**Sysops Project**

**Setting Up a Website on Cloud – Course-End Project 2**

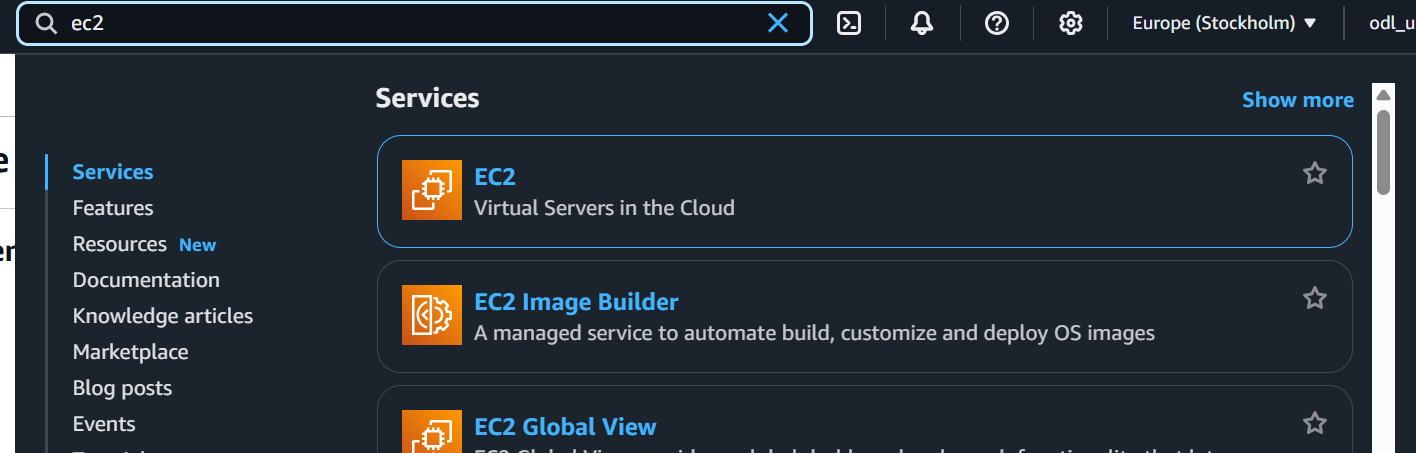
**Introduction**

As a Cloud Engineer, our task is to set up a web server on AWS Cloud for a new client. We will be creating an EC2 instance, installing IIS, deploying a static website, and configuring a load balancer. This ensures that the client's website is accessible and scalable. The following steps will guide us through the entire process.

**Step 1: Creating an EC2 Instance (Windows Server 2012 R2 Base)**

**1.1 Log in to AWS Console**

* Navigate to **AWS Management Console** and sign in.
* Open **EC2 Dashboard** by searching for "EC2" in the AWS services search bar.

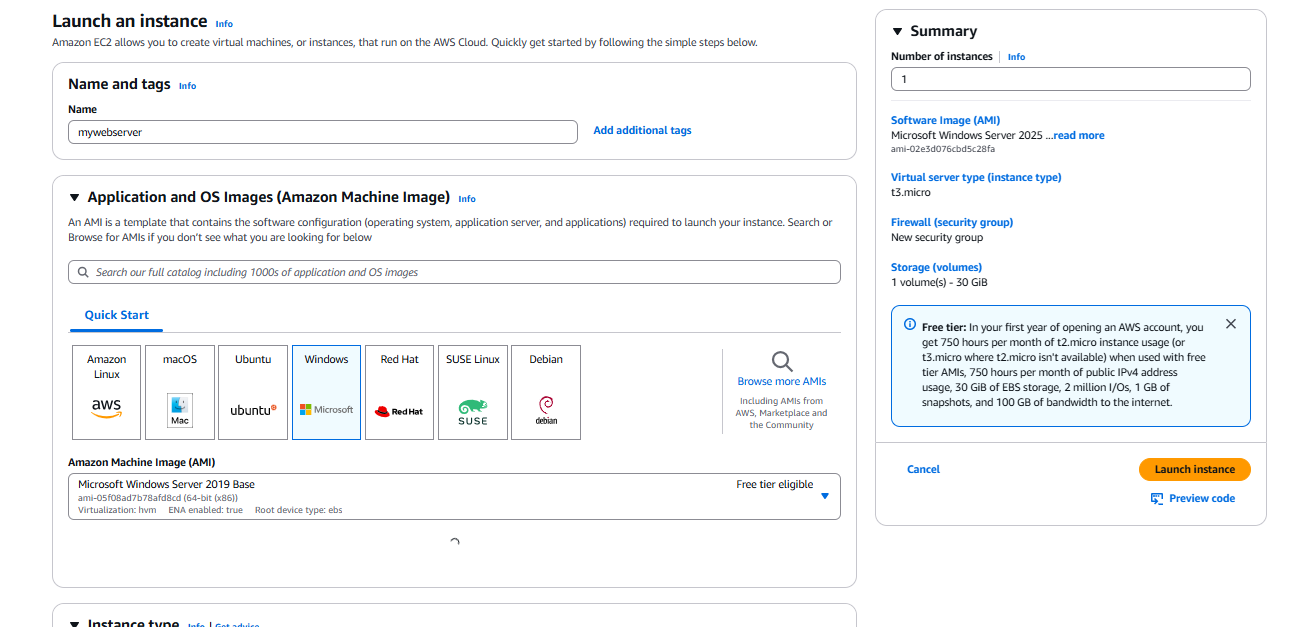


**1.2 Launch an EC2 Instance**

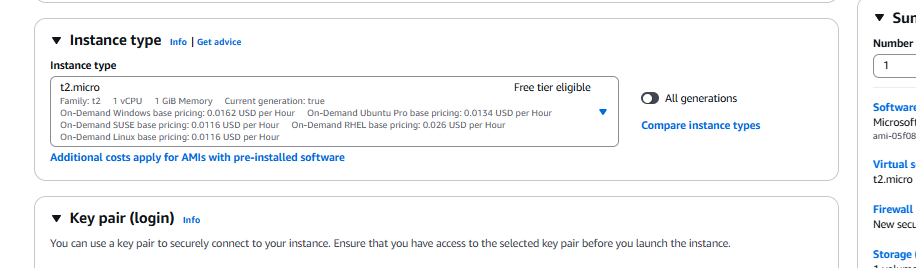
* Click on **Launch Instance**.



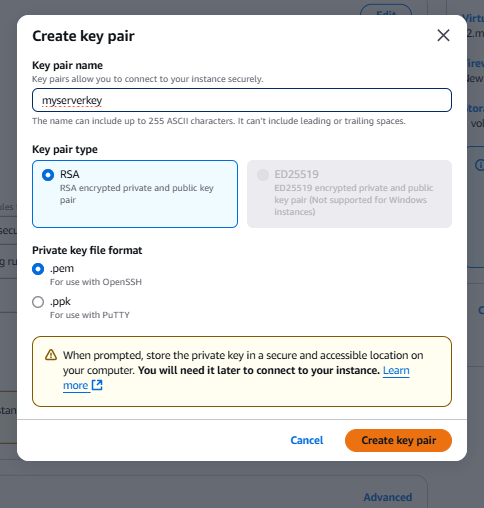
* Choose the **Amazon Machine Image (AMI)**: Select **Windows Server 2012 R2 Base**.



* Choose an instance type: Select **t2.micro** (for free-tier) or a higher tier based on client requirements.

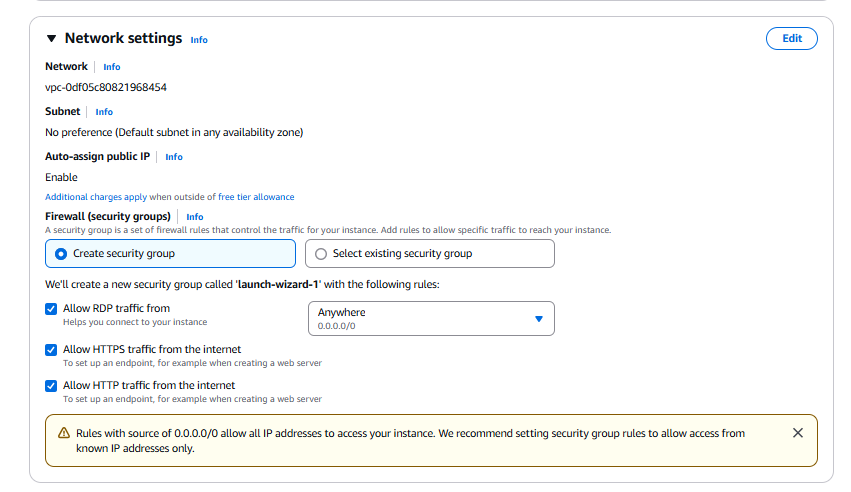


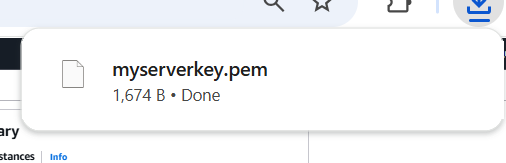
* Configure instance details:
  + Set the **number of instances** to 1.
  + Select **default VPC** or create a new VPC if required.
  + Assign a public IP address.
  + Launch Key pair



**1.3 Configure Security Group**

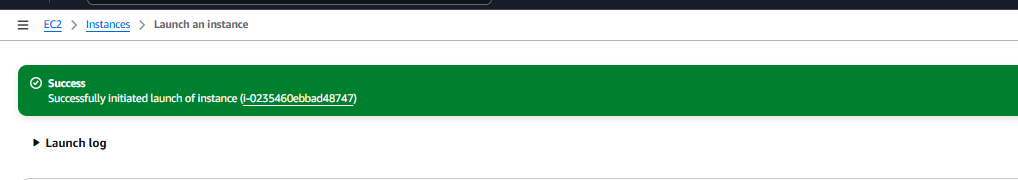
* Create a new **Security Group** with the following rules:
  + **RDP (TCP 3389)** → Allow from **your IP** (for remote access).
  + **HTTP (TCP 80)** → Allow from **Anywhere (0.0.0.0/0)** (to access the website).
  + **HTTPS (TCP 443)** → Allow from **Anywhere (0.0.0.0/0)** (for secure browsing).



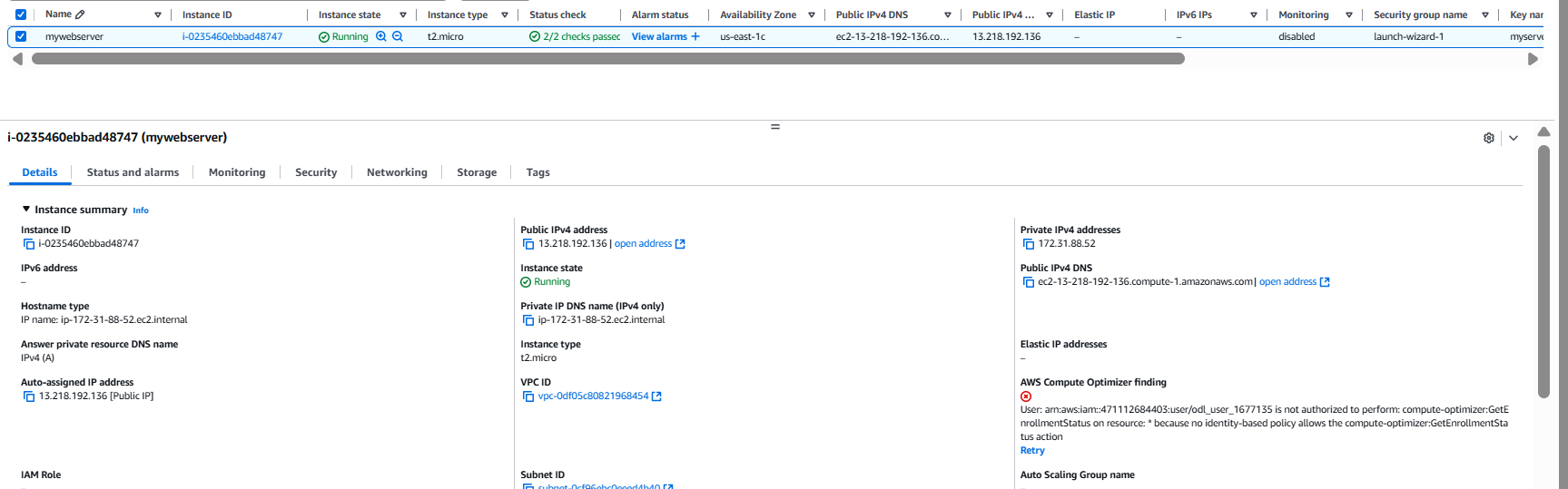


**1.4 Launch and Connect to the Instance**

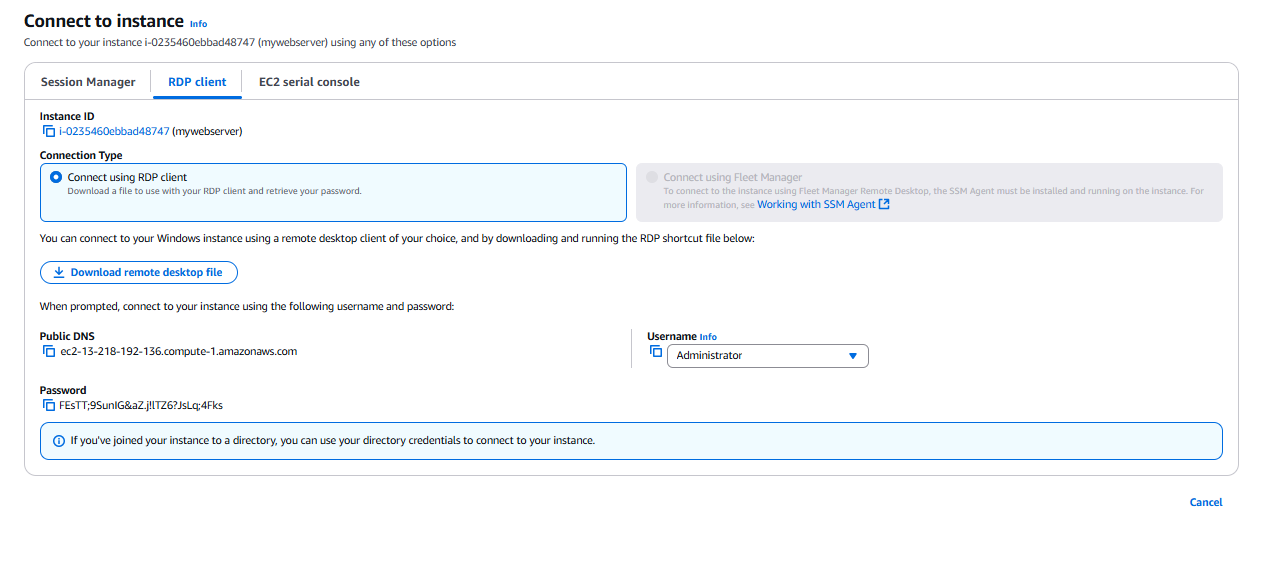
* Click **Launch**, then create



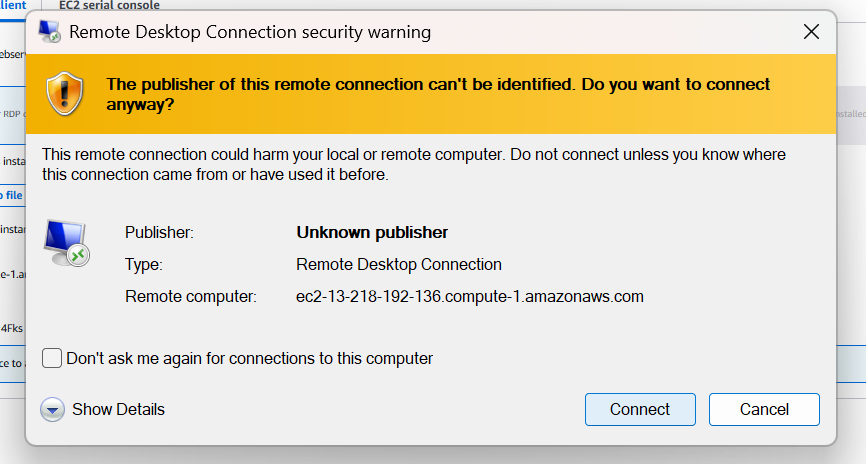
* Once the instance is running,

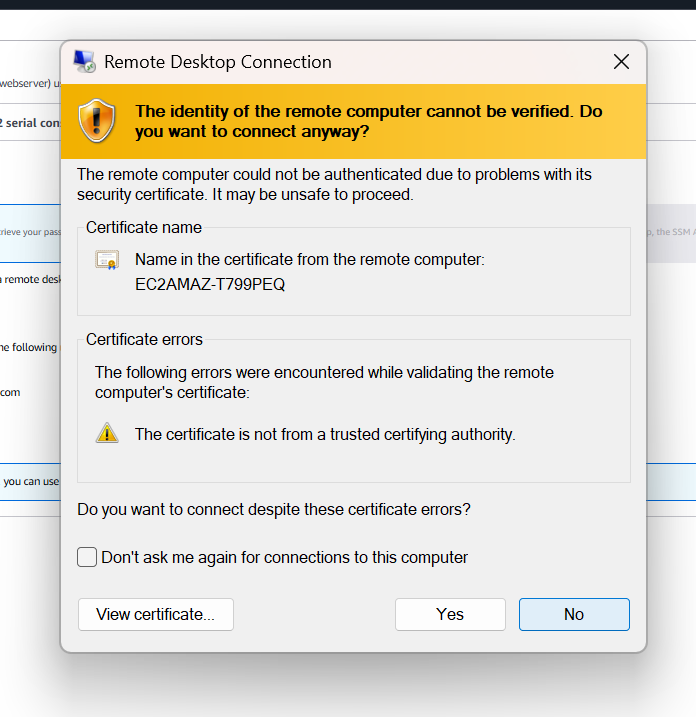


* connect using **Remote Desktop Protocol (RDP)**.



After decrypting the password using the key pair , connect to the server as shown below:



Click yes on the screen below  


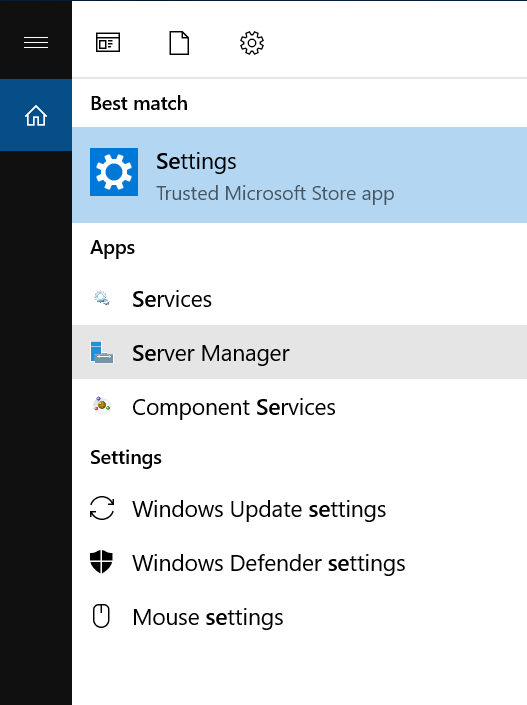
Finally connected to the server via RDP as shown below:



**Step 2: Installing IIS Server Role**

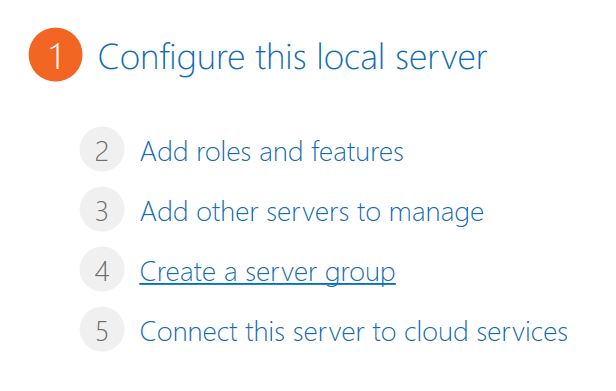
**2.1 Open Server Manager**

* After connecting to the instance via RDP, open **Server Manager** from the start menu.

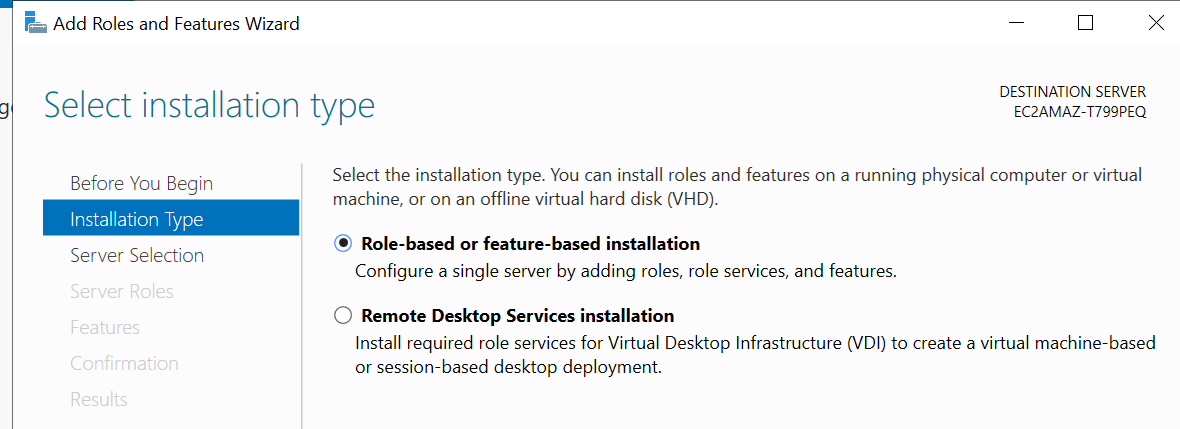


**2.2 Add IIS Role**

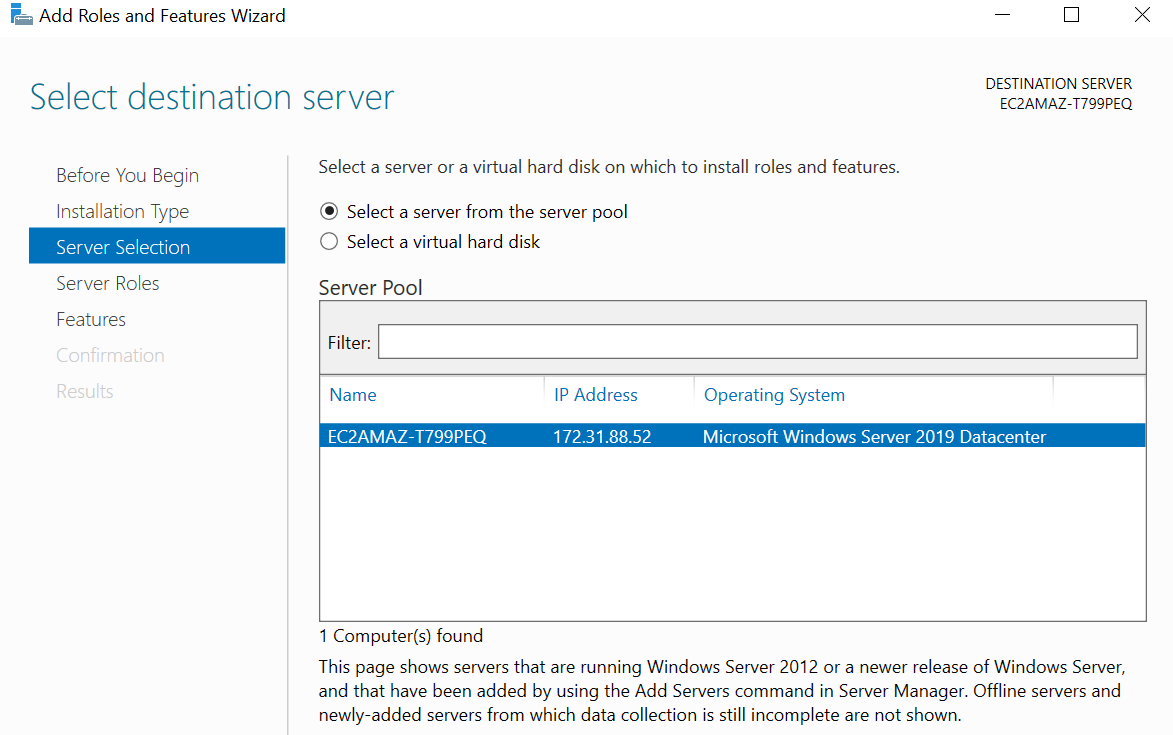
* Click **Manage** → **Add Roles and Features**.



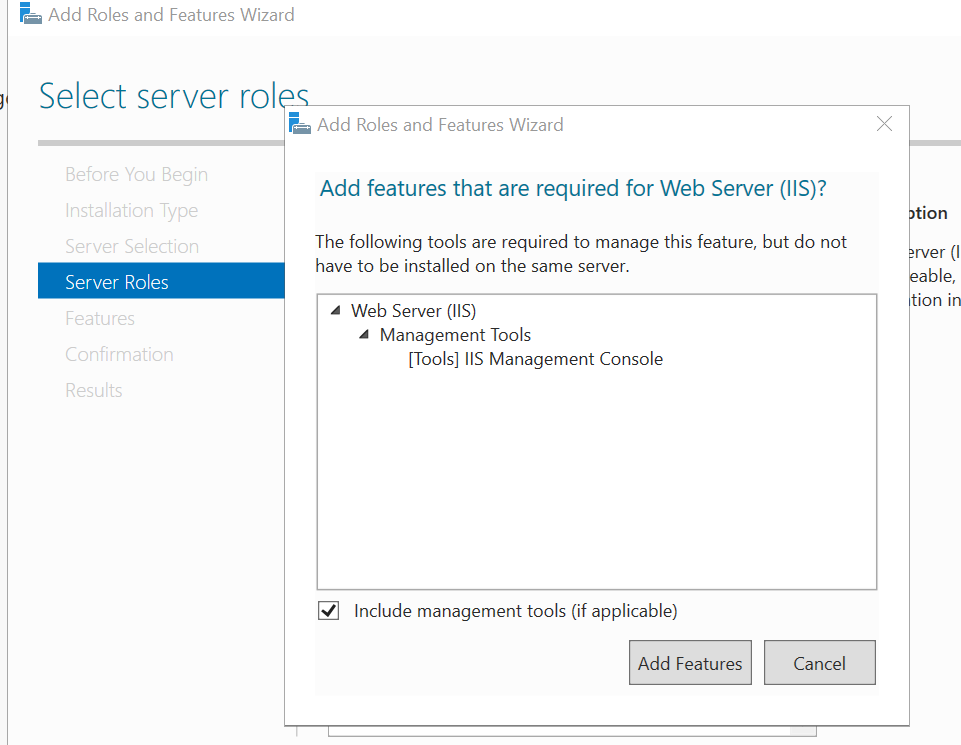
* Choose **Role-based or feature-based installation**.

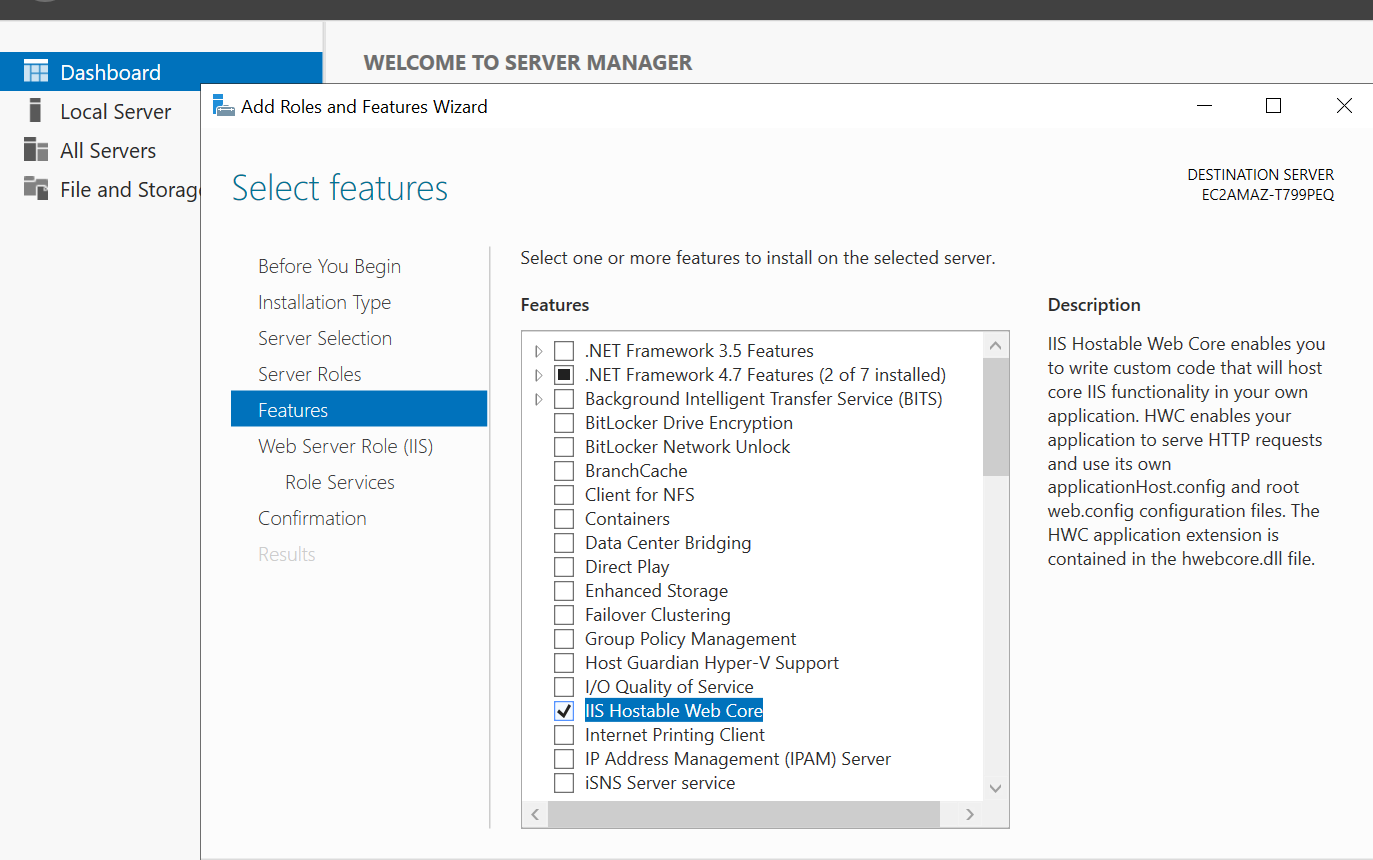


* Select **Windows Server** instance.

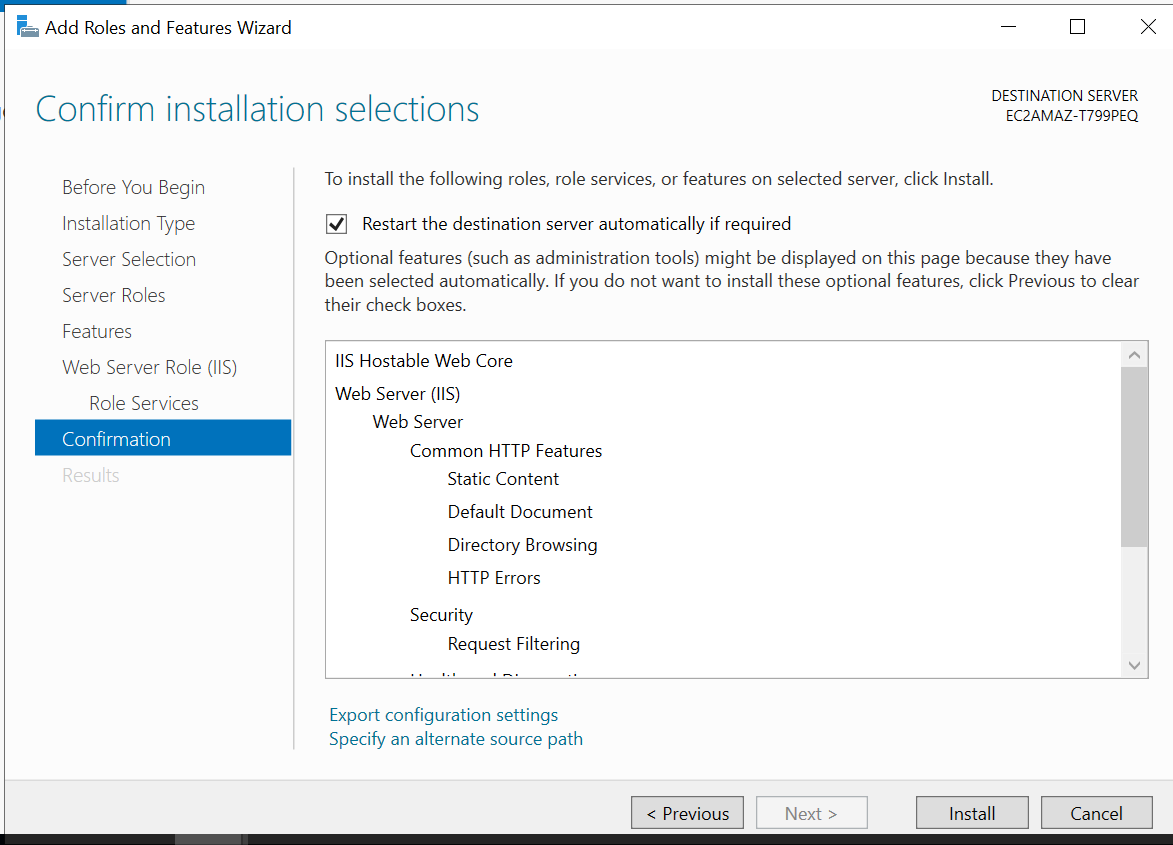


* Under **Server Roles**, select **Web Server (IIS)**.

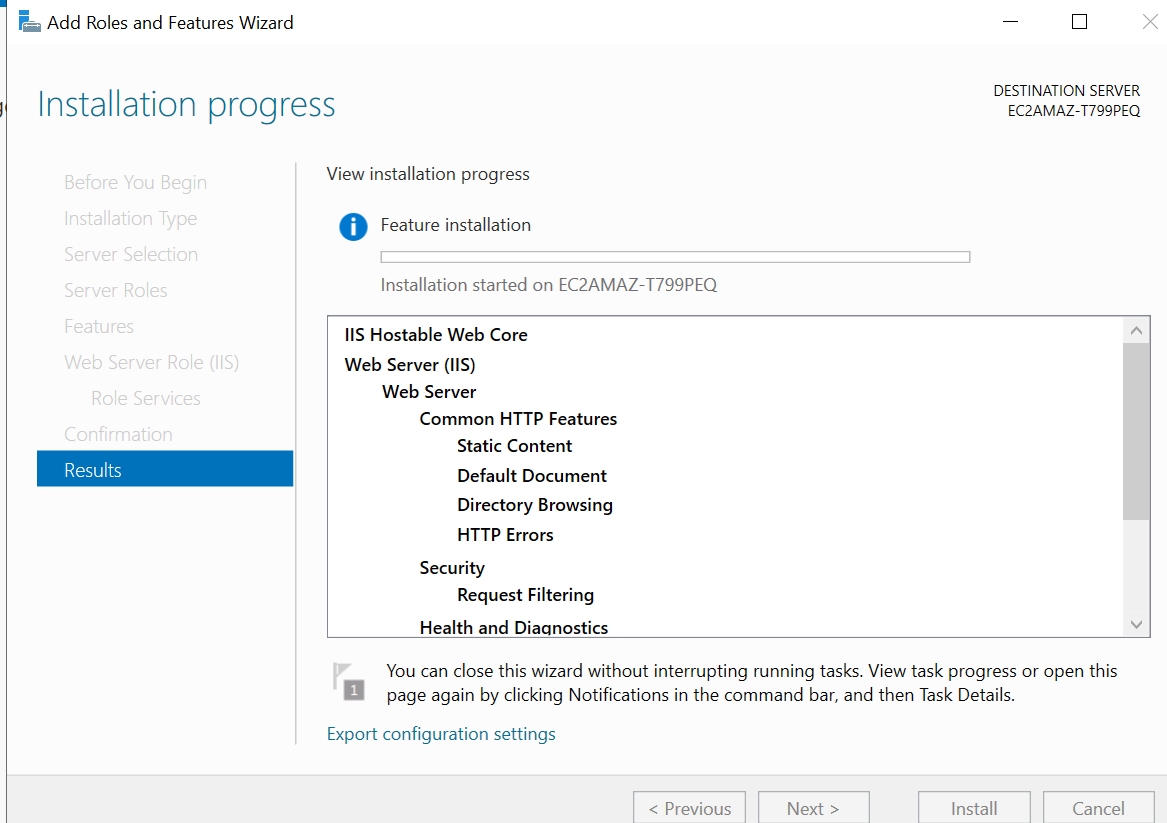




* Click **Next**, then **Install**.



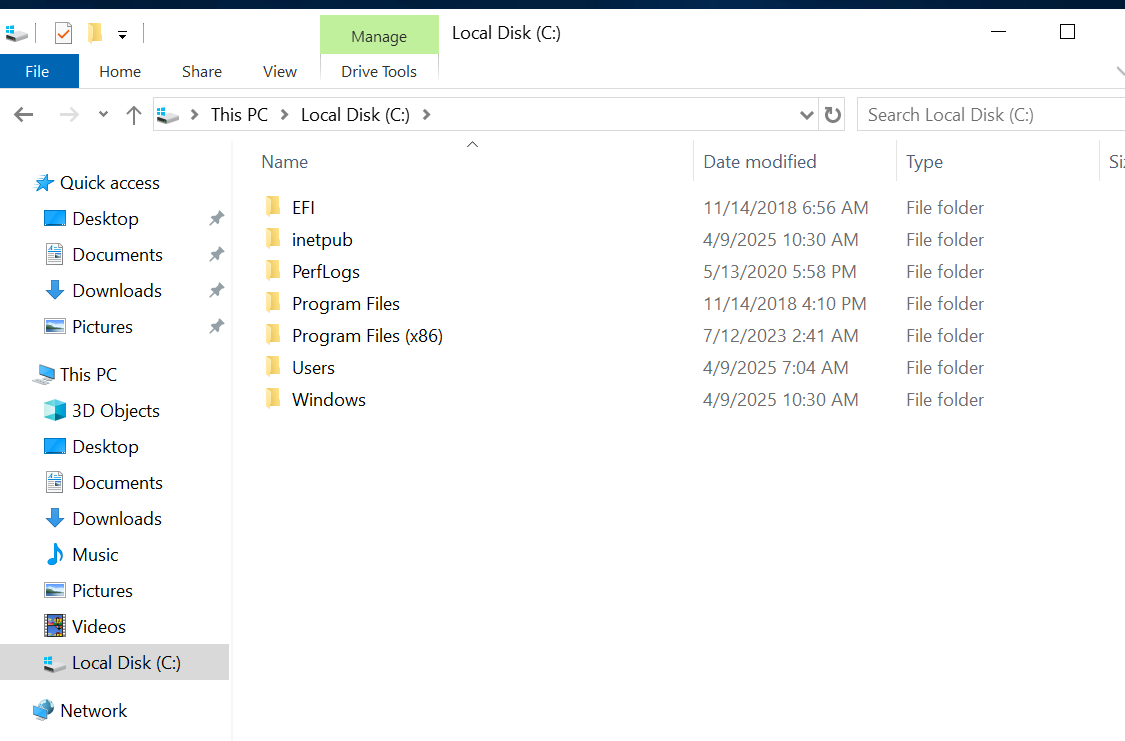
* Wait for the installation to complete and then restart the server if required.

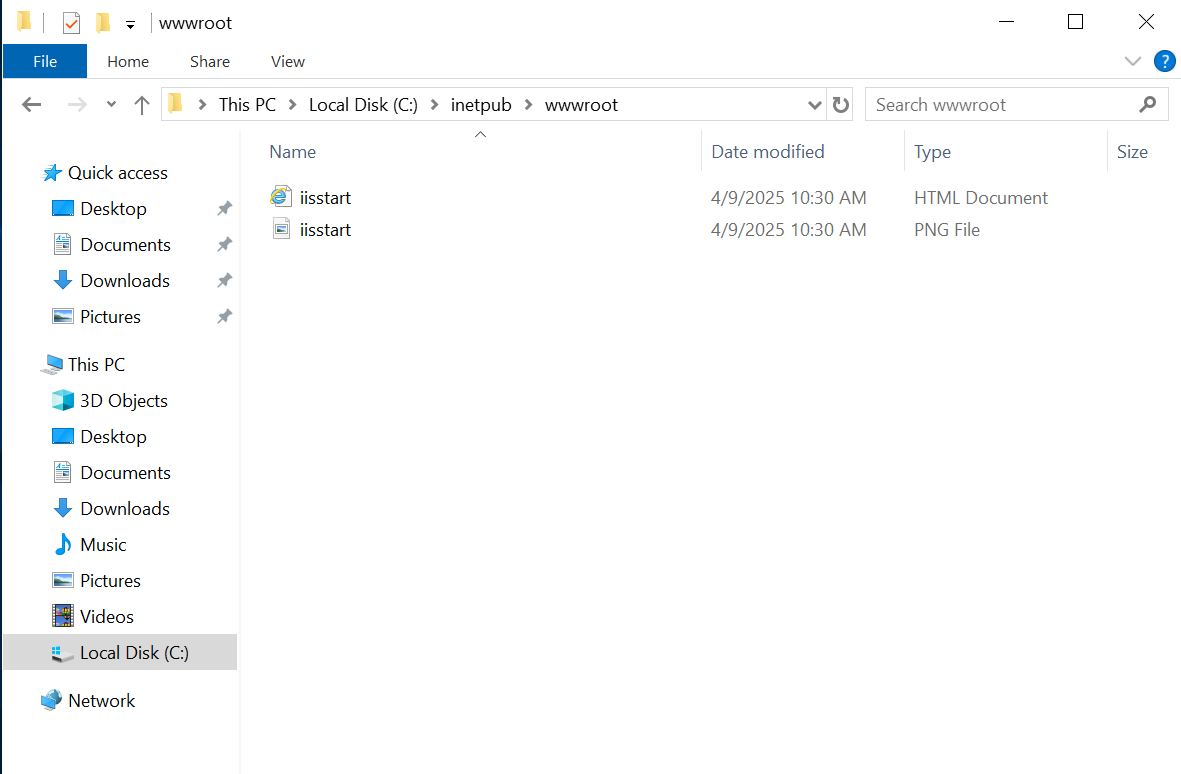


**Step 3: Create a Static Website and Check on Localhost**

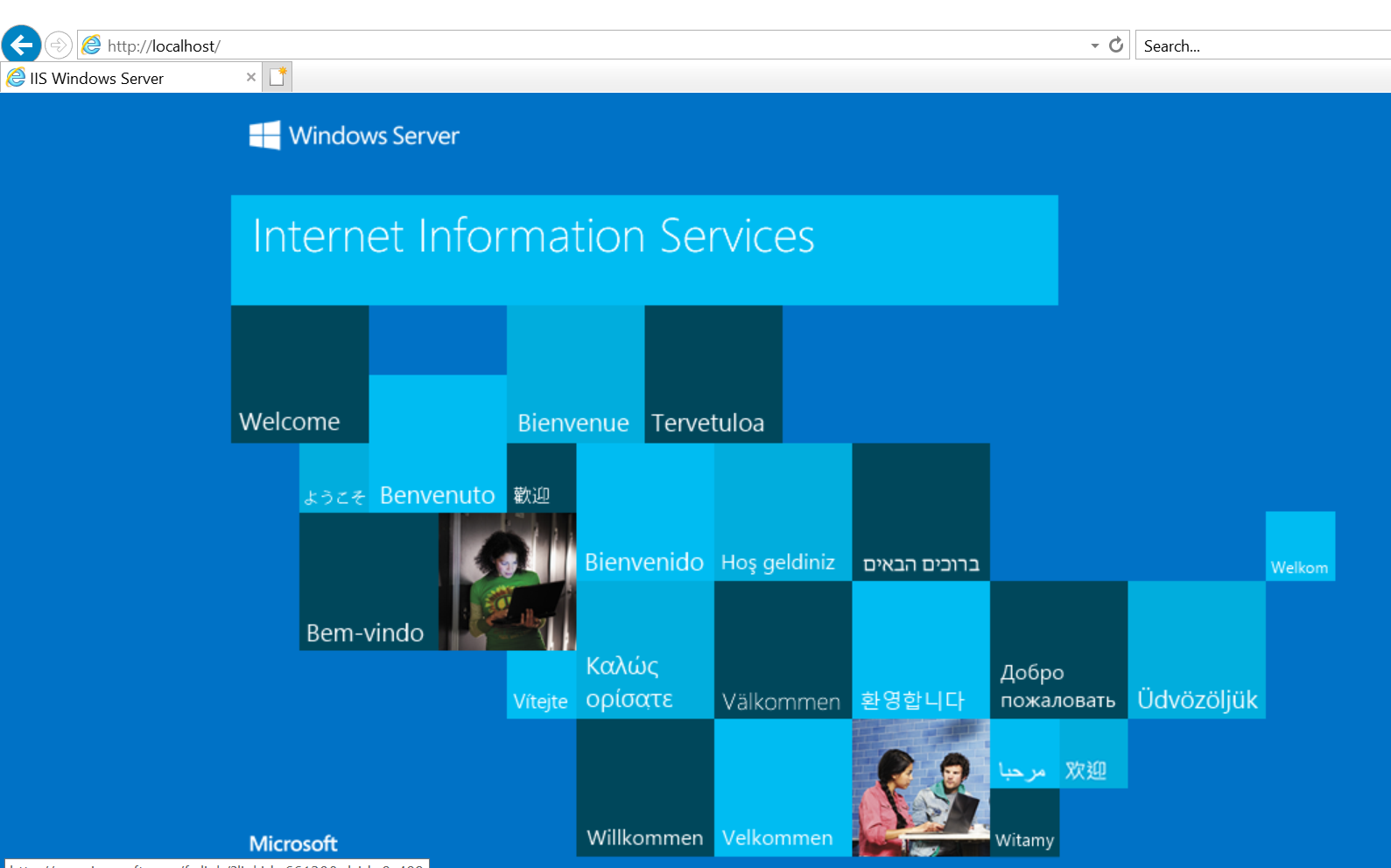
**3.1 Create a Static Website**

* Navigate to **C:\inetpub\wwwroot**.





* Open **Internet Explorer** and enter [**http://localhost**](http://localhost/).



* The created webpage should load successfully.
* Create an **index.html** file using Notepad or any text editor.

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />

<title>IIS Windows Server</title>

<style type="text/css">

<!--

body {

color:#000000;

background-color:#0072C6;

margin:0;

}

#container {

margin-left:auto;

margin-right:auto;

text-align:center;

}

a img {

border:none;

}

-->

</style>

</head>

<body> Hello, this is a test website running on IIS in AWS

<div id="container">

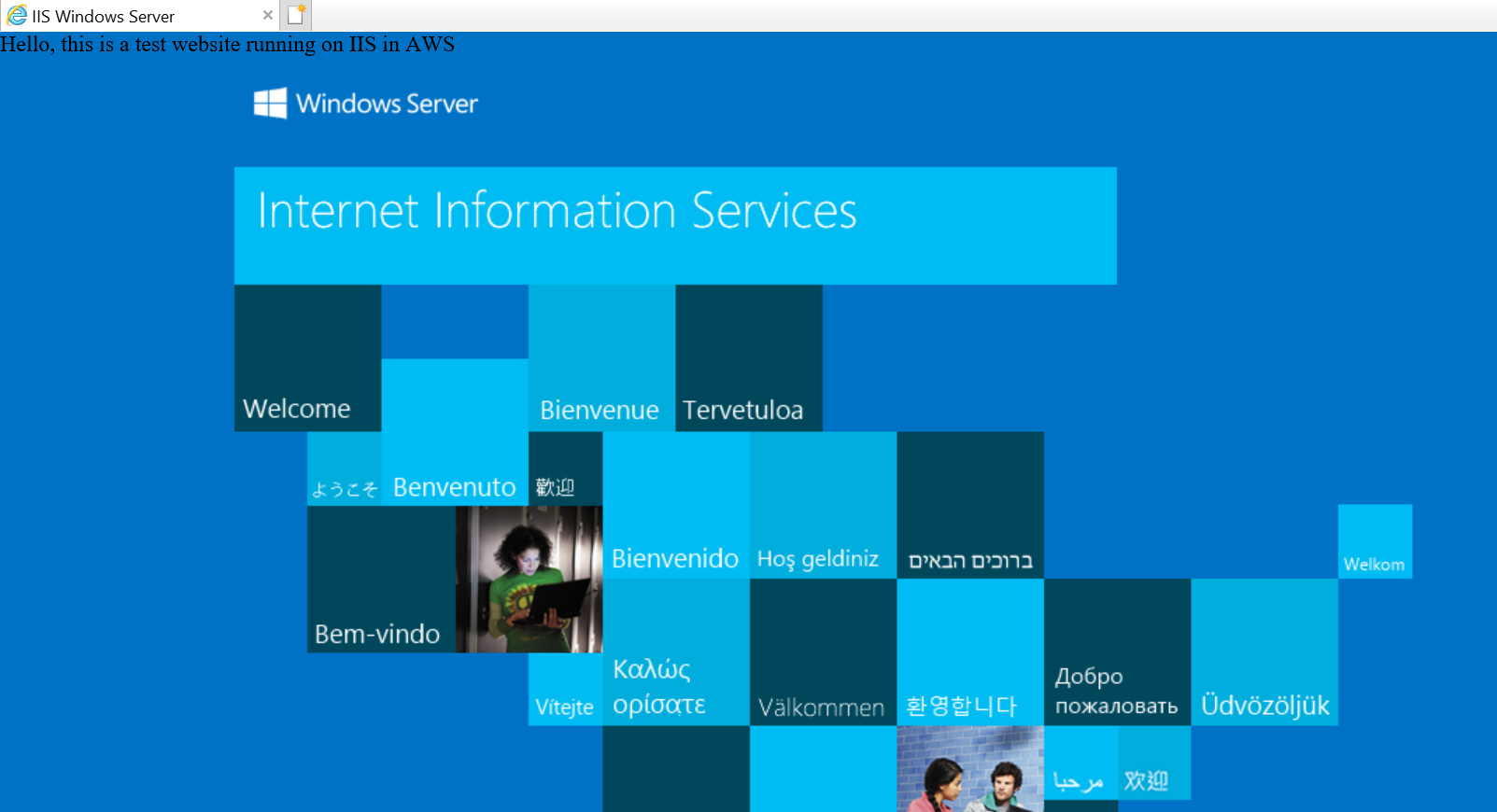
<a href="http://go.microsoft.com/fwlink/?linkid=66138&amp;clcid=0x409"><img src="iisstart.png" alt="IIS" width="960" height="600" /></a>

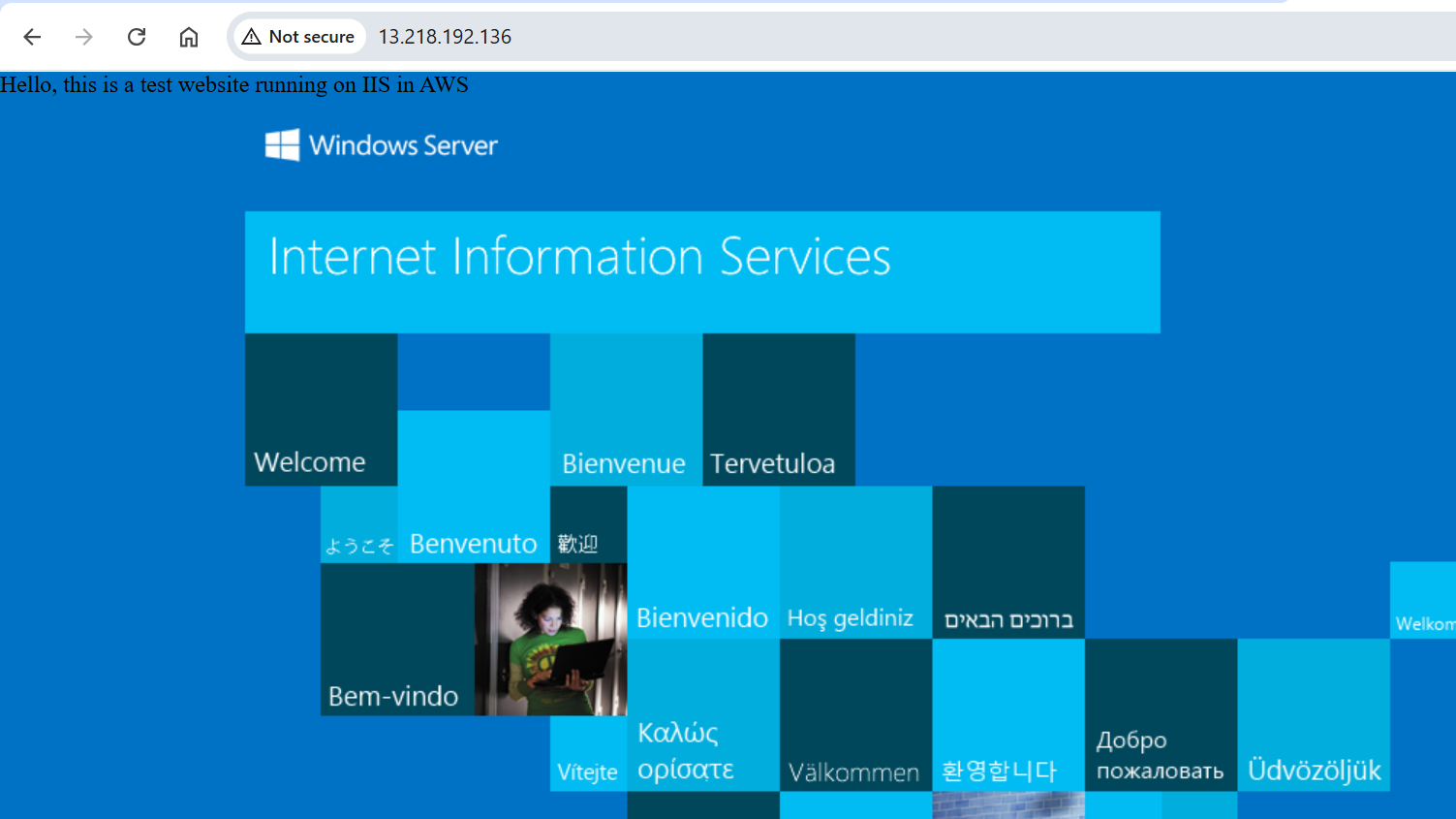
</div>

</body>

</html>

* After adding the index.html file verify the same by going on the local host and verify the same by accessing the public IP address of the instance:



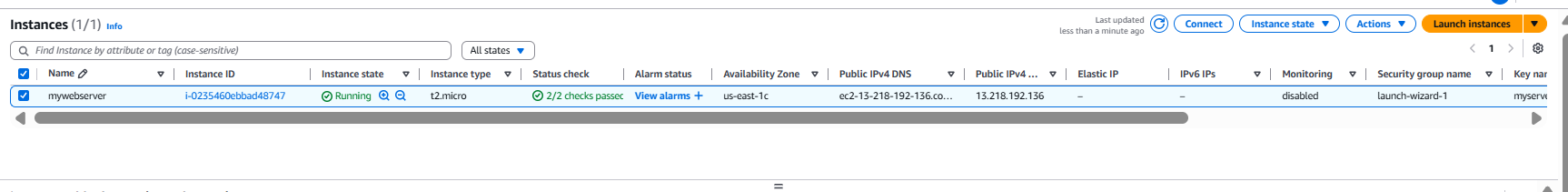
****

**Hence the web server successfully installed**

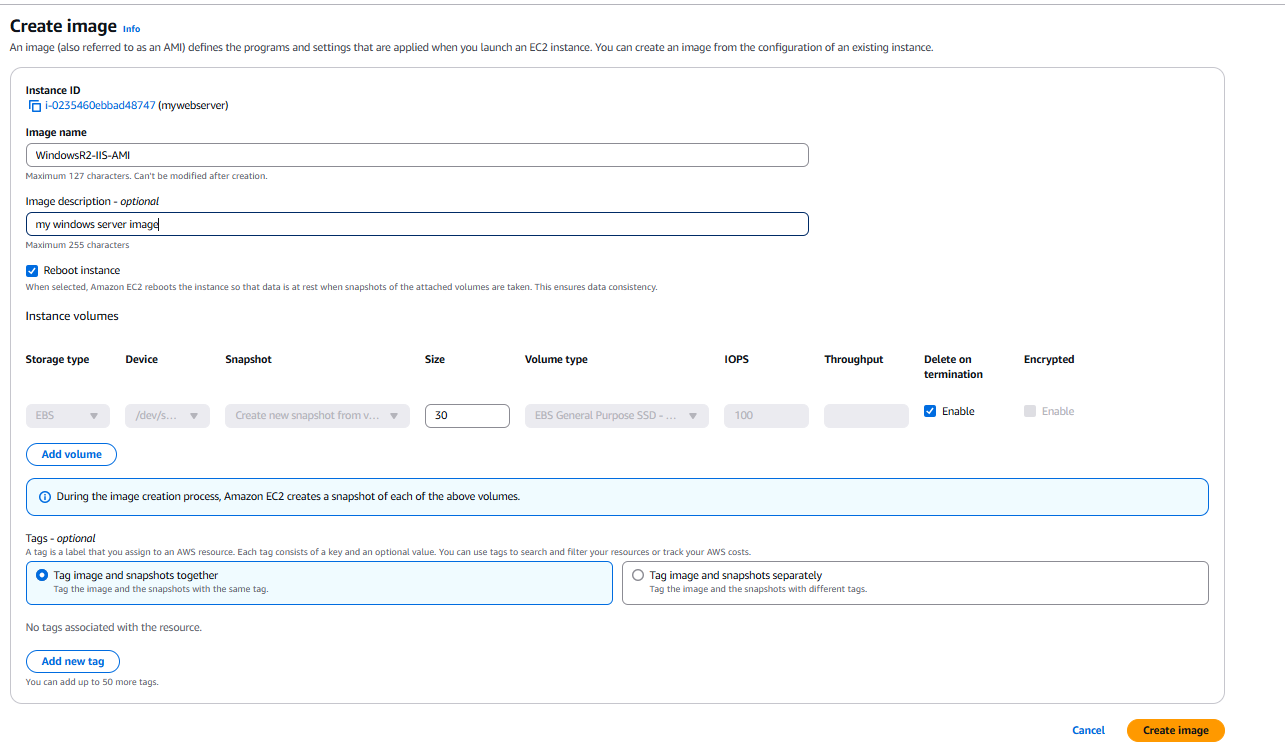
**Step 4: Create an Image of the Machine and Save the AMI**

**4.1 Create an AMI from the Configured Instance**

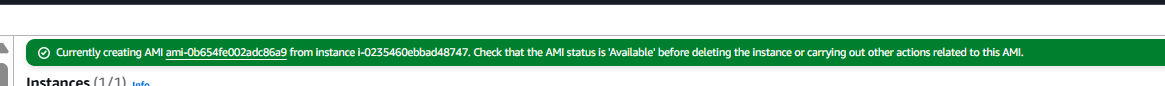
* Go to **AWS EC2 Dashboard**.
* Select the configured instance.

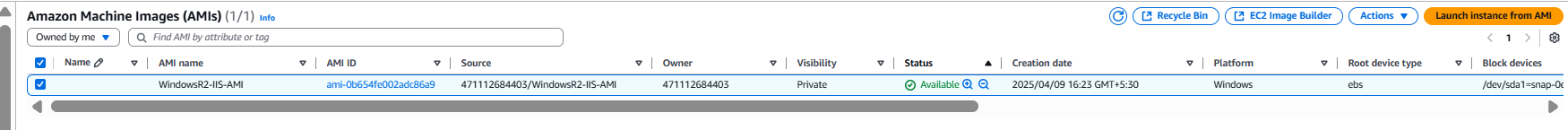


* Click on **Actions** → **Image and Templates** → **Create Image**.



* Provide a name (e.g., "WindowsR2-IIS-AMI").
* Click **Create Image** and wait for the AMI to be available under **AMI section**.

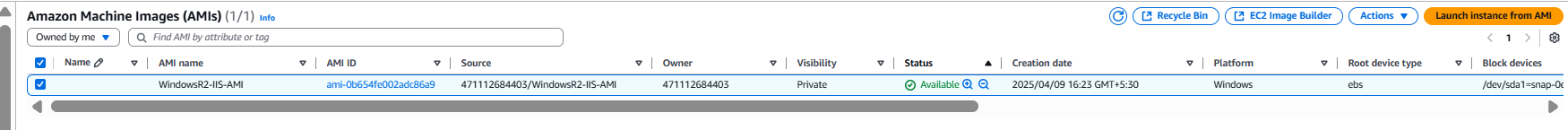


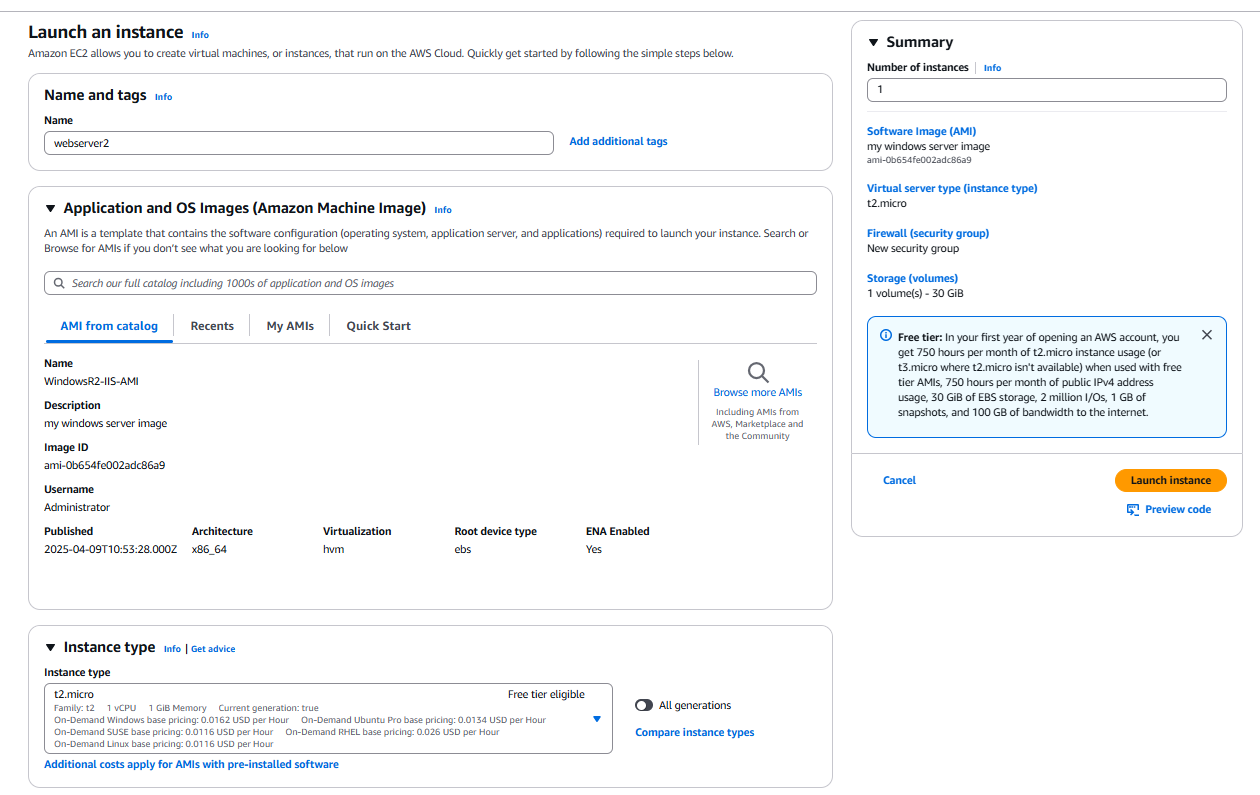


Hence, the AMI successfully launched with the windows server previously created.

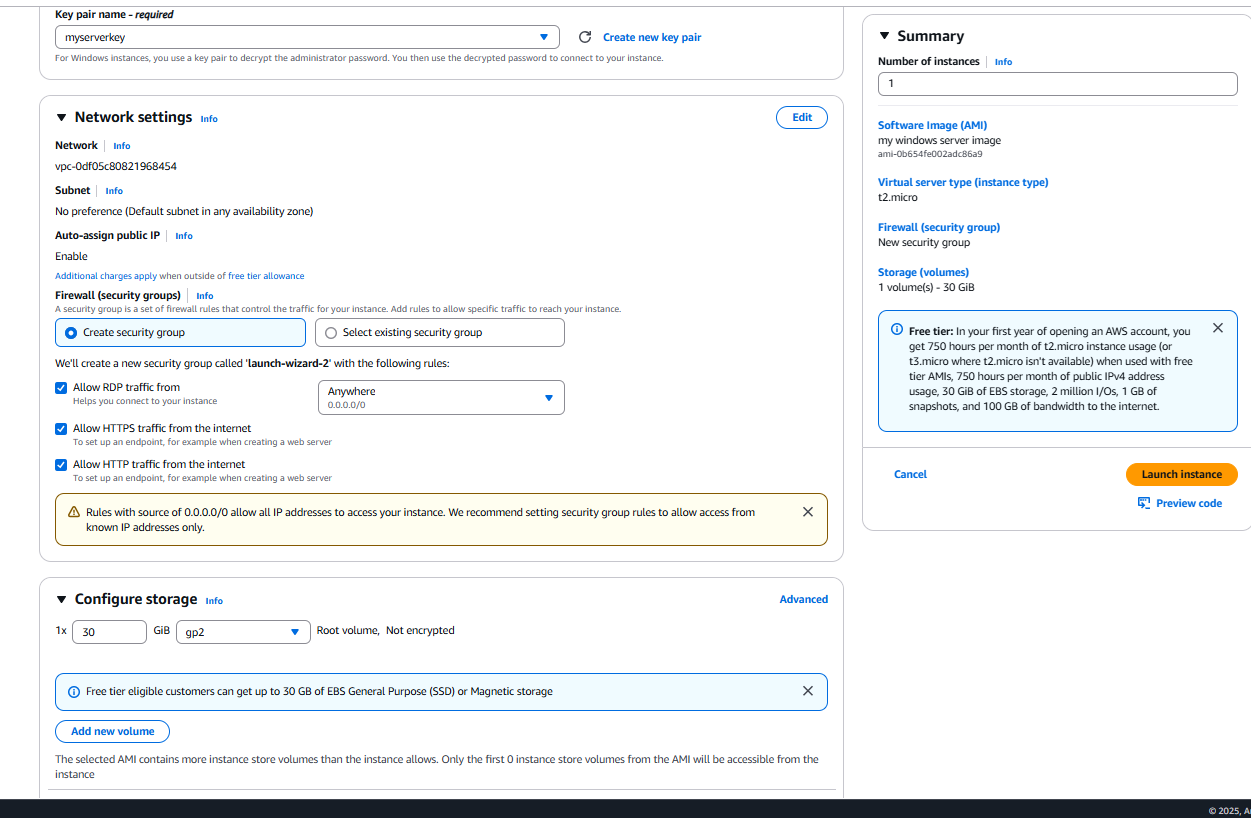
**Step 5: Launch a New Instance Using the Saved AMI**

**5.1 Create a New Instance**

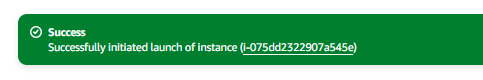
* Go to **EC2 Dashboard** → **AMIs**.
* Select the saved AMI and click **Launch Instance from Image**.
* 
* Configure the instance similar to Step 1.

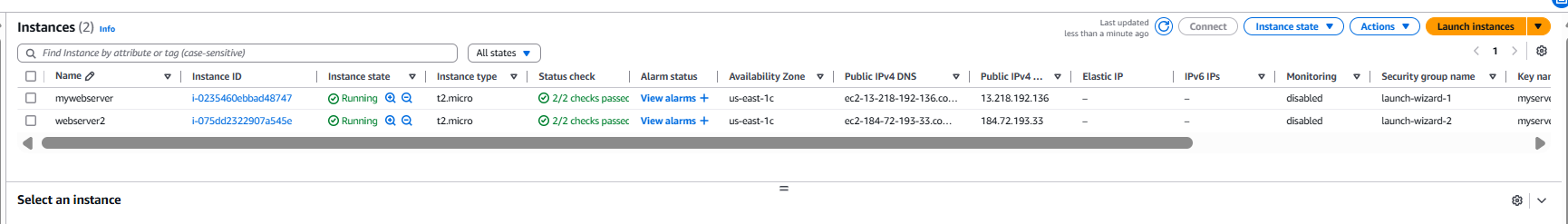


* Use the existing keypair and also assign the appropriate **security group** to allow HTTP access.



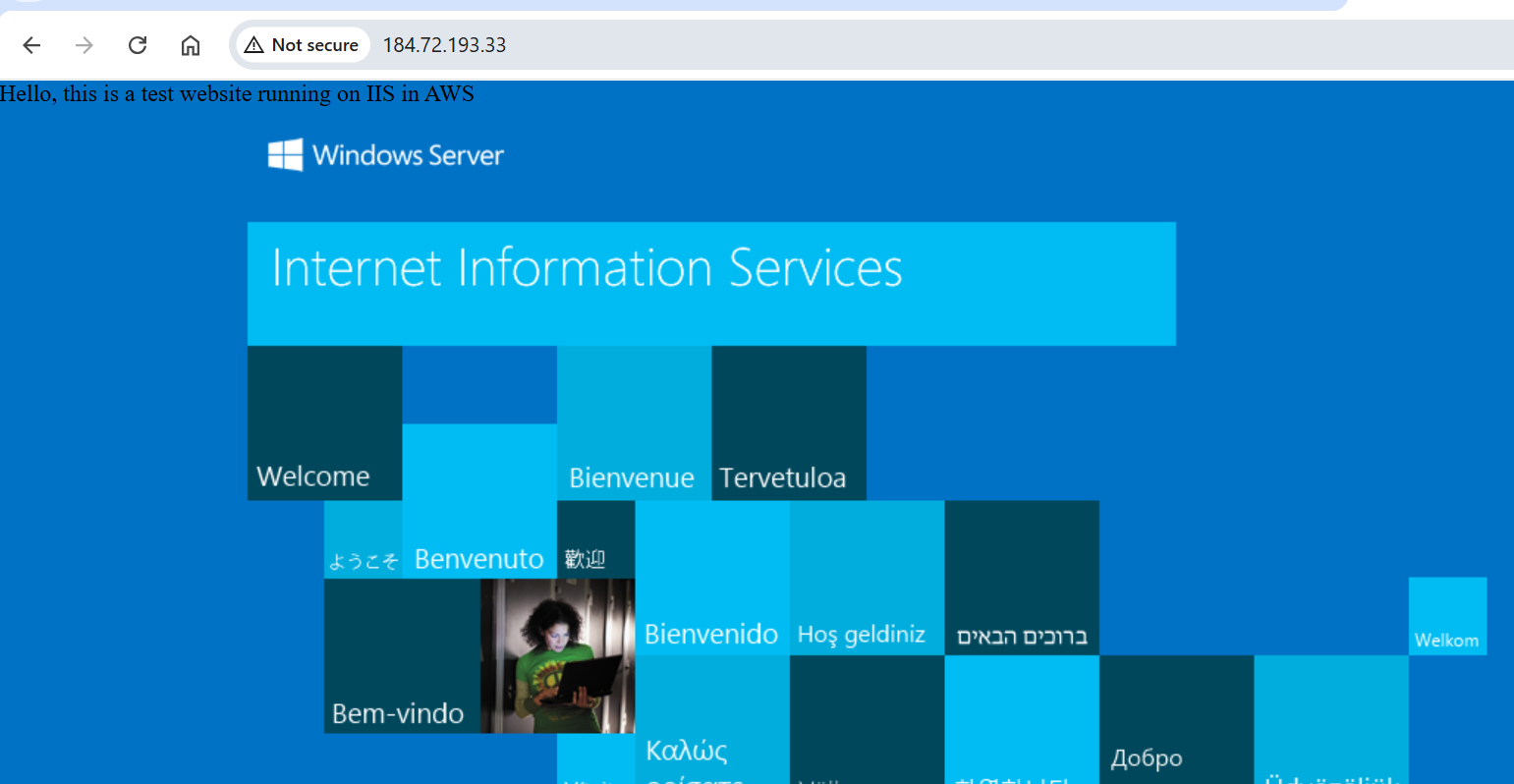
* Launch and connect to the instance.



Hence, the instance is successfully created by using the existing instance  


**5.2 Verify the Web Server**

* Open **Internet Explorer** on the new instance.

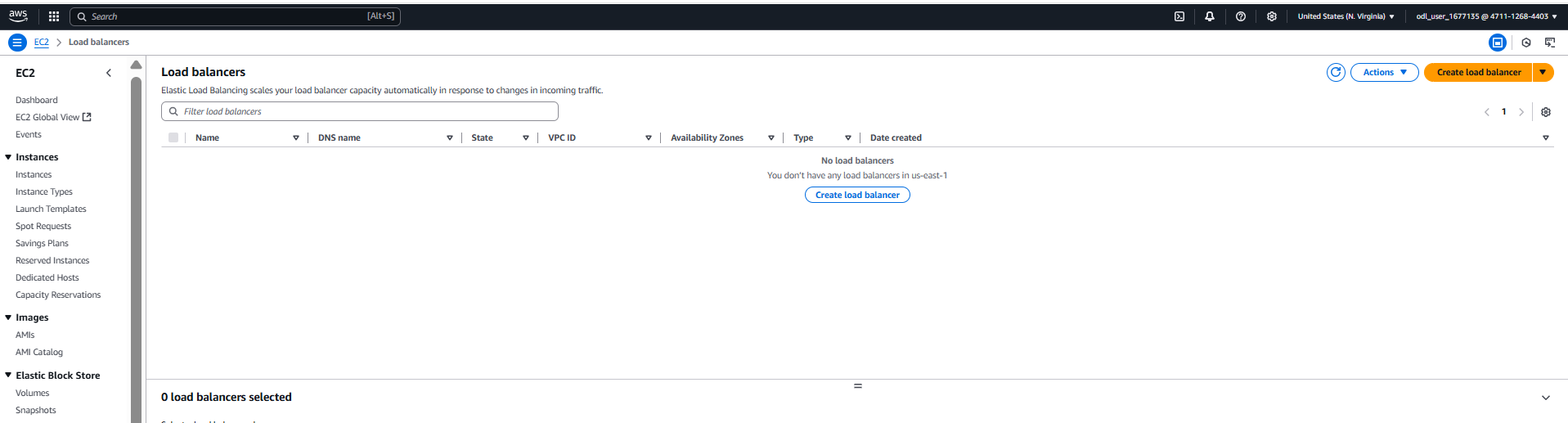


* The static website loads correctly as shown in the above screenshot.

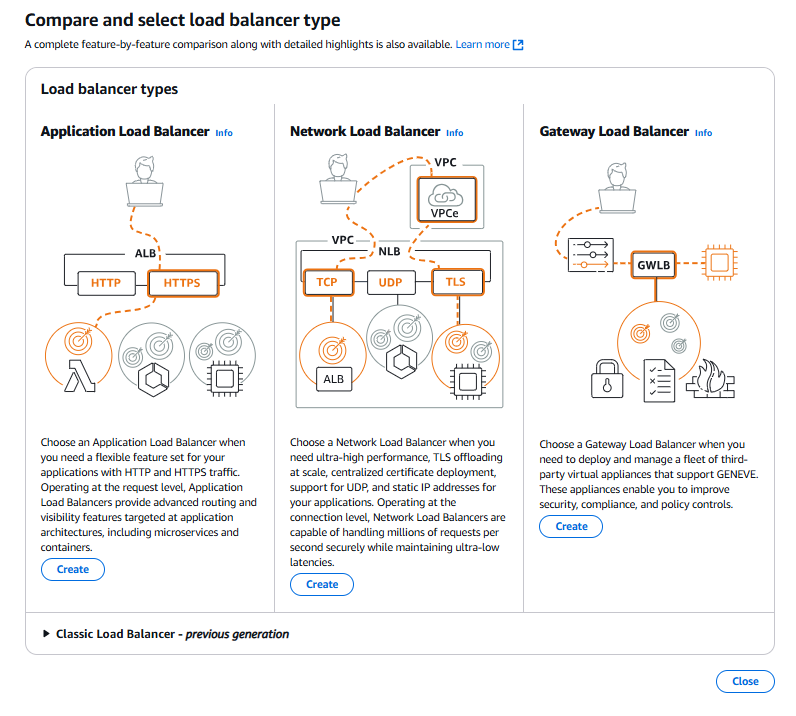
**Step 6: Create a Load Balancer and Attach the Instances**

**6.1 Navigate to Load Balancer**

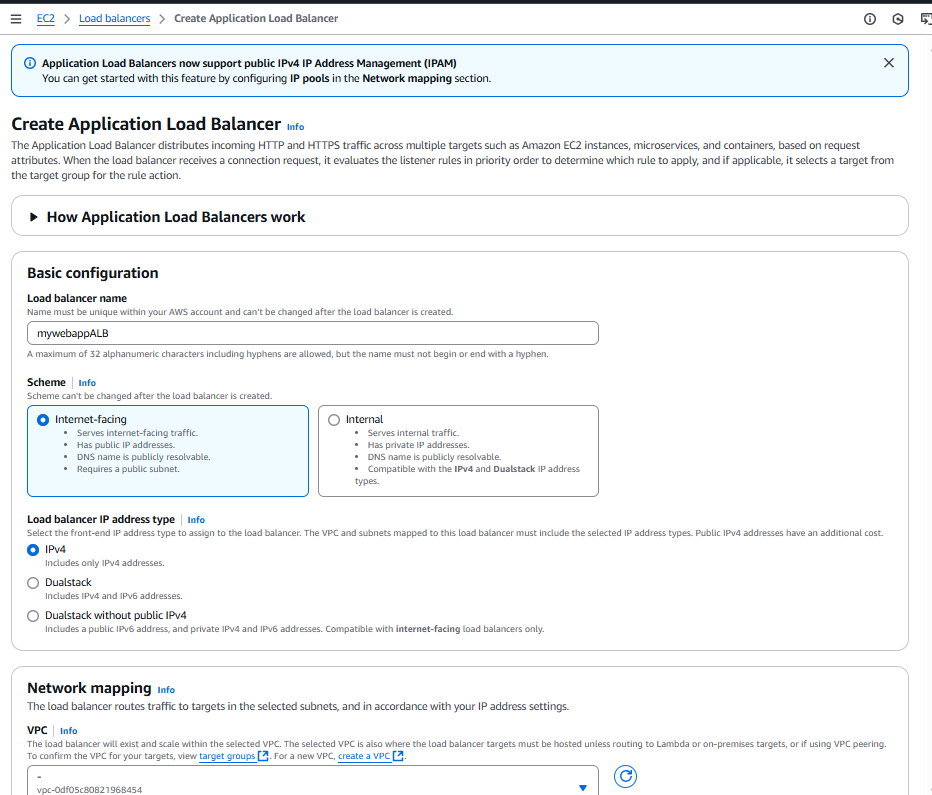
* Open the **EC2 Dashboard** → Click **Load Balancers** → **Create Load Balancer**.

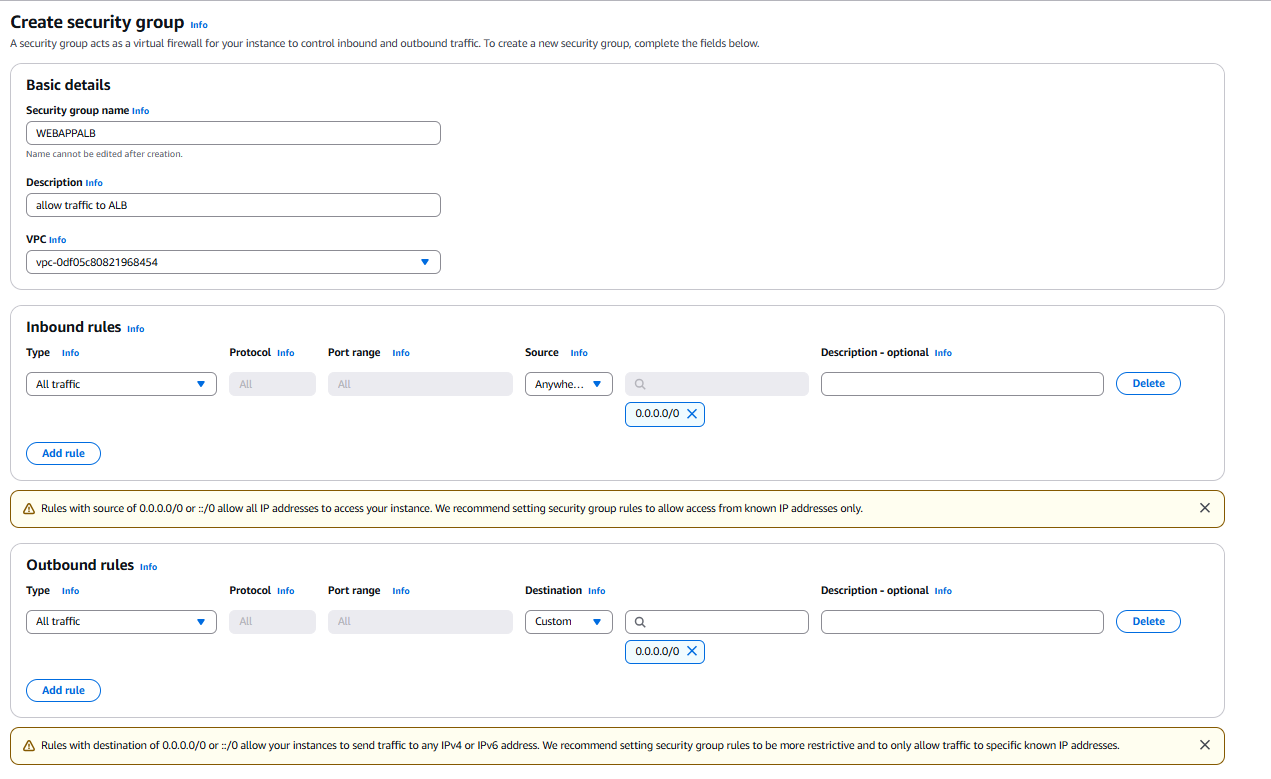


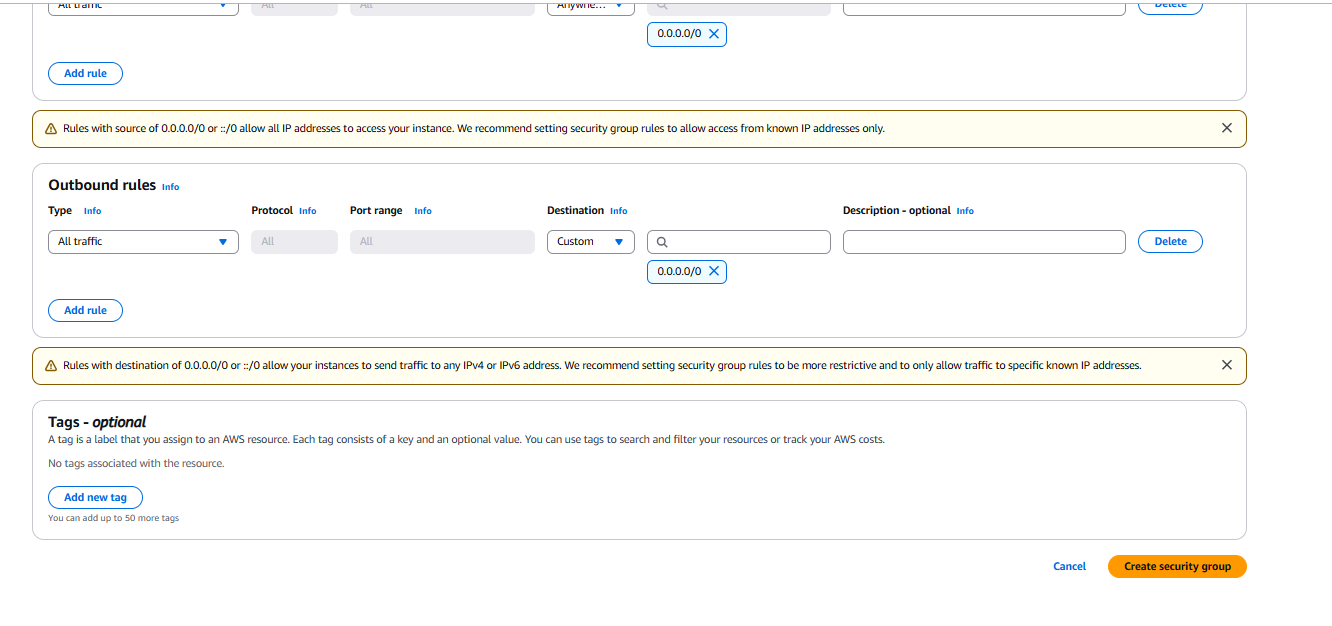
* Choose **Application Load Balancer**.

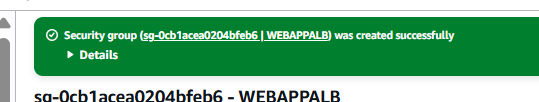


* Provide a **name** (e.g., "mywebappALB") and Set the **VPC** (DEFAULT)and select **subnets** across different Availability Zones.

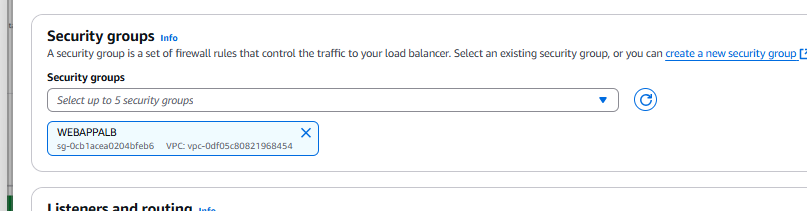


* Also create security group as shown:  
  





Also attach the security group created with the ALB

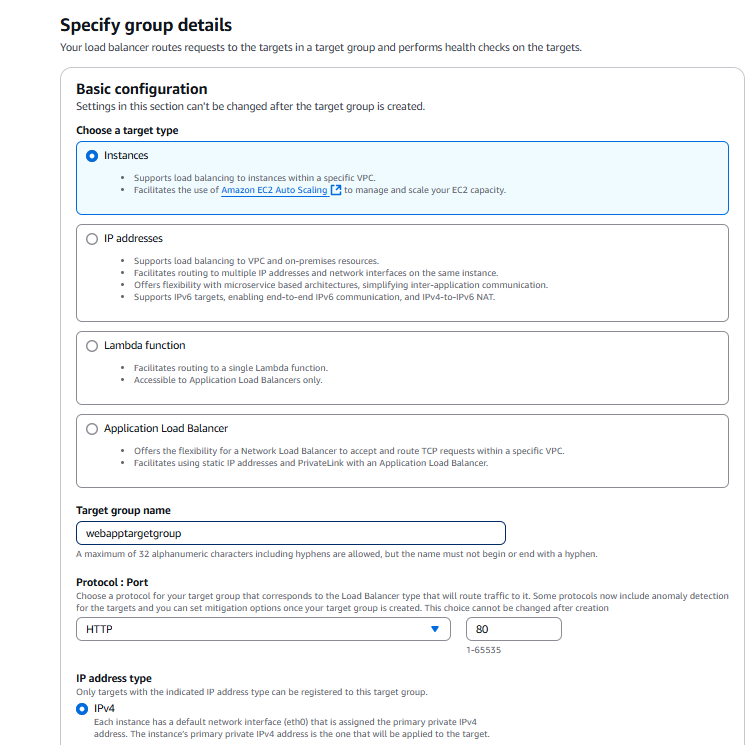


**6.2 Configure Target Group**

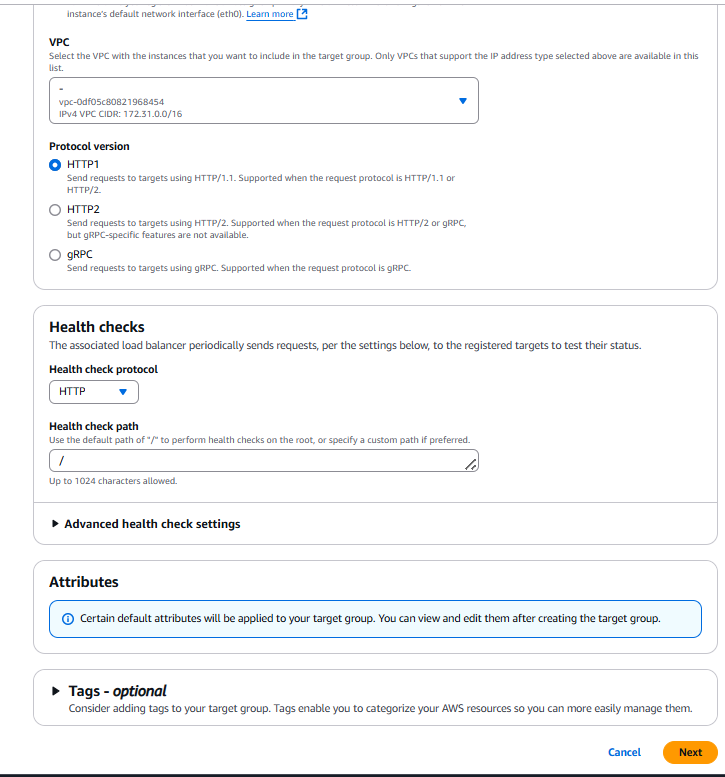
* Create a new **Target Group** with:



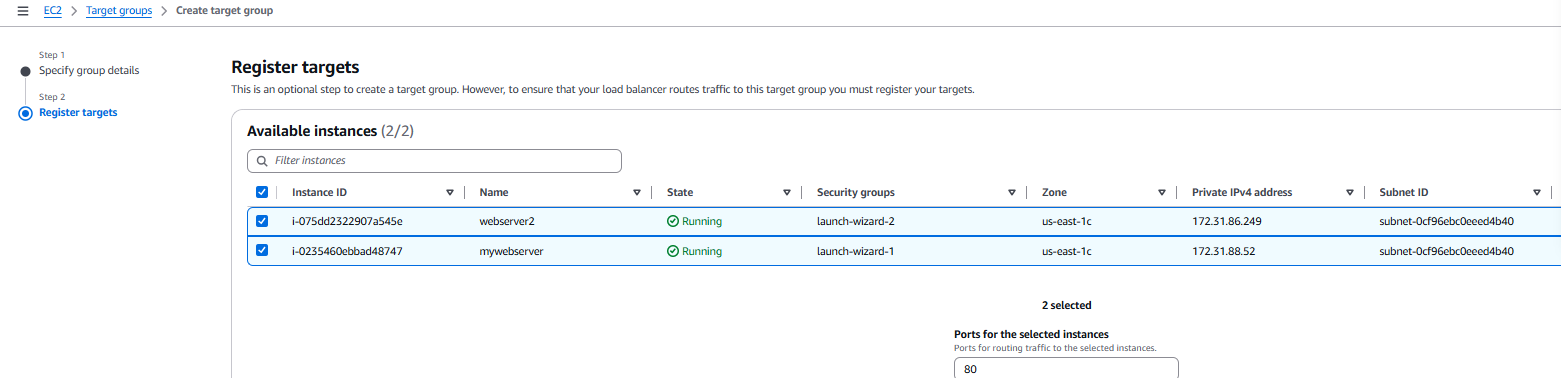
* + **Target type**: Instance
  + **Protocol**: HTTP (Port 80)



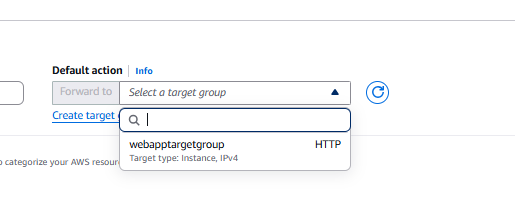
* + **Health Check Path**: /index.html

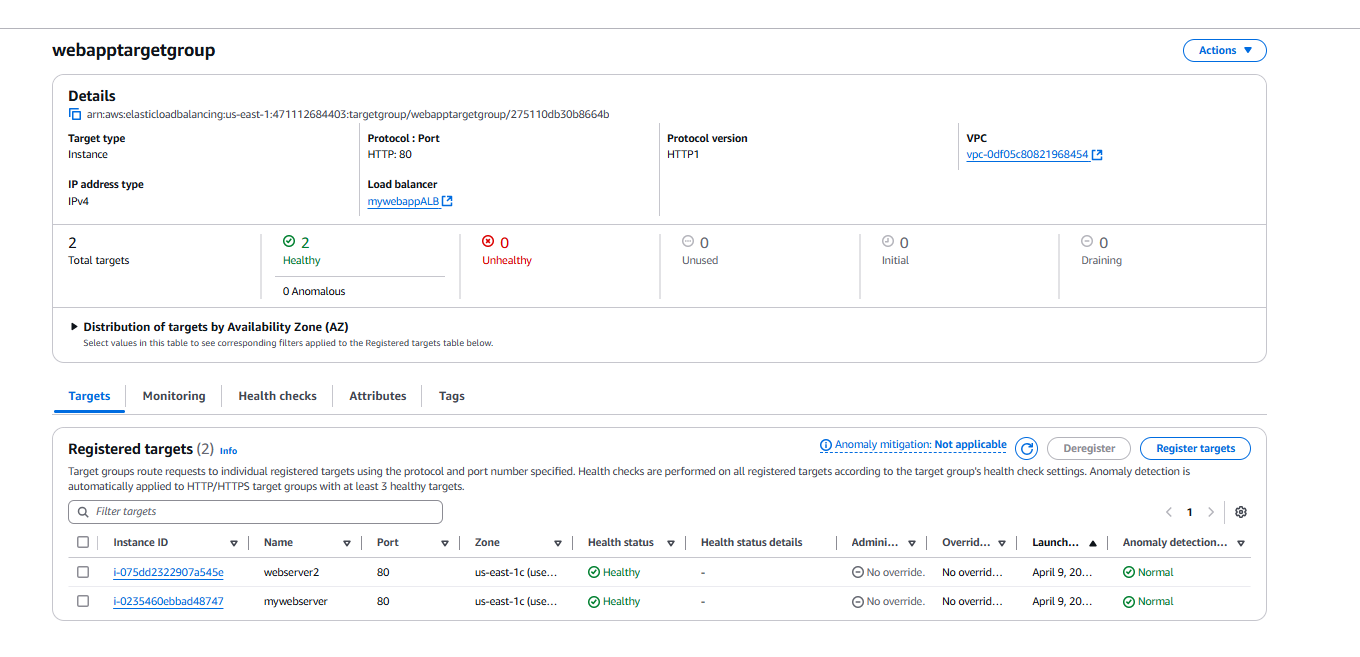


* Register both instances as targets.



* Click **Create** and attach the target group to the load balancer.

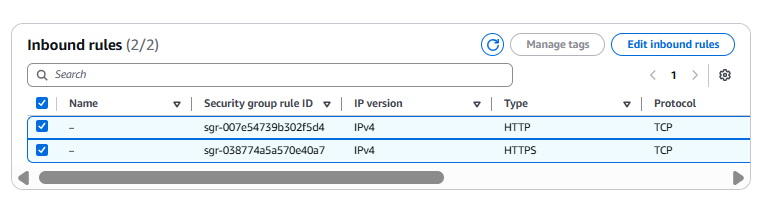




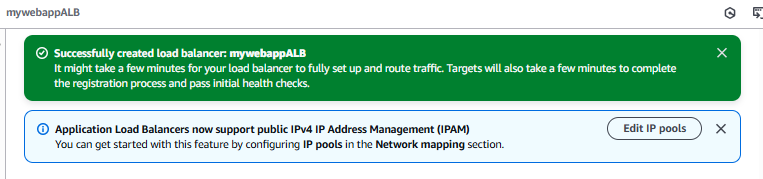
Target group successfully attached to both the existing instances.

**6.3 Review and Launch Load Balancer**

* Assign a **security group** allowing HTTP/HTTPS.



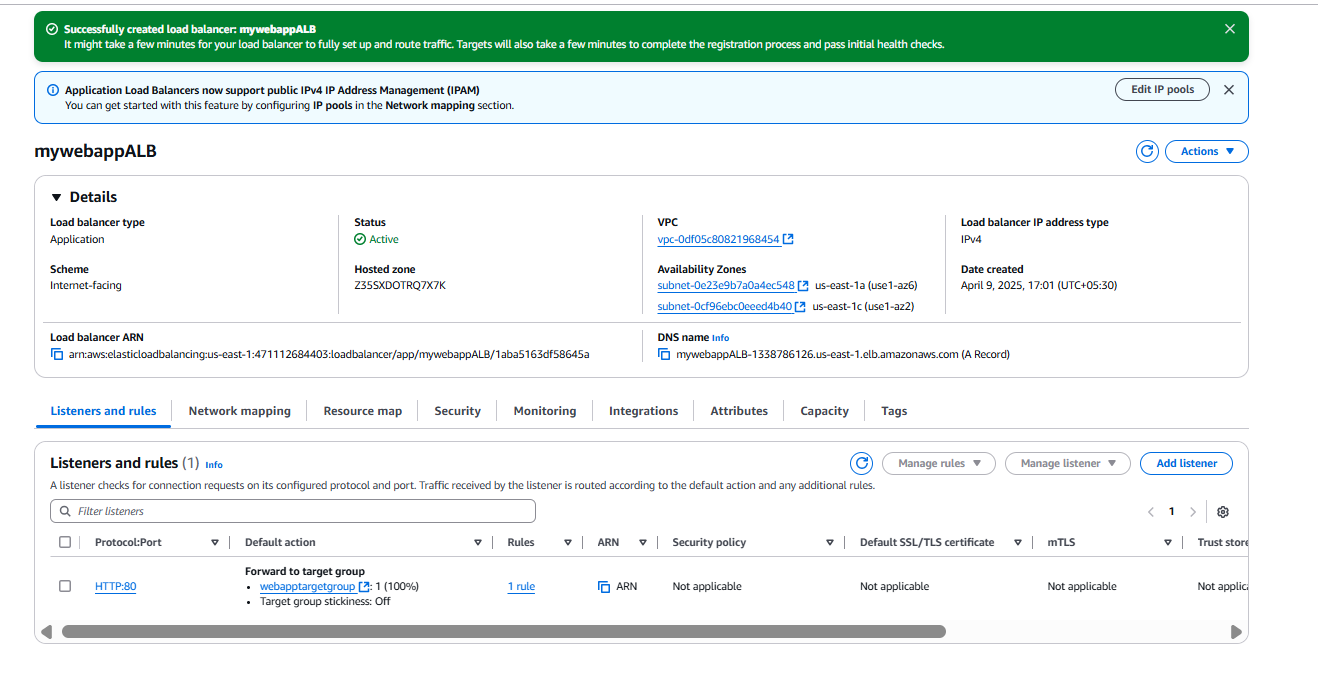
* Click **Create Load Balancer**.



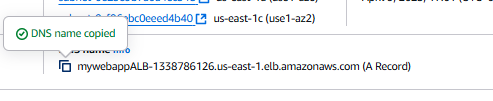
**Step 7: Check the DNS Name on the Browser**

**7.1 Get Load Balancer DNS Name**

* Navigate to **Load Balancers** in AWS Console.

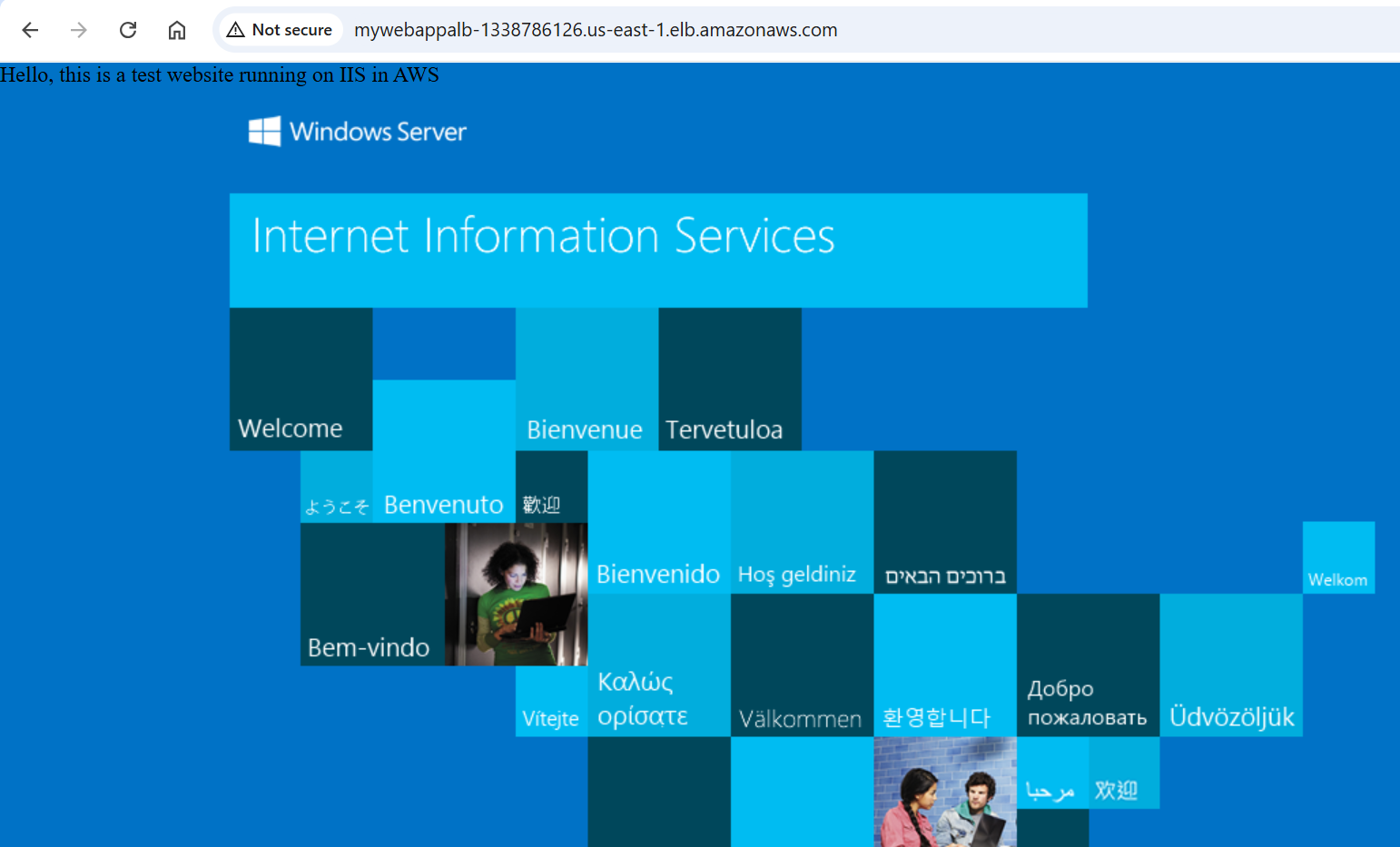


* Copy the **DNS name** of the Load Balancer.



**7.2 Open Website on Browser**

* Open a web browser and enter the DNS copied:



* The static website should be accessible from the browser.

**Conclusion**

By following these steps, we successfully deployed a web server on AWS Cloud. The process included creating an EC2 instance, installing IIS, setting up a static website, creating an AMI, deploying multiple instances, and configuring a load balancer. The website is now accessible globally through the Load Balancer's DNS. This setup ensures scalability, availability, and reliability for the client’s website.