Infrastructure Overview

This documentation provides a comprehensive guide for deploying a web application infrastructure on Amazon Web Services (AWS) using Terraform as the Infrastructure as Code (IaC) tool. The infrastructure setup is designed to run a web server with proper networking, security, and scalability considerations. (Note: All text in red represents commands.)

This deployment creates:

- EC2 instance (t2.medium)
  - 2 vCPUs
  - 4GB RAM
  - 30GB GP3 SSD storage
  - Ubuntu 24.04 LTS
- VPC with 10.0.0.0/16 CIDR
- Public subnet with 10.0.1.0/24 CIDR
- Apache web server installed via user data script

Setup

1. AWS Account Setup

1. **Create AWS Account:**
   - Go to aws.amazon.com
   - Click "Create an AWS Account"
   - Provide email and AWS account name
   - Complete verification process
   - Add payment method
2. **Create IAM User:**
   - Login to AWS Console
   - Search for "IAM"
   - Click "Users" → "Create User"

- Username: terraform-deployer
- Select "Programmatic access"
- Attach policy: "AdministratorAccess" (for testing; restrict for production)
- Save the Access Key ID and Secret Access Key securely

3. Create EC2 Key Pair:

- Go to EC2 Dashboard
- Click "Key Pairs" under "Network & Security"
- Click "Create Key Pair"
- Name it "task1" (as referenced in compute.tf)
- Select .pem format
- Download and secure the key file:

- `chmod 400 task1.pem`
- `mv task1.pem ~/.ssh/`

4. AWS CLI Configuration

1. Configure AWS credentials:
   - **aws configure**

Enter the following when prompted:

- AWS Access Key ID: [Your Access Key]
- AWS Secret Access Key: [Your Secret Key]
- Default region name: region
- Default output format: json

Infrastructure Details

VPC Configuration

- Main VPC (10.0.0.0/16):
  - Enables DNS hostnames
  - Enables DNS support
  - Spans entire region

Subnet Layout

- Public Subnet (10.0.1.0/24):
  - Located in ap-southeast-1a
  - Auto-assigns public IPs

    - Connected to Internet Gateway

Security Configuration

- Inbound Rules:
    - Port 80 (HTTP): 0.0.0.0/0
    - Port 22 (SSH): 0.0.0.0/0 (restrict this in production)
- Outbound Rules:
    - All traffic allowed: 0.0.0.0/0

EC2 Instance

- AMI: Ubuntu 24.04 LTS (ami-047126e50991d067b)
- Type: t2.medium
    - Suitable for development/testing
    - 2 vCPUs, 4GB RAM
- Storage: 30GB GP3 encrypted EBS
- User Data:
    - Installs Apache2
    - Enables Apache2 service
    - Creates basic index.html

Deployment Process

1. Code Organization

Current file structure:

```
./
├── backend.tf    // State configuration
├── compute.tf    // EC2 configuration
├── network.tf    // VPC and subnet configuration
├── output.tf     // Output values
├── providers.tf  // AWS provider configuration
├── security.tf   // Security group rules
└── variables.tf  // Variable definitions
```

2. Steps for deployment:

1. **Update system packages**:

   ```
   sudo apt update && sudo apt upgrade -y
   ```

2. **Install required packages:**

   sudo apt install -y software-properties-common gnupg2 curl

3. **Install Terraform**:
   - `curl -fsSL https://apt.releases.hashicorp.com/gpg | sudo apt-key add -`
   - `sudo apt-add-repository "deb [arch=amd64]https://apt.releases.hashicorp.com $(lsb_release -cs) main"`
   - `sudo apt update && sudo apt install terraform -y`
   - `terraform --version`

```
# Initialize Terraform and download providers
```

- `terraform init`

```
# Check code formatting
```

- `terraform fmt`

```
# Validate configuration
```

- `terraform validate`

```
# Preview changes
```

- `terraform plan`

```
# Apply changes (enter 'yes' when prompted)
```

- `terraform apply`