Publishing Application to Azure Web App

Azure Web App is a Platform-as-a-Service (PaaS) offering from Microsoft Azure that allows developers to easily host, build, deploy, and scale web applications and APIs. It abstracts away the complexities of managing the underlying infrastructure, enabling developers to focus entirely on application development and deployment.

Azure Web App is ideal for businesses or developers who want a **reliable, scalable, and secure platform** to host their web applications without the overhead of managing infrastructure. It's a robust solution for modern cloud-native applications.

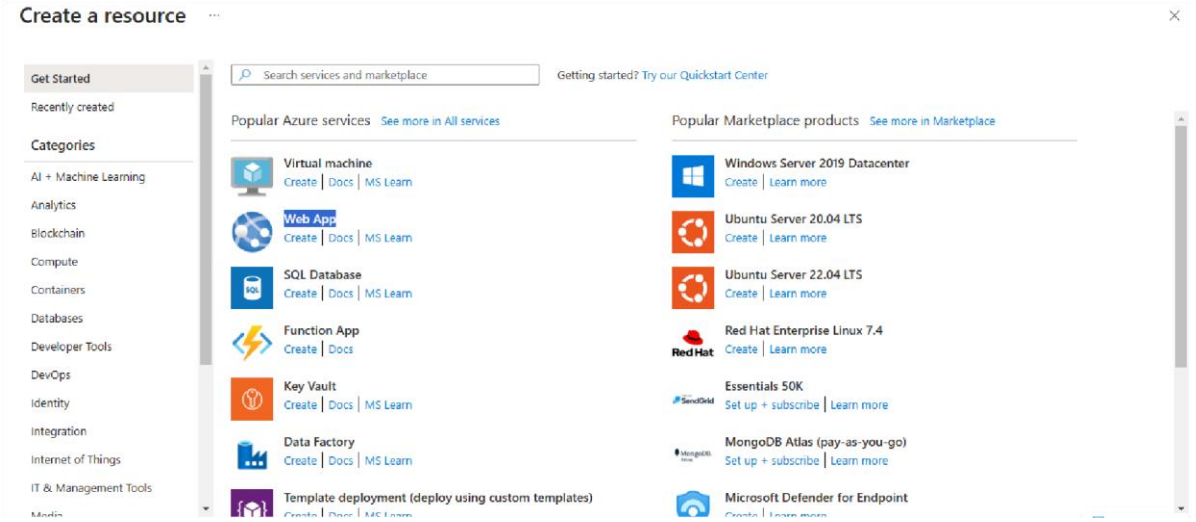
Use Cases of Azure Web App

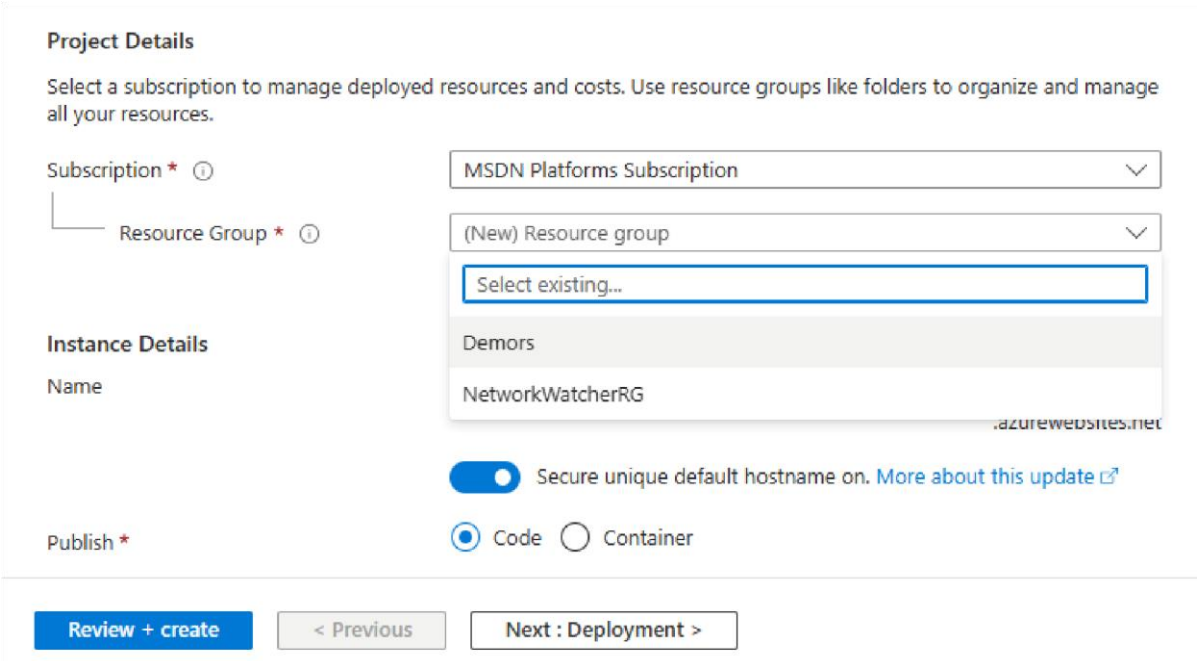
1. Hosting dynamic websites and web applications, such as corporate websites, blogs, or e-commerce platforms.
2. Deploying RESTful APIs for mobile and web applications, or as part of a microservices architecture.
3. Running user-facing dashboards, internal portals, or collaboration platforms.
4. Hosting server-side rendered applications using frameworks like Next.js, Nuxt.js, or ASP.NET.
5. Deploying Software-as-a-Service (SaaS) applications with multi-tenant support.
6. Serving static websites and Jamstack applications with integrations to headless CMS.
7. Supporting continuous deployment for staging, testing, and production environments.
8. Running event-driven applications like real-time chat or notification systems.
9. Deploying multi-region applications for low-latency global access.
10. Managing scalable workloads to handle fluctuating traffic, such as seasonal e-commerce or promotional campaigns.

The end goal of this lab is to successfully publish a .NET application onto an Azure Web App, a platform-as-a-service offering. By creating and configuring a Web App resource in Azure, choosing the appropriate runtime stack (.NET 6), and generating a new publish profile in Visual Studio, the application is seamlessly deployed to the Web App. Once published, the application is accessible via the Web App's default domain, showcasing its functionality. This process eliminates the need to manage underlying infrastructure, providing a scalable, managed hosting solution while demonstrating Azure's ease of use for application deployment.

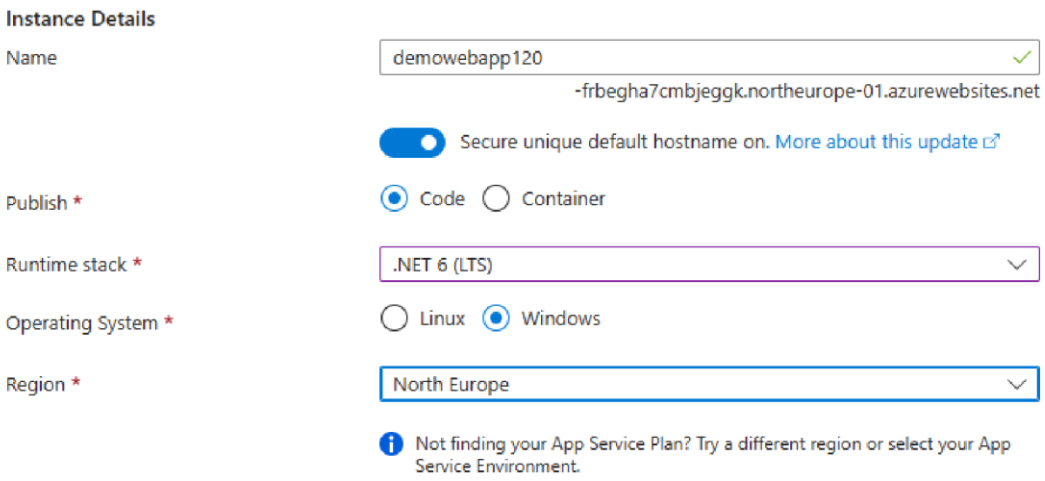
To begin with the lab

1. First, we need to create a webapp. To create one navigate to the Azure portal under "Popular Azure Services" and select the **Web App** option.
2. Select an existing resource group.

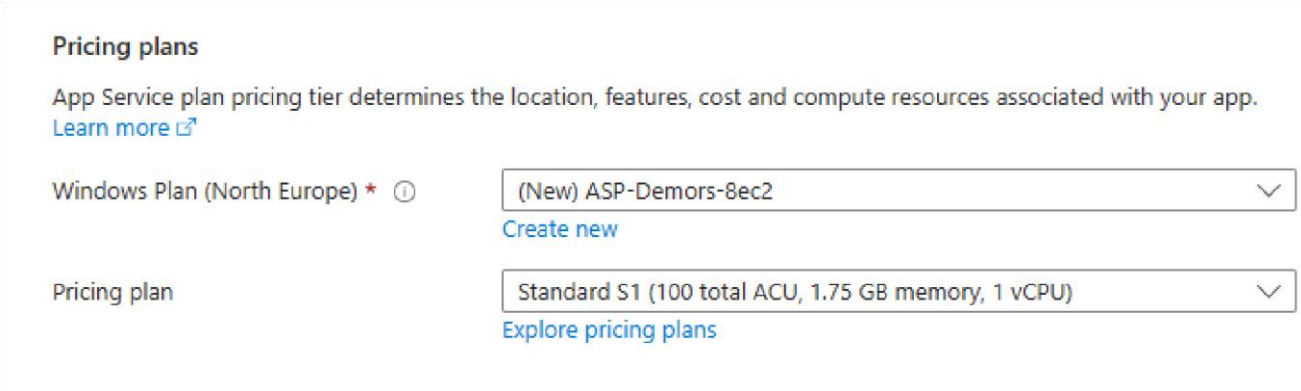




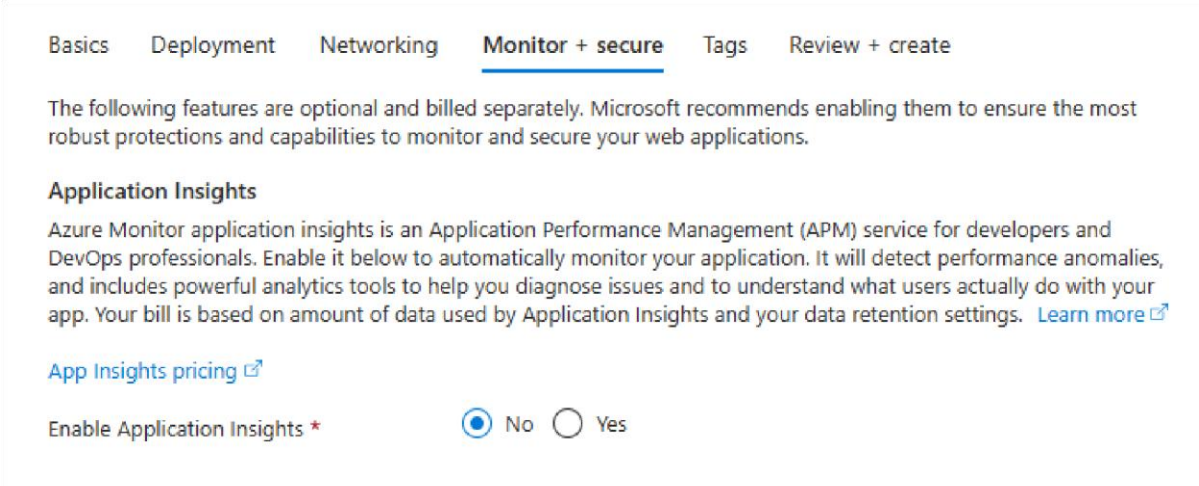
3. Provide a globally unique name for your web app. Choose **.NET6** to align with the project's framework and then select region.(eg. North Europe).



4. Choose your pricing accordingly. Default is **Standard S1**, but the **Free F1** plan limits compute to 60 minutes/day. So, select **Basic P1** to ensure the application runs without time limitations.

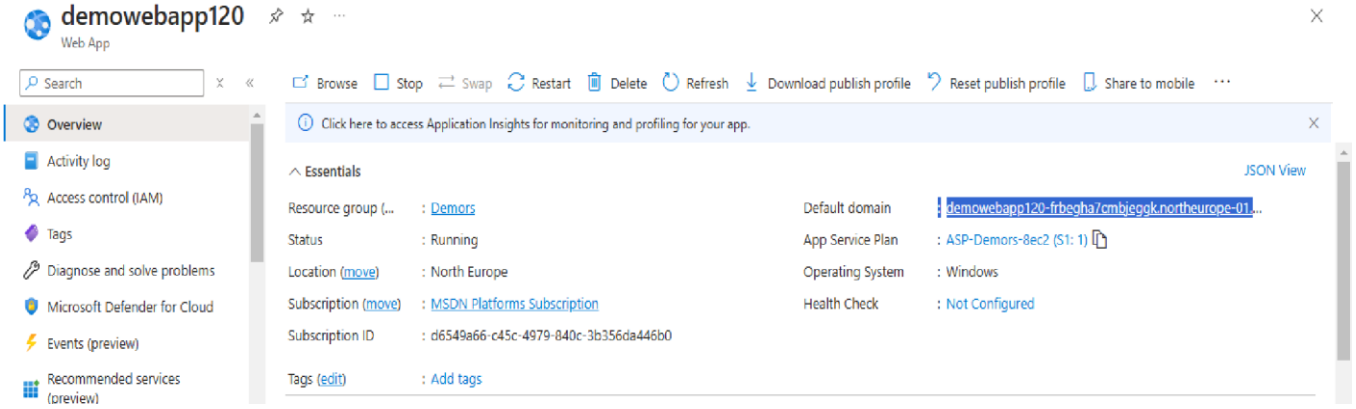


5. Disable **Application Insights** during setup (can be enabled later for diagnostics) insights and then move to review page and create your webapp.

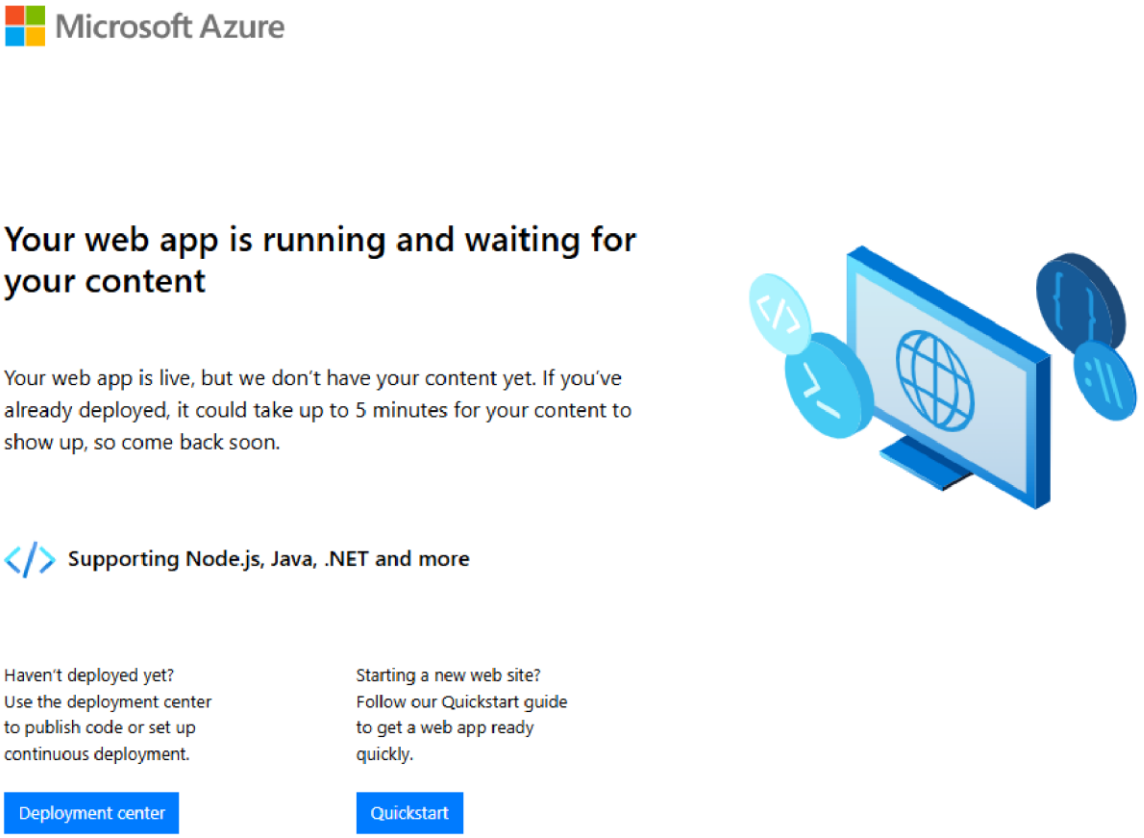


6. Review and finalize the configuration, then click **Create**.

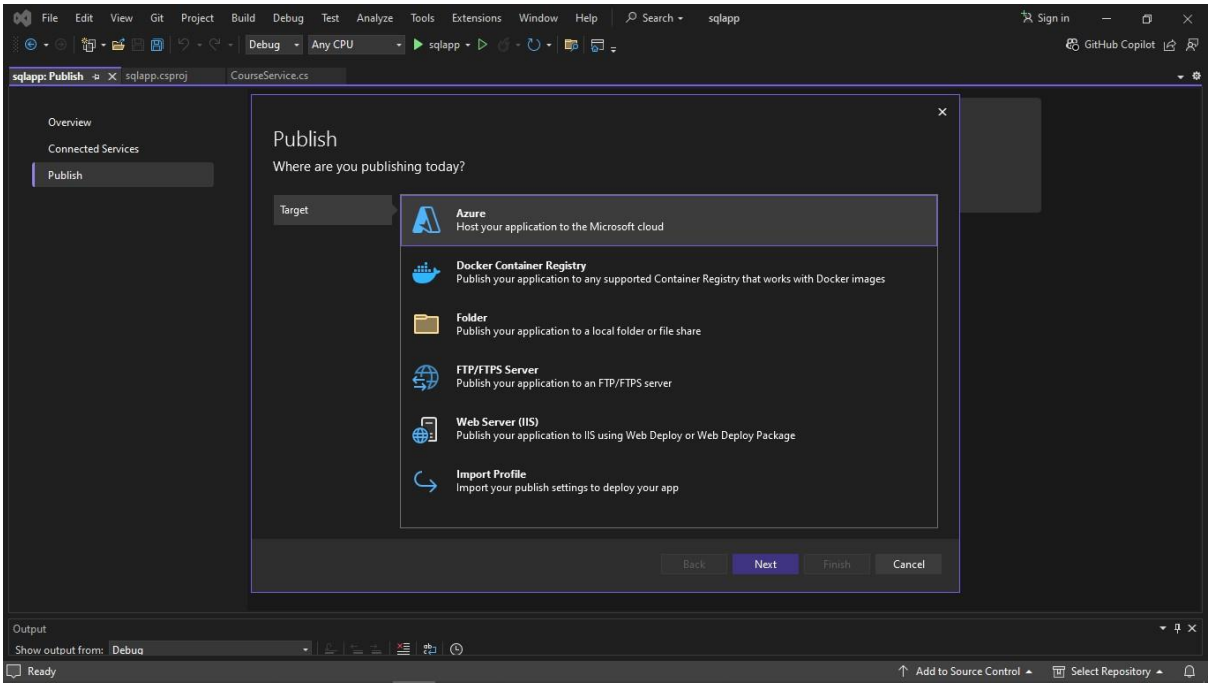
7. Once your webapp is deployed successfully, Access the default domain name in a new tab provided to see a default sample page confirming deployment success.

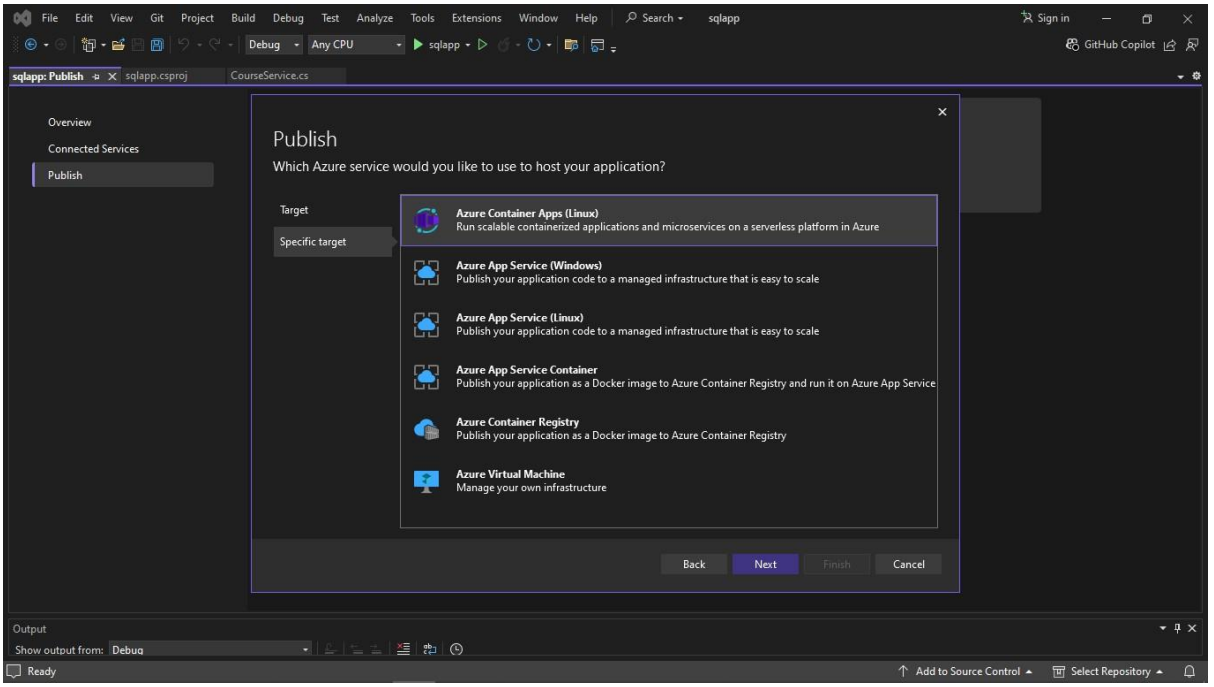


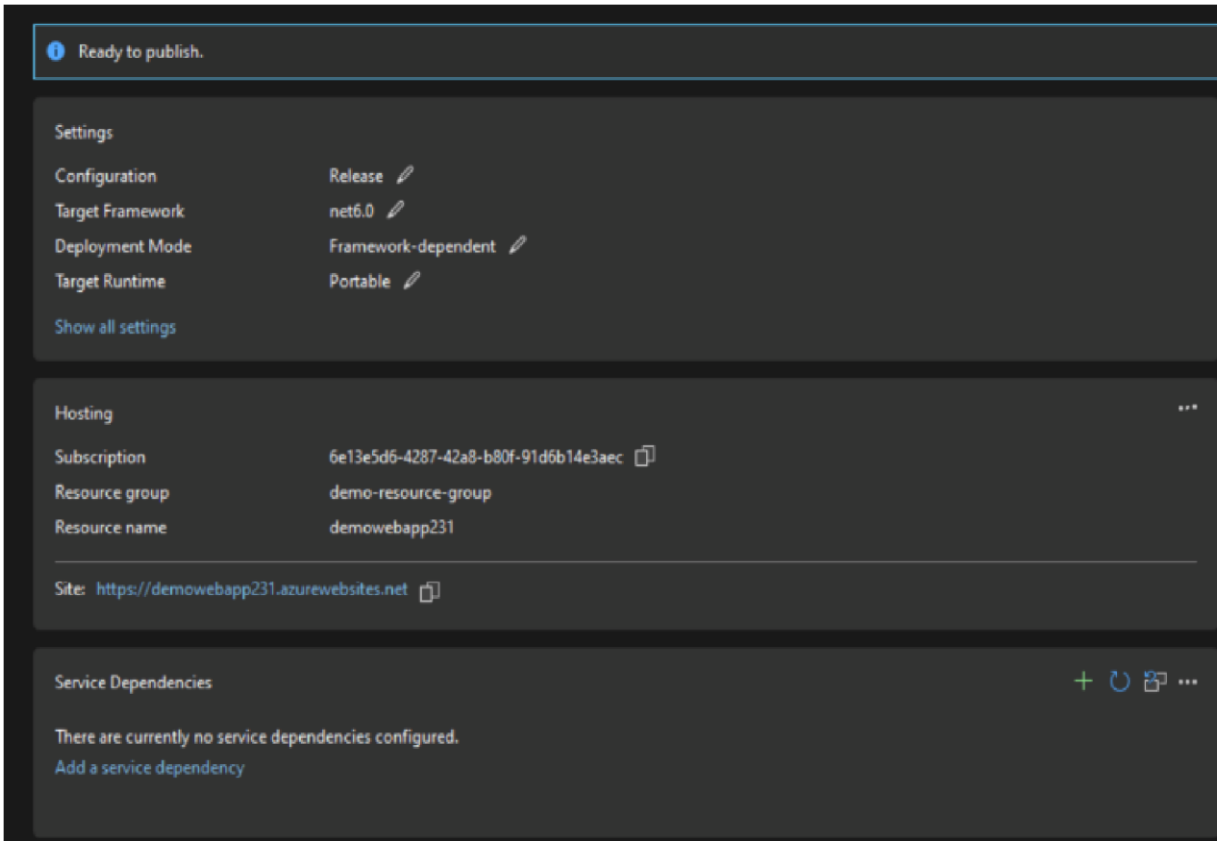
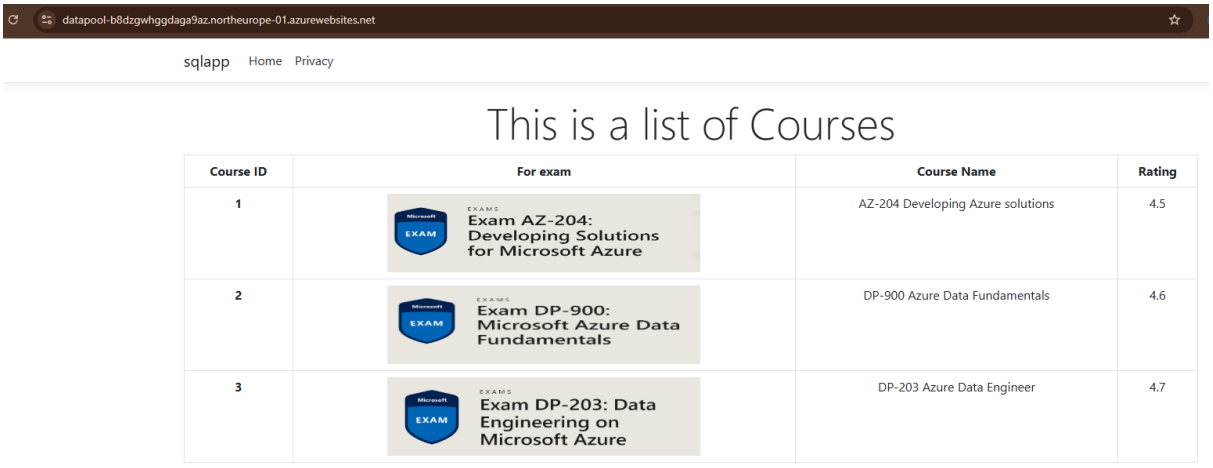
8. You will land to the newly created webapp.



9. In the Publish section, create a new profile Select **Azure** as the destination. Choose **Azure App Service (Windows)**. Select the subscription, resource group, and newly created web app. Click **Finish** to generate the publish profile.
10. Once the profile is generated you need to Publish it then wait for some time.


File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search
sqlapp
Sign in
Debug
Any CPU
sqlapp
GitHub Copilot
sqlapp: Publish
sqlapp.csproj
CourseService.cs
Overview
Connected Services
Publish
Publish
Where are you publishing today?
Target
Azure
Host your application to the Microsoft cloud
Docker Container Registry
Publish your application to any supported Container Registry that works with Docker images
Folder
Publish your application to a local folder or file share
FTP/FTPS Server
Publish your application to an FTP/FTPS server
Web Server (IIS)
Publish your application to IIS using Web Deploy or Web Deploy Package
Import Profile
Import your publish settings to deploy your app
Back
Next
Finish
Cancel
Output
Show output from: Debug
Ready
Add to Source Control
Select Repository


File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search
sqlapp
Sign in
Debug
Any CPU
sqlapp
GitHub Copilot
sqlapp: Publish
sqlapp.csproj
CourseService.cs
Overview
Connected Services
Publish
Publish
Which Azure service would you like to use to host your application?
Target
Specific target
Azure Container Apps (Linux)
Run scalable containerized applications and microservices on a serverless platform in Azure
Azure App Service (Windows)
Publish your application code to a managed infrastructure that is easy to scale
Azure App Service (Linux)
Publish your application code to a managed infrastructure that is easy to scale
Azure App Service Container
Publish your application as a Docker image to Azure Container Registry and run it on Azure App Service
Azure Container Registry
Publish your application as a Docker image to Azure Container Registry
Azure Virtual Machine
Manage your own infrastructure
Back
Next
Finish
Cancel
Output
Show output from: Debug
Ready
Add to Source Control
Select Repository



11. Refresh the web app's domain in a browser to see the deployed application running live. If the web page is not displayed then wait for some more time because it can take up to 5-10 minutes.