Terraform AWS Web Application Stack: Project Overview

This repository contains Terraform code to provision a scalable, highly available, and secure web application infrastructure on AWS. The entire stack is defined as code, promoting consistency, repeatability, and version control for your infrastructure.

Architecture

The core architecture consists of the following components, deployed across multiple Availability Zones for resilience:

1. **VPC**: A custom Virtual Private Cloud (VPC) with public and private subnets provides a secure, isolated network environment.
2. **Application Load Balancer (ALB)**: An internet-facing ALB is deployed in the public subnets to receive incoming traffic and distribute it to the application instances.
3. **EC2 Auto Scaling Group (ASG)**: Application instances are managed by an Auto Scaling Group within the **public** subnets. This allows the application to automatically scale in or out based on CPU load and enables direct access for maintenance via SSH.
4. **EC2 Launch Template**: The ASG uses a launch template to define instance configuration. This template is configured to use the latest Amazon Linux 2023 AMI and a dynamically generated EC2 Key Pair for SSH access.
5. **Security**: A multi-layered security approach is used:
   - The ALB's security group allows web traffic from the internet.
   - The EC2 instances' security group allows traffic from the ALB and **SSH traffic from your local machine's IP address**.

Key Features

- **High Availability**: Resources are deployed across at least two Availability Zones.
- **Scalability**: CPU-based auto-scaling policies automatically adjust capacity to meet demand.
- **SSH Access**: Secure remote access to instances is enabled via a generated SSH key pair. Ingress rules are automatically configured to allow access only from the IP address of the machine running Terraform.
- **Modularity**: The project is broken down into reusable modules (`vpc`, `alb`, `launch_template`, `asg`) for clarity and maintainability.
- **Automation**: A set of helper scripts (`terraform_init.sh`, `terraform_plan.sh`, `terraform_apply.sh`, `terraform_destroy.s` streamline the deployment lifecycle for different environments.

Getting Started

Prerequisites

- Terraform v1.0+
- AWS CLI with configured credentials
- An SSH client

Recommended Workflow

The included helper scripts are the recommended way to interact with this project.

1. **Make scripts executable:**

   ```
   chmod +x *.sh
   ```

2. **Initialize Terraform for an environment:** (This configures the backend and downloads providers).

   ```
   ./terraform_init.sh dev
   ```

3. **Plan changes:**

   ```
   ./terraform_plan.sh dev
   ```

4. **Apply changes:**

   ```
   ./terraform_apply.sh dev
   ```

5. **Destroy infrastructure:**

   ```
   ./terraform_destroy.sh dev
   ```

For detailed documentation on each module, please see the `README.md` file located within the respective module's directory (e.g., `./terraform/modules/basic_vpc/README.md`).