



Azure Web Apps - Deployment Slots

Deployment slots in **Azure Web Apps** allow you to host multiple versions of your app within the same **App Service Plan**. These slots enable **staged deployments**, allowing you to test changes before pushing them to production.

Key Features of Deployment Slots

1. **Multiple Environments:** You can create slots for **staging, testing, or pre-production** while keeping your production environment stable.
2. **Slot Swapping:** You can swap slots to push a tested version into production with **zero downtime**.
3. **Sticky Settings:** Certain settings (like connection strings) can be **slot-specific**, ensuring proper configuration for each environment.
4. **Rollback Capability:** If something goes wrong after deployment, you can swap back to the previous stable slot.
5. **Traffic Routing:** You can direct a percentage of traffic to a specific slot for **gradual rollouts and A/B testing**.

Default Slots

- **Production Slot (Default):** The live version of your app.
- **Staging Slot (Optional):** A pre-production slot for testing changes before swapping to production.

Example Use Case

1. You deploy your app to the **staging slot**.
2. You test it thoroughly.
3. If everything works fine, you **swap the staging slot with production**.
4. If issues arise, you **swap back** quickly without downtime.

The process starts by creating an Azure Web App in the North Europe region with .NET 6 and the Standard S1 plan. After setting up the web app, a .NET 6 ASP.NET project is created in Visual Studio 2022, modified, and published to Azure. To test a new version before deployment, a staging deployment slot is created. The updated version is published to the staging slot, ensuring it works correctly. Finally, the swap feature exchanges the staging and production slots, promoting the tested version to production with zero downtime. The goal is safe, controlled, and seamless deployment.

The end goal is to enable seamless, zero-downtime deployments by using Azure deployment slots. This ensures that new application versions are thoroughly tested in a staging environment before being promoted to production. If any issues arise, you can quickly roll back by swapping slots. This approach improves application reliability,

reduces downtime, and enhances the development workflow, making updates smoother and safer for end users.

😊 To begin with the Lab:

1. First, we start by creating a web app in our Azure Portal. So, choose your resource group and give a name to your web app instance, and choose dot net 6 for the runtime stack in the North Europe region.

Subscription * ⓘ

MSDN Platforms Subscription

Resource Group * ⓘ

NewRG

Create new

Instance Details

Name

newapp2w ✓

-b5dxbcgghmdafmgv.northeurope-01.azurewebsites.net

Secure unique default hostname on. [More about this update ↗](#)

Code Container

Publish *

Runtime stack *

.NET 6 (LTS) ✓

Operating System *

Linux Windows

Region *

North Europe ✓

i Not finding your App Service Plan? Try a different region or select your App Service Environment.

2. For the pricing plan choose Standard S1 and move to the review page to create your web app.

Pricing plans

App Service plan pricing tier determines the location, features, cost and compute resources associated with your app.
[Learn more ↗](#)

Windows Plan (North Europe) * ⓘ

(New) ASP-NewRG-8bb0 ✓

Create new

Pricing plan

Standard S1 (100 total ACU, 1.75 GB memory, 1 vCPU) ✓

[Explore pricing plans](#)

Zone redundancy

An App Service plan can be deployed as a zone redundant service in the regions that support it. This is a deployment time only decision. You can't make an App Service plan zone redundant after it has been deployed [Learn more ↗](#)

- Zone redundancy
- Enabled:** Your App Service plan and the apps in it will be zone redundant. The minimum App Service plan instance count will be three.
- Disabled:** Your App Service Plan and the apps in it will not be zone redundant. The minimum App Service plan instance count will be one.

3. Below you can see that our web app has been created and it is up and running.

The screenshot shows the Microsoft Azure portal's deployment overview for a web app named "Microsoft.Web-WebApp-Portal-a71910e0-90ed". The status is "Your deployment is complete". Deployment details include a name, subscription, and resource group. Next steps provide links to manage deployments, protect with authentication, and add a deployment slot. A "Go to resource" button is at the bottom, and the URL "newapp2w-b5dxbcgghmdafmgv.northeurope-01.azurewebsites.net" is shown in the address bar.



Your web app is running and waiting for your content

Your web app is live, but we don't have your content yet. If you've already deployed, it could take up to 5 minutes for your content to show up, so come back soon.



 Supporting Node.js, Java, .NET and more

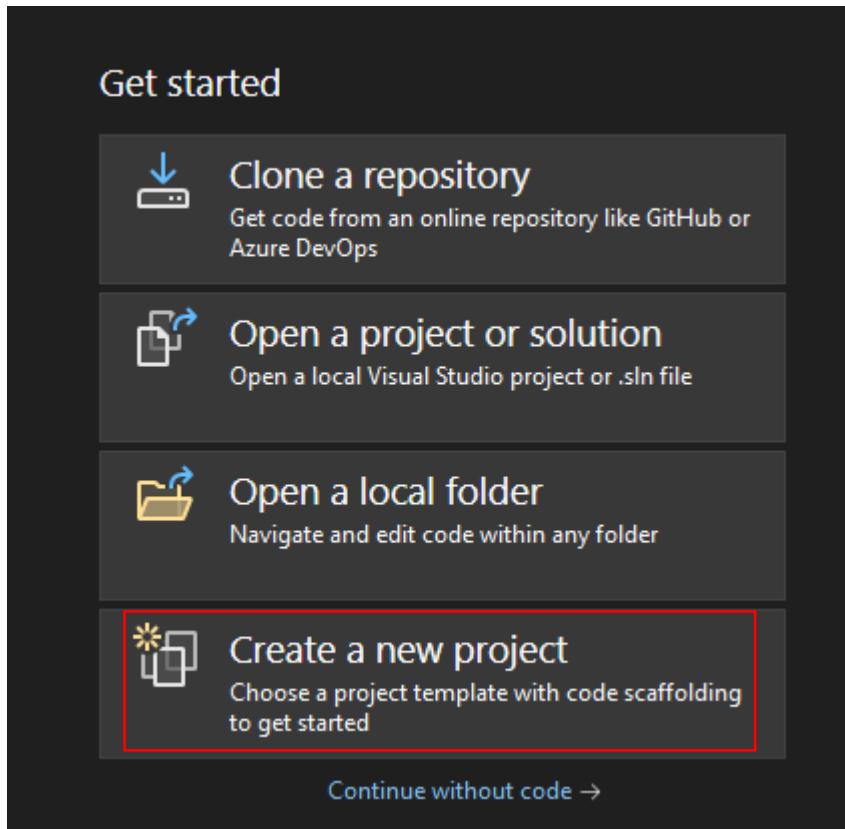
Haven't deployed yet?
Use the deployment center to publish code or set up continuous deployment.

Starting a new web site?
Follow our Quickstart guide to get a web app ready quickly.

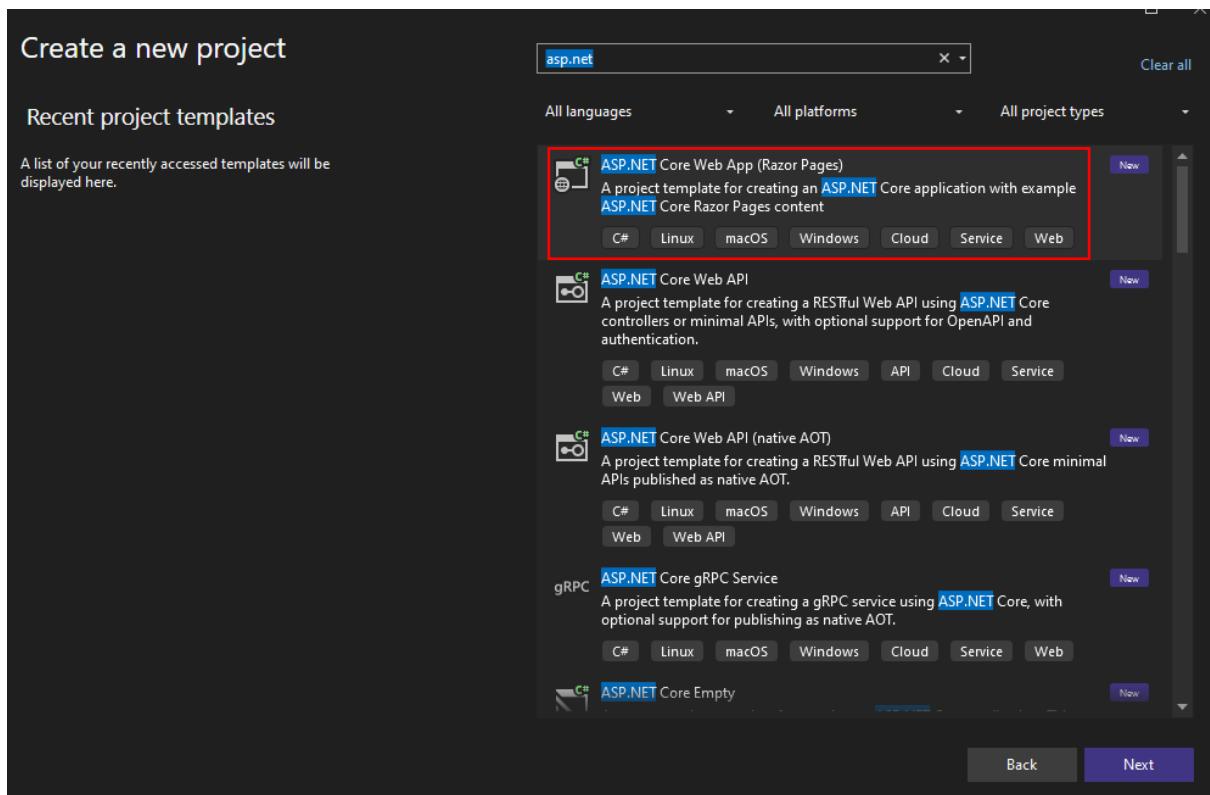
[Deployment center](#)

[Quickstart](#)

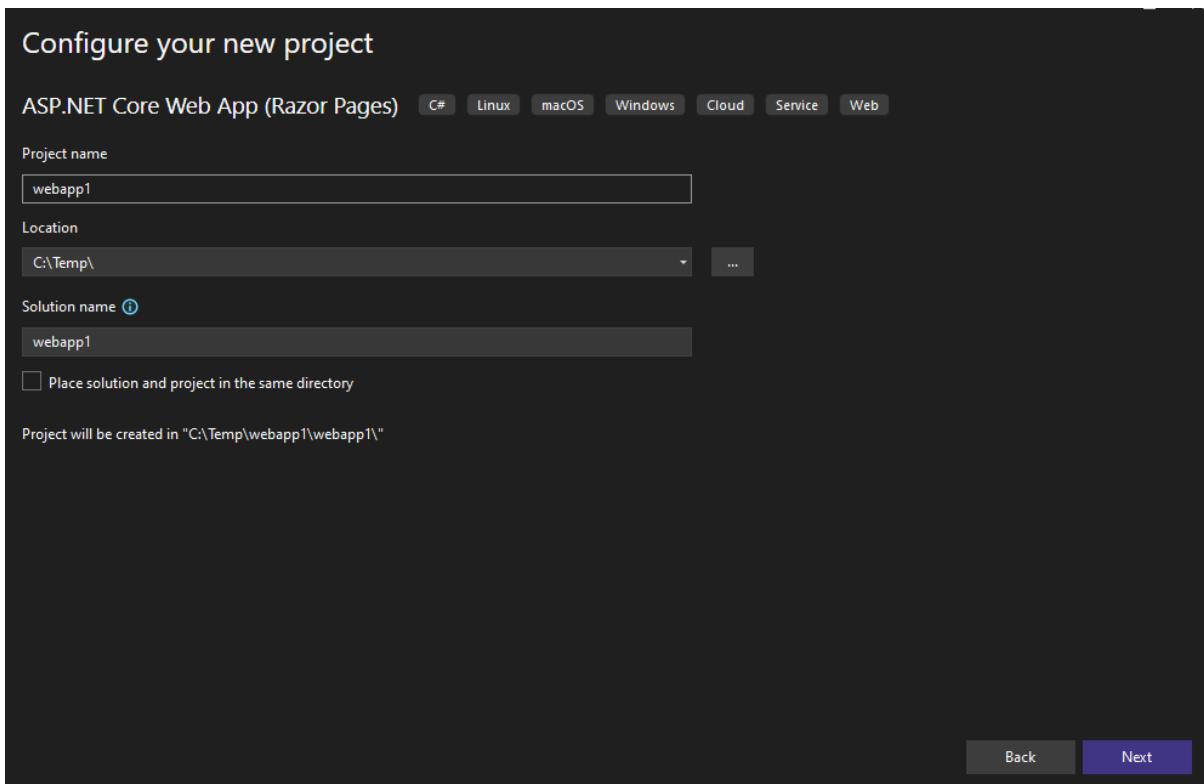
4. Now open Visual Studio 2022 on your laptop and create a project. On the home screen of Visual Studio click on Create a new project.



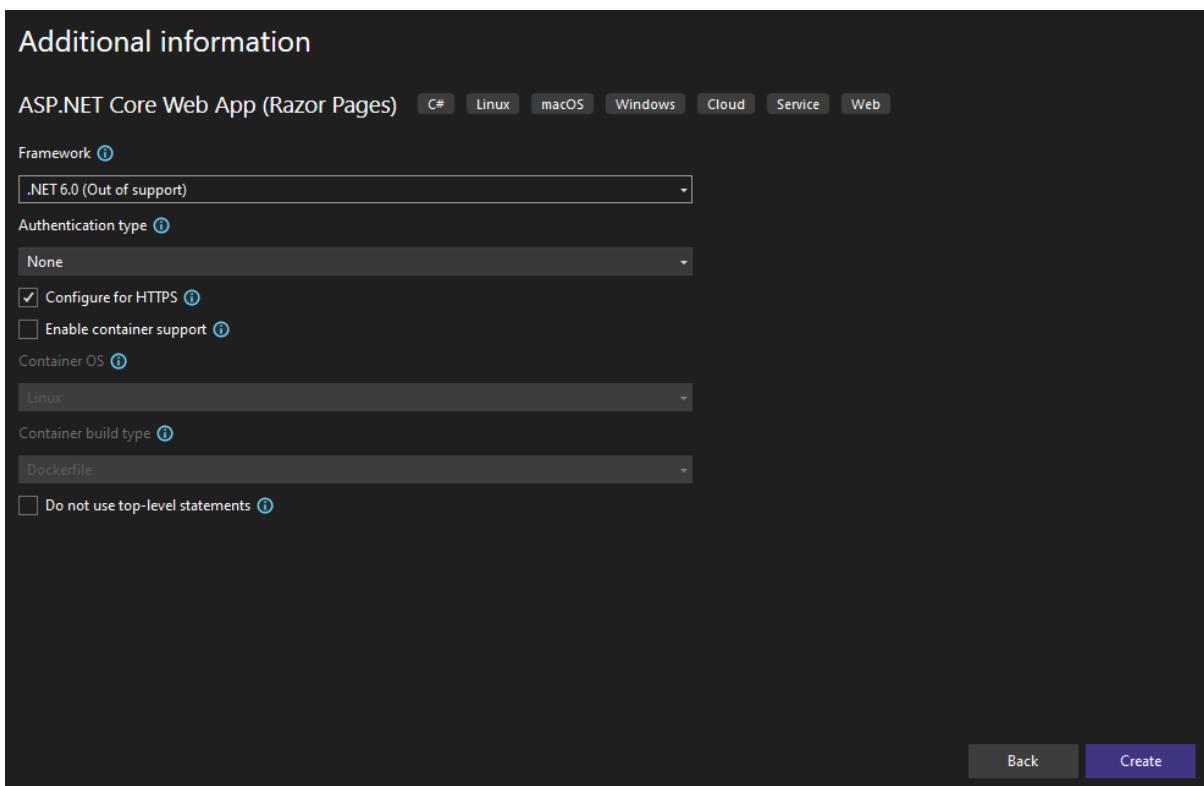
5. Search for asp.net and choose the highlighted web app.



6. Then give a name to your project and click on next.



7. In the end choose the .net 6 framework and create your web app.



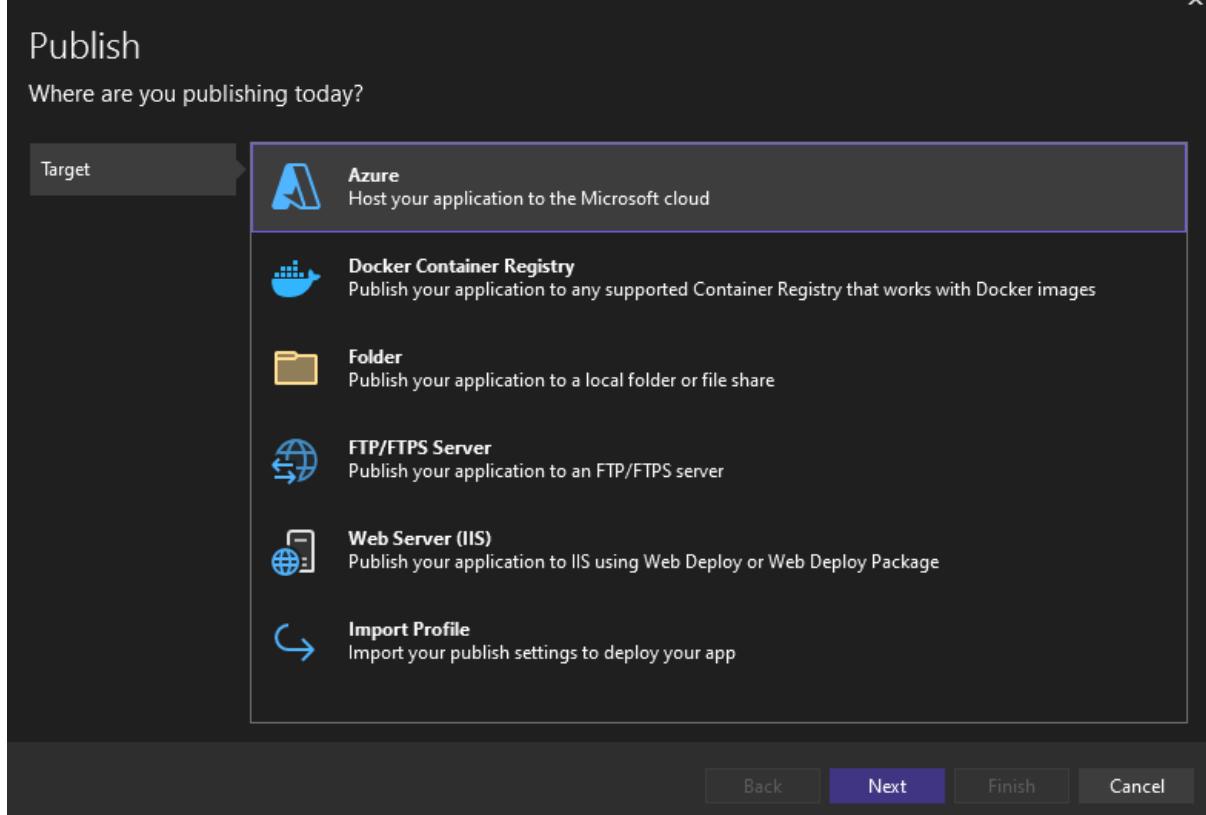
8. Once you are in the application expand pages from the solution explorer and go to index.cshtml then on line number 9 write the same sentence. Save it.

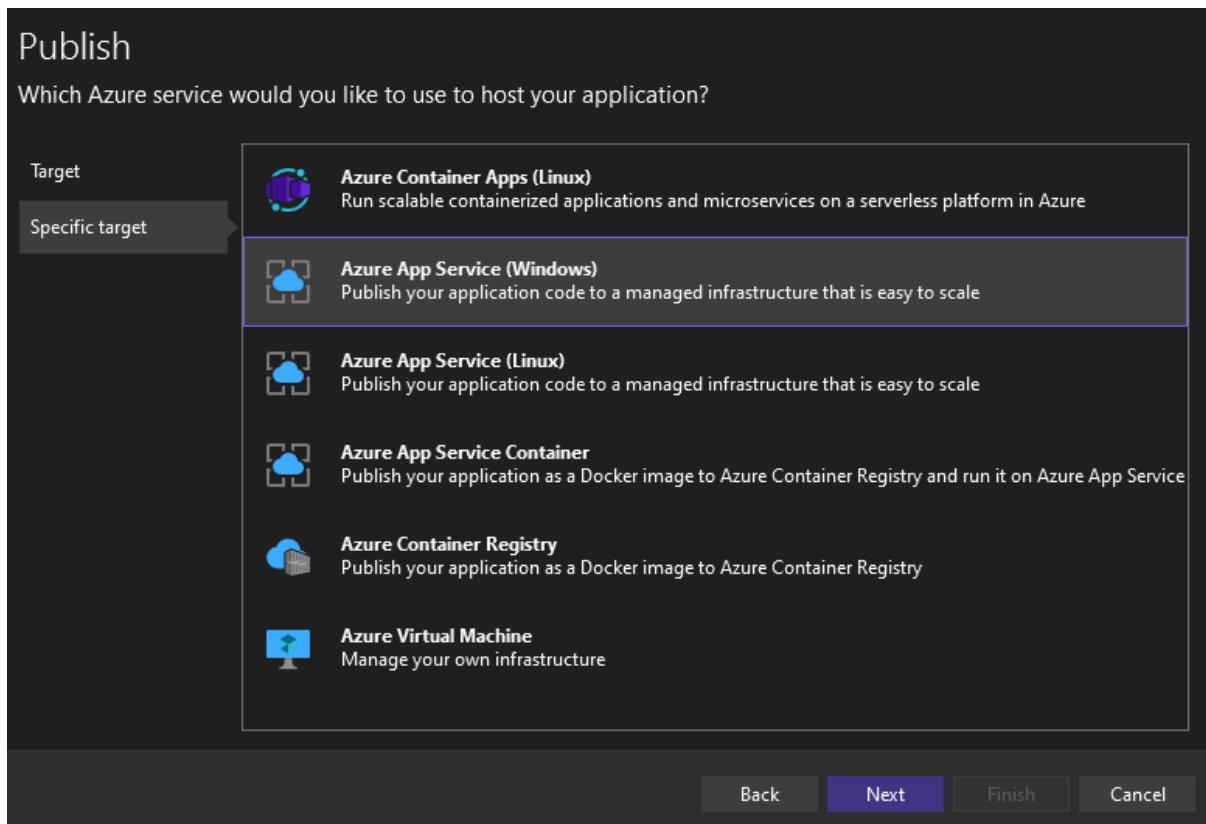
The screenshot shows the Visual Studio interface. On the left, the code editor displays the `Index.cshtml` file with the following content:

```
1  @page
2  @model IndexModel
3  @{
4      ViewData["Title"] = "Home page";
5  }
6
7  <div class="text-center">
8      <h1 class="display-4">Welcome</h1>
9      <p>This is version 1 of the Application</p>
10 </div>
```

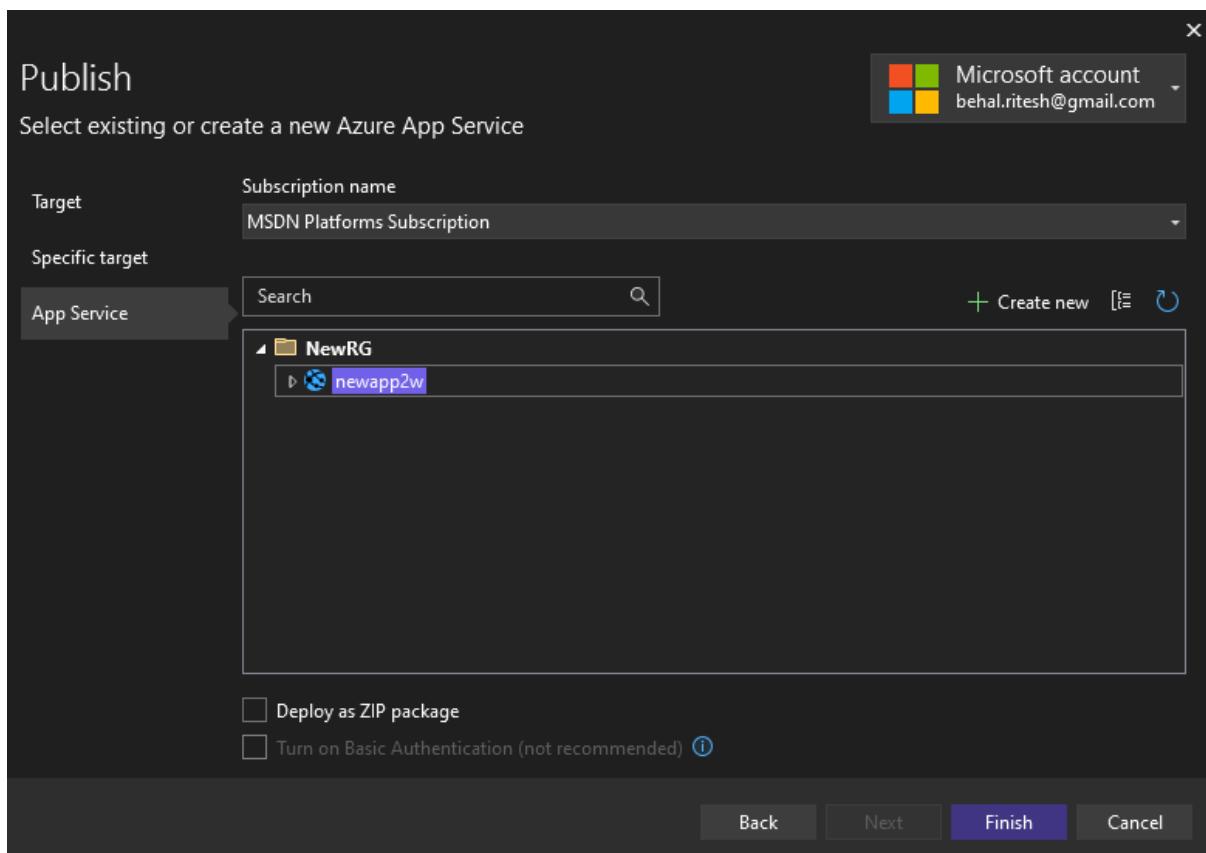
A red box highlights the line `<p>This is version 1 of the Application</p>`. On the right, the Solution Explorer shows a project named `webapp1` containing files like `Index.cshtml`, `ViewImports.cshtml`, `ViewStart.cshtml`, and `Program.cs`.

9. Now we are going to publish this web application from our Visual Studio to Web app on the Azure portal. So, right-click on the web app in the solution explorer and choose to publish.
10. Choose Azure as your target then choose Azure App Services (Windows).





11. After that you will see your resource group and inside of it your web app. Choose it and click on finish.



12. Once your publish profile is created then click on publish button.

The screenshot shows the Azure App Service (Windows) deployment center. At the top, there's a navigation bar with a cloud icon, the name "newapp2w - Web Deploy.pubxml", and a "Publish" button. Below the navigation bar, there are links for "New profile" and "More actions". A message box says "Ready to publish.". The main area is divided into sections: "Settings" (Configuration: Release, Target Framework: net6.0, Deployment Mode: Framework-dependent, Target Runtime: Portable), "Hosting" (Account: behal.ritesh@gmail.com (Microsoft account), Subscription: d6549a66-c45c-4979-840c-3b356da446b0, Resource group: NewRG, Resource name: newapp2w, Site URL: https://newapp2w-b5dxbcgghmdafmvg.northeurope-01.azurewebsites.net), and "Service Dependencies" (which is currently empty). There are also "Add", "Edit", and "Delete" buttons for service dependencies.

13. Wait for some time and you will see your web app in action.

The screenshot shows the "Welcome" page of the Azure web app. The URL in the browser is "https://newapp2w-b5dxbcgghmdafmvg.northeurope-01.azurewebsites.net". The page content includes "Welcome" and "This is version 1 of the Application".

14. If I now had to develop a newer version of the application, so obviously before publishing it or promoting it on the Azure web app, you'd want your developers or your end users to first test out the application itself.
15. So, for the Azure Web app, what you can do is you can work with deployment slots.
16. As you can see in the deployment slots we have no slots but you will see a production web app as well so, we will click on the Add button.

newapp2w | Deployment slots

No slots have been added.

Deployment slots are live apps with their own hostnames. App content and configurations elements can be swapped between two deployment slots, including the production slot. [Learn more](#)

Add slot

17. Give it a name and click on add. We're just adding another place in which we can deploy a newer version of our application.

Add Slot

Name

newapp2w-stagingwebapp-b9fnfnhuhkcxhfax.northeurope-01.azurewebsites.net

Clone settings from:

Do not clone settings

18. Here you can see that our staging web app is now ready. Click on it and you will see this is just another web app. But this is just a slot, an area within your current Azure web application itself. You even get another default domain name that's basically linked onto your Azure web app itself.

newapp2w | Deployment slots

Name	Status	App service plan	Traffic %
newapp2w PRODUCTION	Running	ASP-NewRG-8bb0	100
newapp2w-stagingwebapp	Running	ASP-NewRG-8bb0	0

stagingwebapp (newapp2w/stagingwebapp)

App Service (Slot)

Search

Browse Stop Swap Restart Delete Refresh Download publish profile Reset publish profile Share to mobile Send us your feedback

Overview Essentials JSON View

Activity log Resource group (move) : newrg Status : Running Default domain : newapp2w-stagingwebapp-b9fnfnhuhkcxhfax.northeurope-01.azure App Service Plan : ASP-NewRG-Bbb0 (S1: 1) Tags Location (move) : North Europe Operating System : Windows Diagnose and solve problems Subscription (move) : MSDN Platforms Subscription Health Check : Not Configured Microsoft Defender for Cloud Subscription ID : d6549a66-c45c-4979-840c-3b356da446b0 Recommended services (preview) Tags (edit) : Add tags Deployment Properties Monitoring Logs Capabilities Notifications Recommendations

Web app Name : newapp2w/stagingwebapp Publishing model : Code Deployment Center Deployment logs View logs Last deployment Loading deployments... Deployment provider : None Domains Default domain : newapp2w-stagingwebapp-b9fnfnhuhkcxhfax.northeurope-01.azurewebsites.net Application Insights Name : Enable Application Insights Custom domain : Add custom domain Networking Virtual IP address : 20.50.64.29 Hosting

19. If we click on the default name we can see the default web app in place.

newapp2w-stagingwebapp-b9fnfnhuhkcxhfax.northeurope-01.azurewebsites.net

Microsoft Azure

Your web app is running and waiting for your content

Your web app is live, but we don't have your content yet. If you've already deployed, it could take up to 5 minutes for your content to show up, so come back soon.

</> Supporting Node.js, Java, .NET and more



Haven't deployed yet? Use the deployment center to publish code or set up continuous deployment.

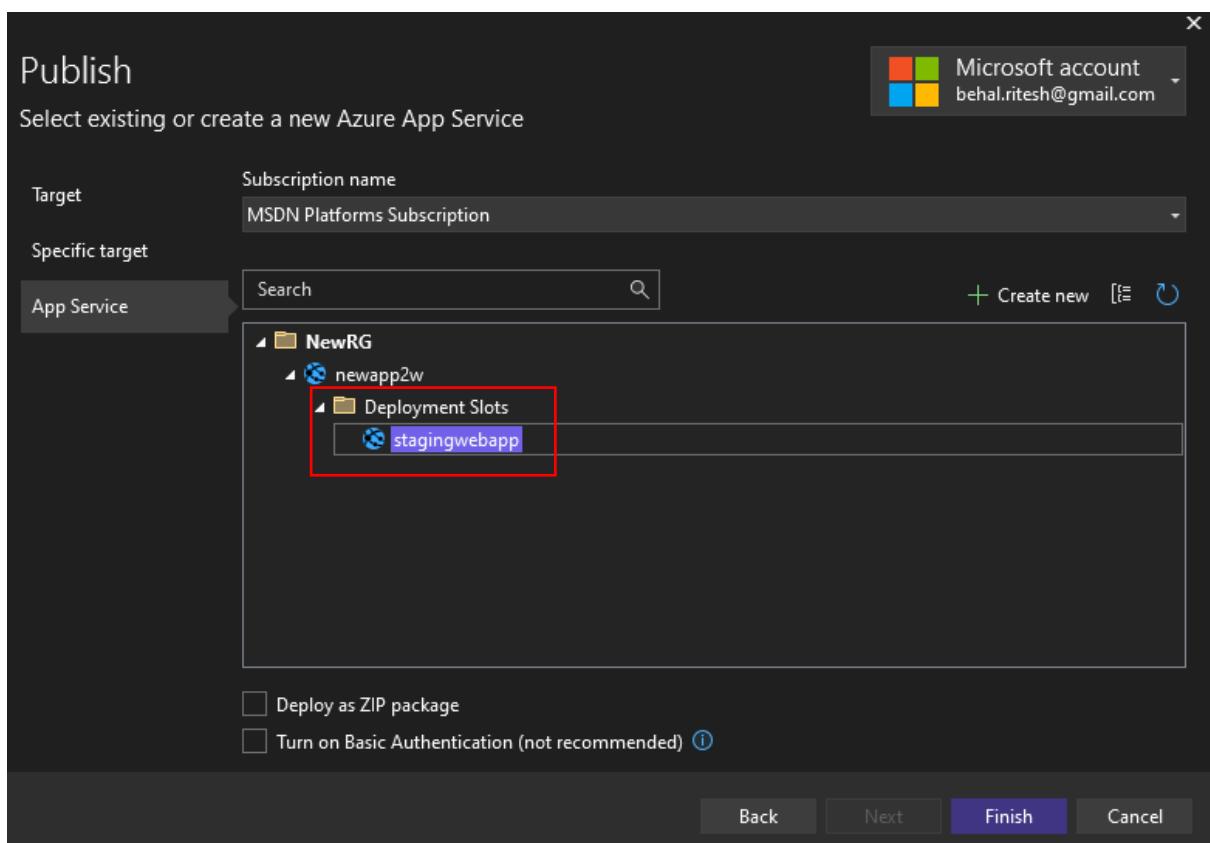
Starting a new web site? Follow our Quickstart guide to get a web app ready quickly.

Deployment center Quickstart

20. Now we went back to the visual studio and changes the version from 1 to 2 and save it.

```
1 @page
2 @model IndexModel
3 @{
4     ViewData["Title"] = "Home page";
5 }
6
7 <div class="text-center">
8     <h1 class="display-4">Welcome</h1>
9     <p>This is version 2 of the Application</p>
10 </div>
11
```

21. From the visual studio we will again publish our web app. But this time we will create a new profile and choose our staging web app from the deployment slots.
22. When the publish profile has been created just click on publish button to publish your web app.



23. Once your web app is published you will see the version 2 of it.



24. So, I have this running in my staging slot and I have my version one application running in my production slot. So now let's say that testing is complete for the newer version of your application, and you want to now replace that version with your production slot. You can go onto the deployment slots of your Azure web app and click on the swap button. You can swap your staging with your production slot.

This screenshot shows the deployment slots configuration for the 'newapp2w-stagingwebapp' web app. It lists two slots: 'newapp2w' (labeled 'PRODUCTION') and 'newapp2w-stagingwebapp'. Both slots are shown as 'Running'. The 'Traffic %' column indicates that 100% of traffic is currently being directed to the production slot, while 0% is directed to the staging slot. The interface includes standard Azure navigation and management controls at the top.

Name	Status	App service plan	Traffic %
newapp2w PRODUCTION	Running	ASP-NewRG-8bb0	100
newapp2w-stagingwebapp	Running	ASP-NewRG-8bb0	0

● Source
newapp2w-stagingwebapp

● Target **PRODUCTION**
newapp2w

(i) Swap with preview can only be used with sites that have deployment slot settings enabled.

Perform swap with preview

Config Changes

This is a summary of the final set of configuration changes on the source and target deployment slots after the swap has completed.

● Source slot changes

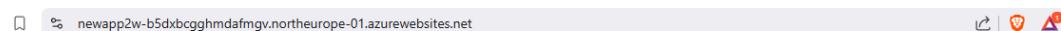
● Target slot changes

Setting	Type	Old Value	New Value
PhpVersion	General	5.6	
WEBSITE_NODE_DEFAULT_...	AppSetting	6.9.1	Not set

Start Swap

Close

25. Once the swapping is completed you will see that your production is now version 2 and staging is version 1 web app.



webapp1 Home Privacy

Welcome

This is version 2 of the Application

C

newapp2w-stagingwebapp-b9fnfnhuhkcxhfax.northeurope-01.azurewebsites.net

webapp1 Home Privacy

Welcome

This is version 1 of the Application