Step-by-Step Guide for Deploying a Publicly Accessible Web Server on AWS EC2

Introduction: Basic Setup of an EC2 Web Server

This guide demonstrates the process of setting up a basic web server on an Amazon Web Services (AWS) EC2 instance. By leveraging AWS EC2 (Elastic Compute Cloud), you can easily deploy a web server that is both scalable and accessible to users around the world.

In this setup, we will walk through the deployment of an Ubuntu-based server using the AWS Free Tier, which allows you to host your website or application at no cost, as long as you stay within the specified usage limits. The server will be configured with **Apache2**, a widely-used open-source web server, and an **Elastic IP** will be used to maintain a static IP address, ensuring consistent and reliable access.

Additionally, we'll secure the server with **TLS/SSL** encryption using **Certbot**, making your website secure for all users by encrypting data traffic between the server and the client. Finally, we will use **Route53** to link your custom domain name to the Elastic IP, making your site accessible via a user-friendly web address.

Objective: Deploy an Ubuntu-based web server on AWS EC2 with Apache2, configure a static IP using Elastic IP, set up a TLS/SSL certificate using Certbot, configure Route 53 for domain mapping, and become a web developer with little to no coding experience.

1. Launch an EC2 Instance

- Login to AWS Console: Go to the AWS Management Console and log in.
- Launch Instance:
    - Navigate to EC2 under the Services menu.
    - Click Launch Instance to create a new instance.
    - Choose Ubuntu Server as the AMI (Amazon Machine Image) from the list of available operating systems.
    - Select Free Tier as the instance type (t2.micro).
    - Configure the instance details, keeping default settings unless you need to make specific adjustments.
    - Add storage if needed (the default should be fine for basic setups).
    - Configure security groups to allow inbound HTTP (port 80), HTTPS (port 443), and SSH (port 22) traffic.
- Key Pair: Select or create a key pair to SSH into your server after launch.
- Launch Instance: Review your settings and click Launch.

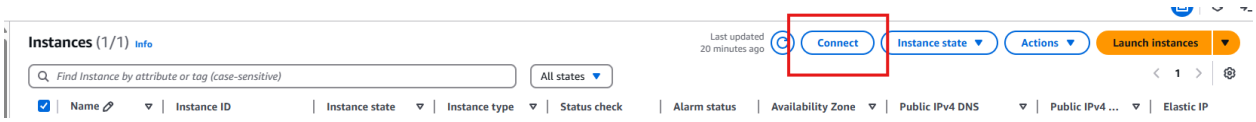
2. Allocate and Associate Elastic IP

- Go to Elastic IPs: In the EC2 Dashboard, select Elastic IPs under Network & Security.
- Allocate New Elastic IP: Click Allocate Elastic IP address to get a static IP.
- Associate Elastic IP:

- After allocation, select the Elastic IP and click Associate.
- Choose the EC2 instance to associate it with.
- This ensures your instance maintains a static IP even if it gets restarted.

3. Connect to Your EC2 Instance

You can connect via the connect button in the instances home page or



Open a terminal on your local machine and use SSH to connect to the instance:

- ssh -i "your-key.pem" ubuntu@<Elastic-IP>

Replace your-key.pem with your private key file and <Elastic-IP> with the Elastic IP you just allocated.

4. Install Apache2 Web Server

Once logged into your EC2 instance, update package lists:

- sudo apt update

Install Apache2:

- sudo apt install apache2 -y

Check if Apache2 is running by accessing http://<Elastic-IP> in a browser. This may take a moment

5. Configure Route 53 and Apply A Record

If you choose to buy a domain name, mine costs around $14 per year which is a good investment in my eyes, here is how you can apply the purchased domain name. I purchased mine through AWS route53.

- Log in to AWS Route 53: Go to the Route 53 console in the AWS Management Console.
- Create a Hosted Zone: If you don't already have one, click Create Hosted Zone and set the domain name that you purchased.
- Create an A Record:
    - In the hosted zone, click Create Record.
    - Select A – IPv4 address as the record type.
    - In the Value field, enter the Elastic IP you assigned to your instance.
    - Set the TTL (Time To Live) and save the record.

6. Set Up TLS/SSL with Certbot

Install Certbot:

- sudo apt install certbot python3-certbot-apache -y

Obtain and install the SSL certificate for your domain:

- sudo certbot --apache -d examplewesbite.com

Follow the prompts to complete the process, which will also configure Apache to use the SSL certificate.

7. Creating the content for the webserver

How to Use AI to Create an index.html File for Your Website

Objective: Leverage AI tools to generate an HTML structure and design for your website, making it easy to customize your site's look and feel without needing to code manually.

For my website, I decided to take the portfolio route and create a web server that is tailored to my experience. I created different directories in the ubuntu server which can be accessed via the buttons on the website. I created one for my personal projects, my resume and my linkedin profile. Here is how you can get creative.

Firstly navigate to the directory where the web server's root files are located:

- cd /var/www/html

Open index.html using a text editor like nano:

- sudo nano index.html

Now ask AI to tailor the website to you, example prompt can be, Can you create an index.html page for my personal cybersecurity portfolio website? It should include a header with a navigation bar, a section for my skills and certifications, and a section for my cybersecurity projects with links to PDFs for each project. I'd also like the design to be modern with a professional look, incorporating my color scheme of blue, white, and black."

Using a prompt like this, you can take the code and keep parsing it through AI based on your personal preferences. For example, if you don't like the color scheme, you can ask AI to change it to something that better suits your style. Here's how you can do it:

Example AI Request for Color Scheme:

"I don't like the current color scheme. Can you change the background to light gray, the header to a gradient of dark blue to teal, and the text color to black for better readability?"

Other Customizations:

Font Style: "Can you change the fonts to something more modern, like 'Roboto' or 'Open Sans'?"

Spacing and Margins: "Can you adjust the margins on the main content to give it more space and reduce clutter?"

Button Design: "Can you redesign the buttons to have rounded corners and a hover effect that changes their color to orange?"