Develop and Deploy C# Application in Windows Virtual Machine (LAB-M01-02)

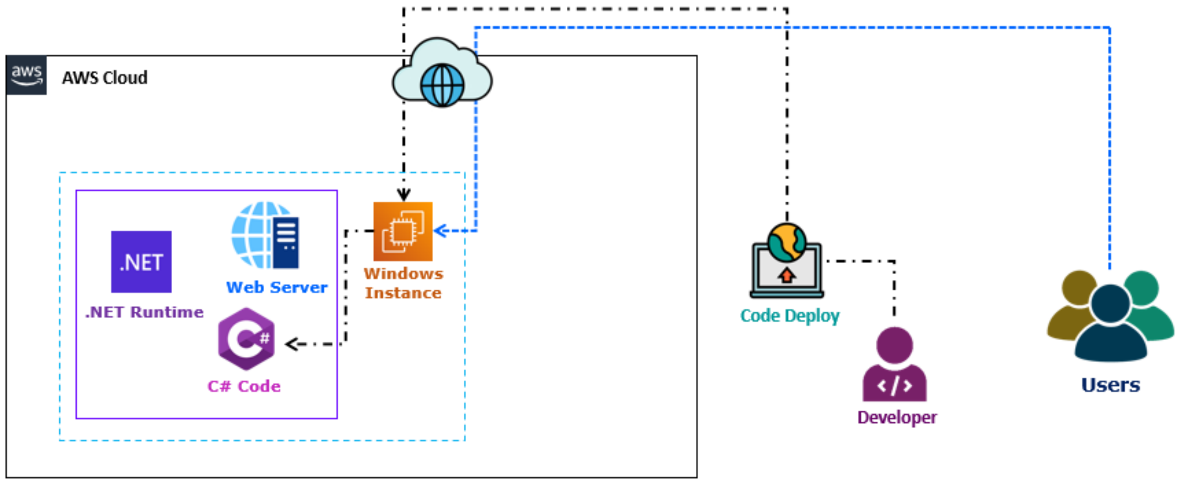
Lab scenario

You're preparing to deploy web application in AWS. As a development group, your team has decided to use C# (dot net based) application to deploy in the Windows environment in AWS.

Objectives

After you complete this lab, you will be able to:

- Create new virtual machine.
- Build the Run-time environment.
- Deploy the C# code.
- Access your web application.



Task 1: Develop C# Application

In this task, you will develop the C# (dot net based) code for deployment.

Step 1: Develop the Code to Display the Server IP Address

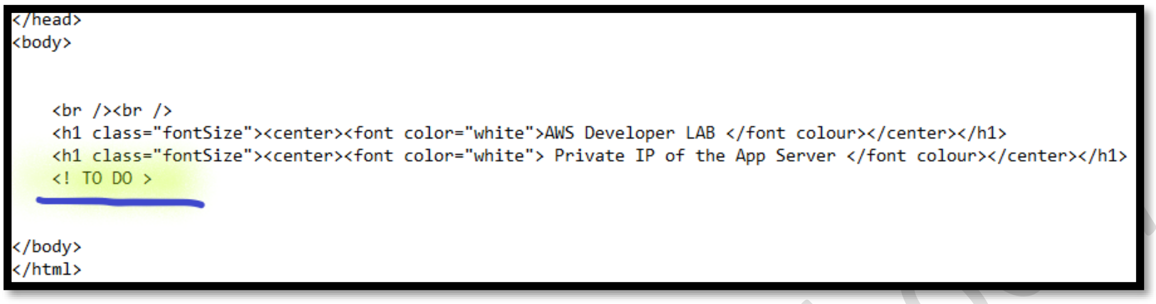
1. **Unzip** the **LAB-01-02.zip** (**C# code**).

Note: **Lab-01-02.zip** code file is available with the Lab Manual.

2. Open the **views** folder.

3. Open the **home** folder.
4. Open the **index.cshtml** in the **notepad**.
5. **Update** the **code** to **display** the *server Private address*.

    i. Look for the **TO DO** section.



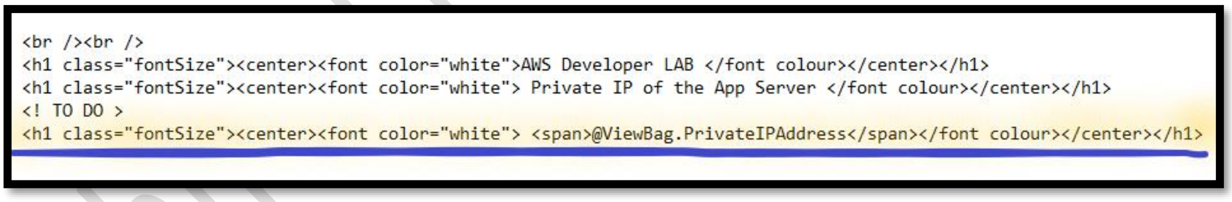
```
</head>
<body>


    <br /><br />
    <h1 class="fontSize"><center><font color="white">AWS Developer LAB </font colour></center></h1>
    <h1 class="fontSize"><center><font color="white"> Private IP of the App Server </font colour></center></h1>
    <! TO DO >


</body>
</html>
```

    ii. **Add** the **Code** after **TO DO** to display the server Private IP address.

Info:

1. You can also use the below code to display the Server Private IP Address.
   <h1 class="fontSize"><center><font color="white">
   <span>@ViewBag.PrivateIPAddress</span></font colour></center></h1>

2. Add the above code below to **<! TODO >** in the index.cshtml.



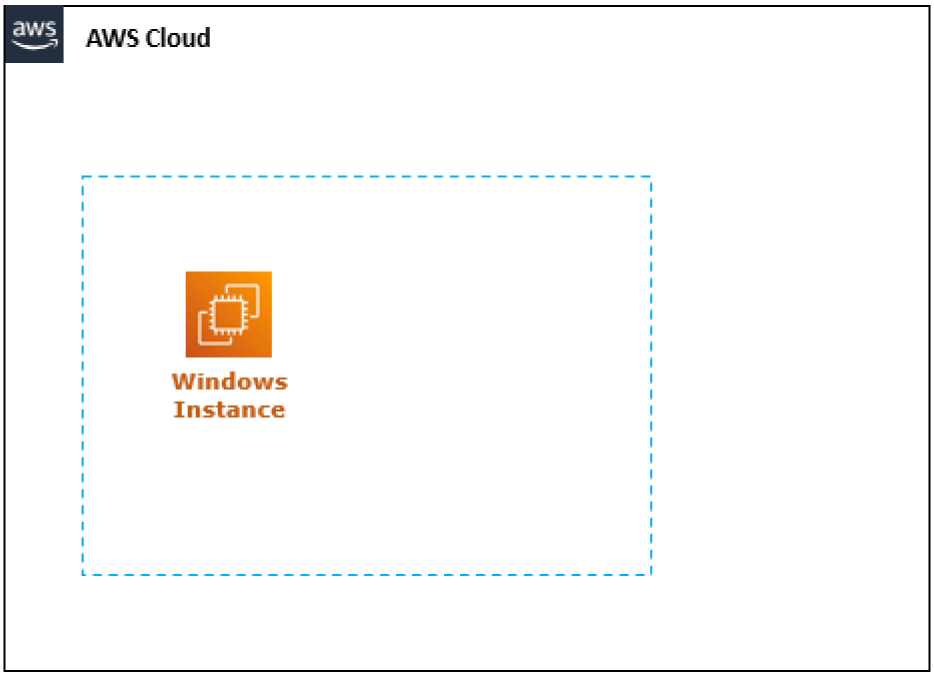
```
<br /><br />
<h1 class="fontSize"><center><font color="white">AWS Developer LAB </font colour></center></h1>
<h1 class="fontSize"><center><font color="white"> Private IP of the App Server </font colour></center></h1>
<! TO DO >
<h1 class="fontSize"><center><font color="white"> <span>@ViewBag.PrivateIPAddress</span></font colour></center></h1>
```

    iii. Select the **File**.

    iv. Select **Save**.

Task 2: Create Server to C# Application

In this task, you will launch an Amazon EC2 instance using the management console. The instance will be used to deploy the C# Application.



Step 1: Create EC2 Instance

6. Choose the **US East (N. Virginia)** region list to the right of your account information on the navigation bar.

7. In the **AWS Management Console**, on the **Services** menu **Search** and **Select EC2**.

8. Click **Launch Instance**.

   i. Click **Launch Instance**.

9. **Choose Amazon Machine Image (AMI)** section:

   i. **Go below** and search for **Microsoft Windows Server 2019 Base**.

   ii. Click on **Select**.

10. **Choose Instance Type** section:

    i. Choose an **t2.micro**.

    ii. Click **Next: Configure Instance Details**.

11. **Configure Instance Details** section:

    i. Select **Next: Add Storage**.

Note: Leave the detail as default.

12. **Add Storage** section:

    i. Select **Next: Add Tags**.

Note: Leave the detail as default.

13. **Add Tags** section:

    i. Select **Add Tag**.

    - **Key Name**: Write **Name**.
    - **Value**: Write **C# App Server**.

    ii. Choose **Next: Configure security group**.

14. **Configure Security Group** section:

Note: The wizard automatically defines the launch-wizard-*x* security group and creates an inbound rule to allow you to connect to your instance over RDP (**port 3389**).

    i. **Assign a security group**: Select **Create a new security group**.

    a. **Security group name**: Write **C# App Server**.
    b. **Description**: Write **C# App Server Security Group**.
    c. Select **Add Rule**.

Note: Don't remove or update the RDP rule. Add the new rule.

    - **Type**: Dropdown and Select **HTTP**.
    - **Source**: Dropdown and Select **Anywhere**.

    ii. Click **Review and Launch**.

15. Click **Launch**

Note: New window will pop-up for Key pair.

16. **Select an existing key pair or create a new key pair** section:

    i. In the **popover**, dropdown and select **Choose an existing key pair**.

        a. **Select a Key pair**: Dropdown and Select **My-Dev-LAB-KP**.

        b. **Select** I acknowledge …

17. Click **Launch Instances**.

Step 2: Check the C# App Server Status

18. In the **AWS Management Console**, on the **Services** menu, click **EC2**.
19. Click **Instance**.
20. Select **C# App Server**.

    i. **Wait** for the **Instance State** to change to **Running** state.

    ii. **Wait** for the **Status check** to change to **2/2 checks passed**.

Task 3: Connect to C# App Server

In this task, you will log into the Dot Net Application (Windows) Server that you just created.

Step 1: Copy Dot Net Application Server Public IP address

21. In the **AWS Management Console**, on the **Services** menu, click **EC2**.
22. Click **Instances**.
23. Select **C# App Server**.

    i. Go below and click on **Details**.

    ii. **Copy** the **Public IP address**.

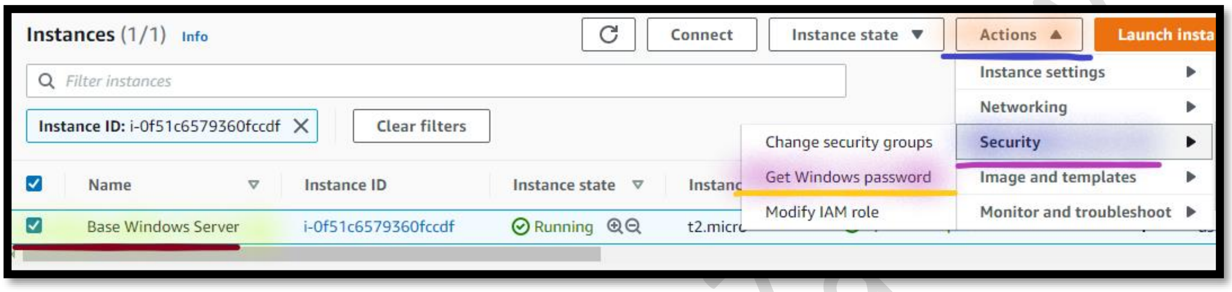
Step 2: Generate Windows Password

24. To generate windows password, select **C# App Server** (windows) virtual machine.

    i. Select **Actions**.

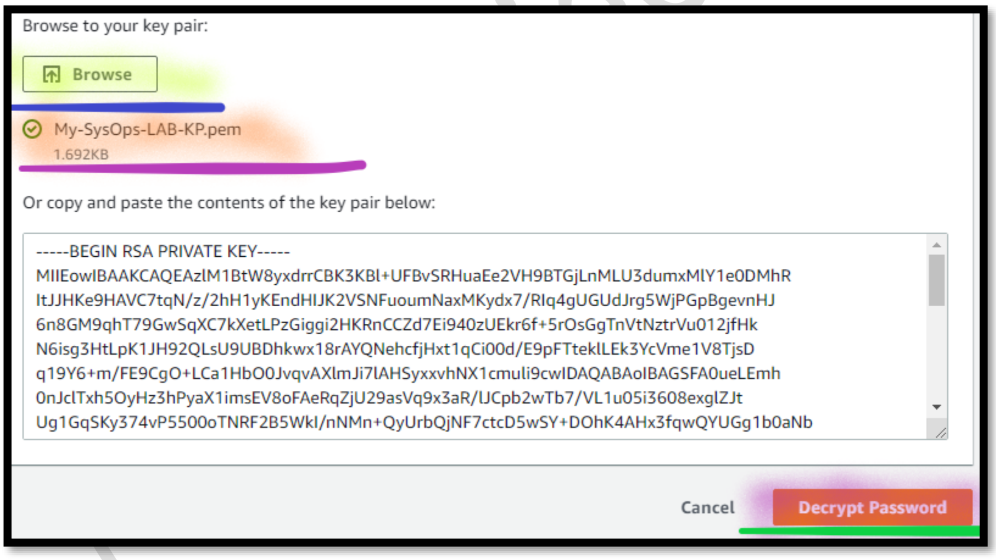
    ii. Select **Security**.

    iii. Select **Get Windows Password**.



    iv. **Browse**: Navigate and Select **My-Dev-LAB-KP**.pem key pair.

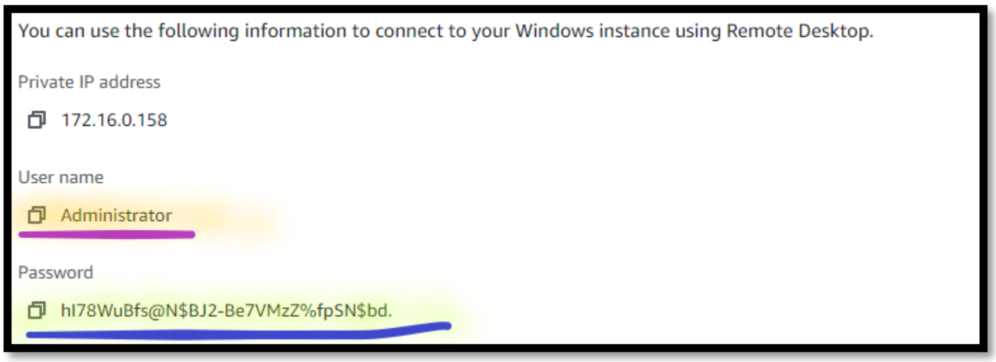
    v. Click on **Decrypt Password**.


Browse to your key pair:

Or copy and paste the contents of the key pair below:

```
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAzlM1BtW8yxdrrCBK3KBl+UFBvSRHuaEe2VH9BTGjLnMLU3dumxMlY1e0DMhR
ItJJHKe9HAVC7tqN/z/2hH1yKEndHIJK2VSNFuoumNaxMKydx7/RIq4gUGUdJrg5WjPGpBgevnHJ
6n8GM9qhT79GwSqXC7kXetLPzGiggi2HKRnCCZd7Ei940zUEkr6f+5rOsGgTnVtNztrVu012jfHk
N6isg3HtLpK1JH92QLsU9UBDhkwx18rAYQNehcfjHxt1qCi00d/E9pFTteklLEk3YcVme1V8TjsD
q19Y6+m/FE9CgO+LCa1HbO0JvqvAXlmJi7lAHSyxxvhNX1cmuli9cwIDAQABAoIBAGSFA0ueLEmh
0nJclTxh5OyHz3hPyaX1imsEV8oFAeRqZjU29asVq9x3aR/lJCpb2wTb7/VL1u05i3608exglZJt
Ug1GqSKy374vP5500oTNRF2B5WkI/nNMn+QyUrbQjNF7ctcD5wSY+DOhK4AHx3fqwQYUGg1b0aNb
```

Note: Windows will pop-up with **user name** and **password**.


You can use the following information to connect to your Windows instance using Remote Desktop.

Private IP address

172.16.0.158

User name

Administrator

Password

hI78WuBfs@N$BJ2-Be7VMzZ%fpSN$bd.

Note: Copy the **user name** and **password** in **Notepad**.

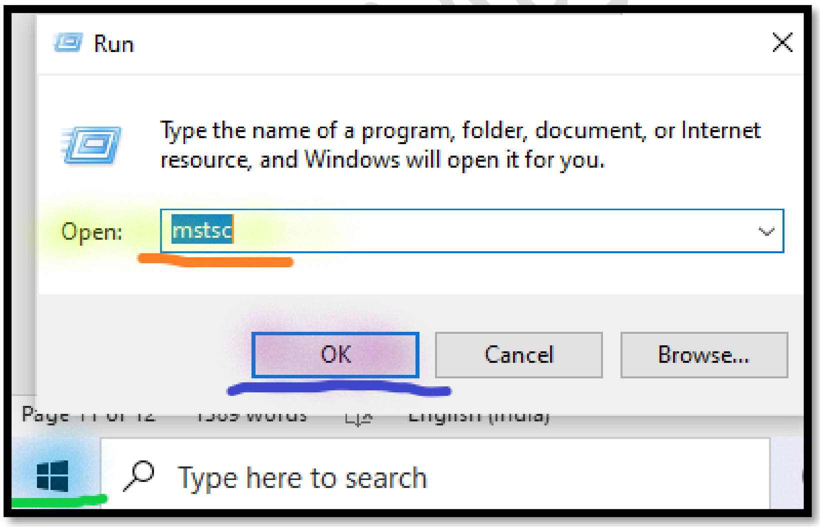
vi. Select **Close**.

Step 3: Remote Desktop from Windows Desktop/ Laptop

Note: If you are using **Mac** Operating System, go below to follow the **Step 4**.

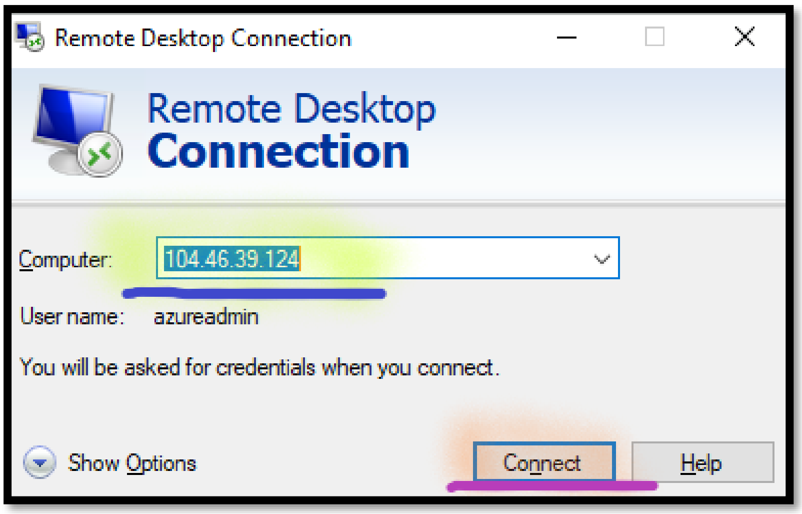
25. From the **local Desktop/ Laptop** (Windows), right click on **Start** & **Run**.

26. In the open, write **mstsc**, press **Ok**.


Run
Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.
Open: mstsc
OK
Cancel
Browse...
Type here to search

i. **Type** the **Public IP Address** of the **C# App Server** instance.

ii. Click **Connect**.


Remote Desktop Connection
Remote Desktop Connection
Computer:
104.46.39.124
User name:
azureadmin
You will be asked for credentials when you connect.
Show Options
Connect
Help

iii. **Type** the **Username** and **Password** of the **C# App Server** instance and click **Ok**.

iv. Click on **Yes** to confirm this connection, if prompted with the security message.

Note: Go to the **Task 4**, **But don't close the Windows console**.

Step 4: Remote Desktop from Mac Desktop/ Laptop

Note: If you are using **Windows** Operating System, go below to follow the **Step 3**.

27. **Download** & **Install** the Microsoft Remote Desktop client from the Mac App Store.

https://apps.apple.com/us/app/microsoft-remote-desktop-8/id715768417

28. Open the **Remote Desktop client**.

i. **Type** the **Username** and **Password** of the **C# App Server** instance.

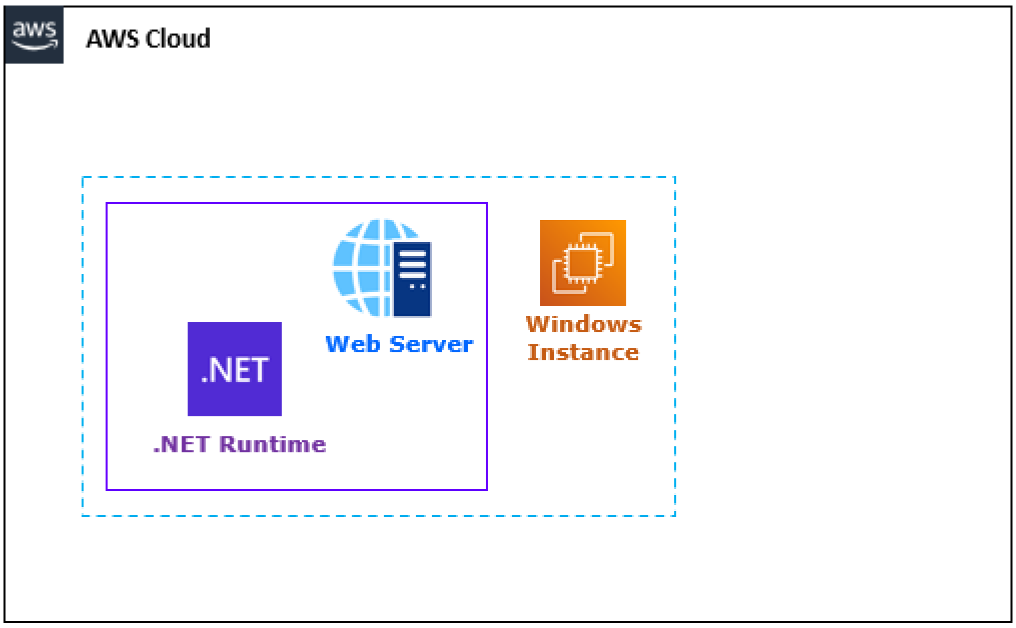
Note: Go to the next task **(but don't close the Windows console)**.

Task 4: Deploy the C# Code

In this task, you will install the runtime environment for C# (dot net based) code.

Step 1: Install the Runtime Environment

In this step, you will install the Web Server and Dot Net Run-time environment to deploy the C# code.



29.From the **C# App Server** (Windows virtual machine) instance ***console***:

i. Right click on **Start** & **Run**.

ii. In the open, write **powershell.exe**, press **Ok**.

iii. **Install** the below commands (one by one) to ready the runtime environment.

Note: You need to wait after every command to complete succesfully before executing the next command.

Note: You can also copy the command from **Runtime Environment Command**.txt file, which is available with the Lab Manual.

a. Enable-WindowsOptionalFeature -Online -FeatureName IIS-WebServerRole

b. Enable-WindowsOptionalFeature -Online -FeatureName IIS-WebServer

c. Enable-WindowsOptionalFeature -Online -FeatureName IIS-ApplicationDevelopment

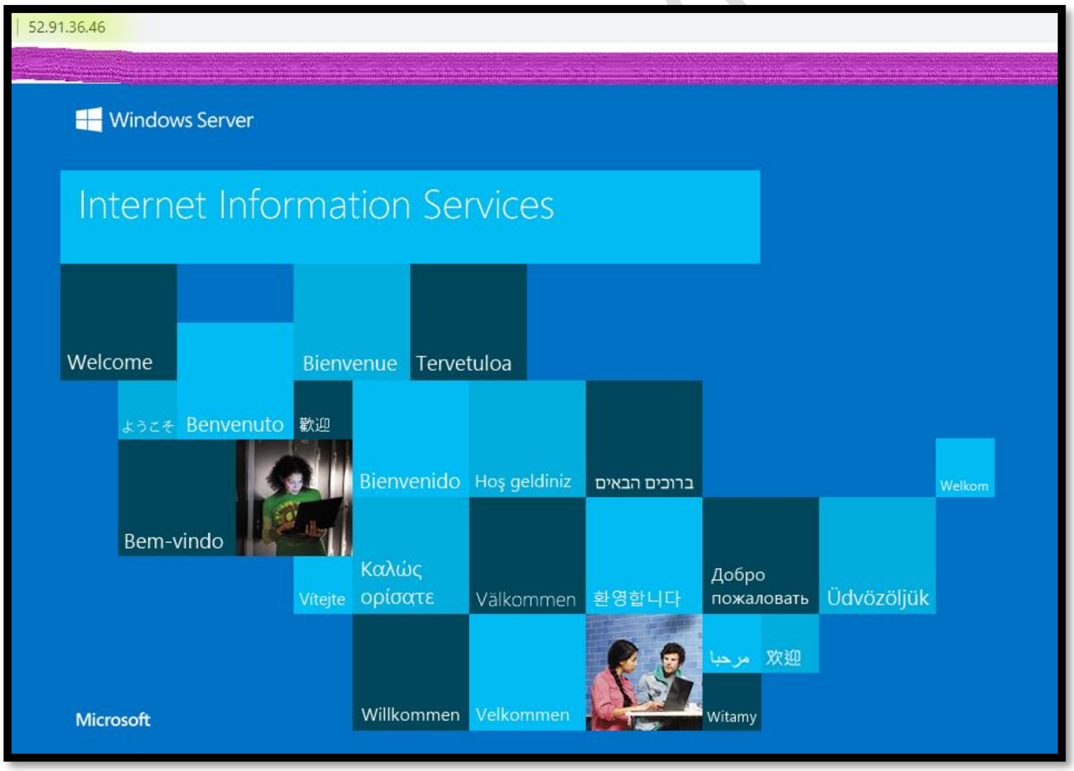
d. Enable-WindowsOptionalFeature -Online -FeatureName IIS-IIS6ManagementCompatibility

e. Enable-WindowsOptionalFeature -Online -FeatureName IIS-ASPNet45 -All

f. Enable-WindowsOptionalFeature -Online -FeatureName NetFx4Extended-ASPNET45

Step 2: Access the C# App Server

30. From the **Web browser**, type **Public IP Address** of **Dot Net Application Server** (Windows virtual machine) and access your **Dot Net website.**

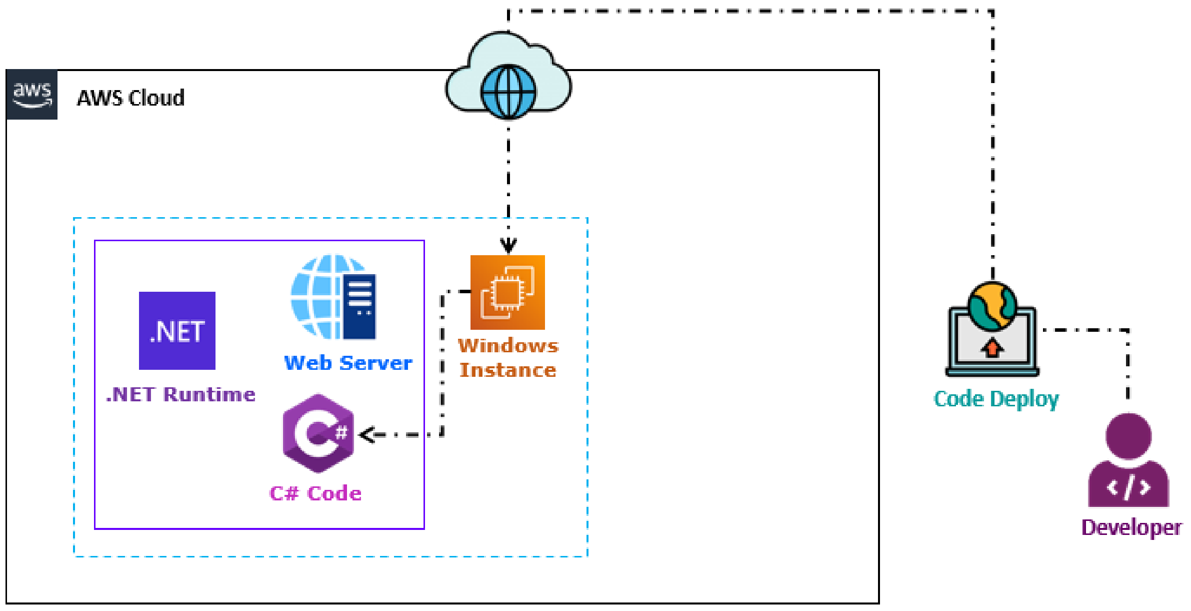


Note: You will see the Default IIS web page.

Note: Go to the next step, **But Don't close the IIS web page**.

Step 4: Deploy the C# Code

In this step, you will deploy the C# code.



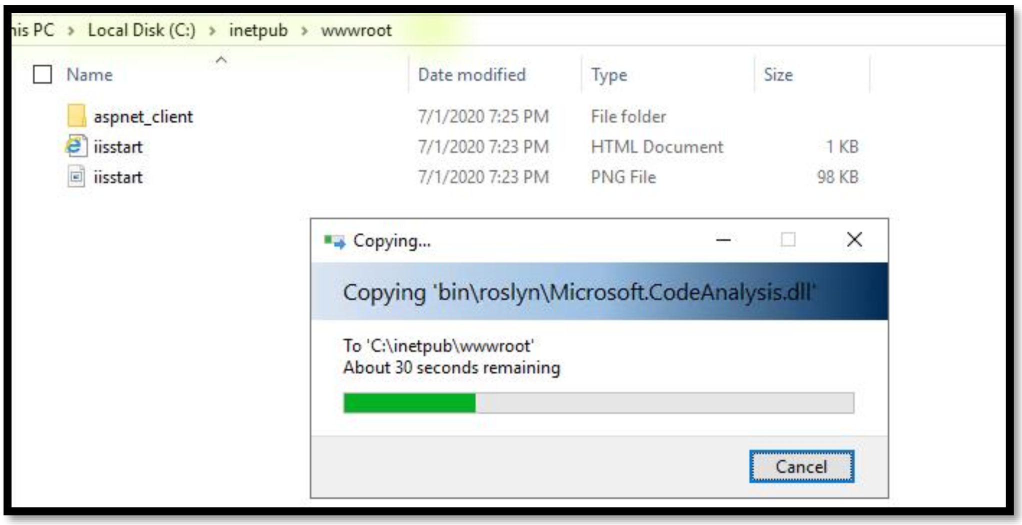
31. **Return** to the **C# App Server** (Windows 2019).

  i. From the **C# App Server** (Windows 2019), right click on **Start** & **Run**.

  ii. In the **Open**, type **c:\inetpub\wwwroot**, Press **Ok**.

  iii. **Copy** the code structure from local laptop/ desktop to the **wwwroot** folder.

Note: You need to copy the code structure (folder and files), not the zip file.


is PC > Local Disk (C:) > inetpub > wwwroot

Name	Date modified	Type	Size
aspnet_client	7/1/2020 7:25 PM	File folder	
iisstart	7/1/2020 7:23 PM	HTML Document	1 KB
iisstart	7/1/2020 7:23 PM	PNG File	98 KB

Copying...

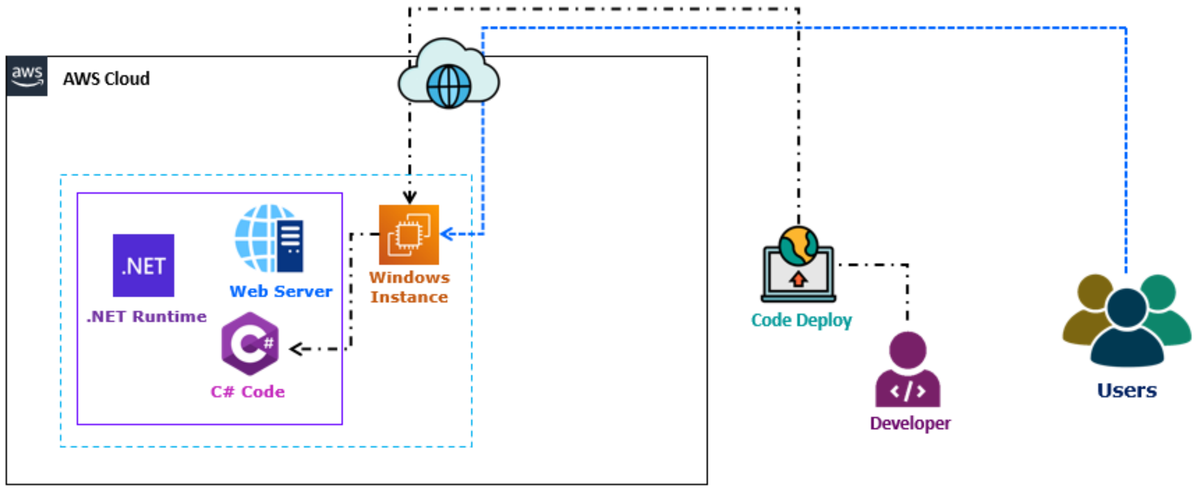
Copying 'bin\roslyn\Microsoft.CodeAnalysis.dll'

To 'C:\inetpub\wwwroot'
About 30 seconds remaining

Cancel

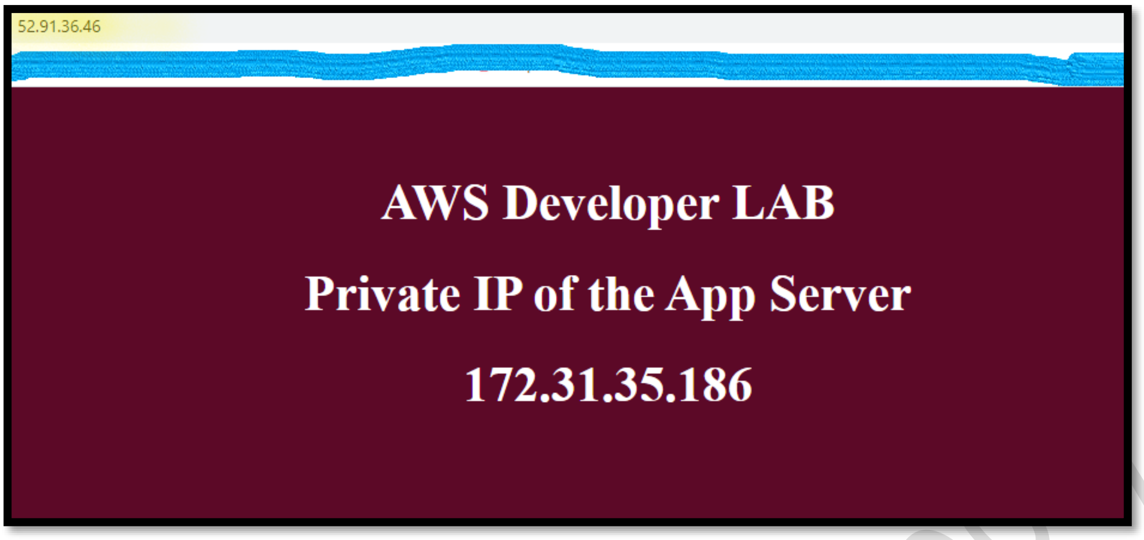
Step 5: Access the Dot App Server

In this step, you will access your web server deployed with C# code.



32. **Return** to the **Web browser** and **Refresh** the web page.

Note: You will see the C# application web page.



52.91.36.46

AWS Developer LAB

Private IP of the App Server

172.31.35.186

Note: Dot Net Application web page also display the **Dot Net Application Server** (Windows virtual machine) Private IP address.

Task 5: Clean up the Environment

Step 1: Terminate EC2 Instances

33. In the **AWS Management Console**, on the **Services** menu, click **EC2**.
34. Click **Instances**.
35. Select **C# App Server**.
    i. Click on **Instance state**.
    ii. Select **Terminate instance**.
    iii. Select **Terminate**.