Project Documentation: Sufle-Terraform-Architecture

Overview

This Terraform-managed project demonstrates a secure architecture within AWS where private EC2 instances handle application workloads. The private instances do not have direct internet access but are connected through a NAT gateway for outbound internet connectivity. A public-facing Application Load Balancer (ALB) is used to balance traffic across instances. The architecture ensures that traffic is handled securely and efficiently while maintaining high availability and scalability.

Infrastructure Components

1. **VPC (Virtual Private Cloud)**:

   The VPC is the main network where all AWS resources are hosted. It separates resources into public and private subnets.

2. **Subnets**:

   - **Public Subnet**:
     Used for resources that need direct internet access, such as the NAT gateway and the Application Load Balancer (ALB).
   - **Private Subnet**:
     Contains EC2 instances that handle application workloads. These instances do not have direct internet access for security reasons and communicate with the internet via the NAT gateway.

3. **Internet Gateway (IGW)**:

   Connects the VPC's public subnet to the internet, allowing the ALB and NAT gateway to communicate with the internet.

4. **Elastic IP (EIP)**:

   An Elastic IP address is allocated for the NAT gateway to ensure that the private subnet instances have outbound internet access.

5. **NAT Gateway**:

   The NAT gateway is created in the public subnet, providing outbound internet access to the EC2 instances in the private subnet. This is essential for updates, package installations, or other outbound connections from private EC2 instances.

6. **Route Tables**:

   - **Public Route Table**:
     Associated with the public subnet, it allows traffic from the internet to flow through the Internet Gateway and to public-facing resources.
   - **Private Route Table**:
     Associated with the private subnet, it ensures that private instances send outbound traffic through the NAT gateway for internet access without direct exposure.

7. **Security Groups**:

   - **Bastion Host Security Group**:
     Controls SSH access to the bastion host from trusted IPs.
   - **EC2 Security Group**:
     Allows traffic from the load balancer and SSH access from the bastion host.
   - **Load Balancer Security Group**:
     Manages HTTP and HTTPS traffic from the internet to the ALB.

8. **Bastion Host**:

   A publicly accessible EC2 instance used as a jump server to access private EC2 instances via SSH.

9. **Application Load Balancer (ALB)**:

   Distributes incoming traffic across the EC2 instances. The ALB is located in the public subnet and handles traffic from the internet.

10. **Target Groups**:

    Target groups are used by the ALB to route traffic to the EC2 instances in the private subnet. Each instance is registered with a target group, and the ALB forwards incoming traffic to the appropriate targets based on the load balancing rules.

11. **Listeners**:
    Listeners define how the ALB listens for incoming traffic. For example, you can configure an HTTP listener on port 80 or HTTPS on port 443 to listen for traffic and forward it to the appropriate target group

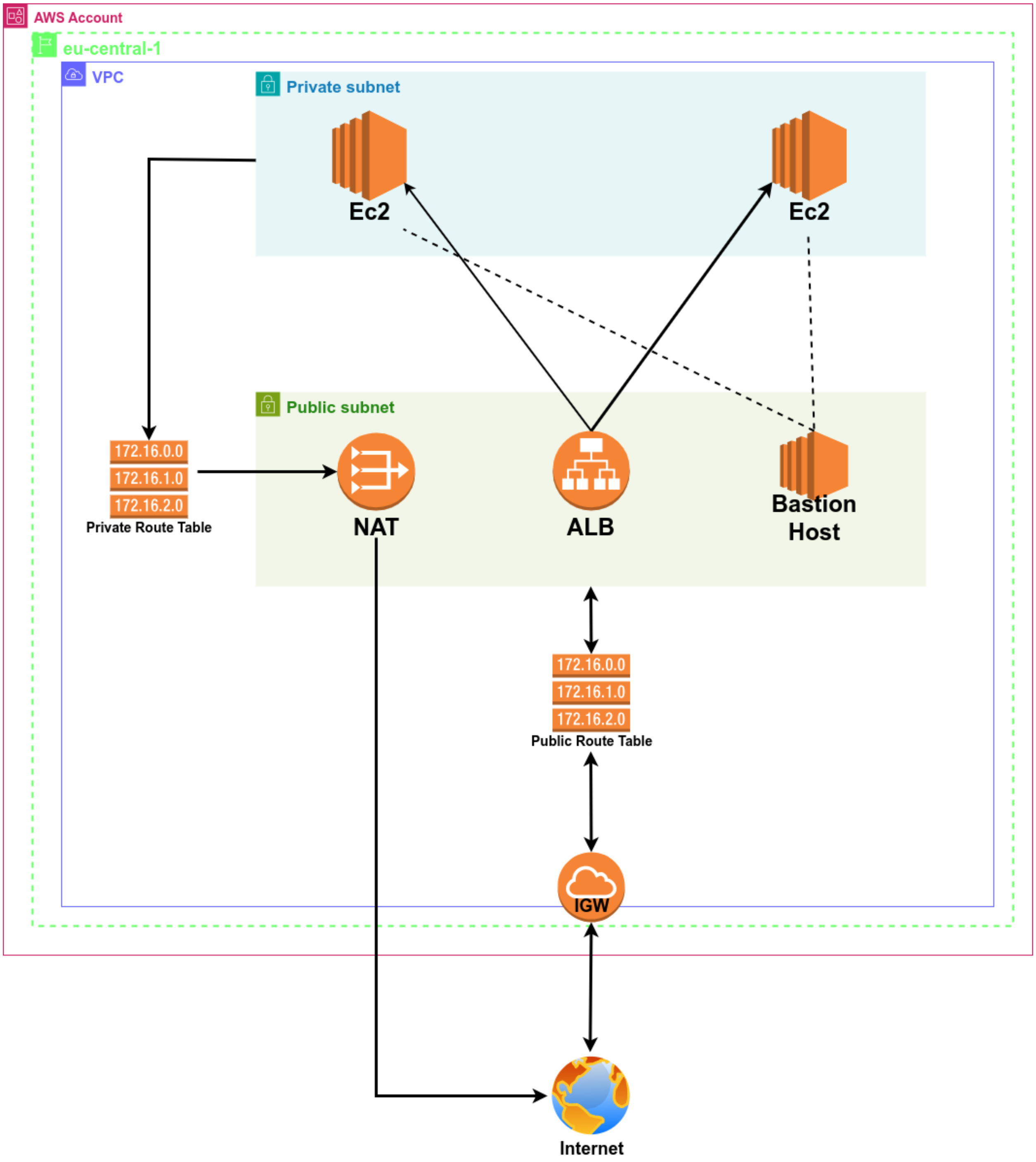
12. **Private EC2 Instances**:

    These are the instances that handle application workloads. They run in private subnets without direct internet access for security purposes but have outbound connectivity through the NAT gateway.

13. **SSM (AWS Systems Manager)**:

    This allows secure remote access to the private EC2 instances without the need for SSH or a bastion host.

Diagram model of architecture



Step-by-Step Creation Process

1. **Create a VPC**

   Start by creating a VPC to isolate your infrastructure.

2. **Set Up Public and Private Subnets**

   Public subnets are used for resources that need internet access, such as the ALB and NAT gateway. Private subnets are used for EC2 instances, which do not require direct internet access.

3. **Set Up an Internet Gateway (IGW)**

   Attach the internet gateway to the VPC to allow the ALB and NAT gateway to communicate with the internet.

4. **Set up Route Table:**

   - **Global Route Table**

     Associate this with the public subnet so that traffic from the internet can reach it

   - **Private Route Table**

     Associate this with the private subnet. Configure it so that private instances route outbound traffic through the NAT gateway for secure internet access.

5. **Create and Attach the NAT Gateway**

   The NAT gateway is deployed in the public subnet and attached to an Elastic IP to provide internet access to the EC2 instances in the private subnet.

6. **Set Up Security Groups**

   Define rules for inbound and outbound traffic, ensuring that only necessary traffic is allowed between the bastion host, EC2 instances, and the load balancer.

7. **Deploy EC2 Instances in Private Subnets**

   These instances are responsible for processing application requests. They can only be accessed from the bastion host or load balancer.

8. **Deploy the Application Load Balancer (ALB)**

   It will distribute traffic evenly across the private EC2 instances.

9. **Configure SSM for Remote Management**

Set up Systems Manager to manage EC2 instances without requiring SSH access, improving security and reducing dependency on the bastion host.

10. **Connect to ec2s with SSM and create your static web file**

    Create your static web file with NginX or a simple python web server

---

Future Enhancements

- **Add SSL to the Load Balancer**:
  Use SSL for HTTPS traffic to enhance security.

Conclusion

This architecture ensures a secure, scalable, and highly available environment for running applications on AWS. By separating the public and private subnets, using NAT for outbound access, and utilizing a load balancer for traffic distribution, it offers both security and efficiency.