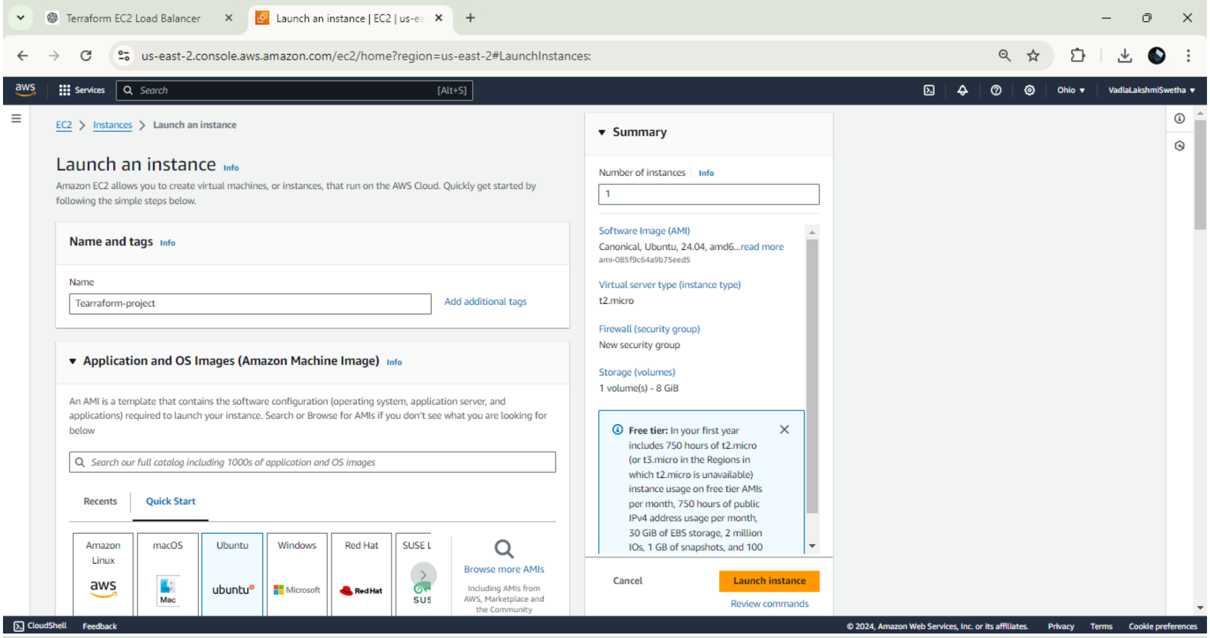
PROJECT-TERRAFORM

Create below infra using terraform

1. Create two virtual machines in east us (web servers).

2. Configure load balancer for above servers.


Terraform EC2 Load Balancer
Launch an instance | EC2 | us-ea
us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#LaunchInstances:
aws
Services
Search
[Alt+S]
Ohio ▾
VadlaLakshmiSwetha ▾
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: Tearraform-project

Add additional tags

▾ Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

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

▾ Summary

Number of instances Info: 1

Software Image (AMI): Canonical, Ubuntu, 24.04, amd6...read more
ami-085f9c64a9b75eed5

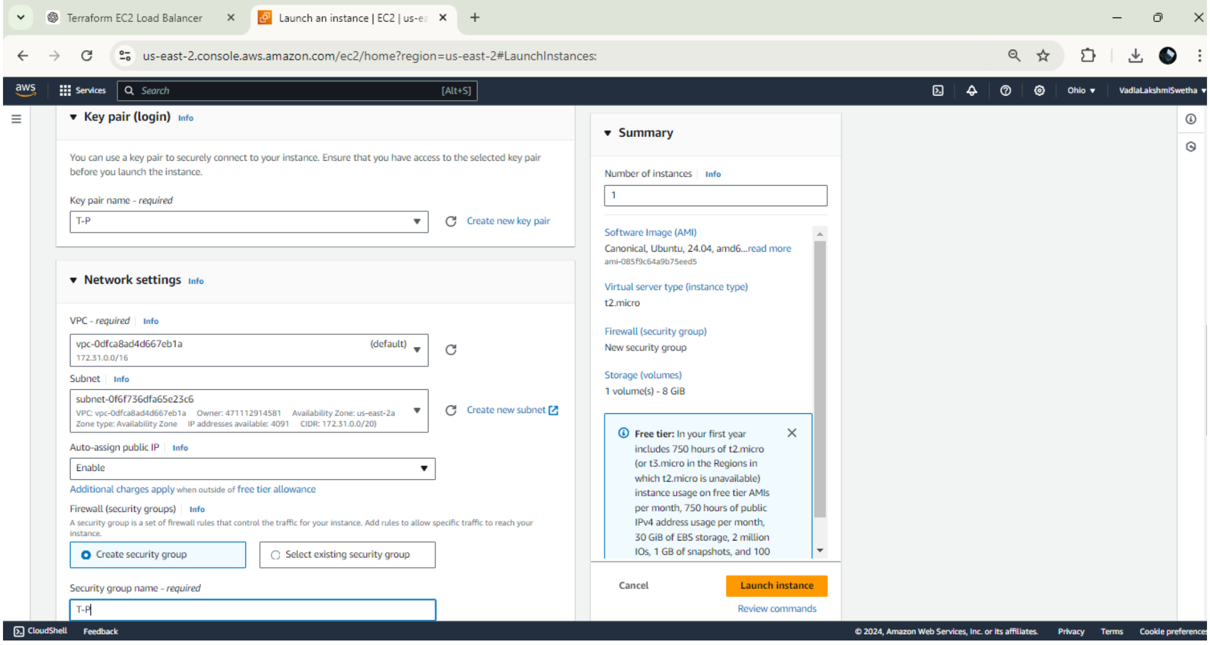
Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100

Cancel
Launch instance
Review commands
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Terraform EC2 Load Balancer
Launch an instance | EC2 | us-ea
us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#LaunchInstances:
aws
Services
Search
[Alt+S]
Ohio ▾
VadlaLakshmiSwetha ▾

▾ Key pair (login) Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*: T-P

Create new key pair

▾ Network settings Info

VPC - *required* Info: vpc-0dfca8ad4d667eb1a (default)
172.31.0.0/16

Subnet Info: subnet-0f6f736dfa65e23c6
VPC: vpc-0dfca8ad4d667eb1a Owner: 471112914581 Availability Zone: us-east-2a
Zone type: Availability Zone IP addresses available: 4091 CIDR: 172.31.0.0/20)

Create new subnet

Auto-assign public IP Info: Enable

Additional charges apply when outside of free tier allowance

Firewall (security groups) Info

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*: T-P

▾ Summary

Number of instances Info: 1

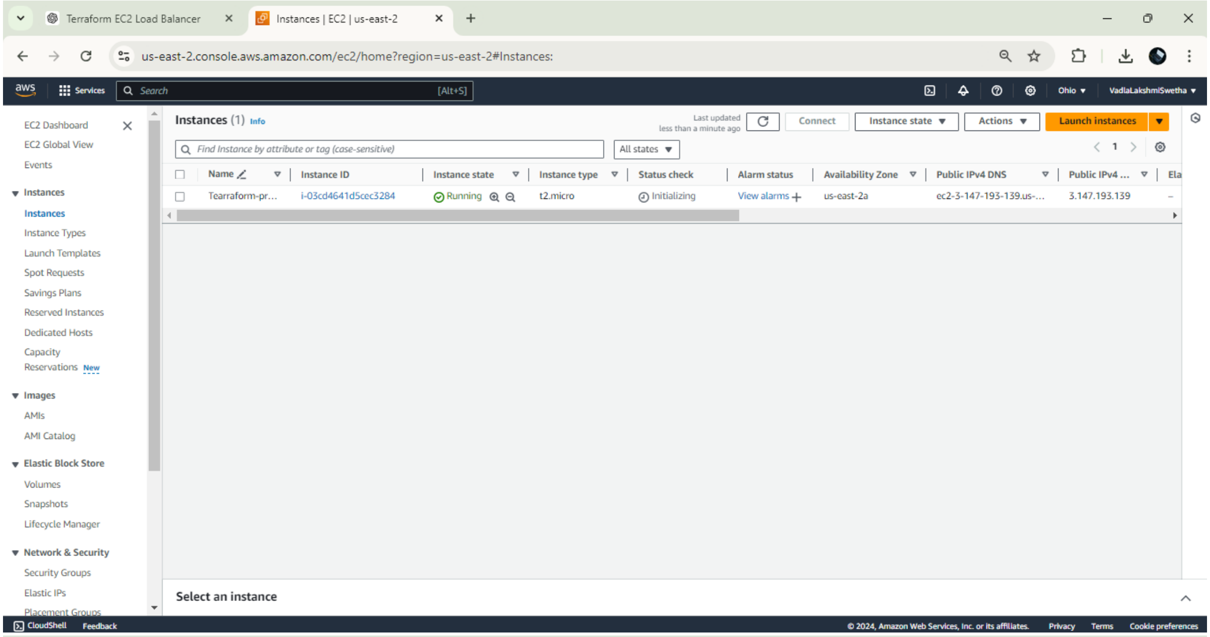
Software Image (AMI): Canonical, Ubuntu, 24.04, amd6...read more
ami-085f9c64a9b75eed5

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100

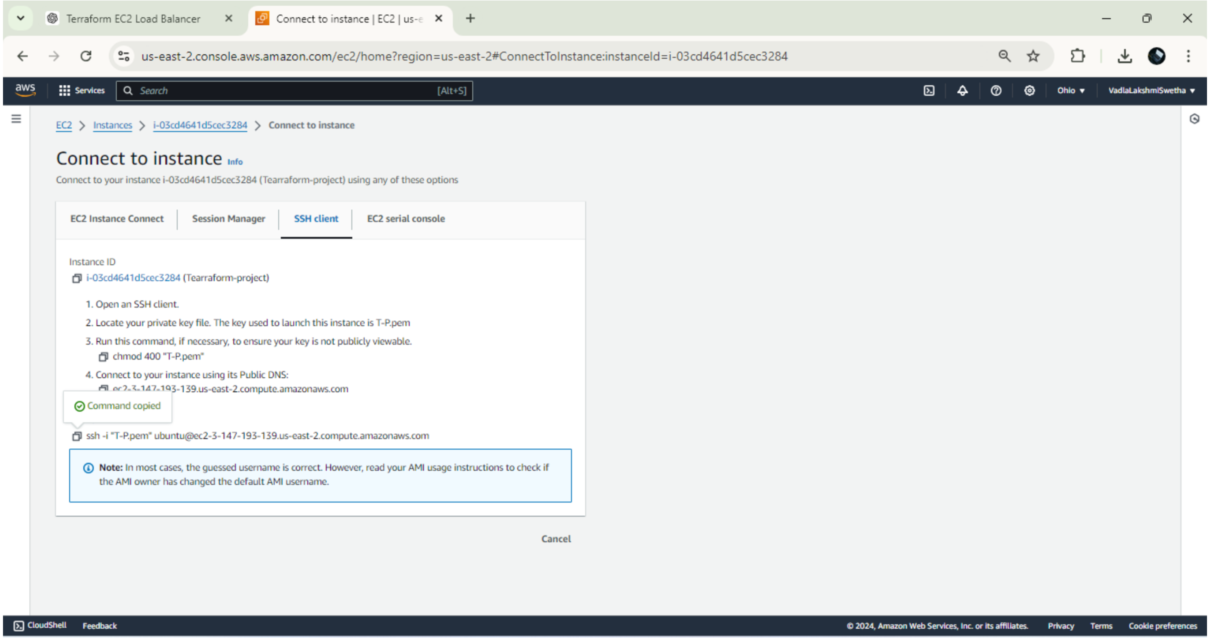


Instances (1) Info

Find Instance by attribute or tag (case-sensitive)

□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Ela
□	Tearraform-pr...	i-03cd4641d5cec3284	Running	t2.micro	Initializing	View alarms +	us-east-2a	ec2-3-147-193-139.us-...	3.147.193.139	–

Select an instance


Terraform EC2 Load Balancer
Connect to instance | EC2 | us-e
us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#ConnectToInstance:instanceId=i-03cd4641d5cec3284
Services
Search
[Alt+S]
Ohio
VadlaLakshmiSwetha
EC2 > Instances > i-03cd4641d5cec3284 > Connect to instance
EC2 Instance Connect
Session Manager
SSH client
EC2 serial console
Command copied
Cancel
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Connect to instance Info

Connect to your instance i-03cd4641d5cec3284 (Tearraform-project) using any of these options

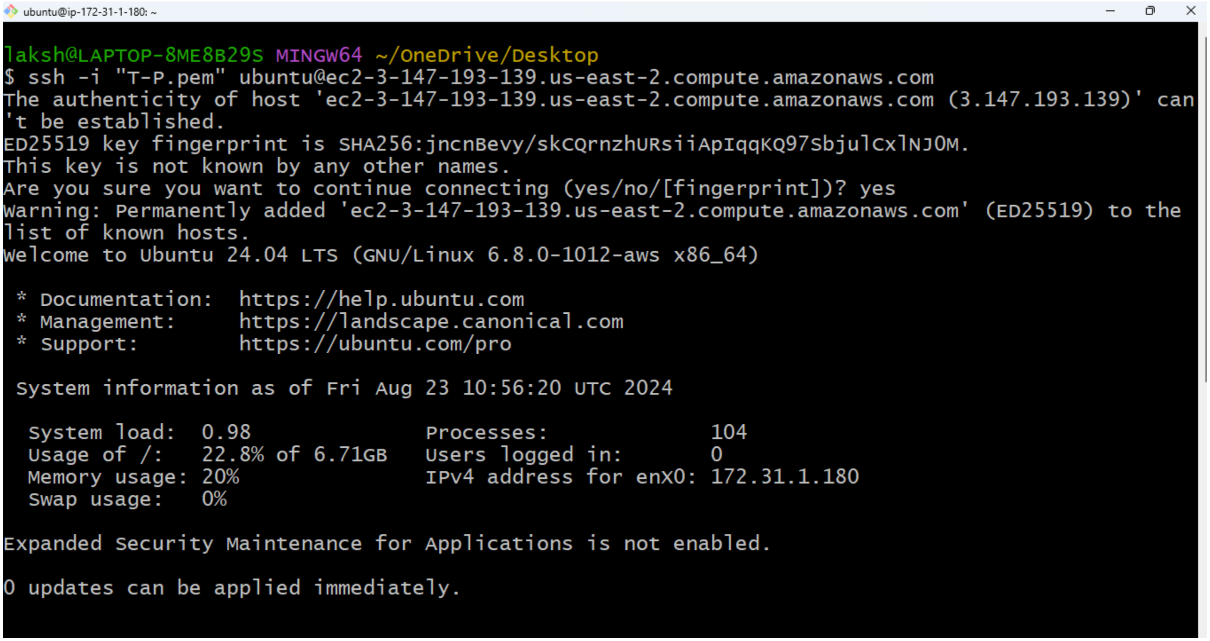
Instance ID

i-03cd4641d5cec3284 (Tearraform-project)

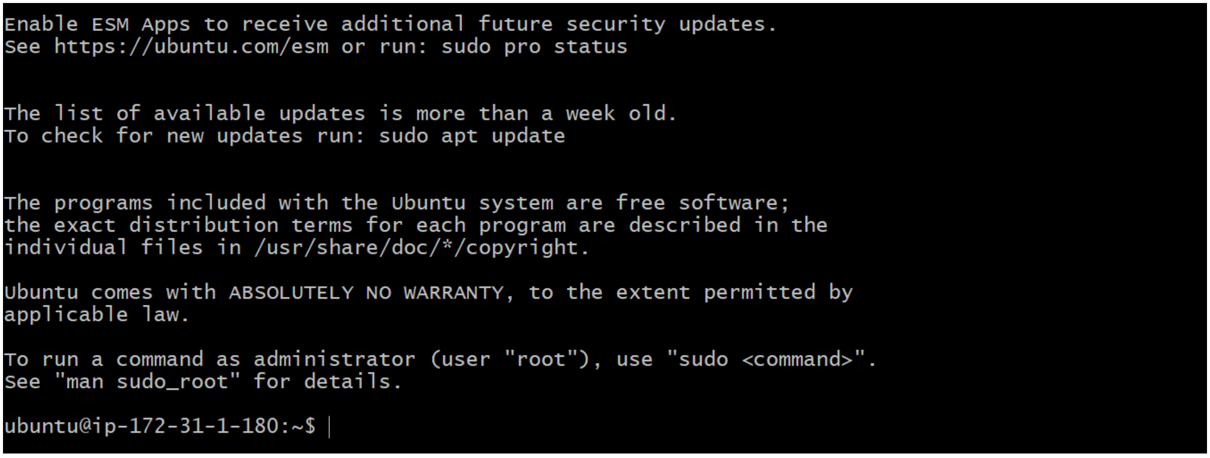
1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is T-P.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
   ```
   chmod 400 "T-P.pem"
   ```
4. Connect to your instance using its Public DNS:
   ec2-3-147-193-139.us-east-2.compute.amazonaws.com

```
ssh -i "T-P.pem" ubuntu@ec2-3-147-193-139.us-east-2.compute.amazonaws.com
```

Note: In most cases, the guessed username is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.



```
laksh@LAPTOP-8ME8B29S MINGW64 ~/OneDrive/Desktop
$ ssh -i "T-P.pem" ubuntu@ec2-3-147-193-139.us-east-2.compute.amazonaws.com
The authenticity of host 'ec2-3-147-193-139.us-east-2.compute.amazonaws.com (3.147.193.139)' can
't be established.
ED25519 key fingerprint is SHA256:jncnBevy/skCQrnzhURsiiApIqqKQ97SbjulCxlNJOM.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-147-193-139.us-east-2.compute.amazonaws.com' (ED25519) to the
list of known hosts.
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1012-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Fri Aug 23 10:56:20 UTC 2024

  System load:  0.98              Processes:             104
  Usage of /:   22.8% of 6.71GB   Users logged in:       0
  Memory usage: 20%               IPv4 address for enX0: 172.31.1.180
  Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.
```



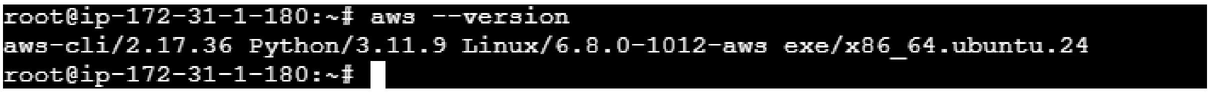
```
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-1-180:~$ |
```

AWS CLI INSTALL:

curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"

unzip awscliv2.zip

sudo ./aws/install



```
root@ip-172-31-1-180:~# aws --version
aws-cli/2.17.36 Python/3.11.9 Linux/6.8.0-1012-aws exe/x86_64.ubuntu.24
root@ip-172-31-1-180:~#
```

TERRAFORM DOWNLOAD:-

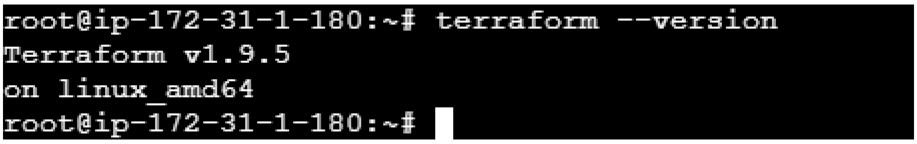
sudo apt-get update

sudo apt-get install -y gnupg software-properties-common curl

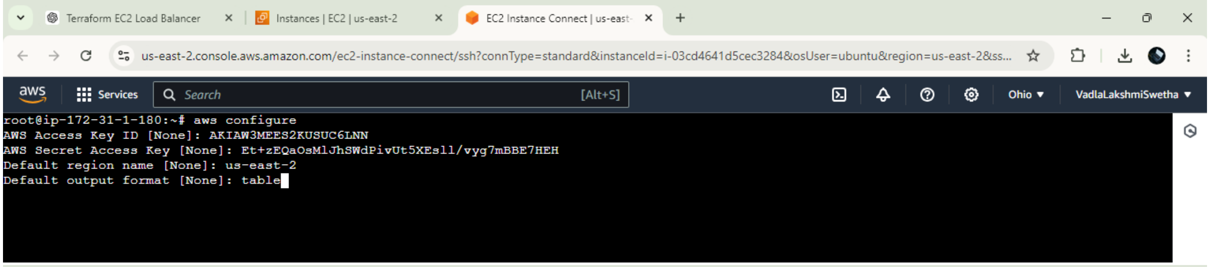
curl -fsSL https://apt.releases.hashicorp.com/gpg | sudo gpg --dearmor -o /usr/share/keyrings/hashicorp-archive-keyring.gpg echo "deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] https://apt.releases.hashicorp.com $(lsb_release -cs) main" | sudo tee /etc/apt/sources.list.d/hashicorp.list

sudo apt-get update

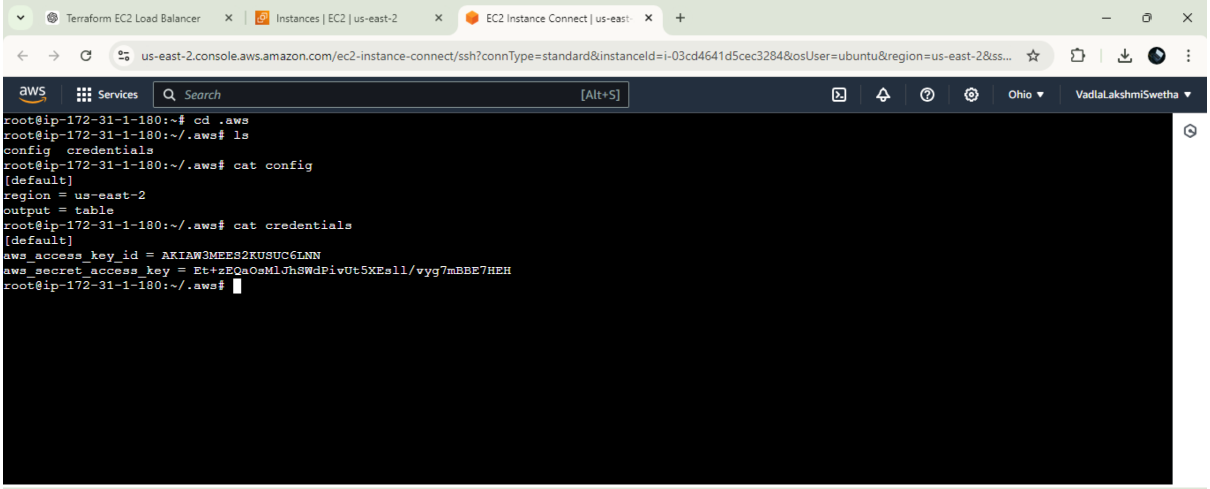
sudo apt-get install terraform



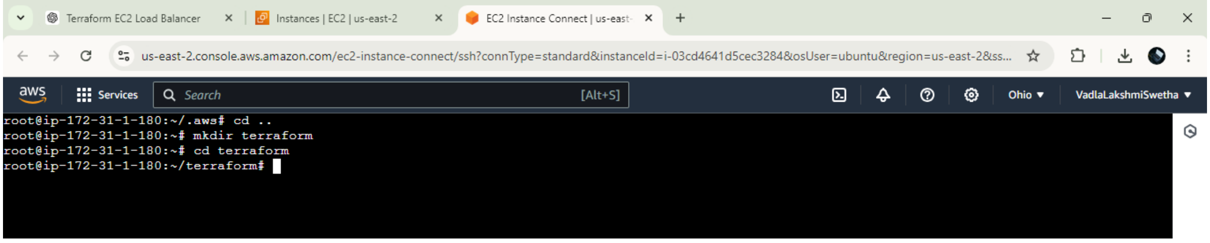
```
root@ip-172-31-1-180:~# terraform --version
Terraform v1.9.5
on linux_amd64
root@ip-172-31-1-180:~# 
```


Terraform EC2 Load Balancer
Instances | EC2 | us-east-2
EC2 Instance Connect | us-east
us-east-2.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-03cd4641d5cec3284&osUser=ubuntu&region=us-east-2&ss...
aws
Services
Search
[Alt+S]
Ohio
VadlaLakshmiSwetha

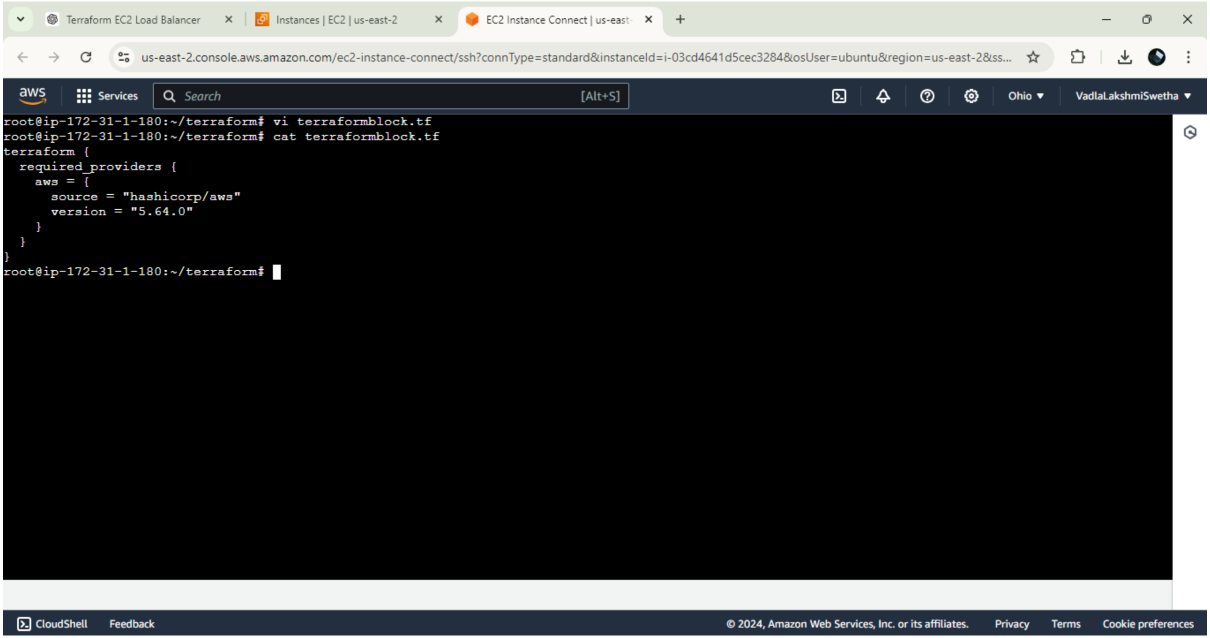
```
root@ip-172-31-1-180:~# aws configure
AWS Access Key ID [None]: AKIAW3MEES2KUSUC6LNN
AWS Secret Access Key [None]: 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
Default region name [None]: us-east-2
Default output format [None]: table
```


Terraform EC2 Load Balancer
Instances | EC2 | us-east-2
EC2 Instance Connect | us-east
us-east-2.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-03cd4641d5cec3284&osUser=ubuntu&region=us-east-2&ss...
aws
Services
Search
[Alt+S]
Ohio
VadlaLakshmiSwetha

```
root@ip-172-31-1-180:~# cd .aws
root@ip-172-31-1-180:~/.aws# ls
config  credentials
root@ip-172-31-1-180:~/.aws# cat config
[default]
region = us-east-2
output = table
root@ip-172-31-1-180:~/.aws# cat credentials
[default]
aws_access_key_id = AKIAW3MEES2KUSUC6LNN
aws_secret_access_key = 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
root@ip-172-31-1-180:~/.aws# 
```

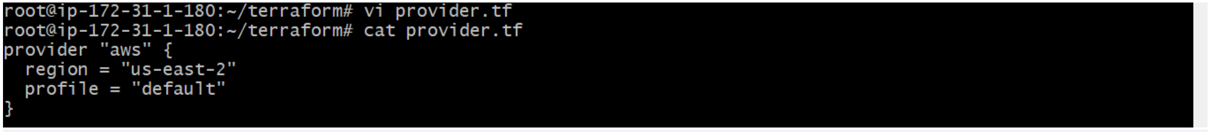

Terraform EC2 Load Balancer
Instances | EC2 | us-east-2
EC2 Instance Connect | us-east
us-east-2.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-03cd4641d5cec3284&osUser=ubuntu&region=us-east-2&ss...
aws
Services
Search
[Alt+S]
Ohio
VadlaLakshmiSwetha

```
root@ip-172-31-1-180:~/.aws# cd ..
root@ip-172-31-1-180:~# mkdir terraform
root@ip-172-31-1-180:~# cd terraform
root@ip-172-31-1-180:~/terraform# 
```

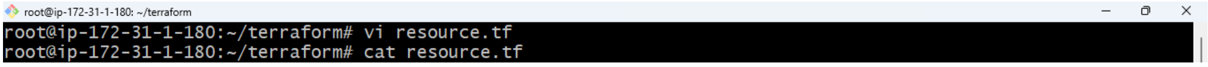


Terraform EC2 Load Balancer
Instances | EC2 | us-east-2
EC2 Instance Connect | us-east-
us-east-2.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-03cd4641d5cec3284&osUser=ubuntu&region=us-east-2&ss...
aws
Services
Search
[Alt+S]
Ohio ▾
VadlaLakshmiSwetha ▾
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

```
root@ip-172-31-1-180:~/terraform# vi terraformblock.tf
root@ip-172-31-1-180:~/terraform# cat terraformblock.tf
terraform {
  required_providers {
    aws = {
      source = "hashicorp/aws"
      version = "5.64.0"
    }
  }
}
root@ip-172-31-1-180:~/terraform# 
```



```
root@ip-172-31-1-180:~/terraform# vi provider.tf
root@ip-172-31-1-180:~/terraform# cat provider.tf
provider "aws" {
  region = "us-east-2"
  profile = "default"
}
```

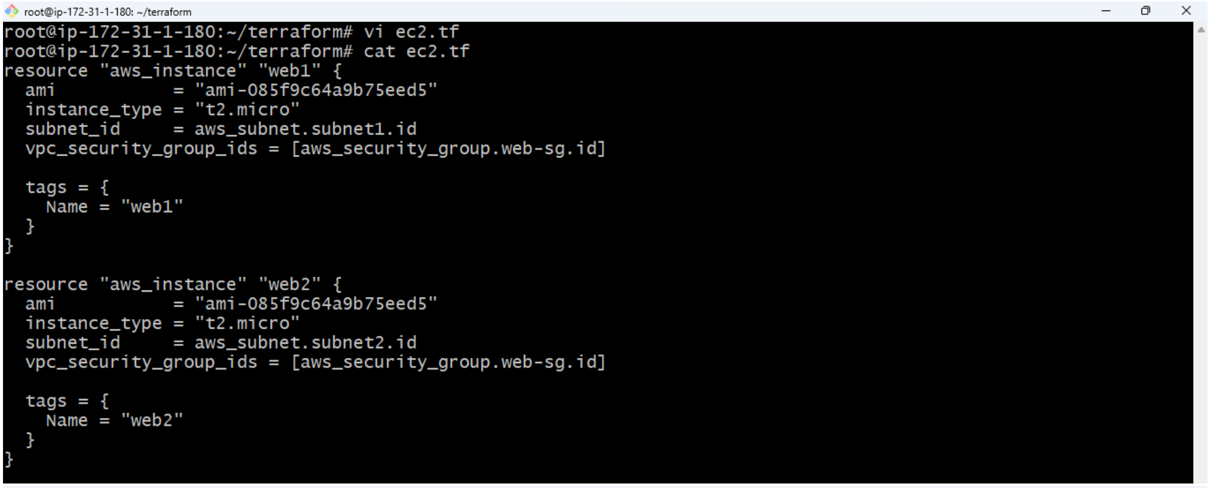


```
root@ip-172-31-1-180:~/terraform# vi resource.tf
root@ip-172-31-1-180:~/terraform# cat resource.tf
```

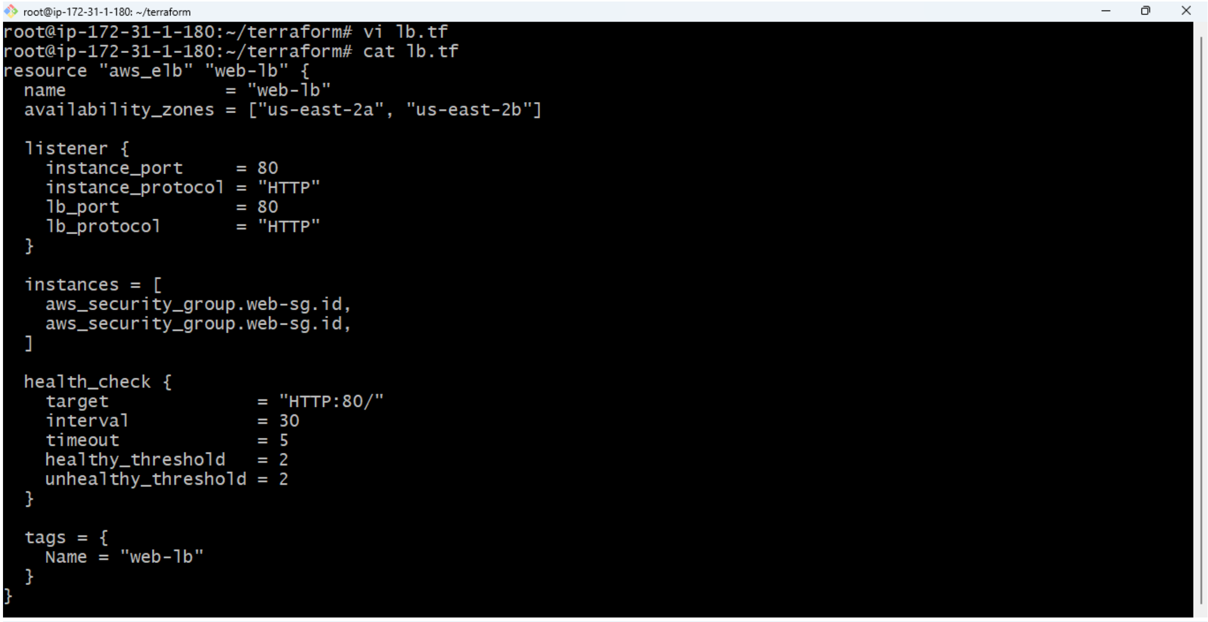
```
resource "aws_vpc" "main" {
  cidr_block = "10.0.0.0/16"

  tags = {
    Name = "main-vpc"
  }
}
resource "aws_subnet" "subnet1" {
  vpc_id            = aws_vpc.main.id
  cidr_block        = "10.0.1.0/24"
  availability_zone = "us-east-2a"

  tags = {
    Name = "subnet1"
  }
}
```

```
resource "aws_subnet" "subnet2" {
  vpc_id            = aws_vpc.main.id
  cidr_block        = "10.0.2.0/24"
  availability_zone = "us-east-2b"

  tags = {
    Name = "subnet2"
  }
}
resource "aws_internet_gateway" "igw" {
  vpc_id = aws_vpc.main.id

  tags = {
    Name = "main-igw"
  }
}
resource "aws_route_table" "rt" {
  vpc_id = aws_vpc.main.id

  route {
    cidr_block = "0.0.0.0/0"
    gateway_id = aws_internet_gateway.igw.id
  }

  tags = {
    Name = "main-route-table"
  }
}
```

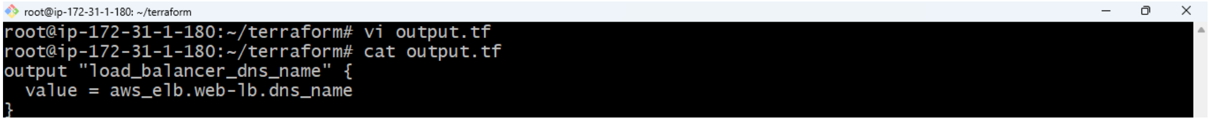
```
resource "aws_route_table_association" "subnet1-association" {
  subnet_id      = aws_subnet.subnet1.id
  route_table_id = aws_route_table.rt.id
}

resource "aws_route_table_association" "subnet2-association" {
  subnet_id      = aws_subnet.subnet2.id
  route_table_id = aws_route_table.rt.id
}
resource "aws_security_group" "web-sg" {
  vpc_id = aws_vpc.main.id

  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"]
  }

  tags = {
    Name = "web-sg"
  }
}
```



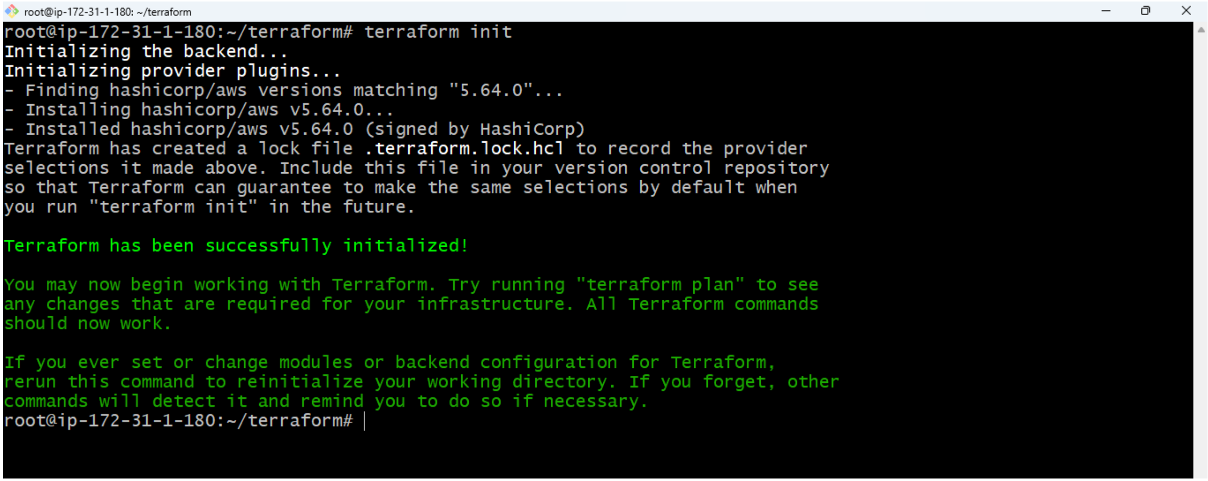
```
root@ip-172-31-1-180:~/terraform# vi ec2.tf
root@ip-172-31-1-180:~/terraform# cat ec2.tf
resource "aws_instance" "web1" {
  ami           = "ami-085f9c64a9b75eed5"
  instance_type = "t2.micro"
  subnet_id     = aws_subnet.subnet1.id
  vpc_security_group_ids = [aws_security_group.web-sg.id]

  tags = {
    Name = "web1"
  }
}

resource "aws_instance" "web2" {
  ami           = "ami-085f9c64a9b75eed5"
  instance_type = "t2.micro"
  subnet_id     = aws_subnet.subnet2.id
  vpc_security_group_ids = [aws_security_group.web-sg.id]

  tags = {
    Name = "web2"
  }
}
```



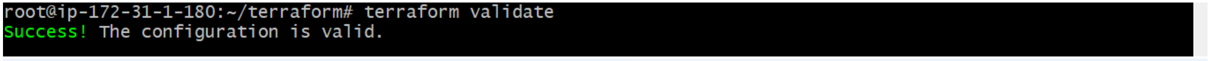
```
root@ip-172-31-1-180:~/terraform# vi lb.tf
root@ip-172-31-1-180:~/terraform# cat lb.tf
resource "aws_elb" "web-lb" {
  name               = "web-lb"
  availability_zones = ["us-east-2a", "us-east-2b"]

  listener {
    instance_port     = 80
    instance_protocol = "HTTP"
    lb_port           = 80
    lb_protocol       = "HTTP"
  }

  instances = [
    aws_security_group.web-sg.id,
    aws_security_group.web-sg.id,
  ]

  health_check {
    target              = "HTTP:80/"
    interval            = 30
    timeout             = 5
    healthy_threshold   = 2
    unhealthy_threshold = 2
  }

  tags = {
    Name = "web-lb"
  }
}
```



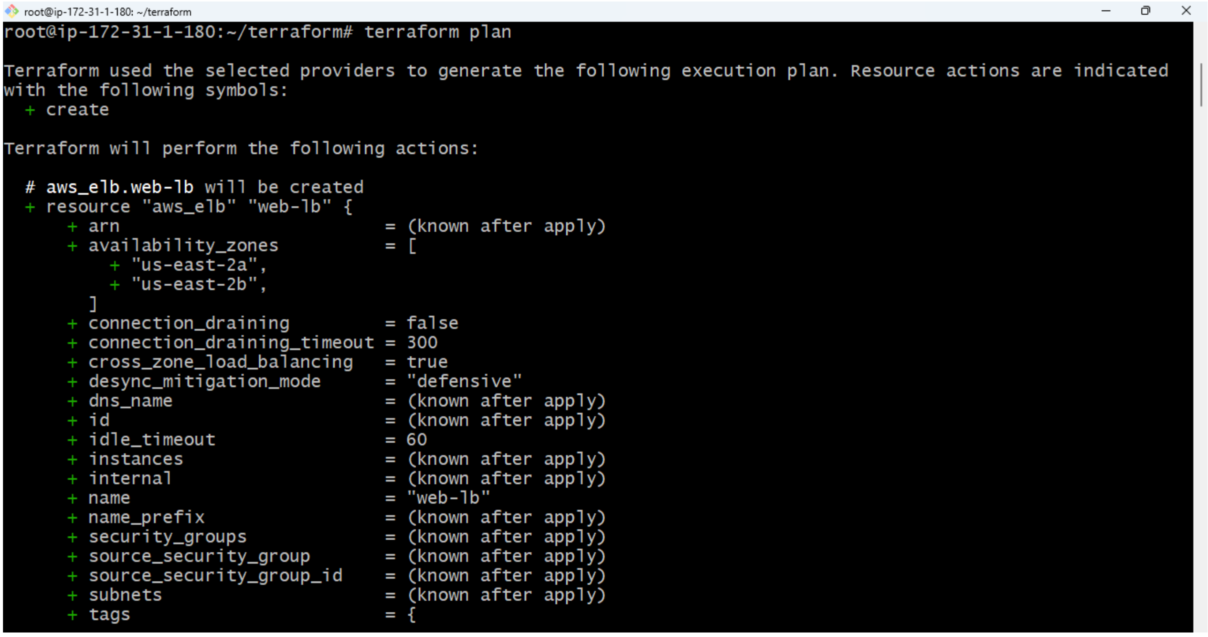
```
root@ip-172-31-1-180:~/terraform# vi output.tf
root@ip-172-31-1-180:~/terraform# cat output.tf
output "load_balancer_dns_name" {
  value = aws_elb.web-lb.dns_name
}
```



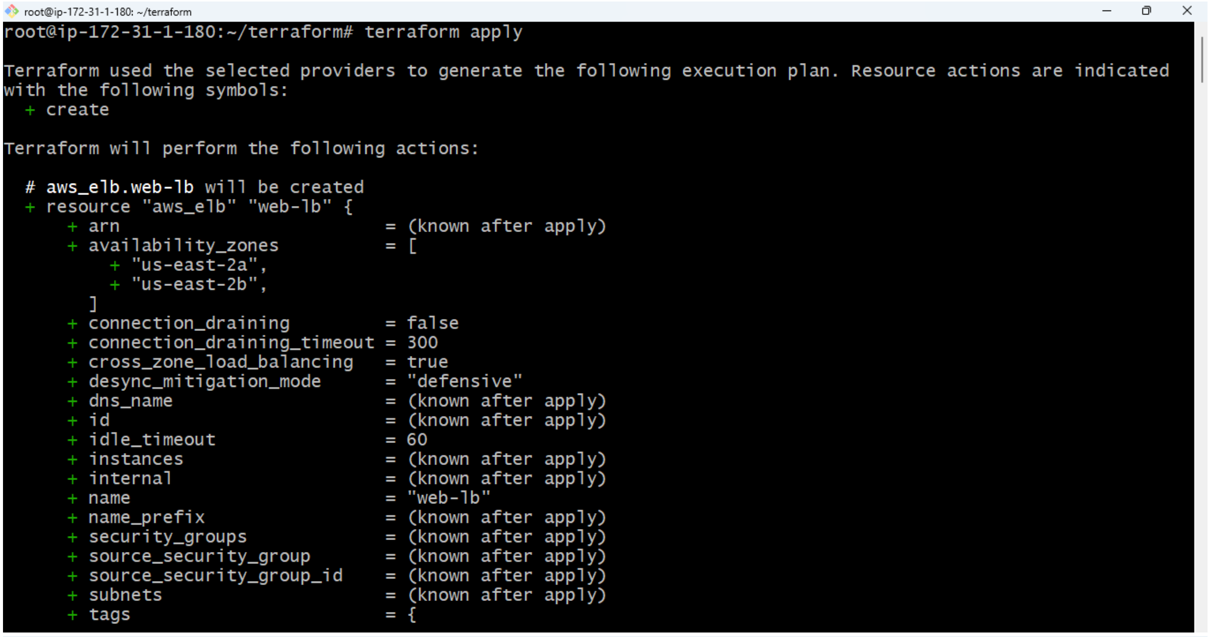
```
root@ip-172-31-1-180:~/terraform# terraform init
Initializing the backend...
Initializing provider plugins...
- Finding hashicorp/aws versions matching "5.64.0"...
- Installing hashicorp/aws v5.64.0...
- Installed hashicorp/aws v5.64.0 (signed by HashiCorp)
Terraform has created a lock file .terraform.lock.hcl 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@ip-172-31-1-180:~/terraform# 
```



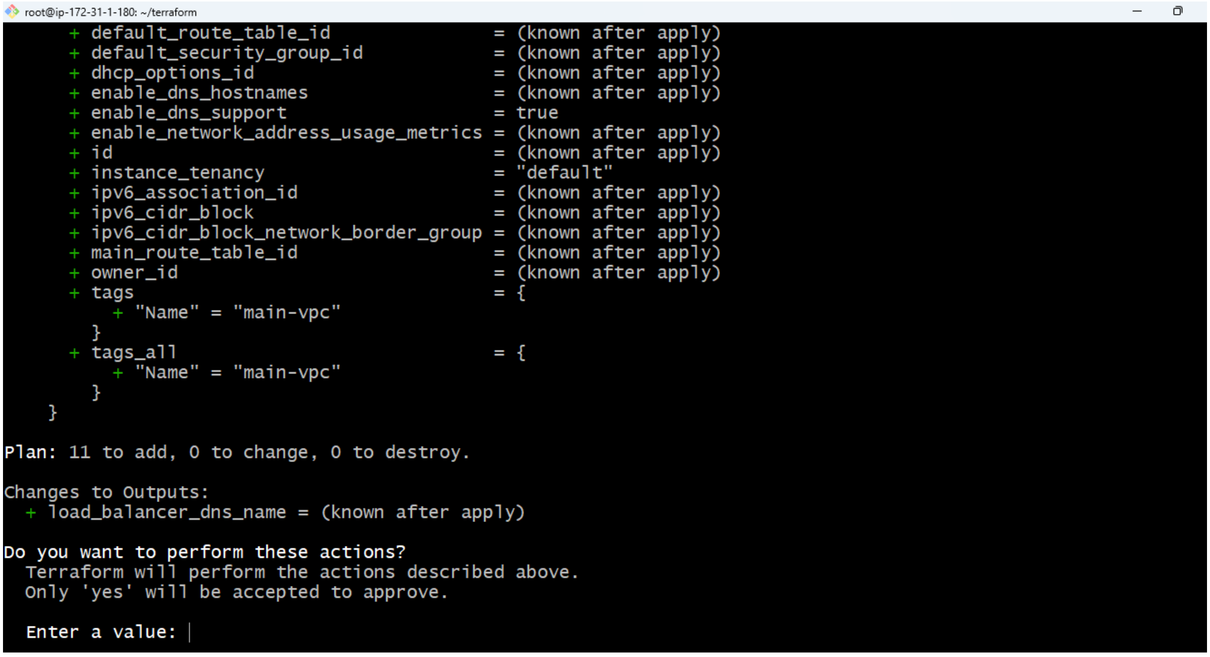
```
root@ip-172-31-1-180:~/terraform# terraform validate
Success! The configuration is valid.
```



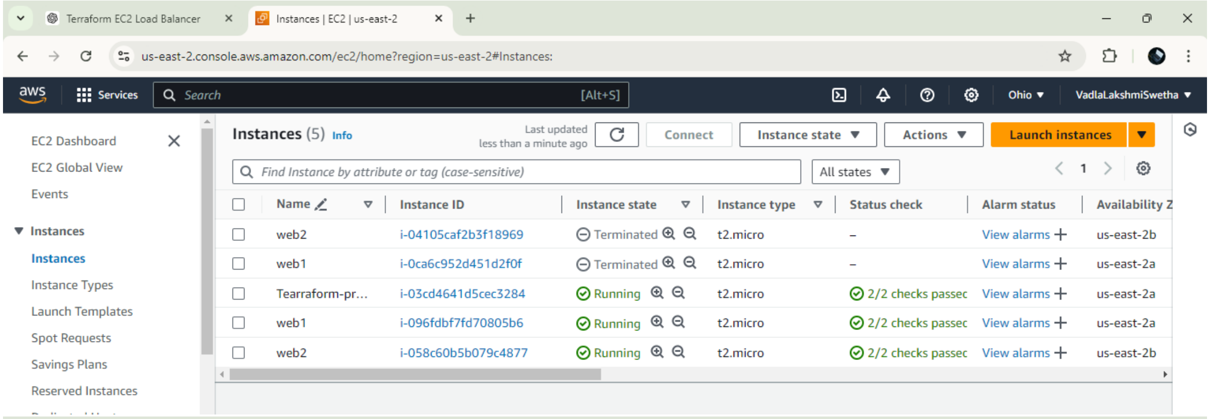
```
root@ip-172-31-1-180:~/terraform# terraform plan

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated
with the following symbols:
  + create

Terraform will perform the following actions:

  # aws_elb.web-lb will be created
  + resource "aws_elb" "web-lb" {
      + arn                         = (known after apply)
      + availability_zones          = [
          + "us-east-2a",
          + "us-east-2b",
        ]
      + connection_draining         = false
      + connection_draining_timeout = 300
      + cross_zone_load_balancing   = true
      + desync_mitigation_mode      = "defensive"
      + dns_name                    = (known after apply)
      + id                          = (known after apply)
      + idle_timeout                = 60
      + instances                   = (known after apply)
      + internal                    = (known after apply)
      + name                        = "web-lb"
      + name_prefix                 = (known after apply)
      + security_groups             = (known after apply)
      + source_security_group       = (known after apply)
      + source_security_group_id    = (known after apply)
      + subnets                     = (known after apply)
      + tags                        = {
```



```
root@ip-172-31-1-180:~/terraform# terraform apply

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated
with the following symbols:
  + create

Terraform will perform the following actions:

  # aws_elb.web-lb will be created
  + resource "aws_elb" "web-lb" {
      + arn                         = (known after apply)
      + availability_zones          = [
          + "us-east-2a",
          + "us-east-2b",
        ]
      + connection_draining         = false
      + connection_draining_timeout = 300
      + cross_zone_load_balancing   = true
      + desync_mitigation_mode      = "defensive"
      + dns_name                    = (known after apply)
      + id                          = (known after apply)
      + idle_timeout                = 60
      + instances                   = (known after apply)
      + internal                    = (known after apply)
      + name                        = "web-lb"
      + name_prefix                 = (known after apply)
      + security_groups             = (known after apply)
      + source_security_group       = (known after apply)
      + source_security_group_id    = (known after apply)
      + subnets                     = (known after apply)
      + tags                        = {
```

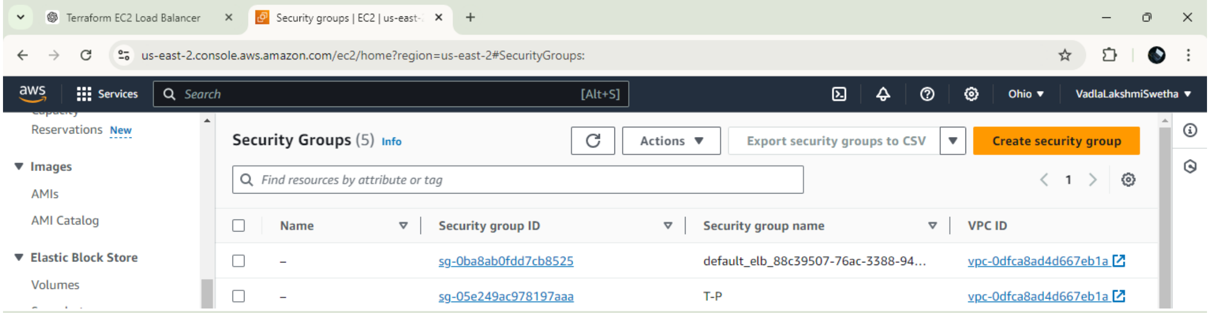


```
      + default_route_table_id               = (known after apply)
      + default_security_group_id            = (known after apply)
      + dhcp_options_id                      = (known after apply)
      + enable_dns_hostnames                 = (known after apply)
      + enable_dns_support                   = true
      + enable_network_address_usage_metrics = (known after apply)
      + id                                   = (known after apply)
      + instance_tenancy                     = "default"
      + ipv6_association_id                  = (known after apply)
      + ipv6_cidr_block                      = (known after apply)
      + ipv6_cidr_block_network_border_group = (known after apply)
      + main_route_table_id                  = (known after apply)
      + owner_id                             = (known after apply)
      + tags                                 = {
          + "Name" = "main-vpc"
        }
      + tags_all                             = {
          + "Name" = "main-vpc"
        }
    }

Plan: 11 to add, 0 to change, 0 to destroy.

Changes to Outputs:
  + load_balancer_dns_name = (known after apply)

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

  Enter a value:
```



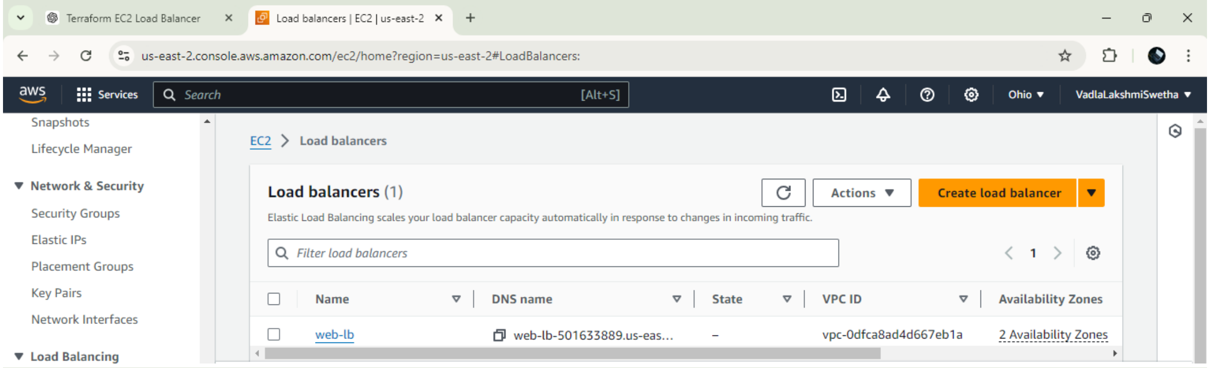
Instances (5) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Z
web2	i-04105caf2b3f18969	Terminated	t2.micro	–	View alarms	us-east-2b
web1	i-0ca6c952d451d2f0f	Terminated	t2.micro	–	View alarms	us-east-2a
Tearraform-pr...	i-03cd4641d5cec3284	Running	t2.micro	2/2 checks passec	View alarms	us-east-2a
web1	i-096fdbf7fd70805b6	Running	t2.micro	2/2 checks passec	View alarms	us-east-2a
web2	i-058c60b5b079c4877	Running	t2.micro	2/2 checks passec	View alarms	us-east-2b



Security Groups (5) Info

Name	Security group ID	Security group name	VPC ID
–	sg-0ba8ab0fdd7cb8525	default_elb_88c39507-76ac-3388-94...	vpc-0dfca8ad4d667eb1a
–	sg-05e249ac978197aaa	T-P	vpc-0dfca8ad4d667eb1a

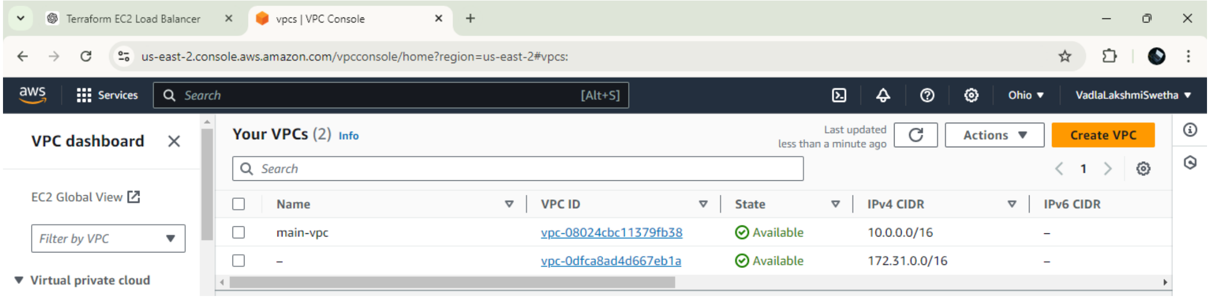


EC2 > Load balancers

Load balancers (1)

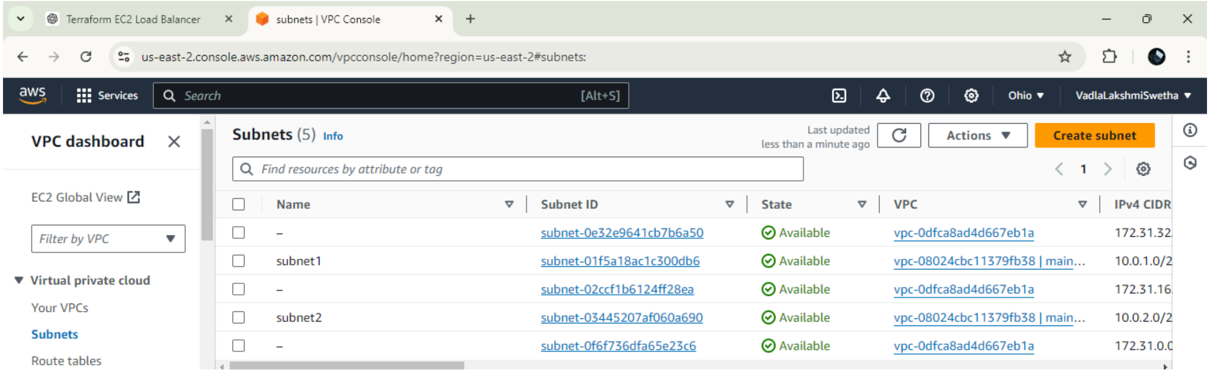
Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.

Name	DNS name	State	VPC ID	Availability Zones
web-lb	web-lb-501633889.us-eas...	–	vpc-0dfca8ad4d667eb1a	2 Availability Zones



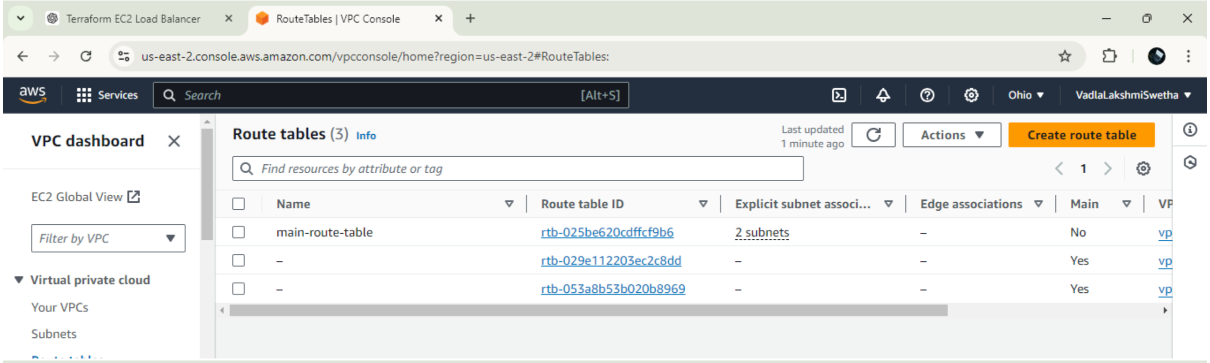
Your VPCs (2) Info

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR
main-vpc	vpc-08024cbc11379fb38	Available	10.0.0.0/16	–
–	vpc-0dfca8ad4d667eb1a	Available	172.31.0.0/16	–



Subnets (5) Info

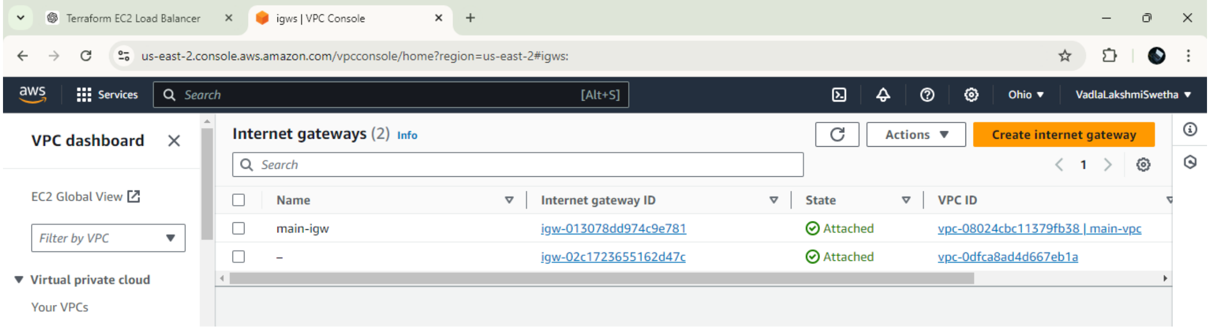
Name	Subnet ID	State	VPC	IPv4 CIDR
–	subnet-0e32e9641cb7b6a50	Available	vpc-0dfca8ad4d667eb1a	172.31.32
subnet1	subnet-01f5a18ac1c300db6	Available	vpc-08024cbc11379fb38 | main...	10.0.1.0/2
–	subnet-02ccf1b6124ff28ea	Available	vpc-0dfca8ad4d667eb1a	172.31.16
subnet2	subnet-03445207af060a690	Available	vpc-08024cbc11379fb38 | main...	10.0.2.0/2
–	subnet-0f6f736dfa65e23c6	Available	vpc-0dfca8ad4d667eb1a	172.31.0.0



Terraform EC2 Load Balancer
RouteTables | VPC Console
us-east-2.console.aws.amazon.com/vpcconsole/home?region=us-east-2#RouteTables:
Services
Search
[Alt+S]
Ohio
VadlaLakshmiSwetha
VPC dashboard
EC2 Global View
Filter by VPC
Virtual private cloud
Your VPCs
Subnets
Last updated 1 minute ago
Actions
Create route table
Find resources by attribute or tag

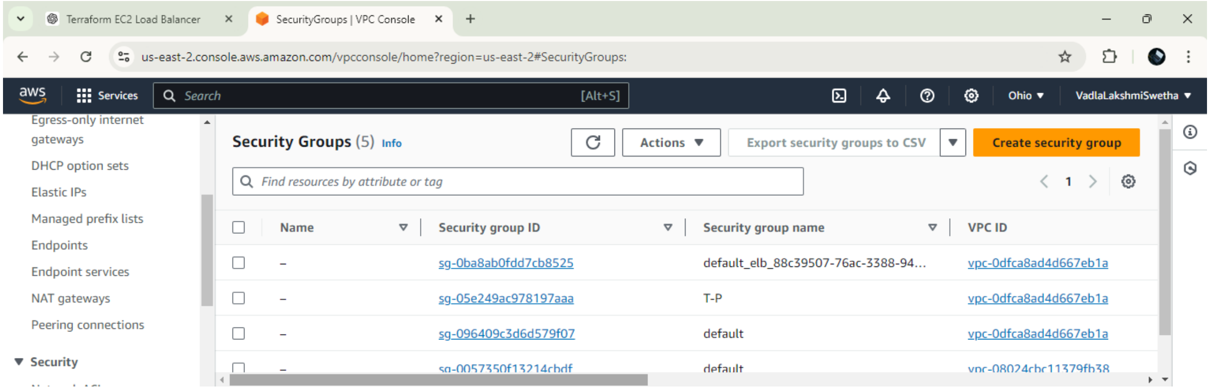
Route tables (3) Info

☐	Name	Route table ID	Explicit subnet associ...	Edge associations	Main	VP
☐	main-route-table	rtb-025be620cdffcf9b6	2 subnets	–	No	vp
☐	–	rtb-029e112203ec2c8dd	–	–	Yes	vp
☐	–	rtb-053a8b53b020b8969	–	–	Yes	vp



Internet gateways (2) Info

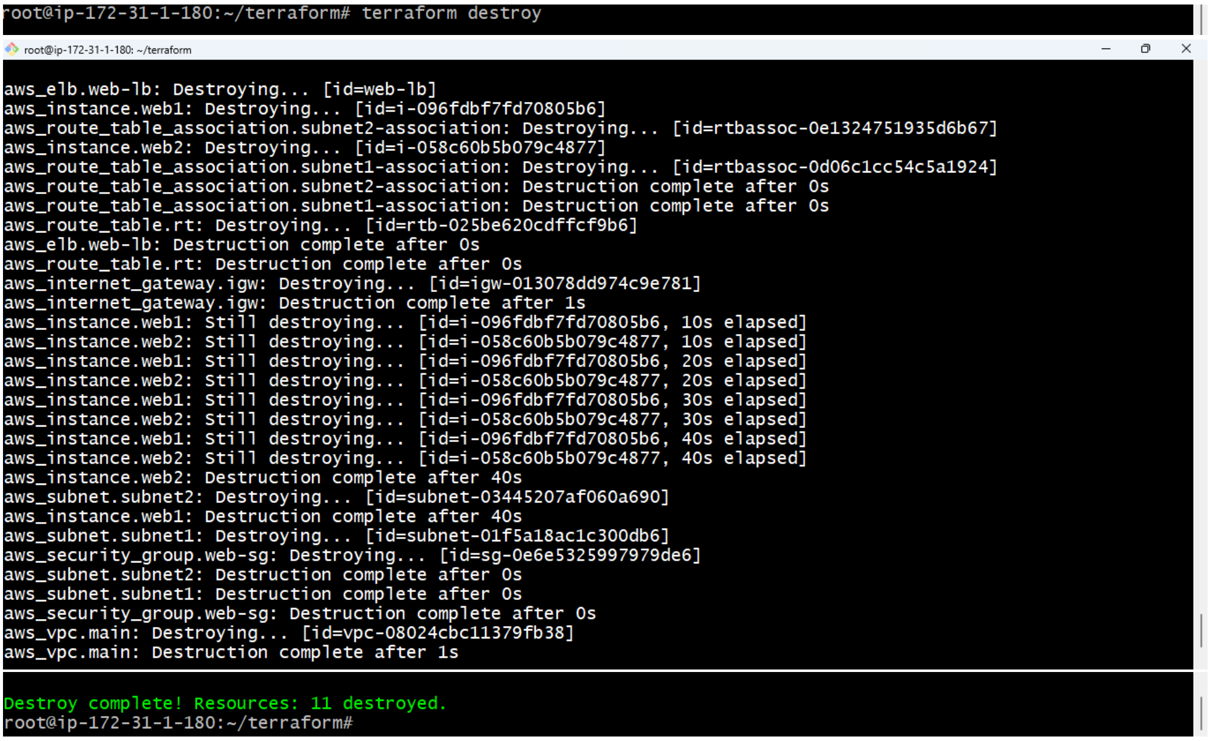
☐	Name	Internet gateway ID	State	VPC ID
☐	main-igw	igw-013078dd974c9e781	✓ Attached	vpc-08024cbc11379fb38 | main-vpc
☐	–	igw-02c1723655162d47c	✓ Attached	vpc-0dfca8ad4d667eb1a



Terraform EC2 Load Balancer
SecurityGroups | VPC Console
us-east-2.console.aws.amazon.com/vpcconsole/home?region=us-east-2#SecurityGroups:
Services
Search
[Alt+S]
Ohio
VadlaLakshmiSwetha
Egress-only internet gateways
DHCP option sets
Elastic IPs
Managed prefix lists
Endpoints
Endpoint services
NAT gateways
Peering connections
Security
Actions
Export security groups to CSV
Create security group
Find resources by attribute or tag

Security Groups (5) Info

☐	Name	Security group ID	Security group name	VPC ID
☐	–	sg-0ba8ab0fdd7cb8525	default_elb_88c39507-76ac-3388-94...	vpc-0dfca8ad4d667eb1a
☐	–	sg-05e249ac978197aaa	T-P	vpc-0dfca8ad4d667eb1a
☐	–	sg-096409c3d6d579f07	default	vpc-0dfca8ad4d667eb1a
☐	–	sg-0057350f13214cbdf	default	vpc-08024cbc11379fb38



```
root@ip-172-31-1-180:~/terraform# terraform destroy
```

```
aws_elb.web-lb: Destroying... [id=web-lb]
aws_instance.web1: Destroying... [id=i-096fdbf7fd70805b6]
aws_route_table_association.subnet2-association: Destroying... [id=rtbassoc-0e1324751935d6b67]
aws_instance.web2: Destroying... [id=i-058c60b5b079c4877]
aws_route_table_association.subnet1-association: Destroying... [id=rtbassoc-0d06c1cc54c5a1924]
aws_route_table_association.subnet2-association: Destruction complete after 0s
aws_route_table_association.subnet1-association: Destruction complete after 0s
aws_route_table.rt: Destroying... [id=rtb-025be620cdffcf9b6]
aws_elb.web-lb: Destruction complete after 0s
aws_route_table.rt: Destruction complete after 0s
aws_internet_gateway.igw: Destroying... [id=igw-013078dd974c9e781]
aws_internet_gateway.igw: Destruction complete after 1s
aws_instance.web1: Still destroying... [id=i-096fdbf7fd70805b6, 10s elapsed]
aws_instance.web2: Still destroying... [id=i-058c60b5b079c4877, 10s elapsed]
aws_instance.web1: Still destroying... [id=i-096fdbf7fd70805b6, 20s elapsed]
aws_instance.web2: Still destroying... [id=i-058c60b5b079c4877, 20s elapsed]
aws_instance.web1: Still destroying... [id=i-096fdbf7fd70805b6, 30s elapsed]
aws_instance.web2: Still destroying... [id=i-058c60b5b079c4877, 30s elapsed]
aws_instance.web1: Still destroying... [id=i-096fdbf7fd70805b6, 40s elapsed]
aws_instance.web2: Still destroying... [id=i-058c60b5b079c4877, 40s elapsed]
aws_instance.web2: Destruction complete after 40s
aws_subnet.subnet2: Destroying... [id=subnet-03445207af060a690]
aws_instance.web1: Destruction complete after 40s
aws_subnet.subnet1: Destroying... [id=subnet-01f5a18ac1c300db6]
aws_security_group.web-sg: Destroying... [id=sg-0e6e5325997979de6]
aws_subnet.subnet2: Destruction complete after 0s
aws_subnet.subnet1: Destruction complete after 0s
aws_security_group.web-sg: Destruction complete after 0s
aws_vpc.main: Destroying... [id=vpc-08024cbc11379fb38]
aws_vpc.main: Destruction complete after 1s

Destroy complete! Resources: 11 destroyed.
root@ip-172-31-1-180:~/terraform#
```