

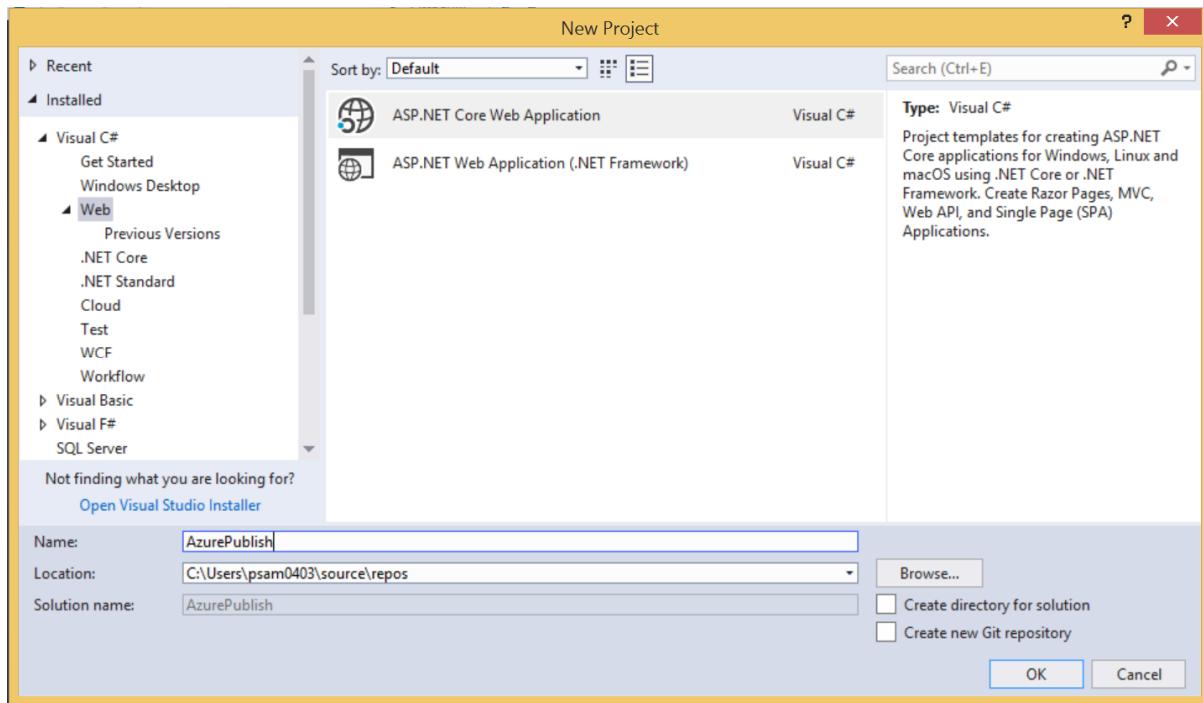
How to Publish ASP.NET Core App to Azure through FTP

In this tutorial I'm going to show how to:

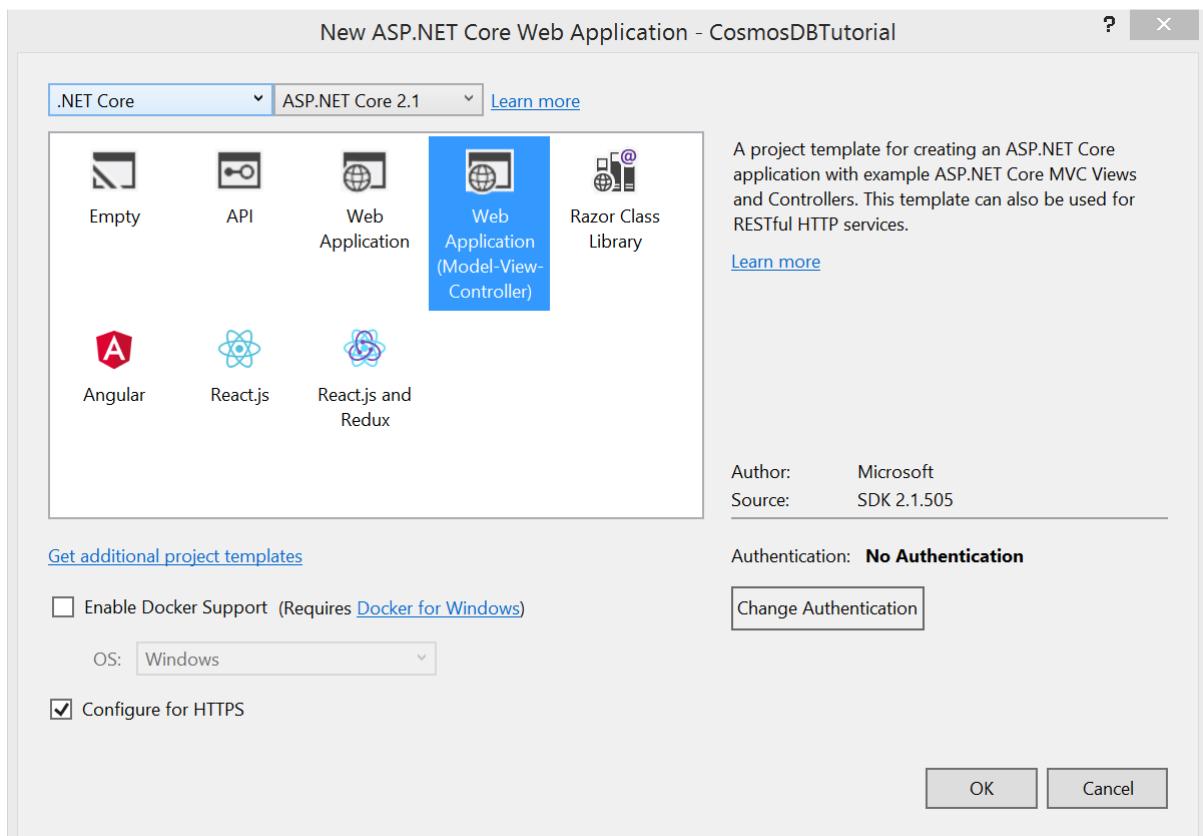
- Create an ASP.NET Core MVC Application
- Create an App Service Plan
- Create an App Service
- Publish the App using Folder option
- Publish the code to Azure through FTP Client
- Publish the code to Azure through FTP Deployment
- Use FTP to modify the content

Step 01: Create an ASP.NET Core MVC Application

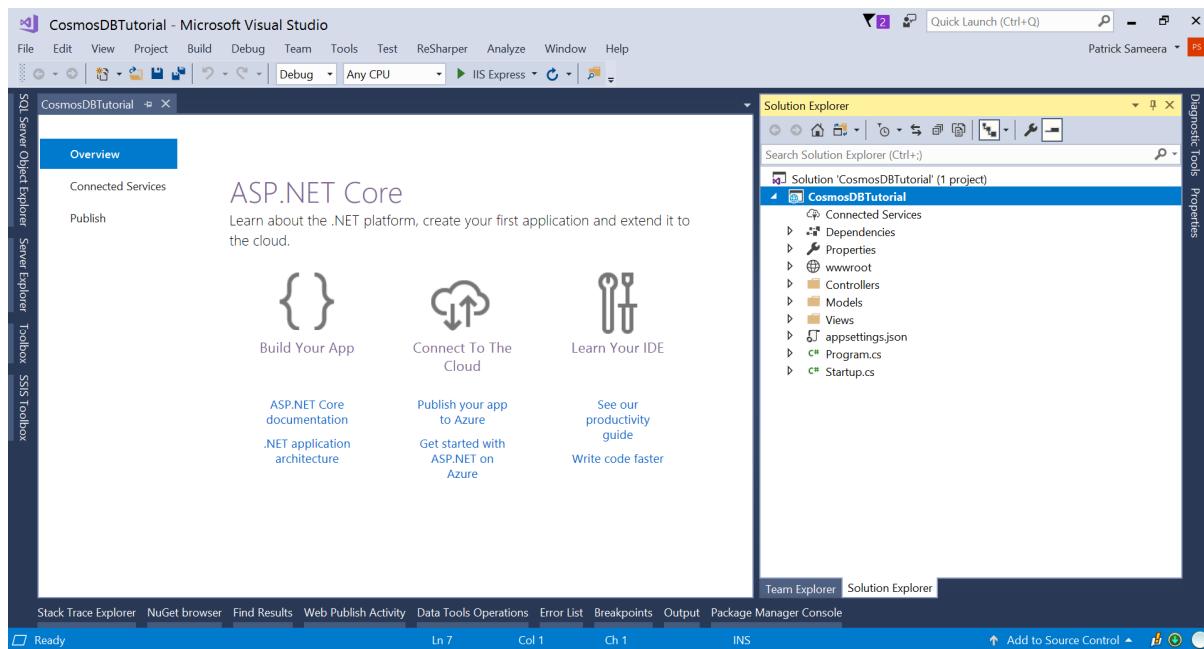
Create a new ASP .Net Core Web Project – Name it AzurePublish.



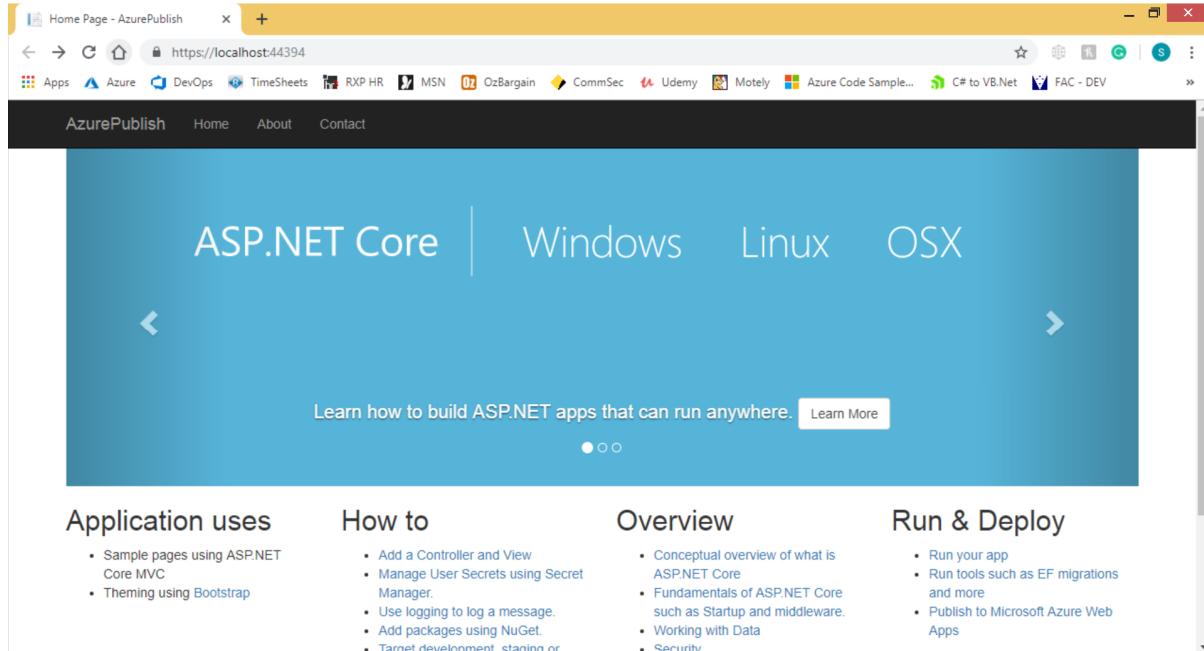
- Net Core 2.1
- MVC



Once Project gets created Click F5 to run the Project, just to make sure it compiles properly.



It runs properly.



Step 02: Create an App Service Plan

I have a Resource Group already existing and I will be using that resource group for this tutorial.

- Name: patsam-rg
- Location: Central US

The screenshot shows the 'Resource groups' blade in the Azure portal. At the top, there are buttons for '+ Add', 'Edit columns', 'Refresh', 'Export to CSV', 'Assign tags', and 'Feedback'. Below these are filters for 'Filter by name...', 'Subscription == all', 'Location == all', and an 'Add filter' button. A dropdown menu shows 'No grouping'. The main table lists one record: 'patsam-rg' under 'Name', 'Visual Studio Enterprise - MPN' under 'Subscription', and 'Central US' under 'Location'. There are also 'Subscription ↑↓' and 'Location ↑↓' sorting arrows.

Go to App Service Plans and click + Add button to create a new App Service Plan.

The screenshot shows the 'App Service plans' blade in the Azure portal. At the top, there are buttons for '+ Add', 'Edit columns', 'Refresh', and 'Assign tags'. A message 'Subscriptions: Visual Studio Enterprise – MPN – Don't see a subscription? [Open Directory + Subscription settings]' is displayed. Below are filters for 'Filter by name...', 'All resource groups', 'All locations', 'All tags', and 'No grouping'. The table header includes columns for 'Name ↑↓', 'Apps', 'Pricing Tier', 'Resource group ↑↓', and 'Subscription ↑↓'. The main area displays a placeholder icon and the text 'No app service plans to display'. A note below explains what App Service plans are and provides a link to learn more. A prominent blue button at the bottom center says 'Create app service plan'.

Provide relevant details.

- Resource group: patsam-rg
- App Service Plan Name: patsam-ps
- Operating system: Windows
- Sku and size: Free F1

The screenshot shows the 'App Service Plan' creation page in the Azure portal. The 'Basics' tab is selected. In the 'Project Details' section, 'Subscription' is set to 'Visual Studio Enterprise - MPN', 'Resource Group' is set to 'patsam-rg', and 'Name' is set to 'patsam-ps'. Under 'Operating System', 'Windows' is selected. The 'Region' is set to 'Central US'. In the 'Pricing Tier' section, 'Free F1' is selected. At the bottom, there are buttons for 'Review + create', '< Previous', and 'Next : Tags >'.

Click on Change Size to select the pricing tier. For this example, I'll be using Free F1.

The screenshot shows the 'Spec Picker' dialog. It has three main sections: 'Dev / Test' (orange), 'Production' (grey), and 'Isolated' (green). The 'Production' section is currently selected. Below it, the 'Recommended pricing tiers' section shows three options: 'F1 Shared infrastructure, 1 GB memory, 60 minutes/day compute, Free' (highlighted with a blue border), 'D1 Shared infrastructure, 1 GB memory, 240 minutes/day compute, 13.03 AUD/Month (Estimated)' (purple), and 'B1 100 total ACU, 1.75 GB memory, A-Series compute equivalent, 44.10 AUD/Month (Estimated)' (green). A link 'See additional options' is shown below. To the right, under 'Included hardware', there are sections for 'Azure Compute Units (ACU)', 'Memory', and 'Storage'. At the bottom is a large 'Apply' button.

Click Review + Create button.

The screenshot shows the 'Microsoft.Web-ASP-Portal-75fa8464-8519 - Overview' page. On the left, there's a navigation menu with 'Overview', 'Inputs', 'Outputs', and 'Template'. The main area has a green checkmark icon and the message 'Your deployment is complete'. Below it, deployment details are listed: Deployment name: Microsoft.Web-ASP-Portal-75fa8464-8519, Subscription: Visual Studio Enterprise – MPN, Start time: 11/14/2019, 9:36:42 PM, Correlation ID: dae92244-1b0b-487b-b758-d479eead3694, and Resource group: patsam-rg. There are sections for 'Deployment details' (with a 'Download' link) and 'Next steps' (with a 'Go to resource' button). A search bar at the top left says 'Search (Ctrl+I)'.

We can see the newly created App Service Plan.

The screenshot shows the 'App Service plans' page. At the top, it says 'App Service plans' and 'RDP Services Ltd'. Below that are buttons for '+ Add', 'Edit columns', 'Refresh', and 'Assign tags'. A note says 'Subscriptions: Visual Studio Enterprise – MPN – Don't see a subscription? Open Directory + Subscription settings'. There are filters for 'Filter by name...', 'All resource groups', 'All locations', 'All tags', and 'No grouping'. The table below shows one item: 'Name' (patsam-sp), 'Apps' (1), 'Pricing Tier' (patsam-sp (F1: Free)), 'Resource group' (patsam-rg), and 'Subscription' (Visual Studio Enterprise – M...).

Name	Apps	Pricing Tier	Resource group	Subscription
patsam-sp	1	patsam-sp (F1: Free)	patsam-rg	Visual Studio Enterprise – M...

Step 03: Create an App Service

Go to App Service and click + Add button to create a new App Service.

The screenshot shows the 'App Services' blade in the Azure portal. At the top, there are navigation links for 'Home > App Services' and a 'RXP Services Ltd' account indicator. Below the header is a toolbar with buttons for '+ Add', 'Edit columns', 'Refresh', 'Export to CSV', 'Assign tags', 'Start', 'Restart', 'Stop', 'Delete', 'Feedback', and 'More'. A search bar labeled 'Filter by name...' is highlighted with a blue border. To its right are filters for 'Subscription == all', 'Resource group == all', 'Location == all', and a 'No grouping' dropdown. Below the toolbar, it says 'Showing 0 to 0 of 0 records.' and lists columns: Name ↑↓, Status ↑↓, Location ↑↓, Pricing Tier ↑↓, App Service Plan ↑↓, Subscription ↑↓, and App. In the center, there's a large circular placeholder icon with a network-like pattern. Below the icon, the text 'No app services to display' is centered. At the bottom, there's a descriptive paragraph about building web, mobile, and API apps using .NET, Java, Node.js, PHP, and Python, followed by a 'Learn more' link. A prominent blue 'Create app service' button is located at the bottom center.

Provide relevant details.

- Resource group: patsam-rg
- App Service Name: patsam
- Publish: Code
- Runtime stack: .Net Core 2.1 (since we created a .Net Core 2.1 MVC Web Application)
- Operating system: Windows
- App Service Plan: patsam-sp (App Service Plan we created earlier)

The screenshot shows the 'Basics' step of the 'Web App' creation wizard in the Azure portal. The page title is 'Home > App Services > Web App'. The sub-section title is 'Web App'. The tabs at the top are 'Basics' (selected), 'Monitoring', 'Tags', and 'Review + create'. The main content area is titled 'Project Details' and includes fields for 'Subscription' (Visual Studio Enterprise – MPN) and 'Resource Group' (patsam-rg). Below this is the 'Instance Details' section with fields for 'Name' (patsam), 'Publish' (Code selected), 'Runtime stack' (.NET Core 2.1 (LTS)), 'Operating System' (Windows selected), and 'Region' (Central US). The 'App Service Plan' section shows a dropdown for 'Windows Plan (Central US)' which is currently empty. At the bottom are buttons for 'Review + create', '< Previous', and 'Next : Monitoring >'.

Click Review + Create button.

The screenshot shows the 'Microsoft.Web-WebApp-Portal-1a7942a8-8d9e - Overview' page. At the top, there's a search bar and navigation buttons for Delete, Cancel, Redeploy, and Refresh. On the left, a sidebar has 'Overview' selected, along with 'Inputs', 'Outputs', and 'Template'. The main content area displays a green checkmark icon and the message 'Your deployment is complete'. It provides deployment details: Deployment name: Microsoft.Web-WebApp-Portal-1a7942a8-8d9e, Start time: 11/14/2019, 9:45:40 PM, Subscription: Visual Studio Enterprise – MPN, Correlation ID: 6c1d0f77-207d-49d9-b08d-a14b6b072, and Resource group: patsam-rg. There are collapsed sections for 'Deployment details' (with a download link) and 'Next steps', and a 'Go to resource' button.

We can see the newly created App Service.

The screenshot shows the 'App Services' blade under the 'App Services' section. It includes standard filtering and sorting options like 'Add', 'Edit columns', 'Refresh', 'Export to CSV', 'Assign tags', 'Start', 'Restart', 'Stop', 'Delete', 'Feedback', and 'Leave preview'. Below these are filters for 'Name', 'Subscription', 'Resource group', 'Location', and an 'Add filter' button, along with a 'No grouping' dropdown. The table lists one record: 'patsam' (Status: Running, Location: Central US, Pricing Tier: Free, App Service Plan: patsam-sp, Subscription: Visual Studio Enterpri..., App Type: Web App).

Next go inside the newly created App Service. Click on it.

The screenshot shows the 'App Services' blade in the Azure portal. It lists one record: 'patsam'. The details are as follows:

Name	Status	Location	Pricing Tier	App Service Plan	Subscription	App Type
patsam	Running	Central US	Free	patsam-sp	Visual Studio Enterprise - MPN	Web App

Click on Browse button to check whether we have created the App Service successfully.

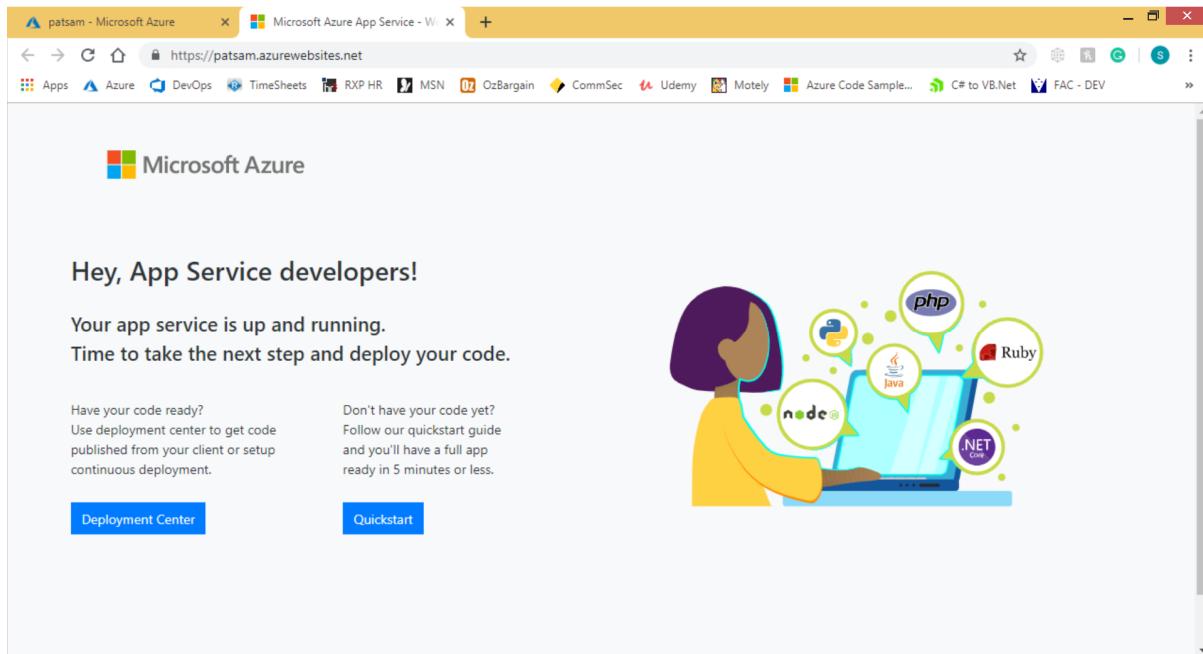
The screenshot shows the 'Overview' page for the 'patsam' app service. Key details displayed include:

- Resource group: patsam-rg
- Status: Running
- Location: Central US
- Subscription: Visual Studio Enterprise - MPN
- URL: https://patsam.azurewebsites.net
- App Service Plan: patsam-sp (F1: Free)
- FTP/deployment username: patsam\deploy-patrick-sameera
- FTP hostname: ftp://waws-prod-dm1-155.ftp.azurewebsites.windows.net
- FTPS hostname: https://waws-prod-dm1-155.ftp.azurewebsites.windows.net

On the left sidebar, the 'Overview' tab is selected. Other tabs include Activity log, Access control (IAM), Tags, Diagnose and solve problems, Security, Deployment (Quickstart, Deployment slots, Deployment Center), Settings, and Configuration.

We can see it's working fine.

- <https://patsam.azurewebsites.net/>



Next go to the App Service Plan we created. Click on it.

Home > App Service plans

App Service plans

RXP Services Ltd

+ Add Edit columns Refresh Assign tags

Subscriptions: Visual Studio Enterprise – MPN – Don't see a subscription? [Open Directory + Subscription settings](#)

Filter by name... All resource groups All locations All tags No grouping

1 items

Name	Apps	Pricing Tier	Resource group	Subscription	...
patsam-sp	1	patsam-sp (F1: Free)	patsam-rg	Visual Studio Enterprise – MPN	...

Under Apps tab now we can see the App Service (Web App) we created earlier.

Home > App Service plans > patsam-sp - Apps

patsam-sp - Apps App Service plan

Search (Ctrl+ /)

Overview Activity log Access control (IAM) Tags Diagnose and solve problems

Settings

Apps

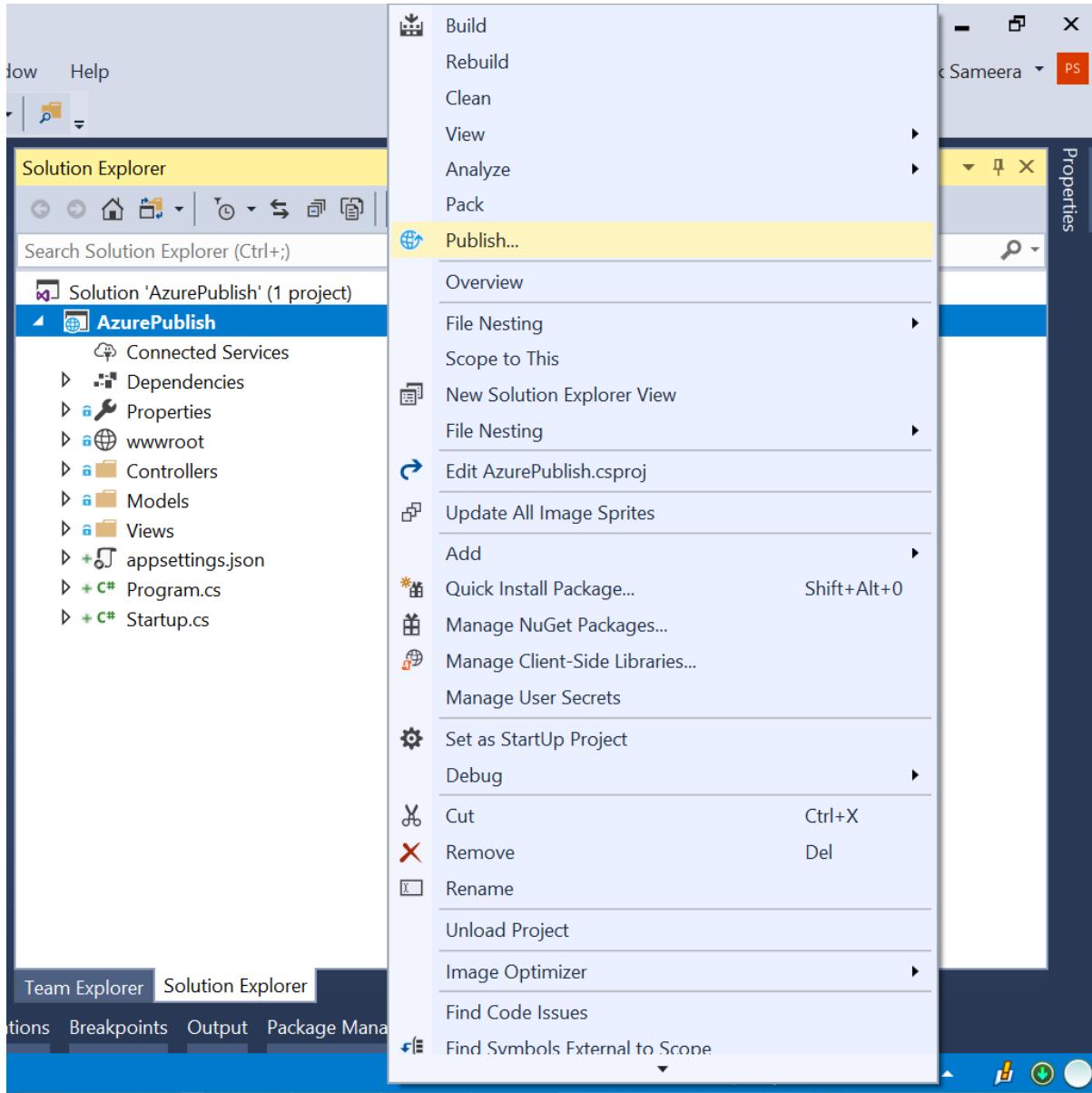
Name Type Resource Group Status

Name	Type	Resource Group	Status
patsam	app	patsam-rg	Running

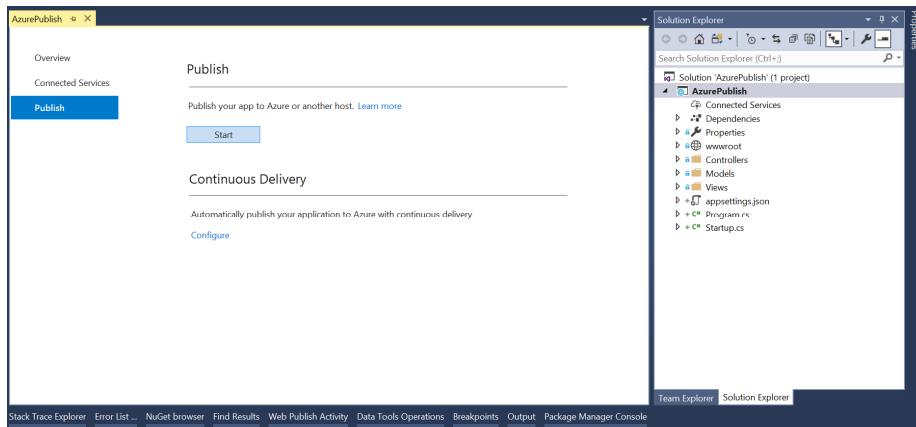
Step 04: Publish the App using Folder option

Next, we need to Publish the App to a local Folder.

Right click on the project and click Publish.



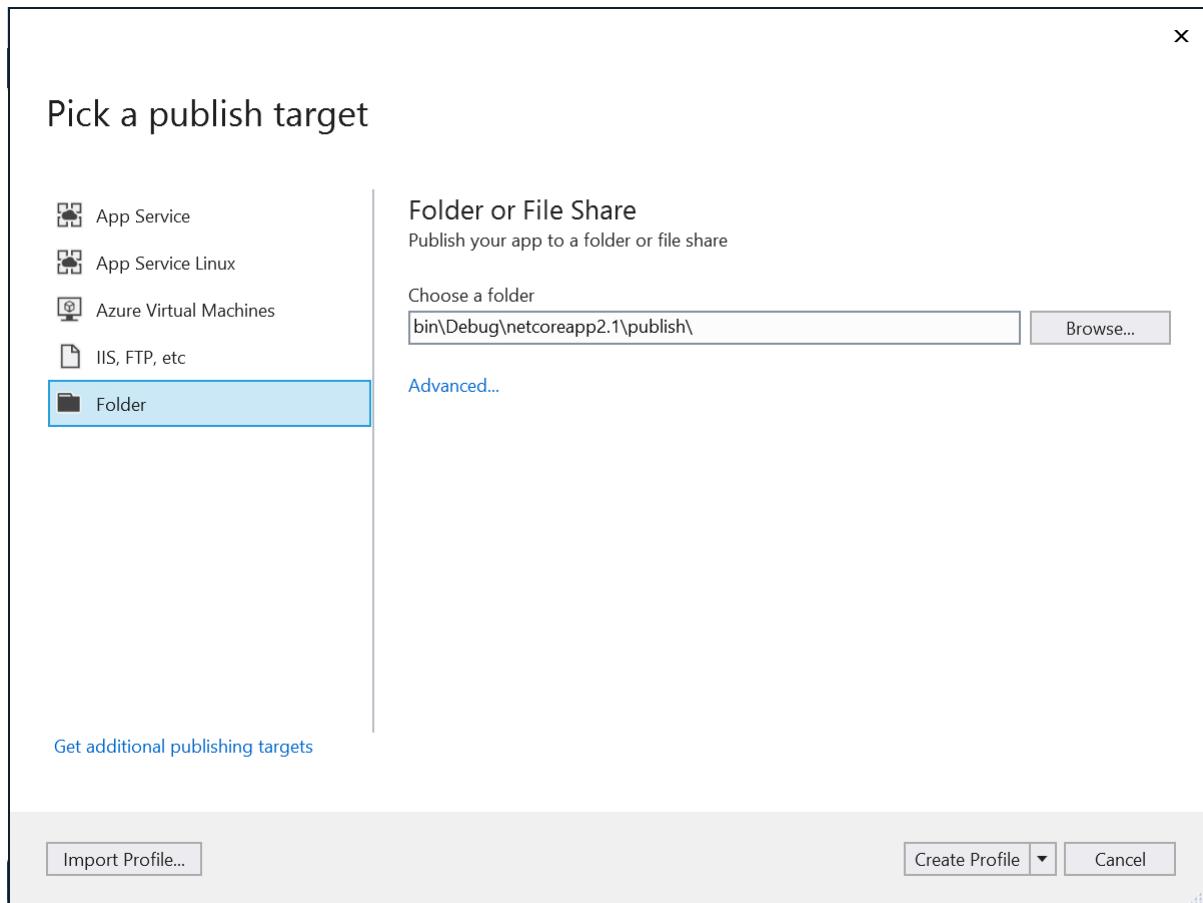
If we haven't set a Publish Profile, then click Start.



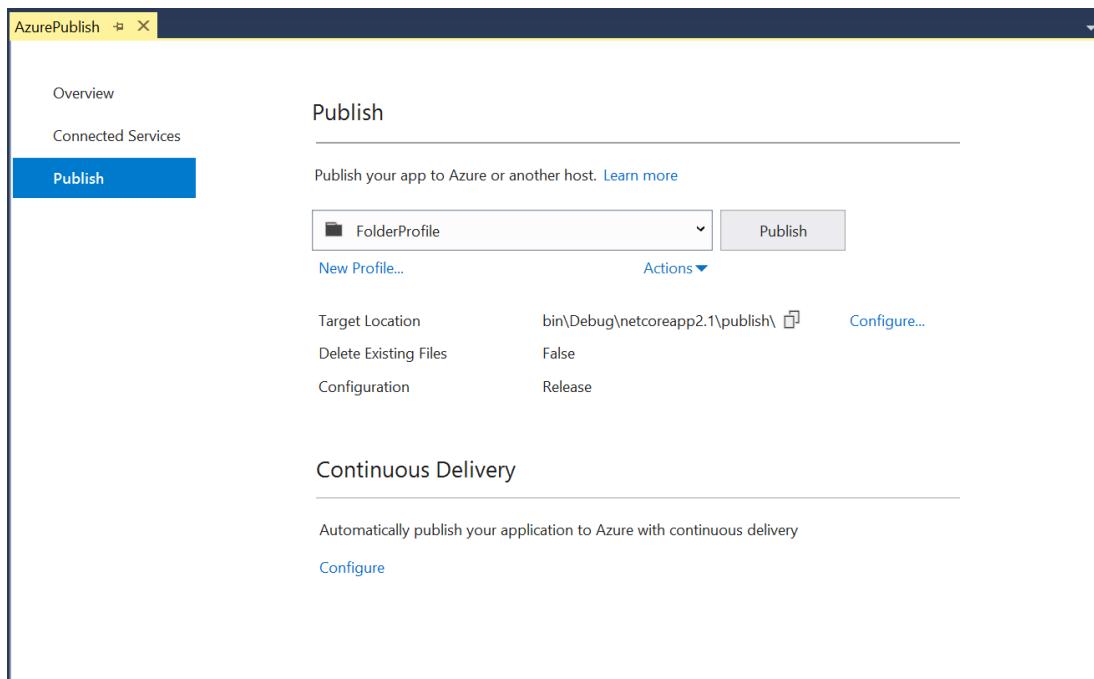
Select Folder option and set the location and click Create Profile button.

In my case I have set `bin\Debug\netcoreapp2.1\publish\`

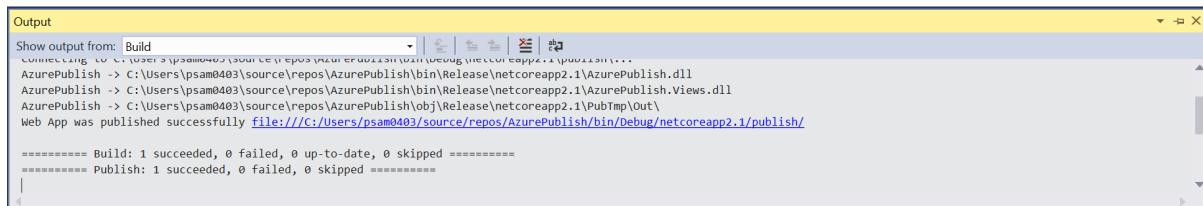
- `C:\Users\psam0403\source\repos\AzurePublish\bin\Debug\netcoreapp2.1\publish\`



Click Publish.

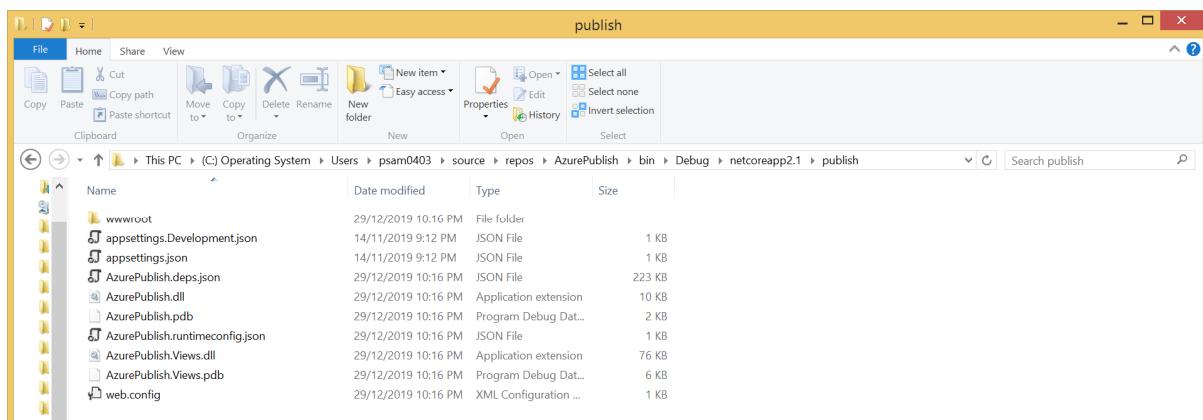


Make sure it gets published correctly.



Go to the Publish folder location and make sure published folder exists.

- `C:\Users\psam0403\source\repos\AzurePublish\bin\Debug\netcoreapp2.1\publish`



Step 05: Publish the code to Azure through FTP Client

Click on Deployment Centre tab under the App Service we created earlier.

The screenshot shows the Azure Deployment Center interface for an app service named 'patsam'. The left sidebar has 'Deployment Center' selected under the 'Deployment' section. The main area displays four deployment methods: OneDrive, Dropbox, External, and FTP. The 'FTP' option is highlighted with a blue border, indicating it is the active choice.

Select FTP and click Dashboard.

This screenshot is similar to the previous one, but the 'Dashboard' button at the bottom right of the main content area is highlighted with a blue border, indicating it is the next step to be clicked.

It will provide you the FTP Credential details.

- Endpoint URL: `ftps://waws-prod-dm1-155.ftp.azurewebsites.windows.net/site/wwwroot`
- Username: `patsam\$patsam`
- Password: `***`

The screenshot shows the 'FTP' section of the Azure App Service configuration. It displays the following information:

- FTPS Endpoint:** `ftps://waws-prod-dm1-155.ftp.azurewebsites.windows.net` (with a `Copy` button)
- App Credentials:** This tab is selected, showing:
 - Application Credentials:** Auto-generated for the specific app or deployment slot, used for FTP, Local Git, and WebDeploy. They cannot be manually configured but can be reset.
 - Username:** `patsam\$patsam` (with a `Copy` button)
 - Password:** A masked password (with a `Show` and `Copy` button).
- User Credentials:** This tab is also present but not selected.

Mind you these credentials are user level credentials. Which means these credentials can be used across all apps in all subscriptions associated with the Azure account.

Also, we can get Publish Profile details through Overview tab of the App Service.

Click on Get Publish Profile button to download the profile details.

The screenshot shows the Azure App Service Overview page for the 'patsam' app service. The left sidebar includes links for Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Security, Deployment (Quickstart, Deployment slots, Deployment Center), and Settings (Configuration). The main content area displays deployment details: Resource group (patsam-rg), Status (Running), Location (Central US), Subscription (Visual Studio Enterprise – MPN), Subscription ID (3822b9c0-bf7e-455e-a1d1-8717b1b4150b), and Tags. It also shows publishing information: URL (<https://patsam.azurewebsites.net>), App Service Plan (patsam-sp (F1: Free)), FTP/deployment username (patsam\deploy-patrick-sameera), FTP hostname (<ftp://waws-prod-dm1-155.ftp.azurewebsites.windows.net>), and HTTPS hostname (<https://waws-prod-dm1-155.ftp.azurewebsites.windows.net>). A purple bar at the top right says 'Click here to access Application Insights for monitoring and profiling for your ASP.NET Core app. →'. Below the main content are three cards: 'Diagnose and solve problems', 'Application Insights', and 'App Service Advisor'.

Once the download completes open the file.

- Endpoint URL: <ftp://waws-prod-dm1-155.ftp.azurewebsites.windows.net/site/wwwroot>
- Username: patsam\\$patsam
- Password: ***

The screenshot shows a Notepad++ window titled 'C:\Users\psam0403\Downloads\patsam.PublishSettings - Notepad++ [Administrator]'. The file contains XML code for a publish profile:

```
<publishProfile profileName="patsam - Web Deploy" publishMethod="MSDeploy" publishUrl="patsam.scm.azurewebsites.net:443" msdeploySite="patsam" userName="$patsam" userPWD="jhRcwC9eGXhQ26RtJpg2Ycd29E4mR6zHJ9eqRg4E689anQcpbQsvQjAhk" destinationAppUrl="http://patsam.azurewebsites.net" SQLServerDBConnectionString="" mySQLDBConnectionString="" hostingProviderForForumLink="" controlPanelLink="http://windows.azure.com" webSystem="WebSites"><database></database></publishProfile><publishProfile profileName="patsam - FTP" publishMethod="FTP" publishUrl="ftp://waws-prod-dm1-155.ftp.azurewebsites.windows.net/site/wwwroot" ftpPassiveMode="True" userName="patsam\$patsam" userPWD="jhRcwC9eGXhQ26RtJpg2Ycd29E4mR6zHJ9eqRg4E689anQcpbQsvQjAhk" destinationAppUrl="http://patsam.azurewebsites.net" SQLServerDBConnectionString="" mySQLDBConnectionString="" hostingProviderForForumLink="" controlPanelLink="http://windows.azure.com" webSystem="WebSites"><database></database></publishProfile><publishProfile profileName="patsam - ReadOnly - FTP" publishMethod="FTP" publishUrl="ftp://waws-prod-dm1-155.ftp.azurewebsites.windows.net/site/wwwroot" ftpPassiveMode="True" userName="patsam\$patsam" userPWD="jhRcwC9eGXhQ26RtJpg2Ycd29E4mR6zHJ9eqRg4E689anQcpbQsvQjAhk" destinationAppUrl="http://patsam.azurewebsites.net" SQLServerDBConnectionString="" mySQLDBConnectionString="" hostingProviderForForumLink="" controlPanelLink="http://windows.azure.com" webSystem="WebSites"><database></database></publishProfile></publishData>
```

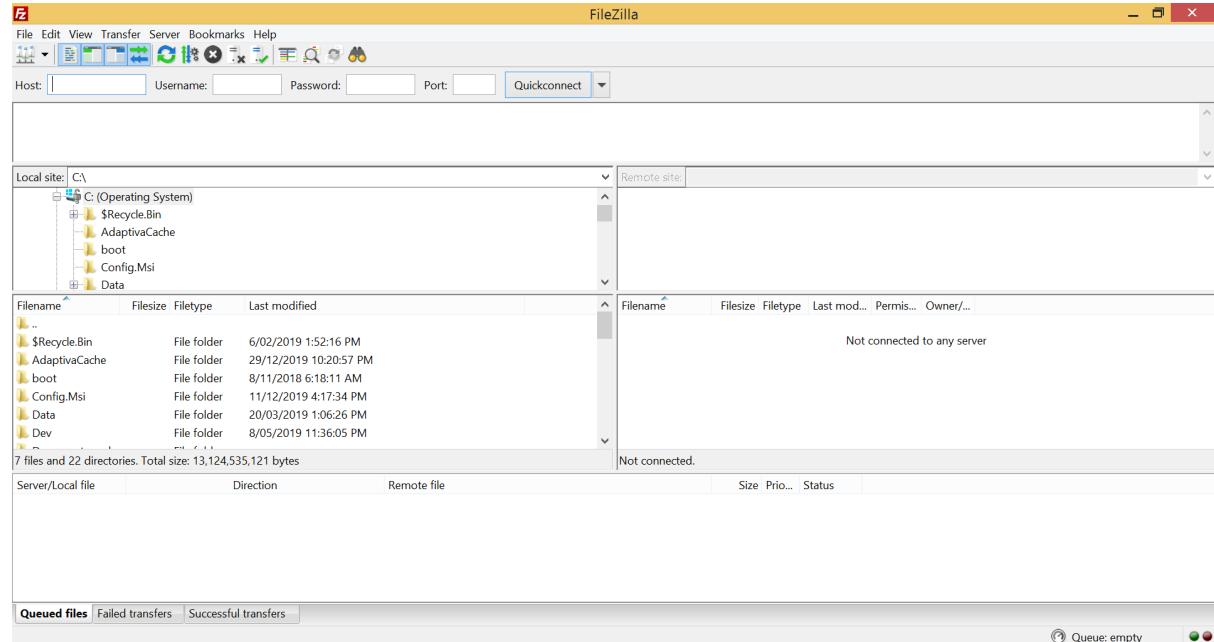
Mind you these credentials are app level credentials. Which means these credentials can be used only for this application.

Next, we need to use an FTP Client to connect to Azure FTP location. I'll be using FileZilla as my FTP Client.

You can download FileZilla here:

<https://filezilla-project.org/download.php?type=server>

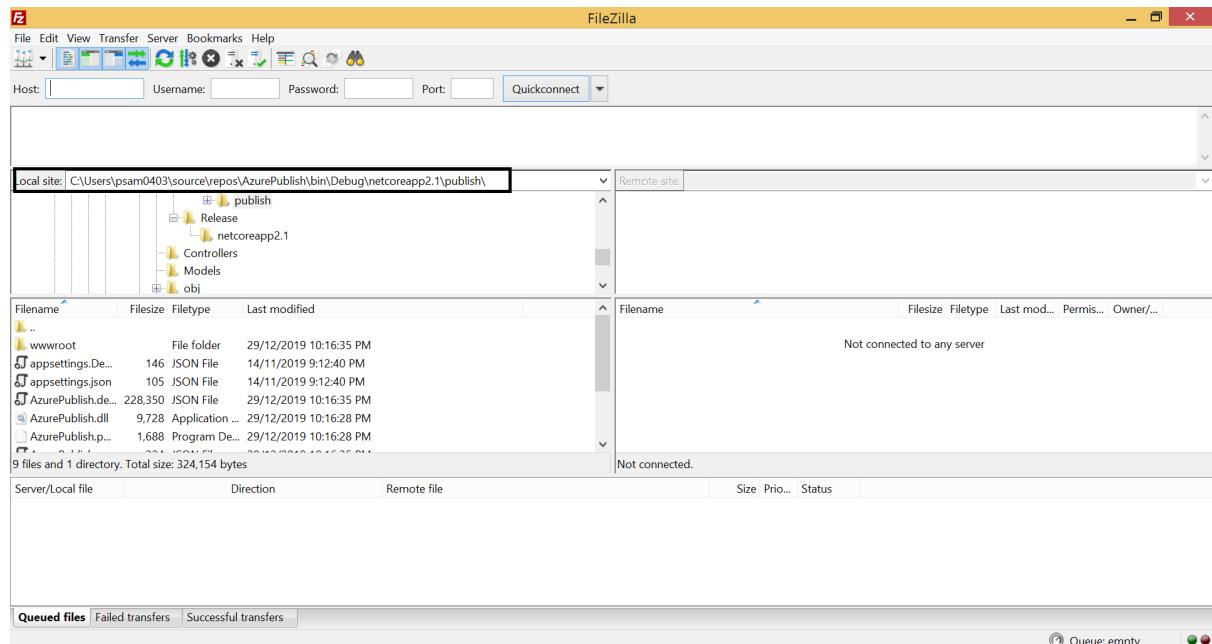
Open FileZilla.



Input Local Site and Remote Site details.

Local Site:

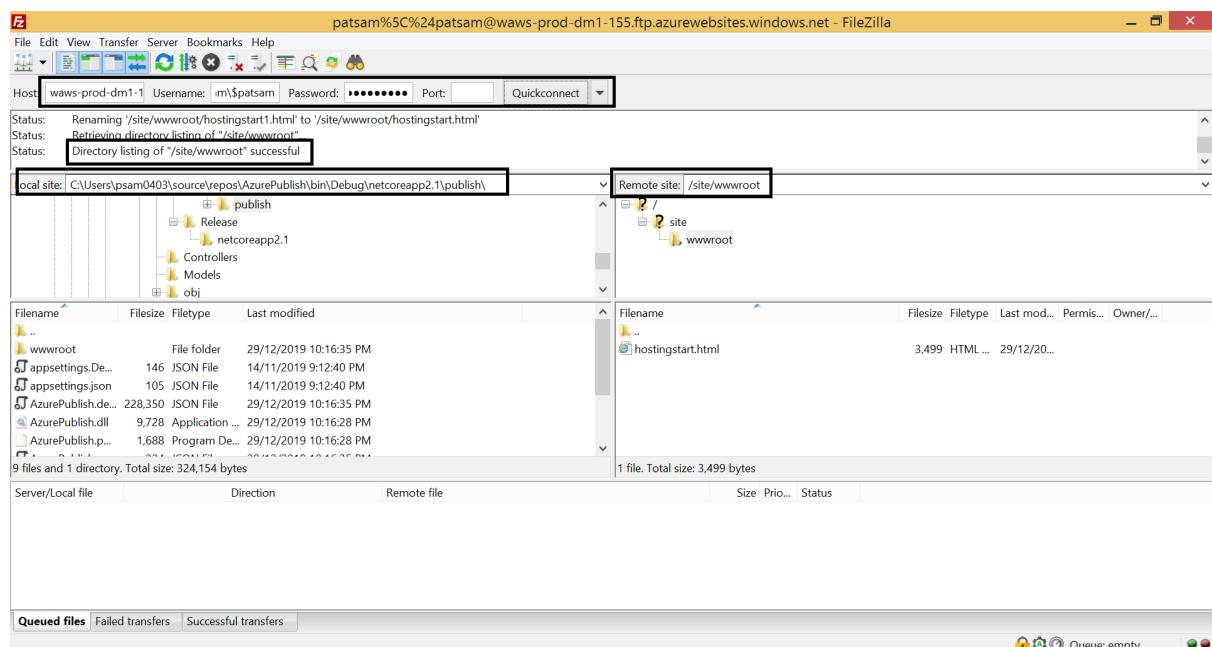
- C:\Users\psam0403\source\repos\AzurePublish\bin\Debug\netcoreapp2.1\publish



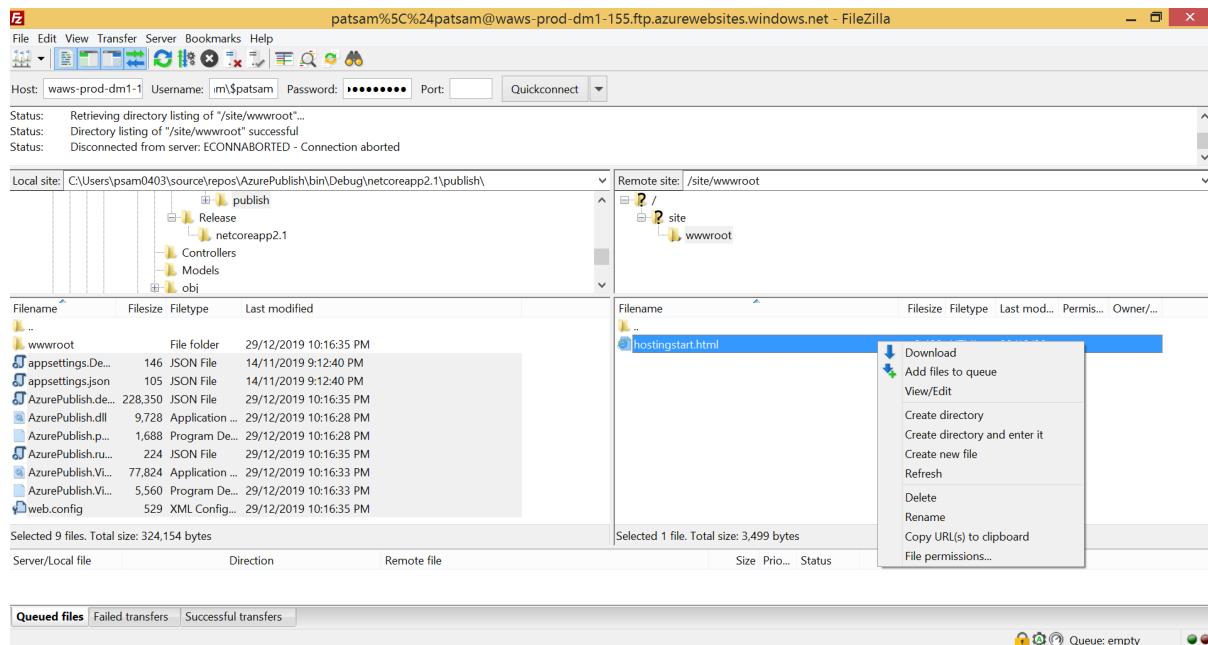
Remote Site:

- Endpoint URL: `ftps://waws-prod-dm1-155.ftp.azurewebsites.windows.net/site/wwwroot`
- Username: patsam\\$patsam
- Password: ***

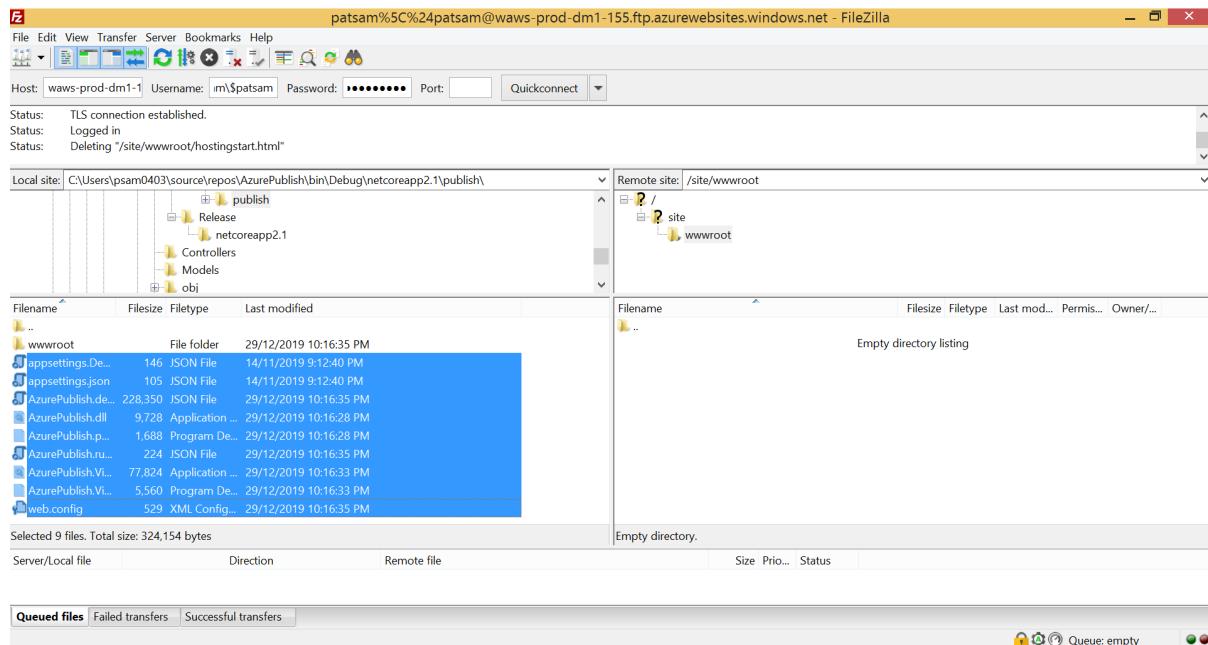
Click QuickConnect.



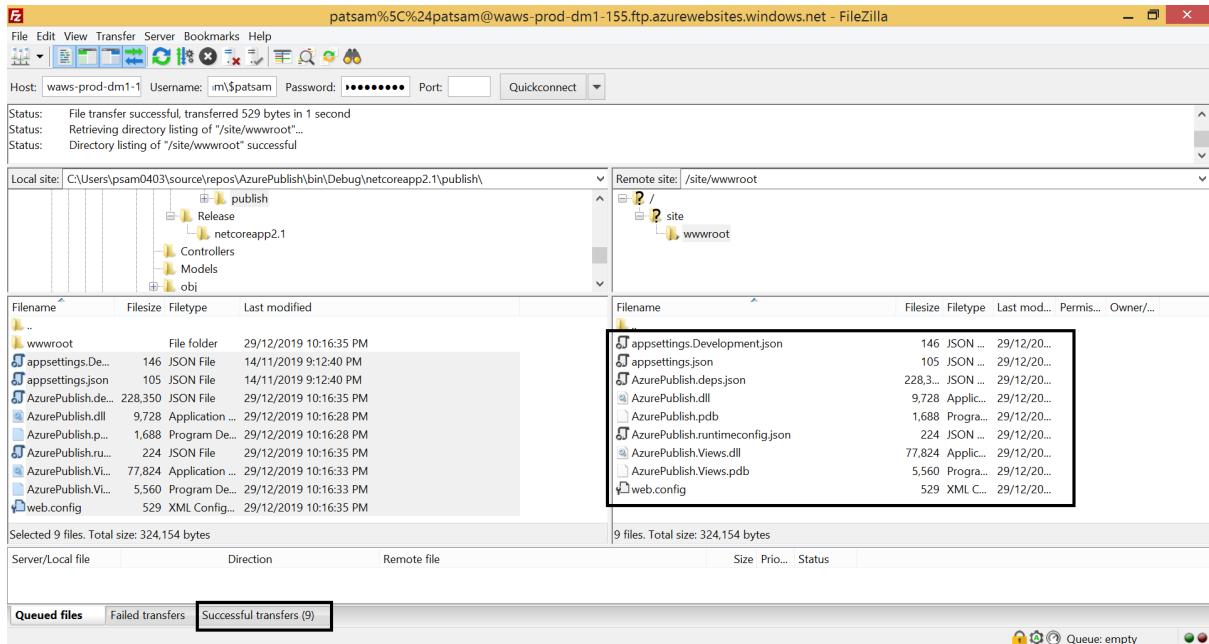
Right click on the hostingstart.html file and delete it.



Next select all the files from the Local Site and drag them to Remote Site.



Make sure file transfer completes successfully.



Browse the URL:

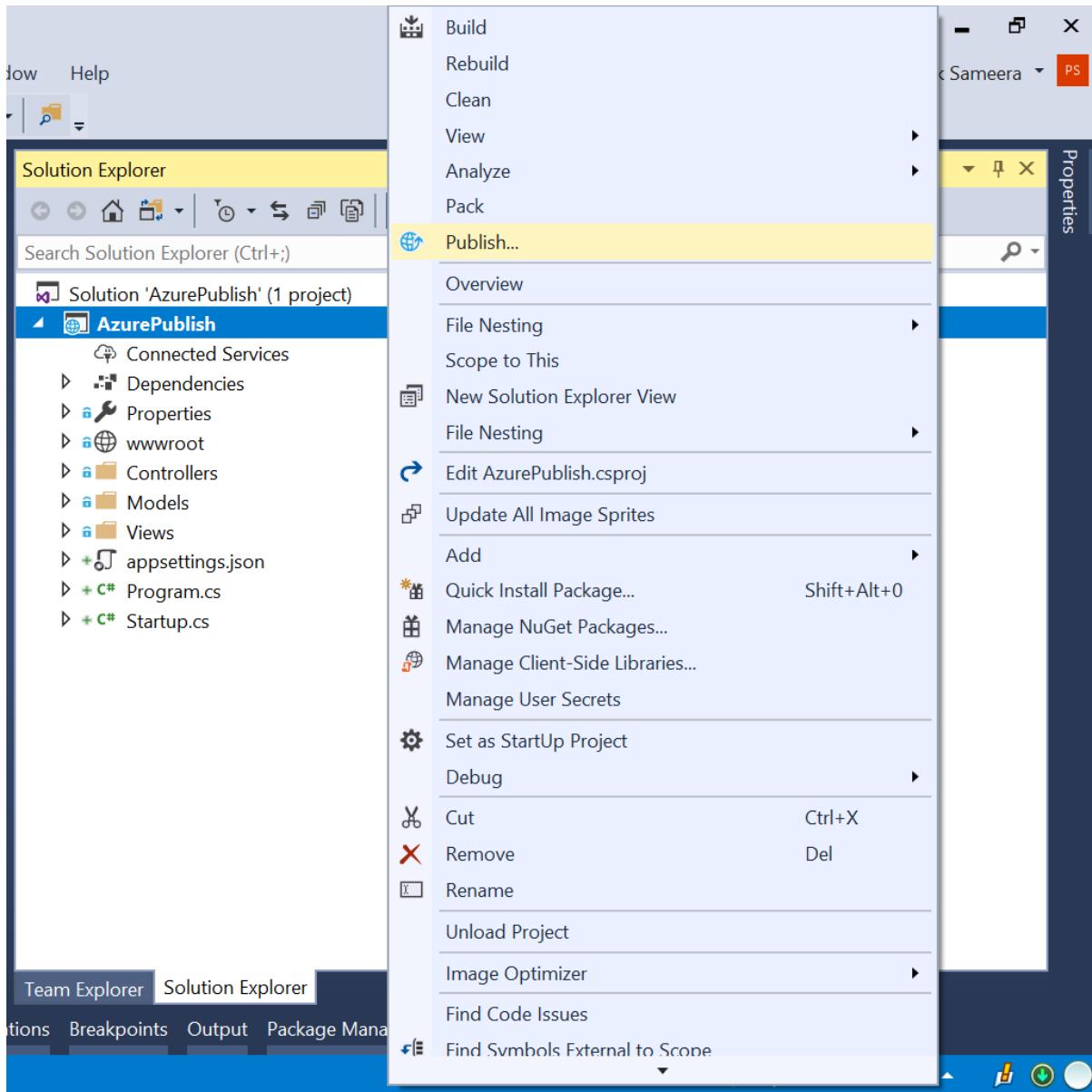
- <https://patsam.azurewebsites.net/>

The screenshot shows a web browser window with the URL 'https://patsam.azurewebsites.net/'. The page title is 'Home Page - AzurePublish'. The page content includes a navigation bar with links for 'Home', 'About', and 'Contact'. Below the navigation bar, there are four main sections: 'Application uses', 'How to', 'Overview', and 'Run & Deploy'. The 'Application uses' section lists: 'Sample pages using ASP.NET Core MVC', 'Theming using Bootstrap'. The 'How to' section lists: 'Add a Controller and View', 'Manage User Secrets using Secret Manager.', 'Use logging to log a message.', 'Add packages using NuGet.', 'Target development, staging or production environment.'. The 'Overview' section lists: 'Conceptual overview of what is ASP.NET Core', 'Fundamentals of ASP.NET Core such as Startup and middleware.', 'Working with Data', 'Security', 'Client side development', 'Develop on different platforms', 'Read more on the documentation site'. The 'Run & Deploy' section lists: 'Run your app', 'Run tools such as EF migrations and more', 'Publish to Microsoft Azure Web Apps'. At the bottom of the page, there is a copyright notice: '© 2019 - AzurePublish'.

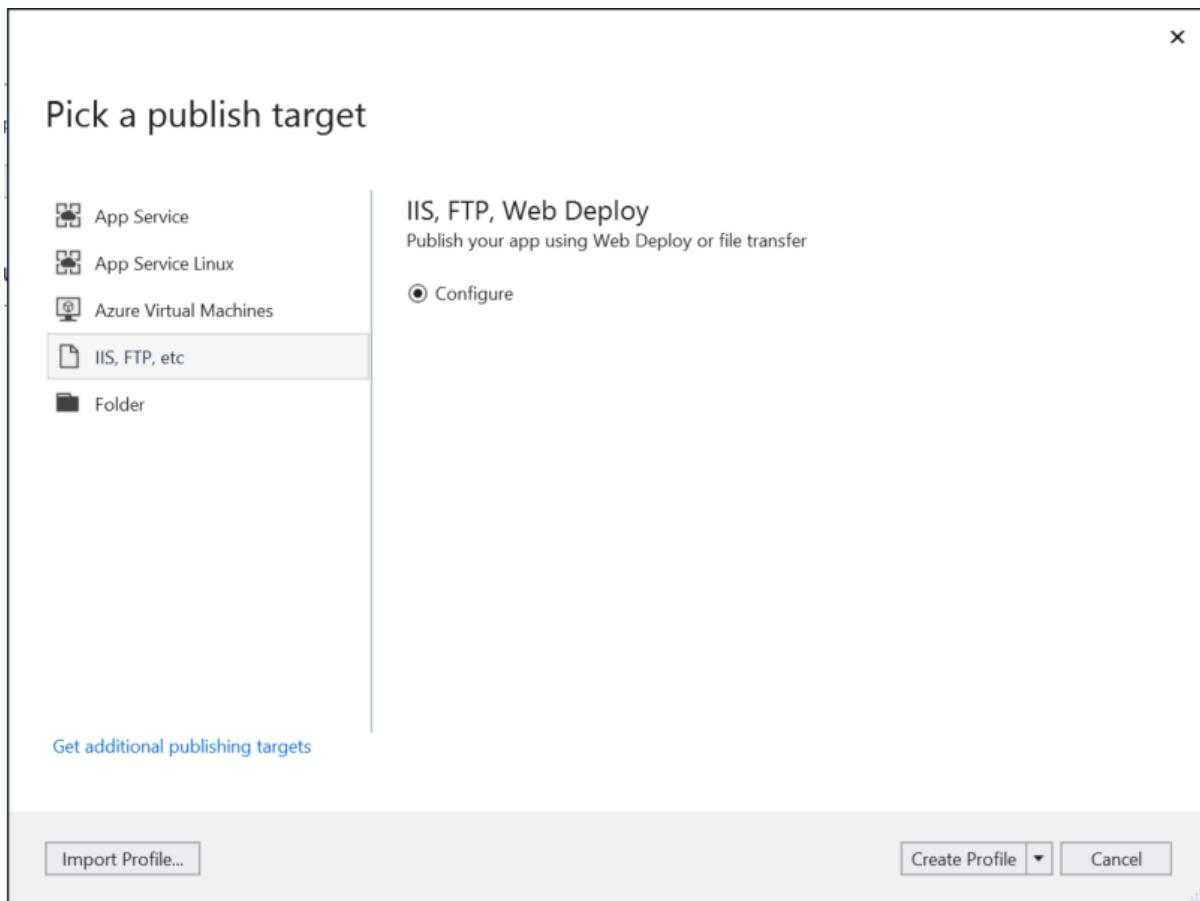
Step 06: Publish the code to Azure through FTP Deployment

Next, we need to Publish the App using FTP Deployment.

Right click on the project and click Publish.

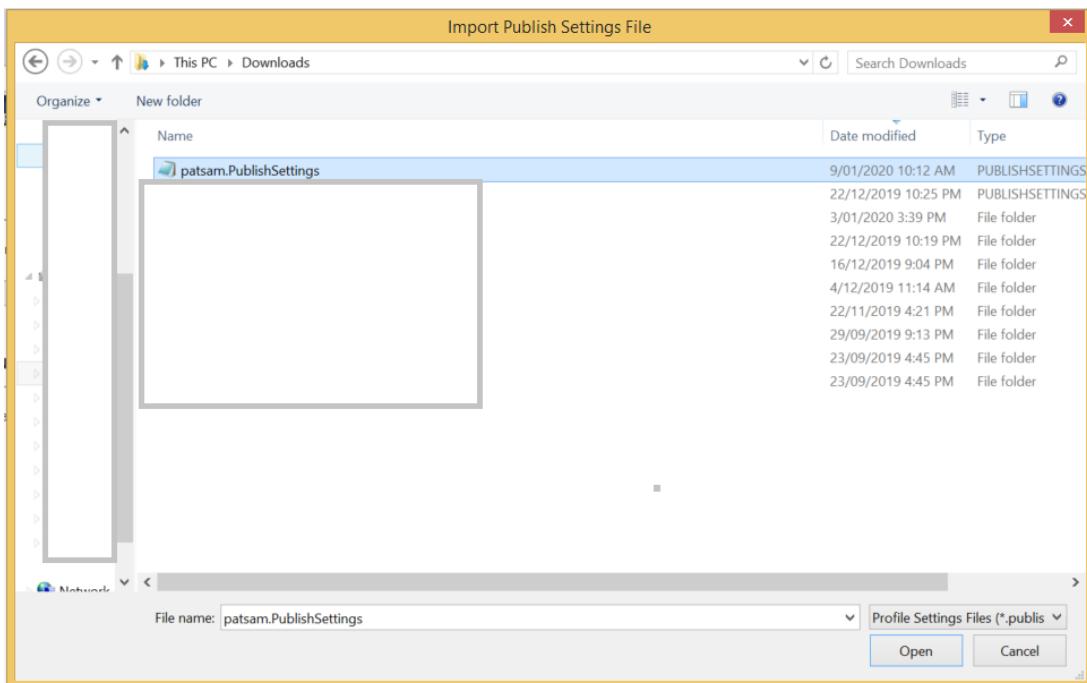


Select IIS, FTP, etc option.



You can click Create Profile button and manually input the FTP details. Or you click on Import Profile button to import the publish profile we downloaded earlier.

We are going to click Import Profile and select the publish profile file.



Then select FTP Publish option.

The screenshot shows the Azure portal's Publish blade. On the left, there's a vertical navigation bar with 'Overview', 'Connected Services', and a blue 'Publish' button. The main area is titled 'Publish' and contains the sub-header 'Publish your app to Azure or another host. [Learn more](#)'. Below this is a dropdown menu set to 'patsam - FTP'. To the right of the dropdown are 'Publish' and 'Actions ▾' buttons. Underneath, there are three rows of publishing details: 'Site URL' (http://patsam.azurewebsites...), 'Configuration' (Release), and 'Troubleshooting Info' (with a 'See Guide' link). At the bottom of the blade, there's a section titled 'Continuous Delivery' with a 'Configure' link.

Click Publish.

Make sure it gets published correctly.

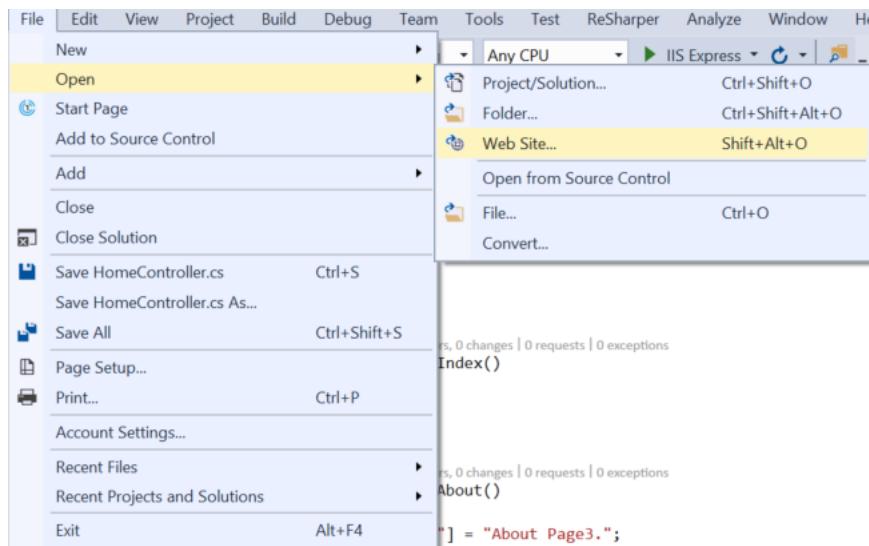
The screenshot shows the Visual Studio Output window with the title 'Output' and a tab for 'Build'. The window displays a log of the publishing process:

```
Output
Show output from: Build
Build started: Project: WebApplication3, Configuration: Release Any CPU -----
1>-->---- Build started: Project: WebApplication3, Configuration: Release Any CPU -----
1>WebApplication3 -> C:\Users\psam403\source\repos\WebApplication3\bin\Release\netcoreapp2.1\WebApplication3.dll
1>WebApplication3 -> C:\Users\psam403\source\repos\WebApplication3\bin\Release\netcoreapp2.1\WebApplication3.Views.dll
2>-->---- Publish started: Project: WebApplication3, Configuration: Release Any CPU -----
Connecting to http://www-prod-dm1-155.ftp.azurewebsites.windows.net/site/wwwroot.
WebApplication3 -> C:\Users\psam403\source\repos\WebApplication3\bin\Release\netcoreapp2.1\WebApplication3.dll
WebApplication3 -> C:\Users\psam403\source\repos\WebApplication3\bin\Release\netcoreapp2.1\WebApplication3.Views.dll
WebApplication3 -> C:\Users\psam403\source\repos\WebApplication3\obj\Release\netcoreapp2.1\PubTmp\Out\
Publishing folder /...
Publishing folder wwwroot...
Publishing folder wwwroot/css...
Publishing folder wwwroot/images...
Publishing folder wwwroot/js...
Publishing folder wwwroot/lib...
Publishing folder wwwroot/lib/bootstrap...
Publishing folder wwwroot/lib/bootstrap/dist...
Publishing folder wwwroot/lib/bootstrap/dist/css...
Publishing folder wwwroot/lib/bootstrap/dist/fonts...
Publishing folder wwwroot/lib/bootstrap/dist/js...
Publishing folder wwwroot/lib/jquery...
Publishing folder wwwroot/lib/jquery/dist...
Publishing folder wwwroot/lib/jquery-validation...
Publishing folder wwwroot/lib/jquery-validation-unobtrusive...
Web App was published successfully http://www-prod-dm1-155.ftp.azurewebsites.windows.net/site/wwwroot
Web App was published successfully http://patsam.azurewebsites.net/
===== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped ======
===== Publish: 1 succeeded, 0 failed, 0 skipped ======
```

Step 07: Use FTP to modify the content

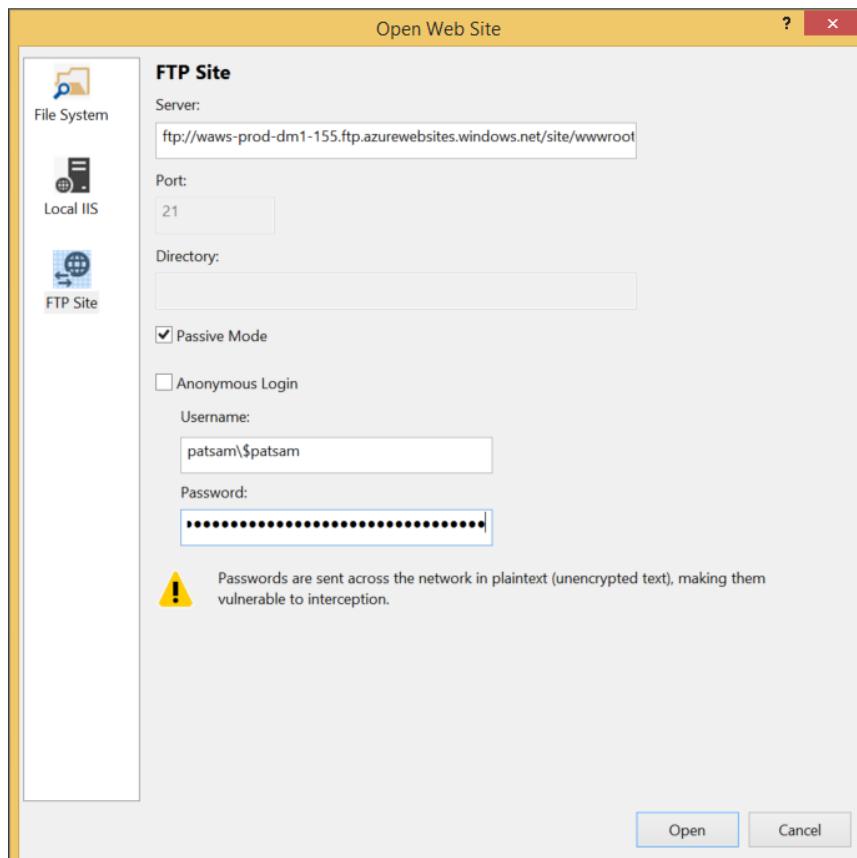
We are going to open the Web Site through FTP.

On Visual Studio click Open → Web Site



Select FTP Site option and input FTP details and click Open.

- Endpoint URL: `ftp://waws-prod-dm1-155.ftp.azurewebsites.windows.net/site/wwwroot`
- Username: `patsam\$patsam`
- Password: `***`



Once connected, it will show the Web Site content on the Solution Explorer window.



Now you can edit the content inside FTP.

The screenshot shows the Microsoft Visual Studio interface. On the left, the code editor displays the `hostingstart.html` file, which contains the static content for an Azure App Service. The code includes logic to handle different browser types and replace specific URLs. On the right, the Solution Explorer pane shows the project structure for "wwwroot" and "WebApplication4". The bottom navigation bar includes links like Stack Trace Explorer, Error List, NuGet browser, Find Results, Web Publish Activity, Data Tools Operations, Breakpoints, Output, and Package Manager Console.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8" name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <title>Microsoft Azure App Service - Welcome</title>
    <link href="https://azuredotnetstorage.blob.core.windows.net/images/app-service/v3/icon.ico" type="image/x-icon" />
    <link rel="stylesheet" href="https://azuredotnetstorage.blob.core.windows.net/css/bootstrap/4.1.1/css/bootstrap_min.css" crossorigin="anonymous">
    <link rel="stylesheet" type="text/css" href="https://appservice.azureedge.net/css/app-service/v3/main.css" />
    <script src="https://appservice.azureedge.net/script/app-service/v3/loc_min.js" crossorigin="anonymous"></script>
    <script type="text/javascript">window.onload = function() {
        try {
            var a = window.location.hostname;
            if (a.includes('.azuresites.net')) {
                a = a.replace('.azuresites.net', '');
            }
            var b = document.getElementById("appCenterLink");
            b.setAttribute("href", b.getAttribute("href") + "&siteName=" + a);
            document.cookie = `siteName=${a};path=/`;
        } catch (d) {
        }
    }</script>
</head>
<body>
    <nav class="navbar navbar-light bg-light">
        <div class="container pl-4 ml-3">
            
        </div></div></nav>
        <div class="container-fluid container-height mt-2" style="height: 20px; width: 100%; clear: both;"></div>
        <div class="row pb-10 mt-10 mb-10 d-xs-none d-sm-none d-md-none d-lg-block d-xl-block" style="height: 20px; width: 100%; clear: both;"></div>
        <div class="row">
            <div class="col-xxl-12 col-sm-12 d-block d-lg-none d-xl-none d-md-block d-xs-block">
                <div class="text-center" style="font-size: 1.5em; color: #0072bc; margin-bottom: 10px;">Hey, App Service developer!
```