Terraform AWS ECS Infrastructure

Contents

1 Introduction

This document outlines the key components of an AWS-based architecture, explaining their purpose and providing best practices for implementation in your project. The architecture leverages a Virtual Private Cloud (VPC) with subnets, networking components, and AWS services like ECS, RDS, and ALB to create a secure, scalable, and resilient system.

2 Key Components Explained

2.1 Virtual Private Cloud (VPC)

- **Purpose**: Provides an isolated network within the AWS cloud where all resources reside.
- **CIDR Block**: Use `10.0.0.0/16` to allocate sufficient IP addresses for future scalability.

2.2 Subnets

Subnets segment the VPC to enhance security and organization.

2.2.1 Public Subnets (`10.0.1.0/24`)

- **Purpose**: Host resources requiring direct internet access, such as Application Load Balancers (ALB) or Bastion Hosts.
- **Key**: Associated with a route table directing traffic to an Internet Gateway (IGW).
- **Components**: Application Load Balancer (ALB).

2.2.2 Private Subnets (`10.0.2.0/24`)

- **Purpose**: Host resources like backend ECS containers that need outbound internet access (e.g., for pulling Docker images or sending logs) but should not be directly accessible from the internet.
- **Key**: Associated with a route table routing internet-bound traffic through a NAT Gateway.
- **Components**: Backend ECS Containers.

2.2.3 Database Subnets (`10.0.3.0/24`)

- **Purpose**: Dedicated subnets for RDS database instances, requiring a DB Subnet Group spanning at least two Availability Zones (AZs) for high availability.
- **Key**: Fully private with no direct internet access for enhanced security.
- **Components**: RDS Database.

2.3 Internet Gateway (IGW)

- **Purpose**: Enables resources in public subnets to connect to the internet and vice versa.
- **Association**: Directly attached to the VPC.

2.4 NAT Gateway (NAT GW)

- **Purpose**: Allows private subnet instances to initiate outbound connections (e.g., to AWS services like ECR or S3) while preventing inbound internet connections.
- **Placement**: Deployed in a public subnet.

2.5 Route Tables

- **Purpose**: Direct network traffic from subnets.
- **Public Route Table**: Routes `0.0.0.0/0` to the Internet Gateway.
- **Private Route Table**: Routes `0.0.0.0/0` to the NAT Gateway.
- **Database Route Table**: Routes local VPC traffic internally, without a `0.0.0.0/0` route to prevent internet access.

2.6 Application Load Balancer (ALB)

- **Purpose**: Distributes incoming traffic across frontend ECS containers for high availability and fault tolerance.
- **Placement**: Deployed across public subnets for internet-facing access.
- **Listeners**: Configured for ports 80 (HTTP) and 443 (HTTPS) in production.

2.7 Elastic Container Service (ECS)

2.7.1 Frontend ECS Service/Container

- **Placement**: Launched in public subnets, accessible via the ALB.
- **Role**: Serves the user interface or API gateway for external requests.

2.7.2 Backend ECS Service/Container

- **Placement**: Launched in private subnets for enhanced security.
- **Role**: Handles business logic and interacts with the database.

2.8 Relational Database Service (RDS)

- **Purpose**: Managed relational database service (e.g., PostgreSQL, MySQL).
- **Placement**: Deployed in private database subnets, with a DB Subnet Group across multiple AZs for high availability.

2.9 Security Groups (SGs)

Security Groups act as virtual firewalls controlling traffic at the instance level.

2.9.1 SG_LB (Load Balancer SG)

- **Inbound**: Allows traffic from `0.0.0.0/0` on ports 80 and 443.
- **Outbound**: Allows traffic to SG_FE on the frontend container port (e.g., 8080).

2.9.2 SG_FE (Frontend ECS SG)

- **Inbound**: Allows traffic from SG_LB on the application port (e.g., 8080 or 80).
- **Outbound**: Allows traffic to SG_BE and NAT Gateway for internet access.

2.9.3 SG_BE (Backend ECS SG)

- **Inbound**: Allows traffic from SG_FE on the application port (e.g., 8080).
- **Outbound**: Allows traffic to DB_SG on the database port (e.g., 3306 for MySQL, 5432 for PostgreSQL) and NAT Gateway.

2.9.4 DB_SG (RDS Database SG)

- **Inbound**: Allows traffic from SG_BE on the database port (e.g., 3306 or 5432).
- **Outbound**: Typically allows all outbound traffic, but can be restricted.

3 Project Best Practices

- **Multi-AZ Deployment**: Spread subnets across at least two Availability Zones for resilience. ALB and RDS automatically leverage this setup.
- **ECS Fargate vs. EC2**:
  - **Fargate**: Serverless, low-maintenance option for containers, ideal for simplified management.
  - **EC2**: Offers more control but requires managing underlying instances.
- **Service Discovery**: Use AWS Cloud Map for dynamic discovery of backend and database services, avoiding hardcoded IPs.
- **Observability**:
  - **CloudWatch Logs**: Centralize container logs.
  - **CloudWatch Metrics**: Monitor ALB, ECS, and RDS performance.
  - **AWS X-Ray**: Trace performance across services.
- **CI/CD Pipeline**: Use AWS CodePipeline, CodeBuild, and CodeDeploy to automate building, pushing, and deploying container images to ECR and ECS.

- **Infrastructure as Code (IaC)**: Define all resources (VPC, subnets, SGs, ECS, ALB, RDS) using Terraform or AWS CloudFormation for reproducibility.
- **Parameter Store/Secrets Manager**: Securely store sensitive data (e.g., database credentials) and inject into ECS containers at runtime.
- **Cost Optimization**: Consider Fargate Spot, appropriate instance types, and data transfer cost management.