# NGINX on EC2 with Terraform

## Introduction

This project provisions a simple infrastructure on AWS using Terraform. It deploys a single Amazon EC2 instance running NGINX on port 80. The Terraform configuration also sets up a remote backend using Amazon S3 with Object Lock enabled.

## Architecture Overview

The setup consists of the following components:

- A new VPC with a public subnet and internet gateway.  
- A security group that allows inbound HTTP (port 80) traffic.  
- An Amazon EC2 instance running Amazon Linux 2023 with NGINX installed via user data.  
- Terraform state stored remotely in an S3 bucket with Object Lock enabled.

## Deployment Steps

1. Create the S3 bucket in your AWS account with a unique name.  
2. Configure the Terraform backend to use the S3   
3. Deploy the main stack, which provisions the networking components, security group, and EC2 instance with NGINX installed.  
4. Access the deployed NGINX server over HTTP using the public IP or DNS output by Terraform.

## Recommendations

The following recommendations should be considered for production deployments:

1. The current setup is a proof of concept with a basic NGINX server running on port 80.

2. No SSL certificates are attached, so all traffic is only available on HTTP port 80.

3. SSH port access is not enabled. Instead, it is recommended to use the AWS Systems Manager (SSM) Agent, which is pre-installed on Amazon Linux. By attaching an SSM Instance Role, administrators can securely manage instances without exposing SSH.

4. In production, instances should not be exposed directly to the internet. Only the load balancer or API Gateway (depending on the application infra) should be public facing.

5. Security groups should allow inbound traffic only from the load balancer to the instances. The load balancer's security group should allow public access with proper HTTP to HTTPS redirection configured as rules in LB.

6. SSL certificates should be handled at the Application Load Balancer (ALB) level. AWS Certificate Manager (ACM) can be used to provision or import certificates to serve traffic securely over HTTPS.

7. Instances should be deployed in private subnets. The only allowed inbound port for instance SG should be HTTP (80) on the ALB security group, since SSL termination occurs at the ALB level.

8. All these best practices can be implemented using Terraform. However, this project was intentionally kept minimalistic to demonstrate a basic setup without modules.