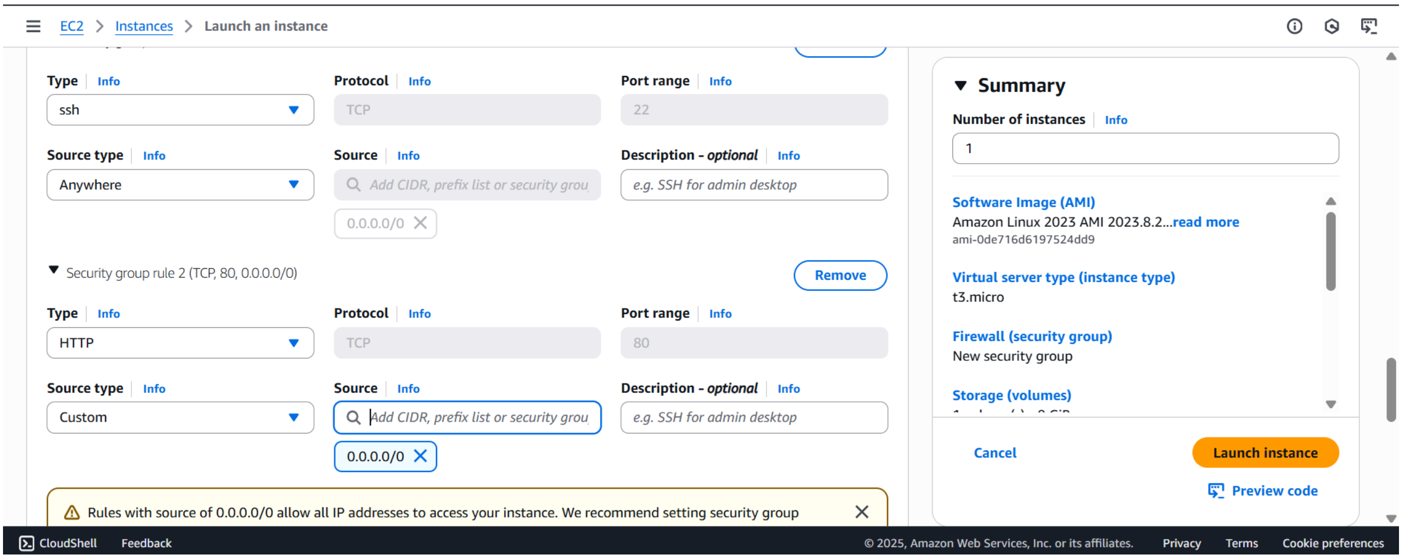
Number of instances: 1

Software Image (AMI): Amazon Linux 2023 AMI 2023.8.2...read more
ami-0de716d6197524dd9

Virtual server type (instance type): t3.micro

Firewall (security group): New security group

Storage (volumes)



EC2 > Instances > Launch an instance

Type: ssh | Protocol: TCP | Port range: 22

Source type: Anywhere | Source: 0.0.0.0/0 | Description - optional: e.g. SSH for admin desktop

Security group rule 2 (TCP, 80, 0.0.0.0/0)

Type: HTTP | Protocol: TCP | Port range: 80

Source type: Custom | Source: 0.0.0.0/0 | Description - optional: e.g. SSH for admin desktop

⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group

Summary

Number of instances: 1

Software Image (AMI): Amazon Linux 2023 AMI 2023.8.2...read more
ami-0de716d6197524dd9

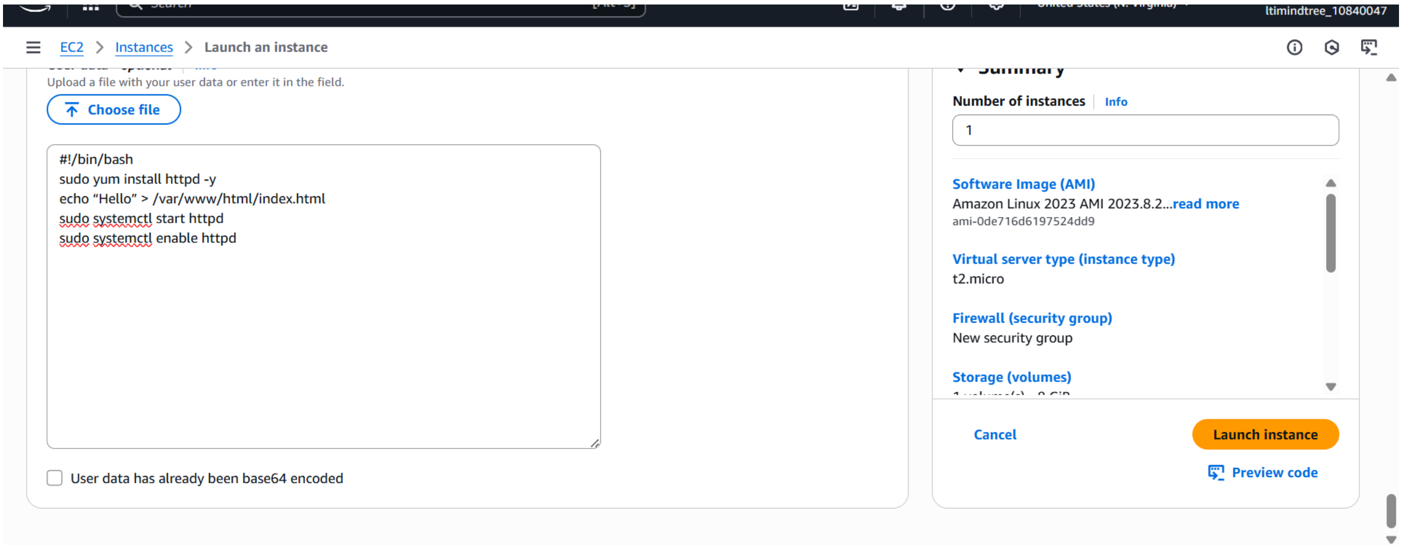
Virtual server type (instance type): t3.micro

Firewall (security group): New security group

Storage (volumes)

Remove
Add CIDR, prefix list or security grou
Cancel
Launch instance
Preview code
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Wrote script in advanced settings.



Upload a file with your user data or enter it in the field.

```
#!/bin/bash
sudo yum install httpd -y
echo “Hello” > /var/www/html/index.html
sudo systemctl start httpd
sudo systemctl enable httpd
```

□ User data has already been base64 encoded

Summary

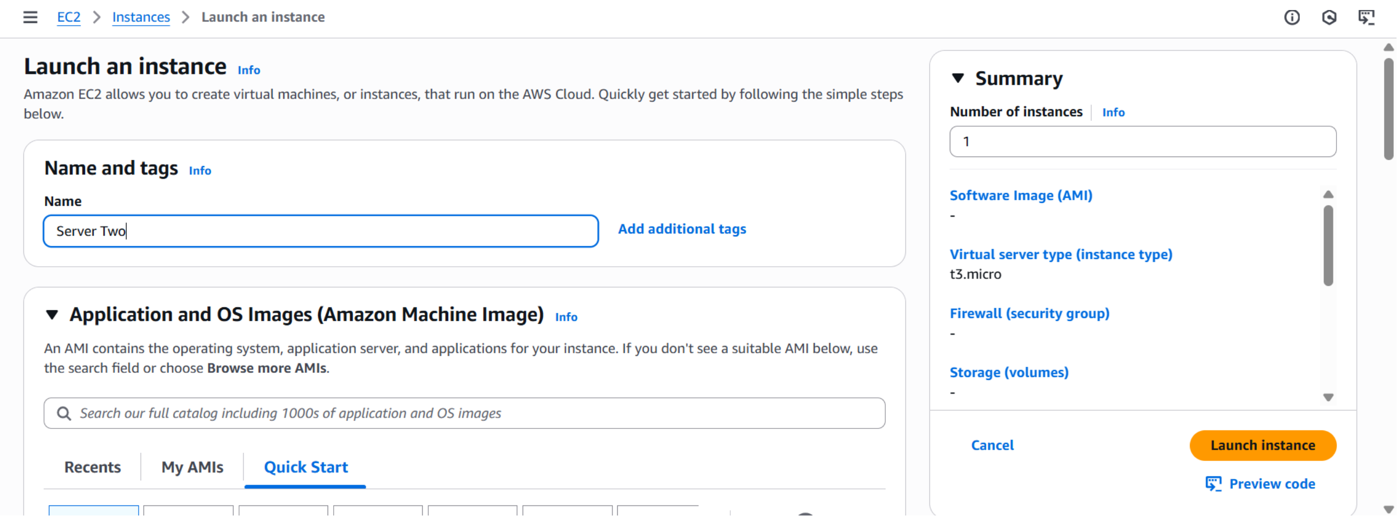
Number of instances: 1

Software Image (AMI): Amazon Linux 2023 AMI 2023.8.2...read more ami-0de716d6197524dd9

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes)



Launch an instance

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags

Name: Server Two

Application and OS Images (Amazon Machine Image)

An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose **Browse more AMIs**.

Search our full catalog including 1000s of application and OS images
Recents
My AMIs
Quick Start

Summary

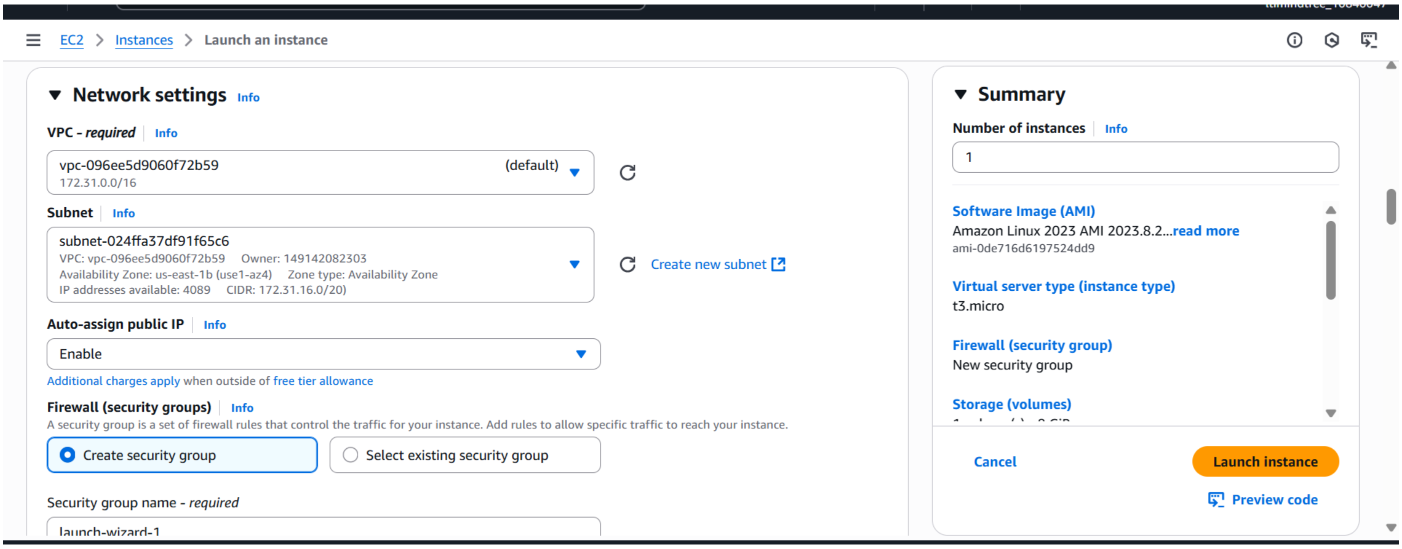
Number of instances: 1

Software Image (AMI): -

Virtual server type (instance type): t3.micro

Firewall (security group): -

Storage (volumes): -



Network settings

VPC - *required*: vpc-096ee5d9060f72b59 (default) 172.31.0.0/16

Subnet: subnet-024ffa37df91f65c6
VPC: vpc-096ee5d9060f72b59 Owner: 149142082303
Availability Zone: us-east-1b (use1-az4) Zone type: Availability Zone
IP addresses available: 4089 CIDR: 172.31.16.0/20)

Auto-assign public IP: Enable

Additional charges apply when outside of free tier allowance

Firewall (security groups)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

● Create security group ○ Select existing security group

Security group name - *required*: launch-wizard-1

Summary

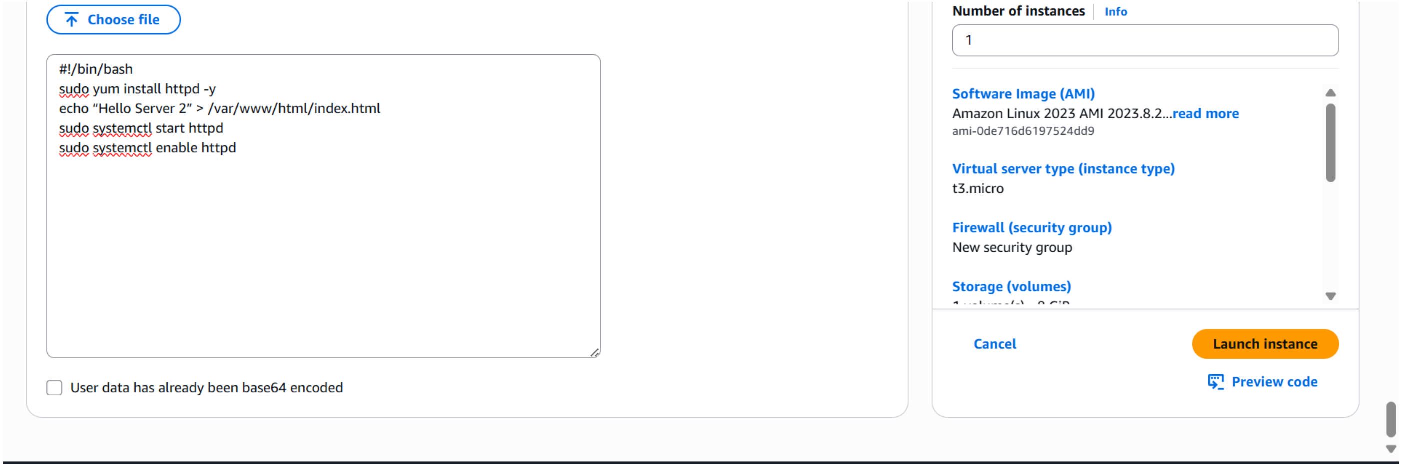
Number of instances: 1

Software Image (AMI): Amazon Linux 2023 AMI 2023.8.2...read more ami-0de716d6197524dd9

Virtual server type (instance type): t3.micro

Firewall (security group): New security group

Storage (volumes)



```
#!/bin/bash
sudo yum install httpd -y
echo "Hello Server 2" > /var/www/html/index.html
sudo systemctl start httpd
sudo systemctl enable httpd
```

□ User data has already been base64 encoded

Number of instances: 1

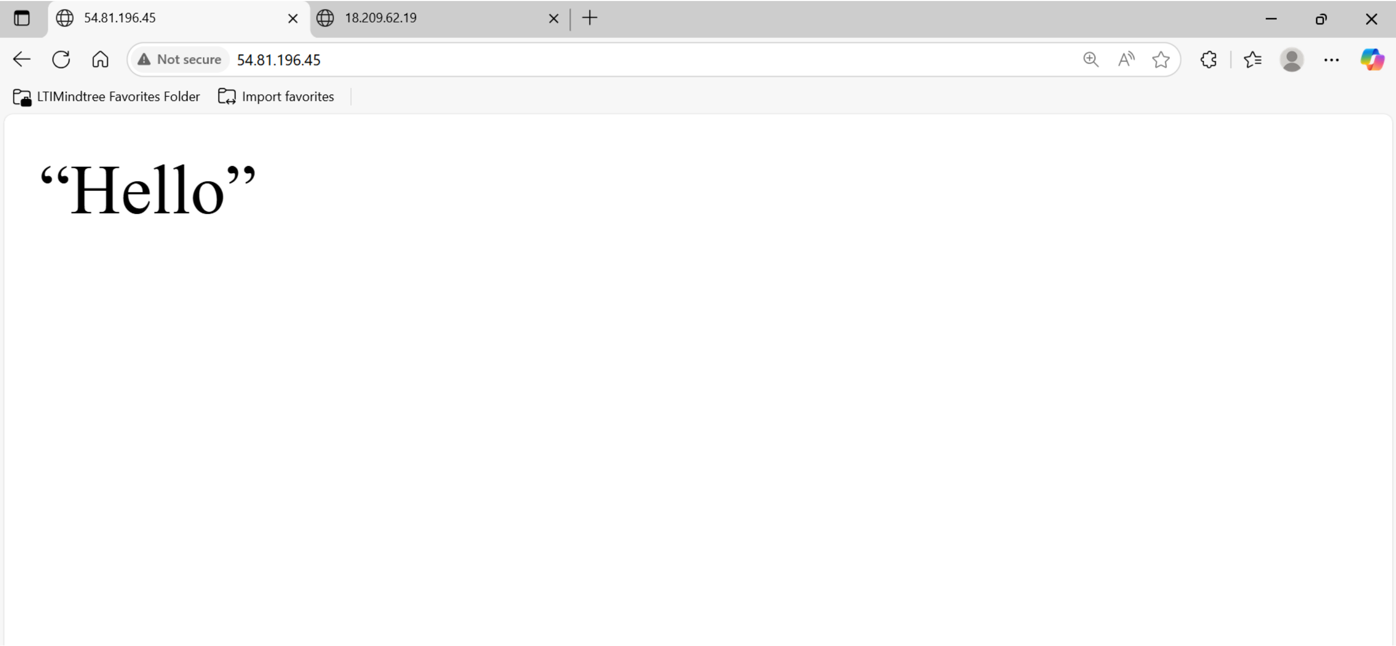
Software Image (AMI)
Amazon Linux 2023 AMI 2023.8.2...read more
ami-0de716d6197524dd9

Virtual server type (instance type)
t3.micro

Firewall (security group)
New security group

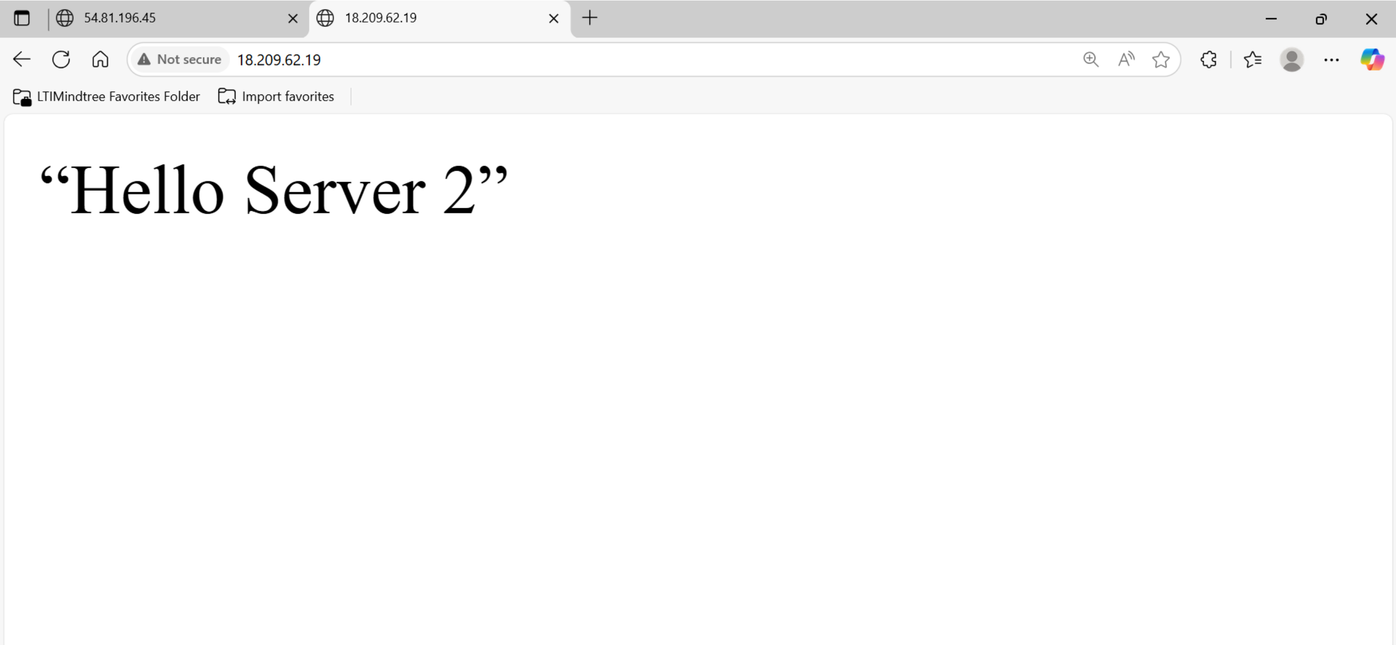
Storage (volumes)

Both working fine in port 80.



54.81.196.45

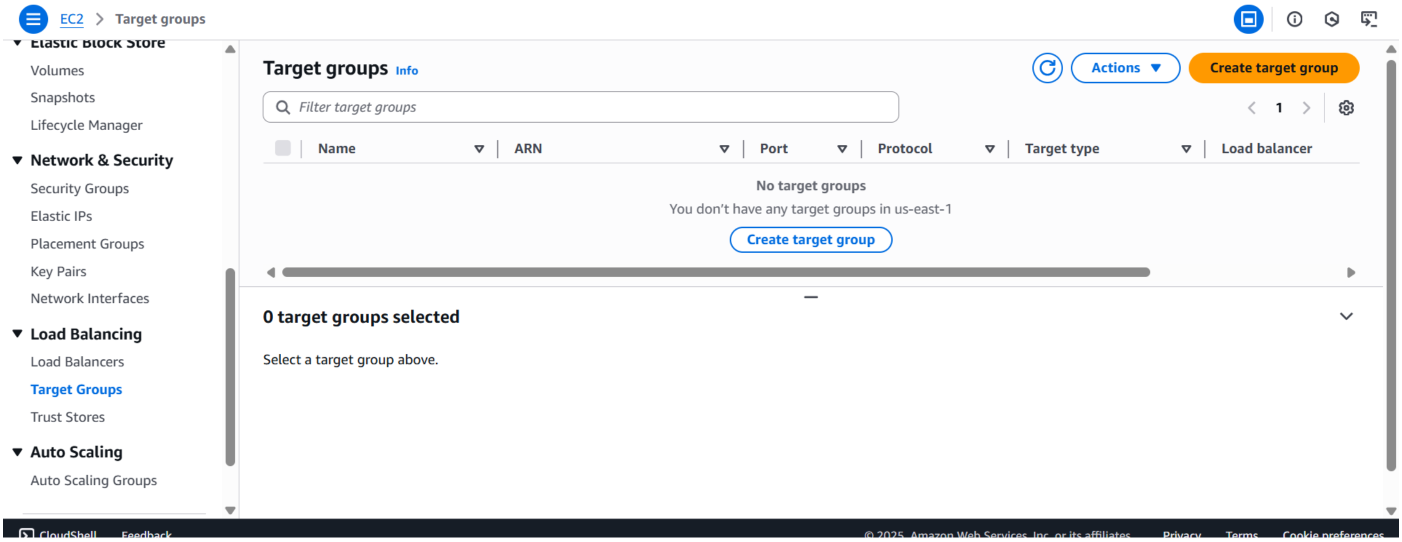
"Hello"



18.209.62.19

"Hello Server 2"

Created a target group.



Target groups

Filter target groups

Name	ARN	Port	Protocol	Target type	Load balancer

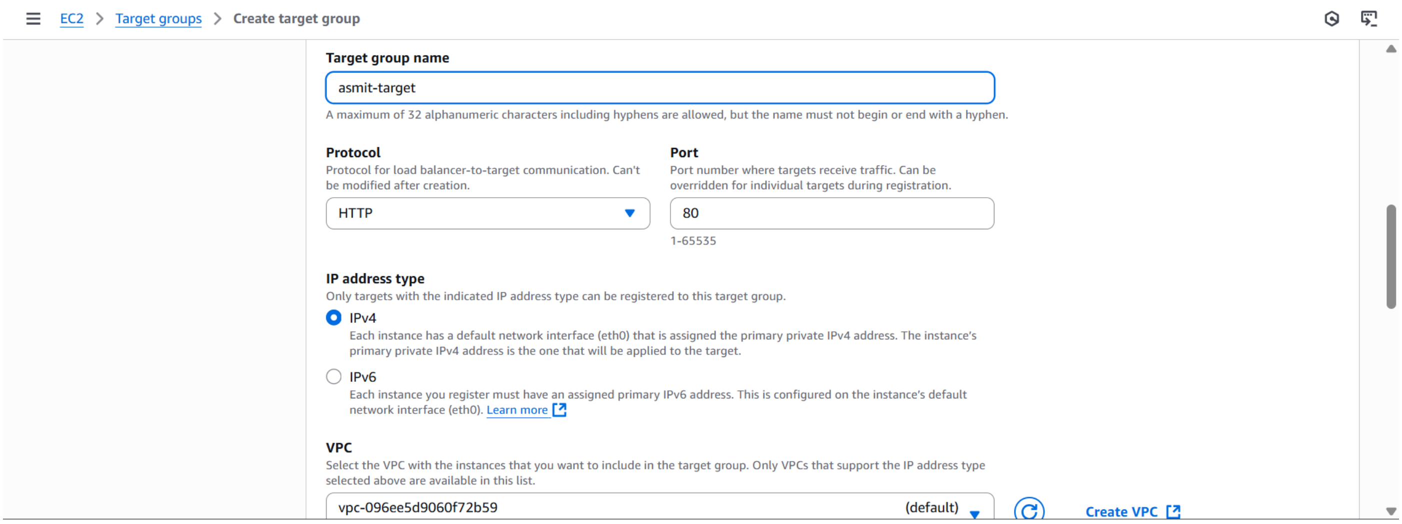
No target groups

You don't have any target groups in us-east-1

Create target group

0 target groups selected

Select a target group above.



Target group name

asmit-target

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Protocol

Protocol for load balancer-to-target communication. Can't be modified after creation.

HTTP

Port

Port number where targets receive traffic. Can be overridden for individual targets during registration.

80

1-65535

IP address type

Only targets with the indicated IP address type can be registered to this target group.

● IPv4

Each instance has a default network interface (eth0) that is assigned the primary private IPv4 address. The instance's primary private IPv4 address is the one that will be applied to the target.

○ IPv6

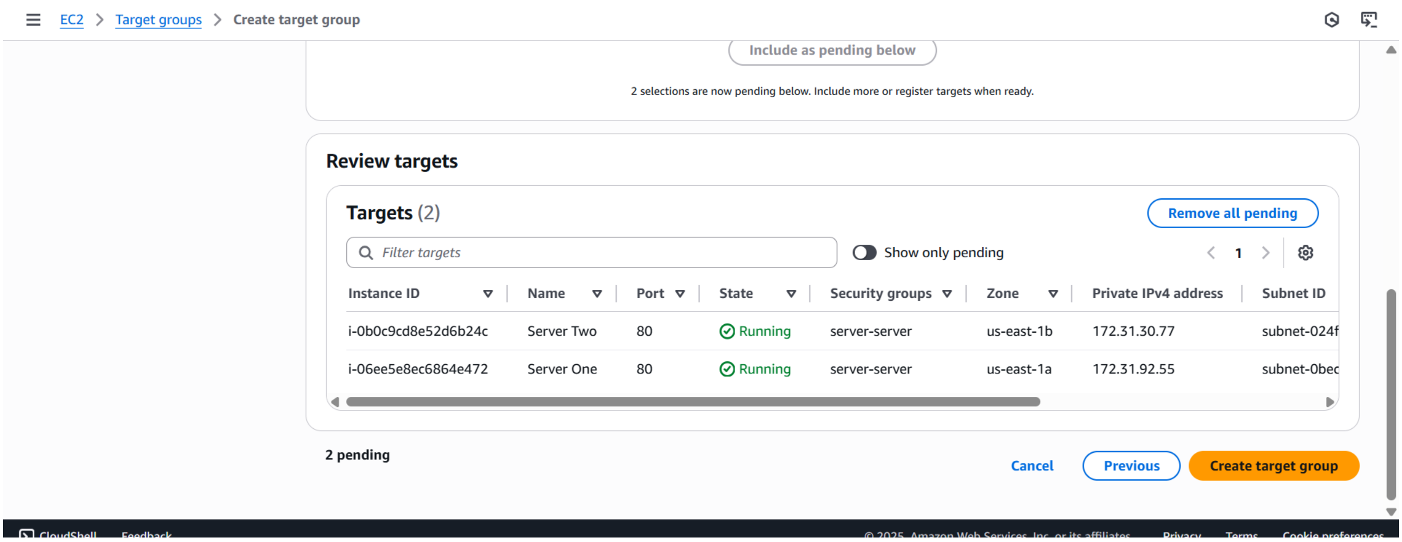
Each instance you register must have an assigned primary IPv6 address. This is configured on the instance's default network interface (eth0). Learn more

VPC

Select the VPC with the instances that you want to include in the target group. Only VPCs that support the IP address type selected above are available in this list.

vpc-096ee5d9060f72b59 (default)

Included both the instances.



2 selections are now pending below. Include more or register targets when ready.

Review targets

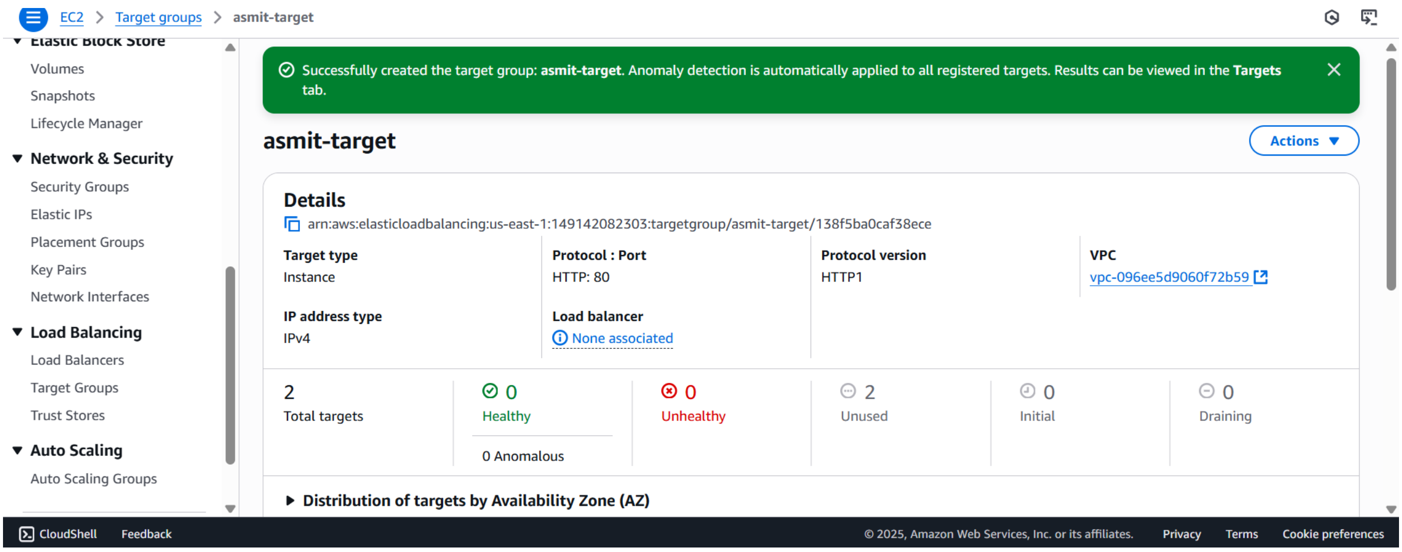
Targets (2)

Filter targets

Show only pending

Instance ID	Name	Port	State	Security groups	Zone	Private IPv4 address	Subnet ID
i-0b0c9cd8e52d6b24c	Server Two	80	✓ Running	server-server	us-east-1b	172.31.30.77	subnet-024f
i-06ee5e8ec6864e472	Server One	80	✓ Running	server-server	us-east-1a	172.31.92.55	subnet-0bec

2 pending



Successfully created the target group: **asmit-target**. Anomaly detection is automatically applied to all registered targets. Results can be viewed in the **Targets** tab.

asmit-target

Details

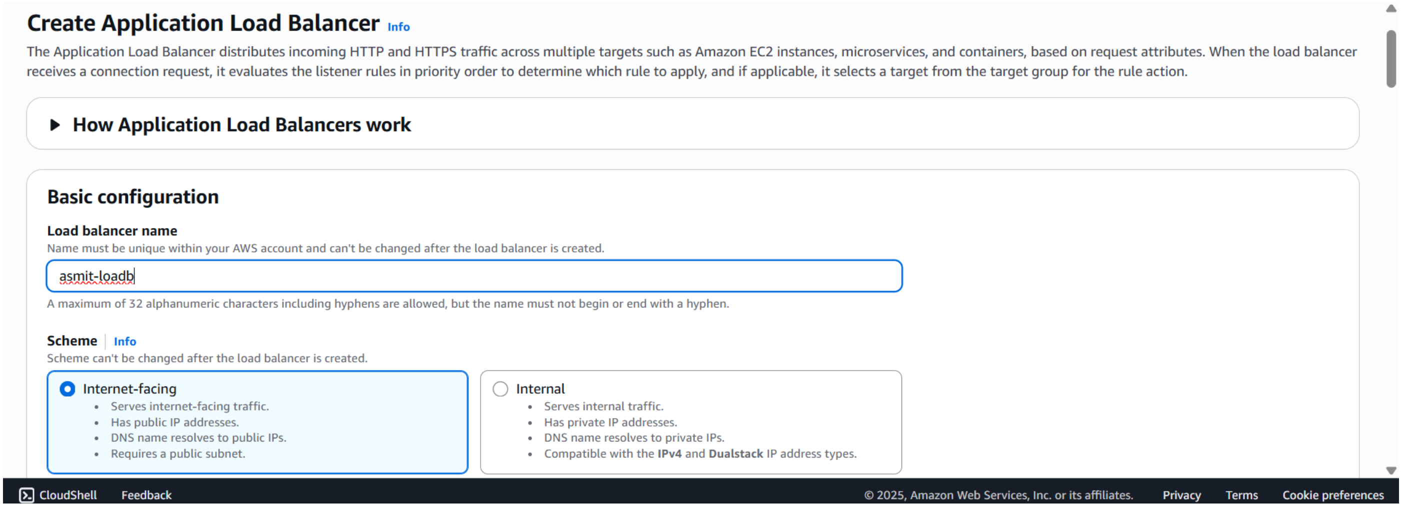
arn:aws:elasticloadbalancing:us-east-1:149142082303:targetgroup/asmit-target/138f5ba0caf38ece

Target type	Protocol : Port	Protocol version	VPC
Instance	HTTP: 80	HTTP1	vpc-096ee5d9060f72b59
IP address type	**Load balancer**		
IPv4	None associated		

2	✓ 0	⊗ 0	2	0	0
Total targets	Healthy 0 Anomalous	Unhealthy	Unused	Initial	Draining

▶ Distribution of targets by Availability Zone (AZ)

Created a load balancer with above target group.



Create Application Load Balancer

The Application Load Balancer distributes incoming HTTP and HTTPS traffic across multiple targets such as Amazon EC2 instances, microservices, and containers, based on request attributes. When the load balancer receives a connection request, it evaluates the listener rules in priority order to determine which rule to apply, and if applicable, it selects a target from the target group for the rule action.

▶ How Application Load Balancers work

Basic configuration

Load balancer name

Name must be unique within your AWS account and can't be changed after the load balancer is created.

asmit-loadb

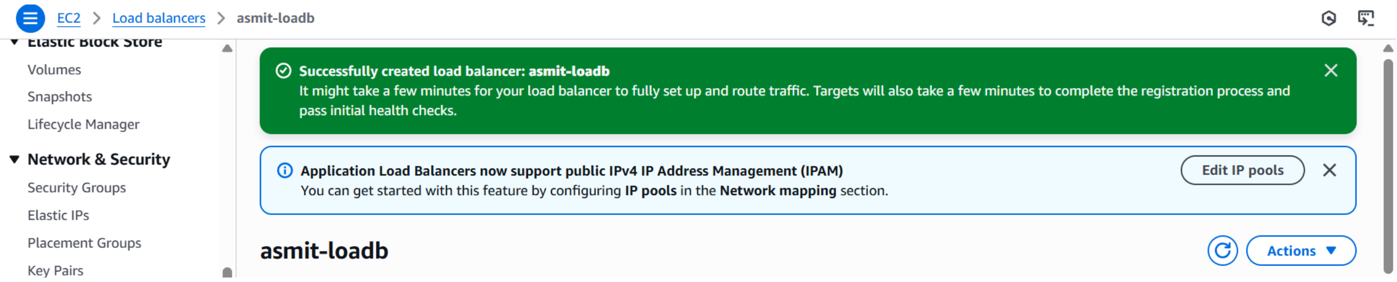
A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Scheme Info

Scheme can't be changed after the load balancer is created.

● Internet-facing
- Serves internet-facing traffic.
- Has public IP addresses.
- DNS name resolves to public IPs.
- Requires a public subnet.

○ Internal
- Serves internal traffic.
- Has private IP addresses.
- DNS name resolves to private IPs.
- Compatible with the **IPv4** and **Dualstack** IP address types.

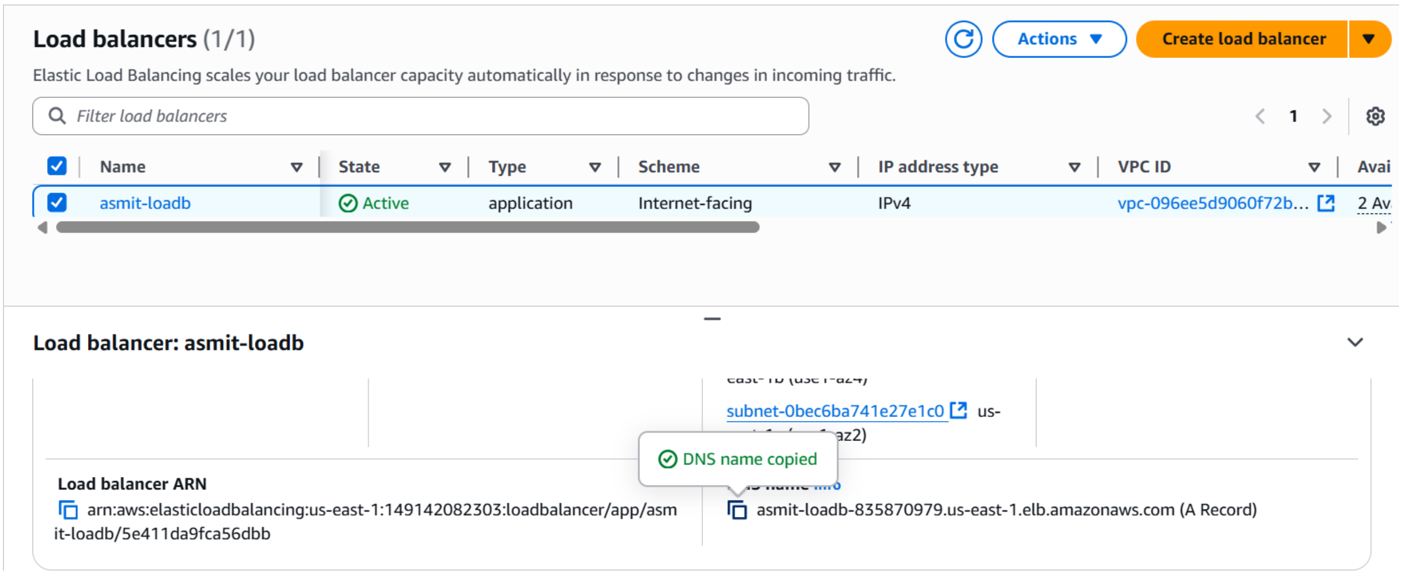


Successfully created load balancer: asmit-loadb
It might take a few minutes for your load balancer to fully set up and route traffic. Targets will also take a few minutes to complete the registration process and pass initial health checks.

Application Load Balancers now support public IPv4 IP Address Management (IPAM)
You can get started with this feature by configuring **IP pools** in the **Network mapping** section.

asmit-loadb

Copied DNS name and pasted on the browser.



Load balancers (1/1)

Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.

☑	Name	State	Type	Scheme	IP address type	VPC ID	Avai
☑	asmit-loadb	✓ Active	application	Internet-facing	IPv4	vpc-096ee5d9060f72b...	2 Av

Load balancer: asmit-loadb

subnet-0bec6ba741e27e1c0 us-

az2)

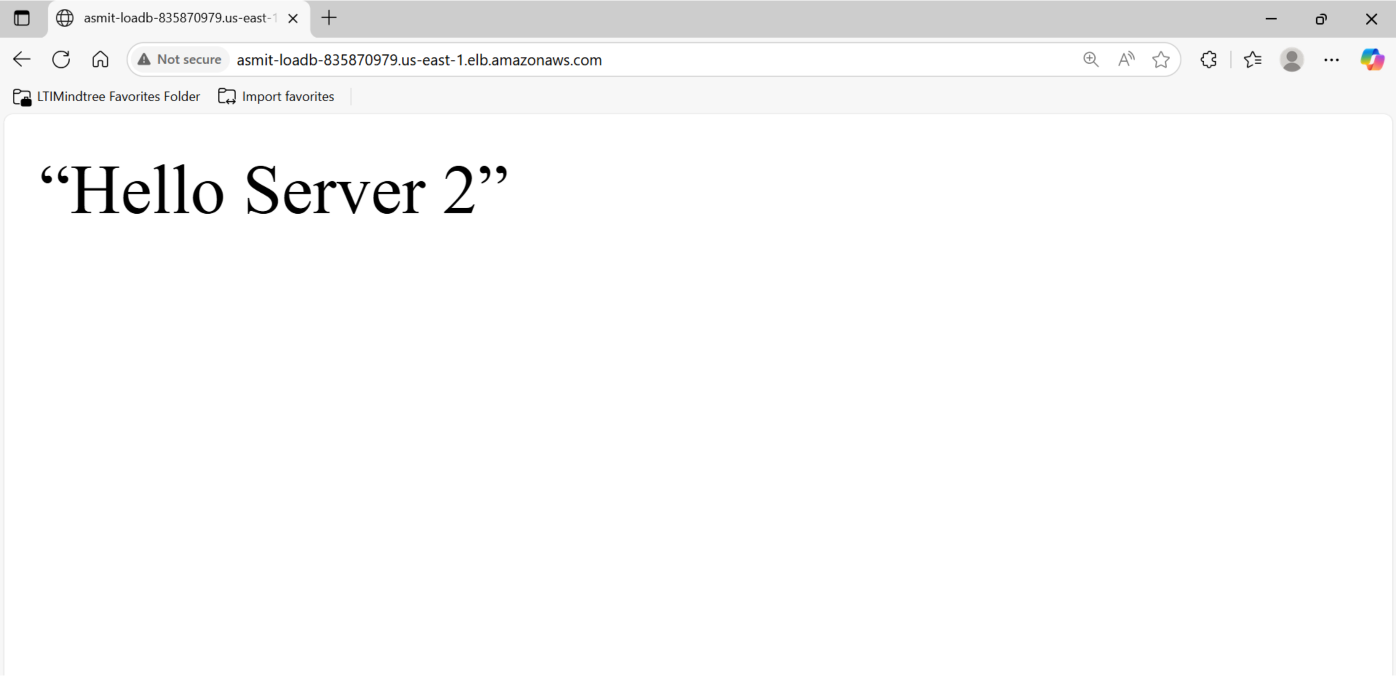
✓ DNS name copied

Load balancer ARN

arn:aws:elasticloadbalancing:us-east-1:149142082303:loadbalancer/app/asmit-loadb/5e411da9fca56dbb

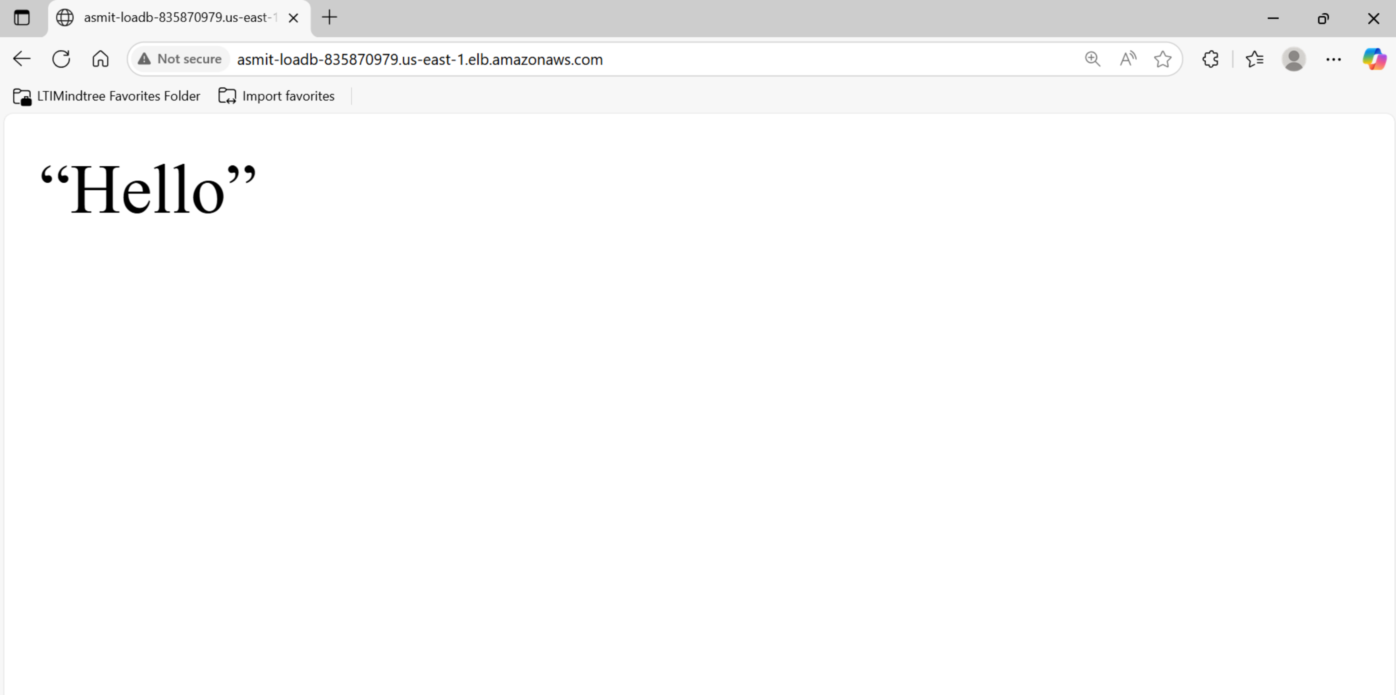
asmit-loadb-835870979.us-east-1.elb.amazonaws.com (A Record)

By refreshing the page the content got changed.



asmit-loadb-835870979.us-east-1.elb.amazonaws.com

"Hello Server 2"



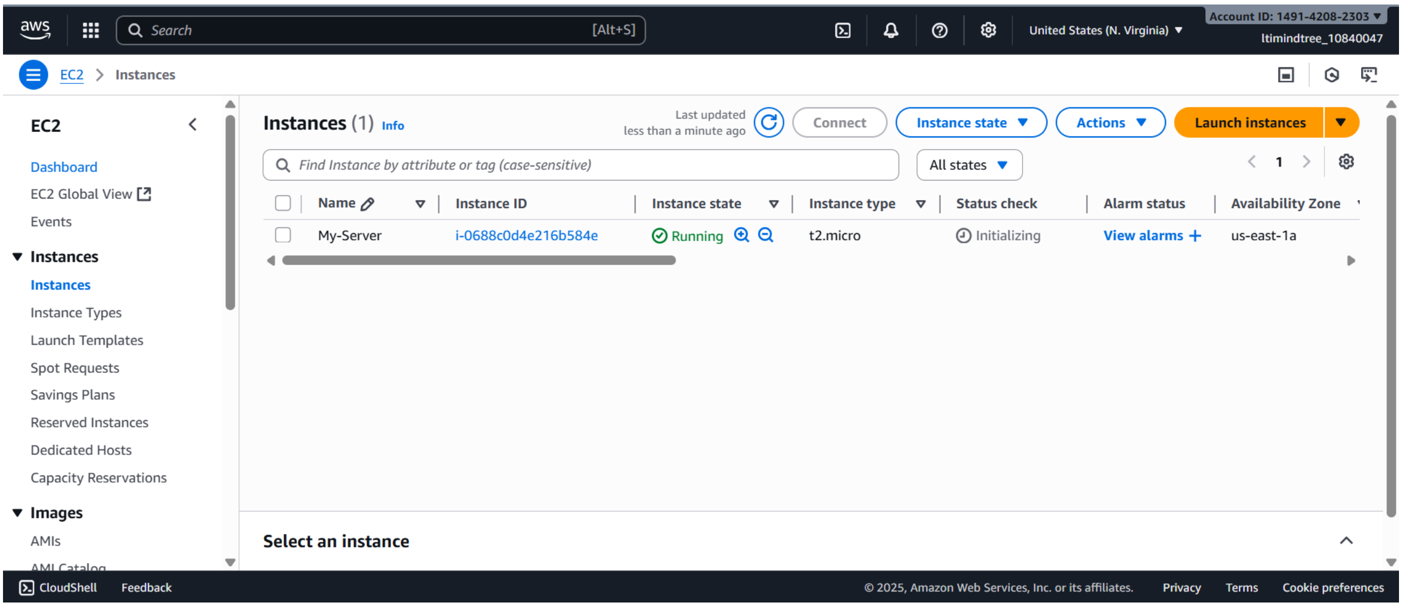
asmit-loadb-835870979.us-east-1
Not secure
asmit-loadb-835870979.us-east-1.elb.amazonaws.com
LTIMindtree Favorites Folder
Import favorites

“Hello”

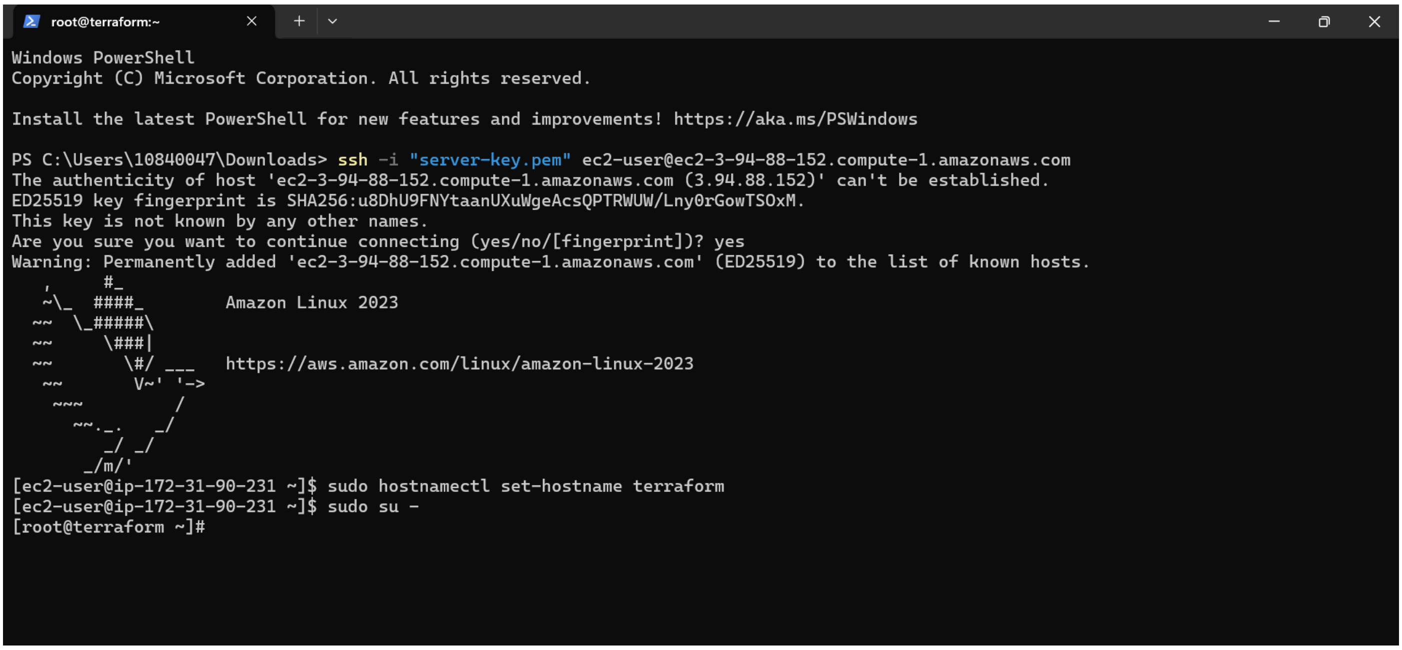
Hence the load balancer is working fine and it’s exposed on port 80.

Create an Ec2 instance in Mumbai region and attach a new security group. Where port number 22 and 80 should be allow. Using of the IAC tool terraform.

Created an instnace and connected to terminal.

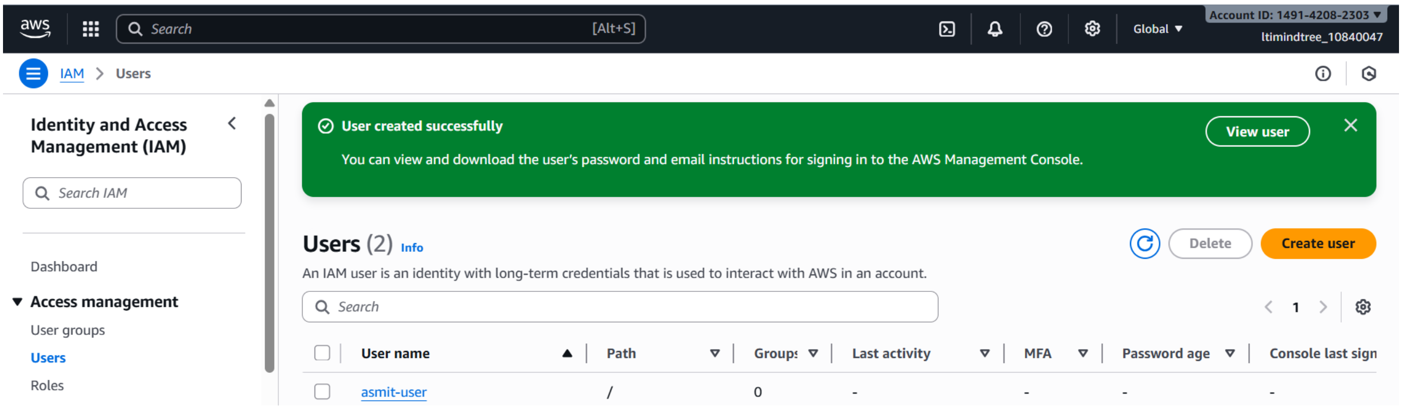


□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
□	My-Server	i-0688c0d4e216b584e	✓ Running	t2.micro	Initializing	View alarms +	us-east-1a



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\10840047\Downloads> ssh -i "server-key.pem" ec2-user@ec2-3-94-88-152.compute-1.amazonaws.com
The authenticity of host 'ec2-3-94-88-152.compute-1.amazonaws.com (3.94.88.152)' can't be established.
ED25519 key fingerprint is SHA256:u8DhU9FNYtaanUXuWgeAcsQPTRWUW/Lny0rGowTSOxM.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-94-88-152.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-90-231 ~]$ sudo hostnamectl set-hostname terraform
[ec2-user@ip-172-31-90-231 ~]$ sudo su -
[root@terraform ~]#
```

Created an IAM user with administrator access policy and generated access keys.



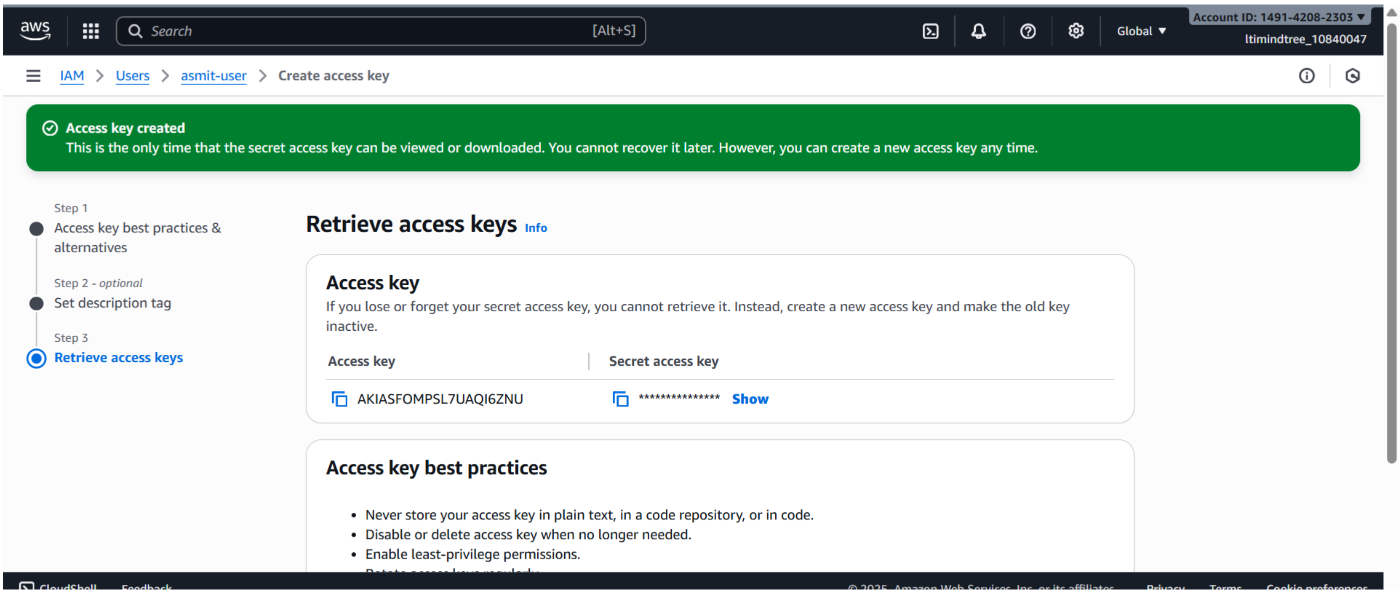
User created successfully

You can view and download the user's password and email instructions for signing in to the AWS Management Console.

Users (2)

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

□	User name ▲	Path	Group:	Last activity	MFA	Password age	Console last sign
□	asmit-user	/	0	-	-	-	-


aws
Search
[Alt+S]
Global ▼
Account ID: 1491-4208-2303 ▼
ltimindtree_10840047
IAM > Users > asmit-user > Create access key
Info
Show
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Access key created

This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time.

Step 1
● Access key best practices & alternatives

Step 2 - *optional*
● Set description tag

Step 3
◉ Retrieve access keys

Retrieve access keys

Access key

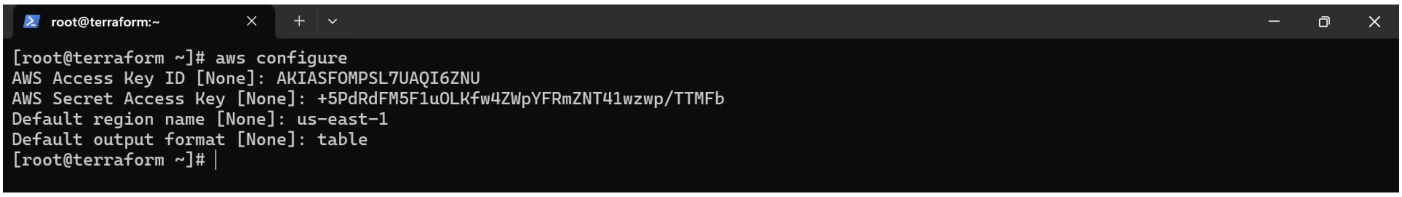
If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

Access key	Secret access key
AKIASFOMPSL7UAQI6ZNU	*************** Show

Access key best practices

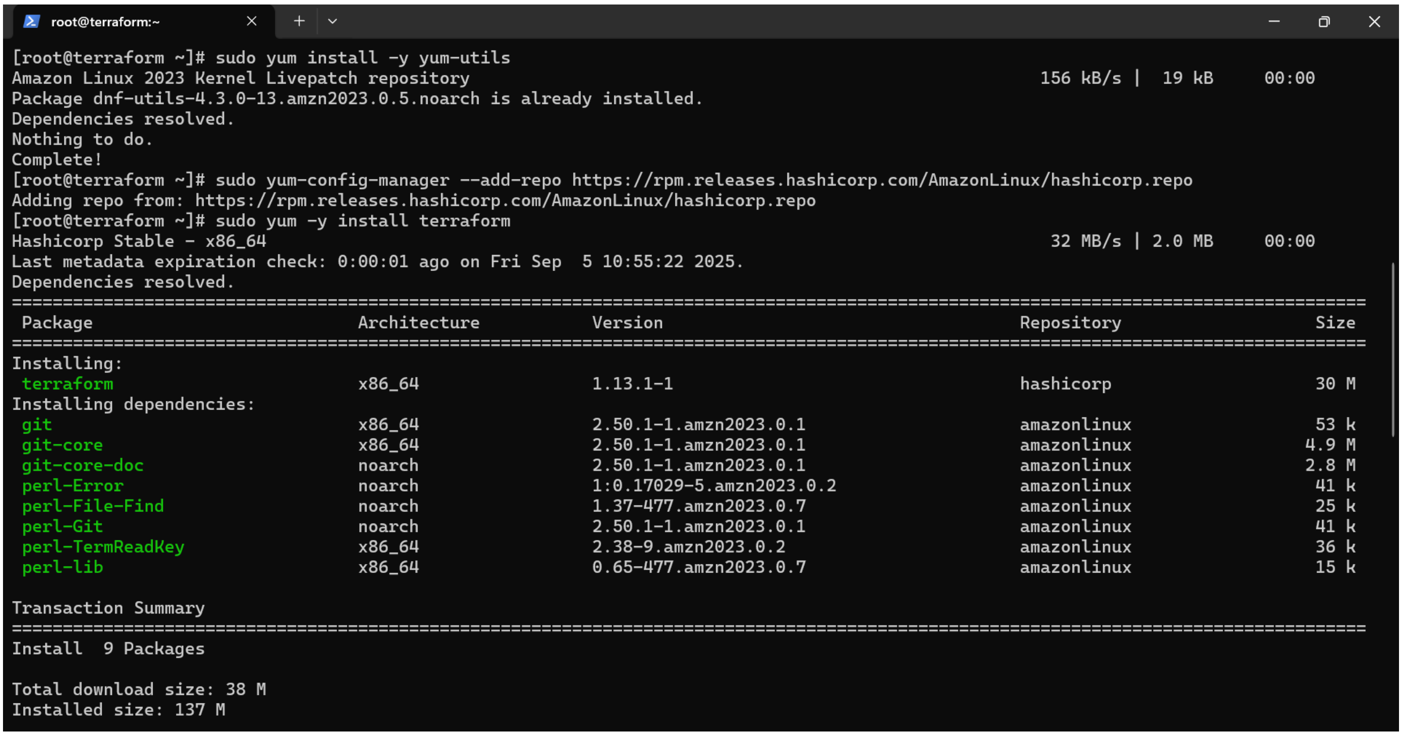
- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access key when no longer needed.
- Enable least-privilege permissions.

Configured aws in the terminal.

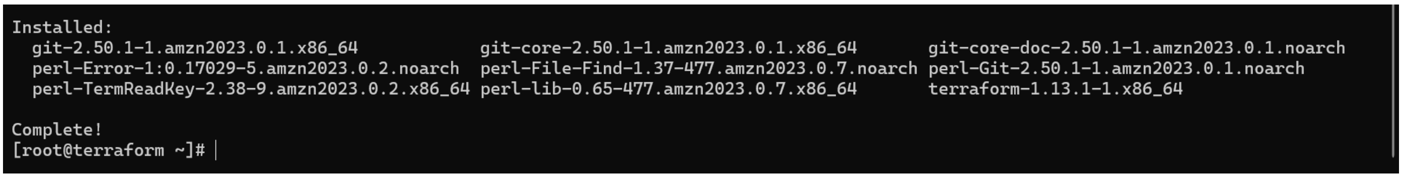


```
[root@terraform ~]# aws configure
AWS Access Key ID [None]: AKIASFOMPSL7UAQI6ZNU
AWS Secret Access Key [None]: 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
Default region name [None]: us-east-1
Default output format [None]: table
[root@terraform ~]# |
```

Installed terraform.

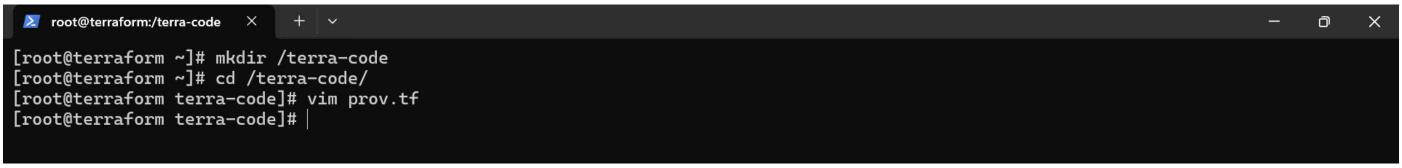


```
[root@terraform ~]# sudo yum install -y yum-utils
Amazon Linux 2023 Kernel Livepatch repository                                                156 kB/s |  19 kB     00:00
Package dnf-utils-4.3.0-13.amzn2023.0.5.noarch is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[root@terraform ~]# sudo yum-config-manager --add-repo https://rpm.releases.hashicorp.com/AmazonLinux/hashicorp.repo
Adding repo from: https://rpm.releases.hashicorp.com/AmazonLinux/hashicorp.repo
[root@terraform ~]# sudo yum -y install terraform
Hashicorp Stable - x86_64                                                                     32 MB/s | 2.0 MB     00:00
Last metadata expiration check: 0:00:01 ago on Fri Sep  5 10:55:22 2025.
Dependencies resolved.
=====================================================================================================================================
 Package                          Architecture          Version                                  Repository                  Size
=====================================================================================================================================
Installing:
 terraform                        x86_64                1.13.1-1                                 hashicorp                   30 M
Installing dependencies:
 git                              x86_64                2.50.1-1.amzn2023.0.1                    amazonlinux                 53 k
 git-core                         x86_64                2.50.1-1.amzn2023.0.1                    amazonlinux                4.9 M
 git-core-doc                     noarch                2.50.1-1.amzn2023.0.1                    amazonlinux                2.8 M
 perl-Error                       noarch                1:0.17029-5.amzn2023.0.2                 amazonlinux                 41 k
 perl-File-Find                   noarch                1.37-477.amzn2023.0.7                    amazonlinux                 25 k
 perl-Git                         noarch                2.50.1-1.amzn2023.0.1                    amazonlinux                 41 k
 perl-TermReadKey                 x86_64                2.38-9.amzn2023.0.2                      amazonlinux                 36 k
 perl-lib                         x86_64                0.65-477.amzn2023.0.7                    amazonlinux                 15 k

Transaction Summary
=====================================================================================================================================
Install  9 Packages

Total download size: 38 M
Installed size: 137 M
```



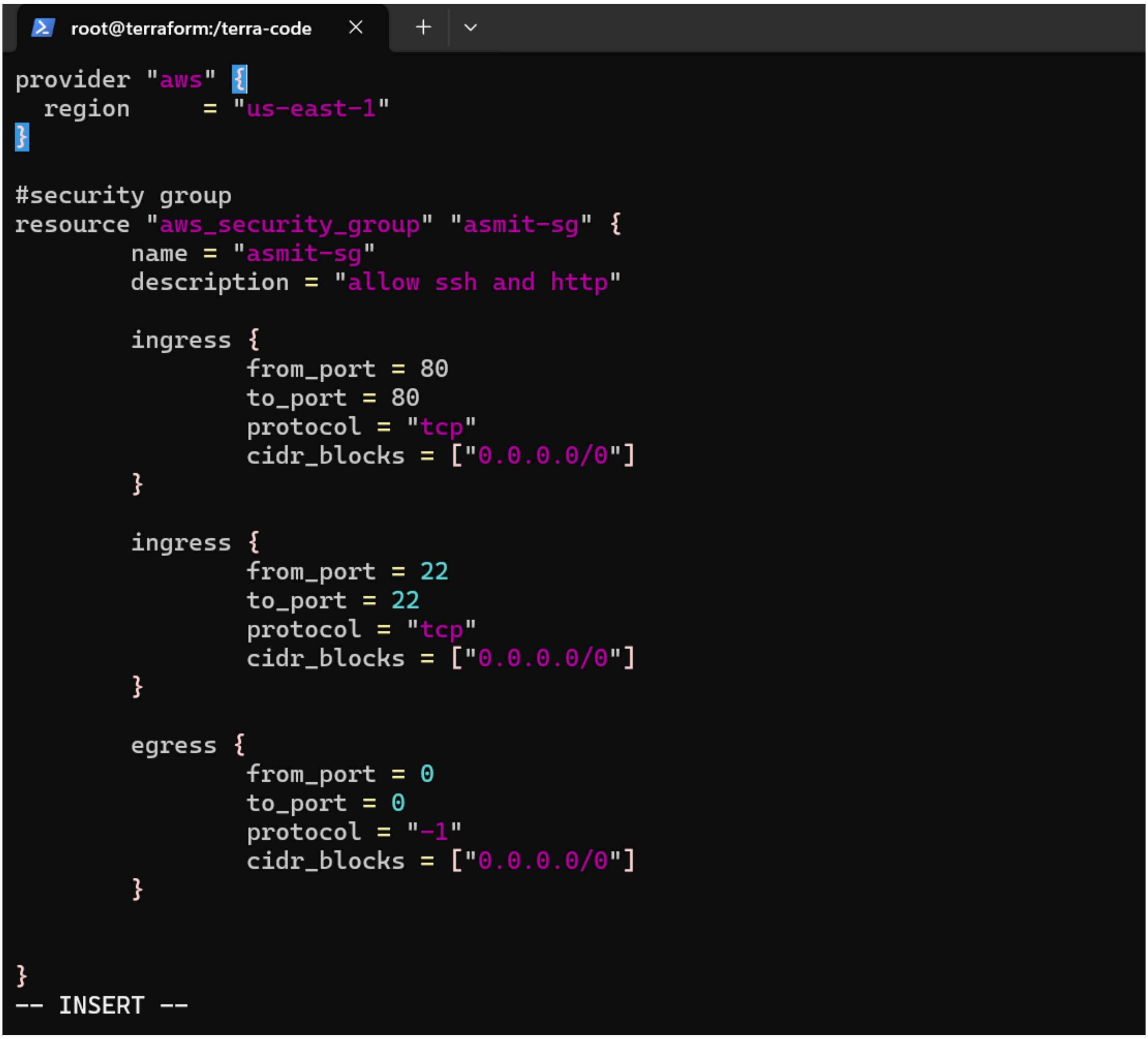
```
Installed:
  git-2.50.1-1.amzn2023.0.1.x86_64            git-core-2.50.1-1.amzn2023.0.1.x86_64       git-core-doc-2.50.1-1.amzn2023.0.1.noarch
  perl-Error-1:0.17029-5.amzn2023.0.2.noarch  perl-File-Find-1.37-477.amzn2023.0.7.noarch perl-Git-2.50.1-1.amzn2023.0.1.noarch
  perl-TermReadKey-2.38-9.amzn2023.0.2.x86_64 perl-lib-0.65-477.amzn2023.0.7.x86_64       terraform-1.13.1-1.x86_64

Complete!
[root@terraform ~]#
```

Created a directory and a .tf file

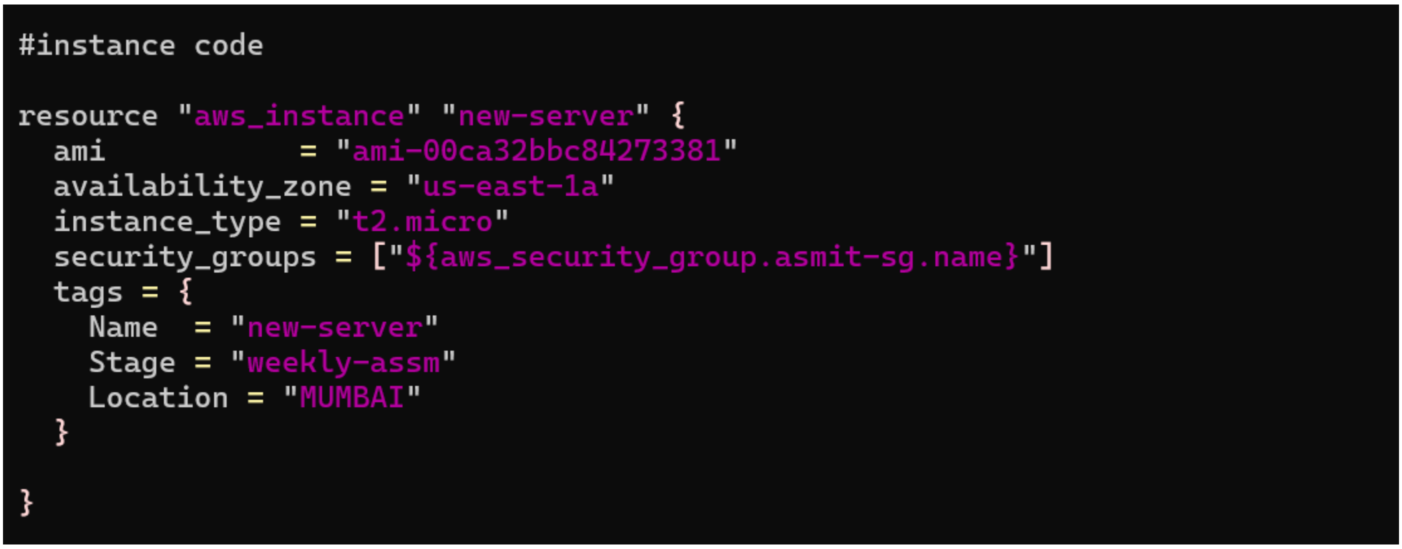


```
[root@terraform ~]# mkdir /terra-code
[root@terraform ~]# cd /terra-code/
[root@terraform terra-code]# vim prov.tf
[root@terraform terra-code]#
```

Wrote the code to create a security group and an ec2 instance.

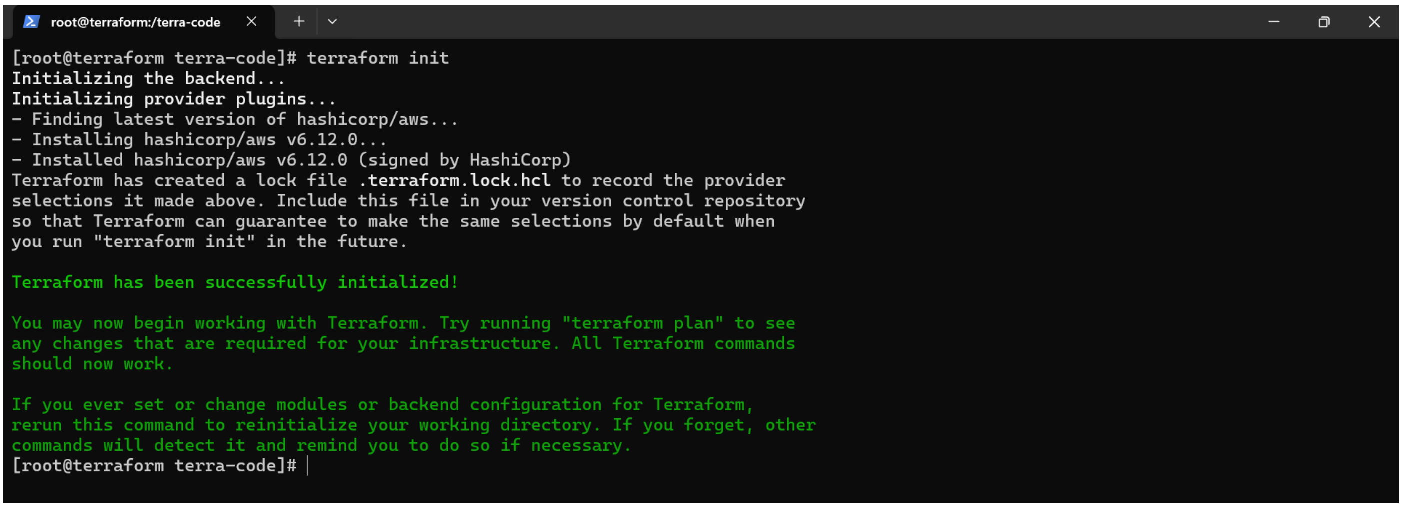


```
provider "aws" {
  region     = "us-east-1"
}

#security group
resource "aws_security_group" "asmit-sg" {
        name = "asmit-sg"
        description = "allow ssh and http"

        ingress {
                from_port = 80
                to_port = 80
                protocol = "tcp"
                cidr_blocks = ["0.0.0.0/0"]
        }

        ingress {
                from_port = 22
                to_port = 22
                protocol = "tcp"
                cidr_blocks = ["0.0.0.0/0"]
        }

        egress {
                from_port = 0
                to_port = 0
                protocol = "-1"
                cidr_blocks = ["0.0.0.0/0"]
        }


}
-- INSERT --
```



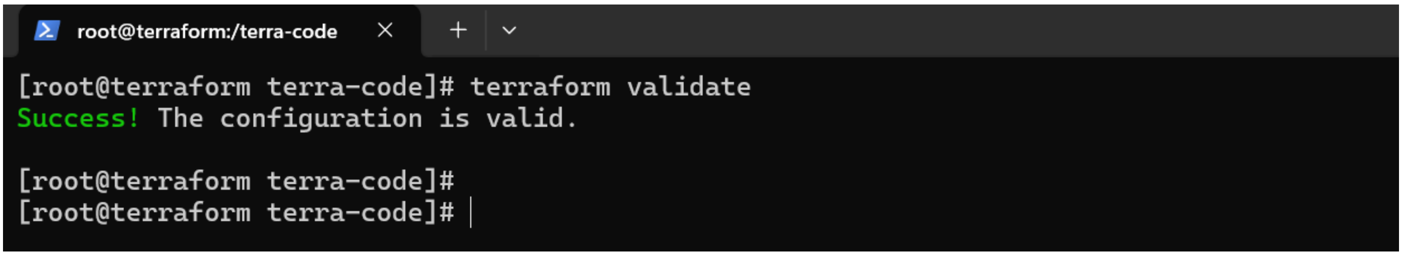
```
#instance code

resource "aws_instance" "new-server" {
  ami           = "ami-00ca32bbc84273381"
  availability_zone = "us-east-1a"
  instance_type = "t2.micro"
  security_groups = ["${aws_security_group.asmit-sg.name}"]
  tags = {
    Name  = "new-server"
    Stage = "weekly-assm"
    Location = "MUMBAI"
  }

}
```

Then initialized terraform in the directory.

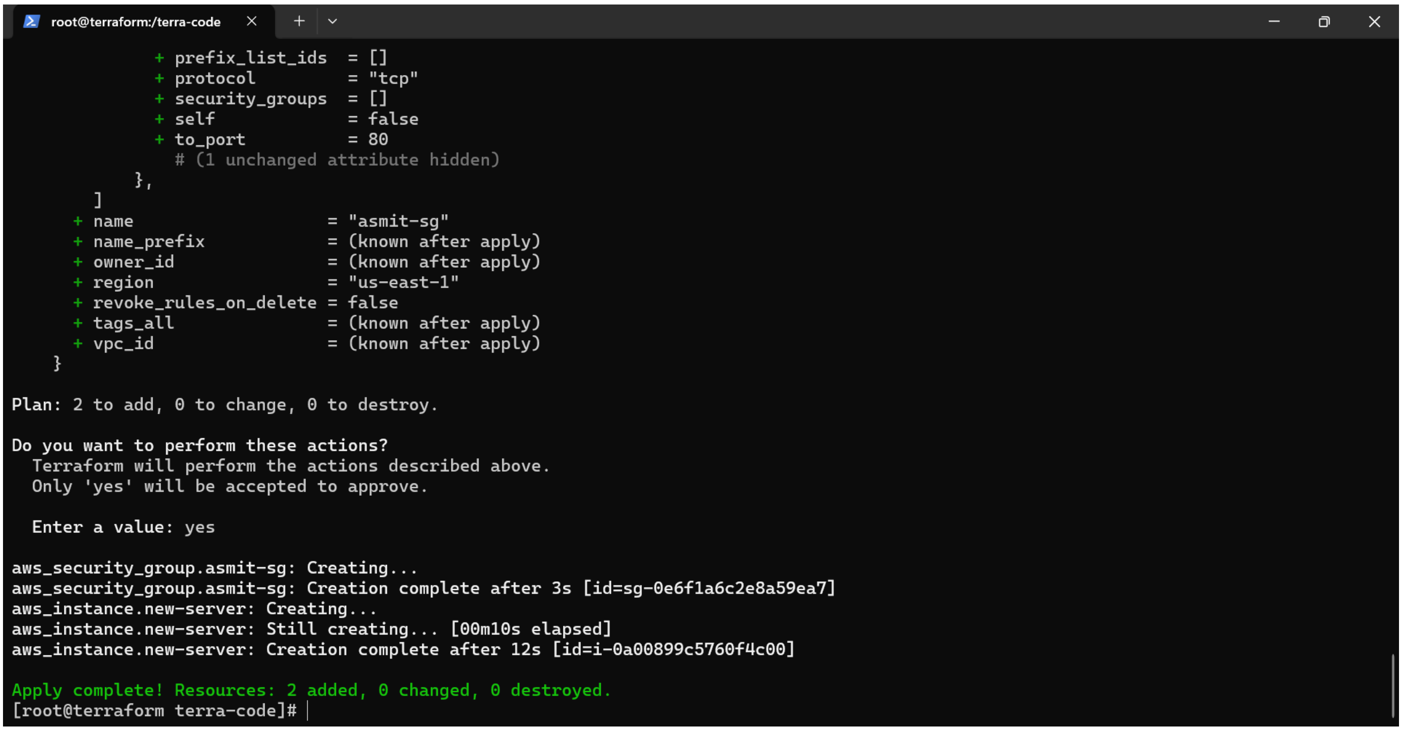


```
[root@terraform terra-code]# terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v6.12.0...
- Installed hashicorp/aws v6.12.0 (signed by HashiCorp)
Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
[root@terraform terra-code]#
```

Validated and applied it.

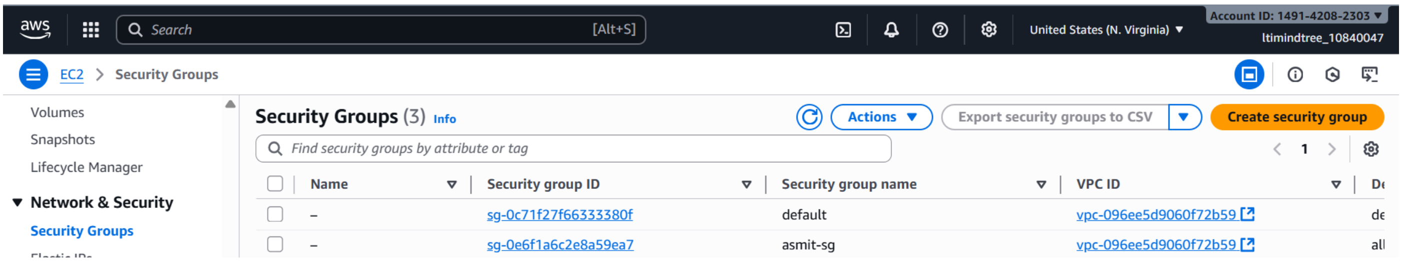


```
[root@terraform terra-code]# terraform validate
Success! The configuration is valid.

[root@terraform terra-code]#
[root@terraform terra-code]#
```

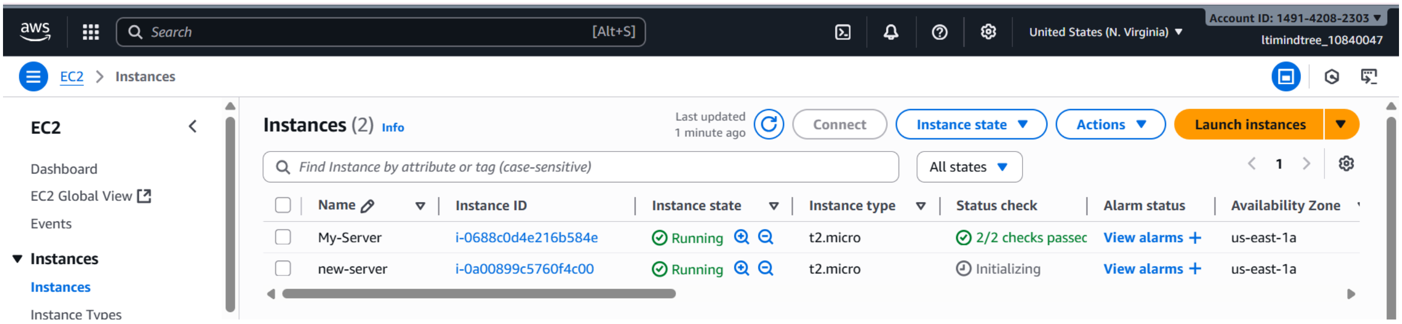


```
              + prefix_list_ids  = []
              + protocol         = "tcp"
              + security_groups  = []
              + self             = false
              + to_port          = 80
                # (1 unchanged attribute hidden)
            },
        ]
      + name                   = "asmit-sg"
      + name_prefix            = (known after apply)
      + owner_id               = (known after apply)
      + region                 = "us-east-1"
      + revoke_rules_on_delete = false
      + tags_all               = (known after apply)
      + vpc_id                 = (known after apply)
    }

Plan: 2 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

aws_security_group.asmit-sg: Creating...
aws_security_group.asmit-sg: Creation complete after 3s [id=sg-0e6f1a6c2e8a59ea7]
aws_instance.new-server: Creating...
aws_instance.new-server: Still creating... [00m10s elapsed]
aws_instance.new-server: Creation complete after 12s [id=i-0a00899c5760f4c00]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
[root@terraform terra-code]#
```

In aws console a new security group named “asmit-sg” and an ec2 instance named “new-server” that I defined in the code is craeted.



Name	Security group ID	Security group name	VPC ID	De
–	sg-0c71f27f66333380f	default	vpc-096ee5d9060f72b59	de
–	sg-0e6f1a6c2e8a59ea7	asmit-sg	vpc-096ee5d9060f72b59	all



Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
My-Server	i-0688c0d4e216b584e	Running	t2.micro	2/2 checks passec	View alarms +	us-east-1a
new-server	i-0a00899c5760f4c00	Running	t2.micro	Initializing	View alarms +	us-east-1a