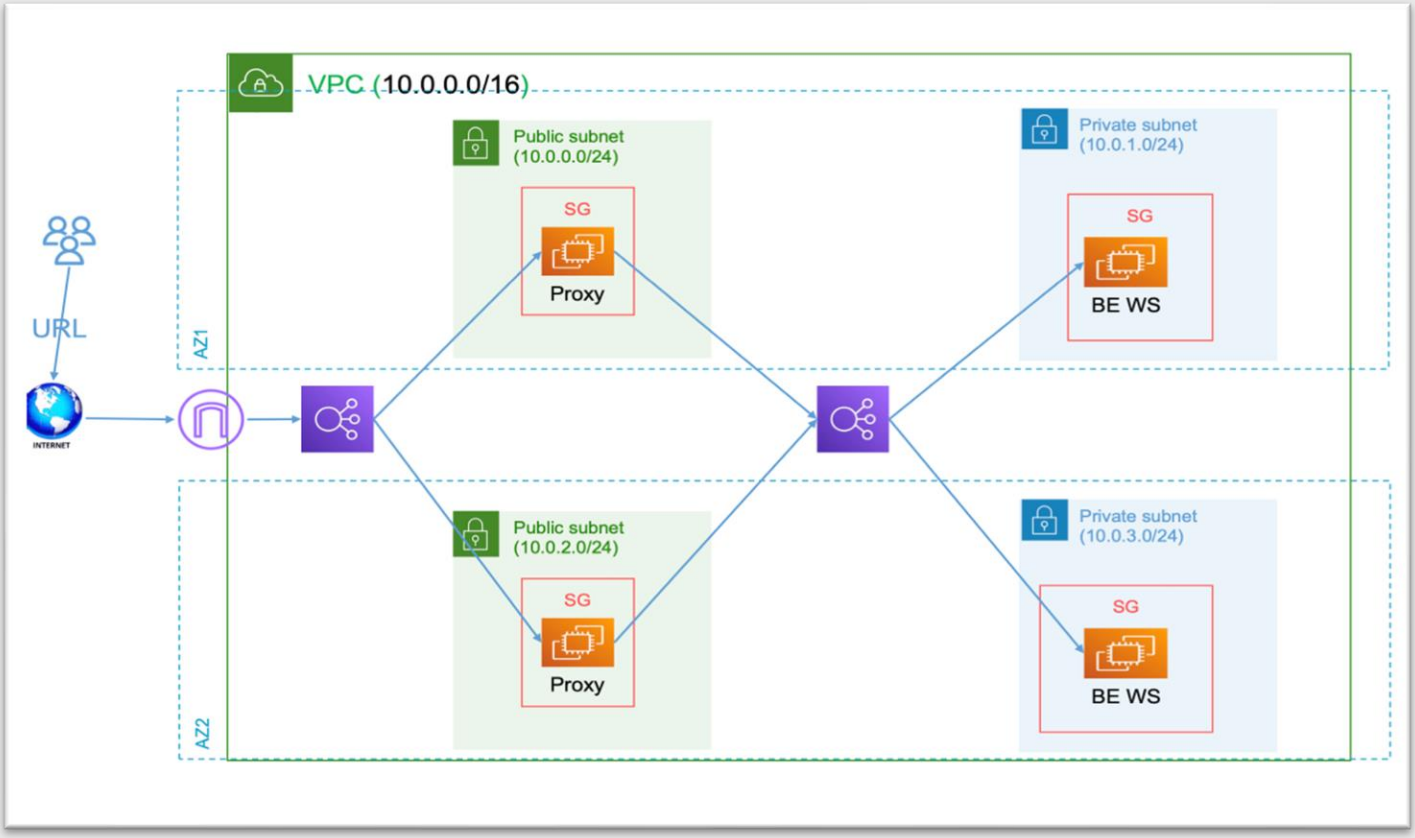
FINAL PROJECT

Basel Ashraf
Track: System Administration
Branch: Aswan

GitHub repository link:

https://github.com/basellashraf/terraform-project

Project diagram:



Project Requirements:

1- Don’t work on the default Workspace Create a new workspace called dev

2- Using customs not public modules to implement the diagram below

3- remote bucket for statefile

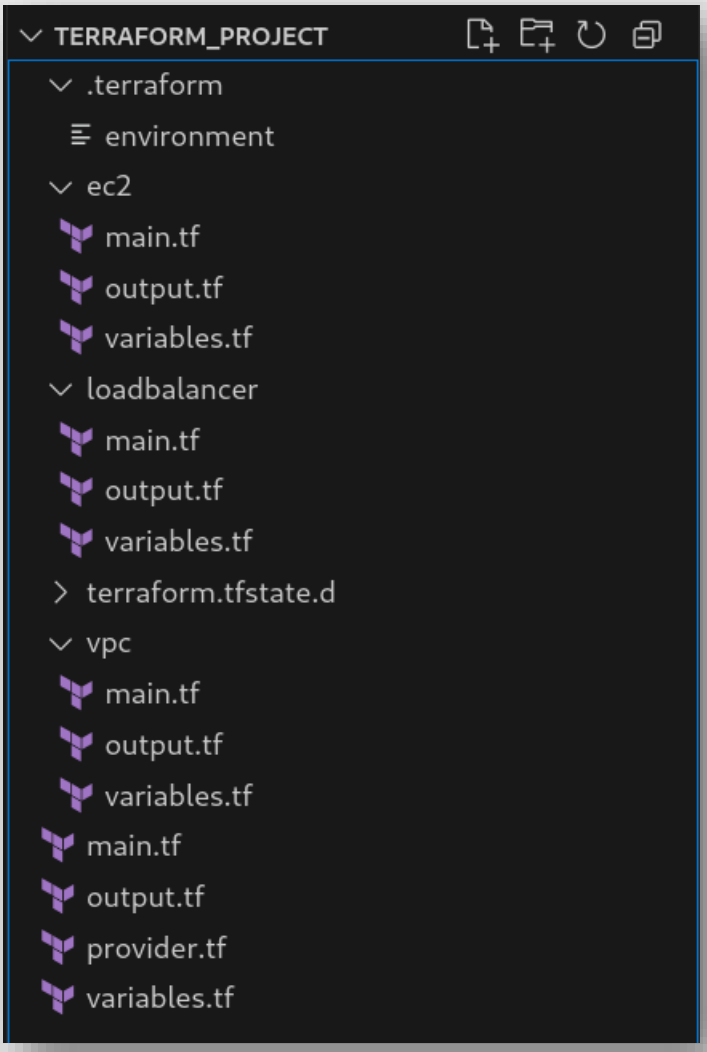
4- Use remote provisioner to install apache or proxy in machines then use local-exec to print all the ips to a file called all-ips.txt with format

public-ip1 1.1.1.1 public-ip2 2.2.2.2

5- Use the datasource to get the image id for ec2

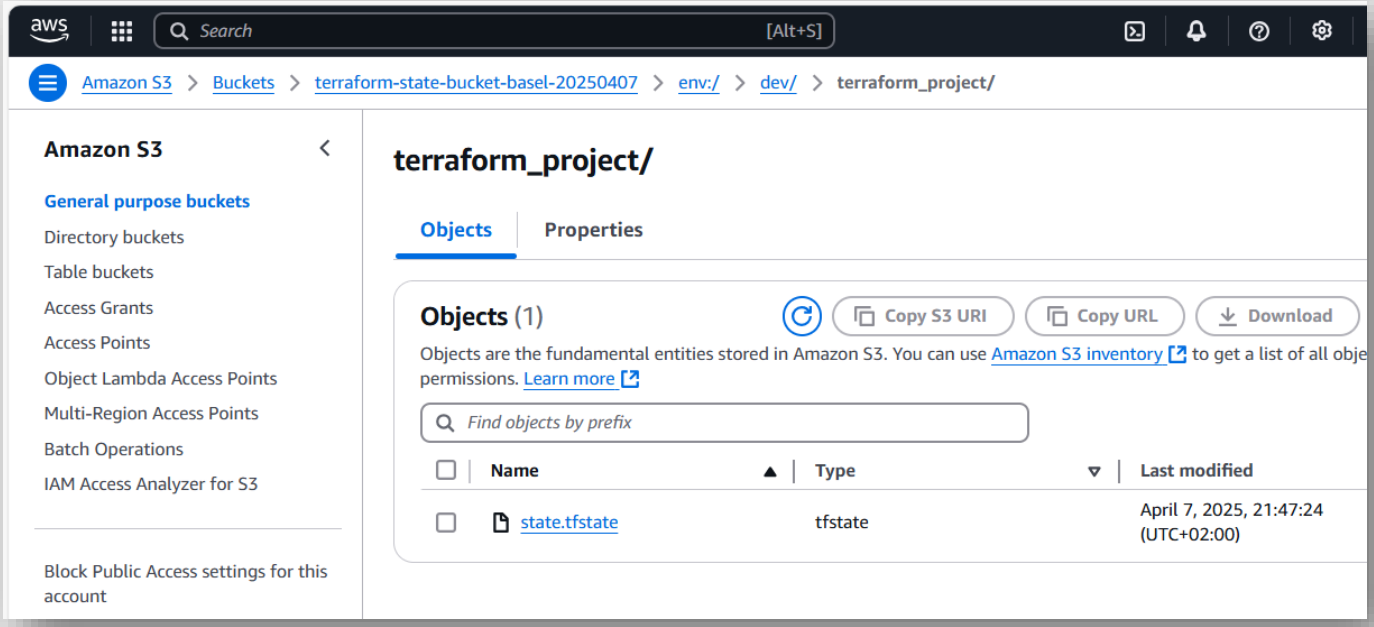
6- The first Loadbalancer is public, and the other one that will send the traffic to the private machines will be private

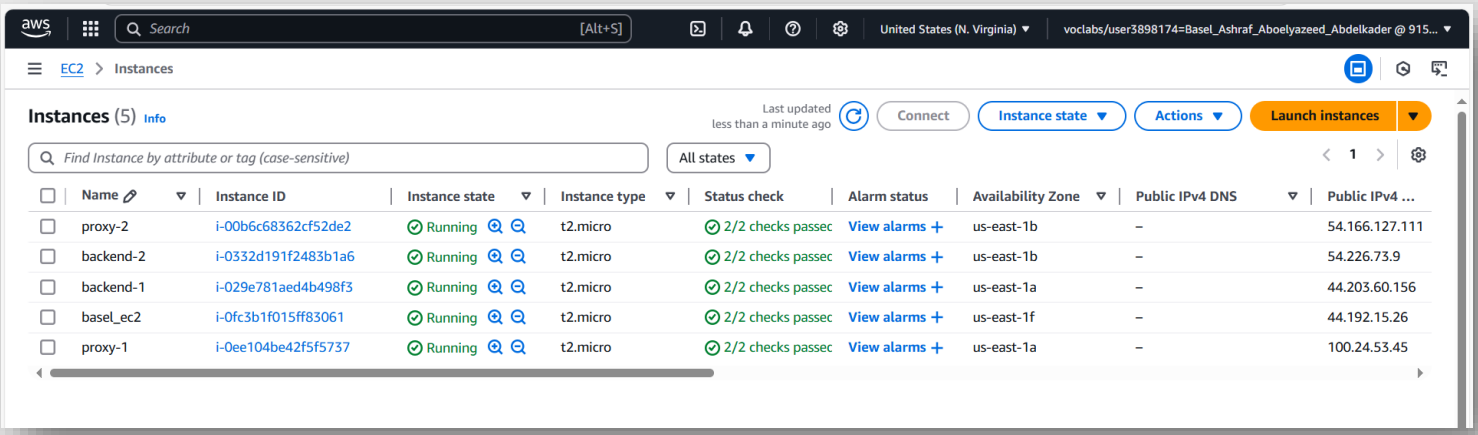
```
Activities  Terminal                                   Apr 7  01:21
                                                basel@serverb:~/terraform_project
[basel@serverb ~]$ cd terraform_project/
[basel@serverb terraform_project]$ terraform init
Initializing the backend...
Initializing provider plugins...

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.
[basel@serverb terraform_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.
[basel@serverb terraform_project]$ terraform workspace list
  default
* dev
```

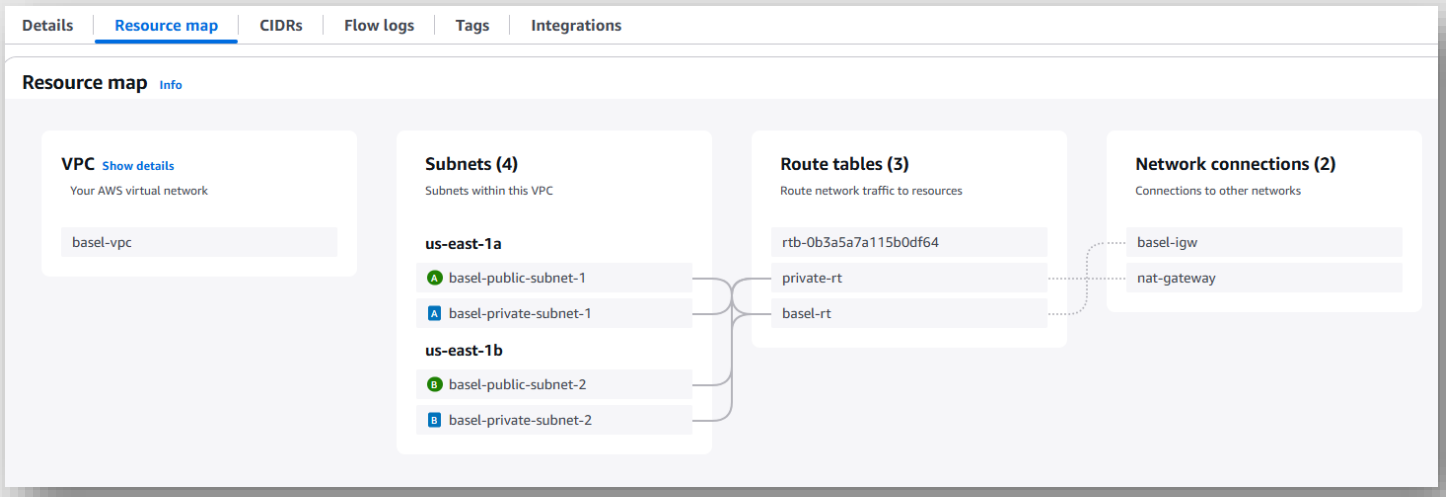


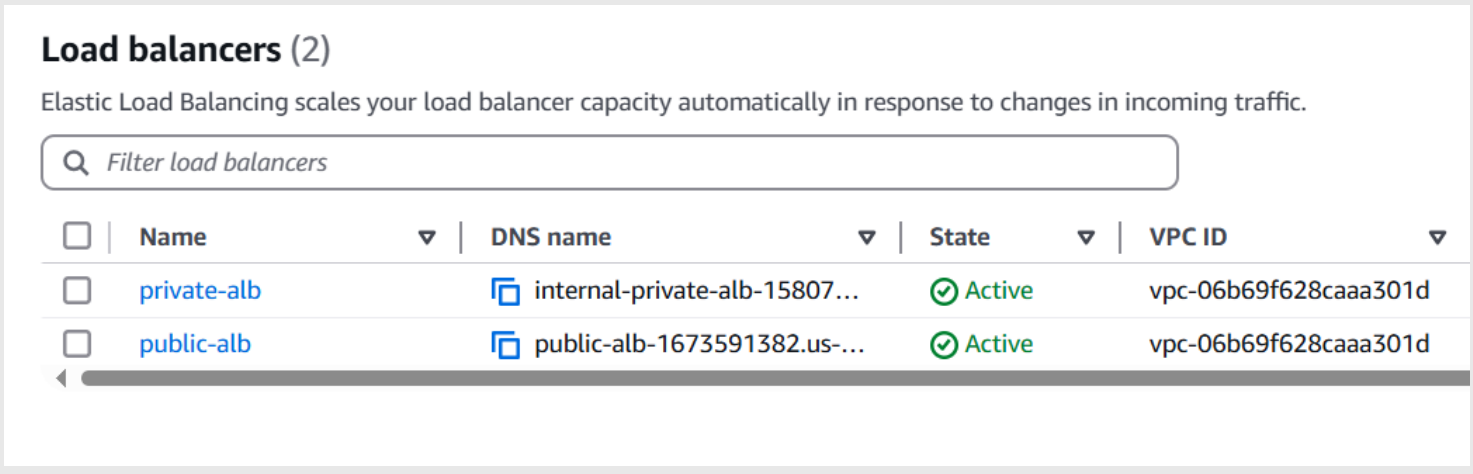
```
[basel@serverb terraform_project]$ aws s3api create-bucket --bucket terraform-state-bucket-basel-20250407 --region us-east-1
{
    "Location": "/terraform-state-bucket-basel-20250407"
}
[basel@serverb terraform_project]$ terraform init
Initializing the backend...
Do you want to migrate all workspaces to "s3"?
  Both the existing "local" backend and the newly configured "s3" backend
  support workspaces. When migrating between backends, Terraform will copy
  all workspaces (with the same names). THIS WILL OVERWRITE any conflicting
  states in the destination.

  Terraform initialization doesn't currently migrate only select workspaces.
  If you want to migrate a select number of workspaces, you must manually
  pull and push those states.

  If you answer "yes", Terraform will migrate all states. If you answer
  "no", Terraform will abort.

  Enter a value: yes
```

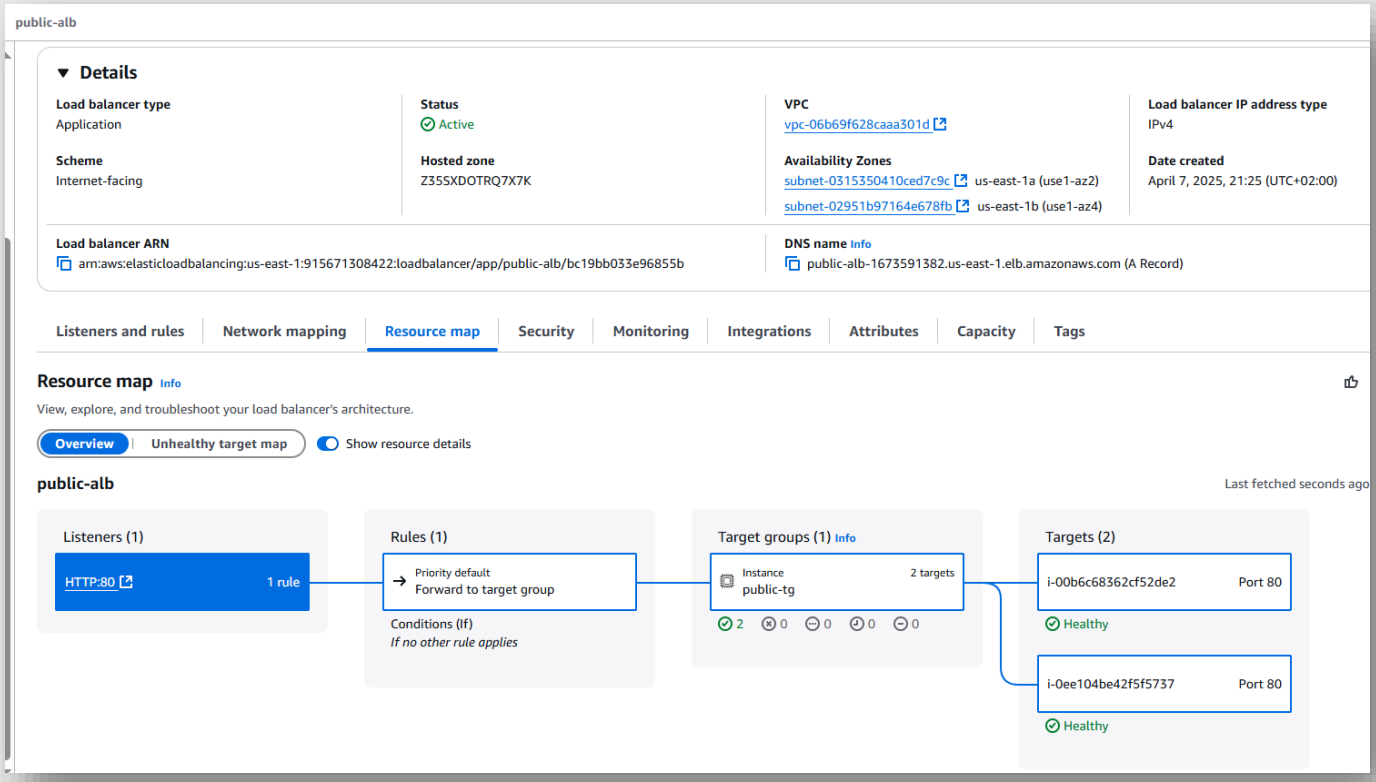
```
Successfully configured the backend "s3"! Terraform will automatically
use this backend unless the backend configuration changes.
Initializing modules...
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.94.1

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.
[basel@serverb terraform_project]$
```

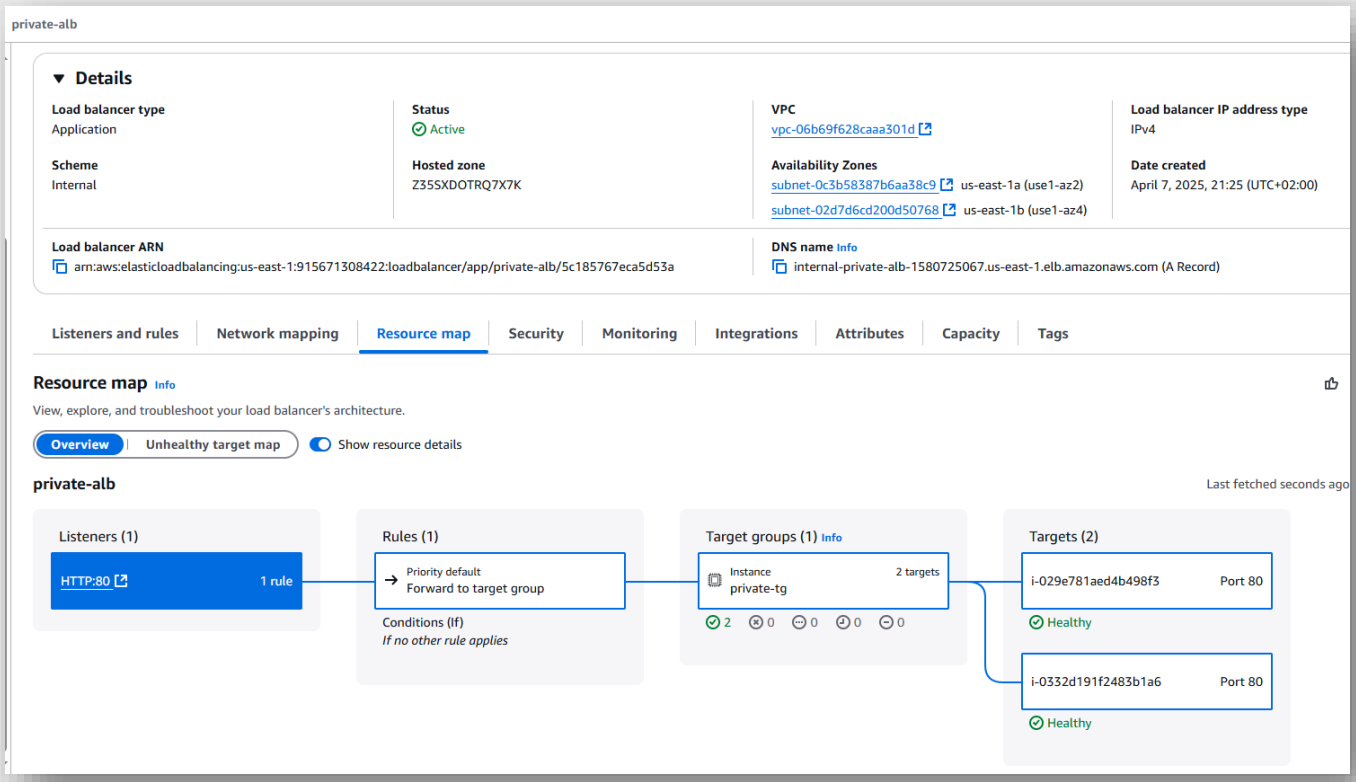


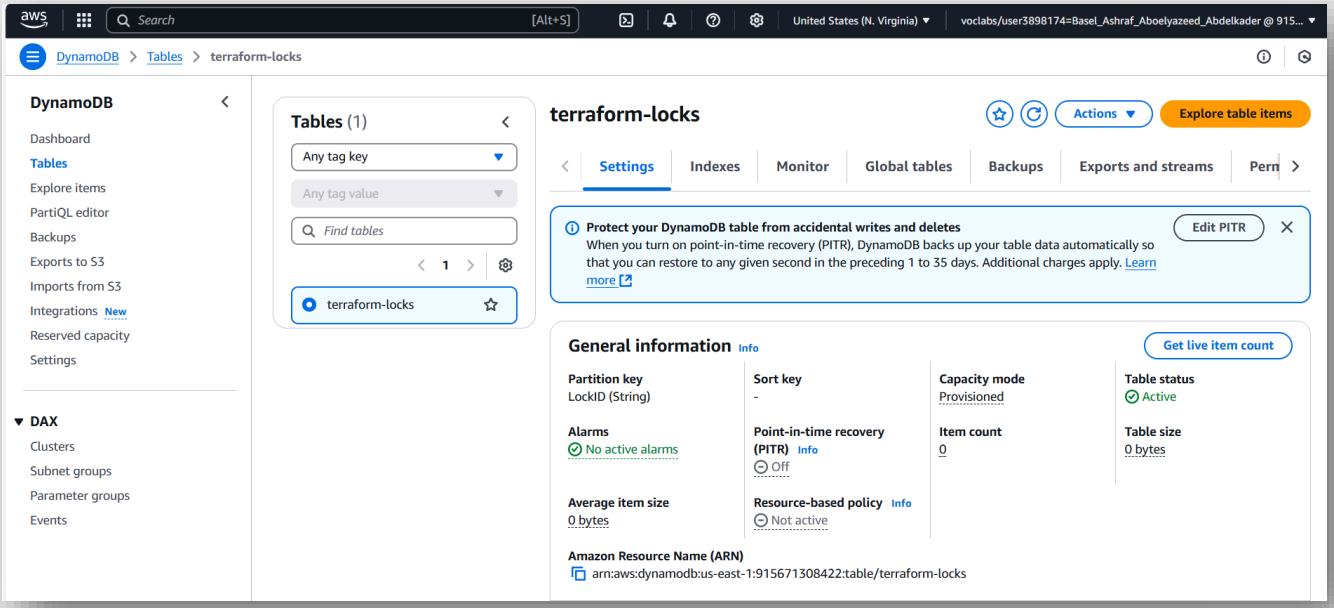

Search [Alt+S]
United States (N. Virginia)
voclabs/user3898174=Basel_Ashraf_Aboelyazeed_Abdelkader @ 915...
EC2 > Instances
Instances (5) Info
Last updated less than a minute ago
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
All states
Name | Instance ID | Instance state | Instance type | Status check | Alarm status | Availability Zone | Public IPv4 DNS | Public IPv4 ...
proxy-2 | i-00b6c68362cf52de2 | Running | t2.micro | 2/2 checks passec | View alarms | us-east-1b | – | 54.166.127.111
backend-2 | i-0332d191f2483b1a6 | Running | t2.micro | 2/2 checks passec | View alarms | us-east-1b | – | 54.226.73.9
backend-1 | i-029e781aed4b498f3 | Running | t2.micro | 2/2 checks passec | View alarms | us-east-1a | – | 44.203.60.156
basel_ec2 | i-0fc3b1f015ff83061 | Running | t2.micro | 2/2 checks passec | View alarms | us-east-1f | – | 44.192.15.26
proxy-1 | i-0ee104be42f5f5737 | Running | t2.micro | 2/2 checks passec | View alarms | us-east-1a | – | 100.24.53.45

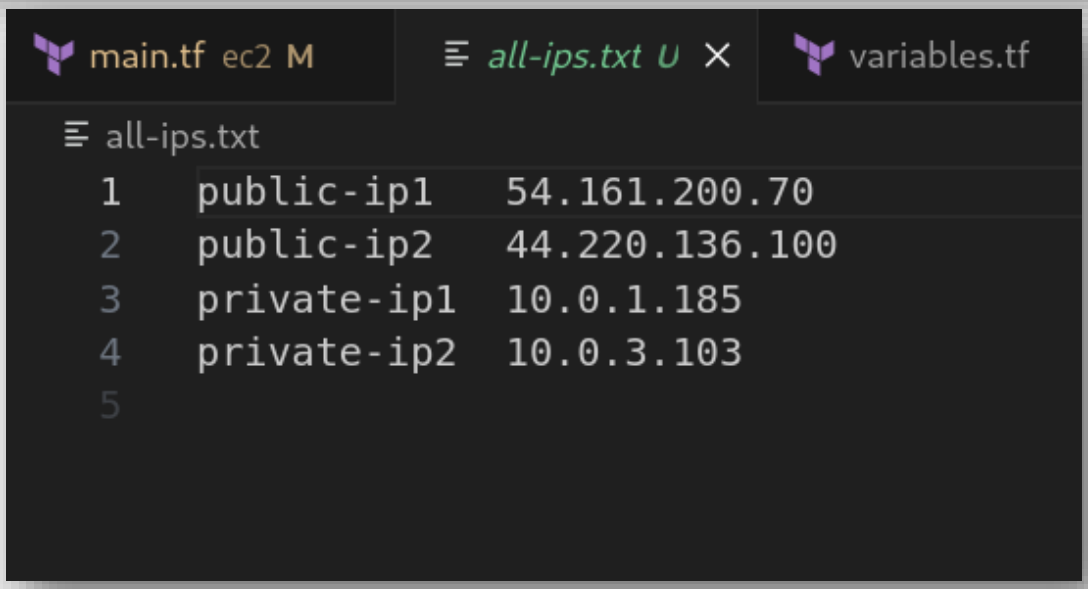

Details
Resource map
CIDRs
Flow logs
Tags
Integrations
Resource map Info
VPC Show details
Your AWS virtual network
basel-vpc
Subnets (4)
Subnets within this VPC
us-east-1a
basel-public-subnet-1
basel-private-subnet-1
us-east-1b
basel-public-subnet-2
basel-private-subnet-2
Route tables (3)
Route network traffic to resources
rtb-0b3a5a7a115b0df64
private-rt
basel-rt
Network connections (2)
Connections to other networks
basel-igw
nat-gateway

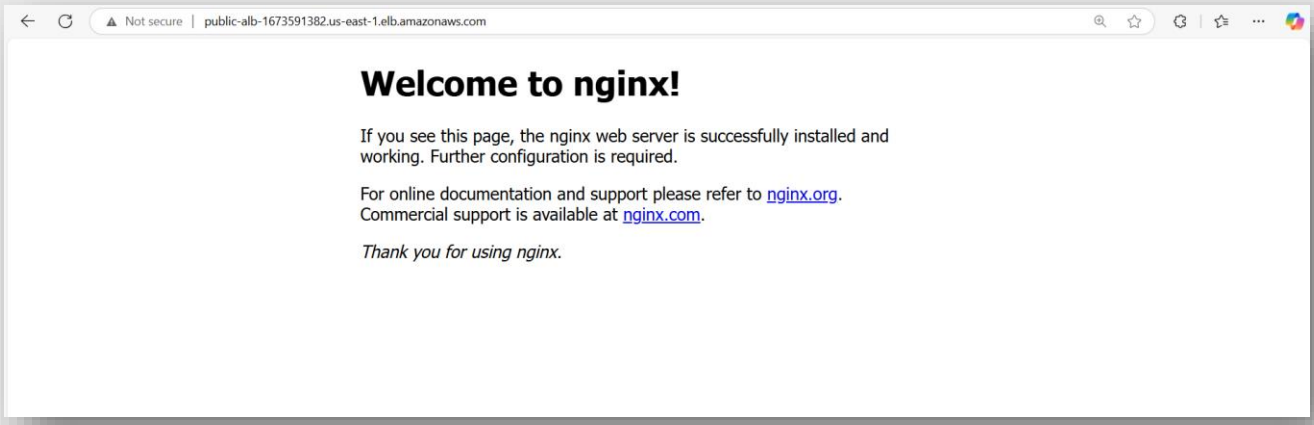

Load balancers (2)
Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.
Filter load balancers
Name | DNS name | State | VPC ID
private-alb | internal-private-alb-15807... | Active | vpc-06b69f628caaa301d
public-alb | public-alb-1673591382.us-... | Active | vpc-06b69f628caaa301d


public-alb
Details
Load balancer type
Application
Status
Active
VPC
vpc-06b69f628caaa301d
Load balancer IP address type
IPv4
Scheme
Internet-facing
Hosted zone
Z35SXDOTRQ7X7K
Availability Zones
subnet-0315350410ced7c9c us-east-1a (use1-az2)
subnet-02951b97164e678fb us-east-1b (use1-az4)
Date created
April 7, 2025, 21:25 (UTC+02:00)
Load balancer ARN
arn:aws:elasticloadbalancing:us-east-1:915671308422:loadbalancer/app/public-alb/bc19bb033e96855b
DNS name Info
public-alb-1673591382.us-east-1.elb.amazonaws.com (A Record)
Listeners and rules
Network mapping
Resource map
Security
Monitoring
Integrations
Attributes
Capacity
Tags
Resource map Info
View, explore, and troubleshoot your load balancer's architecture.
Overview
Unhealthy target map
Show resource details
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
public-tg
2 targets
2 0 0 0 0
Targets (2)
i-00b6c68362cf52de2
Port 80
Healthy
i-0ee104be42f5f5737
Port 80
Healthy


private-alb
Details
Load balancer type
Application
Status
Active
VPC
vpc-06b69f628caaa301d
Load balancer IP address type
IPv4
Scheme
Internal
Hosted zone
Z35SXDOTRQ7X7K
Availability Zones
subnet-0c3b58387b6aa38c9 us-east-1a (use1-az2)
subnet-02d7d6cd200d50768 us-east-1b (use1-az4)
Date created
April 7, 2025, 21:25 (UTC+02:00)
Load balancer ARN
arn:aws:elasticloadbalancing:us-east-1:915671308422:loadbalancer/app/private-alb/5c185767eca5d53a
DNS name Info
internal-private-alb-1580725067.us-east-1.elb.amazonaws.com (A Record)
Listeners and rules
Network mapping
Resource map
Security
Monitoring
Integrations
Attributes
Capacity
Tags
Resource map Info
View, explore, and troubleshoot your load balancer's architecture.
Overview
Unhealthy target map
Show resource details
private-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
private-tg
2 targets
2 0 0 0 0
Targets (2)
i-029e781aed4b498f3
Port 80
Healthy
i-0332d191f2483b1a6
Port 80
Healthy


aws
Search
[Alt+S]
United States (N. Virginia)
voclabs/user3898174=Basel_Ashraf_Aboelyazeed_Abdelkader @ 915...
DynamoDB > Tables > terraform-locks
DynamoDB
Dashboard
Tables
Explore items
PartiQL editor
Backups
Exports to S3
Imports from S3
Integrations New
Reserved capacity
Settings
DAX
Clusters
Subnet groups
Parameter groups
Events
Tables (1)
Any tag key
Any tag value
Find tables
1
terraform-locks
terraform-locks
Actions
Explore table items
Settings
Indexes
Monitor
Global tables
Backups
Exports and streams
Perm
Protect your DynamoDB table from accidental writes and deletes
When you turn on point-in-time recovery (PITR), DynamoDB backs up your table data automatically so that you can restore to any given second in the preceding 1 to 35 days. Additional charges apply. Learn more
Edit PITR
General information Info
Get live item count
Partition key
LockID (String)
Sort key
-
Capacity mode
Provisioned
Table status
Active
Alarms
No active alarms
Point-in-time recovery (PITR) Info
Off
Item count
0
Table size
0 bytes
Average item size
0 bytes
Resource-based policy Info
Not active
Amazon Resource Name (ARN)
arn:aws:dynamodb:us-east-1:915671308422:table/terraform-locks


main.tf ec2 M
all-ips.txt U
variables.tf
all-ips.txt
1 public-ip1 54.161.200.70
2 public-ip2 44.220.136.100
3 private-ip1 10.0.1.185
4 private-ip2 10.0.3.103
5


Not secure | public-alb-1673591382.us-east-1.elb.amazonaws.com
Welcome to nginx!
If you see this page, the nginx web server is successfully installed and working. Further configuration is required.
For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.
Thank you for using nginx.