Deploying VPC, EC2, ALB, and Auto Scaling Group using Terraform

Overview

This document provides a step-by-step guide to deploying a cloud infrastructure setup using Terraform. The setup includes:

- A Virtual Private Cloud (VPC)
- An Elastic Load Balancer (ALB)
- An Auto Scaling Group (ASG) with EC2 instances

Terraform is used to automate the infrastructure provisioning process.

Prerequisites

Before proceeding, ensure the following requirements are met:

- Terraform installed on your local machine
- AWS CLI configured with appropriate credentials
- A working AWS account with permissions to create VPCs, EC2 instances, ALBs, and ASGs

Architecture Overview

The infrastructure consists of:

1. A VPC with public and private subnets
2. An Internet Gateway to allow external access
3. A Load Balancer for distributing traffic across EC2 instances
4. An Auto Scaling Group (ASG) to manage instance scaling
5. EC2 instances running Apache with a simple HTML page

Terraform File Structure

The project consists of the following Terraform files:

- main.tf - Defines the primary infrastructure resources
- variables.tf - Contains input variables for customization
- output.tf - Specifies output values for easy reference
- alb.tf - Configures the Application Load Balancer
- ec2.tf - Defines EC2 instance configurations

- userdata.sh - A script to install and configure Apache on EC2 instances

Deployment Steps

Step 1: Initialize Terraform

Navigate to the directory containing Terraform files and initialize Terraform:

terraform init

```
PS I:\Devops Projects\Project-x> terraform init
Initializing the backend...
Initializing modules...
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.91.0

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.
PS I:\Devops Projects\Project-x> terraform plan
```

Step 2: Plan the Infrastructure

Review the resources that will be created:

terraform plan

Step 3: Apply the Configuration

Deploy the infrastructure:

terraform apply -auto-approve

```
      }
    + tags_all                              = {
        + "Environment" = "production"
        + "Name"        = "stylish-website-vpc"
        + "Project"     = "stylish-website"
        + "Terraform"   = "true"
      }
  }

Plan: 38 to add, 0 to change, 0 to destroy.

Changes to Outputs:
  + alb_dns_name = (known after apply)

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply" now.
PS I:\Devops Projects\Project-x> terraform apply -auto-approve
```

Step 4: Retrieve Outputs

After deployment, retrieve essential output values such as Load Balancer DNS:

terraform output

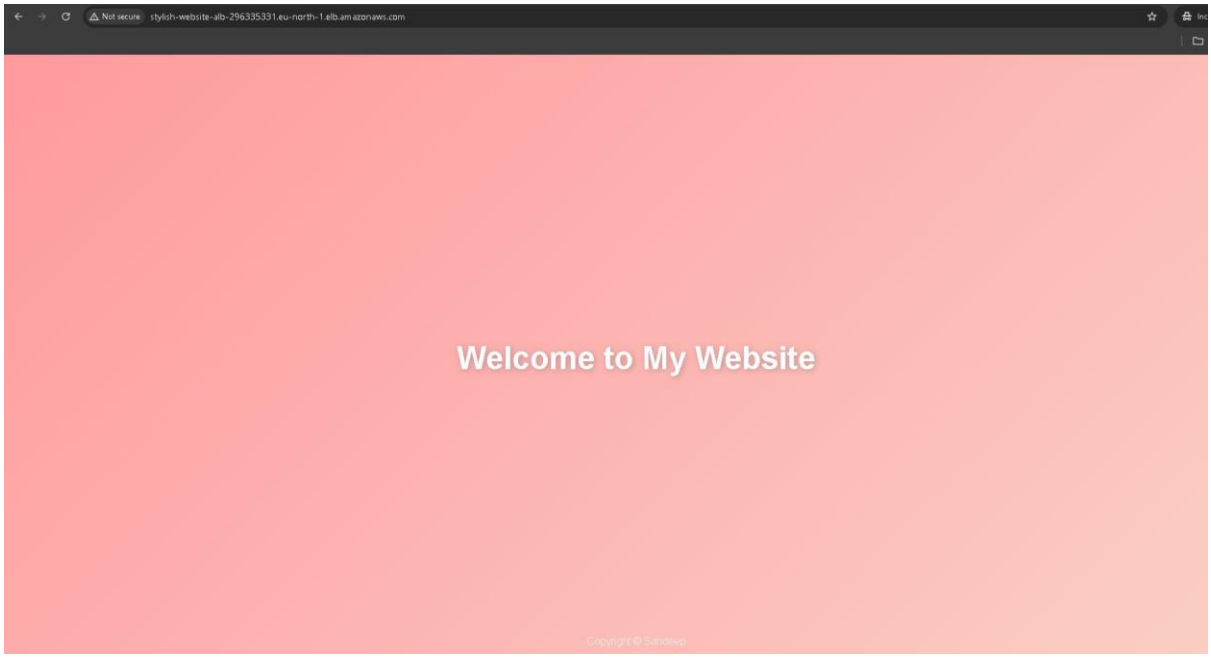
```
Outputs:

alb_dns_name = "stylish-website-alb-296335331.eu-north-1.elb.amazonaws.com"
PS I:\Devops Projects\Project-x>
```

Step 5: Verify the Deployment

- Navigate to the Load Balancer URL from the Terraform output.
- The webpage should display "Welcome to My Website."

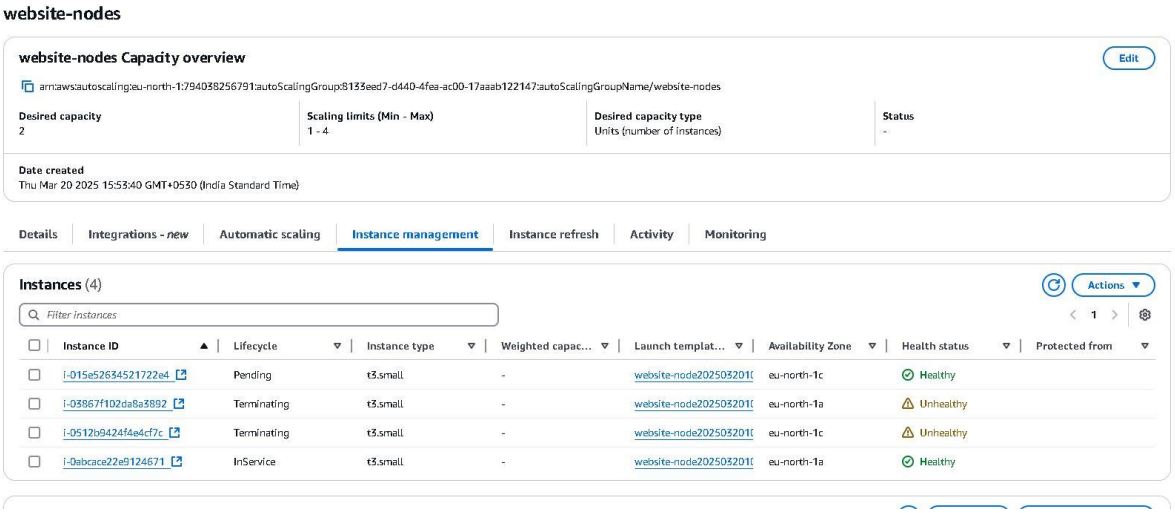
Access the website using the ALB DNS

http://stylish-website-alb-296335331.eu-north-1.elb.amazonaws.com

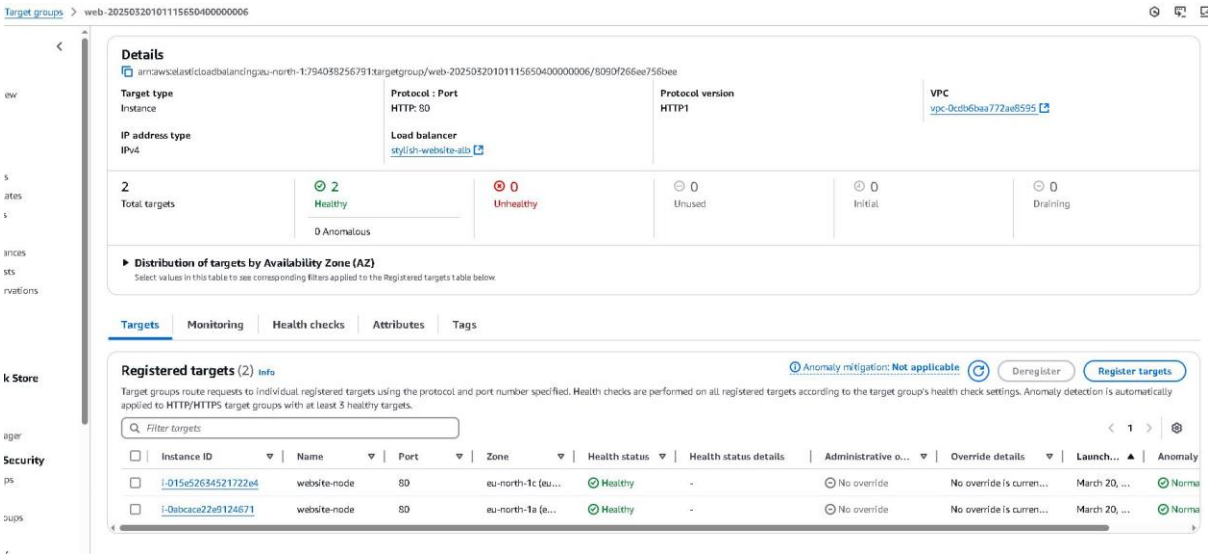


Step 6: Auto Scaling Group Validation

- Delete two instances from the Auto Scaling Group (ASG), and verify that it automatically spins up two new instances.



Ensure that the Target Groups update dynamically with the newly created instances.



Cleanup

To destroy the infrastructure when no longer needed:

terraform destroy -auto-approve

Conclusion

This guide helps deploy a scalable, highly available setup using Terraform. It ensures efficient traffic distribution using ALB and dynamic resource allocation through ASG.