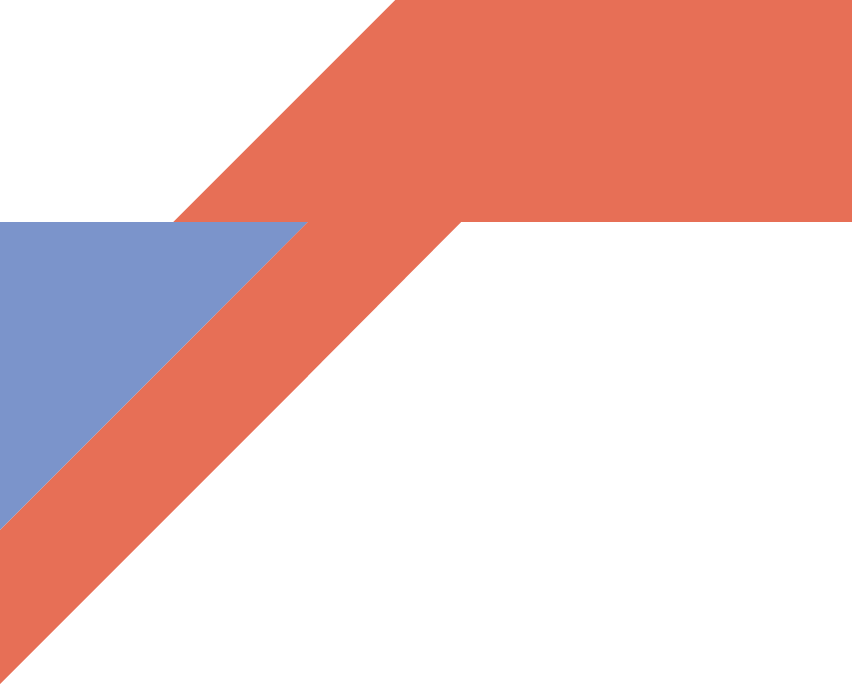
Project by Rishabh Jha





Web Hosting

AWS project



Contents

A)Introduction ......................................................................... 3

B)Technologies Used ....................................................4

C)Project Steps with screenshots ............................................................ 5

a)Launching an EC2 Instance .........................................................................5

b)Connecting to the EC2 Instance .............................................11

c)Configuring the web server.............................................................. 16

D)Conclusion ......................................................19

Introduction

This project aims to demonstrate how to host a website using Amazon Web Services (AWS) Elastic Compute Cloud (EC2) with a Windows server. AWS provides a comprehensive, scalable, and cost-effective cloud computing platform that enables individuals and organizations to deploy and manage applications in a secure and reliable environment.

In this project, we will utilize AWS EC2, a web service that provides resizable compute capacity in the cloud, to host a website. EC2 allows users to run virtual servers in the cloud, providing the necessary resources and infrastructure to deploy and manage web applications efficiently. By launching a Windows-based EC2 instance, we can take advantage of the robust features and flexibility that AWS offers for hosting web applications.

The primary goal of this project is to set up an EC2 instance with a Windows server, install the necessary web server software, upload the website files, and ensure the website is accessible via the instance's public IP address. This process involves several detailed steps, which will be thoroughly documented in this report, including screenshots to illustrate the procedure.

AWS provides several advantages for hosting websites:

- **Scalability**: EC2 instances can be scaled up or down based on demand, ensuring that the website can handle varying levels of traffic efficiently.

- **Reliability:** AWS's infrastructure is designed for high availability, providing a reliable platform for hosting web applications.

- **Cost-Effectiveness:** With AWS's pay-as-you-go pricing model, we only pay for the resources we use, making it a cost-effective solution for hosting websites.

- **Security:** AWS offers robust security features, including network firewalls, encryption, and access controls, to protect our web applications and data.

By leveraging AWS EC2, we can deploy a powerful and flexible web hosting solution. This project report will outline the steps taken to achieve this, providing a comprehensive guide for anyone looking to host a website on AWS using a Windows server. This knowledge can be applied to a variety of web hosting scenarios, demonstrating the flexibility and efficiency of AWS for cloud-based solutions.

Technologies Used

This project utilized several key technologies to successfully host a website on AWS using a Windows server:

1. **Amazon Web Services (AWS)**:

AWS is a leading cloud computing platform that offers a wide range of services. We used AWS to deploy and manage our website, specifically utilizing EC2 to create and configure the virtual server.

2. **Amazon EC2 (Elastic Compute Cloud)**:

Amazon EC2 provides scalable virtual servers in the cloud. For this project, we launched a Windows-based EC2 instance to host our website. EC2 allows flexibility in choosing the instance type and configuring security settings, making it an ideal solution for hosting web applications.

3. **Microsoft Windows Server**:

Windows Server is the operating system used for the EC2 instance. It provides a stable and familiar environment for web hosting, supporting a variety of applications and services.

4. **Remote Desktop Protocol (RDP):**

RDP allows us to connect to and manage the Windows-based EC2 instance remotely. This enabled us to install necessary software and configure the web server directly from our local machine.

5. **Internet Information Services (IIS):**

IIS is the web server software used to host the website. Included with Windows Server, IIS supports various web protocols and frameworks, allowing us to serve web content efficiently.

6. **HTML, CSS, and JavaScript:**

These frontend technologies were used to build the website hosted on the EC2 instance. HTML provides structure, CSS handles styling, and JavaScript adds interactivity to the website.

7. **Security Groups:**

Security groups are virtual firewalls used to control access to the EC2 instance. We configured them to allow RDP and HTTP traffic, ensuring secure access to the server and website.

8. **Key Pair Authentication:**

AWS uses key pairs for secure login to EC2 instances. A key pair was created to authenticate and access the Windows server securely.

These technologies collectively enabled the successful deployment and management of a cloud-based website hosted on AWS EC2.

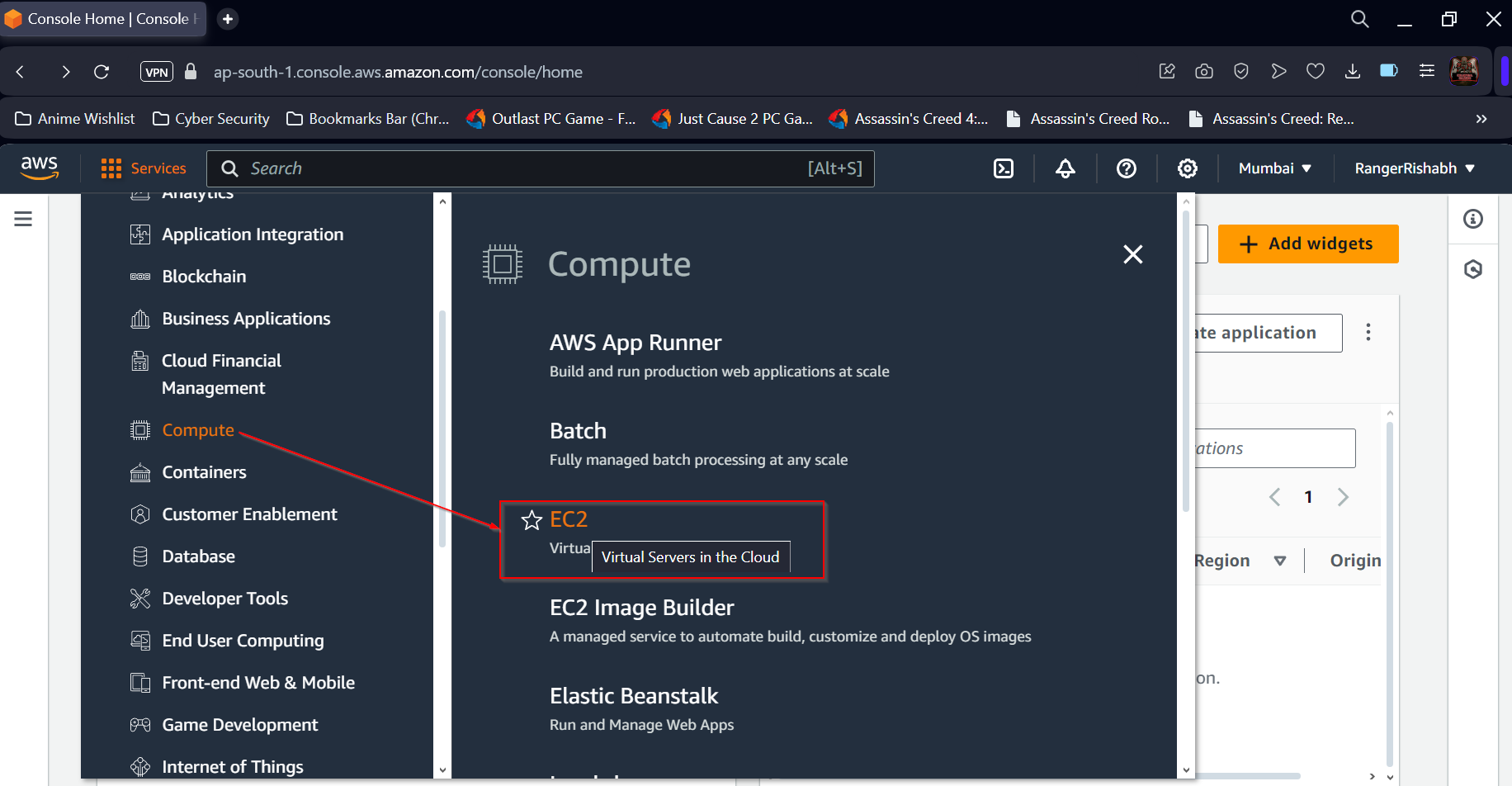
Project Steps with Screenshots

The following steps outline the complete process of hosting a website on AWS using an EC2 instance with a Windows server. Each step is accompanied by a detailed explanation and includes screenshots to guide you through the process.

**Step 1: Launching an EC2 Instance**

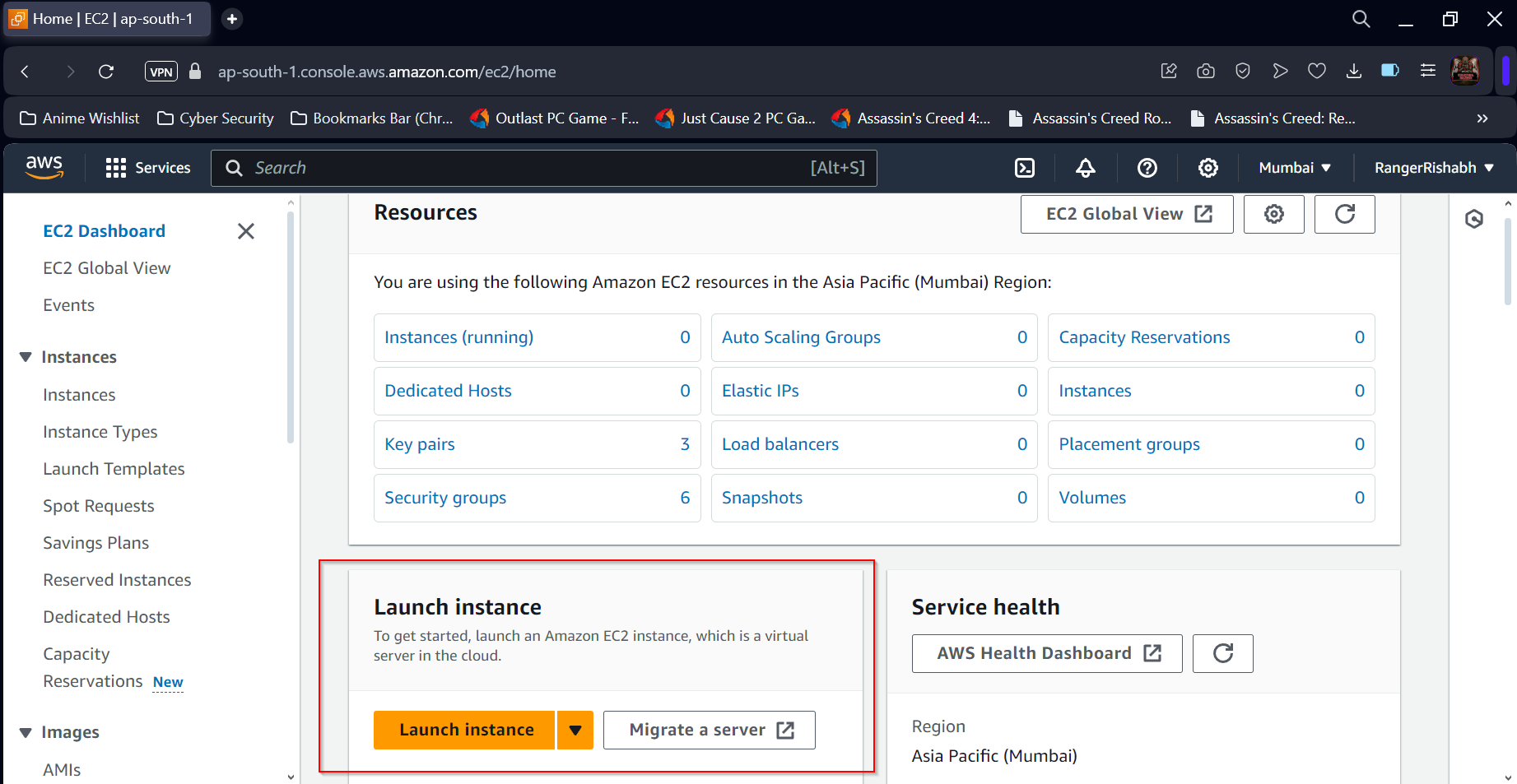
1. ***Login to AWS Management Console:***

* Begin by logging into the AWS Management Console. From the dashboard, navigate to the EC2 service under the Compute section.



1. ***Launch an Instance***

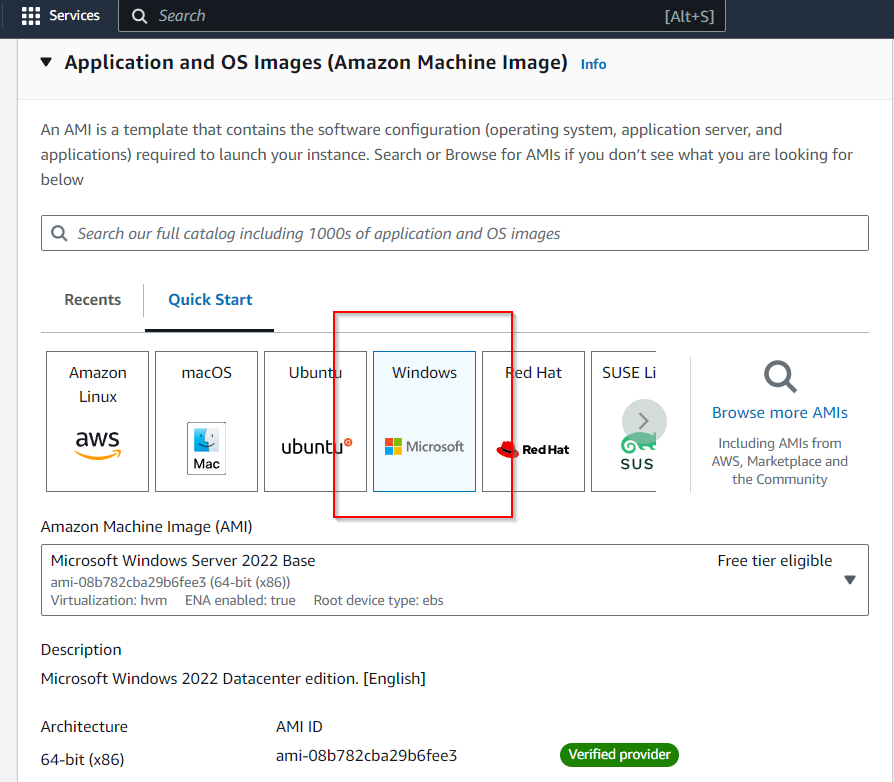
* Click on the “Launch Instance” button to start the process of creating a new EC2 instance. You'll be guided through several configuration steps to set up your instance.



* Write Name of the Server then proceed

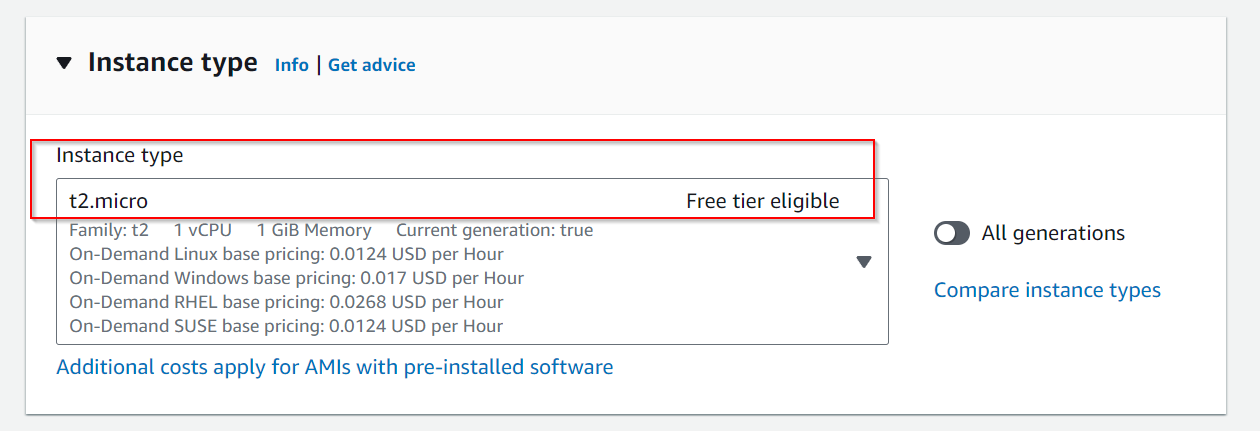
1. ***Choose an Amazon Machine Image (AMI):***

* Select a Windows Server AMI from the list of available options. Windows Server AMIs come preconfigured with the Windows operating system, making them suitable for web hosting.



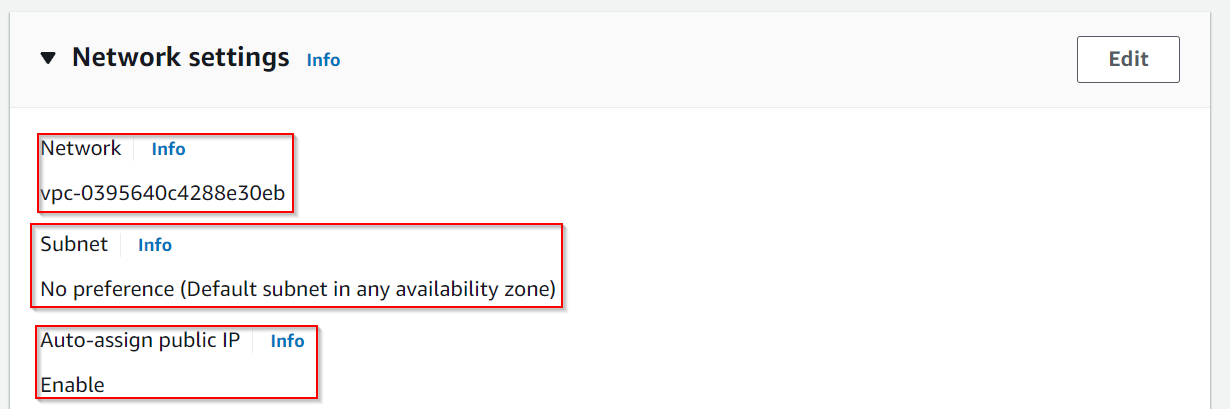
1. ***Select Instance Type:***

* Choose an instance type that fits your requirements. For this project, a t2.micro instance was selected as it falls under the AWS Free Tier and provides sufficient resources for basic website hosting.



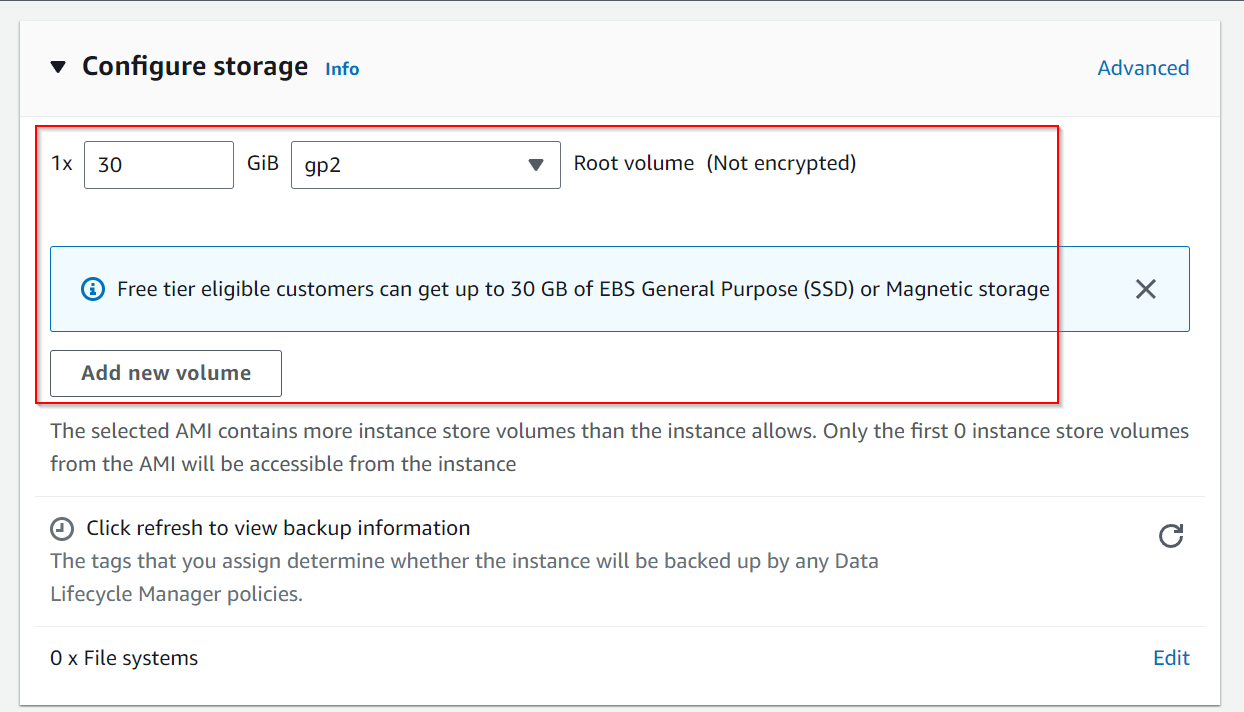
1. ***Configure Instance Details:***

* In this step, configure instance details such as the number of instances, network settings, and IAM roles. Ensure that the instance is set to launch in the correct Virtual Private Cloud (VPC) and subnet.



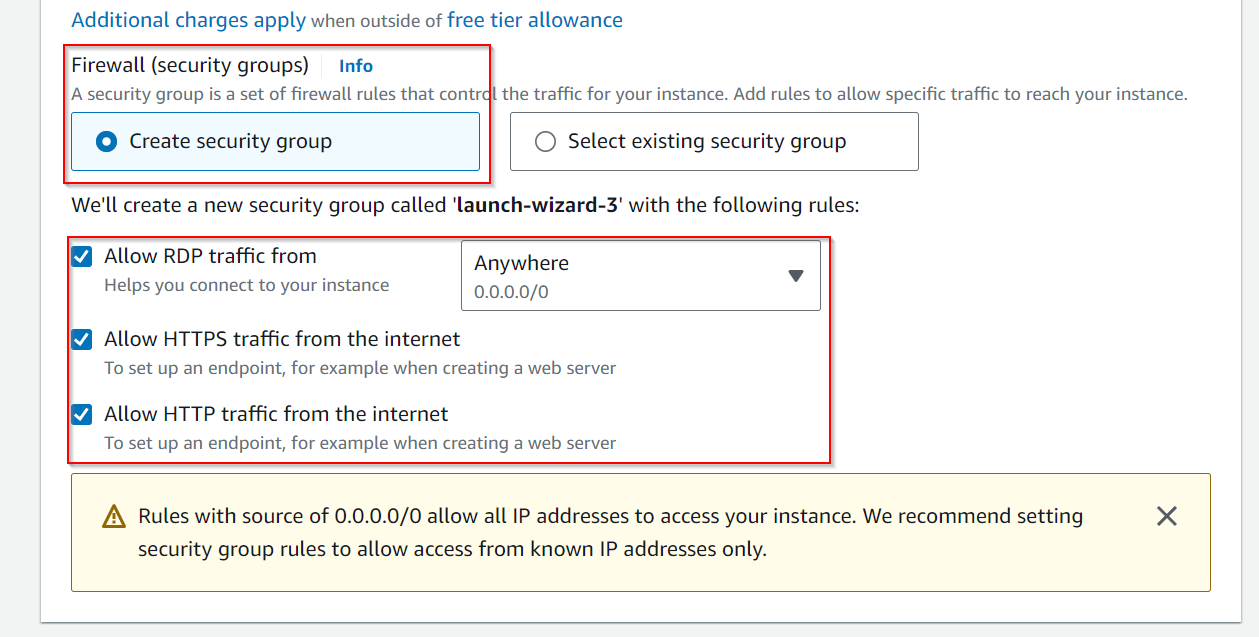
1. ***Add Storage:***

* Specify the storage for your instance. The default storage is usually sufficient, but you can adjust it based on the needs of your website (e.g., adding more disk space for larger sites).



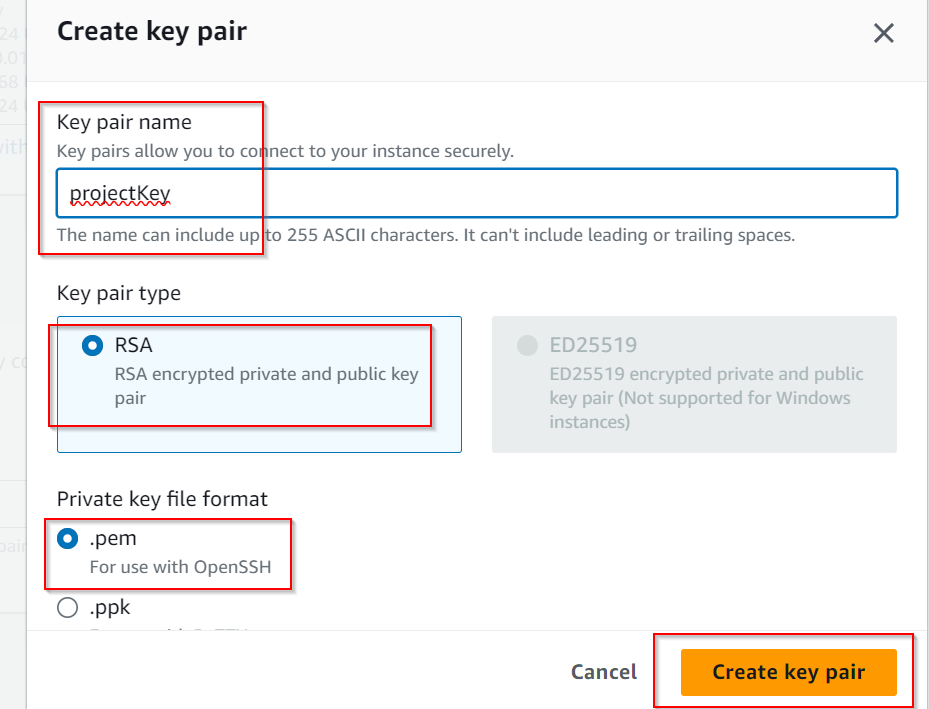
1. ***Configure Security Group:***

* Security groups act as virtual firewalls. Create a new security group or select an existing one. Make sure to allow inbound traffic for RDP (port 3389) to enable remote desktop access and HTTP (port 80) for web traffic.

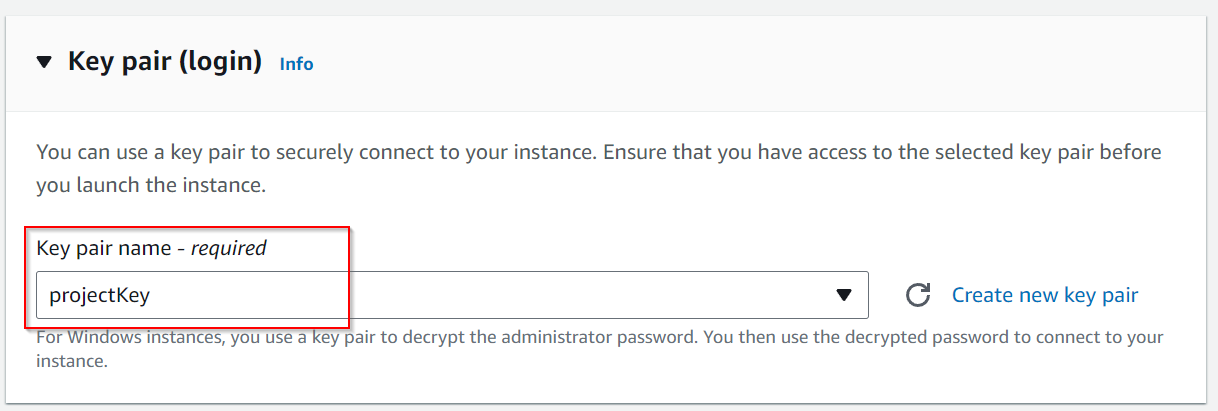


1. ***Key Pair Creation:***

* AWS EC2 uses key pairs for secure access to instances. Create a new key pair or use an existing one. Download the private key (.pem) file, as it will be required to access the instance later.

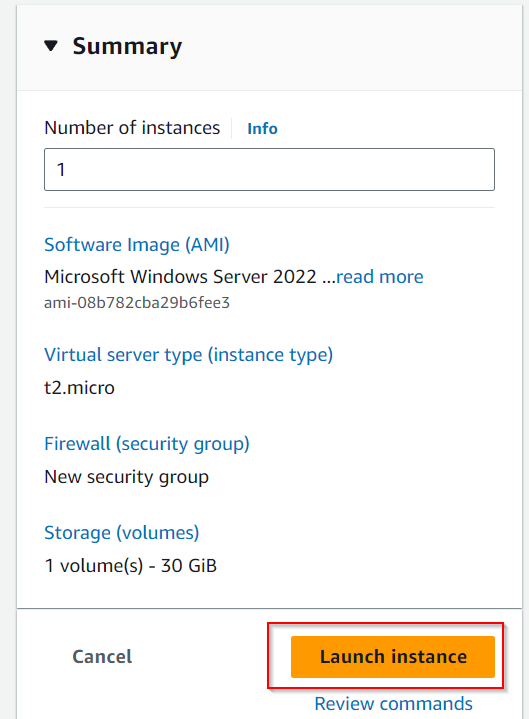


* Type key pair name
* Select key type as RSA
* Select key format as .pem



1. ***Review and Launch:***

* Review all the configured settings and click on "Launch" to start the instance. It may take a few minutes for the instance to be fully initialized.



**Step 2: Connecting to the EC2 Instance**

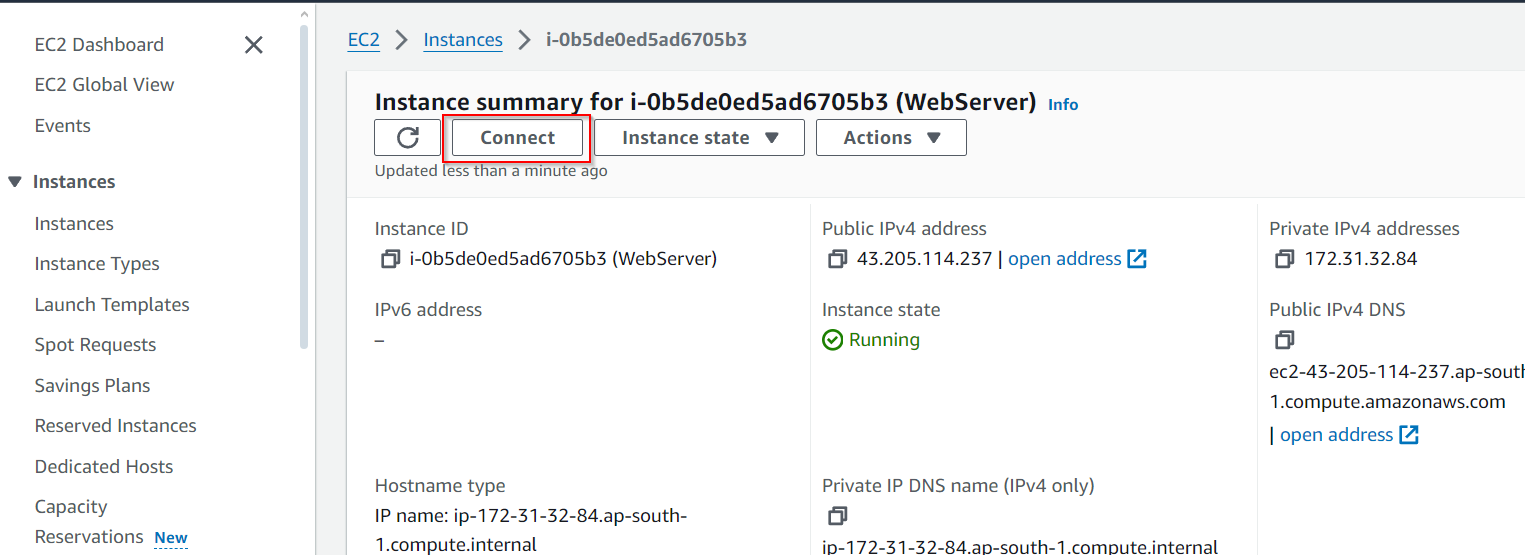
1. ***Obtain Public IP Address:***

* Once the instance is running, navigate to the EC2 dashboard and note the public IP address or DNS name of your instance. This address will be used to access the instance and the hosted website.

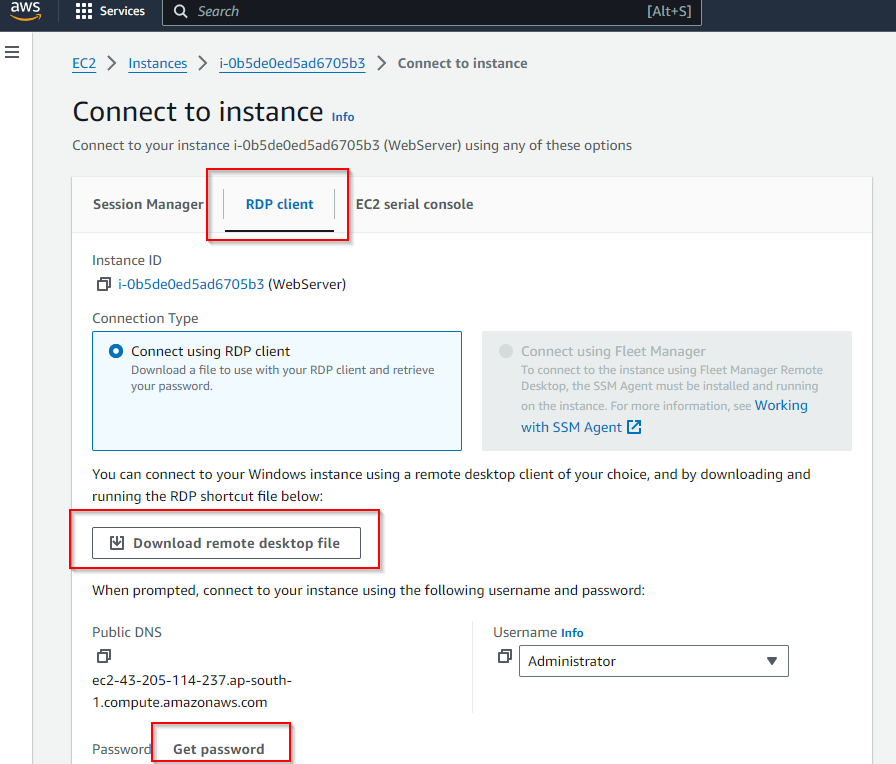


1. ***Connect via Remote Desktop Protocol (RDP):***

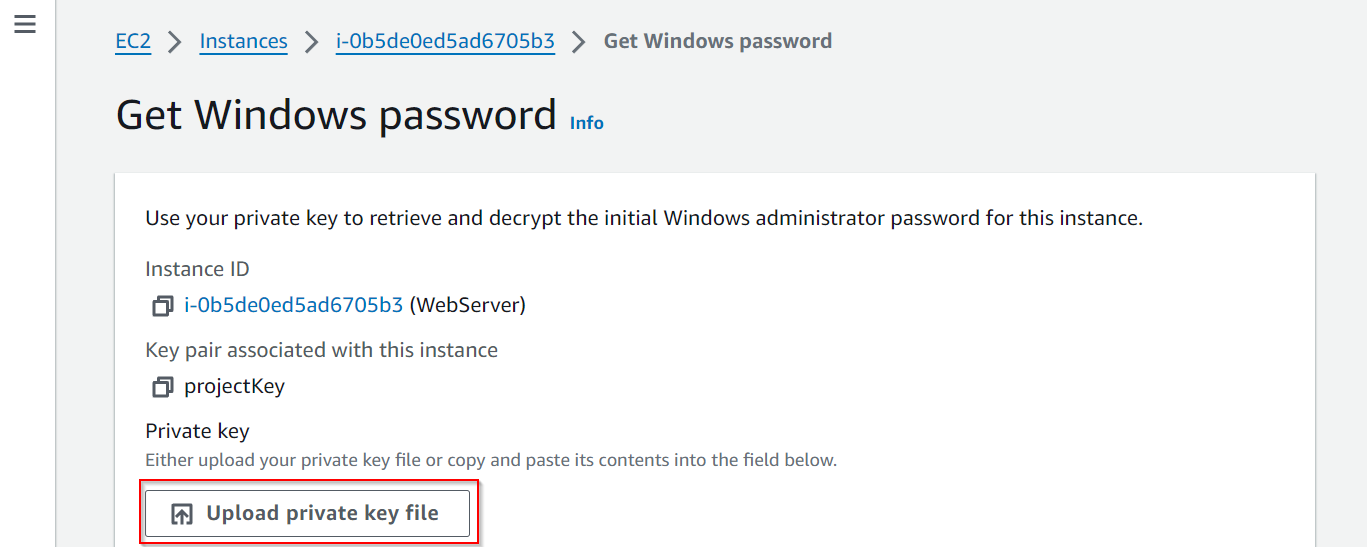
* Open an RDP client on your local machine. Enter the public IP address of the EC2 instance. Use the credentials generated from the key pair file to log in to the Windows Server. This will give you full access to the server environment.



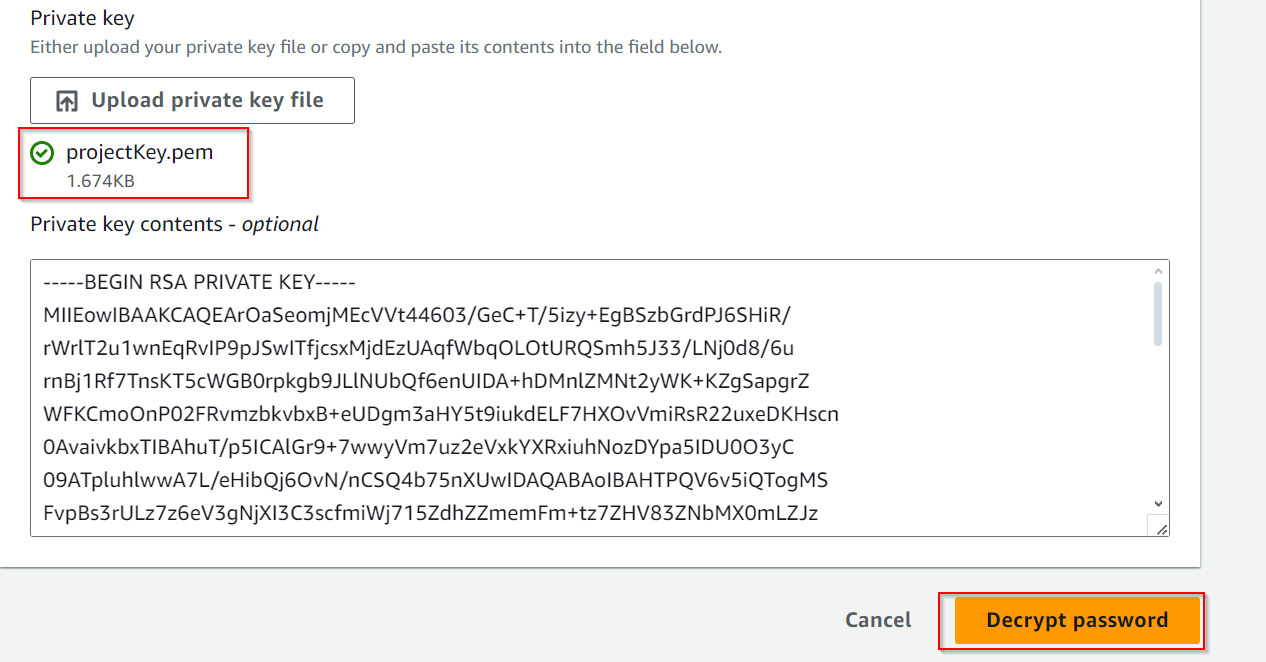
* Click on connect button



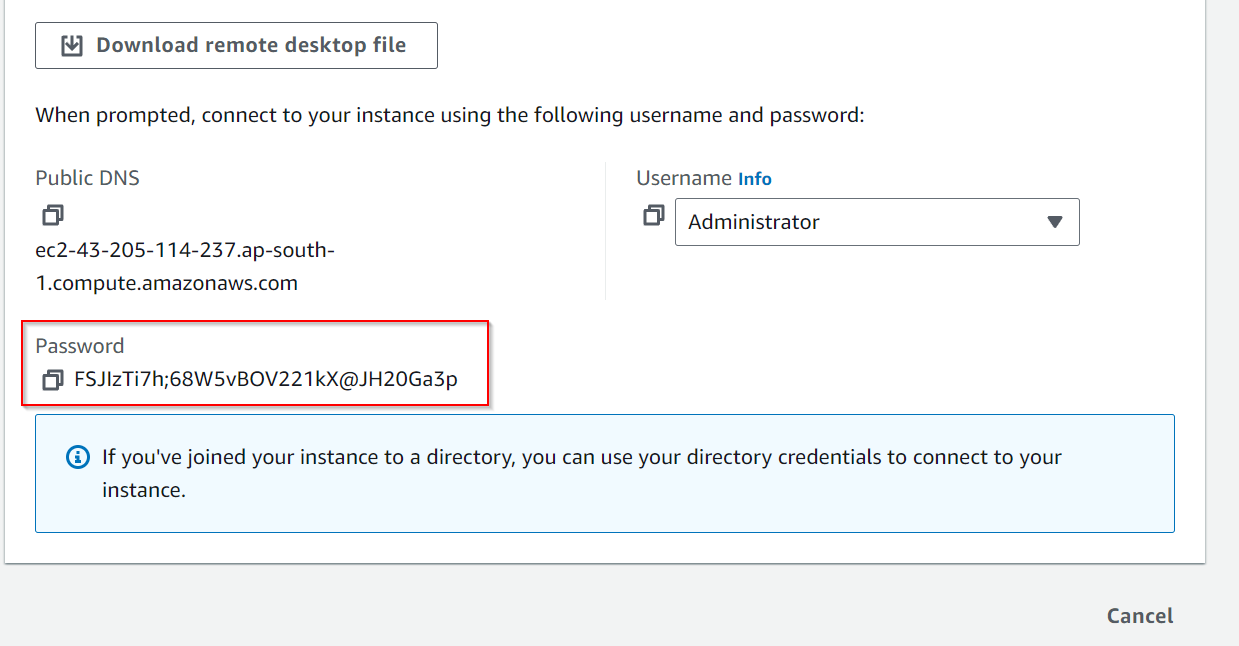
* Go to RDP client section
* Download remote desktop file then select “Get Password”

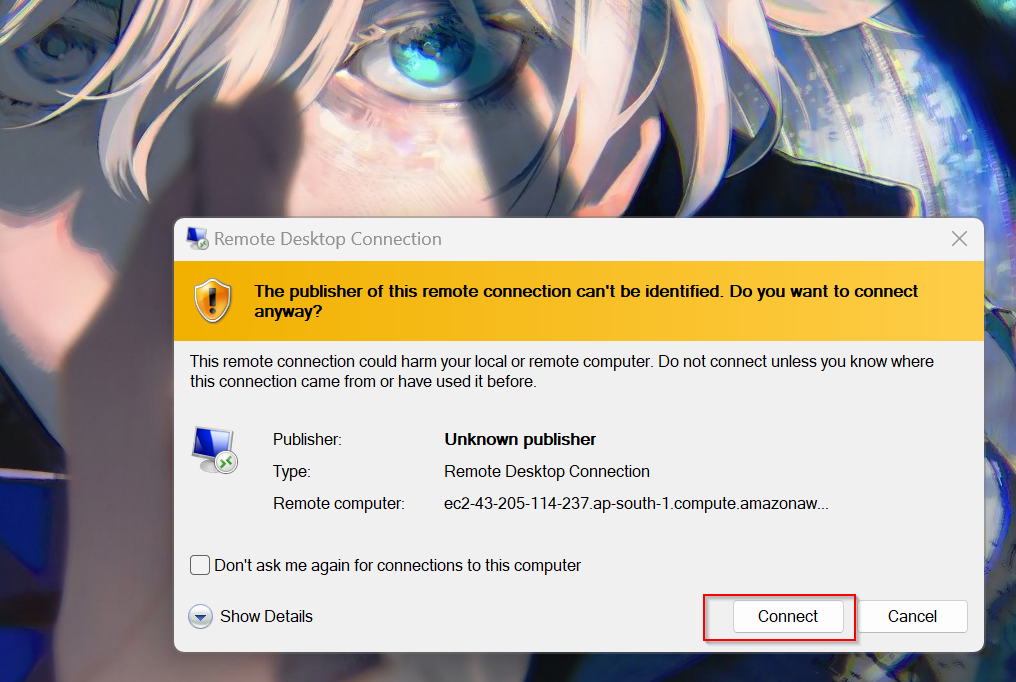


* Upload private key file which was downloaded earlier “projectKey.pem”

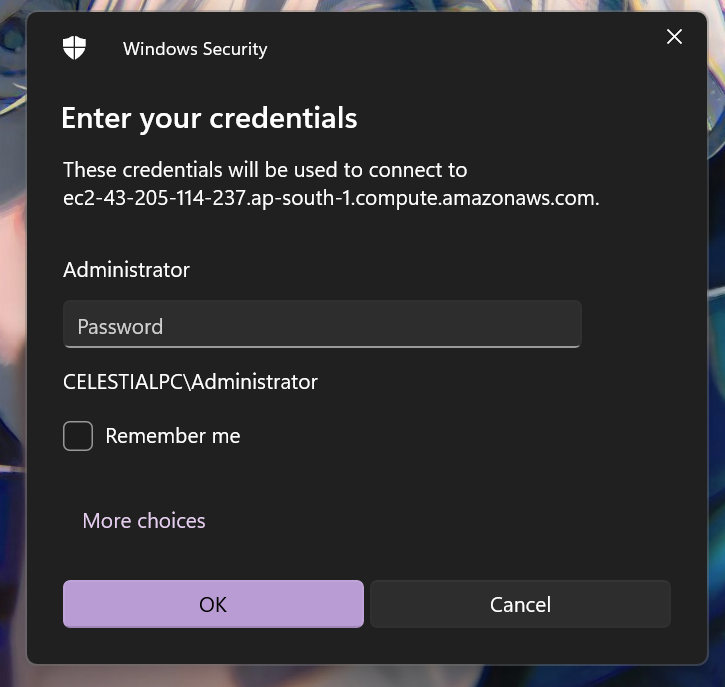


* Select decrypt password to obtain password for remote desktop





* Connect to the instance by opening remote desktop file

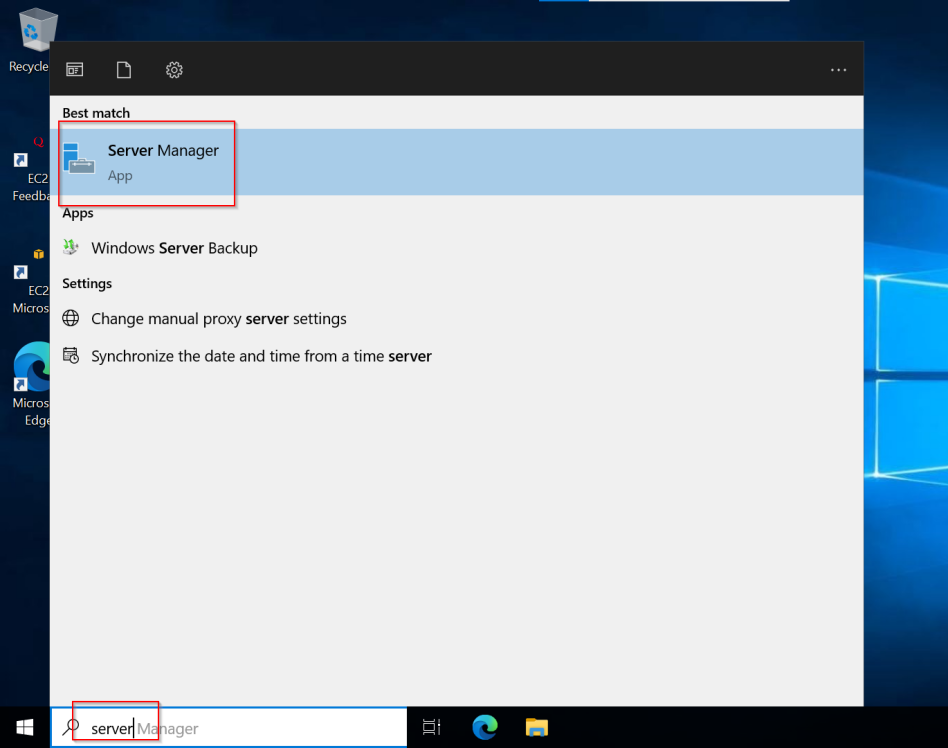


* Enter the password obtained before

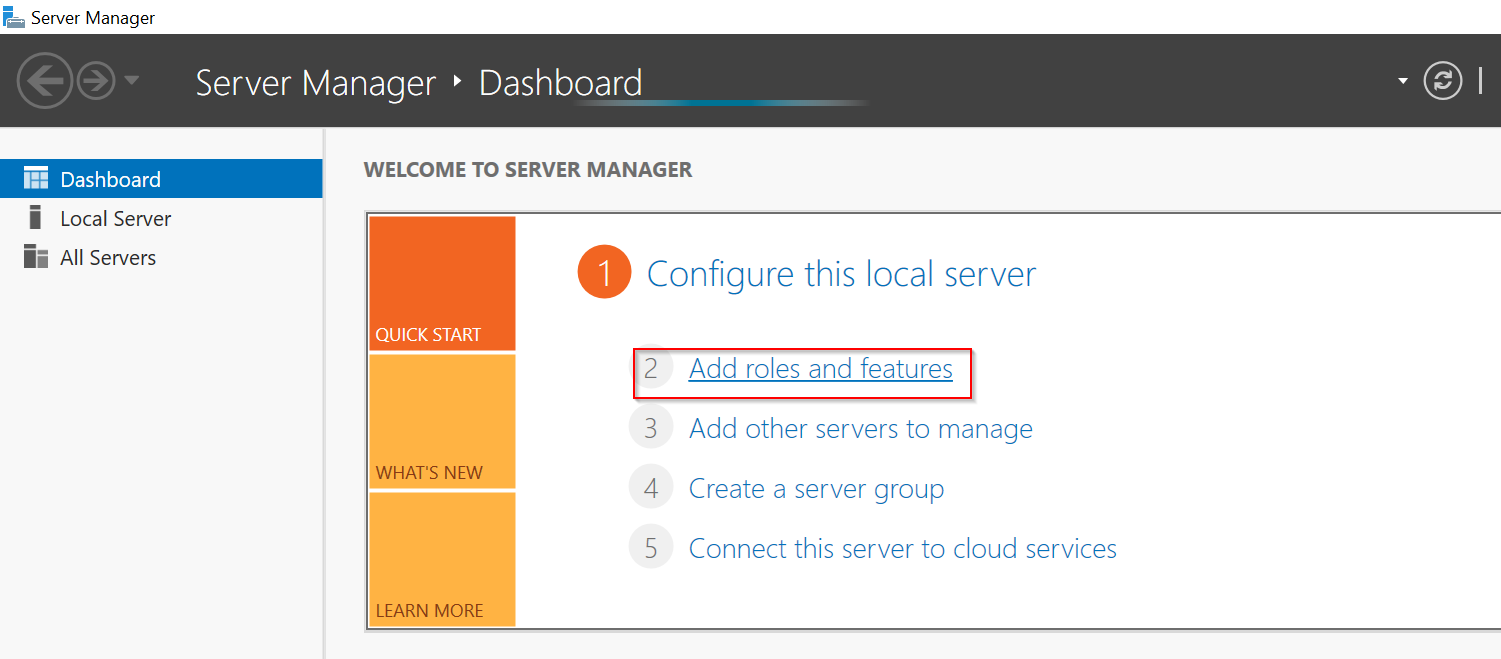
**Step 3: Configuring the Web Server**

1. ***Install Internet Information Services (IIS):***

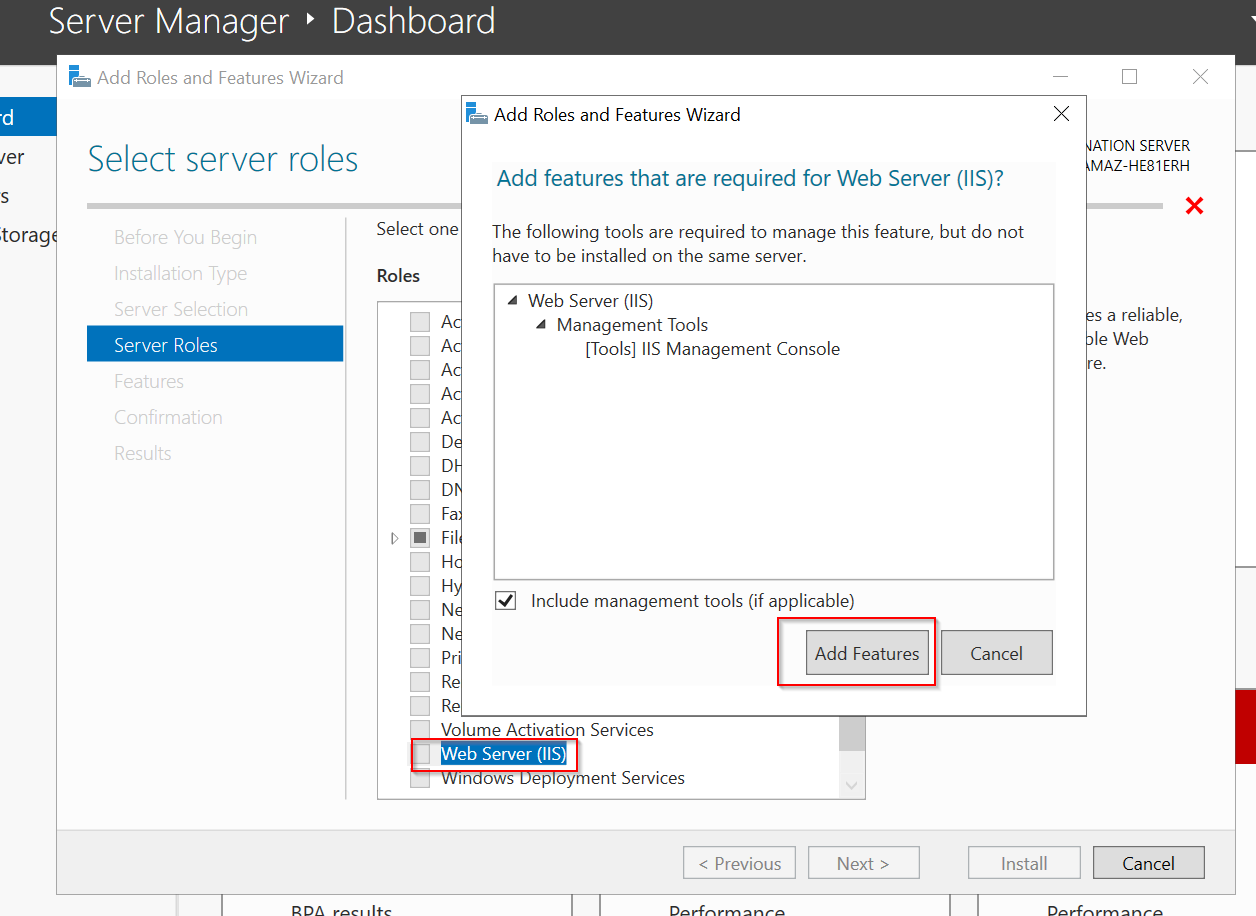
* After connecting to the instance, open the Server Manager dashboard. Navigate to "Add Roles and Features" and select Internet Information Services (IIS). Proceed with the installation, which may take a few minutes.



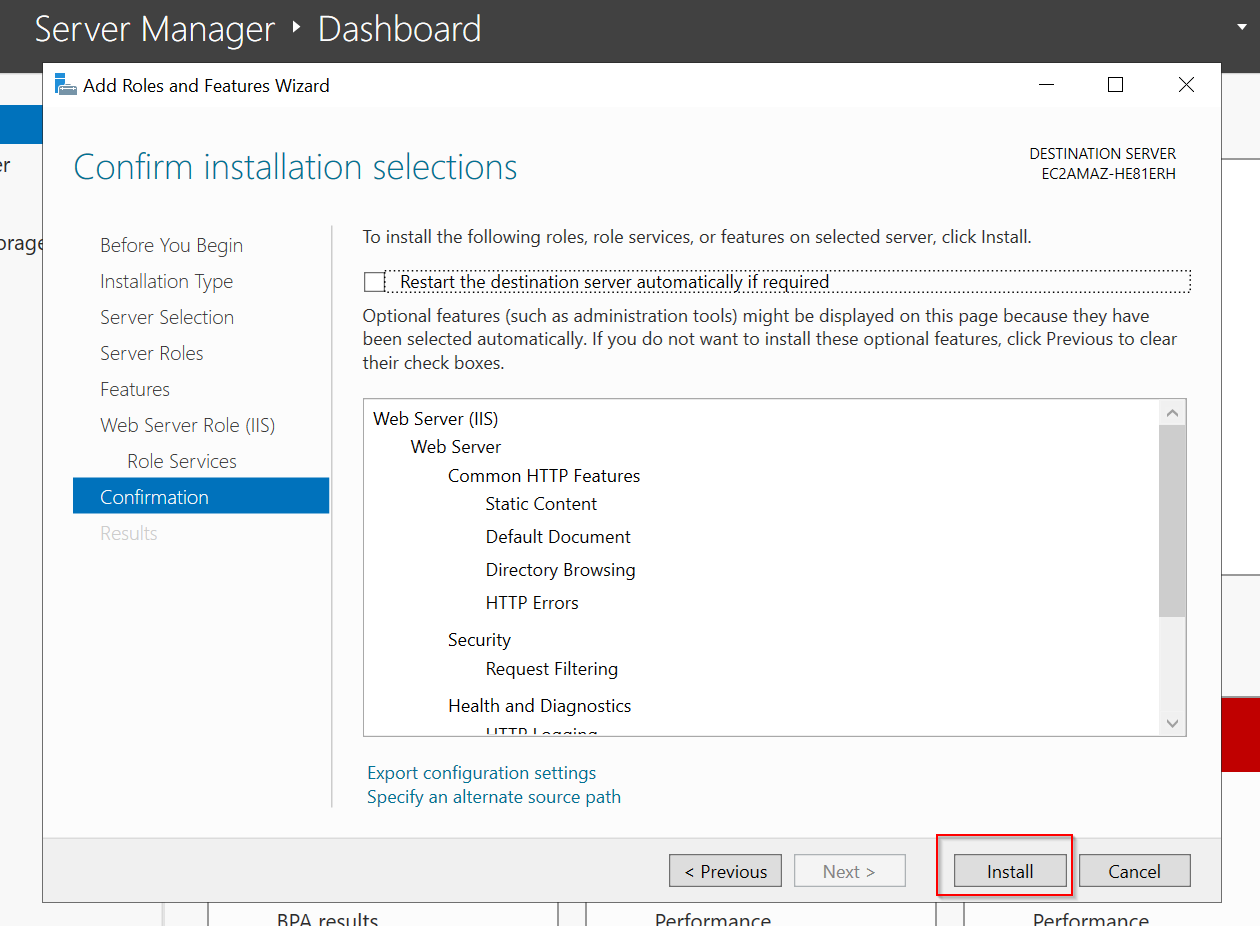
* Search server manager and select the Server Manager App



* Wait for the dashboard to get initalized then select “Add roles and features” option



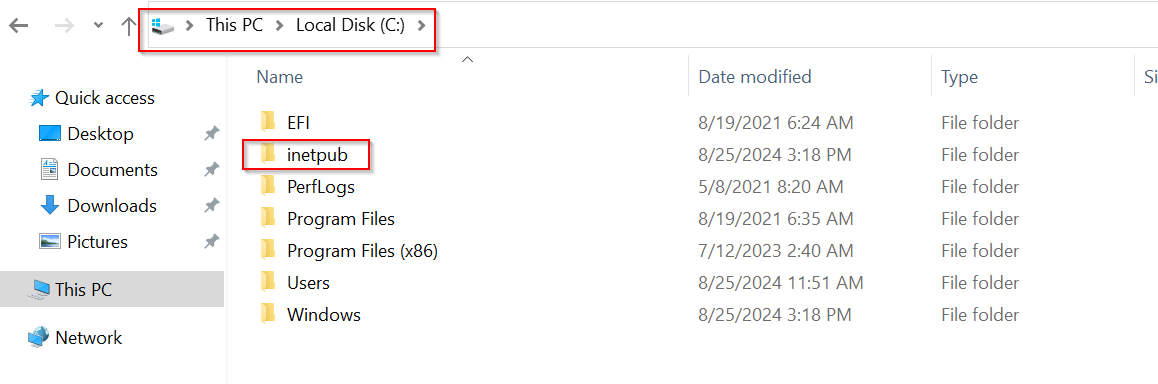
* Keep scrolling by clicking Next until we reach server roles, select Web Server(IIS) from the list and then click on Add features

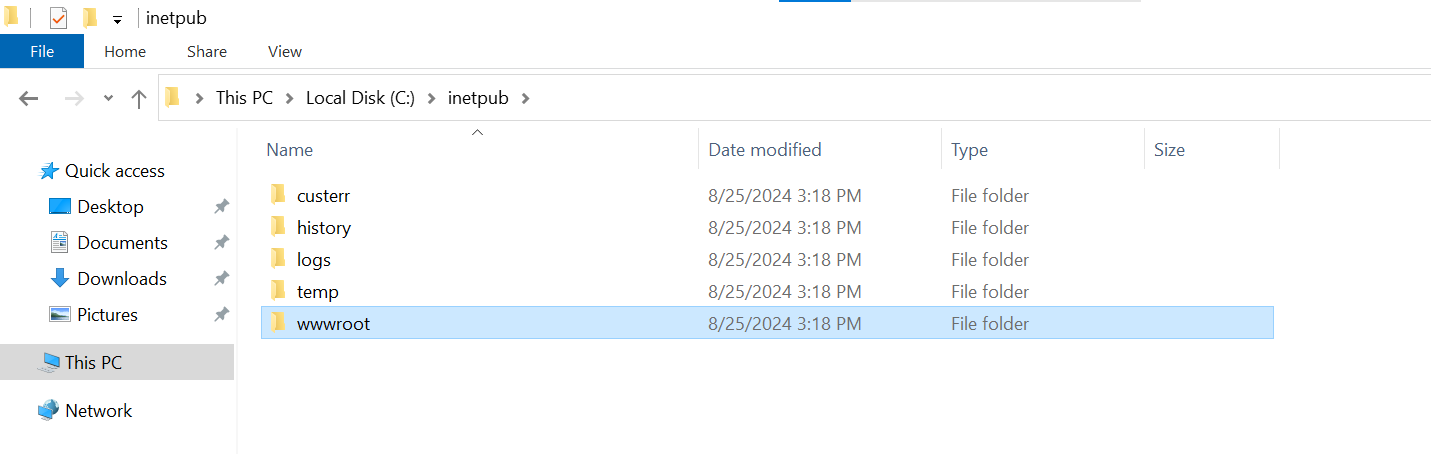


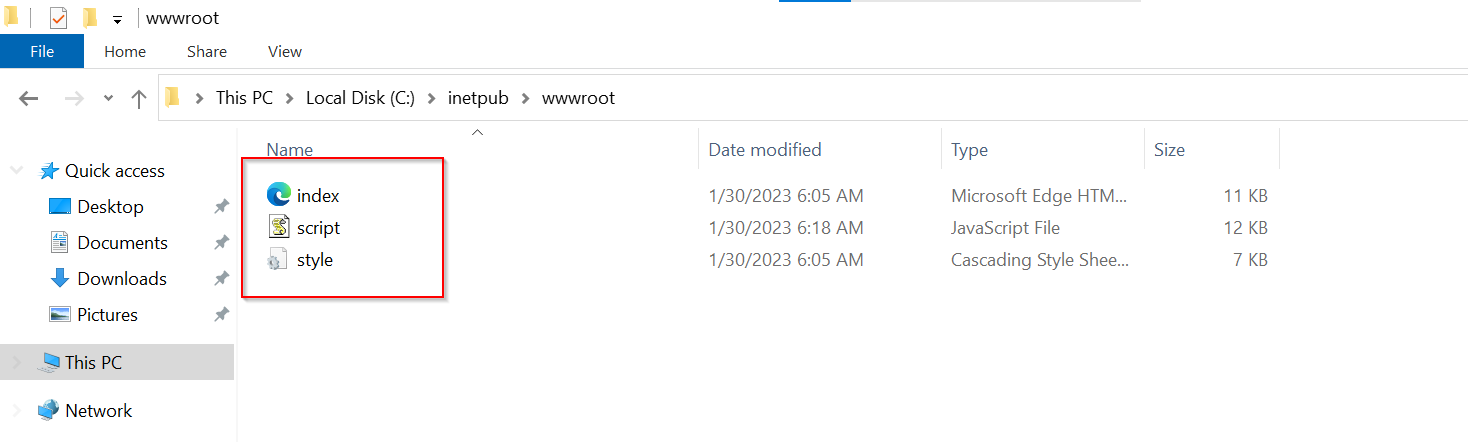
* Then scroll till the end by clicking Next, once completed click on Install button

1. ***Upload Website Files:***

* Use Remote Desktop to upload your website files to the appropriate directory on the server. Typically, the files should be placed in the "C:\inetpub\wwwroot" directory, which is the default location for IIS websites.



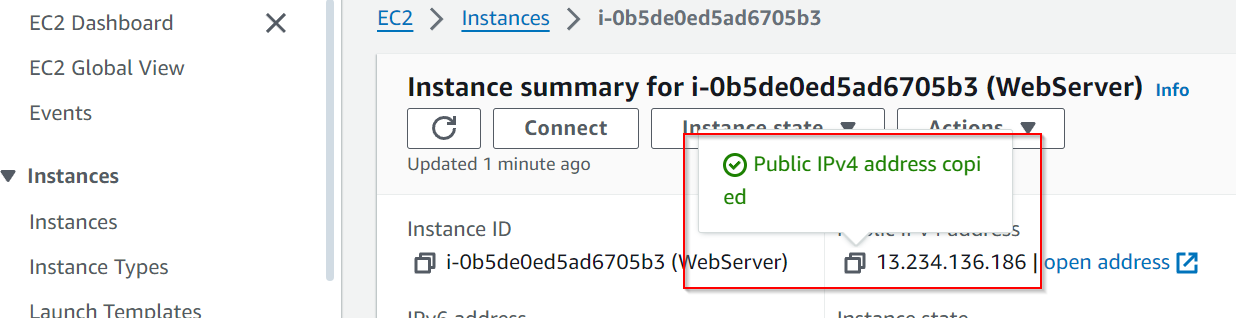


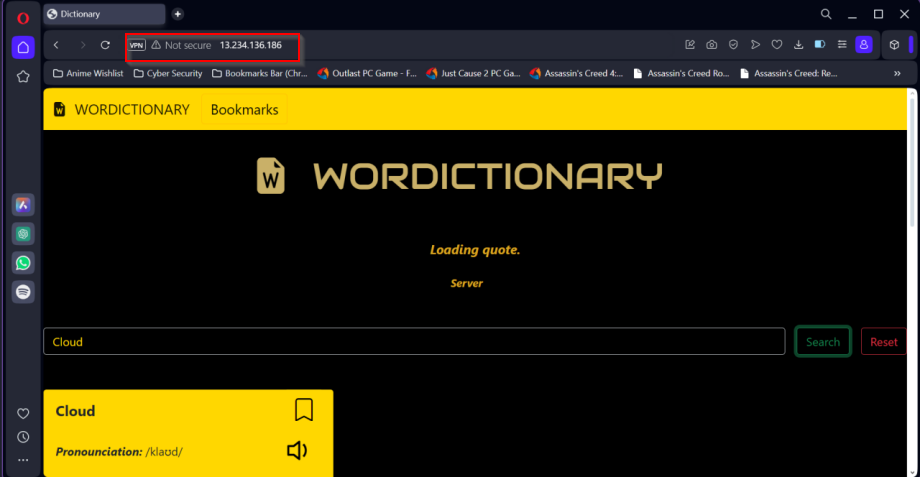


* On reaching the location, place files to be displayed on the web server

1. ***Test the Website:***

* Open a browser on your local machine and enter the public IP address of your EC2 instance. If everything is configured correctly, your website should be accessible online. You can also verify that IIS is serving the correct content by viewing the default IIS start page.





* On opening the public IP on the browser, the website will be displayed

Conclusion

This project demonstrates the effective use of Amazon Web Services (AWS) Elastic Compute Cloud (EC2) for hosting a website on a Windows server. By leveraging AWS EC2, we have successfully deployed a scalable, reliable, and cost-effective web hosting solution, showcasing the flexibility and robustness of AWS’s cloud infrastructure.

Throughout this project, we meticulously followed several critical steps, from launching an EC2 instance to configuring the server and hosting the website. The process began with setting up an EC2 instance using a Windows Server AMI, followed by configuring the instance details and security groups to ensure appropriate network access. By connecting to the instance via Remote Desktop Protocol (RDP), we were able to install Internet Information Services (IIS), a web server software, and upload our website files to the server.

Key takeaways from this project include:

- **Hands-on Experience:** This project provided practical experience in setting up and managing a cloud-based server, enhancing our understanding of cloud computing and web hosting.

- **Comprehensive Guide:** The detailed steps and screenshots included in this report serve as a valuable resource for anyone looking to replicate the process, making it easier to deploy a website on AWS EC2.

- **Flexibility:** AWS EC2's ability to support various configurations and software installations highlights its adaptability for different web hosting needs and applications.

- **Transferable Skills:** The skills and knowledge gained from this project are applicable to a wide range of cloud-based solutions, beneficial for personal projects, business applications, and large-scale enterprise deployments.

In addition to these benefits, the hands-on experience gained from this project enhances our understanding of cloud computing and web hosting. The detailed steps and screenshots provided in this report serve as a practical guide, making it easier for others to replicate the process and deploy their own websites using AWS EC2.

Furthermore, the skills and knowledge acquired through this project are transferable to a wide range of cloud-based solutions. Whether it’s for personal projects, business applications, or large-scale enterprise deployments, the principles and practices demonstrated here can be applied to various scenarios, highlighting the versatility of AWS EC2.

In conclusion, hosting a website on AWS EC2 with a Windows server has proven to be an efficient and effective solution. By following the outlined steps, we have successfully deployed a web server, demonstrating the power and flexibility of AWS’s cloud computing platform. This project not only achieves its primary goal but also serves as a valuable learning experience, equipping us with the skills and knowledge to leverage AWS for future web hosting and cloud computing endeavors.