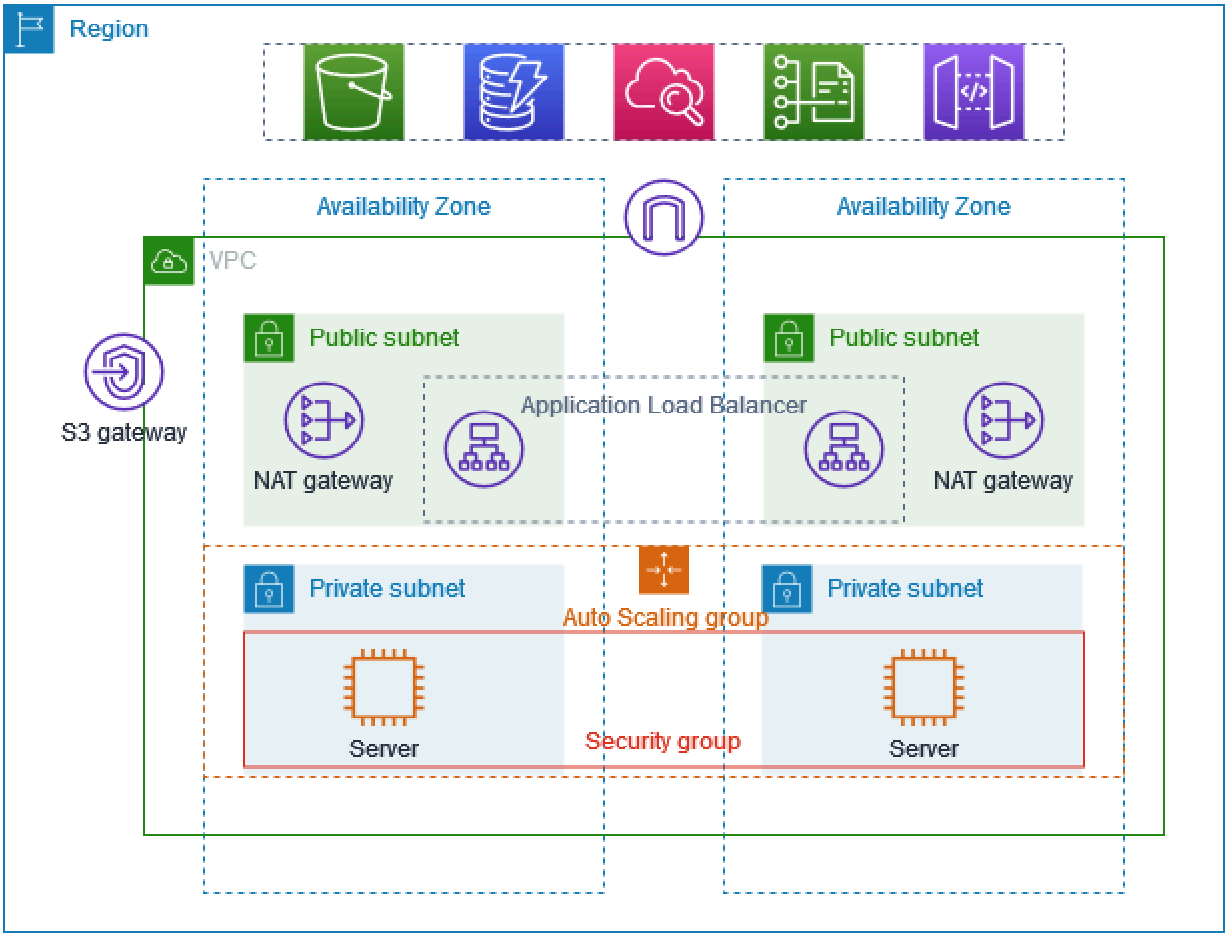
AWS VPC Architecture with Public and Private Subnets, NAT Gateway, Application Load Balancer(ALB) and Auto Scaling Group(ASG)



Introduction

This diagram demonstrates a secure and highly available AWS VPC setup using public and private subnets, NAT Gateways, Application Load Balancer (ALB) and Auto Scaling Group (ASG) to manage incoming traffic. The architecture is designed to scale automatically based on traffic and maintain security by isolating sensitive resources in private subnets.

Detailed Explanation of the Architecture

1. AWS Region and Availability Zones

- The architecture is deployed in a single **AWS Region** but spans across **two Availability Zones (AZs)** for high availability and fault tolerance.
- **Availability Zones** are isolated data centers within an AWS region, ensuring that if one zone experiences issues, the other remains operational.

2. VPC (Virtual Private Cloud)

- The **VPC** provides an isolated networking environment in AWS.
- You define the **CIDR block** (e.g., `10.0.0.0/16`), which determines the range of IP addresses available for your subnets.
- The VPC ensures secure communication between your resources within the cloud environment.

3. Public Subnets

- The two **Public Subnets** in the architecture are designed to allow resources (like the Application Load Balancer and NAT Gateway) to communicate with the public internet.
- These subnets are deployed in each Availability Zone to ensure that services remain operational even if one AZ goes down.
- Public subnets are essential for internet-facing services.

Key Components in Public Subnets:

- **NAT Gateway**:
  - Deployed in each public subnet to allow instances in the private subnet to access the internet for outbound traffic (e.g., downloading software updates).

    - It prevents inbound internet traffic from directly reaching the private instances, ensuring security.
    - The NAT Gateway is highly available and managed by AWS.
- **Application Load Balancer (ALB)**:
    - The ALB is also deployed in the public subnets. It distributes incoming traffic evenly across the backend servers (which are hosted in private subnets).
    - It improves fault tolerance by routing traffic only to healthy instances across different availability zones.
    - The ALB operates at Layer 7 (Application Layer), meaning it can route traffic based on content (e.g., HTTP/HTTPS requests).

4. Private Subnets

- **Private Subnets** host sensitive resources (like EC2 instances) that should not be directly accessible from the internet.
- These subnets are protected and only accessible through the NAT Gateway for outbound requests and the Application Load Balancer for incoming requests.

Key Components in Private Subnets:

- **EC2 Instances (Servers)**:
    - Backend servers are deployed in private subnets to ensure they are not directly exposed to the public internet.
    - These servers handle requests that are routed from the Application Load Balancer.
    - Each private subnet in this architecture has at least one EC2 instance running, ensuring high availability across the Availability Zones.
- **Auto Scaling Group (ASG)**:
    - The Auto Scaling Group automatically adjusts the number of EC2 instances based on traffic and load.

    - If demand increases (e.g., more users accessing your application), the ASG will launch additional instances.
    - Conversely, it can scale down to save costs when traffic is low.
    - ASG ensures that the application remains responsive even under high traffic conditions.

5. Security Groups

- **Security Groups** act as virtual firewalls for your EC2 instances.
- In this setup, security groups are applied to the EC2 instances in private subnets and the ALB in public subnets.

Security Group Details:

- **ALB Security Group**:
    - Allows inbound traffic on HTTP (port 80) and HTTPS (port 443) from the internet.
    - Only forwards traffic to instances in private subnets that are healthy and meet the health check criteria.
- **EC2 Instance Security Group**:
    - Restricts inbound traffic to only allow requests from the ALB (on relevant ports like 80 or 443).
    - Ensures that no traffic can directly reach the instances from the internet.

6.Network ACLs (NACLs)

- **Network ACLs (NACLs)** are an additional layer of security implemented at the subnet level. They provide **stateless** filtering of traffic, allowing you to explicitly allow or deny traffic entering or leaving your subnets.

Details of NACLs in this Architecture:

- **Inbound and Outbound Rules**:
  NACLs are applied to both public and private subnets in this architecture. They contain rules to:
  - Allow inbound HTTP and HTTPS traffic to the public subnets (where the ALB and NAT Gateways are located).
  - Allow outbound traffic from private subnets to the NAT Gateway for internet-bound requests.
  - Restrict unauthorized traffic by denying access to ports that are not in use.
- **Stateful vs Stateless**:
  Unlike security groups, which are stateful (meaning that return traffic is automatically allowed), NACLs are stateless, requiring explicit rules for both inbound and outbound traffic. This makes NACLs useful for adding an extra layer of granular control over the network.

7. Application Load Balancer (ALB)

- The **Application Load Balancer** is a key component in handling incoming traffic.
- It is an internet-facing load balancer that distributes traffic across the backend servers located in the private subnets.
- The ALB improves availability by distributing traffic evenly across multiple EC2 instances in different AZs.
- Additionally, it performs **health checks** to ensure traffic is only routed to healthy instances.

8. NAT Gateway

- The **NAT Gateway** is a managed AWS service that enables instances in the private subnets to initiate outbound internet traffic, such as downloading updates, without exposing them to inbound traffic from the internet.
- This ensures that private subnet instances remain secure while still having access to the internet for necessary tasks.
- Each public subnet has its own NAT Gateway to maintain high availability and ensure efficient routing.

9. Auto Scaling Group (ASG)

- The **Auto Scaling Group** is responsible for ensuring that the number of running EC2 instances matches the current demand.
- When traffic increases, the ASG automatically launches more instances.
- When traffic decreases, the ASG terminates instances to reduce costs.
- The ASG ensures both high availability and cost-effectiveness, allowing the architecture to dynamically respond to varying traffic loads.

10. S3 Gateway (Not Implemented in My Setup)

- Though the diagram includes an **S3 Gateway**, I did not implement it as part of my setup.
- An S3 Gateway typically allows instances in private subnets to access Amazon S3 storage without traversing the public internet, using VPC Endpoints to ensure secure and fast access.

11. High Availability and Fault Tolerance

- By deploying resources across multiple **Availability Zones**, the architecture is resilient to failures in a single data center.
- The **Application Load Balancer** ensures that traffic is distributed only to healthy instances, improving fault tolerance.

- **Auto Scaling** allows the architecture to handle varying levels of traffic without manual intervention, ensuring both availability and scalability.

Conclusion

This AWS VPC setup provides a secure, scalable, and highly available architecture for running applications in a cloud environment. With the combination of public and private subnets, an Application Load Balancer, NAT Gateways, and Auto Scaling, this architecture is ideal for applications requiring secure backend services, scalability, and fault tolerance.

Future Enhancements

In the future, this architecture could be further enhanced by:

- Adding an **S3 Gateway** for secure, private access to Amazon S3.
- Implementing **Terraform** or **AWS CloudFormation** for infrastructure as code (IaC) to automate the deployment process.
- Integrating **AWS RDS** (Relational Database Service) in private subnets to securely manage databases.