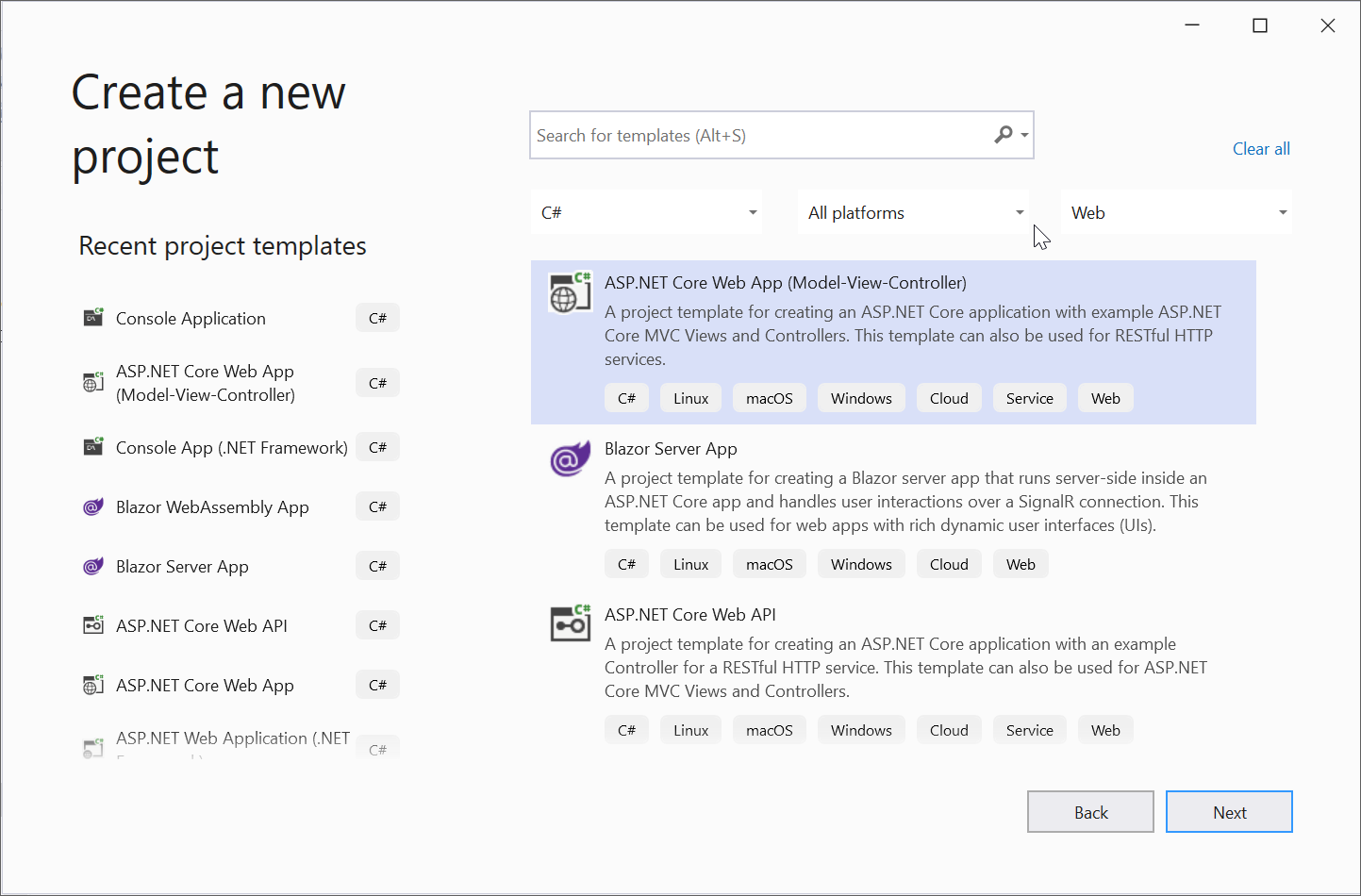
**Convert HTML to PDF in Azure App Service Using .NET Core for Reliable Document Rendering**

Syncfusion® [HTML to PDF converter](https://www.syncfusion.com/document-processing/pdf-framework/net/html-to-pdf) is a powerful [.NET library](https://www.syncfusion.com/document-processing/pdf-framework/net) designed to convert HTML, webpages, and various document formats to PDF while maintaining the original layout and content integrity. This guide explains how to employ this converter in an Azure App Service on Linux using ASP.NET Core.

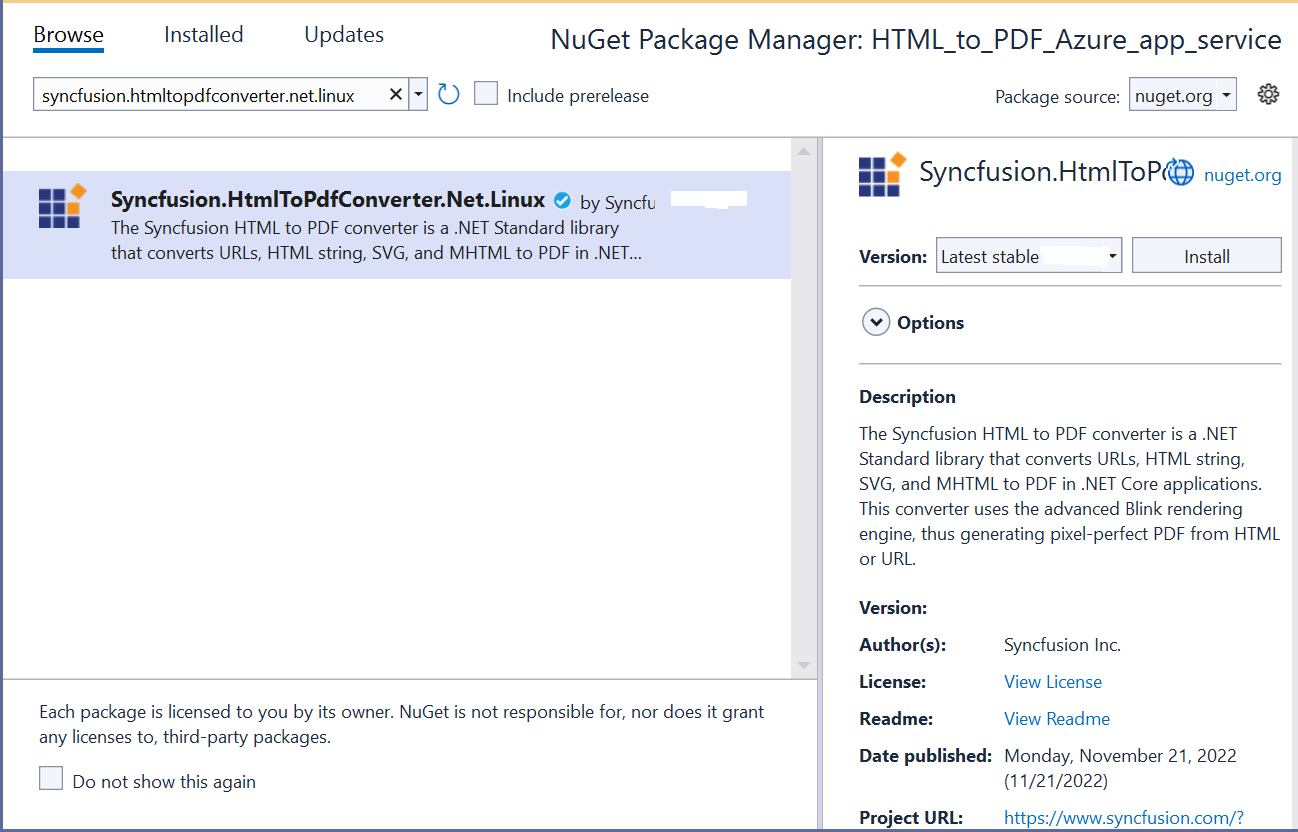
**Steps to Convert HTML to PDF in Azure App Service on Linux**

1. **Create an ASP.NET Core MVC Application**: Initiate a project using ASP.NET Core Model-View-Controller pattern.

A screenshot of a computer

AI-generated content may be incorrect.

2. **Install Syncfusion Package**: Add [Syncfusion.HtmlToPdfConverter.Net.Linux](https://www.nuget.org/packages/Syncfusion.HtmlToPdfConverter.Net.Linux/) from NuGet as a project reference.

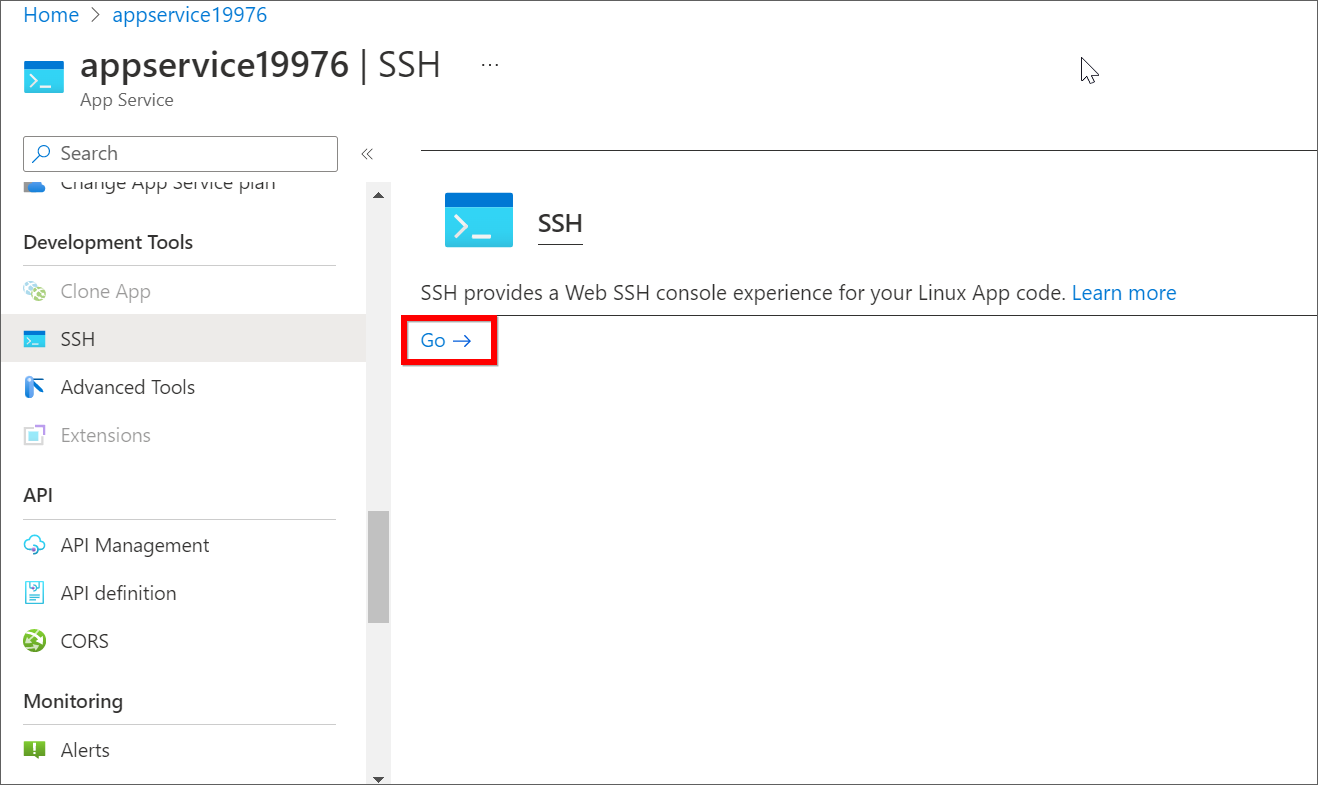


3. **Install Dependency Packages**: There are two methods to install the required packages on Azure:

**3.1 Using SSH command line:**

1.After publishing the web application, log in to the [Azure Portal](https://portal.azure.com) and open the published App Service.

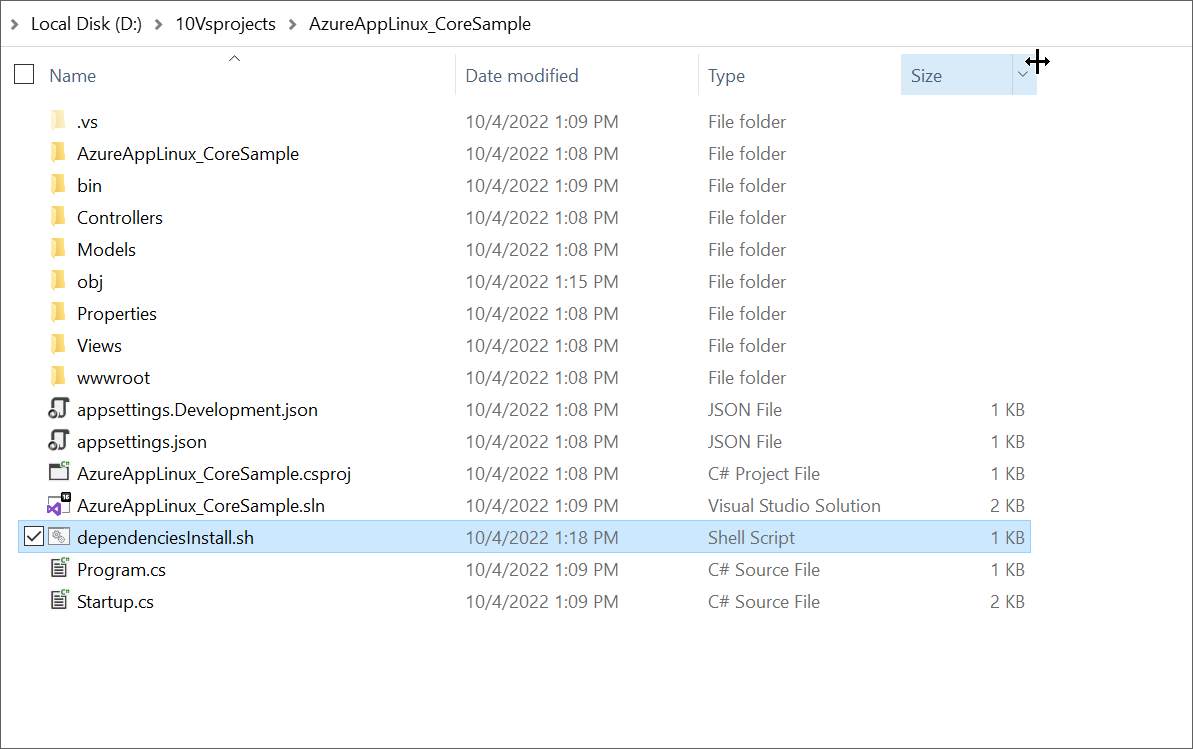
2.Under the **Development Tools** section, select **SSH**, then click the **Go** link to launch the terminal.

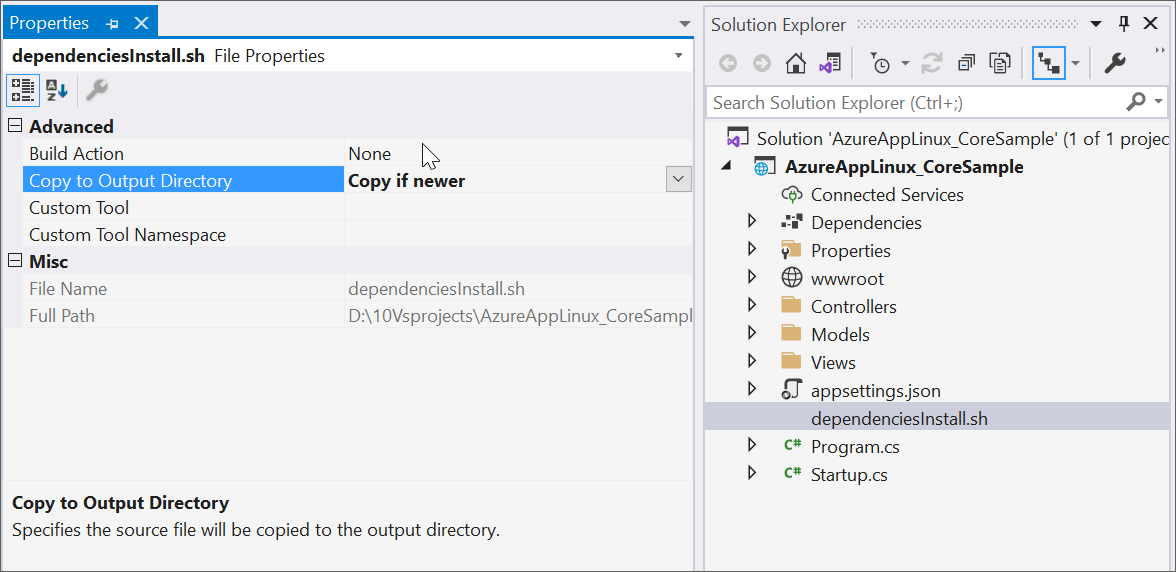
 

3. From the terminal window, you can install the required dependency packages. Use the following single command to install all necessary dependencies:

|  |
| --- |
| apt-get update && apt-get install -yq --no-install-recommends libasound2 libatk1.0-0 libc6 libcairo2 libcups2 libdbus-1-3 libexpat1 libfontconfig1 libgcc1 libgconf-2-4 libgdk-pixbuf2.0-0 libglib2.0-0 libgtk-3-0 libnspr4 libpango-1.0-0 libpangocairo-1.0-0 libstdc++6 libx11-6 libx11-xcb1 libxcb1 libxcursor1 libxdamage1 libxext6 libxfixes3 libxi6 libxrandr2 libxrender1 libxss1 libxtst6 libnss3 libgbm1 |

**3.2 Running the commands from C#**

1.Create a shell script named dependenciesInstall.sh in your project, containing the necessary commands to install dependency packages.

2. To ensure the script is included during deployment, set the **Copy to Output Directory** property of the dependenciesInstall.sh file to **Copy if newer**.

3. To execute the script during the startup application, include the following code in the Configure method of the Startup.cs file:

|  |
| --- |
| private void InstallDependencies(string shellFilePath)  {  Process process = new Process  {  StartInfo = new ProcessStartInfo  {  FileName = "/bin/bash",  Arguments = "-c " + shellFilePath,  CreateNoWindow = true,  UseShellExecute = false,  }  };  process.Start();  process.WaitForExit();  } |

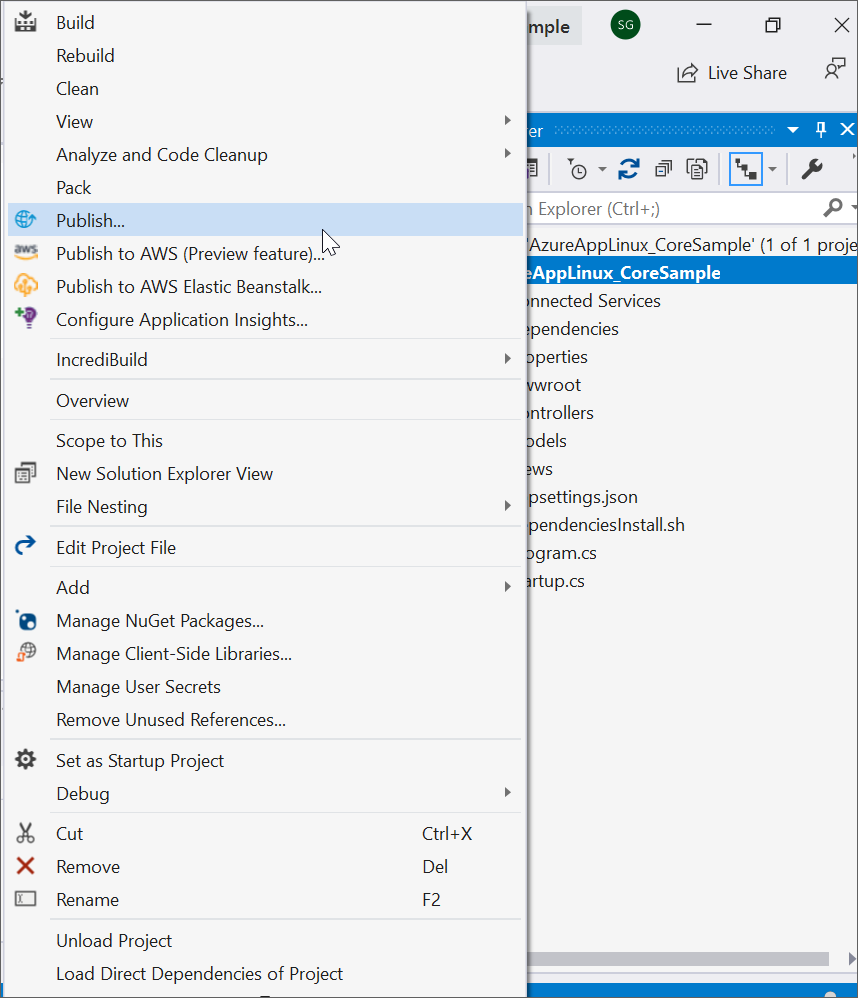
4. **Implement HTML to PDF Conversion**: Add an export button in index.cshtml and include conversion logic in the controller:

|  |
| --- |
| @{ Html.BeginForm("ExportToPDF", "Home", FormMethod.Post); }  <input type="submit" value="Export To PDF" class="btn" />  @Html.EndForm() |

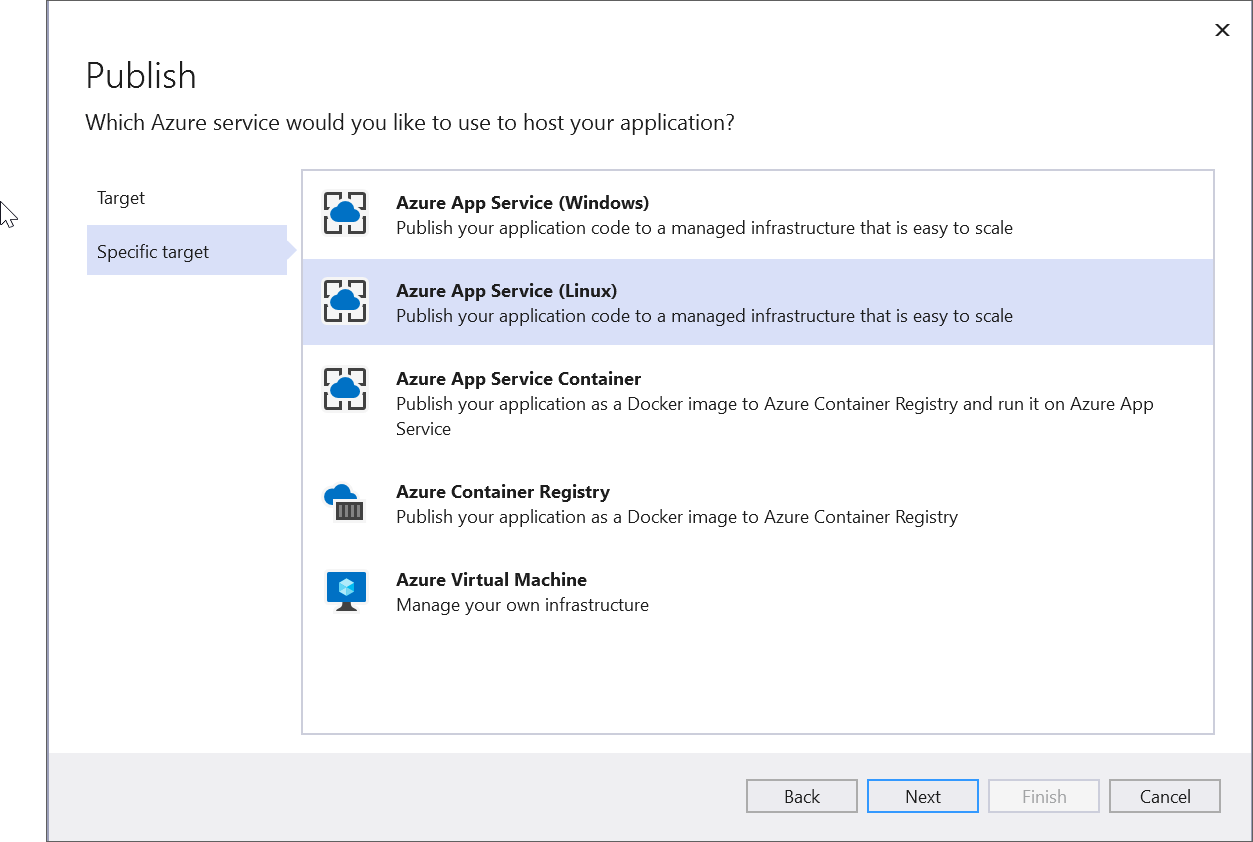
|  |
| --- |
| public IActionResult ExportToPDF()  {  Environment.SetEnvironmentVariable("ASPNETCORE\_ENVIRONMENT", "Development");    HtmlToPdfConverter htmlConverter = new HtmlToPdfConverter();  BlinkConverterSettings settings = new BlinkConverterSettings();  settings.CommandLineArguments.Add("--no-sandbox");  settings.CommandLineArguments.Add("--disable-setuid-sandbox");    htmlConverter.ConverterSettings = settings;    PdfDocument document = htmlConverter.Convert("http://www.syncfusion.com");  MemoryStream stream = new MemoryStream();  document.Save(stream);  stream.Position = 0;  document.Close(true);    string contentType = "application/pdf";  string fileName = "Output.pdf";  return File(stream, contentType, fileName);  } |

**Refer to the following steps to publish as Azure App Linux:**

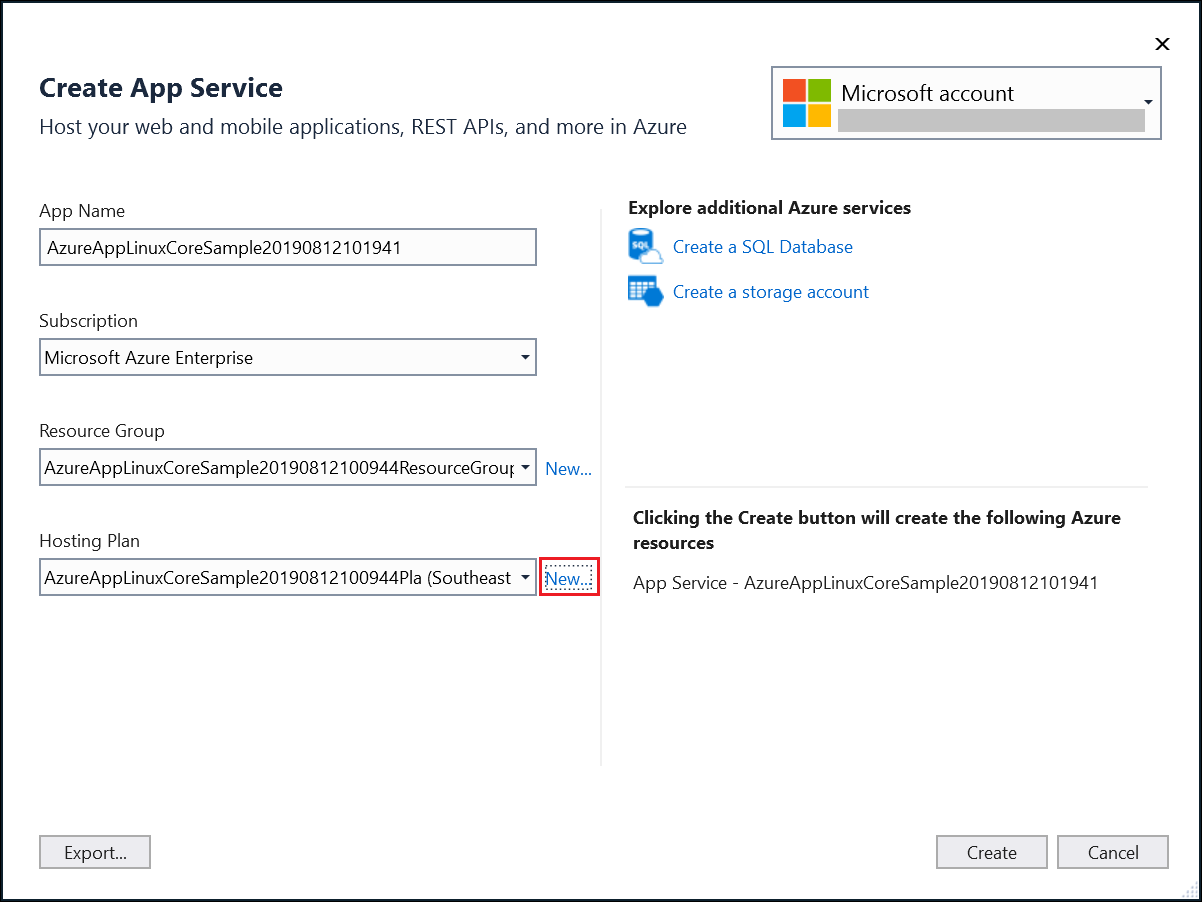
5.Right-click the project and select **Publish.**



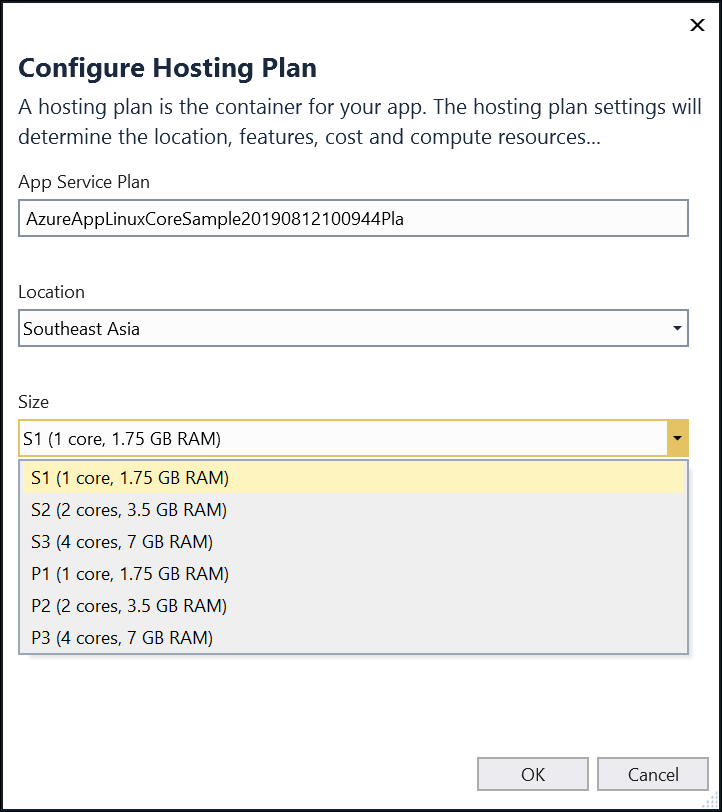
6.Create a new profile in publish target window.



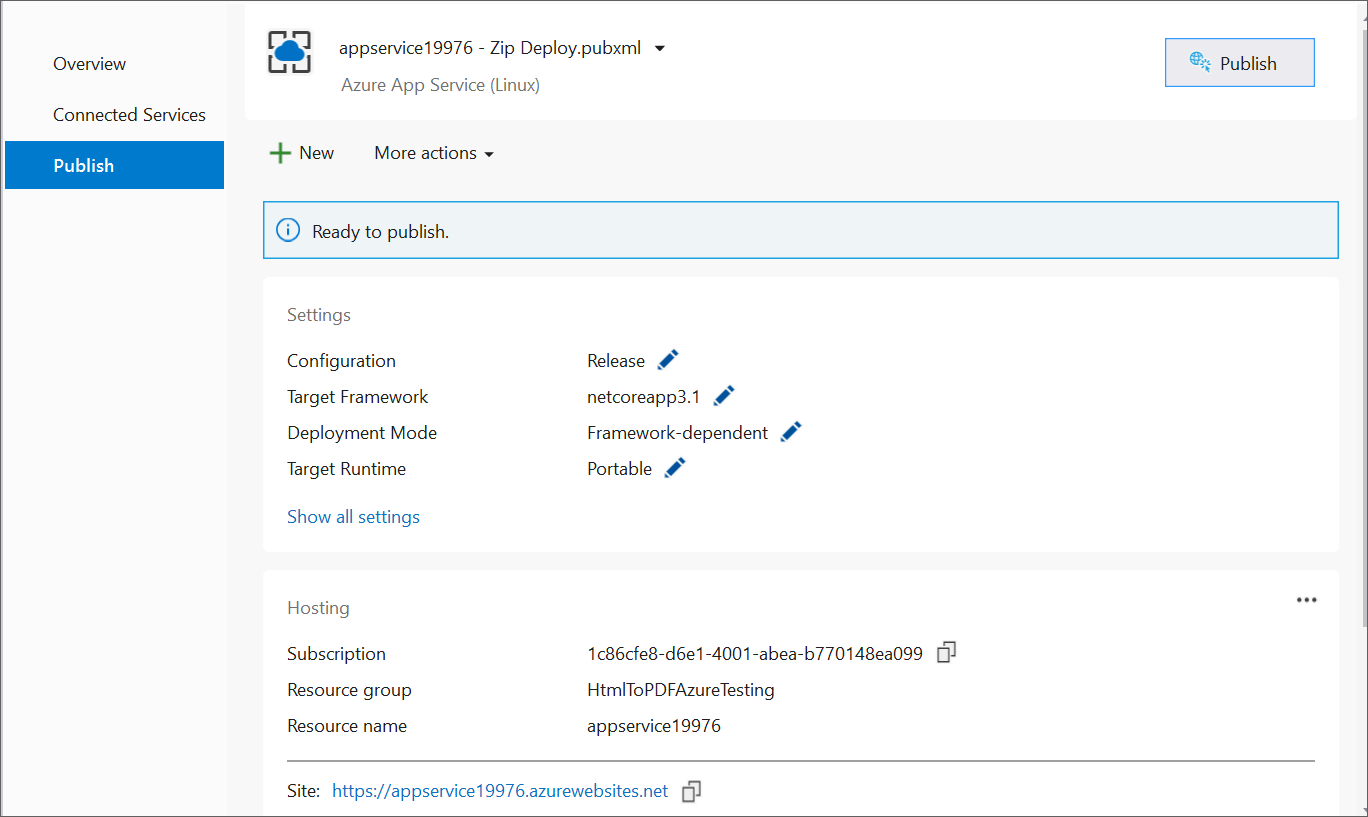
7.Create App service using Azure subscription and select a hosting plan.



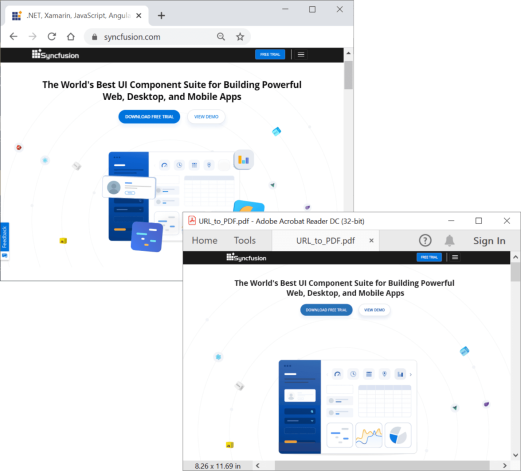
Note: HTML to PDF conversion is supported only on **Basic (B1) to Premium (P3)** hosting plans. It will **not work** on **Free** or **Shared** hosting plans. Please ensure you select an appropriate hosting plan based on your requirements.



9.After creating a profile, click the publish button.



10.Now, the published webpage will open in the browser. Click **Export to PDF** to convert the Syncfusion® webpage to a PDF.



A complete working sample for converting HTML to PDF in an Azure App Service on Linux can be downloaded from [AzureAppLinux\_CoreSample.zip](https://www.syncfusion.com/downloads/support/directtrac/general/ze/AzureAppLinux_CoreSample-2144256654).

Take a moment to explore the [documentation](https://help.syncfusion.com/file-formats/pdf/converting-html-to-pdf), where you'll find detailed guidance on converting HTML pages to PDF documents, along with various customization options and advanced features.  
**Conclusion**

I hope you enjoyed learning about how to HTML to PDF in Azure app using ASP.NET Core.

You can refer to our [**ASP.NET Core PDF**](https://www.syncfusion.com/document-processing/pdf-framework/net-core) feature tour page to know about its other groundbreaking feature representations and [**documentation**](https://help.syncfusion.com/aspnet-core/pdf/getting-started), and how to quickly get started for configuration specifications. You can also explore our [**ASP.NET Core PDF example**](https://www.syncfusion.com/demos/fileformats/pdf-library) to understand how to create and manipulate data in the .NET PDF.

For current customers, you can check out our Document processing libraries from the [**License and Downloads**](https://www.syncfusion.com/account/downloads) page. If you are new to Syncfusion®, you can try our 30-day [**free trial**](https://www.syncfusion.com/downloads/aspnetcore-js2) to check out our ASP.NET Core PDF and other .NET Core controls.