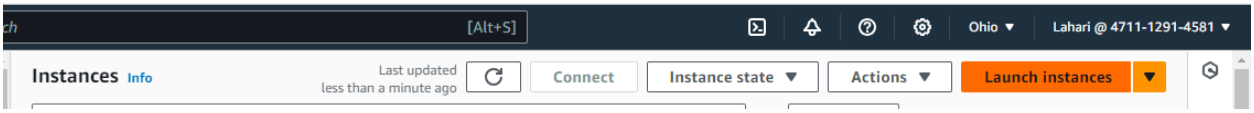
TERRAFORM

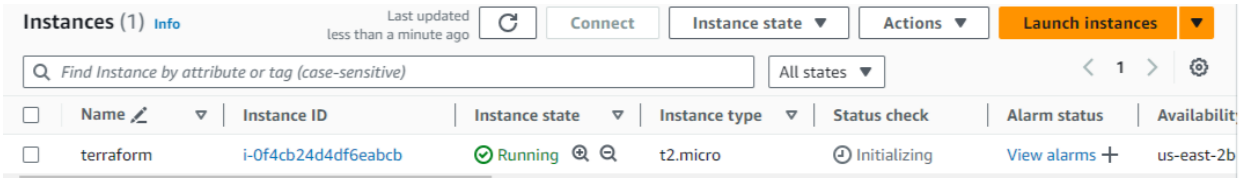
Create below infra using terraform

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

2. Configure the load balancer for the above servers

First launch a instance and connect the instance





To become root user

```
ubuntu@ip-172-31-28-100:~$ sudo -i
```

To install terraform we need to install

- Install AWS CLI
- Install Terraform CLI

```
root@ip-172-31-28-100:~# curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
unzip awscliv2.zip
sudo ./aws/install
```

To know the version of AWS

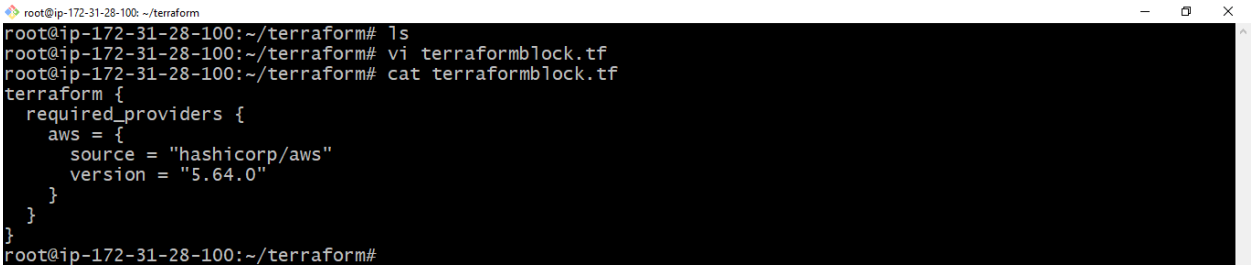
```
root@ip-172-31-28-100:~# aws --version
aws-cli/2.17.37 Python/3.11.9 Linux/6.8.0-1012-aws exe/x86_64.ubuntu.24
root@ip-172-31-28-100:~#
```

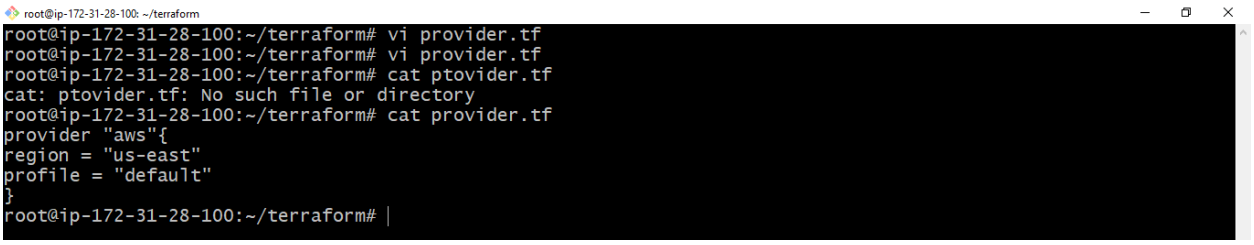
Install terraform CLI

```
root@ip-172-31-28-100:~# wget -O- https://apt.releases.hashicorp.com/gpg | sudo gpg --dearmor -o /usr/share/keyr
ings/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 update && sudo apt install terraform
```

Version of terraform we install by and configure the AWS by using IAM

Login to terraform flatform





Type the code for resource.tf

```
resource "aws_security_group" "example_sg" {
  name        = "example_sg"
  description = "Allow inbound traffic on port 22 (SSH) and port 80 (HTTP)"

  ingress {
    from_port   = 22
    to_port     = 22
    protocol    = "tcp"
```

```
    cidr_blocks = ["0.0.0.0/0"]
  }

  ingress {
    from_port   = 80
    to_port     = 80
    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"]
  }
}

resource "aws_instance" "web" {
  count         = 2
  ami           = "ami-0c55b159cbfafe1f0"  # Replace with a valid AMI ID for your region
  instance_type = "t2.micro"
  security_groups = [aws_security_group.example_sg.name]

  tags = {
    Name = "web-instance-${count.index + 1}"
  }
}
```

```
resource "aws_elb" "example_lb" {
  name               = "example-lb"
  availability_zones = data.aws_availability_zones.available.names

  listener {
    instance_port     = 80
    instance_protocol = "HTTP"
    protocol          = "HTTP"
    port              = 80
  }

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

  security_groups = [aws_security_group.example_sg.id]
  instances       = aws_instance.web[*].id

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

```
data "aws_availability_zones" "available" {}


output "instance_ids" {
  value = aws_instance.web[*].id
}


output "load_balancer_dns" {
  value = aws_elb.example_lb.dns_name
}
}
```

Apply the commands to terraform

- Terraform init
- Terraform validate
- Terraform plan
- Terraform apply
- Terraform destroy

```
root@ip-172-31-28-100:~/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!
```

Initialized Successfully

```
root@ip-172-31-28-100:~/terraform# terraform validate
Success! The configuration is valid.

root@ip-172-31-28-100:~/terraform#
```

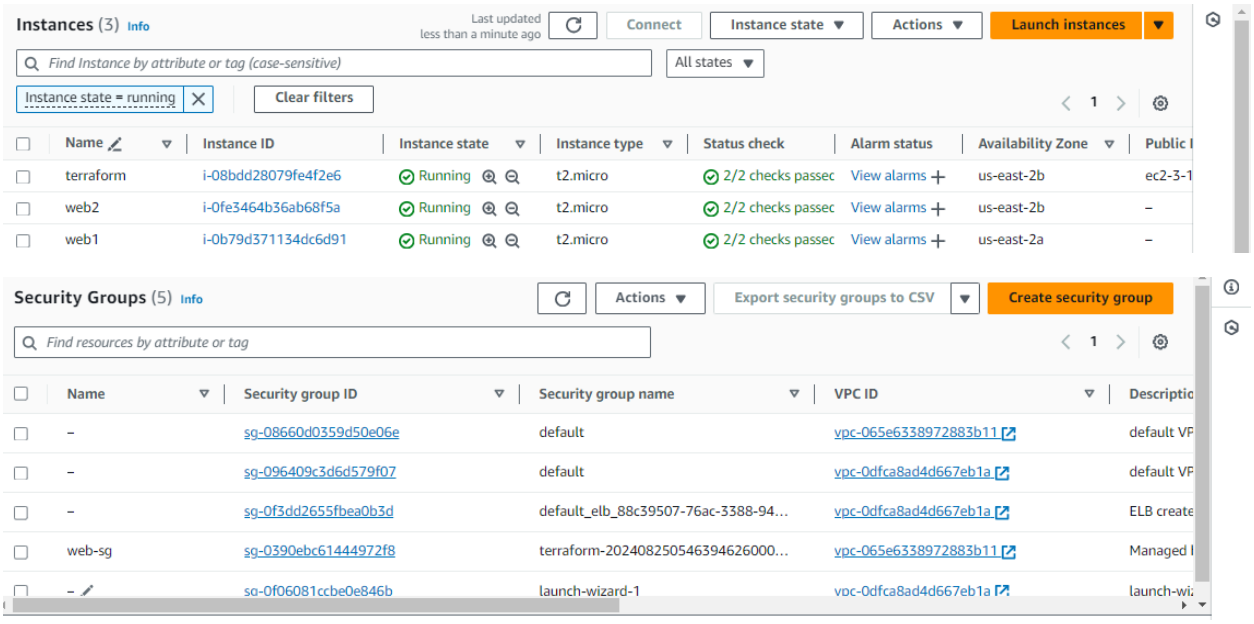
root@ip-172-31-28-100: ~/terraform

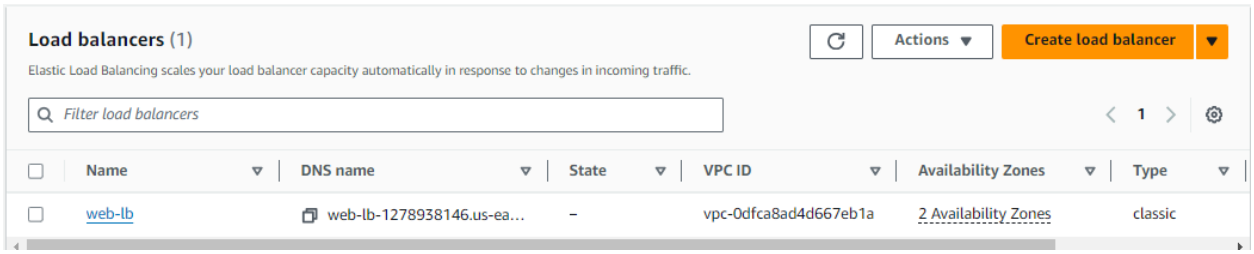
```
  # aws_vpc.main will be created
  + resource "aws_vpc" "main" {
      + arn                                  = (known after apply)
      + cidr_block                           = "10.0.0.0/16"
      + default_network_acl_id               = (known after apply)
      + default_route_table_id               = (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)
```

Finally, 11 to add two instances, one security group, one load balancer, one Vpc, two subnets, one route table, one internet gateway, and defaults





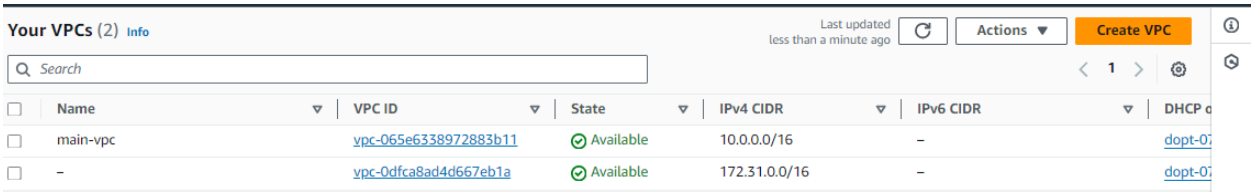
Load balancers (1)

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

Actions | Create load balancer

Filter load balancers

Name	DNS name	State	VPC ID	Availability Zones	Type
web-lb	web-lb-1278938146.us-ea…	–	vpc-0dfca8ad4d667eb1a	2 Availability Zones	classic

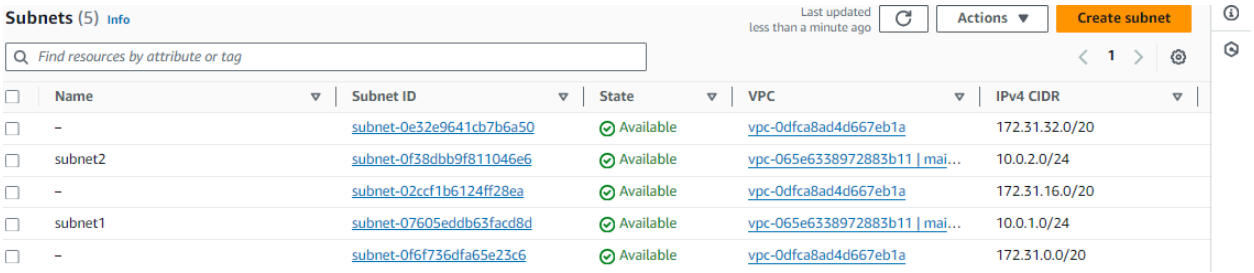


Your VPCs (2) Info

Last updated less than a minute ago | Actions | Create VPC

Search

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP o
main-vpc	vpc-065e6338972883b11	Available	10.0.0.0/16	–	dopt-07
–	vpc-0dfca8ad4d667eb1a	Available	172.31.0.0/16	–	dopt-07

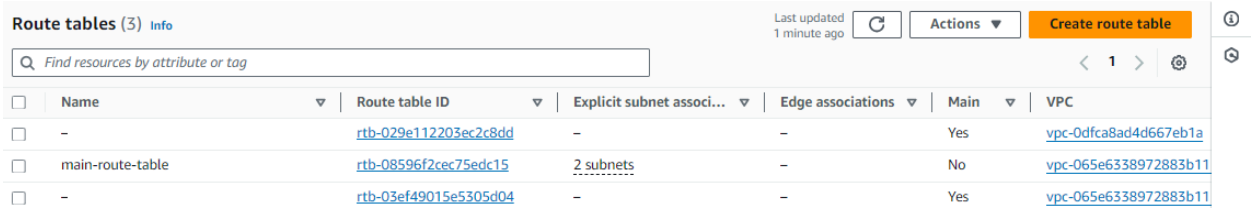


Subnets (5) Info

Last updated less than a minute ago | Actions | Create subnet

Find resources by attribute or tag

Name	Subnet ID	State	VPC	IPv4 CIDR
–	subnet-0e32e9641cb7b6a50	Available	vpc-0dfca8ad4d667eb1a	172.31.32.0/20
subnet2	subnet-0f38dbb9f811046e6	Available	vpc-065e6338972883b11 | mai…	10.0.2.0/24
–	subnet-02ccf1b6124ff28ea	Available	vpc-0dfca8ad4d667eb1a	172.31.16.0/20
subnet1	subnet-07605eddb63facd8d	Available	vpc-065e6338972883b11 | mai…	10.0.1.0/24
–	subnet-0f6f736dfa65e23c6	Available	vpc-0dfca8ad4d667eb1a	172.31.0.0/20

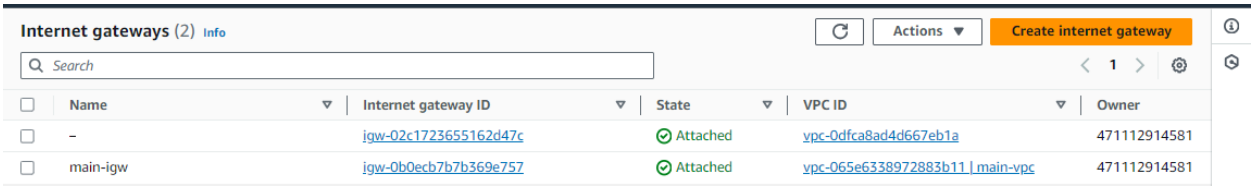


Route tables (3) Info

Last updated 1 minute ago | Actions | Create route table

Find resources by attribute or tag

Name	Route table ID	Explicit subnet associ…	Edge associations	Main	VPC
–	rtb-029e112203ec2c8dd	–	–	Yes	vpc-0dfca8ad4d667eb1a
main-route-table	rtb-08596f2cec75edc15	2 subnets	–	No	vpc-065e6338972883b11
–	rtb-03ef49015e5305d04	–	–	Yes	vpc-065e6338972883b11

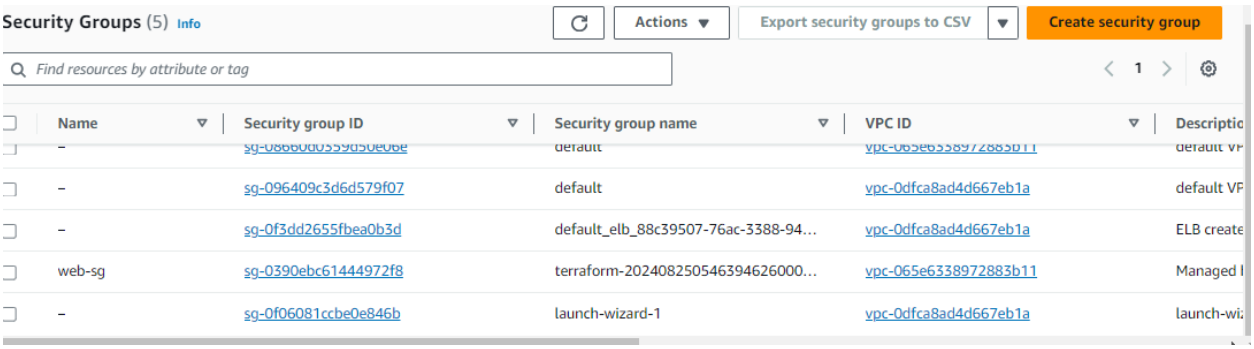


Internet gateways (2) Info

Actions | Create internet gateway

Search

Name	Internet gateway ID	State	VPC ID	Owner
–	igw-02c1723655162d47c	Attached	vpc-0dfca8ad4d667eb1a	471112914581
main-igw	igw-0b0ecb7b7b369e757	Attached	vpc-065e6338972883b11 | main-vpc	471112914581



Security Groups (5) Info

Actions | Export security groups to CSV | Create security group

Find resources by attribute or tag

Name	Security group ID	Security group name	VPC ID	Descriptio
–	sg-0866d0559d50e66e	default	vpc-065e6338972883b11	default VF
–	sg-096409c3d6d579f07	default	vpc-0dfca8ad4d667eb1a	default VF
–	sg-0f3dd2655fbea0b3d	default_elb_88c39507-76ac-3388-94…	vpc-0dfca8ad4d667eb1a	ELB create
web-sg	sg-0390ebc61444972f8	terraform-20240825054639462600…	vpc-065e6338972883b11	Managed I
–	sg-0f06081ccbe0e846b	launch-wizard-1	vpc-0dfca8ad4d667eb1a	launch-wi

At last, destroy it

```
  # aws_vpc.main will be destroyed
  - resource "aws_vpc" "main" {
      - arn                                  = "arn:aws:ec2:us-east-2:471112914581:vpc/vpc-065e6338972883b11" -> null
      - assign_generated_ipv6_cidr_block     = false -> null
      - cidr_block                           = "10.0.0.0/16" -> null
      - default_network_acl_id               = "acl-0effe124feedd5042" -> null
      - default_route_table_id               = "rtb-03ef49015e5305d04" -> null
      - default_security_group_id            = "sg-08660d0359d50e06e" -> null
      - dhcp_options_id                      = "dopt-07fdb849ebafa3398" -> null
      - enable_dns_hostnames                 = false -> null
      - enable_dns_support                   = true -> null
      - enable_network_address_usage_metrics = false -> null
      - id                                   = "vpc-065e6338972883b11" -> null
      - instance_tenancy                     = "default" -> null
      - ipv6_netmask_length                  = 0 -> null
      - main_route_table_id                  = "rtb-03ef49015e5305d04" -> null
      - owner_id                             = "471112914581" -> null
      - tags                                 = {
          - "Name" = "main-vpc"
        } -> null
      - tags_all                             = {
          - "Name" = "main-vpc"
        } -> null
        # (4 unchanged attributes hidden)
    }

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

```
)

Destroy complete! Resources: 11 destroyed.
root@ip-172-31-28-100:~/terraform#
```