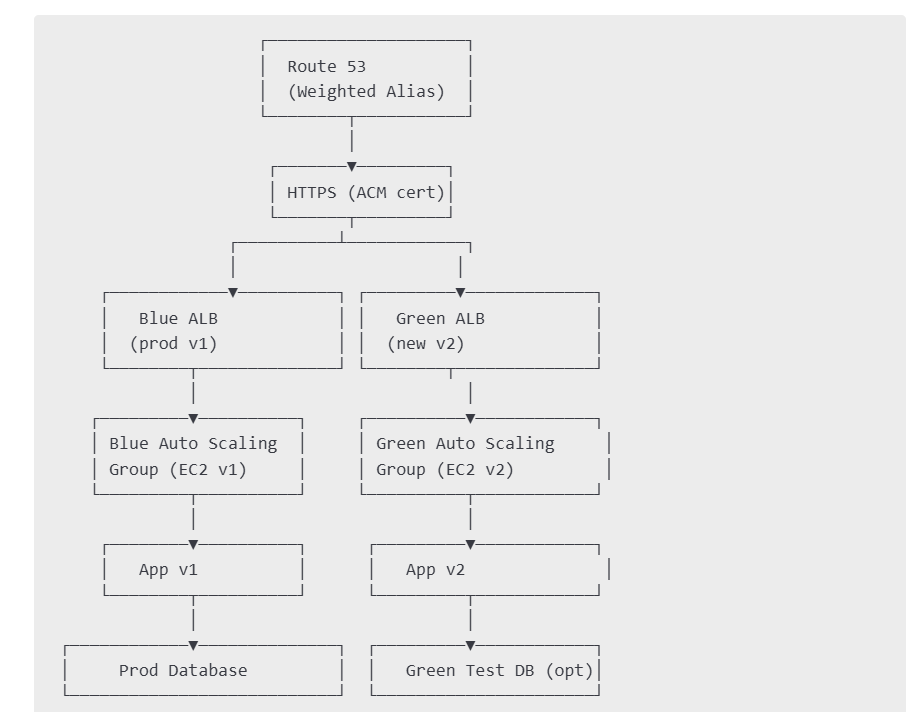
**Case Study-2:**

***Problem Statement:*** *You have been asked to set up a Blue/Green (B/G) solution for a mission-critical web application running on AWS. It should consist of* ***two separate environments****. The blue environment contains Amazon EC2 instances in an Auto Scaling group that runs the current production version of the application. The green environment contains EC2 instances in another Auto Scaling group that runs the new version of the application. Each Auto Scaling group is behind its own Application Load Balancer (ALB), so you can configure* ***two alias records as endpoints*** *in Amazon Route 53 and use a* ***routing policy*** *to gradually shift traffic from the ALB for the blue environment to the ALB for the green environment. Route 53 endpoint must be accessible from the internet through a domain name and* ***HTTPS protocol only****. Setup this combination to achieve B/G deployment and explain how different* ***backend database endpoints*** *will be handled in this situation. Use proper tool to build an infrastructure as code (IaC) solution.*

**Implementation Diagram:**



**Components to Define**

1. **VPC, Subnets, IGW, Route Tables**
2. **Security Groups**
3. **ASGs**
4. **Application Load Balancers (2)**
5. **ACM Certificate (HTTPS)**
6. **Route 53 Hosted Zone + Alias Records**

**\*Created the resources using terraform. Here is the screenshot of the folder structure of terraform and I have used modules for VPC, Route53, Autoscaling, ACM, ALB.**

**folder structure**

**blue-green-deployment/**

**│**

**├── main.tf # Root module entrypoint**

**├── variables.tf # Input variables**

**├── outputs.tf # Global outputs**

**├── terraform.tfvars # Actual values for variables**

**├── terraform.tfstate # Terraform state file**

**│**

**├── modules/ # All reusable Terraform modules**

**│ ├── acm/ # ACM certificate setup**

**│ │ ├── main.tf**

**│ │ ├── variables.tf**

**│ │ └── outputs.tf**

**│ │**

**│ ├── alb/ # Application Load Balancer module**

**│ │ ├── main.tf**

**│ │ ├── variables.tf**

**│ │ └── outputs.tf**

**│ │**

**│ ├── autoscaling/ # ASG**

**│ │ ├── main.tf**

**│ │ ├── variables.tf**

**│ │ └── outputs.tf**

**│ │**

**│ ├── route53/ # Route 53 records and policies**

**│ │ ├── main.tf**

**│ │ ├── variables.tf**

**│ │ └── outputs.tf**

**│ │**

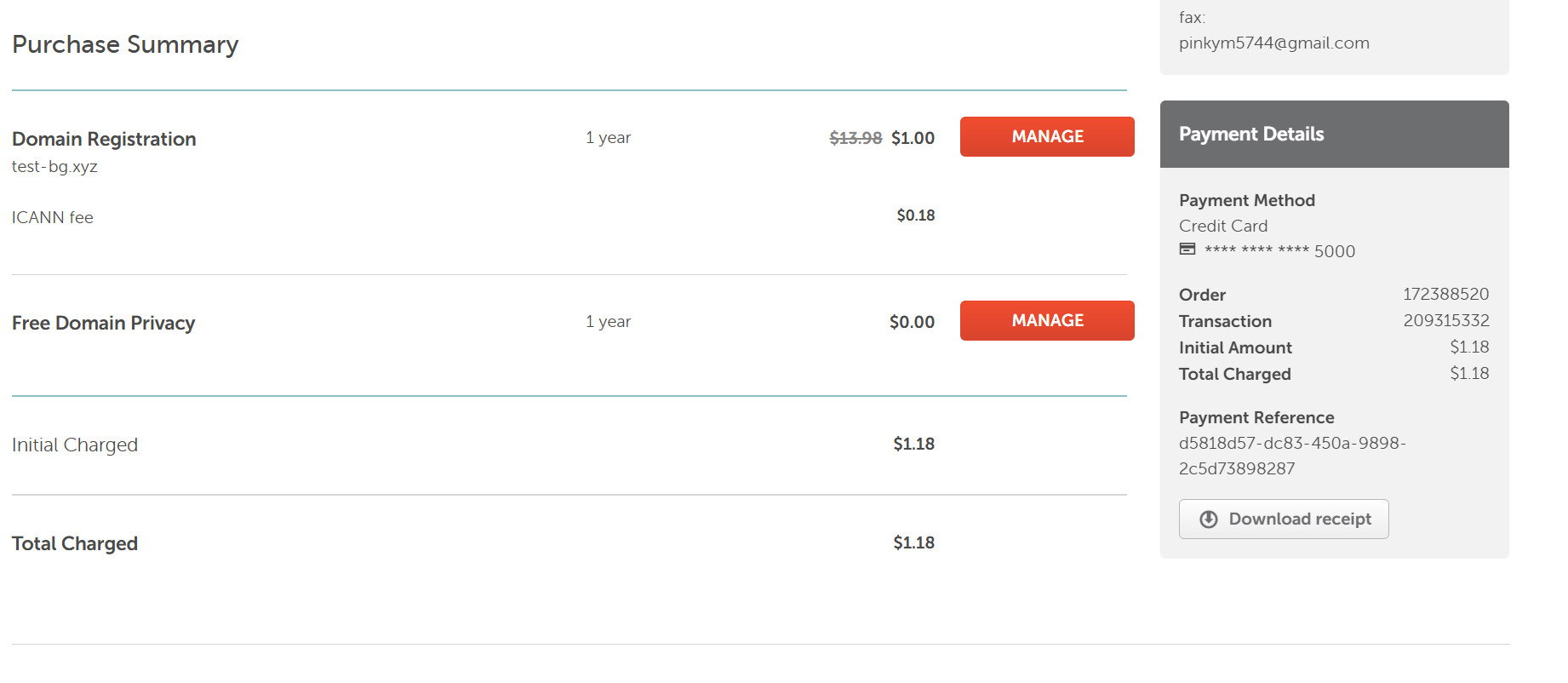
**│ └── vpc/ # VPC, subnets, gateways**

**│ ├── main.tf**

**│ ├── variables.tf**

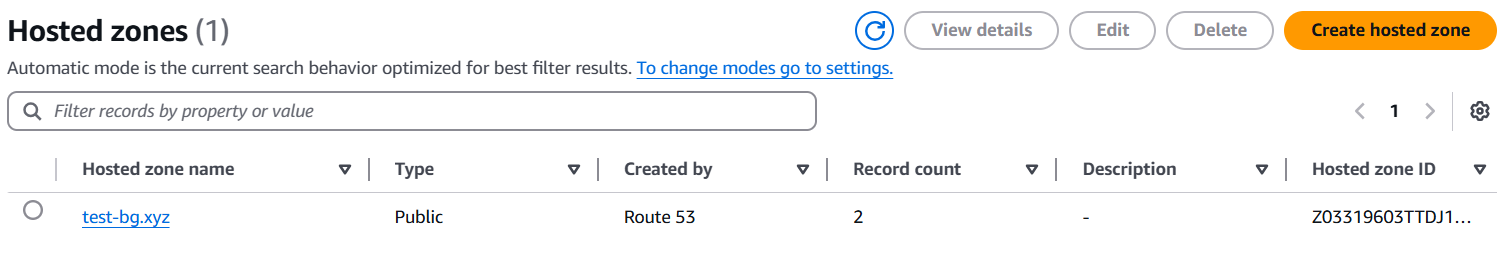
**│ └── outputs.tf**

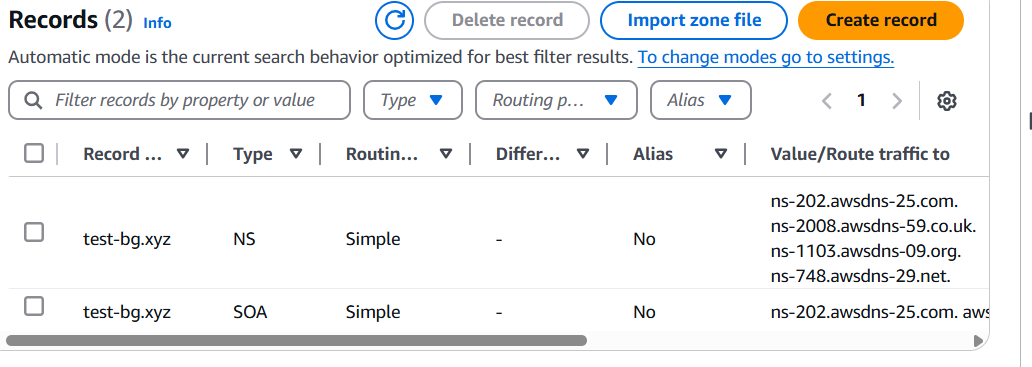
**==============================**



**Purchased domain (test-bg.xyz)from Namecheap website as seen above.**

* Domain is successfully registered via Namecheap.
* I ve assigned AWS Route 53 nameservers
* This means domain is correctly delegated to Route 53
* Created a Hosted Zone and added nameservers list in the domain

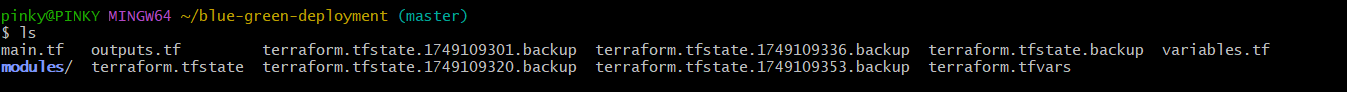


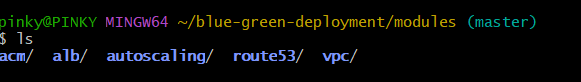


**Nameservers**

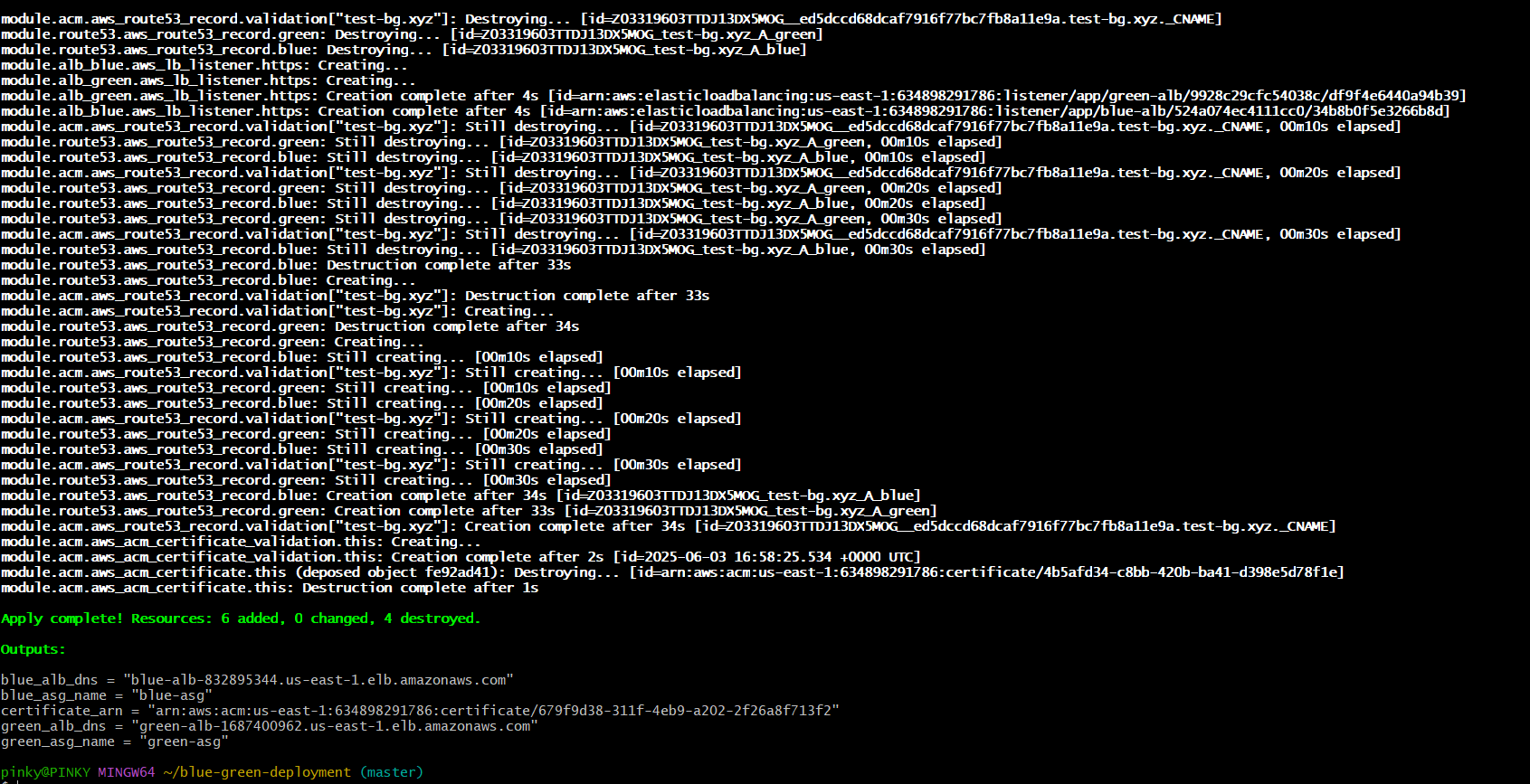
ns-202.awsdns-25.com.  
ns-2008.awsdns-59.co.uk.  
ns-1103.awsdns-09.org.  
ns-748.awsdns-29.net.

**Started creating the resources using terraform.**

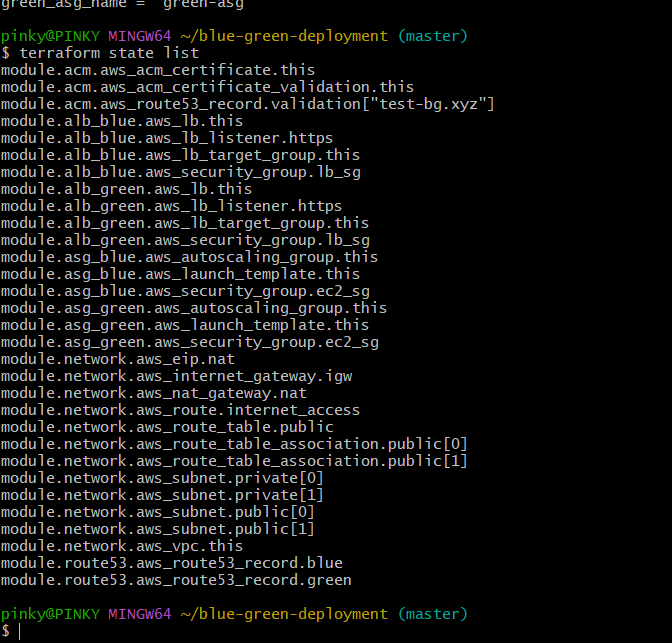


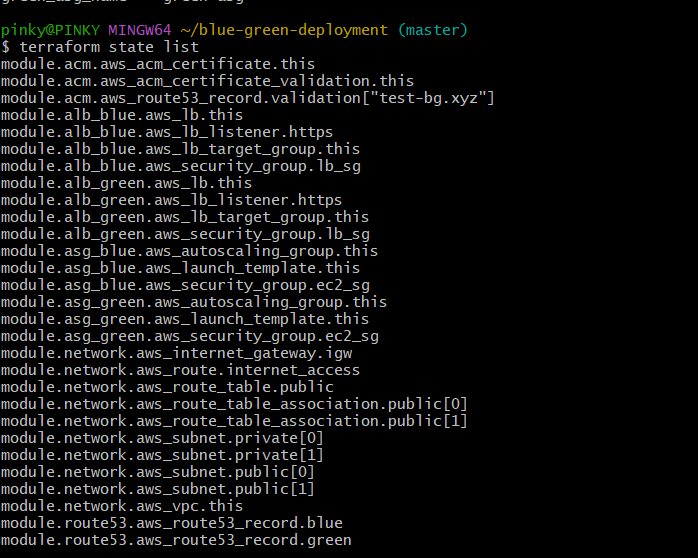


**Modules as seen above:**



**Output of terraform state list:**



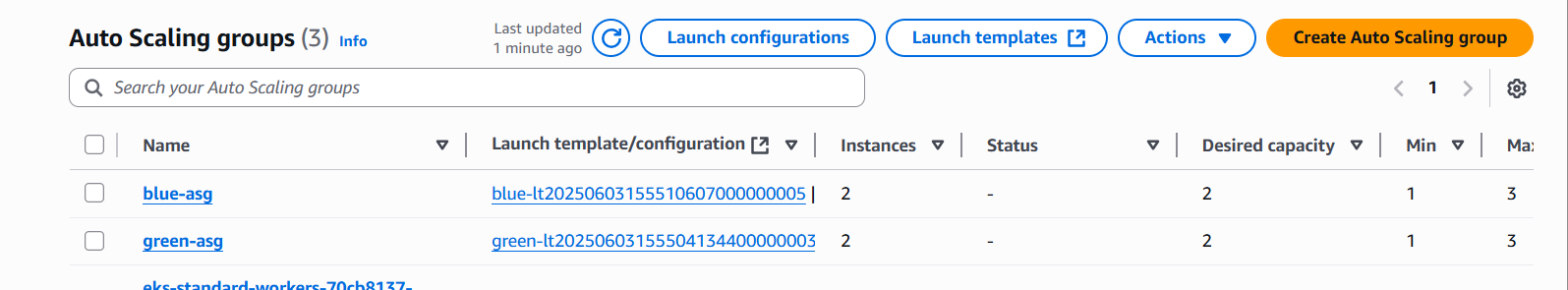


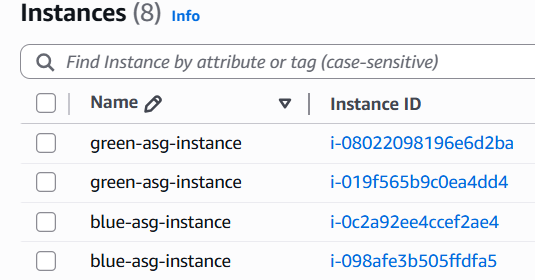
We can see all

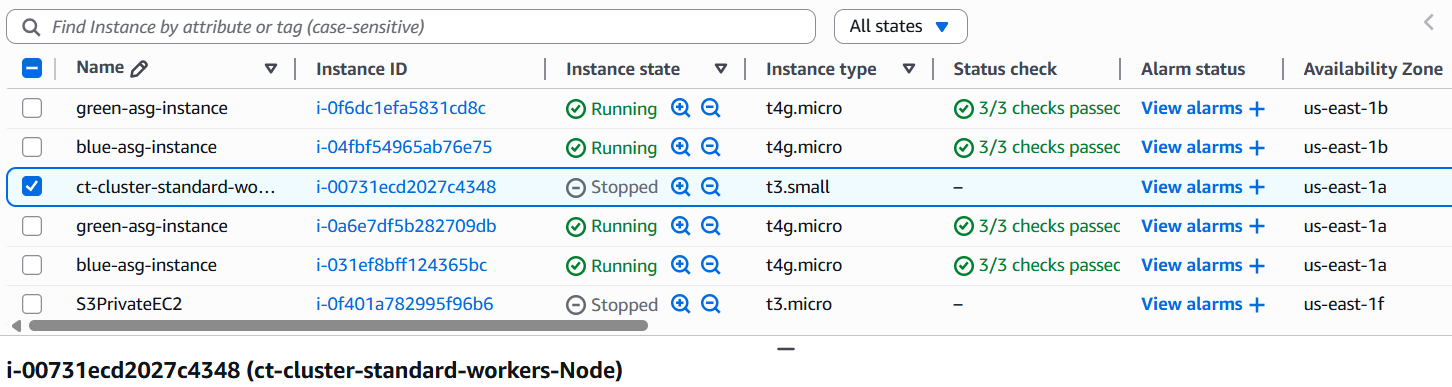
* **VPC, subnets (public/private)**
* **ALBs for blue and green**
* **Auto Scaling groups with Launch Templates**
* **Route 53 records**
* **ACM certificate**
* **Security group**

**Autoscaling group:**

* **One ASG named like blue-asg**
* **Another ASG named like green-asg**
* **each ASG has healthy EC2 instances.**

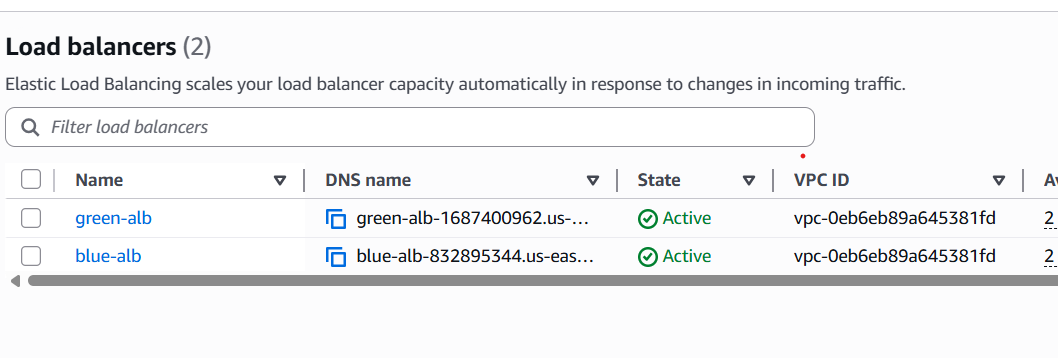
****

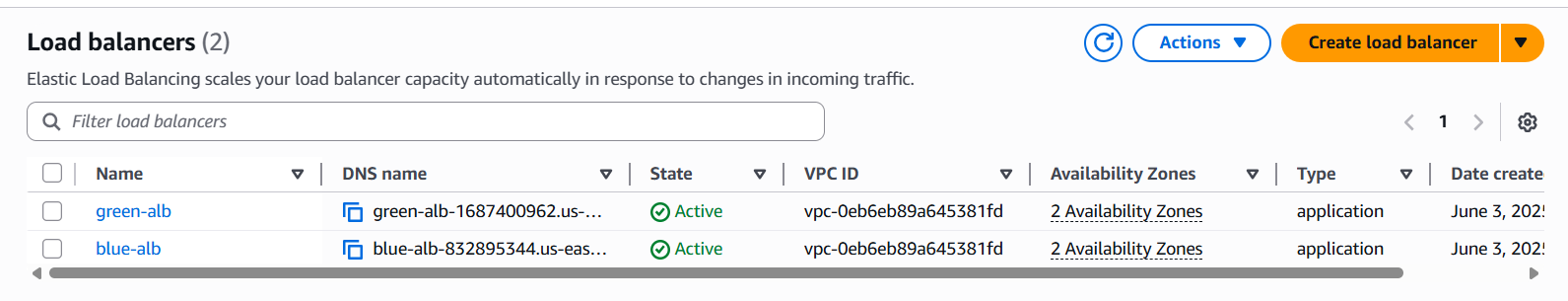
****



**Load Balancers:**

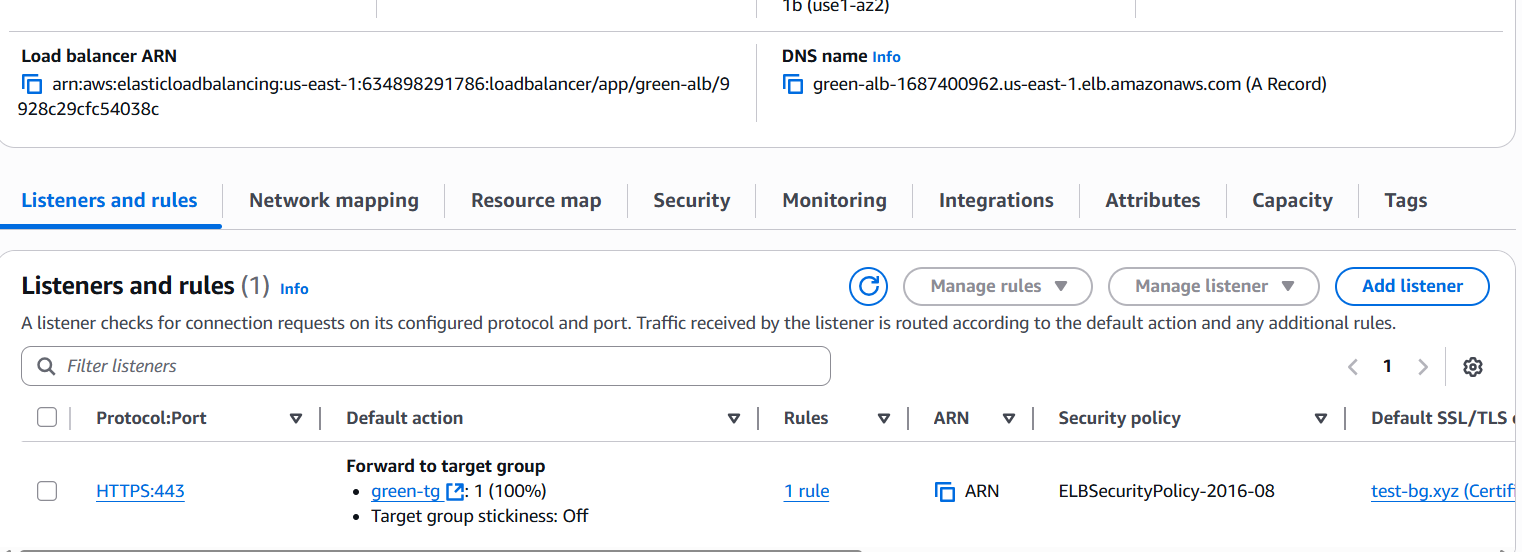
* + - **One ALB for blue**
    - **One ALB for green**

****

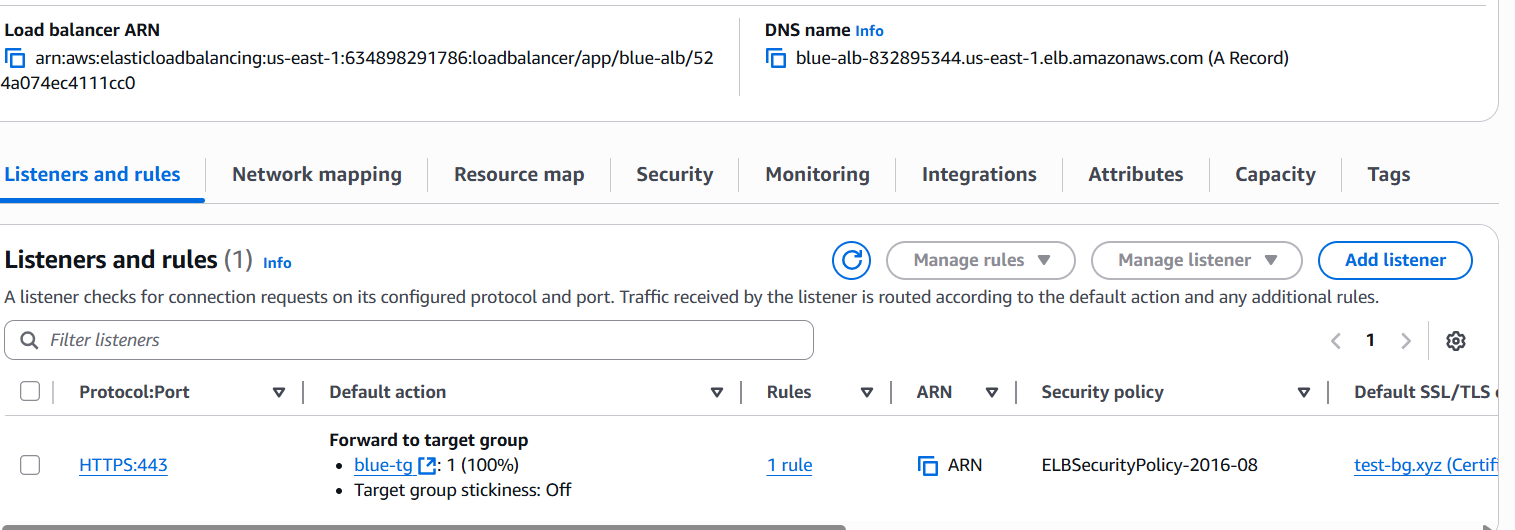
****

**Each ALB:**

* + - **Has listeners on port 443 (HTTPS).**
    - **Points to the correct target group.**
    - **Target group shows healthy instances.**
    - **This is green ALB and its listeners**

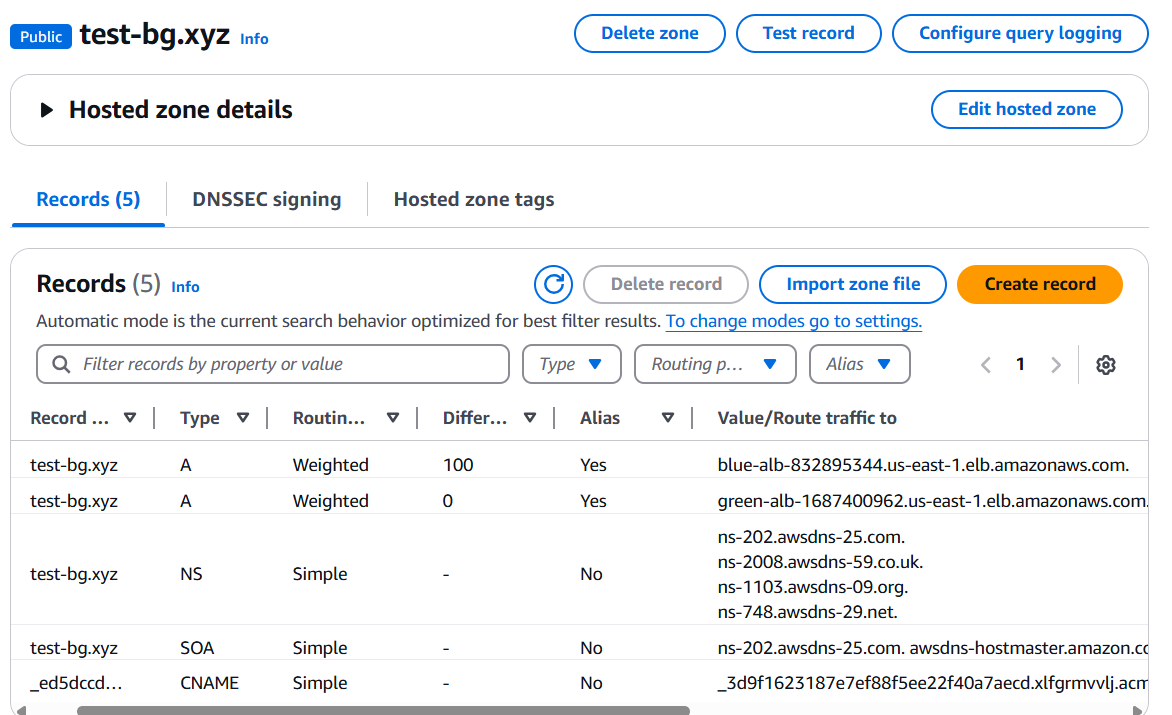
****

**This is Blue ALB and its listeners**

****

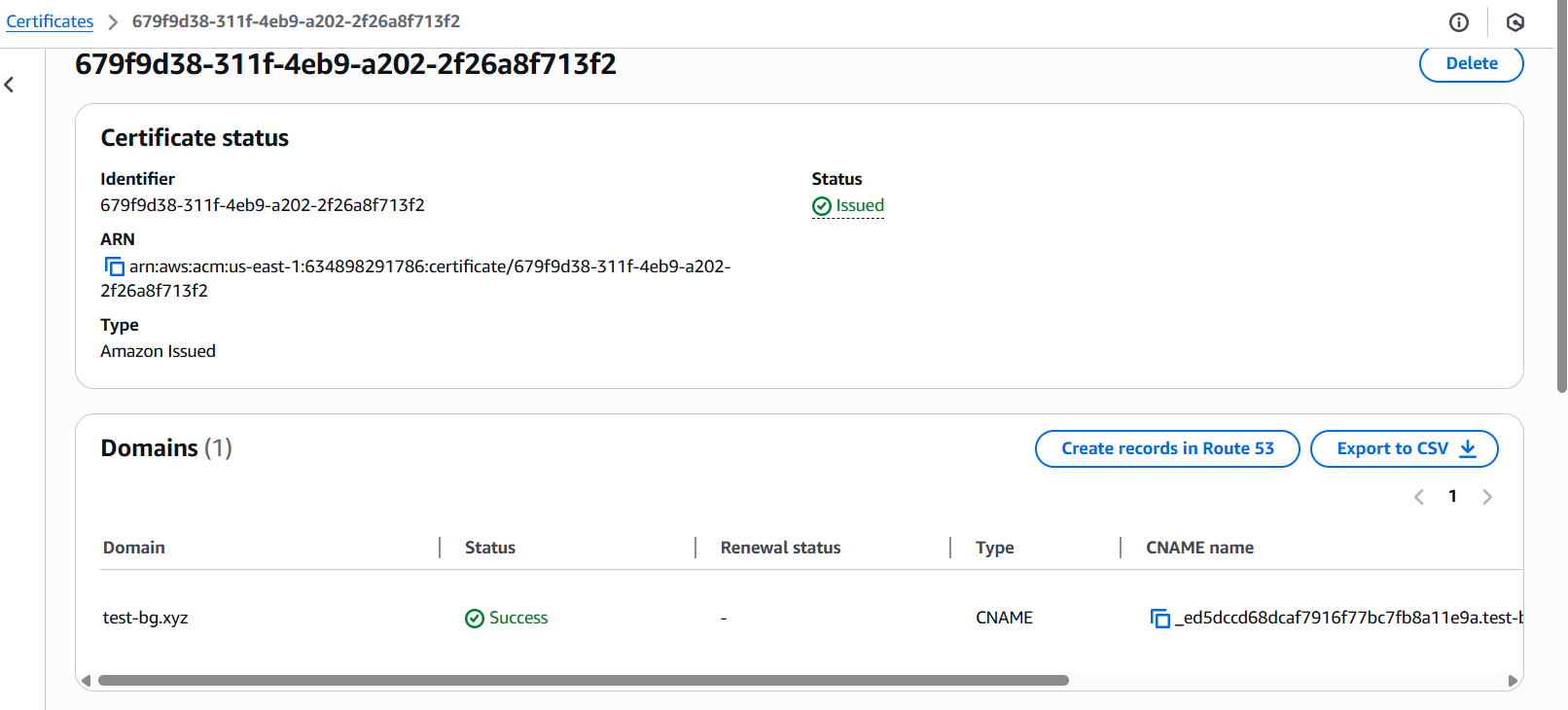
**Blue ALB DNS: blue-alb-832895344.us-east-1.elb.amazonaws.com**

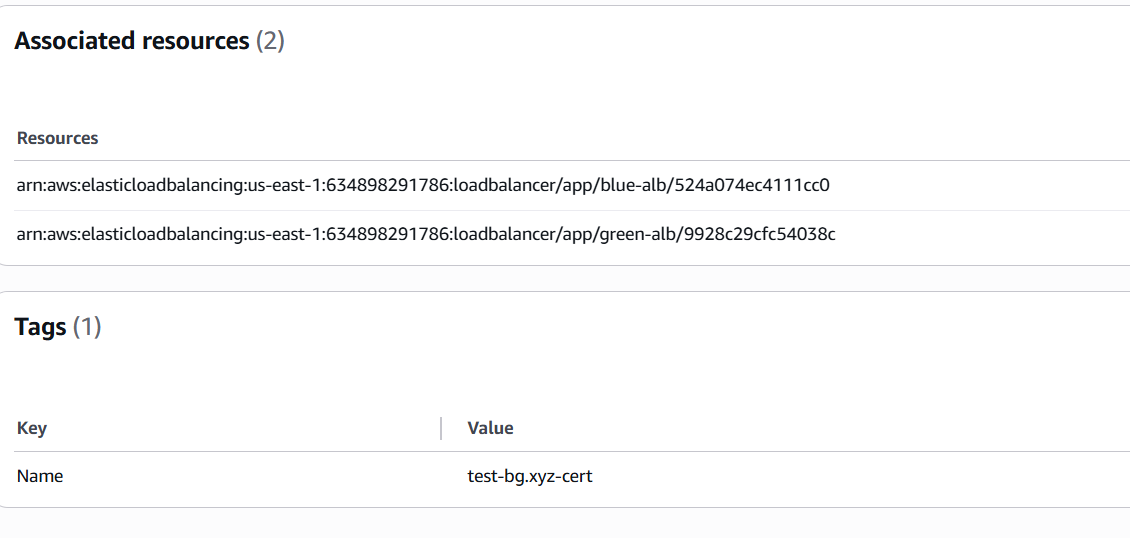
**Green ALB DNS : green-alb-1687400962.us-east-1.elb.amazonaws.com**

****

**ACM : issued status**

**Check the certificate and its associated resources.**

****

****

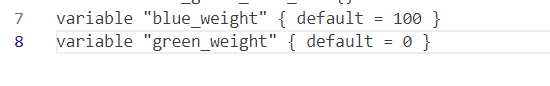
**Implemented weighted routing policy;**

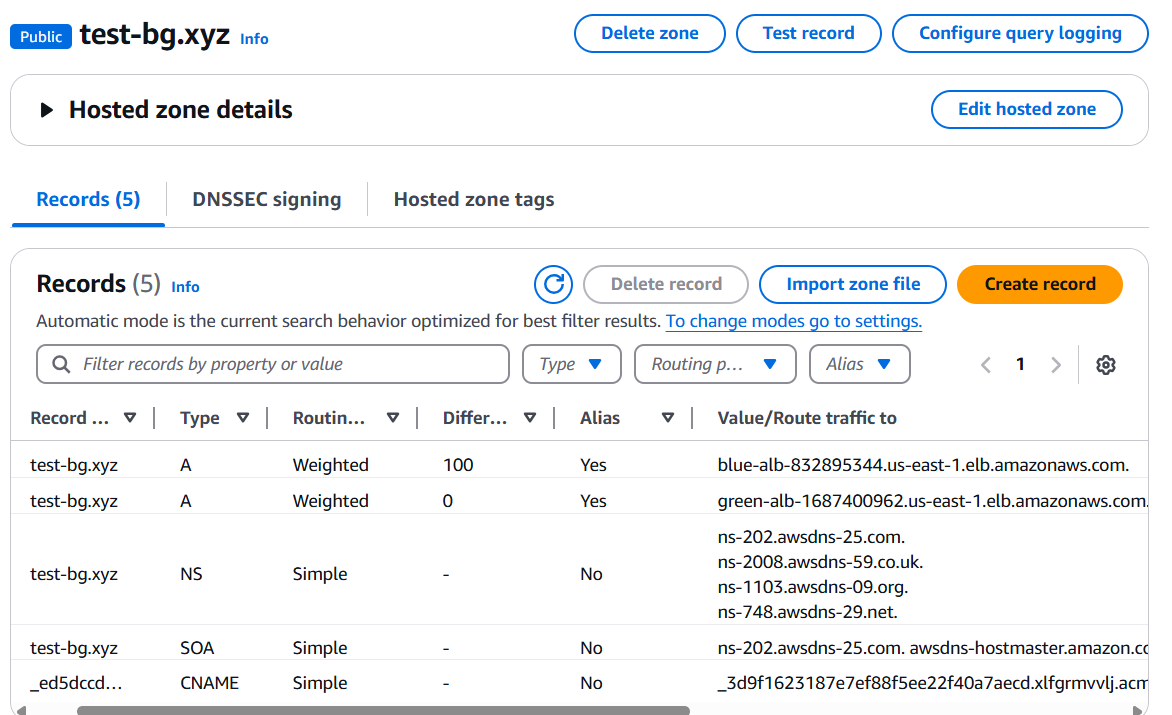
**At First;**

**100% traffic route to Blue-ALB**

**0% traffic route to Green ALB:**

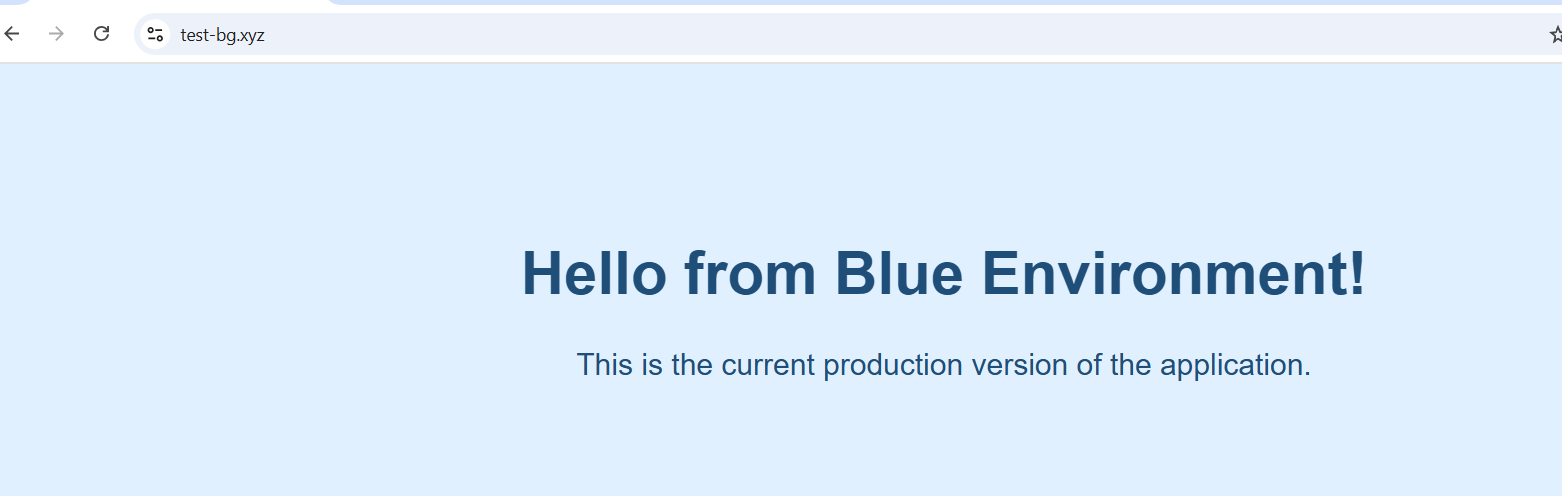
**Here is the weight configuration done in the terraform variables.**

****

****

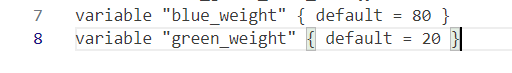
* **So currently, all traffic is going to Blue ALB only.**
* **Only Blue environment is currently receiving traffic**

1. **Open browser: https://test-bg.xyz → Should load app from Blue ALB.**

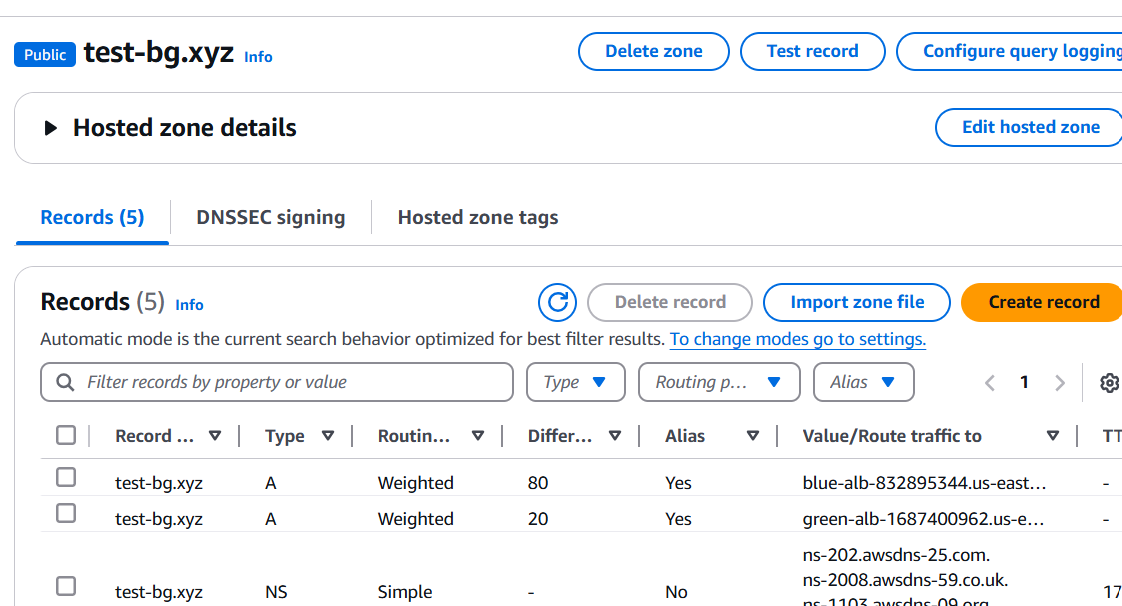
****

**Then ,Change Route 53 weight using terraform**

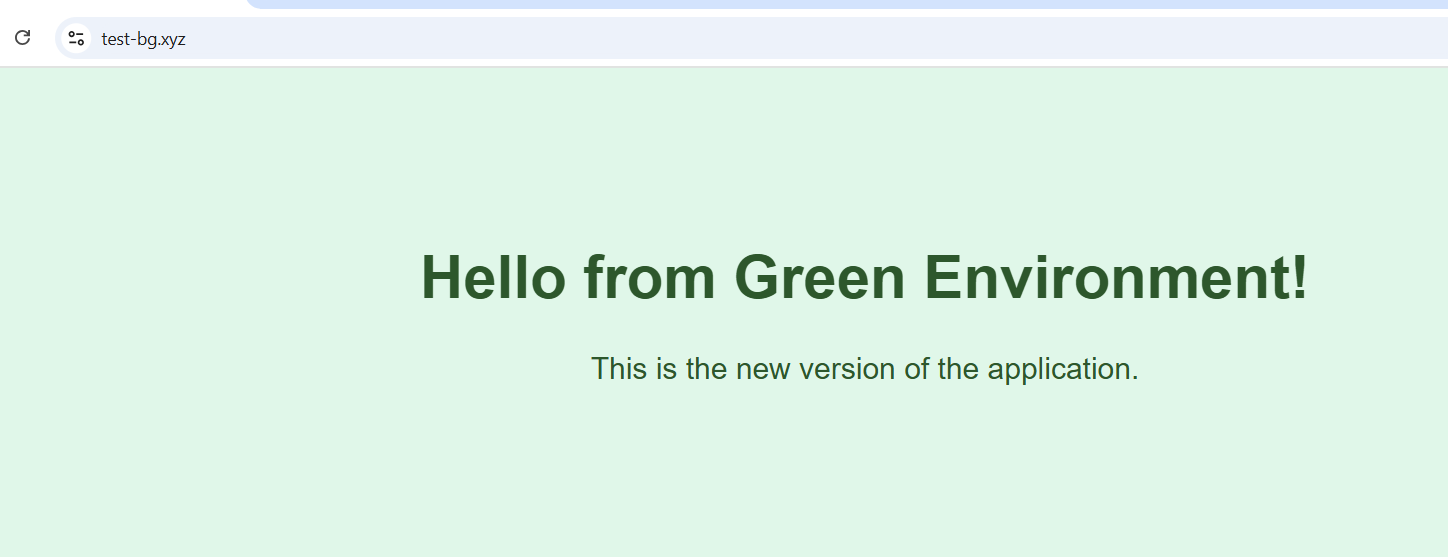
**Set Blue to 80, Green to 20 → Run terraform apply.**

****

**After terraform apply:**

****

**After terraform apply:**

****

**AFTER CHANGING THE WEIGHT TO 20% FOR Green ALB, able to see the blue traffic.**