

# HOSTING PERSONAL SITE ON AWS EC2 INSTANCE

*Prepared in the partial fulfillment of the Summer Internship Program on AWS*

AT



*Under the guidance of*

*Mr. Anil Kumar, APSDC*

*MR. RAMA KRISHNA, APSDC*

*Submitted by*

*TEAM: 80*

*B. Nikhita, 21A21A1263*

*K. Manvitha Lakshmi Sai, 21A21A1228*

*S.S.R Chandrika, 21A21A1251*

*V. Devaki Anandi, 21A21A1258*

*Swarnandhra College of Engineering and Technology (September 2023)*

---

# ***ACKNOWLEDGEMENT***

I would like to express my heartfelt gratitude to all those who have contributed to successfully completing my summer internship project at Andhra Pradesh Skill Development Corporation (APSSDC). This opportunity has been an enriching and transformative experience for me, and I am truly thankful for the support, guidance, and encouragement I have received along the way.

First and foremost, I extend my sincere regards to Mr. Anil Varnasi and Mr. Rama Krishna, my supervisors and mentors, for providing me with valuable insights, constant guidance, and unwavering support throughout the duration of the internship. Their expertise and encouragement have been instrumental in shaping the direction of this project.

I would like to thank the entire team at Andhra Pradesh Skill Development Corporation (APSSDC) for fostering a collaborative and innovative environment. The camaraderie, knowledge sharing, and feedback I received from my colleagues significantly contributed to the development and success of this project.

In conclusion, I am honored to have been a part of this internship program. I look forward to leveraging the skills and knowledge gained to contribute positively to future endeavors.

Thank you.

Sincerely,

B.Nikhita,21A21A1263

K.Maanvitha Lakshmi Sai,21A21A1228

S.S.R Chandrika,21A21A1251

V.Devaki Anandi,21A21A1258

[nikhitabaskarla@gmail.com](mailto:nikhitabaskarla@gmail.com)

# ***ABSTRACT***

With the power of cloud computing, hosting your website has never been easier in today's digital world. Which service to look into? Keep your business competitive as you host a website on AWS EC2!

An EC2 instance is simply a virtual server in amazon web service technology . With an EC2 instance, AWS subscribers can request and provision a computer server within the AWS cloud.

Amazon EC2 provides the broadest and deepest instance choice to match your workload's needs. General purpose, compute optimized, memory optimized, storage optimized, and accelerated computing instance types are available that provide the optimal compute, memory, storage, and networking balance for your workloads.

Features of Amazon EC2 instances provides the following high level features like instances, Amazon Machine Images, instances types, key pairs, instance store volumes, Amazon EBS volumes, regions availability zones, security group rules, elastic IP address, tags etc.

There are many benefits to using AWS EC2 for website hosting, including flexibility, scalability, and cost effectiveness. EC2 allows you to easily add or remove compute capacity as needed, and you only pay for what you use.

## TABLE OF CONTENTS

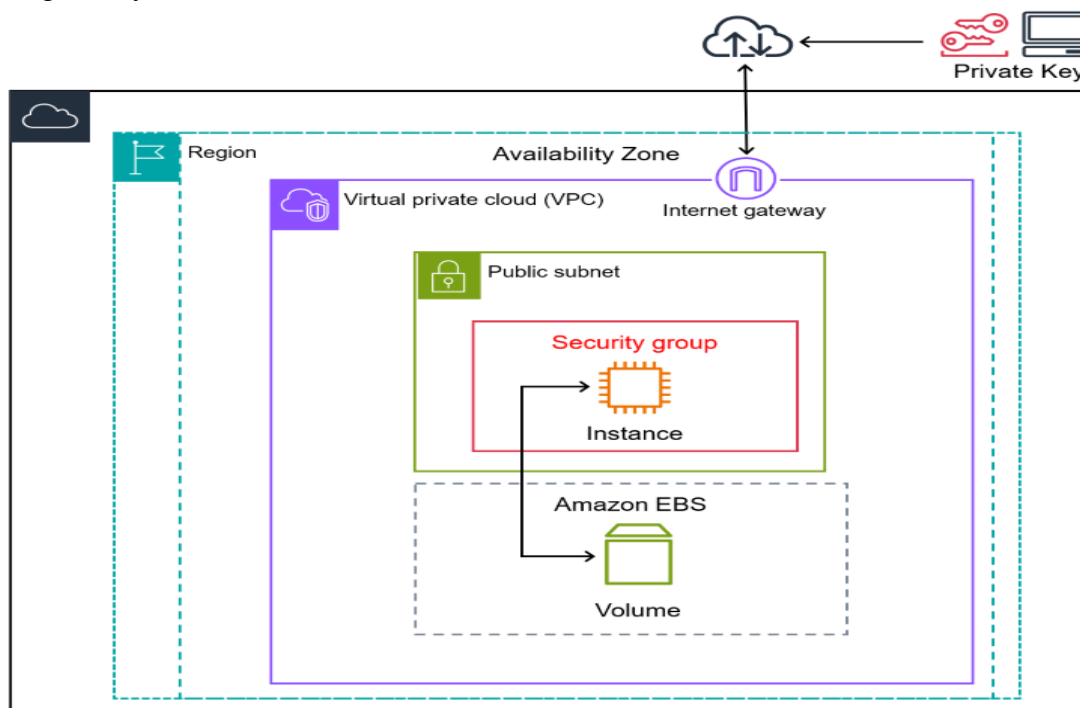
| <b>S.No</b> | <b>Content</b>                    | <b>pg.no</b> |
|-------------|-----------------------------------|--------------|
| 1.          | Introduction.....                 | 5            |
| 2.          | Methodology .....                 | 7            |
| 3.          | System Design / Architecture..... | 9            |
| 4.          | Implementation .....              | 19           |
| 5.          | Results .....                     | 28           |
| 6.          | Conclusion .....                  | 32           |

# 1. INTRODUCTION

Amazon EC2 provides scalable computing capacity in the AWS cloud. Leveraging it enables organizations to develop and deploy applications faster, without needing to invest in hardware upfront. Users can launch virtual servers, configure security and networking, and manage cookies from an intuitive dashboard

Among the vast array of services that Amazon offers, EC2 is the core compute component of the technology stack. In practice, EC2 makes life easier for developers by providing secure, and resizable compute capacity in the cloud. It greatly eases the process of scaling up or down, can be integrated into several other services, and comes with a plan where you only pay for how much you use it.

The following diagram shows a basic architecture of an Amazon EC2 instance deployed within an Amazon Virtual Private Cloud (VPC). In this example, the EC2 instance is within an Availability Zone in the Region. The EC2 instance is secured with a security group, which is a virtual firewall that controls incoming and outgoing traffic. A private key is stored on the local computer and a public key is stored on the instance. Both keys are specified as a key pair to prove the identity of the user. In this scenario, the instance is backed by an Amazon EBS volume. The VPC communicates with the internet using an internet gateway. For more information about Amazon VPC, see the



# *Features of Amazon EC2*

Amazon EC2 provides the following high-level features:

## **Instances**

Virtual servers.

## **Amazon Machine Images (AMIs)**

Preconfigured templates for your instances that package the components you need for your server (including the operating system and additional software).

## **Instance types**

Various configurations of CPU, memory, storage, networking capacity, and graphics hardware for your instances.

## **Key pairs**

Secure login information for your instances. AWS stores the public key, and you store the private key in a secure place.

## **Instance store volumes**

Storage volumes for temporary data that is deleted when you stop, hibernate, or terminate your instance.

## **Amazon EBS volumes**

Persistent storage volumes for your data using Amazon Elastic Block Store (Amazon EBS).

## **Regions, Availability Zones, Local Zones, AWS Outposts, and Wavelength Zones**

Multiple physical locations for your resources, such as instances and Amazon EBS volumes.

## **Security groups**

A virtual firewall that allows you to specify the protocols, ports, and source IP ranges that can reach your instances and the destination IP ranges to which your instances can connect.

## **Elastic IP addresses**

Static IPv4 addresses for dynamic cloud computing.

## **Tags**

Metadata that you can create and assign to your Amazon EC2 resources.

## **Virtual private clouds (VPCs)**

Virtual networks you can create that are logically isolated from the rest of the AWS Cloud. You can optionally connect these virtual networks to your

## ***2. METHODOLOGY***

The hosting of the website on AWS using EC2 instances followed a systematic and iterative methodology to ensure the successful implementation of the project objectives. The methodology encompassed several phases, including requirements gathering, design, implementation, testing, and deployment. The following sections outline each phase of the methodology in detail.

### **Requirements Gathering**

The project commenced with an analysis of the requirements and expectations of the Project . This phase involved interactions with mentors. Feedback sessions were conducted to identify key features.

### **Design**

Based on the collected requirements, a system design was formulated. The architecture of the system was planned, outlining the interaction between the frontend, backend, and database components. Special emphasis was placed on creating an intuitive user experience, seamless navigation, and responsive design to ensure accessibility across various devices.

### **Implementation**

The implementation phase involved translating the design specifications into a functional system. A web-based approach was chosen for its widespread accessibility and ease of use. By using Amazon ec2 instance.

### **Database Design**

It starts with Amazon EC2 instances , you can choose windows or linux/unix operating systems and popular web applications, including , free CSS templates and deploy these with a single click from preconfigured templates.

### **Testing**

The testing phase encompassed integration testing and user acceptance testing. Integration testing ensured seamless communication between frontend and backend modules. User acceptance testing involved real users interacting with the system to identify and rectify any usability issues or bugs.

### **Deployment**

The hosting of website on AWS EC2 instances includes security group rules .The systems components were configured to ensure optimal performance, security, and scalability.

### **Iterative Refinement**

Throughout the development lifecycle, an iterative approach was adopted to incorporate

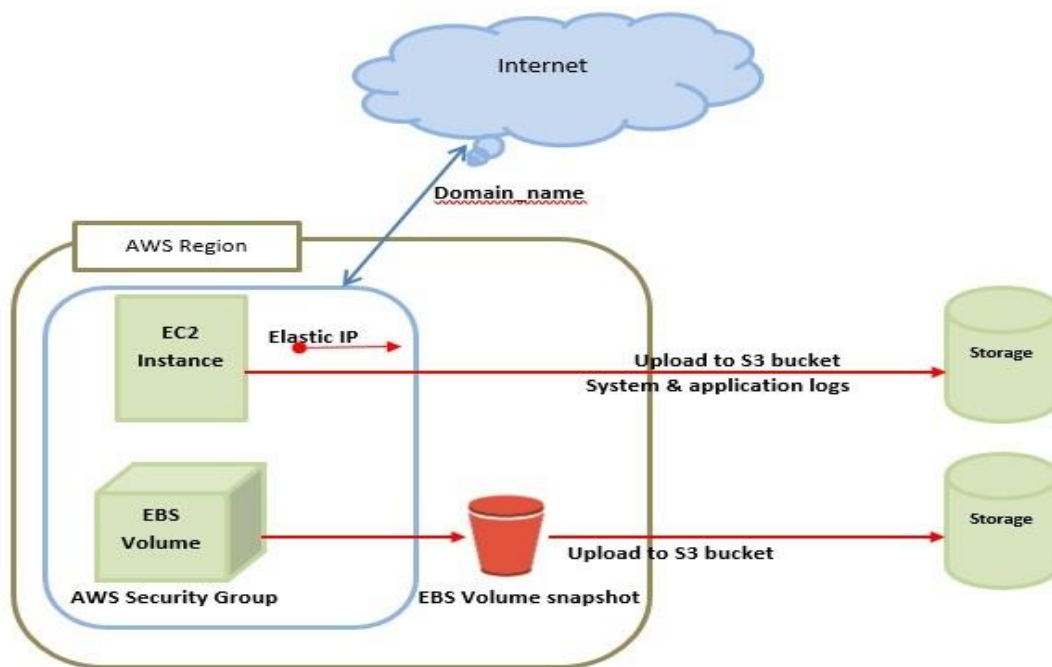
feedback, address challenges, and refine the system. The methodology employed in the hosting website on AWS facilitated a comprehensive and user-centric approach, ensuring the creation of a functional, secure, and intuitive platform for all the students to showcase their achievements and skills effectively.

### 3.SYSTEM DESIGN / ARCHITECTURE

#### Get started with Amazon EC2

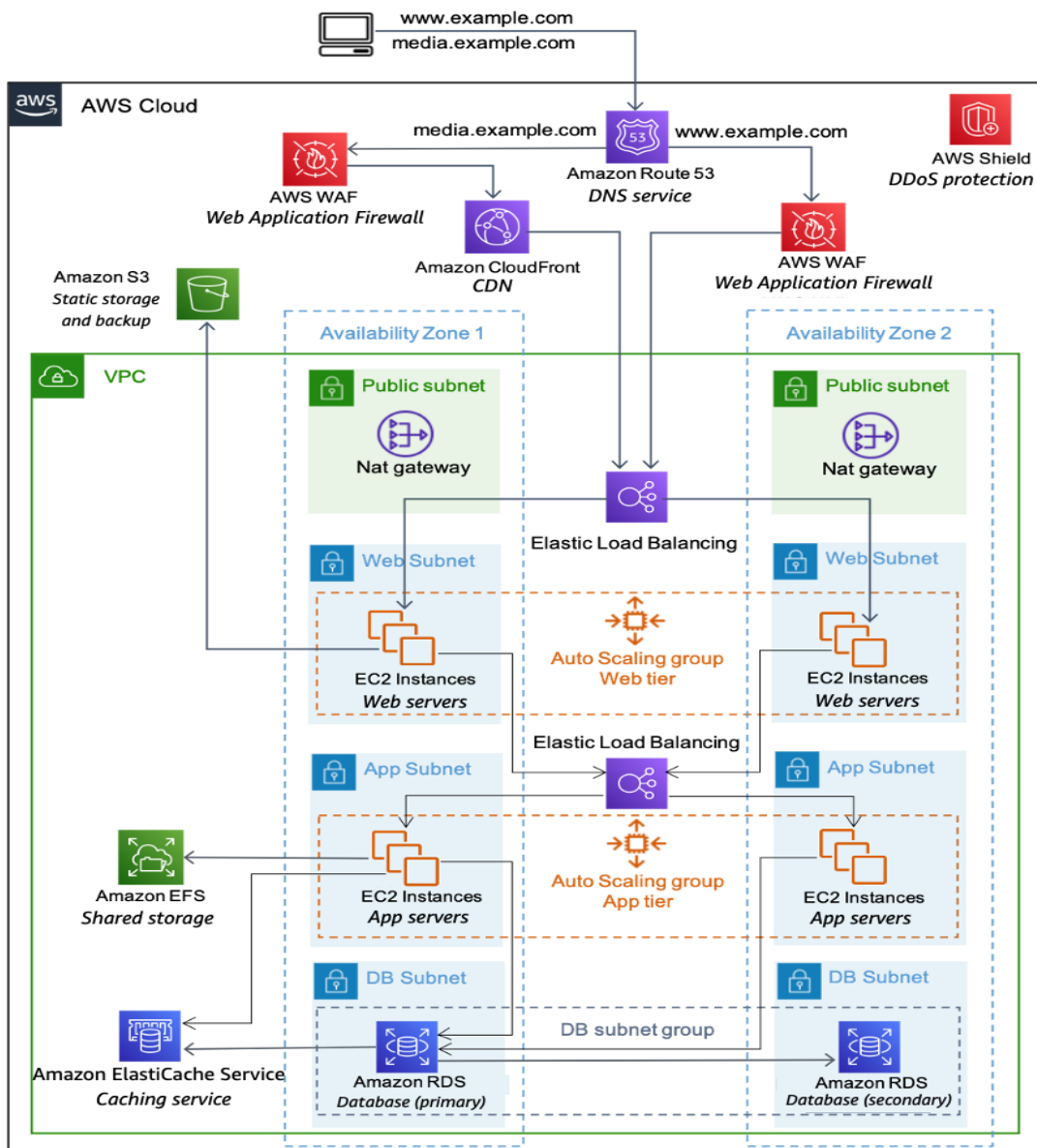
When hosting your website, you need a virtual server on the cloud. AWS provides an EC2 instance as the virtual server.

This is the basic structure of AWS EC2, where EC2 stands for Elastic Compute Cloud. EC2 allow users to use virtual machines of different configurations as per their requirement. It allows various configuration options, mapping of individual server, various pricing options, etc. We will discuss these in detail in AWS Products section. Following is the diagrammatic representation of the architecture.



It is considered as the basic structure of AWS architecture or AWS EC2. Simply, EC2 is also called Elastic Compute cloud which will allow the clients or else the users of using various configurations in their own project or method as per their requirement. There are also different amazing options such as pricing options, individual server mapping, configuration server, etc. S3 which is present in the AWS architecture is called Simple Storage Services. By using this S3, users can easily retrieve or else store data through various data types using Application Programming Interface calls. There will be no computing element for the services as well.





### Key concepts of hosting website on EC2:

These are EC2 instances that you can launch at any time and have it provisioned and available to you within minutes. You can use this instance for a shorter time or for as long as you need before terminating the instance. These instances have a flat rate and is determined on the instance type selected and is paid by the second. On-demand instances are typically used for short term uses where workloads can be irregular and where workload can be interrupted. Many users of AWS use on-demand instances within their testing and development environments. And when you stop or terminate your on-demand instance you'll stop paying for the compute resource.

- Highly Available Infrastructure. Highly available underlying design.

- Flexible and Seamless Expansion Capability. Automatic Scaling in/out with Amazon Auto Scaling.
- Rich Choice of Operating Systems and Software. Rich Choice of Operating Systems and Software.

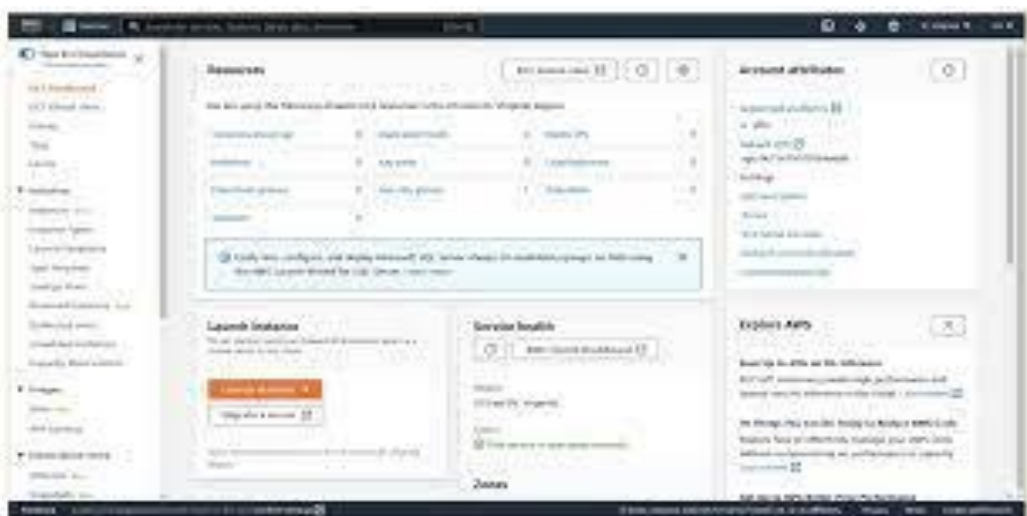
## 4. IMPLEMENTATION

### Setting Up Your EC2 Instance:

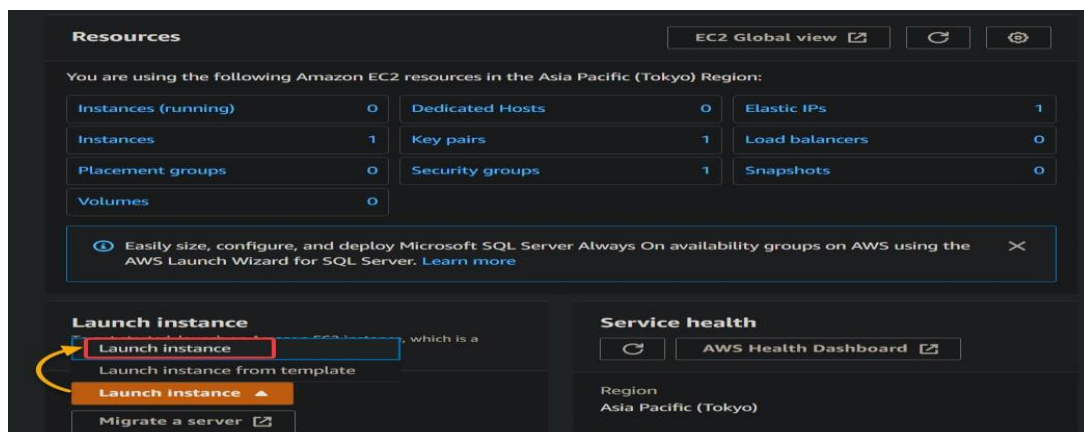
When hosting your website, you need a virtual server on the cloud. AWS provides an EC2 instance as the virtual server.

To set up an EC2 instance, you will create a virtual machine (to host your website) in the cloud that you can access remotely. With an EC2 instance, you can install the server software and applications you need to host your website.

1. Open your favorite web browser, and sign in to your AWS management console.
2. Next, search for and select EC2 from the result list to navigate to the EC2 console.



3. On the EC2 console, click Launch Instance, and choose Launch Instance to begin creating your EC2 instance.



4. Configure the following Name and OS Image for your EC2 instance:

**Launch an instance** [Info](#)

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

**Name and tags** [Info](#)

Name:  [Add additional tags](#)

**Application and OS Images (Amazon Machine Image)** [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below.

**Quick Start**

**Amazon Linux** **macOS** **Ubuntu** **Windows** **Red Hat** **S**

[Browse more AMIs](#)  
Including AMIs from AWS, Marketplace and the Community

5. Scroll down and choose the Instance type that best suits your needs for hosting your website.

**Instance type** [Info](#)

**Instance type**

**t2.micro** **Free tier eligible**

Family: t2 1 vCPU 1 GiB Memory  
On-Demand Linux pricing: 0.0152 USD per Hour  
On-Demand Windows pricing: 0.0198 USD per Hour

[Compare instance types](#)

**Key pair (login)** [Info](#)

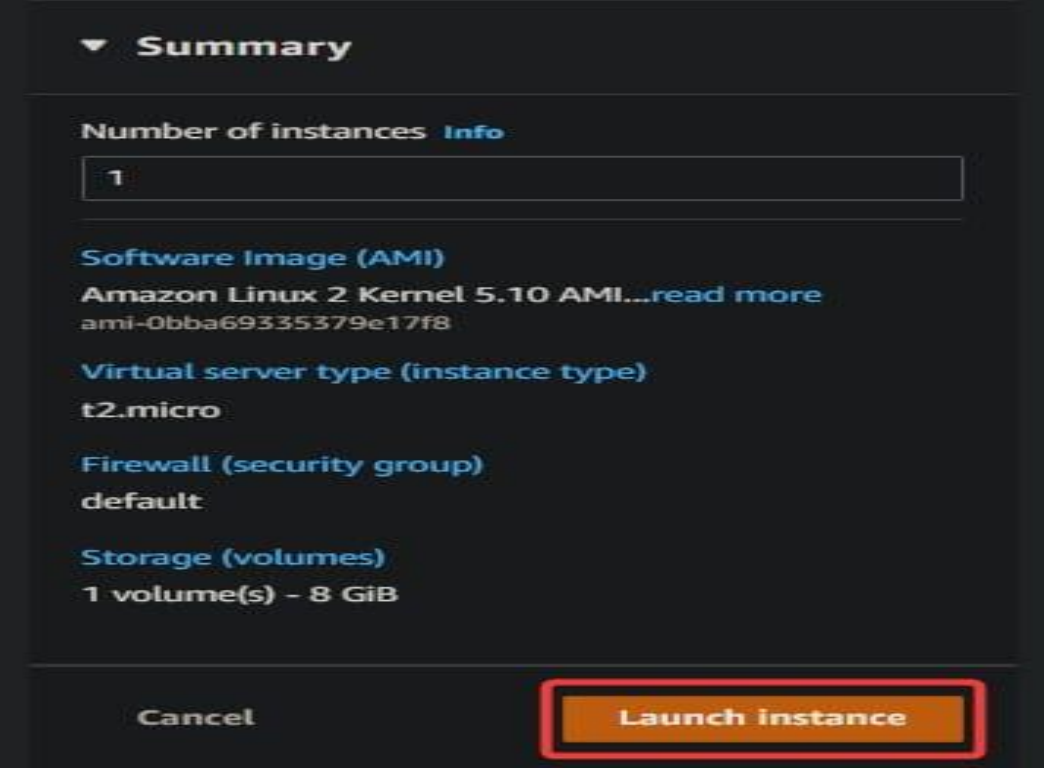
You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

**Key pair name - required**

**Default value** [Create new key pair](#)

6. Next, select an existing security group under the Network settings section or create a new one. Security groups act as virtual firewalls to secure your instance from unauthorized access.

7. Keep other settings as default, and click Launch under the Summary section (right pane).



The screenshot shows the 'Summary' section of the AWS Management Console for launching an EC2 instance. The 'Launch instance' button is highlighted with a red rectangle.

**Summary**

Number of instances [Info](#)

1

**Software Image (AMI)**

Amazon Linux 2 Kernel 5.10 AMI...[read more](#)  
ami-0bba69335379e17f8

**Virtual server type (instance type)**

t2.micro

**Firewall (security group)**

default

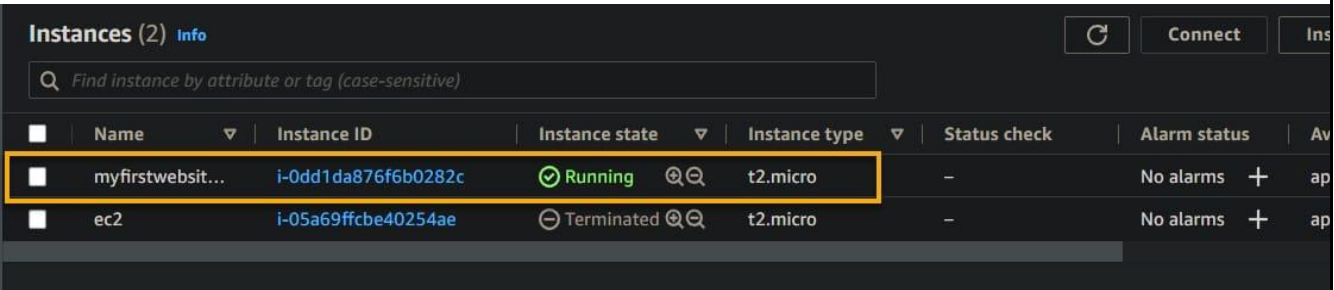
**Storage (volumes)**

1 volume(s) - 8 GiB

Cancel **Launch instance**

8. Lastly, navigate to the EC2 console, and you will see your new instance in the list, as shown below.

Congratulations! You have successfully set up an EC2 instance to host your website.



The screenshot shows the 'Instances' list in the AWS Management Console. The first instance, 'myfirstwebsit...', is highlighted with a yellow box.

|                                     | Name             | Instance ID         | Instance state | Instance type | Status check | Alarm status | Av |
|-------------------------------------|------------------|---------------------|----------------|---------------|--------------|--------------|----|
| <input checked="" type="checkbox"/> | myfirstwebsit... | i-0dd1da876f6b0282c | Running        | t2.micro      | -            | No alarms +  | ap |
| <input checked="" type="checkbox"/> | ec2              | i-05a69ffcbe40254ae | Terminated     | t2.micro      | -            | No alarms +  | ap |

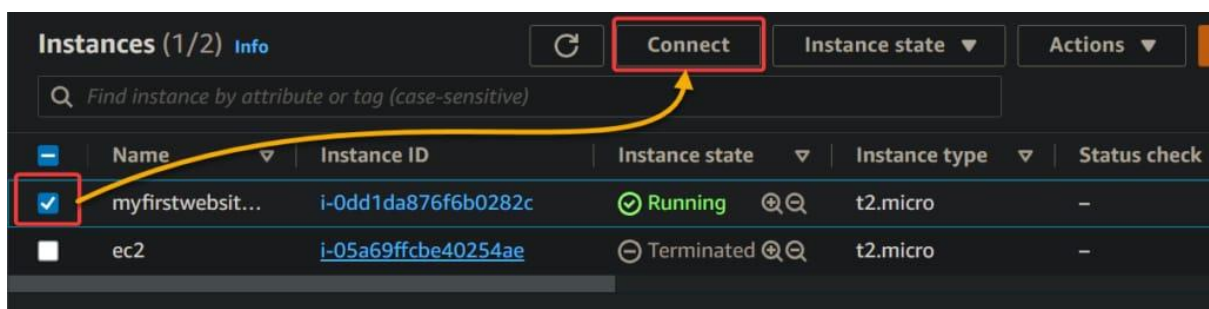
# Installing a Web Server to Host a Website on AWS EC2

With your EC2 instance set up, you now need a way to deliver web content to users over the internet, a web server. As a software program, a web server on your EC2 instance allows you to host your website and make it accessible to users.

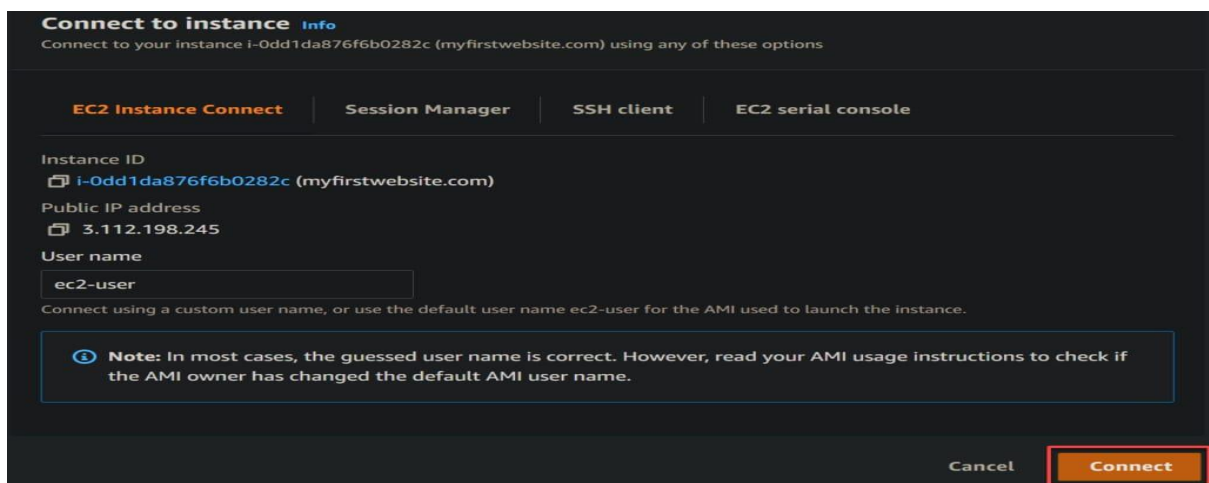
Many different web server options are available, including Apache, NGINX . But this tutorial uses Apache, one of the most popular and widely used web servers, open-source and highly configurable

To install Apache as the web server on your EC2 instance, follow these steps:

1. On your Instances page, tick the box next to your EC2 instance, and click Connect to connect to the selected EC2 instance.



2. Next, keep the settings as default, and click Connect, which opens a console window where you can run commands to your EC2 instance remotely.



The output below signifies you have successfully connected to your EC2 instance.

```
aws Services Search [Alt+S]

  _ | _ | _ )
 _ | ( _ /   Amazon Linux 2 AMI
  _ | \ _ | _ |

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-40-234 ~]$
```

3. Run the below command to gain root access and fully control your EC2 instance. Doing so lets you make any necessary changes or modifications to your web server.

```
aws Services Search [Alt+S]

  _ | _ | _ )
 _ | ( _ /   Amazon Linux 2 AMI
  _ | \ _ | _ |

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-40-234 ~]$ sudo su
[root@ip-172-31-40-234 ec2-user]#
```

4. Now, run the following yum command to update the list of available packages. This command ensures you have the latest version of packages and security updates.

```
[root@ip-172-31-40-234 ec2-user]# yum update -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core | 3.7 kB 00:00:00
```

5. Once updated, run each command below to install the Apache web server (httpd) and the PHP 8.0 (php8.0) package from the amazon-linux-extras repository.

```
[root@ip-172-31-40-234 ec2-user]# yum install httpd -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Package httpd-2.4.54-1.amzn2.x86_64 already installed and latest version
[root@ip-172-31-40-234 ec2-user]# sudo amazon-linux-extras install php8.0 -y
Installing php-pdo, php-fpm, php-mysqlnd, php-cli
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Cleaning repos: amzn2-core amzn2extra-docker amzn2extra-kernel-5.10 amzn2extra-php8.0
17 metadata files removed
6 sqlite files removed
0 metadata files removed
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
```

6. Now, run the below service commands to start the Apache web server (httpd) and check its status.



```
[root@ip-172-31-40-234 ec2-user]# service httpd start
Redirecting to /bin/systemctl start httpd.service
[root@ip-172-31-40-234 ec2-user]# service httpd status
Redirecting to /bin/systemctl status httpd.service
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)
   Drop-In: /usr/lib/systemd/system/httpd.service.d
            └─php-fpm.conf
   Active: active (running) since Mon 2023-01-09 09:58:57 UTC; 48s ago
     Docs: man:httpd.service(8)
  Main PID: 3933 (httpd)
   Status: "Total requests: 0; Idle/Busy workers 100/0; Requests/sec: 0; Bytes served/sec: 0 B/sec"
    CGroup: /system.slice/httpd.service
            └─3933 /usr/sbin/httpd -DFOREGROUND
              └─3939 /usr/sbin/httpd -DFOREGROUND
                └─3940 /usr/sbin/httpd -DFOREGROUND
                  └─3941 /usr/sbin/httpd -DFOREGROUND
                    └─3942 /usr/sbin/httpd -DFOREGROUND
                      └─3943 /usr/sbin/httpd -DFOREGROUND
```

7. Finally, open the Public IPv4 DNS of your EC2 instance in a new browser tab to request web content from the Apache web server.

EC2 > Instances > i-0dd1da876f6b0282c

**Instance summary for i-0dd1da876f6b0282c (myfirstwebsite.com)** Info

Updated less than a minute ago

[Refresh](#) [Connect](#) [Instance state](#) [Actions](#)

|   |  |   |
|---|--|---|
| <b>Instance ID</b><br>i-0dd1da876f6b0282c<br>(myfirstwebsite.com)                 | <b>Public IPv4 address</b><br>3.112.198.245   <a href="#">open address</a>                 | <b>Private IPv4 addresses</b><br>172.31.40.234  |
| <b>IPv6 address</b><br>-  | <b>Instance state</b><br>Running   | <b>Public IPv4 DNS</b><br>ec2-3-112-198-245-ap-northeast-1.compute.amazonaws.com   <a href="#">open address</a> |
| <b>Hostname type</b><br>IP name: ip-172-31-40-234.ap-northeast-1.compute.internal | <b>Private IP DNS name (IPv4 only)</b><br>ip-172-31-40-234.ap-northeast-1.compute.internal |   |

If all goes well, you will see the Apache test page, as shown below, which indicates you have successfully installed and configured the Apache web server.

Not secure | ec2-3-112-198-245-ap-northeast-1.compute.amazonaws.com

## Test Page

This page is used to test the proper operation of the Apache HTTP server after it has been installed. If you can read this page, it means that the Apache HTTP server installed at this site is working properly.

**If you are a member of the general public:**

The fact that you are seeing this page indicates that the website you just visited is either experiencing problems, or is undergoing routine maintenance.


If you would like to let the administrators of this website know that you've seen this page instead of the page you expected, you should send them e-mail. In general, mail sent to the name "webmaster" and directed to the website's domain should reach the appropriate person.

For example, if you experienced problems while visiting www.example.com, you should send e-mail to "webmaster@example.com".

**If you are the website administrator:**

You may now add content to the directory `/var/www/html/`. Note that until you do so, people visiting your website will see this page, and not your content. To prevent this page from ever being used, follow the instructions in the file `/etc/httpd/conf/20welcome.conf`.

You are free to use the image below on web sites powered by the Apache HTTP Server.



## 5. *RESULT*

### Creating Your First Website

Now that you have installed a fully functional Apache web server, you are ready to host your website. Well, almost, since you first need a website to host. In this tutorial, you will create a simple dynamic web page using PHP, generated by a web server, and can display different content on each access.

Creating your dynamic web page involves creating and placing a PHP script in the appropriate directory on your EC2 instance. Doing so lets the Apache web server serve your website to users.

1. Navigate to the root directory of your web server (/var/www/html), which is main directory where all of the web content for your website is stored.

When a user accesses your website, the web server looks in the root directory to find the requested content.

2. Next, create a new file called index.php with your preferred editor. This file is an interactive landing page for your website, displaying different content based on user input.

3. Add the following code to your index.php file, save the changes, and close the file. The code below creates a simple PHP script that generates a dynamic web page with a Click Me button. When users click the Click Me, the script displays a different message each time as defined in the array of messages.

At this point, your web page is ready to be served by the Apache web server.

```
<html>
<body>

<?php
// Create an array of messages
$messages = array("Thank you for clicking the button!", "Thanks for your interest!", "Glad
you enjoyed the website!");

// Check if the "click" variable is set in the POST data
if (isset($_POST['click'])) {
    // If the "click" variable is set, select a random message from the array
    $message = $messages[array_rand($messages)];
} else {
    // If the "click" variable is not set, use the default message
    $message = "Hello, World!";
}

// Output the message to the web page
echo $message;
```



?>

<!-- Create a form with a "Click Me" button that submits a POST request -->

<form method="post">

<input type="submit" name="click" value="Click Me">

</form>

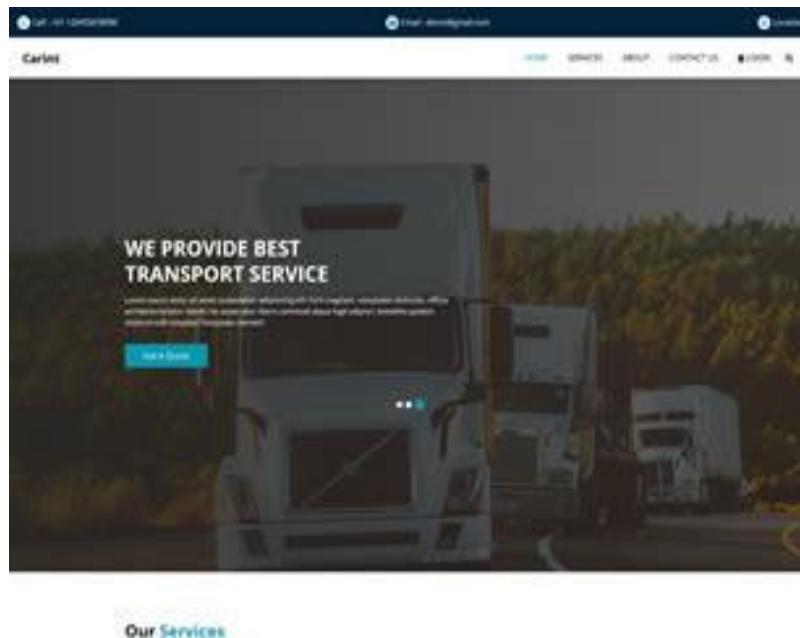
</body>

</html>

4. Ultimately, reopen the Public IPv4 DNS of your EC2 instance in a new browser tab, and click the Click Me button to see your website in action.

As shown below, you will see a different message each time you click the button.

Congratulations! You have created your first dynamic web page using PHP and Apache web servers.



# ***CONCLUSION***

Hosting a website on AWS using an EC2 instance is a common and effective choice. In conclusion, here are some key points to consider:

Depending on your preference, you can install a web server software like Apache, Nginx, or any other web server you prefer. For example, on Amazon Linux, you can use the following command to install Apache.

You can upload your website files to your EC2 instance using SCP, FTP, or any other method you prefer. Place your website files in the appropriate directory for your web server (e.g., /var/www/html for Apache).

In conclusion, hosting a website on AWS using an EC2 instance offers scalability, flexibility, and a wide range of services to meet your specific needs. Proper planning, security, and maintenance are essential for a successful deployment.