Name : Harish Aeyya

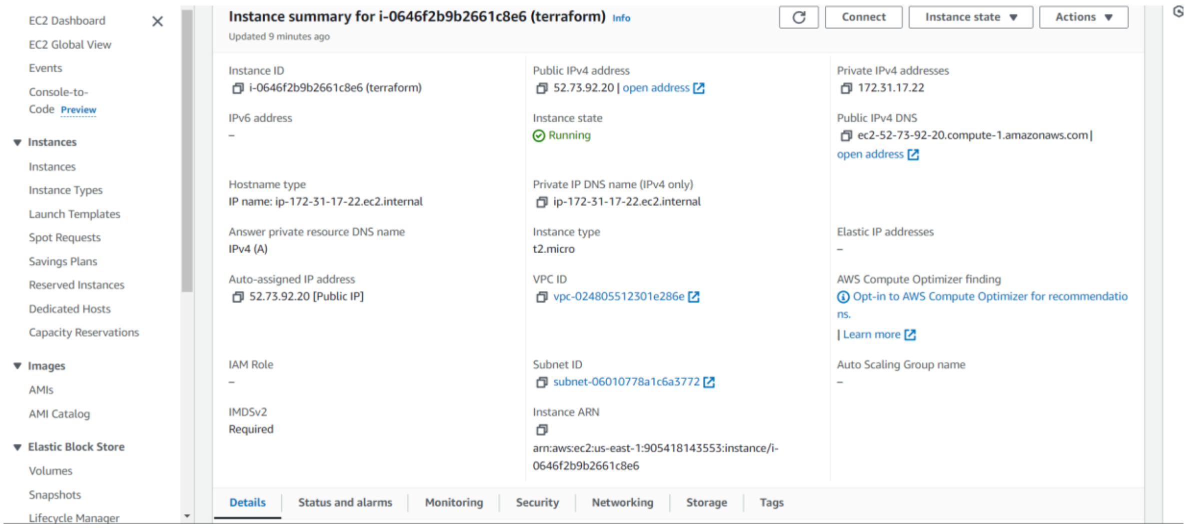
Email : harishaeyya@gmail.com

Mobile : 8184891562

Create two virtual machines in east us (webservers) and configure load balancer for above servers using terraform

Terraform: Terraform is an las software tool that provides a consistent command line interface workflow to manage hundreds of cloud services.

- Lets launch an EC2 instance



Instance summary for i-0646f2b9b2661c8e6 (terraform) Info

Updated 9 minutes ago

Instance ID
i-0646f2b9b2661c8e6 (terraform)

IPv6 address
–

Hostname type
IP name: ip-172-31-17-22.ec2.internal

Answer private resource DNS name
IPv4 (A)

Auto-assigned IP address
52.73.92.20 [Public IP]

IAM Role
–

IMDSv2
Required

Public IPv4 address
52.73.92.20 | open address

Instance state
Running

Private IP DNS name (IPv4 only)
ip-172-31-17-22.ec2.internal

Instance type
t2.micro

VPC ID
vpc-024805512301e286e

Subnet ID
subnet-06010778a1c6a3772

Instance ARN
arn:aws:ec2:us-east-1:905418143553:instance/i-0646f2b9b2661c8e6

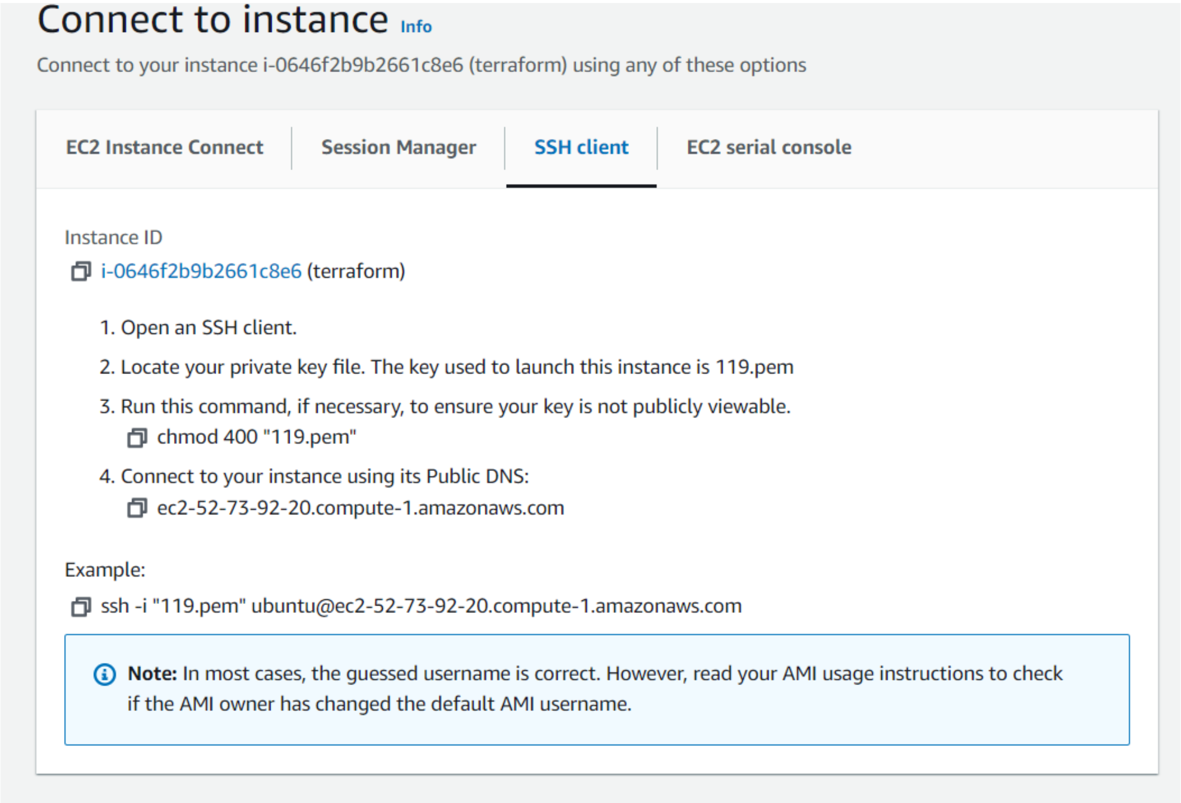
Private IPv4 addresses
172.31.17.22

Public IPv4 DNS
ec2-52-73-92-20.compute-1.amazonaws.com | open address

Elastic IP addresses
–

AWS Compute Optimizer finding
Opt-in to AWS Compute Optimizer for recommendations.
| Learn more

Auto Scaling Group name
–



Connect to instance Info

Connect to your instance i-0646f2b9b2661c8e6 (terraform) using any of these options

Instance ID

i-0646f2b9b2661c8e6 (terraform)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is 119.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
   ```
   chmod 400 "119.pem"
   ```
4. Connect to your instance using its Public DNS:
   ```
   ec2-52-73-92-20.compute-1.amazonaws.com
   ```

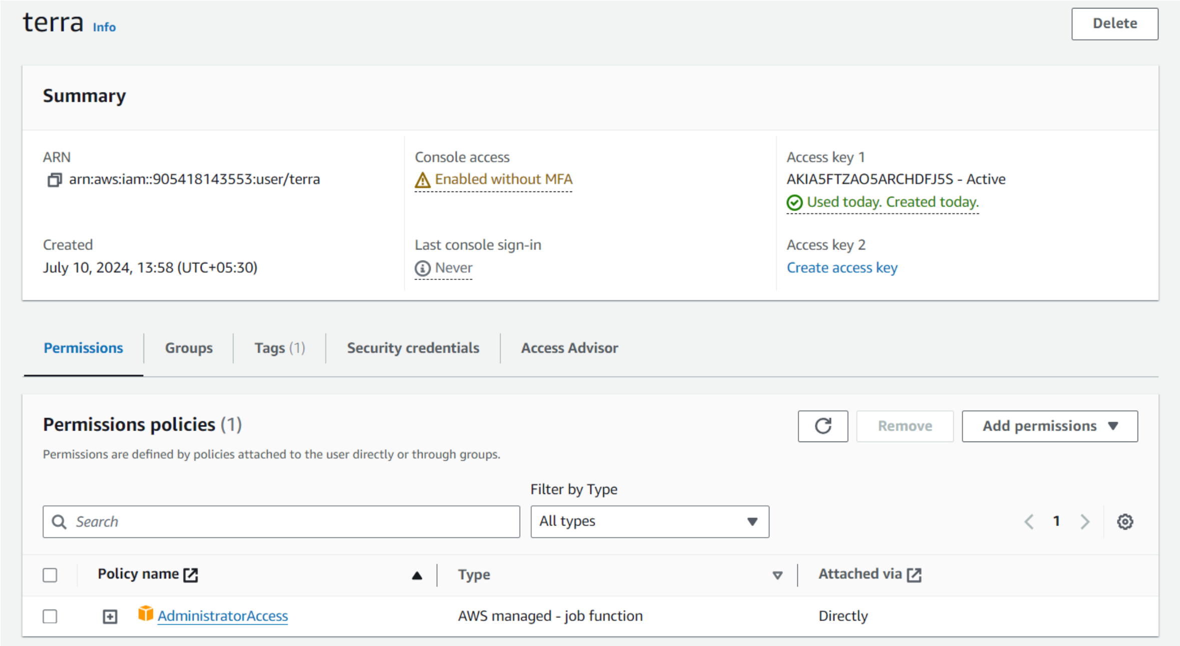
Example:

```
ssh -i "119.pem" ubuntu@ec2-52-73-92-20.compute-1.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.

process:

- create a iam user with administration access.
- now create access key



terra Info

Summary

ARN
arn:aws:iam::905418143553:user/terra

Console access
⚠ Enabled without MFA

Access key 1
AKIA5FTZAO5ARCHDFJ5S - Active
✓ Used today. Created today.

Created
July 10, 2024, 13:58 (UTC+05:30)

Last console sign-in
Never

Access key 2
Create access key

Permissions policies (1)

Permissions are defined by policies attached to the user directly or through groups.

Filter by Type: All types

Policy name	Type	Attached via
AdministratorAccess	AWS managed - job function	Directly

apt install unzip -y

aws cli install on ubuntu

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

unzip awscliv2.zip

sudo ./aws/install

configure your access key id and security key to the resoures

access key:

secret access key:

install the terraform hasicorp on ubuntu.

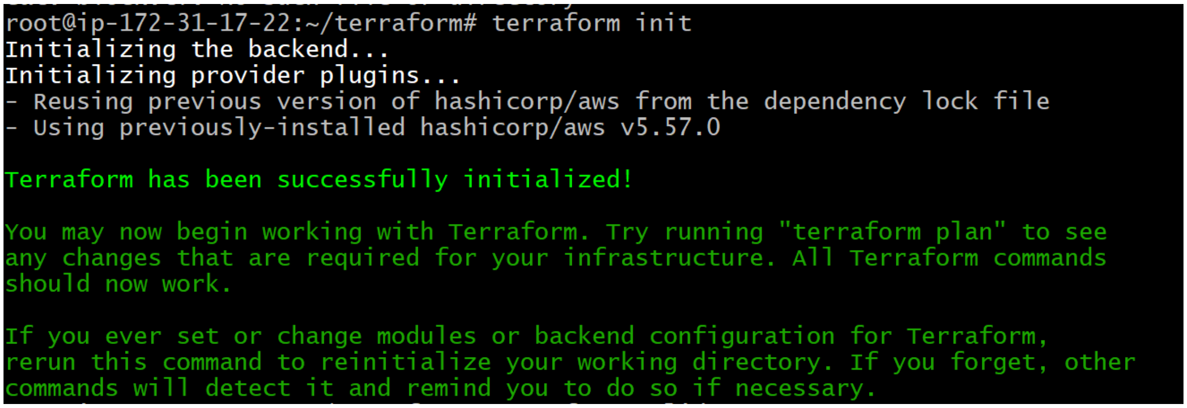
wget -O- https://apt.releases.hashicorp.com/gpg | sudo gpg --dearmor -o /usr/share/keyrings/hashicorp-archive-keyring.gpg

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

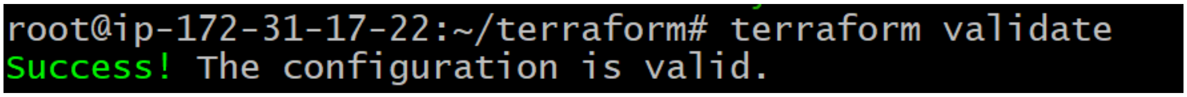
sudo apt update && sudo apt install terraform create a directory and changed to root directory

```
mkdir terraform

cd terraform

vi terraformblock.tf

terraform {

  required_providers {

    aws = {

      source = "hashicorp/aws"

      version = "5.42.0"

    }

  }

}

vi provider.tf

provider "aws" {

  region = "ap-southeast-1"

  profile = "default"

}

vi resource.tf

provider "aws" {

  region = "us-east-1"

}


resource "aws_instance" "web" {

  count         = 2
  ami           = "ami-0c55b159cbfafe1f0"  # Amazon Linux 2 AMI (replace with the latest AMI ID)
  instance_type = "t2.micro"

  tags = {
    Name = "WebServer${count.index + 1}"
  }

  user_data = <<-EOF
              #!/bin/bash
              apt update -y
              apt install -y httpd
              systemctl start httpd
              systemctl enable httpd
              echo "Hello, World from $(hostname -f)" > /var/www/html/index.html
              EOF
}

resource "aws_elb" "web_lb" {
  name               = "web-load-balancer"
  availability_zones = ["us-east-1a", "us-east-1b"]

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

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

  instances = aws_instance.web[*].id

  tags = {
    Name = "WebLoadBalancer"
  }
}

output "elb_dns_name" {
  value = aws_elb.web_lb.dns_name
}
```

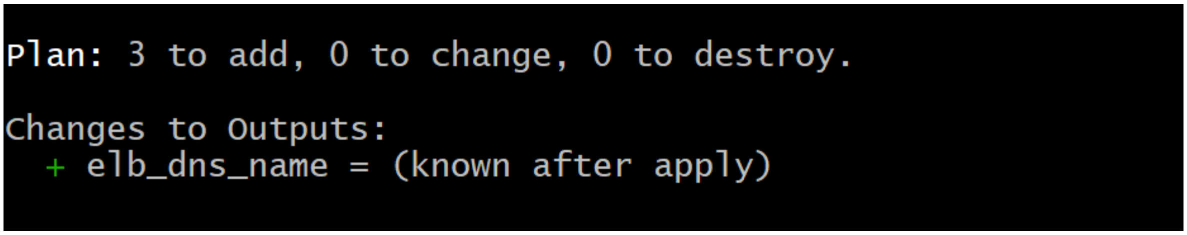
lets following the terraform commands



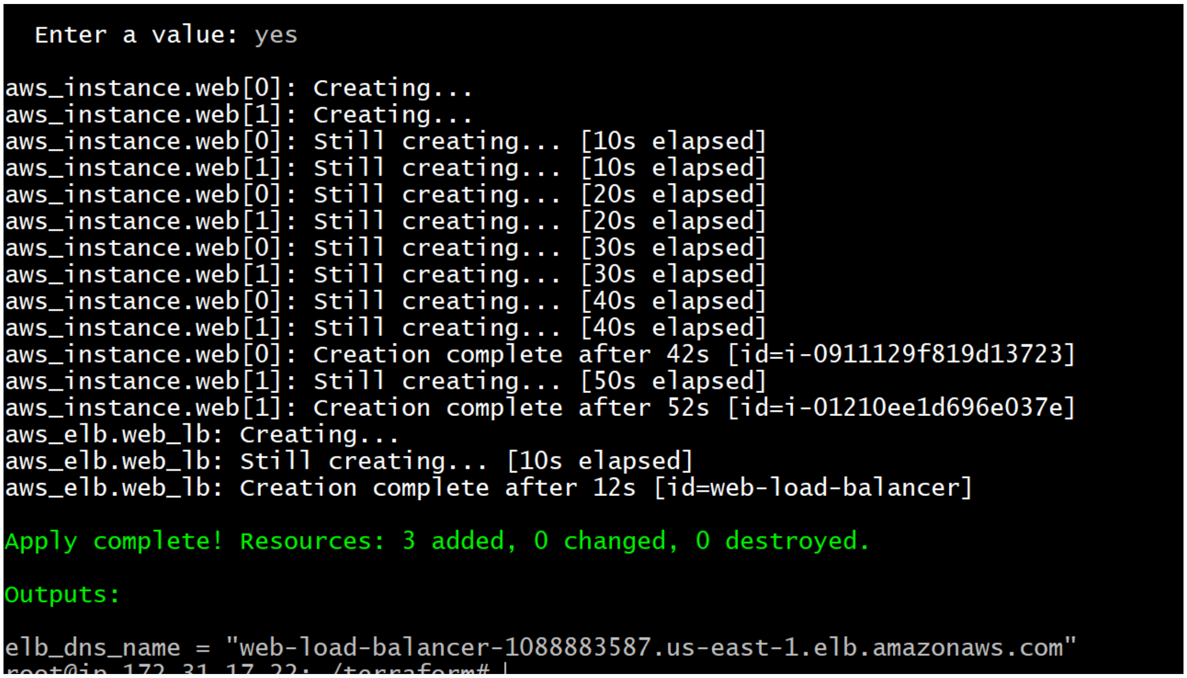
```
root@ip-172-31-17-22:~/terraform# terraform init
Initializing the backend...
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.57.0

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.
```



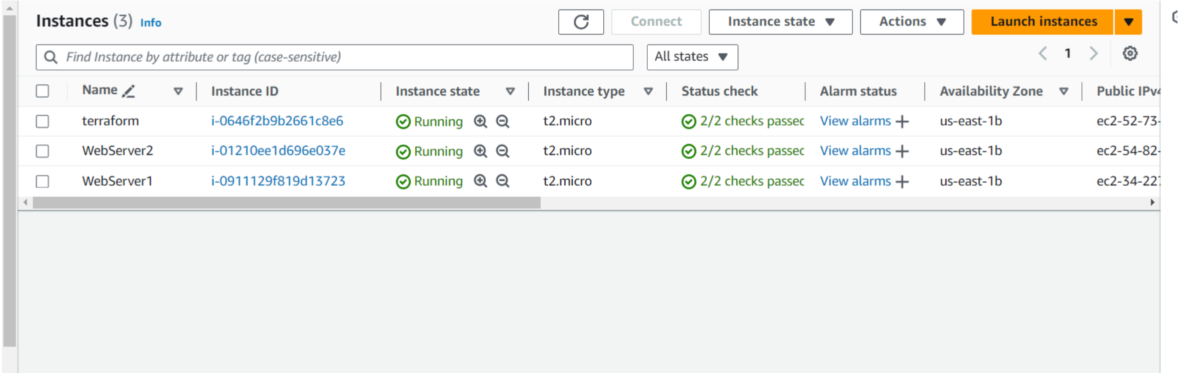
```
root@ip-172-31-17-22:~/terraform# terraform validate
Success! The configuration is valid.
```



```
Plan: 3 to add, 0 to change, 0 to destroy.

Changes to Outputs:
  + elb_dns_name = (known after apply)
```

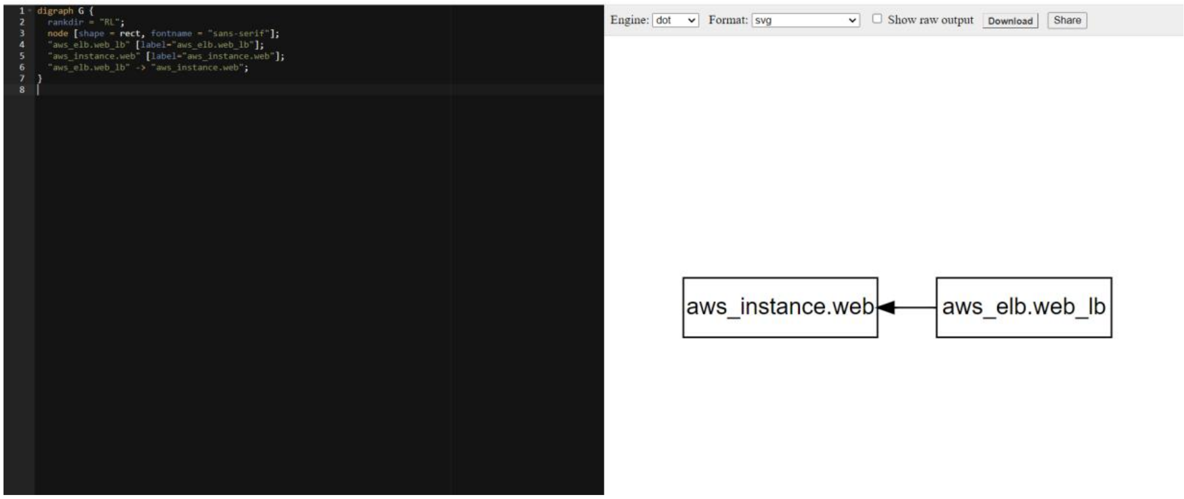


```
  Enter a value: yes

aws_instance.web[0]: Creating...
aws_instance.web[1]: Creating...
aws_instance.web[0]: Still creating... [10s elapsed]
aws_instance.web[1]: Still creating... [10s elapsed]
aws_instance.web[0]: Still creating... [20s elapsed]
aws_instance.web[1]: Still creating... [20s elapsed]
aws_instance.web[0]: Still creating... [30s elapsed]
aws_instance.web[1]: Still creating... [30s elapsed]
aws_instance.web[0]: Still creating... [40s elapsed]
aws_instance.web[1]: Still creating... [40s elapsed]
aws_instance.web[0]: Creation complete after 42s [id=i-0911129f819d13723]
aws_instance.web[1]: Still creating... [50s elapsed]
aws_instance.web[1]: Creation complete after 52s [id=i-01210ee1d696e037e]
aws_elb.web_lb: Creating...
aws_elb.web_lb: Still creating... [10s elapsed]
aws_elb.web_lb: Creation complete after 12s [id=web-load-balancer]

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.

Outputs:

elb_dns_name = "web-load-balancer-1088883587.us-east-1.elb.amazonaws.com"
root@ip-172-31-17-22:~/terraform#
```



□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
□	terraform	i-0646f2b9b2661c8e6	✓ Running	t2.micro	✓ 2/2 checks passec	View alarms +	us-east-1b	ec2-52-73-
□	WebServer2	i-01210ee1d696e037e	✓ Running	t2.micro	✓ 2/2 checks passec	View alarms +	us-east-1b	ec2-54-82-
□	WebServer1	i-0911129f819d13723	✓ Running	t2.micro	✓ 2/2 checks passec	View alarms +	us-east-1b	ec2-34-22

```
root@ip-172-31-17-22:~/terraform# terraform graph

digraph G {

  rankdir = "RL";

  node [shape = rect, fontname = "sans-serif"];

  "aws_elb.web_lb" [label="aws_elb.web_lb"];

  "aws_instance.web" [label="aws_instance.web"];

  "aws_elb.web_lb" -> "aws_instance.web";

}
```



```
digraph G {
  rankdir = "RL";
  node [shape = rect, fontname = "sans-serif"];
  "aws_elb.web_lb" [label="aws_elb.web_lb"];
  "aws_instance.web" [label="aws_instance.web"];
  "aws_elb.web_lb" -> "aws_instance.web";
}
```

Engine: dot Format: svg □ Show raw output