7/16/2025

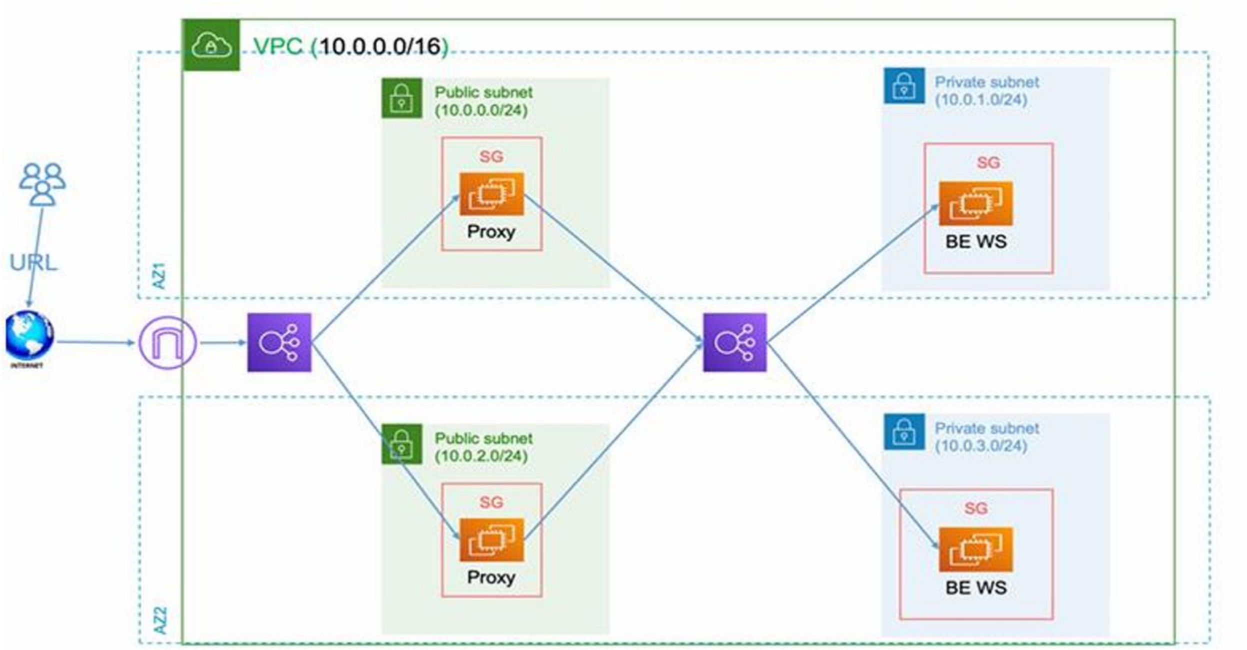
Secure Web App with Public Proxy + Private Backend on AWS

Terraform Project

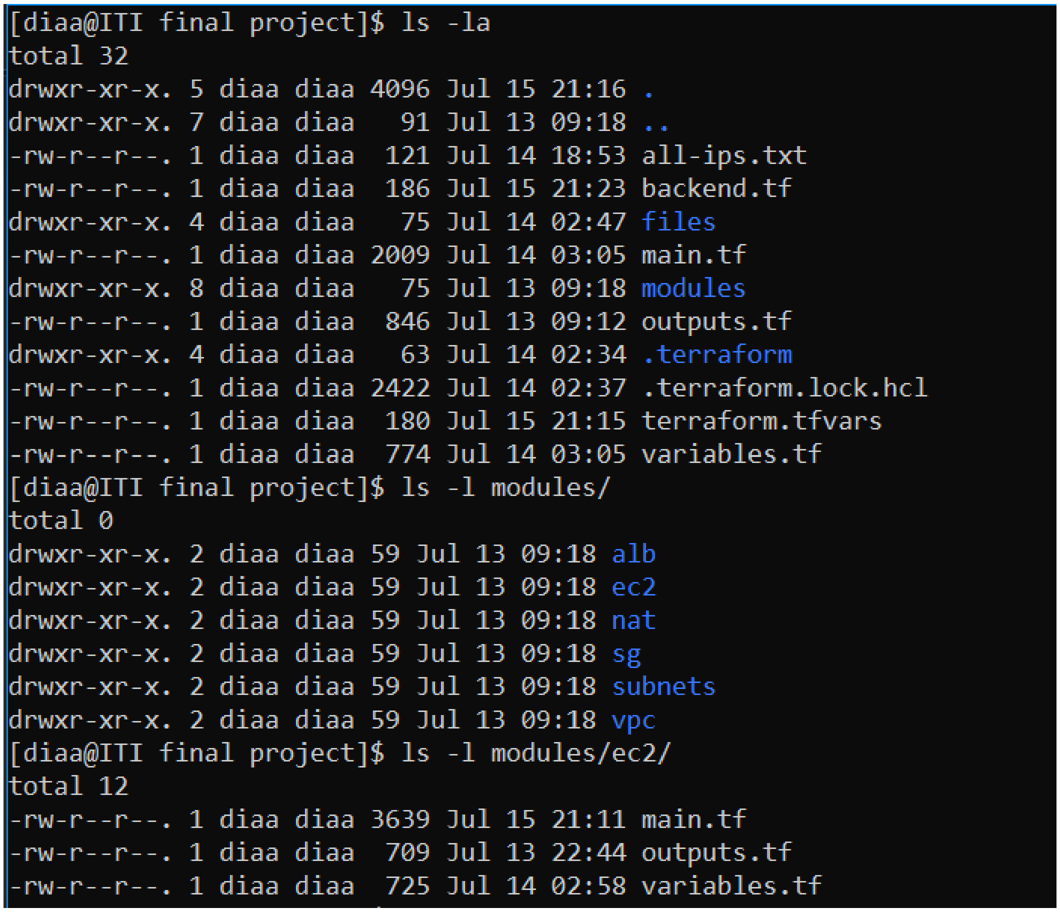
Diaa Qassem
ITI -CAIRO UNIVERSITY

Project Description:

This project consists of a VPC that is deployed in two different availability zones (AZ1 & AZ2). It consists of two public subnets, each containing an EC2 instance that acts as a reverse proxy. They are both connected to a public ALB that takes requests from an internet gateway. The reverse proxy EC2s are connected to two EC2 instances in the two private subnets via an internal ALB.

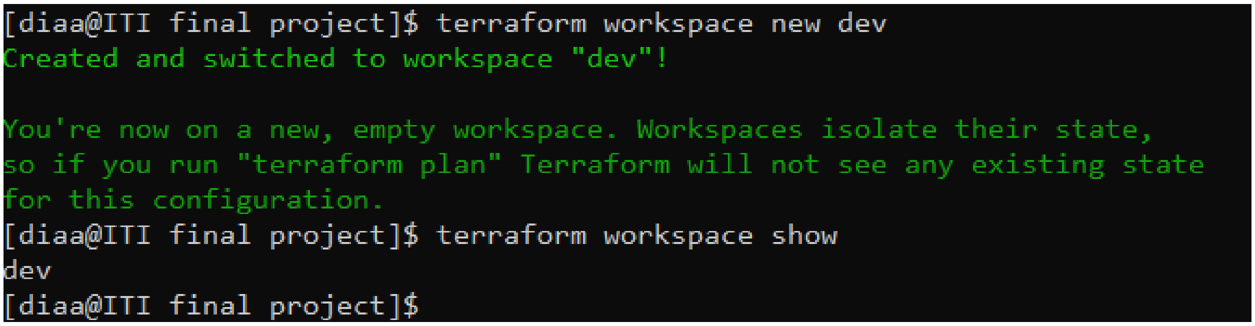

VPC (10.0.0.0/16)
URL
INTERNET
AZ1
AZ2
Public subnet (10.0.0.0/24)
SG
Proxy
Public subnet (10.0.2.0/24)
SG
Proxy
Private subnet (10.0.1.0/24)
SG
BE WS
Private subnet (10.0.3.0/24)
SG
BE WS

Since we are using modules to create this project, the following picture shows the hierarchy of the files that contain the project code.



```
[diaa@ITI final project]$ ls -la
total 32
drwxr-xr-x. 5 diaa diaa 4096 Jul 15 21:16 .
drwxr-xr-x. 7 diaa diaa   91 Jul 13 09:18 ..
-rw-r--r--. 1 diaa diaa  121 Jul 14 18:53 all-ips.txt
-rw-r--r--. 1 diaa diaa  186 Jul 15 21:23 backend.tf
drwxr-xr-x. 4 diaa diaa   75 Jul 14 02:47 files
-rw-r--r--. 1 diaa diaa 2009 Jul 14 03:05 main.tf
drwxr-xr-x. 8 diaa diaa   75 Jul 13 09:18 modules
-rw-r--r--. 1 diaa diaa  846 Jul 13 09:12 outputs.tf
drwxr-xr-x. 4 diaa diaa   63 Jul 14 02:34 .terraform
-rw-r--r--. 1 diaa diaa 2422 Jul 14 02:37 .terraform.lock.hcl
-rw-r--r--. 1 diaa diaa  180 Jul 15 21:15 terraform.tfvars
-rw-r--r--. 1 diaa diaa  774 Jul 14 03:05 variables.tf
[diaa@ITI final project]$ ls -l modules/
total 0
drwxr-xr-x. 2 diaa diaa 59 Jul 13 09:18 alb
drwxr-xr-x. 2 diaa diaa 59 Jul 13 09:18 ec2
drwxr-xr-x. 2 diaa diaa 59 Jul 13 09:18 nat
drwxr-xr-x. 2 diaa diaa 59 Jul 13 09:18 sg
drwxr-xr-x. 2 diaa diaa 59 Jul 13 09:18 subnets
drwxr-xr-x. 2 diaa diaa 59 Jul 13 09:18 vpc
[diaa@ITI final project]$ ls -l modules/ec2/
total 12
-rw-r--r--. 1 diaa diaa 3639 Jul 15 21:11 main.tf
-rw-r--r--. 1 diaa diaa  709 Jul 13 22:44 outputs.tf
-rw-r--r--. 1 diaa diaa  725 Jul 14 02:58 variables.tf
```

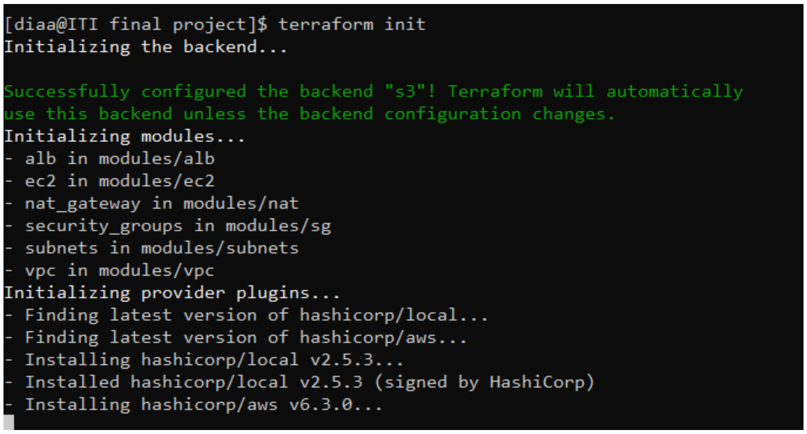
Now, I will create a workspace named “dev”, and verify that it is created successfully.



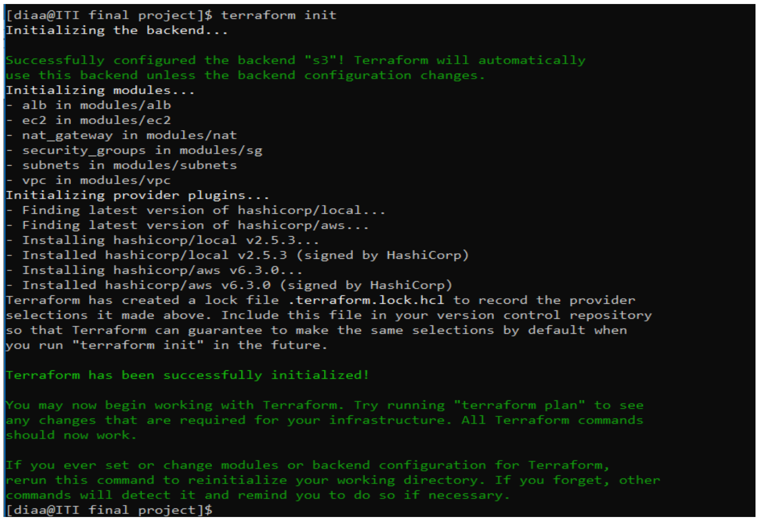
```
[diaa@ITI final project]$ terraform workspace new dev
Created and switched to workspace "dev"!

You're now on a new, empty workspace. Workspaces isolate their state,
so if you run "terraform plan" Terraform will not see any existing state
for this configuration.
[diaa@ITI final project]$ terraform workspace show
dev
[diaa@ITI final project]$
```

I begin with running the “terraform init” command. It initializes the working directory and the backend, setting everything up so Terraform can function properly.

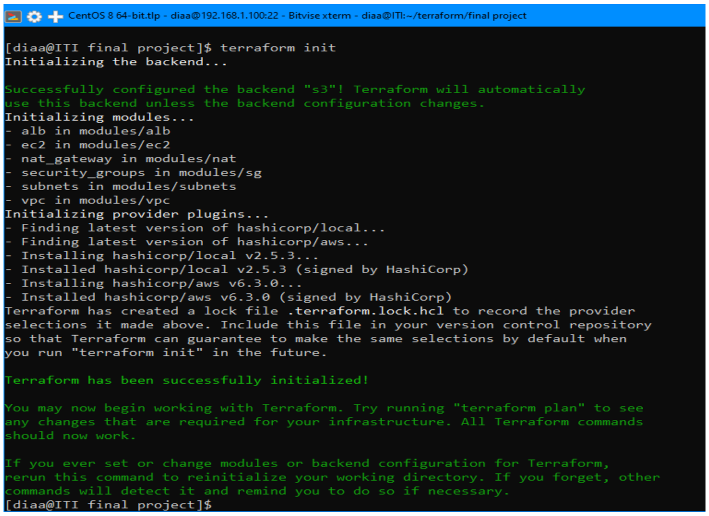
The backend is an S3 bucket and a DynamoDB table that contains a lock ID that prevents two users from making changes on the state file at the same time.



```
[diaa@ITI final project]$ terraform init
Initializing the backend...

Successfully configured the backend "s3"! Terraform will automatically
use this backend unless the backend configuration changes.
Initializing modules...
- alb in modules/alb
- ec2 in modules/ec2
- nat_gateway in modules/nat
- security_groups in modules/sg
- subnets in modules/subnets
- vpc in modules/vpc
Initializing provider plugins...
- Finding latest version of hashicorp/local...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/local v2.5.3...
- Installed hashicorp/local v2.5.3 (signed by HashiCorp)
- Installing hashicorp/aws v6.3.0...
```



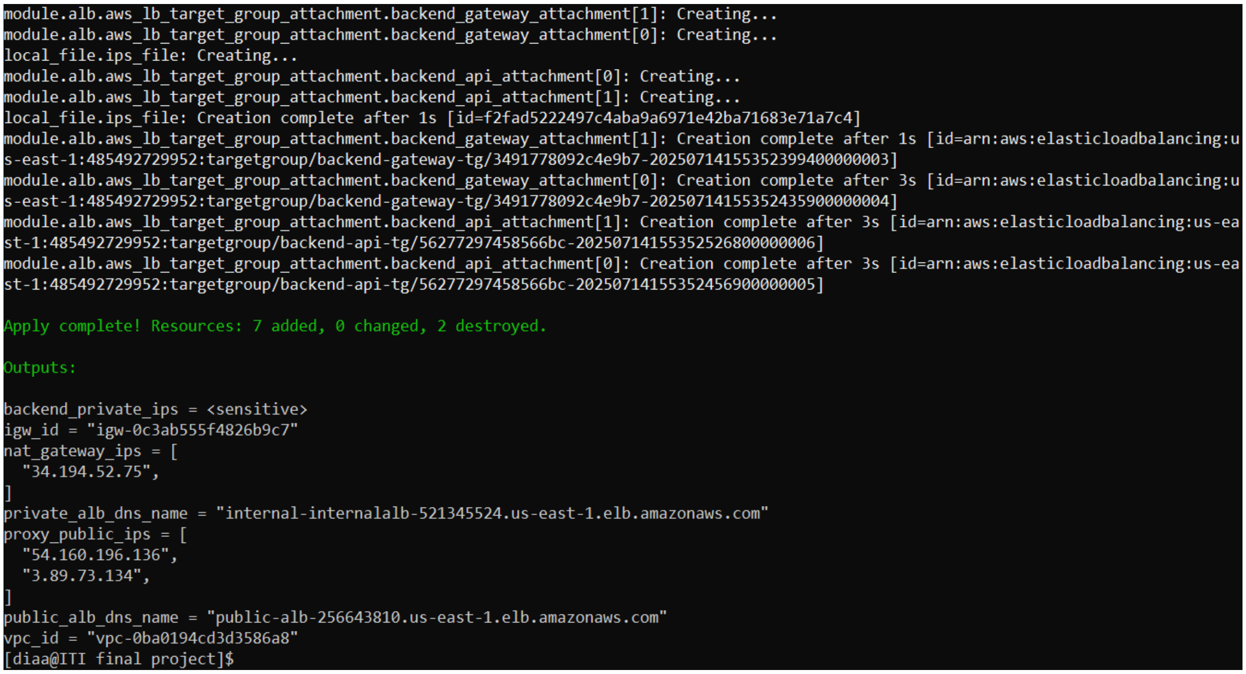
```
[diaa@ITI final project]$ terraform init
Initializing the backend...

Successfully configured the backend "s3"! Terraform will automatically
use this backend unless the backend configuration changes.
Initializing modules...
- alb in modules/alb
- ec2 in modules/ec2
- nat_gateway in modules/nat
- security_groups in modules/sg
- subnets in modules/subnets
- vpc in modules/vpc
Initializing provider plugins...
- Finding latest version of hashicorp/local...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/local v2.5.3...
- Installed hashicorp/local v2.5.3 (signed by HashiCorp)
- Installing hashicorp/aws v6.3.0...
- Installed hashicorp/aws v6.3.0 (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.
[diaa@ITI final project]$
```



CentOS 8 64-bit.tlp - diaa@192.168.1.100:22 - Bitvise xterm - diaa@ITI:~/terraform/final project

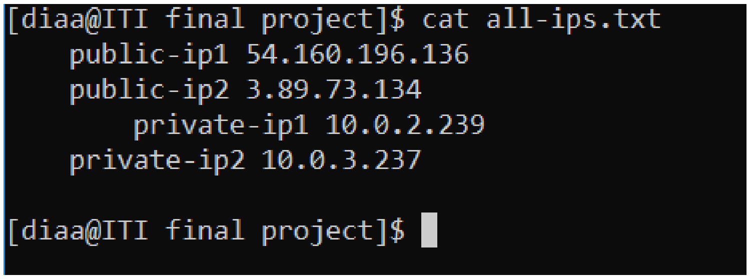
```
[diaa@ITI final project]$ terraform init
Initializing the backend...

Successfully configured the backend "s3"! Terraform will automatically
use this backend unless the backend configuration changes.
Initializing modules...
- alb in modules/alb
- ec2 in modules/ec2
- nat_gateway in modules/nat
- security_groups in modules/sg
- subnets in modules/subnets
- vpc in modules/vpc
Initializing provider plugins...
- Finding latest version of hashicorp/local...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/local v2.5.3...
- Installed hashicorp/local v2.5.3 (signed by HashiCorp)
- Installing hashicorp/aws v6.3.0...
- Installed hashicorp/aws v6.3.0 (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.
[diaa@ITI final project]$
```

After, I'm applying the terraform configurations using the "terraform apply command". Here the number of resources is 7 because I have already applied this file before taking this screenshot.



```
module.alb.aws_lb_target_group_attachment.backend_gateway_attachment[1]: Creating...
module.alb.aws_lb_target_group_attachment.backend_gateway_attachment[0]: Creating...
local_file.ips_file: Creating...
module.alb.aws_lb_target_group_attachment.backend_api_attachment[0]: Creating...
module.alb.aws_lb_target_group_attachment.backend_api_attachment[1]: Creating...
local_file.ips_file: Creation complete after 1s [id=f2fad5222497c4aba9a6971e42ba71683e71a7c4]
module.alb.aws_lb_target_group_attachment.backend_gateway_attachment[1]: Creation complete after 1s [id=arn:aws:elasticloadbalancing:u
s-east-1:485492729952:targetgroup/backend-gateway-tg/3491778092c4e9b7-20250714155352399400000003]
module.alb.aws_lb_target_group_attachment.backend_gateway_attachment[0]: Creation complete after 3s [id=arn:aws:elasticloadbalancing:u
s-east-1:485492729952:targetgroup/backend-gateway-tg/3491778092c4e9b7-20250714155352435900000004]
module.alb.aws_lb_target_group_attachment.backend_api_attachment[1]: Creation complete after 3s [id=arn:aws:elasticloadbalancing:us-ea
st-1:485492729952:targetgroup/backend-api-tg/56277297458566bc-20250714155352526800000006]
module.alb.aws_lb_target_group_attachment.backend_api_attachment[0]: Creation complete after 3s [id=arn:aws:elasticloadbalancing:us-ea
st-1:485492729952:targetgroup/backend-api-tg/56277297458566bc-20250714155352456900000005]

Apply complete! Resources: 7 added, 0 changed, 2 destroyed.

Outputs:

backend_private_ips = <sensitive>
igw_id = "igw-0c3ab555f4826b9c7"
nat_gateway_ips = [
  "34.194.52.75",
]
private_alb_dns_name = "internal-internalalb-521345524.us-east-1.elb.amazonaws.com"
proxy_public_ips = [
  "54.160.196.136",
  "3.89.73.134",
]
public_alb_dns_name = "public-alb-256643810.us-east-1.elb.amazonaws.com"
vpc_id = "vpc-0ba0194cd3d3586a8"
[diaa@ITI final project]$
```

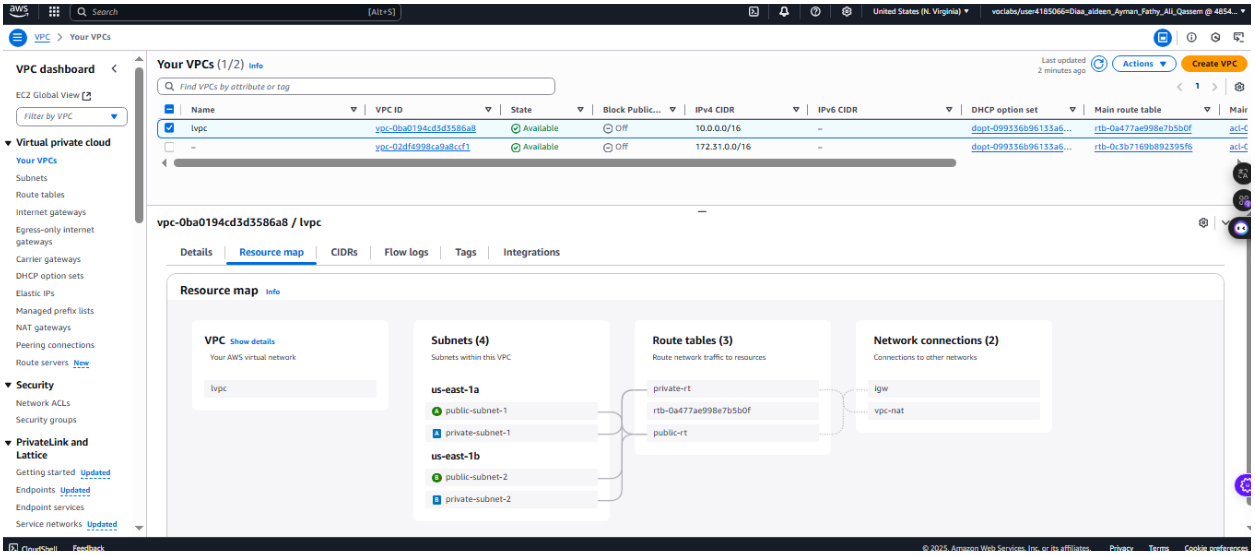
this output file contains the IP associations.



```
[diaa@ITI final project]$ cat all-ips.txt
    public-ip1 54.160.196.136
    public-ip2 3.89.73.134
        private-ip1 10.0.2.239
    private-ip2 10.0.3.237

[diaa@ITI final project]$ 
```

The following picture shows the resources created by Terraform on the console.

1- Resource Map For VPC



Your VPCs (1/2) Info

Last updated 2 minutes ago

☑	Name	VPC ID	State	Block Public...	IPv4 CIDR	IPv6 CIDR	DHCP option set	Main route table	Mair
☑	lvpc	vpc-0ba0194cd3d3586a8	Available	Off	10.0.0.0/16	–	dopt-099336b96133a6...	rtb-0a477ae998e7b5b0f	acl-C
□	–	vpc-02df4998ca9a8ccf1	Available	Off	172.31.0.0/16	–	dopt-099336b96133a6...	rtb-0c3b7169b892395f6	acl-C

vpc-0ba0194cd3d3586a8 / lvpc

Resource map Info

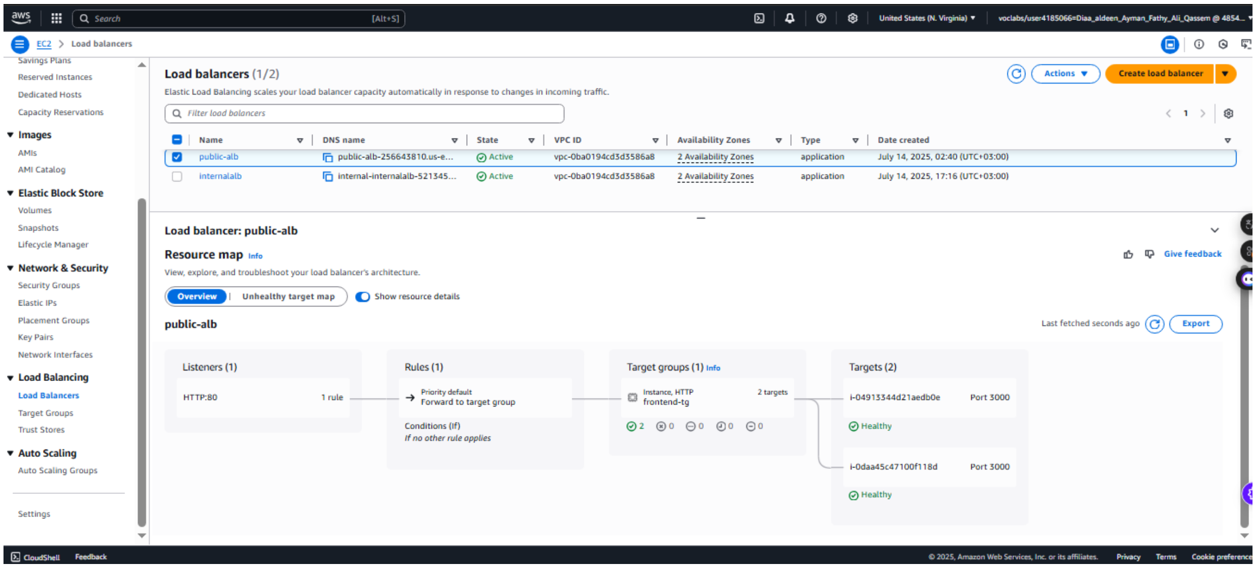
VPC Show details
Your AWS virtual network
lvpc

Subnets (4)
Subnets within this VPC
us-east-1a
public-subnet-1
private-subnet-1
us-east-1b
public-subnet-2
private-subnet-2

Route tables (3)
Route network traffic to resources
private-rt
rtb-0a477ae998e7b5b0f
public-rt

Network connections (2)
Connections to other networks
igw
vpc-nat

2- Resource Map For Public LB



Load balancers (1/2)

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

☑	Name	DNS name	State	VPC ID	Availability Zones	Type	Date created
☑	public-alb	public-alb-256643810.us-e...	Active	vpc-0ba0194cd3d3586a8	2 Availability Zones	application	July 14, 2025, 02:40 (UTC+03:00)
□	internalalb	internal-internalalb-521345...	Active	vpc-0ba0194cd3d3586a8	2 Availability Zones	application	July 14, 2025, 17:16 (UTC+03:00)

Load balancer: public-alb

Resource map Info

View, explore, and troubleshoot your load balancer's architecture.

public-alb

Last fetched seconds ago

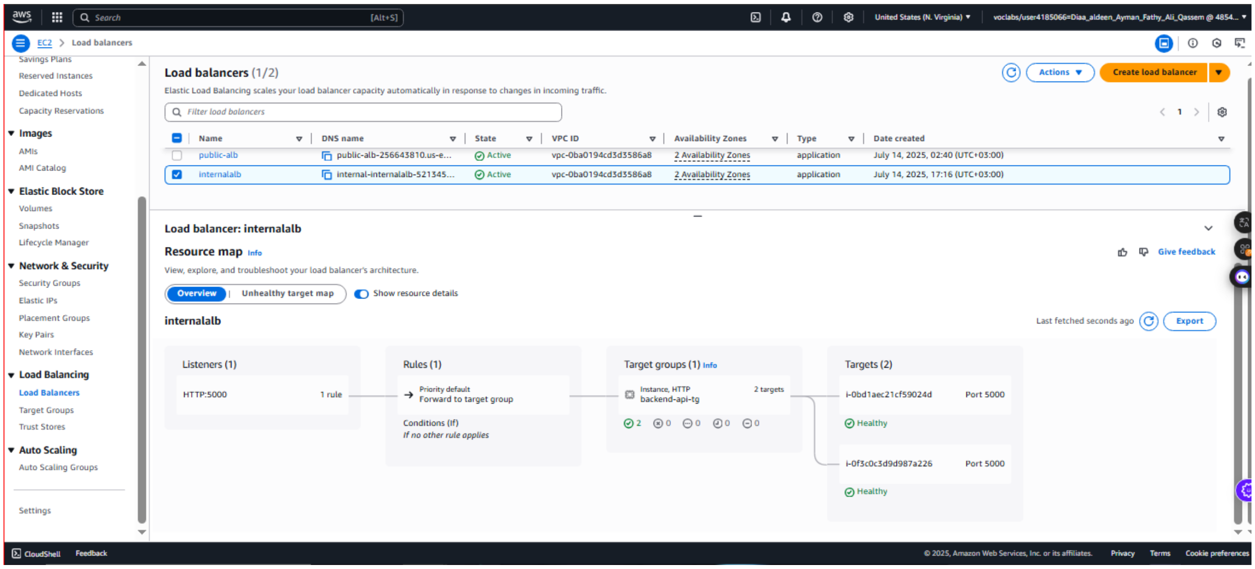
Listeners (1)
HTTP:80 1 rule

Rules (1)
Priority default
Forward to target group
Conditions (If)
If no other rule applies

Target groups (1) Info
Instance, HTTP
frontend-tg
2 targets
2 0 0 0 0

Targets (2)
i-04913344d21aedb0e Port 3000
Healthy
i-0daa45c47100f118d Port 3000
Healthy

3- Resource Map For Internal LB



Load balancers (1/2)

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

☐	Name	DNS name	State	VPC ID	Availability Zones	Type	Date created
☐	public-alb	public-alb-256643810.us-e...	✓ Active	vpc-0ba0194cd3d3586a8	2 Availability Zones	application	July 14, 2025, 02:40 (UTC+03:00)
☑	internalalb	internal-internalalb-521345...	✓ Active	vpc-0ba0194cd3d3586a8	2 Availability Zones	application	July 14, 2025, 17:16 (UTC+03:00)

Load balancer: internalalb

Resource map Info

View, explore, and troubleshoot your load balancer's architecture.

internalalb

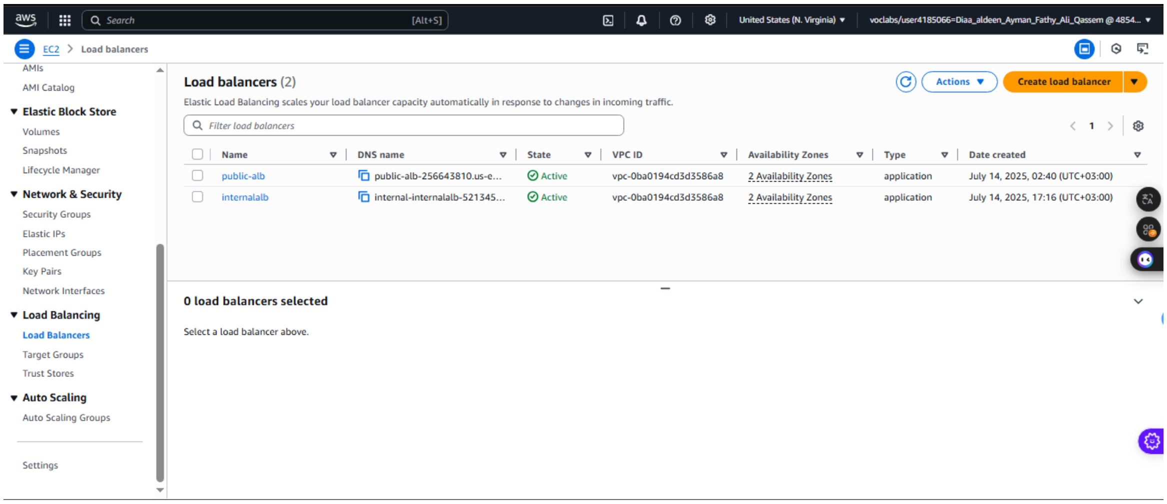
Last fetched seconds ago

Listeners (1)
HTTP:5000 1 rule

Rules (1)
Priority default
Forward to target group
Conditions (If)
If no other rule applies

Target groups (1) Info
Instance, HTTP
backend-api-tg
2 targets
✓ 2 ⊗ 0 ⊖ 0 ◷ 0 ⊖ 0

Targets (2)
i-0bd1aec21cf59024d Port 5000
✓ Healthy
i-0f3c0c3d9d987a226 Port 5000
✓ Healthy



Load balancers (2)

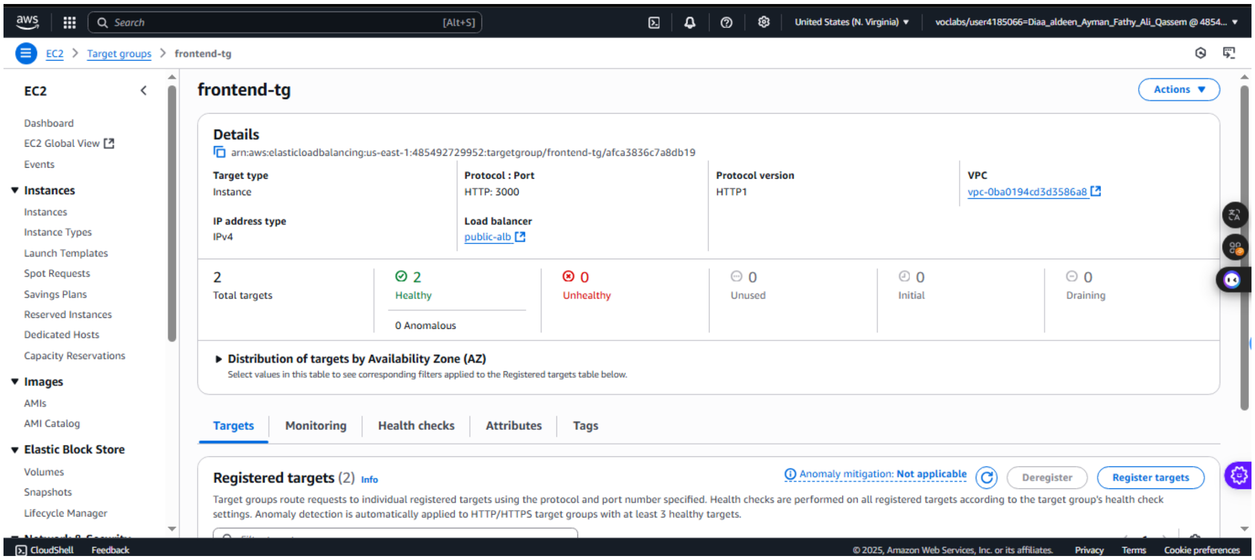
Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.

☐	Name	DNS name	State	VPC ID	Availability Zones	Type	Date created
☐	public-alb	public-alb-256643810.us-e...	✓ Active	vpc-0ba0194cd3d3586a8	2 Availability Zones	application	July 14, 2025, 02:40 (UTC+03:00)
☐	internalalb	internal-internalalb-521345...	✓ Active	vpc-0ba0194cd3d3586a8	2 Availability Zones	application	July 14, 2025, 17:16 (UTC+03:00)

0 load balancers selected

Select a load balancer above.

The frontend target group here shows that both servers are in a healthy state.



frontend-tg

Details

arn:aws:elasticloadbalancing:us-east-1:485492729952:targetgroup/frontend-tg/afca3836c7a8db19

Target type	Protocol : Port	Protocol version	VPC
Instance	HTTP: 3000	HTTP1	vpc-0ba0194cd3d3586a8
IP address type	**Load balancer**		
IPv4	public-alb		

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

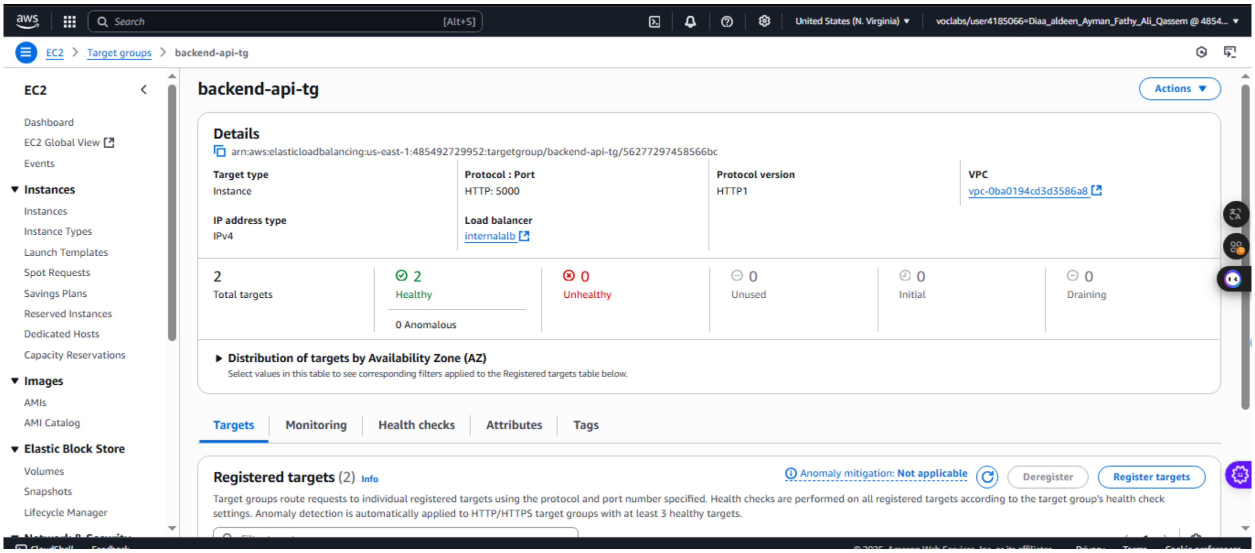
▸ Distribution of targets by Availability Zone (AZ)
Select values in this table to see corresponding filters applied to the Registered targets table below.

Registered targets (2) Info

Anomaly mitigation: Not applicable

Target groups route requests to individual registered targets using the protocol and port number specified. Health checks are performed on all registered targets according to the target group's health check settings. Anomaly detection is automatically applied to HTTP/HTTPS target groups with at least 3 healthy targets.

Also, the backend target group shows that both servers are in a healthy state.



backend-api-tg

Details

arn:aws:elasticloadbalancing:us-east-1:485492729952:targetgroup/backend-api-tg/56277297458566bc

Target type	Protocol : Port	Protocol version	VPC
Instance	HTTP: 5000	HTTP1	vpc-0ba0194cd3d3586a8
IP address type	**Load balancer**		
IPv4	internalalb		

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

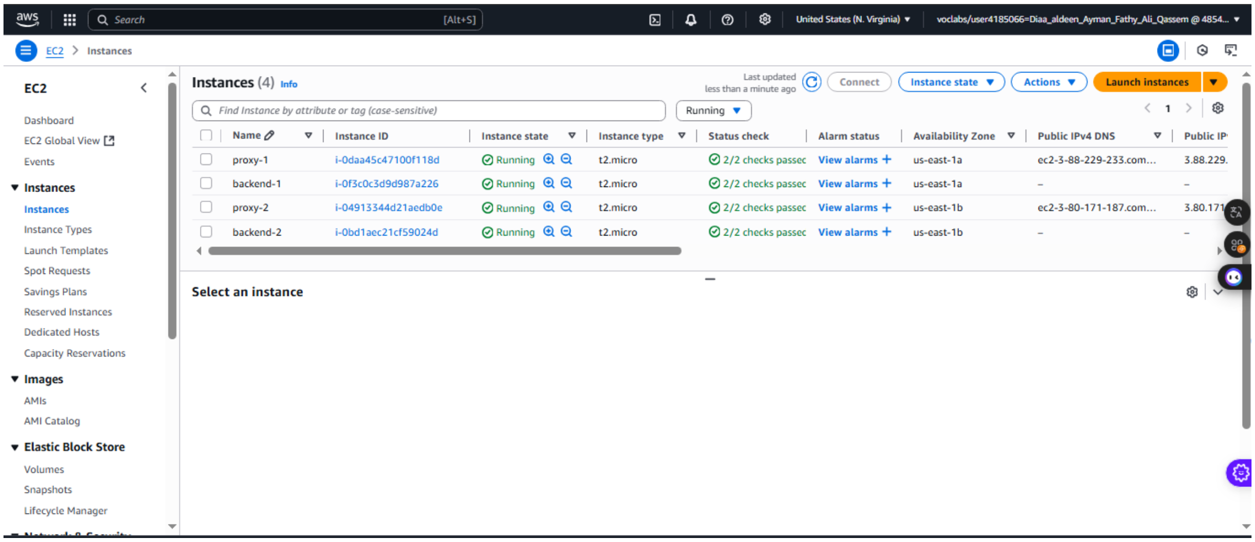
▸ Distribution of targets by Availability Zone (AZ)
Select values in this table to see corresponding filters applied to the Registered targets table below.

Registered targets (2) Info

Anomaly mitigation: Not applicable

Target groups route requests to individual registered targets using the protocol and port number specified. Health checks are performed on all registered targets according to the target group's health check settings. Anomaly detection is automatically applied to HTTP/HTTPS target groups with at least 3 healthy targets.

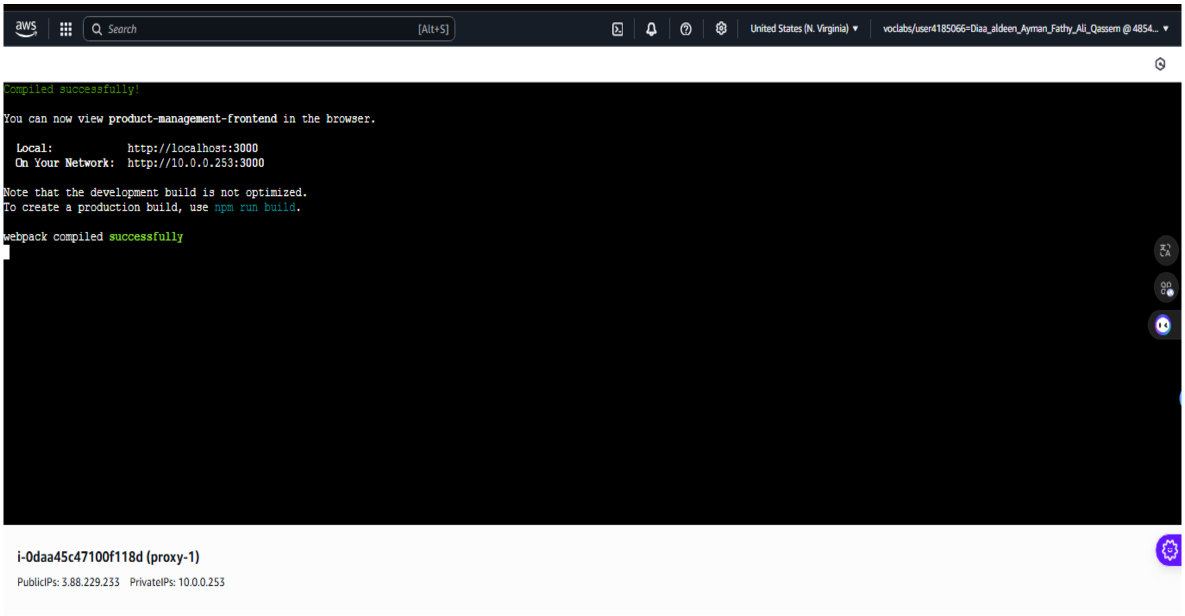
Here, there are four EC2 instances created. Two for the reverse proxy and two for the backend.



Instances (4)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IP
proxy-1	i-0daa45c47100f118d	Running	t2.micro	2/2 checks passec	View alarms +	us-east-1a	ec2-3-88-229-233.com...	3.88.229.
backend-1	i-0f3c0c3d9d987a226	Running	t2.micro	2/2 checks passec	View alarms +	us-east-1a	–	–
proxy-2	i-04913344d21aedb0e	Running	t2.micro	2/2 checks passec	View alarms +	us-east-1b	ec2-3-80-171-187.com...	3.80.171
backend-2	i-0bd1aec21cf59024d	Running	t2.micro	2/2 checks passec	View alarms +	us-east-1b	–	–

The following two snapshots include accessing the two proxy EC2 instances via the console. These instances include the frontend of the application.

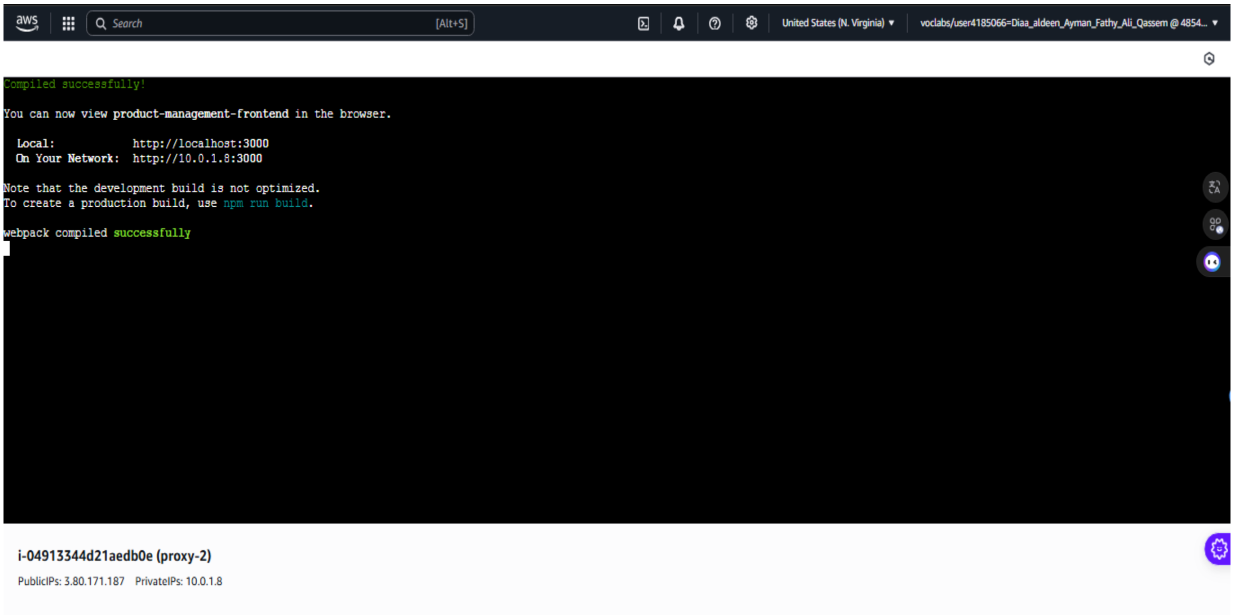


```
Compiled successfully!

You can now view product-management-frontend in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://10.0.0.253:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```

i-0daa45c47100f118d (proxy-1)

PublicIPs: 3.88.229.233   PrivateIPs: 10.0.0.253

Search
[Alt+S]
United States (N. Virginia)
voclabs/user4185066=Diaa_aldeen_Ayman_Fathy_Ali_Qassem @ 4854...

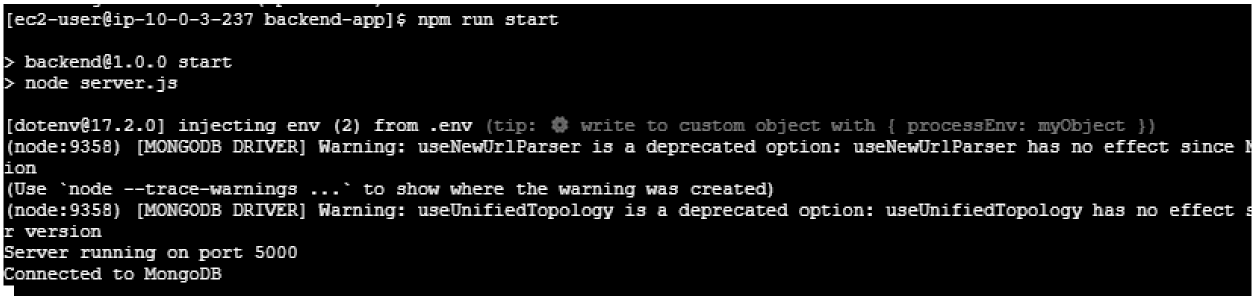

aws
Search [Alt+S]
United States (N. Virginia) ▾
voclabs/user4185066=Diaa_aldeen_Ayman_Fathy_Ali_Qassem @ 4854... ▾

```
Compiled successfully!

You can now view product-management-frontend in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://10.0.1.8:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```

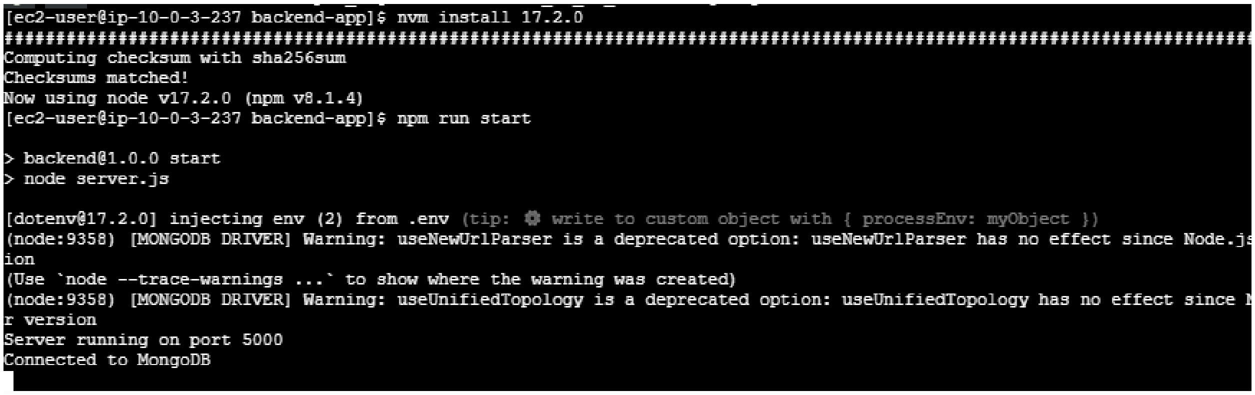
i-04913344d21aedb0e (proxy-2)

PublicIPs: 3.80.171.187 PrivateIPs: 10.0.1.8

Here, I have accessed the backend instances via SSH from the proxy instances by adding the public key of the instances for passwordless access.

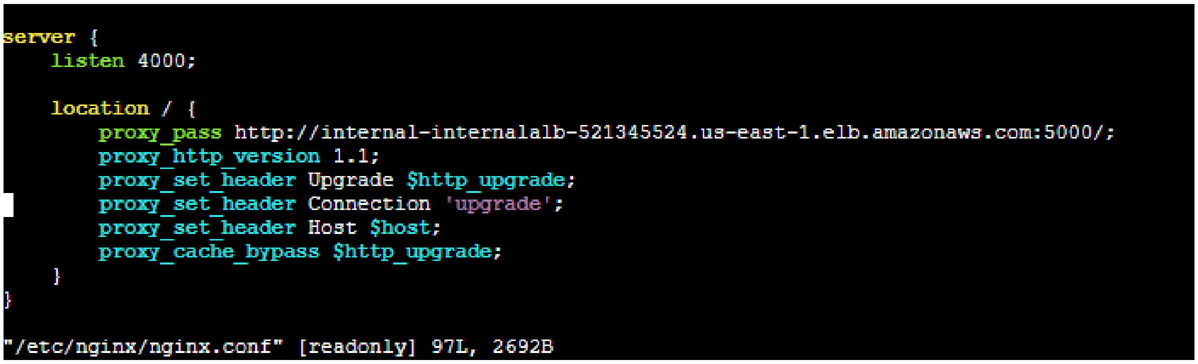


```
[ec2-user@ip-10-0-3-237 backend-app]$ npm run start

> backend@1.0.0 start
> node server.js

[dotenv@17.2.0] injecting env (2) from .env (tip: ⚙ write to custom object with { processEnv: myObject })
(node:9358) [MONGODB DRIVER] Warning: useNewUrlParser is a deprecated option: useNewUrlParser has no effect since N
ion
(Use `node --trace-warnings ...` to show where the warning was created)
(node:9358) [MONGODB DRIVER] Warning: useUnifiedTopology is a deprecated option: useUnifiedTopology has no effect s
r version
Server running on port 5000
Connected to MongoDB
```



```
[ec2-user@ip-10-0-3-237 backend-app]$ nvm install 17.2.0
################################################################################################
Computing checksum with sha256sum
Checksums matched!
Now using node v17.2.0 (npm v8.1.4)
[ec2-user@ip-10-0-3-237 backend-app]$ npm run start

> backend@1.0.0 start
> node server.js

[dotenv@17.2.0] injecting env (2) from .env (tip: ⚙ write to custom object with { processEnv: myObject })
(node:9358) [MONGODB DRIVER] Warning: useNewUrlParser is a deprecated option: useNewUrlParser has no effect since Node.js
ion
(Use `node --trace-warnings ...` to show where the warning was created)
(node:9358) [MONGODB DRIVER] Warning: useUnifiedTopology is a deprecated option: useUnifiedTopology has no effect since N
r version
Server running on port 5000
Connected to MongoDB
```

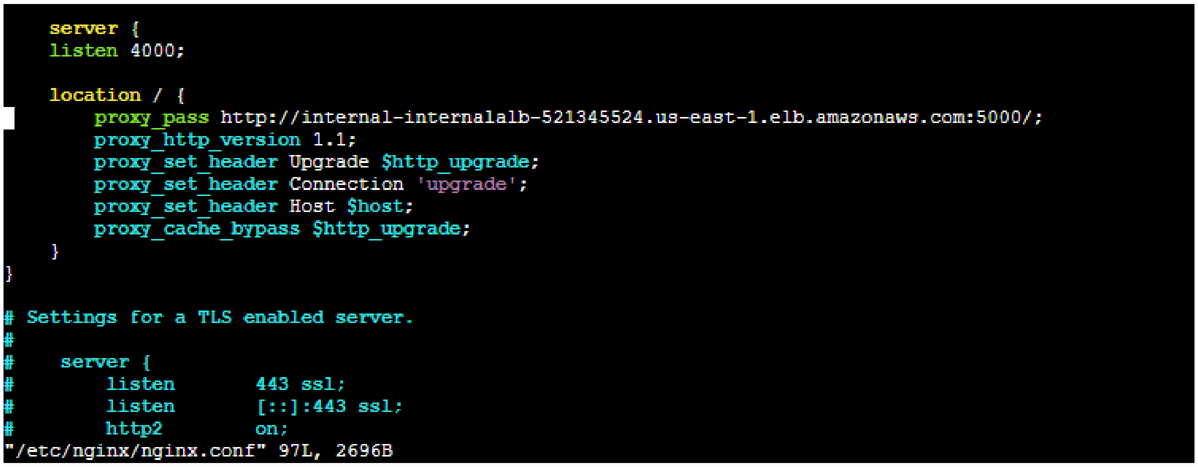
Here, I have configured the Reverse proxy for the instances.



```
server {
    listen 4000;

    location / {
        proxy_pass http://internal-internalalb-521345524.us-east-1.elb.amazonaws.com:5000/;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }
}

"/etc/nginx/nginx.conf" [readonly] 97L, 2692B
```

i-04913344d21aedb0e (proxy-2)

PublicIPs: 3.80.171.187   PrivateIPs: 10.0.1.8

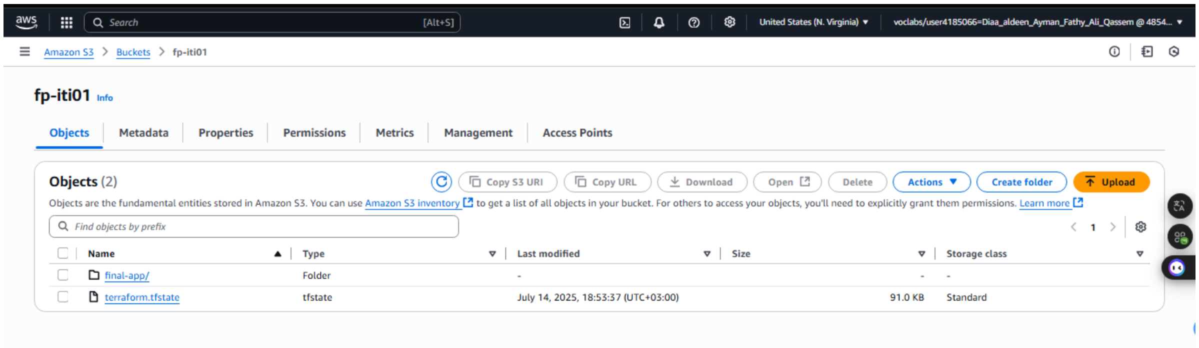


```
    server {
    listen 4000;

    location / {
        proxy_pass http://internal-internalalb-521345524.us-east-1.elb.amazonaws.com:5000/;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }
}

# Settings for a TLS enabled server.
#
#    server {
#        listen       443 ssl;
#        listen       [::]:443 ssl;
#        http2        on;
"/etc/nginx/nginx.conf" 97L, 2696B
```

i-0daa45c47100f118d (proxy-1)

PublicIPs: 3.88.229.233   PrivateIPs: 10.0.0.253

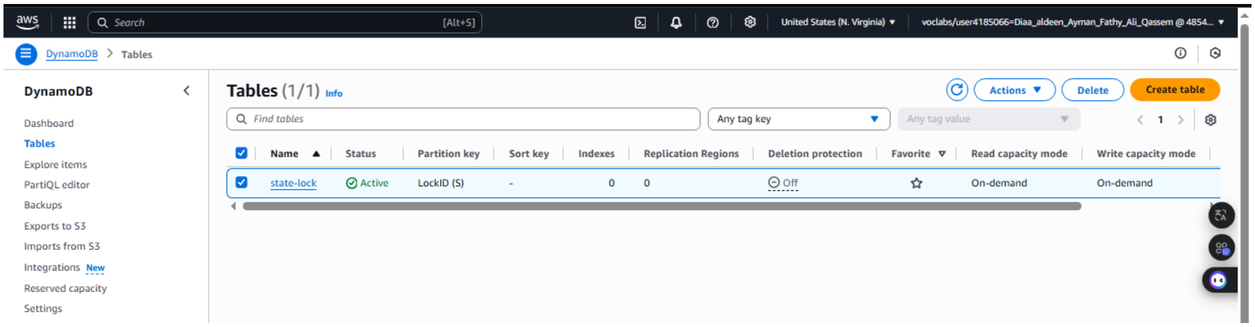
Since we specified the backend as the S3 bucket to store the state file, we can view the state file here.



Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

□	Name	Type	Last modified	Size	Storage class
□	final-app/	Folder	-	-	-
□	terraform.tfstate	tfstate	July 14, 2025, 18:53:37 (UTC+03:00)	91.0 KB	Standard

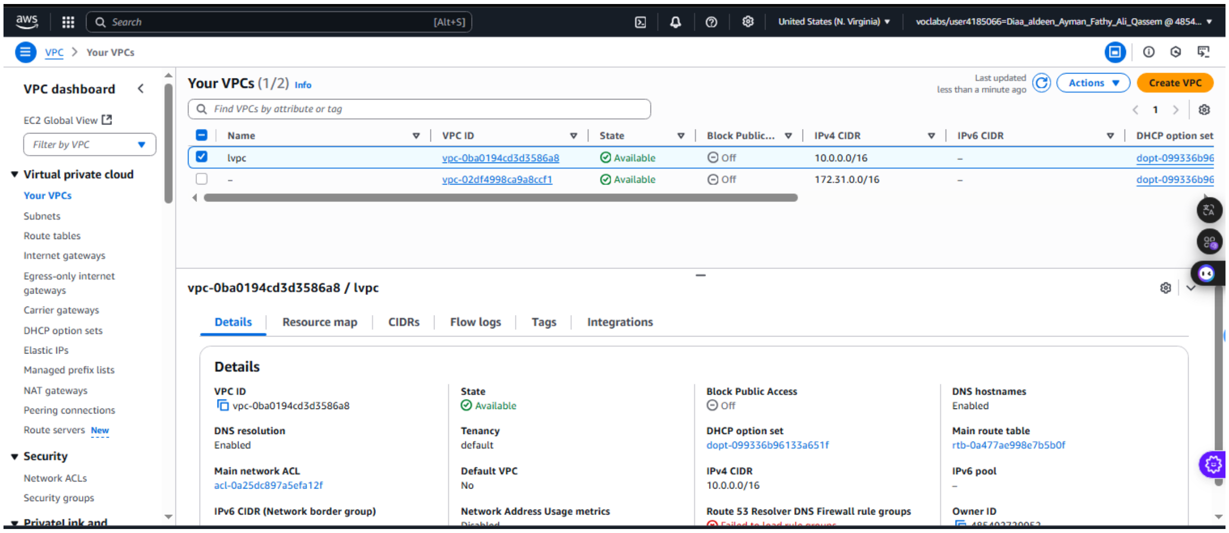
Also, I have created a DynamoDB table to maintain the state of the state file and prevent simultaneous edits on it.



Tables (1/1) Info

☑	Name ▲	Status	Partition key	Sort key	Indexes	Replication Regions	Deletion protection	Favorite	Read capacity mode	Write capacity mode
☑	state-lock	✓ Active	LockID (S)	-	0	0	Off	☆	On-demand	On-demand

The following picture shows the VPC created.



Your VPCs (1/2) Info

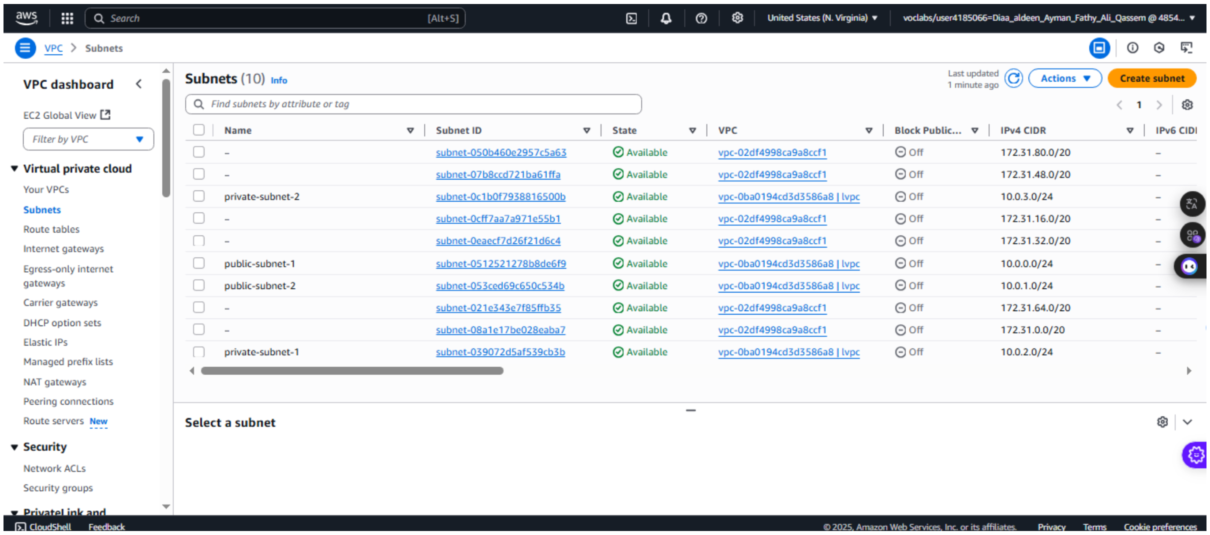
	Name	VPC ID	State	Block Public...	IPv4 CIDR	IPv6 CIDR	DHCP option set
☑	lvpc	vpc-0ba0194cd3d3586a8	✓ Available	Off	10.0.0.0/16	–	dopt-099336b96
□	–	vpc-02df4998ca9a8ccf1	✓ Available	Off	172.31.0.0/16	–	dopt-099336b96

vpc-0ba0194cd3d3586a8 / lvpc

Details

VPC ID vpc-0ba0194cd3d3586a8	State ✓ Available	Block Public Access Off	DNS hostnames Enabled
DNS resolution Enabled	Tenancy default	DHCP option set dopt-099336b96133a651f	Main route table rtb-0a477ae998e7b5b0f
Main network ACL acl-0a25dc897a5efa12f	Default VPC No	IPv4 CIDR 10.0.0.0/16	IPv6 pool –
IPv6 CIDR (Network border group)	Network Address Usage metrics Disabled	Route 53 Resolver DNS Firewall rule groups Failed to load rule groups	Owner ID 485402720052

The VPC contains four subnets. Two private subnets and two public subnets.

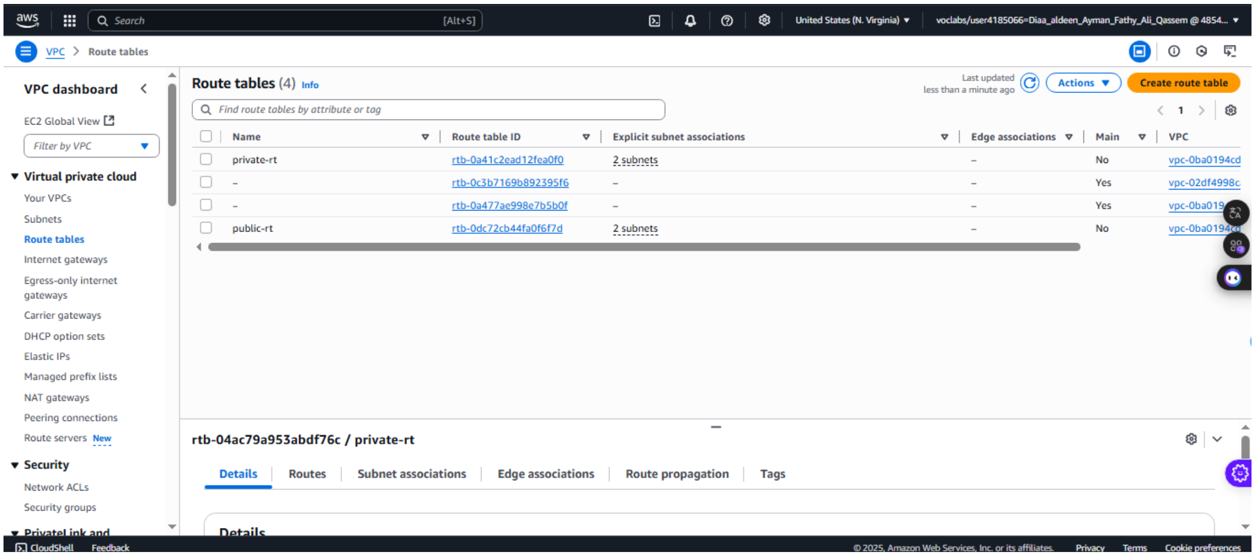


Subnets (10) Info

□	Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR	IPv6 CIDI
□	–	subnet-050b460e2957c5a63	✓ Available	vpc-02df4998ca9a8ccf1	Off	172.31.80.0/20	–
□	–	subnet-07b8ccd721ba61ffa	✓ Available	vpc-02df4998ca9a8ccf1	Off	172.31.48.0/20	–
□	private-subnet-2	subnet-0c1b0f7938816500b	✓ Available	vpc-0ba0194cd3d3586a8 | lvpc	Off	10.0.3.0/24	–
□	–	subnet-0cff7aa7a971e55b1	✓ Available	vpc-02df4998ca9a8ccf1	Off	172.31.16.0/20	–
□	–	subnet-0eaecf7d26f21d6c4	✓ Available	vpc-02df4998ca9a8ccf1	Off	172.31.32.0/20	–
□	public-subnet-1	subnet-0512521278b8de6f9	✓ Available	vpc-0ba0194cd3d3586a8 | lvpc	Off	10.0.0.0/24	–
□	public-subnet-2	subnet-053ced69c650c534b	✓ Available	vpc-0ba0194cd3d3586a8 | lvpc	Off	10.0.1.0/24	–
□	–	subnet-021e343e7f85ffb35	✓ Available	vpc-02df4998ca9a8ccf1	Off	172.31.64.0/20	–
□	–	subnet-08a1e17be028eaba7	✓ Available	vpc-02df4998ca9a8ccf1	Off	172.31.0.0/20	–
□	private-subnet-1	subnet-039072d5af539cb3b	✓ Available	vpc-0ba0194cd3d3586a8 | lvpc	Off	10.0.2.0/24	–

Select a subnet

I also associated the public subnet with a route table called "public-rt", and the private subnet with another route table called "private-rt"



Route tables (4) Info

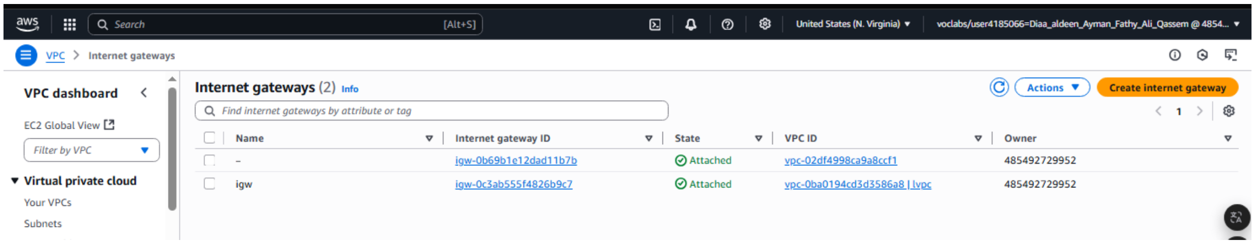
Name	Route table ID	Explicit subnet associations	Edge associations	Main	VPC
private-rt	rtb-0a41c2ead12fea0f0	2 subnets	–	No	vpc-0ba0194cd
–	rtb-0c3b7169b892395f6	–	–	Yes	vpc-02df4998c
–	rtb-0a477ae998e7b5b0f	–	–	Yes	vpc-0ba019
public-rt	rtb-0dc72cb44fa0f6f7d	2 subnets	–	No	vpc-0ba0194c

rtb-04ac79a953abdf76c / private-rt

Details | Routes | Subnet associations | Edge associations | Route propagation | Tags

Details

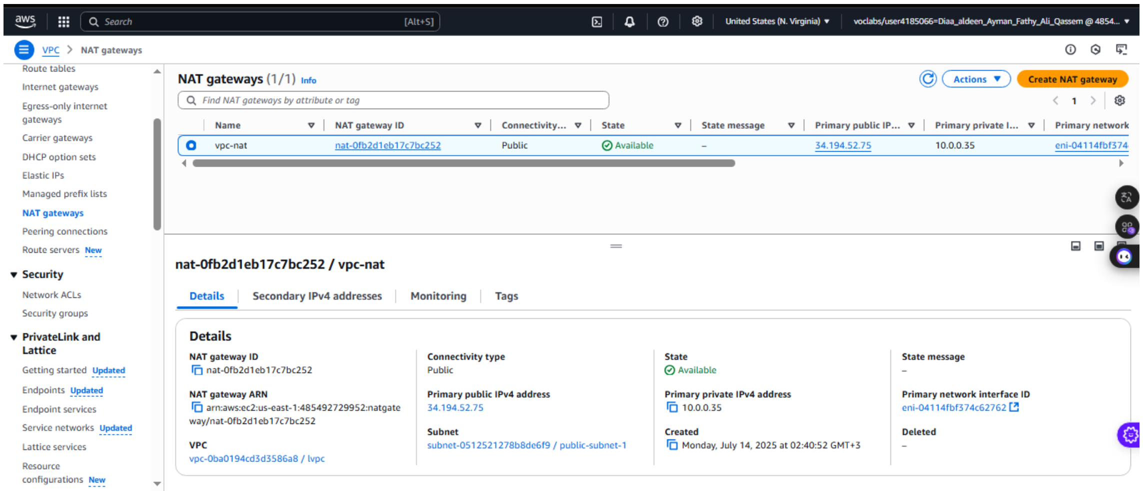
After, I created an internet gateway to allow access for the public EC2 instances on the internet.



Internet gateways (2) Info

Name	Internet gateway ID	State	VPC ID	Owner
–	igw-0b69b1e12dad11b7b	Attached	vpc-02df4998ca9a8ccf1	485492729952
igw	igw-0c3ab555f4826b9c7	Attached	vpc-0ba0194cd3d3586a8 | lvpc	485492729952

The, I created a NAT gateway as well to allow communication for the private EC2 instances



NAT gateways (1/1) Info

Name	NAT gateway ID	Connectivity...	State	State message	Primary public IP...	Primary private I...	Primary network
vpc-nat	nat-0fb2d1eb17c7bc252	Public	Available	–	34.194.52.75	10.0.0.35	eni-04114fbf374

nat-0fb2d1eb17c7bc252 / vpc-nat

Details | Secondary IPv4 addresses | Monitoring | Tags

Details

NAT gateway ID
nat-0fb2d1eb17c7bc252

NAT gateway ARN
arn:aws:ec2:us-east-1:485492729952:natgateway/nat-0fb2d1eb17c7bc252

VPC
vpc-0ba0194cd3d3586a8 / lvpc

Connectivity type
Public

Primary public IPv4 address
34.194.52.75

Subnet
subnet-0512521278b8de6f9 / public-subnet-1

State
Available

Primary private IPv4 address
10.0.0.35

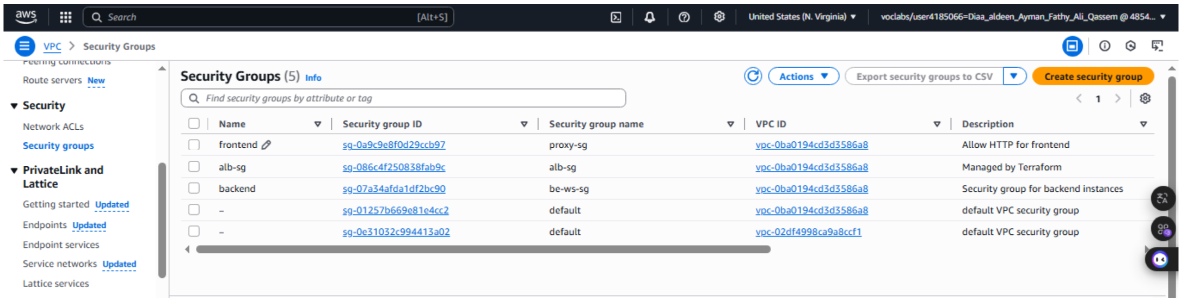
Created
Monday, July 14, 2025 at 02:40:52 GMT+3

State message
–

Primary network interface ID
eni-04114fbf374c62762

Deleted
–

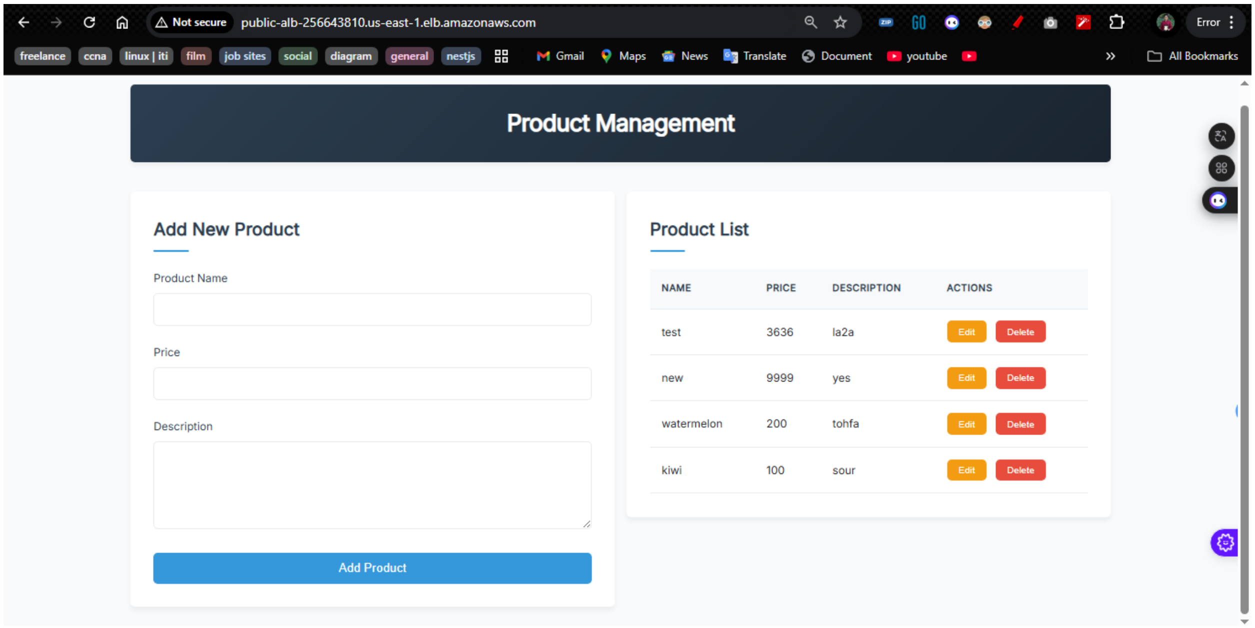
Further, I created security groups for the frontend, the backend, and the ALB, and allowed the needed rules on these SGs.



Security Groups (5) Info

Name	Security group ID	Security group name	VPC ID	Description
frontend	sg-0a9c9e8f0d29ccb97	proxy-sg	vpc-0ba0194cd3d3586a8	Allow HTTP for frontend
alb-sg	sg-086c4f250838fab9c	alb-sg	vpc-0ba0194cd3d3586a8	Managed by Terraform
backend	sg-07a34afda1df2bc90	be-ws-sg	vpc-0ba0194cd3d3586a8	Security group for backend instances
–	sg-01257b669e81e4cc2	default	vpc-0ba0194cd3d3586a8	default VPC security group
–	sg-0e31032c994413a02	default	vpc-02df4998ca9a8ccf1	default VPC security group

The below screenshot shows the application running.



Product Management

Add New Product

Product Name

Price

Description

Add Product

Product List

NAME	PRICE	DESCRIPTION	ACTIONS
test	3636	la2a	Edit Delete
new	9999	yes	Edit Delete
watermelon	200	tohfa	Edit Delete
kiwi	100	sour	Edit Delete