# **Project Documentation: Hosting a Website on a Private Server**

## **Overview**

This project demonstrates how to host a website on a private server within an Amazon Web Services (AWS) Virtual Private Cloud (VPC) while providing public access through a load balancer. The architecture includes a bastion host in a public subnet and a web server in a private subnet.

## **Objectives**

* Create a new VPC for hosting the web server.
* Set up a bastion host in the public subnet for secure access to the private subnet.
* Deploy a web server in the private subnet.
* Use an AWS Load Balancer to provide a public URL for accessing the web server.

## **Prerequisites**

* An AWS account with sufficient permissions to create VPCs, EC2 instances, and Load Balancers.
* Basic knowledge of AWS services, networking concepts, and web server deployment.

## **Technologies Used**

* Amazon VPC
* Amazon EC2
* AWS Elastic Load Balancer (ELB)
* Amazon Route 53 (optional, for domain management)
* HTTP server (e.g., Apache or Nginx)

## **Steps**

### **Step 1: Create a New VPC**

1. **Log in to the AWS Management Console.**
2. Navigate to the **VPC** dashboard.
3. Click on **Create VPC**.
   * **Name tag:** my-vpc
   * **IPv4 CIDR block:** 10.0.0.0/16
   * Click **Create VPC**.

### **Step 2: Create Subnets**

1. **Create a Public Subnet:**
   * Go to **Subnets** and click **Create Subnet**.
   * **Name tag:** public-subnet
   * **VPC:** Select my-vpc
   * **Availability Zone:** Choose any available zone.
   * **IPv4 CIDR block:** 10.0.1.0/24
   * Click **Create Subnet**.
2. **Create a Private Subnet:**
   * Repeat the above steps, but for the private subnet.
   * **Name tag:** private-subnet
   * **IPv4 CIDR block:** 10.0.2.0/24
   * Click **Create Subnet**.

### **Step 3: Configure Internet Gateway and NAT Gateway**

1. Navigate to the **Internet Gateways** section.
2. Click on **Create Internet Gateway**.
   * **Name tag:** my-igw
   * Click **Create**.
3. Attach the Internet Gateway to your VPC:
   * Select the Internet Gateway and click on **Actions** > **Attach to VPC**.
   * Choose my-vpc and click **Attach**.
4. Navigate to the **NAT Gateways** section.
5. Click on **Create NAT Gateway**.
   * **Name tag:** my-nat-gw
   * Assign it to a public subnet and
   * Allocate a elastic IP for the NAT-gateway
   * Click **Create**.

### **Step 4: Configure Route Tables**

1. **Public Route Table:**
   * Go to **Route Tables**, create a new route table, and name it public-rt.
   * Associate it with the public subnet.
   * Add a route for 0.0.0.0/0 targeting the Internet Gateway.
2. **Private Route Table:**
   * Create another route table, name it private-rt, and associate it with the private subnet.
   * Add a route for 0.0.0.0/0 targeting the NAT Gateway.

### **Step 5: Launch EC2 Instances**

1. **Launch the Bastion Host:**
   * Navigate to the **EC2 Dashboard** and click **Launch Instance**.
   * Choose an Amazon Machine Image (AMI), e.g., Amazon Linux 2.
   * Choose instance type (e.g., t2.micro).
   * In **Configure Instance**, select the public subnet.
   * Assign a public IP.
   * Configure security group to allow SSH (port 22) from your IP.
   * Review and launch the instance.
2. **Launch the Web Server:**
   * Repeat the steps to launch another EC2 instance for the web server.
   * Select the private subnet.
   * Configure the security group to allow HTTP (port 80) from the load balancer and SSH (port 22) from the bastion host.
   * Review and launch the instance.

### **Step 6: Install Web Server Software**

**SSH into the Bastion Host:**bash  
Copy code  
ssh -i your-key.pem ec2-user@<bastion-public-ip>

**SSH into the Web Server from the Bastion Host:**bash  
Copy code  
ssh -i your-key.pem ec2-user@<web-server-private-ip>

**Install a Web Server (e.g., Apache):**bash  
Copy code  
sudo yum update -y

sudo yum install httpd -y

sudo systemctl start httpd

sudo systemctl enable httpd

**Place an HTML file to test:**bash  
Copy code  
echo "<h1>Hello from Private Server!</h1>" | sudo tee /var/www/html/index.html

### **Step 7: Set Up the Load Balancer**

1. Navigate to the **EC2 Dashboard** and select **Load Balancers**.
2. Click **Create Load Balancer** and choose **Application Load Balancer**.
   * **Name:** my-load-balancer
   * **Scheme:** Internet-facing
   * **Listeners:** HTTP (port 80)
   * **Availability Zones:** Select the public subnet.
3. Configure security settings as needed and click **Next: Configure Routing**.
4. Create a new target group:
   * **Target type:** Instances
   * **Protocol:** HTTP
   * **Port:** 80
   * Register your web server instance with this target group.
5. Review and create the load balancer.

### **Step 8: Get the Public URL**

1. After the load balancer is created, note the DNS name provided.
2. You can now access your web server via this DNS name in a web browser.

### **Step 9: (Optional) Configure Domain Name**

* If you wish to use a custom domain, you can configure Amazon Route 53 or another DNS service to point to your load balancer’s DNS name.

## **Conclusion**

You have successfully set up a VPC with a private web server accessible via a public URL through a load balancer. This architecture allows for secure management of your web server while providing public access as needed.