



## **UNIVERSITI KUALA LUMPUR KAMPUS KOTA MALAYSIAN INSTITUTE OF INFORMATION TECHNOLOGY**

Name of Course	<b>CLOUD COMPUTING</b>
Course Code	<b>IIB43203</b>
Lecturer	<b>MEGAT NORULAZMI</b>
Semester / Year	<b>MARCH 2025</b>
Student Name & ID	<b>NUR HUMAIRA BINTI ABDUL WAHAB (52224123318) NUR ZAYANEE MAHFUZAH BINTI ZAYZAMRY (52224123241)</b>
Assessment	<b>GROUP ASSIGNMENT REPORT</b>
Weightage	<b>20%</b>

### **YouTube Presentation Link:**

<https://youtu.be/JtzPP4sKz34?si=7aHwny3IVYbZPHKD>

### **INTRODUCTION**

This report details practical exercises in Azure focusing on two critical aspects of managing web applications, manual scaling for performance optimization and advanced deployment techniques for ensuring continuous availability. The first lab explores the fundamental concept of scaling out a web application. Through a simulated hotel reservation system experiencing heavy load, we'll demonstrate how increasing the number of web app instances directly impacts performance metrics like CPU utilization, response times, and error rates. This highlights the importance of proactively scaling in response to anticipated traffic surges or declining application performance.

The second lab then delves into sophisticated deployment strategies using Azure App Service deployment slots. This section illustrates how to achieve seamless updates with minimal to no service interruption, a crucial requirement for critical applications. Furthermore, it covers the process of rapidly rolling back to a previous stable version if a new deployment causes problems. This module also touches upon scaling up, which involves upgrading to more powerful hardware for long-term growth. Together, these labs provide essential insights into managing web app resources and deployments in Azure, equipping us with the knowledge to maintain high performance, ensure stability, and optimize operational costs in a dynamic cloud environment.

## 8-2 Stage and Scale Apps

### Create a web app

1. Sign in to the [Azure portal](#).
2. On the Azure portal menu or from the Home page, select Create a resource. The Create a resource pane appears.

The screenshot shows the Microsoft Azure Portal's Home page. At the top, there's a search bar and a 'Copilot' button. Below the search bar, there's a row of icons for various services: Create a resource, Resource groups, Virtual machines, Subscriptions, Policy, Quickstart Center, Azure AI Foundry, Kubernetes services, App Services, and More services. The 'Create a resource' icon is highlighted with a blue border. Below this is a section titled 'Resources' with tabs for 'Recent' (which is selected) and 'Favorite'. A table lists recent resources: sqlserver52224123241 (SQL server), DB-52224123241 (SQL database), SQLRG-52224123241 (Resource group), MyApp-52224123241 (App Service), AppRG-52224123241 (Resource group), VM-52224123241 (Virtual machine), RG-52224123241 (Resource group), and Azure for Students (Subscription). The 'Last Viewed' column shows the last time each resource was accessed. At the bottom of the 'Recent' table, there's a link to 'See all'.

3. In the left menu, select Web, and then search for and select Web App. The Create Web App pane appears.

The screenshot shows the 'Create a resource' pane. On the left, there's a sidebar with categories like Blockchain, Compute, Containers, Databases, Developer Tools, DevOps, Identity, Integration, Internet of Things, IT & Management Tools, Media, Migration, Mixed Reality, Monitoring & Diagnostics, Networking, Security, Storage, and Web. The 'Web' category is selected and highlighted with a grey background. The main area has a search bar at the top with the placeholder 'Search services and marketplace'. Below the search bar, there are two sections: 'Popular Azure services' (with links to Function App, Web App, Logic App, App Service Plan, Azure AI Search, Azure Bot, Container App, and Application Insights) and 'Popular Marketplace products' (with links to SharePoint Server 2016 Trial, Strapi on App Service, PostgreSQL on Ubuntu 24.04, Wordpress Simple, wordpress-v- 6.0.0, Bitnami package for WordPress, cPanel & WHM Version 124 on Ubuntu - Bring your own license, and Wordpress Ubuntu). At the bottom right of the pane, there's a 'Give feedback' button.

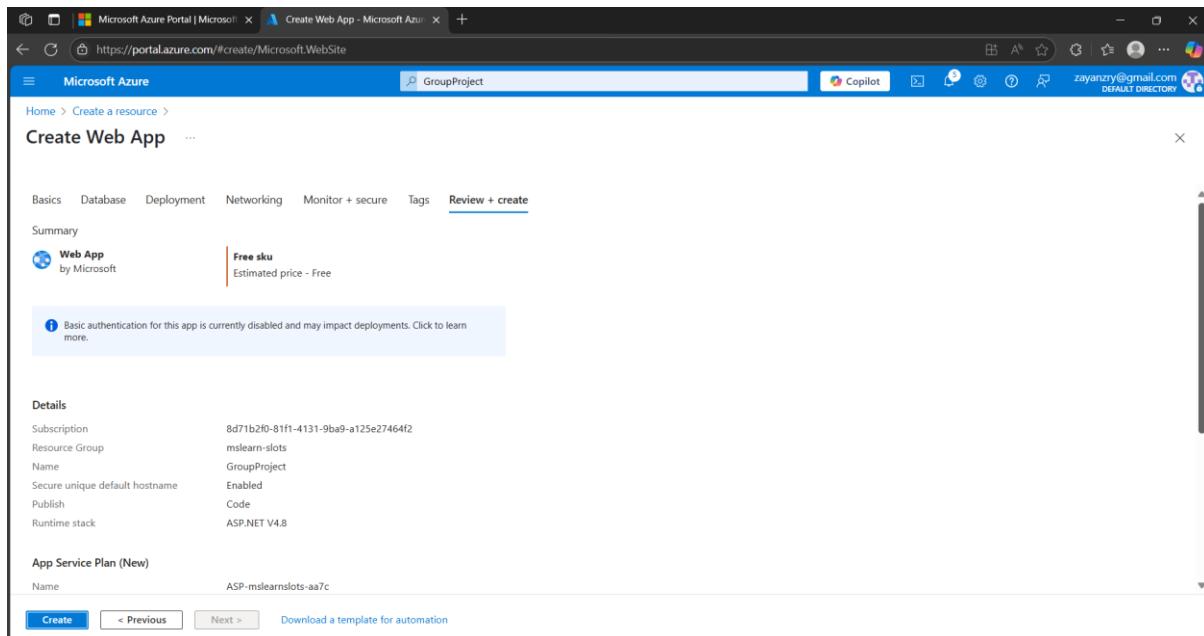
4. On the Basics tab, enter the following values for each setting.

The screenshot shows the 'Create Web App' wizard in the Microsoft Azure Portal. The current step is 'Project Details'. The user has selected 'Subscription' as 'Azure for Students' and 'Resource Group' as '(New) mslearn-slots'. The 'Name' field is set to 'GroupProject.azurewebsites.net'. Under 'Instance Details', the 'Secure unique default hostname' toggle is turned on. The 'Publish' section shows 'Code' selected. The 'Runtime stack' is set to 'ASP.NET V4.8'. The 'Operating System' is 'Windows'. The 'Region' is 'East US 2'. At the bottom, there are buttons for 'Review + create', '< Previous', and 'Next : Database >'.

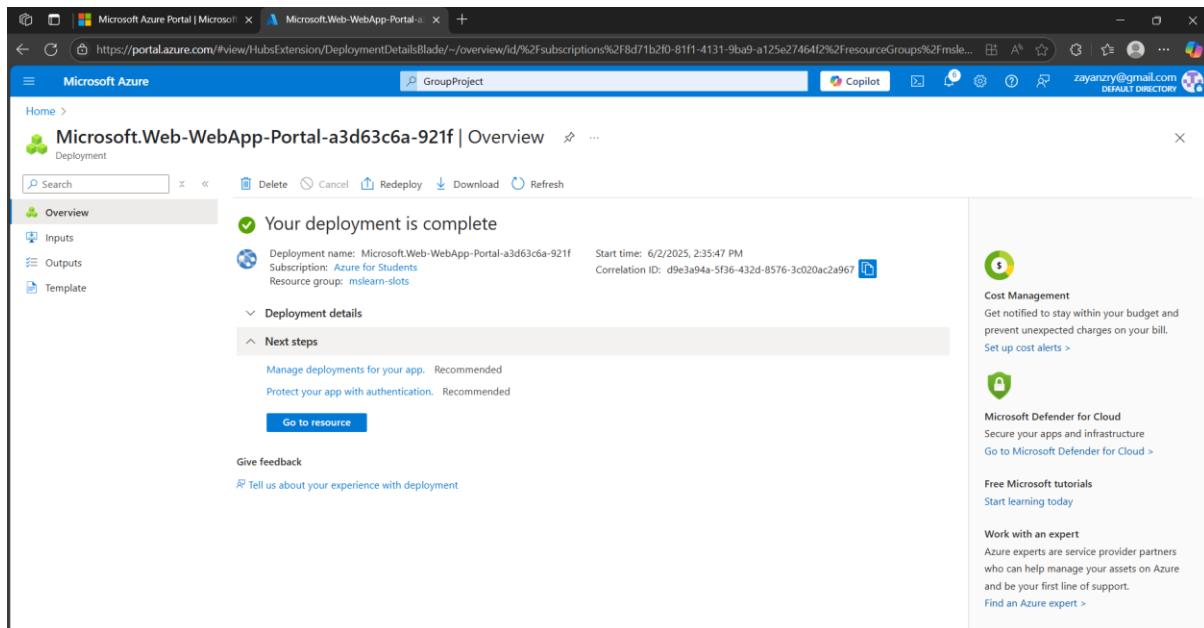
5. Select Next : Monitoring, and enter the following value for the setting.

The screenshot shows the 'Create Web App' wizard in the Microsoft Azure Portal, currently on the 'Monitor + secure' tab. The 'Application Insights' section is visible, showing a description of what it does and an 'App Insights pricing' link. The 'Enable Application Insights' field has 'No' selected. The 'Microsoft Defender for Cloud' section is present, with a note about getting a security solution when adding the service plan. The 'Defender for Cloud pricing' link is shown, and the 'Enable Defender for App Service' checkbox is unchecked. At the bottom, there are buttons for 'Review + create', '< Previous', and 'Next : Tags >'.

6. Select Review + create and when the content is validated, select Create. Wait while Azure creates the web app.

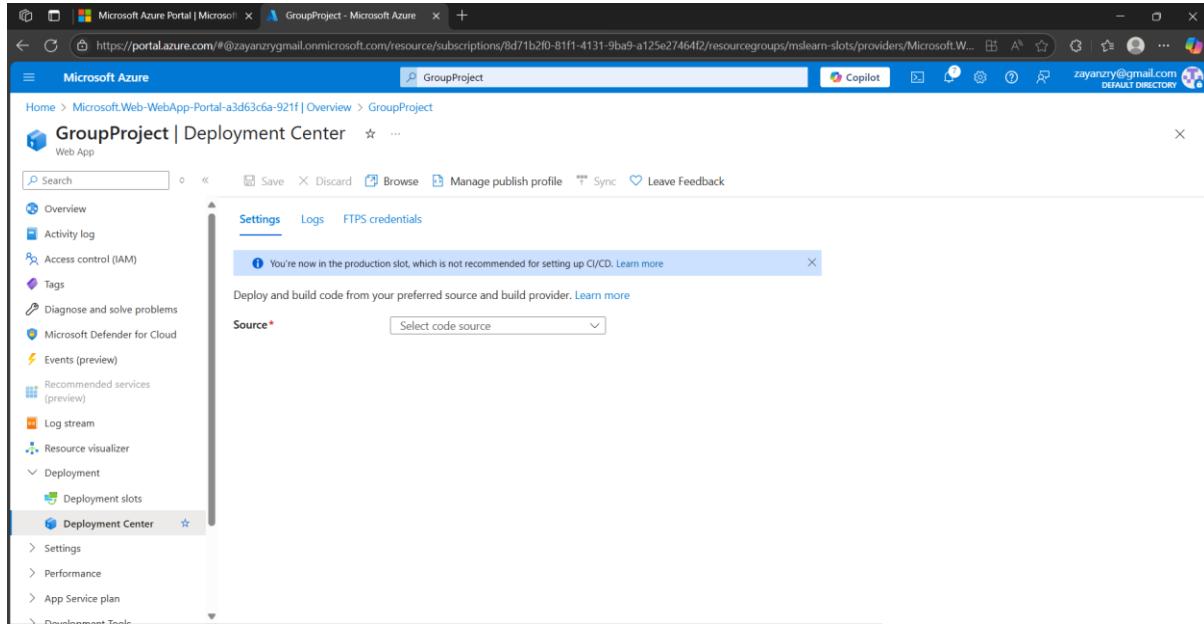


7. When deployment successfully completes, select Go to resource. The App Service pane appears.

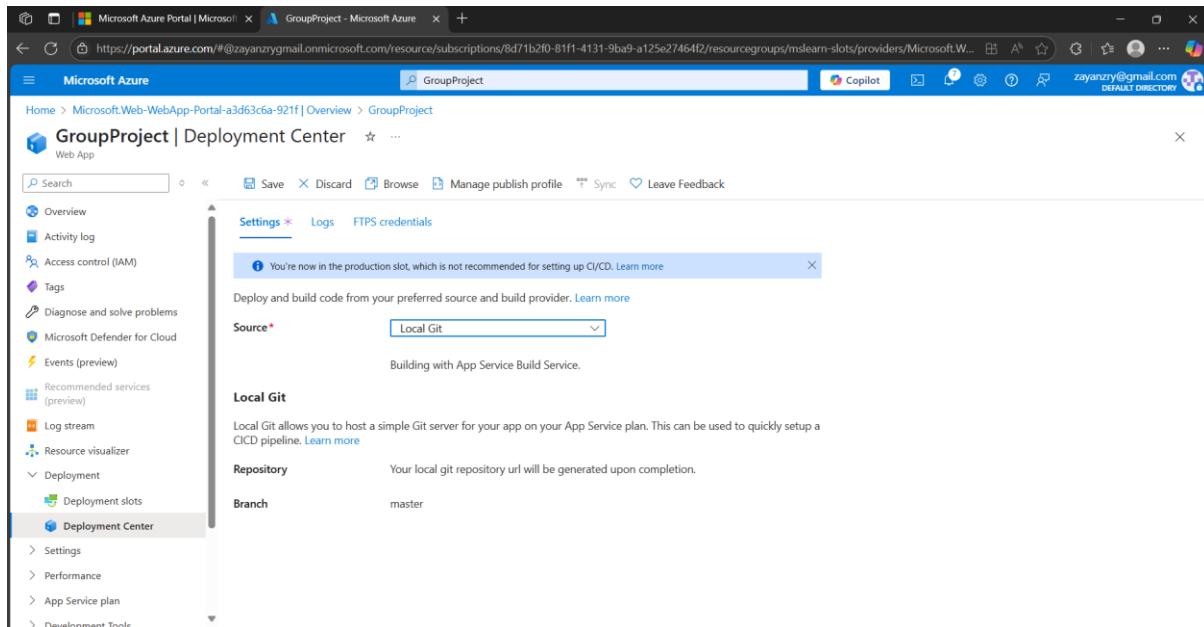


## Configure git deployment

1. On your web app App Service page, in the left menu, under Deployment, select Deployment Center.



2. On the Settings tab, for Source, select Local Git. On the top menu bar, select Save.



GroupProject | Deployment Center

Source Local Git

Git Clone Uri https://groupproject-eqqjagaeb8c7e7fx.scm.azurewebsites.net:443/GroupProject.git

Build

Build provider App Service Build Service

Runtime stack .NET

Version v4.0

3. On the resulting Deployment Center pane, select the Local Git/FTPS credentials tab.

GroupProject | Deployment Center

FTPS endpoint ftps://waws-prod-bn1-233.ftp.azurewebsites.windows.net/site/wwwroot

Git Clone Uri https://groupproject-eqqjagaeb8c7e7fx.scm.azurewebsites.net:443...

Application scope

FTPS Username GroupProject\GroupProject

Local Git Username \$GroupProject

Password [REDACTED]

User scope

4. Under User scope, enter a username and password of your choice, and in the top menu bar, select Save. Make a note of the username and password for later.

#### User scope

User scope credentials are defined by you, the user, and can be used with all the apps to which you have access. These credentials can be used with FTPS, Local Git and WebDeploy. Authenticating to an FTPS endpoint using user-level credentials requires a username in the following format: 'GroupProject\your username'. Authenticating with Git requires only the username '(your username)' defined below. [Learn more](#)

Username	<input type="text" value="ProjectCloudComputing"/>
Password	<input type="password" value="*****"/> <span>Reset</span>
Confirm Password	<input type="password" value="*****"/>

## Configure the git client and clone the web app source code

1. In the Azure portal, open Azure Cloud Shell by selecting its icon on the top toolbar.



2. In the Cloud Shell tool bar, ensure that Bash is selected. Copy the following code to Notepad and replace the values in quotes with your preferred username and email address. These config values aren't associated with any Azure account or sign-up, so you can use whatever values you like.
3. Copy and paste your edited code into the Cloud Shell and run it.

```
nur [ ~ ]$ git config --global user.name "ProjectCloudComputing"  
nur [ ~ ]$ git config --global user.email "zayanzry@gmail.com"
```

4. Create a folder for the source code. Run the following commands.

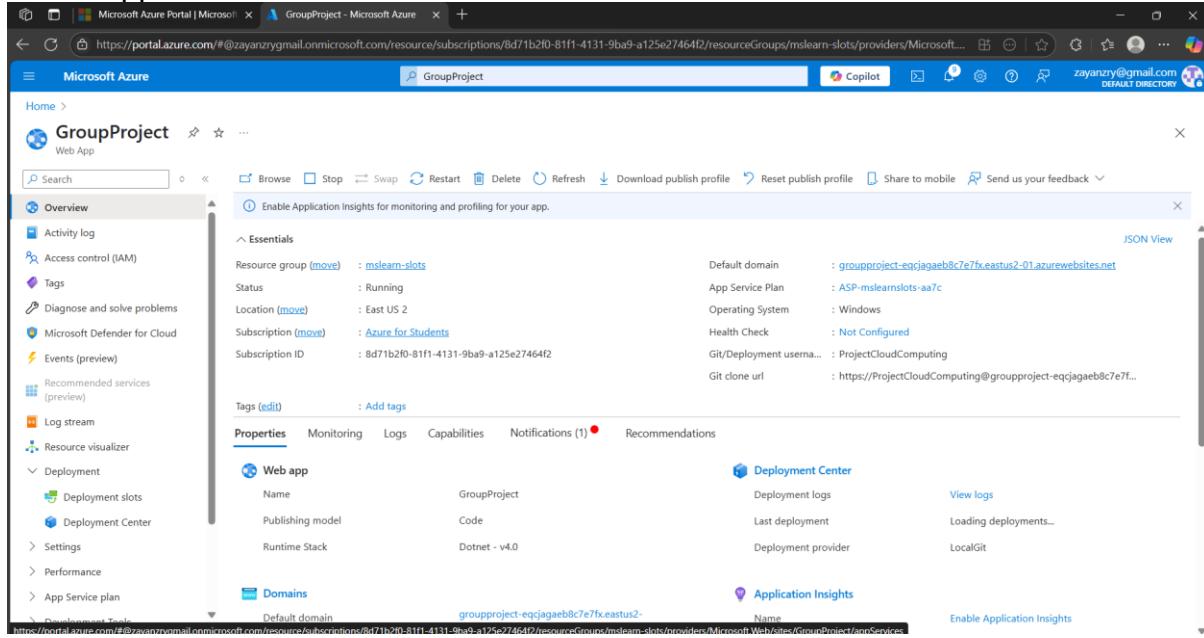
```
nur [ ~ ]$ mkdir demoapp  
nur [ ~ ]$ cd demoapp  
nur [ ~/demoapp ]$
```

5. Clone the source for the web app. Run the following commands. Bash (code text colour may differ in your environment)

```
nur [ ~/demoapp ]$ git clone https://github.com/Azure-Samples/app-service-web-dotnet-get-started.git  
Cloning into 'app-service-web-dotnet-get-started'...  
remote: Enumerating objects: 226, done.  
remote: Total 226 (delta 0), reused 0 (delta 0), pack-reused 226 (from 1)  
Receiving objects: 100% (226/226), 963.26 KiB | 26.03 MiB/s, done.  
Resolving deltas: 100% (99/99), done.  
nur [ ~/demoapp ]$ cd app-service-web-dotnet-get-started  
nur [ ~/demoapp/app-service-web-dotnet-get-started ]$
```

## Configure a git remote to deploy the app to production

1. In the Azure portal, your web app should be active. In the left menu, select Overview.
2. In the Overview pane for your web app note that the Essentials section has a URL and a Git clone url. Hover over the URL and select the *Copy to clipboard* icon. Note that the URL contains your deployment name for the web app.



The screenshot shows the Microsoft Azure Portal interface for a 'GroupProject' web app. The left sidebar shows various navigation options like Activity log, Access control (IAM), Tags, and Deployment. The main 'Overview' tab is selected. In the center, the 'Essentials' section provides key deployment details:

- Resource group (move) : mslearn-slots
- Status : Running
- Location (move) : East US 2
- Subscription (move) : Azure for Students
- Subscription ID : 8d71b2f0-81f1-4131-9ba9-a125e27464f2

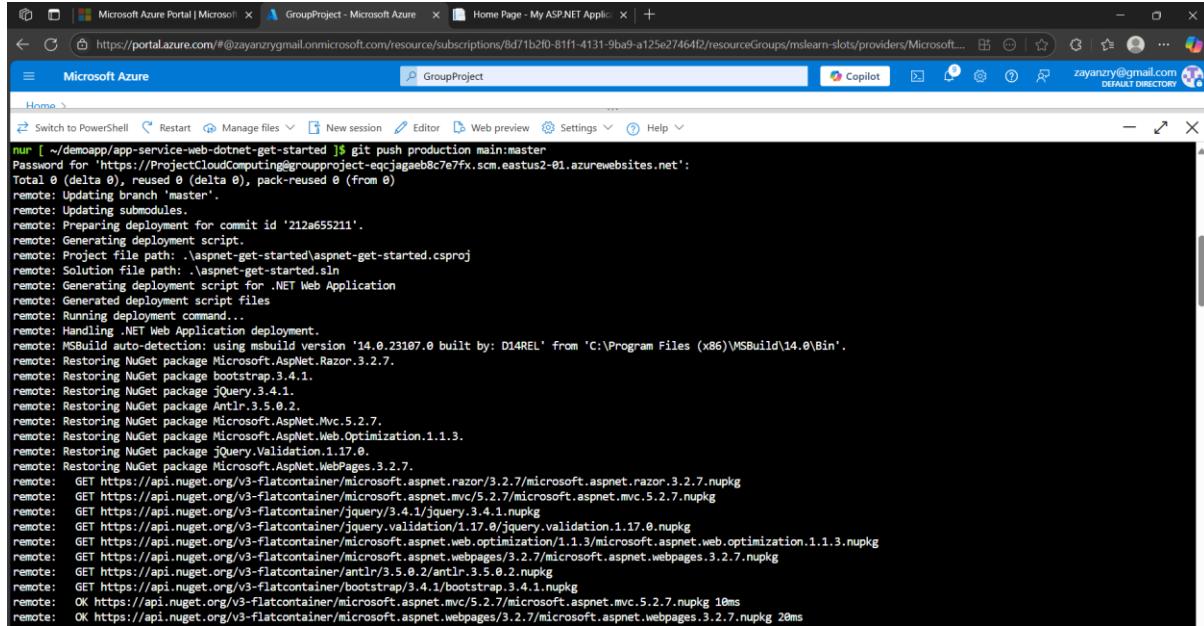
Below this, the 'Properties' tab is selected, showing:

- Web app: Name - GroupProject, Publishing model - Code, Runtime Stack - Dotnet - v4.0
- Deployment Center: Deployment logs, Last deployment, Deployment provider - LocalGit
- Domains: Default domain - groupproject-eqcjagaeb8c7e7fx.eastus2-01.azurewebsites.net
- Application Insights: Name, Enable Application Insights

3. Hover over the Git clone url value and select the Copy to clipboard icon. Note that this value also contains your deployment username.
4. In Cloud Shell, run the following command to configure the git remote with a name "production". Replace git-clone-url with the content you copied to the clipboard from the previous step.

```
nur [ ~/demoapp ]$ cd app-service-web-dotnet-get-started
nur [ ~/demoapp/app-service-web-dotnet-get-started ]$ git remote add production https://ProjectCloudComputing@groupproject-eqcjagaeb8c7e7fx.scm.eastus2-01.azurewebsites.net/GroupProject.git
nur [ ~/demoapp/app-service-web-dotnet-get-started ]$ []
```

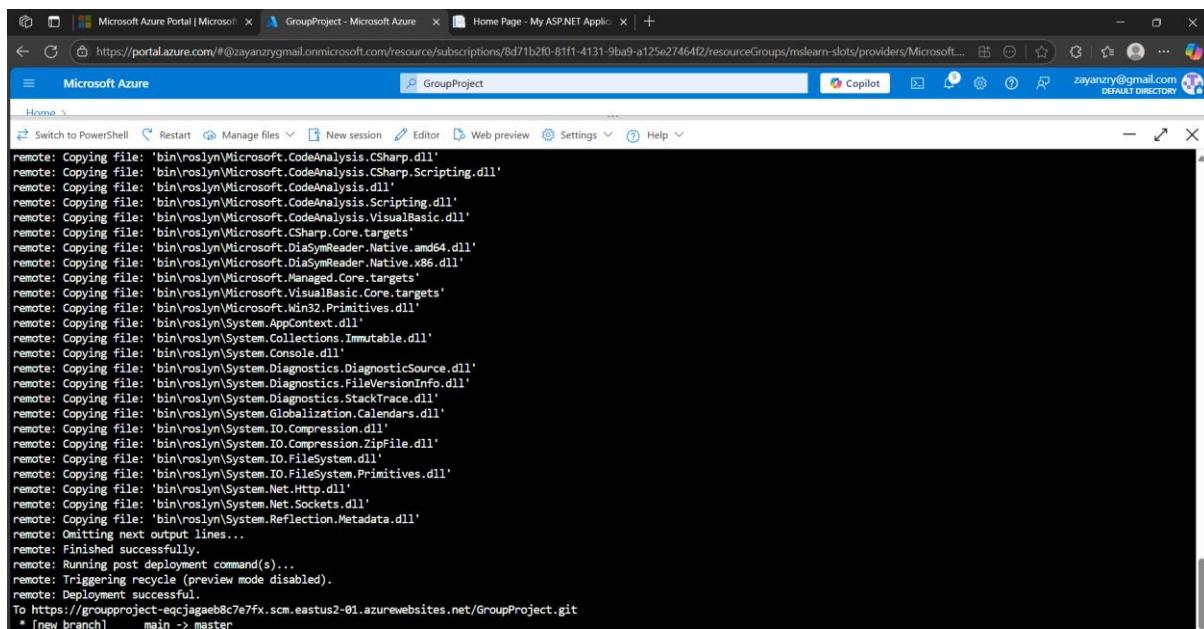
- To deploy the web app to the production slot, run the following command. When you're prompted for the password, enter your deployment password you created previously.



```

Microsoft Azure Portal | Microsoft | GroupProject - Microsoft Azure | Home Page - My ASP.NET App... | +
https://portal.azure.com/#@zayanzy@gmail.onmicrosoft.com/resource/subscriptions/8d71b2f0-81f1-4131-9ba9-a125e27464f2/resourceGroups/mslearn-slots/providers/Microsoft...
Switch to PowerShell Restart Manage files New session Editor Web preview Settings Help
zayanzy@gmail.com DEFAULT DIRECTORY
nur [ ~/demoapp/app-service-web-dotnet-get-started ]$ git push production main:master
Password for 'https://ProjectCloudComputing@groupproject-eqcjagae8c7e7fx.scm.eastus2-01.azurewebsites.net':
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Updating branch `master'.
remote: Updating submodules.
remote: Preparing deployment for commit id '212a655211'.
remote: Generating deployment script.
remote: Project file path: .\aspnet-get-started\aspnet-get-started.csproj
remote: Solution file path: .\aspnet-get-started.sln
remote: Generating deployment script for .NET Web Application
remote: Generated deployment script files
remote: Running deployment command...
remote: Handling .NET Web Application deployment.
remote: MSBuild auto-detection: using msbuild version '14.0.23107.0' built by: D14REL' from 'C:\Program Files (x86)\MSBuild\14.0\Bin'.
remote: Restoring NuGet package Microsoft.AspNet.Razor.3.2.7.
remote: Restoring NuGet package bootstrap.3.4.1.
remote: Restoring NuGet package jQuery.3.4.1.
remote: Restoring NuGet package Antlr.3.5.0.2.
remote: Restoring NuGet package Microsoft.AspNet.Mvc.5.2.7.
remote: Restoring NuGet package Microsoft.AspNet.Web.Optimization.1.1.3.
remote: Restoring NuGet package jQuery.Validation.1.17.0.
remote: Restoring NuGet package Microsoft.AspNet.WebPages.3.2.7.
remote: GET https://api.nuget.org/v3-flatcontainer/microsoft.aspnet.razor/3.2.7/microsoft.aspnet.razor.3.2.7.nupkg
remote: GET https://api.nuget.org/v3-flatcontainer/microsoft.aspnet.mvc/5.2.7/microsoft.aspnet.mvc.5.2.7.nupkg
remote: GET https://api.nuget.org/v3-flatcontainer/jquery/3.4.1/jquery.3.4.1.nupkg
remote: GET https://api.nuget.org/v3-flatcontainer/jquery.validation/1.17.0/jquery.validation.1.17.0.nupkg
remote: GET https://api.nuget.org/v3-flatcontainer/microsoft.aspnet.web.optimization/1.1.3/microsoft.aspnet.web.optimization.1.1.3.nupkg
remote: GET https://api.nuget.org/v3-flatcontainer/microsoft.aspnet.webpages/3.2.7/microsoft.aspnet.webpages.3.2.7.nupkg
remote: GET https://api.nuget.org/v3-flatcontainer/antlr/3.5.0.2/antlr.3.5.0.2.nupkg
remote: GET https://api.nuget.org/v3-flatcontainer/bootstrap/3.4.1/bootstrap.3.4.1.nupkg
remote: OK https://api.nuget.org/v3-flatcontainer/microsoft.aspnet.mvc/5.2.7/microsoft.aspnet.mvc.5.2.7.nupkg 10ms
remote: OK https://api.nuget.org/v3-flatcontainer/microsoft.aspnet.webpages/3.2.7/microsoft.aspnet.webpages.3.2.7.nupkg 20ms

```



```

Microsoft Azure Portal | Microsoft | GroupProject - Microsoft Azure | Home Page - My ASP.NET App... | +
https://portal.azure.com/#@zayanzy@gmail.onmicrosoft.com/resource/subscriptions/8d71b2f0-81f1-4131-9ba9-a125e27464f2/resourceGroups/mslearn-slots/providers/Microsoft...
Switch to PowerShell Restart Manage files New session Editor Web preview Settings Help
zayanzy@gmail.com DEFAULT DIRECTORY
remote: Copying file: 'bin\roslyn\Microsoft.CodeAnalysis.CSharp.dll'
remote: Copying file: 'bin\roslyn\Microsoft.CodeAnalysis.CSharp.Scripting.dll'
remote: Copying file: 'bin\roslyn\Microsoft.CodeAnalysis.dll'
remote: Copying file: 'bin\roslyn\Microsoft.CodeAnalysis.Scripting.dll'
remote: Copying file: 'bin\roslyn\Microsoft.CodeAnalysis.VisualBasic.dll'
remote: Copying file: 'bin\roslyn\Microsoft.CodeAnalysis.VisualBasic.targets'
remote: Copying file: 'bin\roslyn\Microsoft.CSharp.Core.targets'
remote: Copying file: 'bin\roslyn\Microsoft.Diagnostics.DiaSymReader.Native.amd64.dll'
remote: Copying file: 'bin\roslyn\Microsoft.Diagnostics.DiaSymReader.Native.x86.dll'
remote: Copying file: 'bin\roslyn\Microsoft.Managed.Core.targets'
remote: Copying file: 'bin\roslyn\Microsoft.VisualBasic.Core.targets'
remote: Copying file: 'bin\roslyn\Microsoft.Win32.Primitives.dll'
remote: Copying file: 'bin\roslyn\System.AppContext.dll'
remote: Copying file: 'bin\roslyn\System.Collections.Immutable.dll'
remote: Copying file: 'bin\roslyn\System.Console.dll'
remote: Copying file: 'bin\roslyn\System.Diagnostics.DiagnosticSource.dll'
remote: Copying file: 'bin\roslyn\System.Diagnostics.FileVersionInfo.dll'
remote: Copying file: 'bin\roslyn\System.Diagnostics.StackTrace.dll'
remote: Copying file: 'bin\roslyn\System.Globalization.Calendars.dll'
remote: Copying file: 'bin\roslyn\System.IO.Compression.dll'
remote: Copying file: 'bin\roslyn\System.IO.Compression.ZipFile.dll'
remote: Copying file: 'bin\roslyn\System.IO.FileSystem.dll'
remote: Copying file: 'bin\roslyn\System.IO.FileSystem.Primitives.dll'
remote: Copying file: 'bin\roslyn\System.Net.Http.dll'
remote: Copying file: 'bin\roslyn\System.Net.Sockets.dll'
remote: Copying file: 'bin\roslyn\System.Reflection.Metadata.dll'
remote: Omitting next output lines...
remote: Finished successfully.
remote: Running post deployment command(s)...
remote: Triggering recycle (preview mode disabled).
remote: Deployment successful.
To https://groupproject-eqcjagae8c7e7fx.scm.eastus2-01.azurewebsites.net/GroupProject.git
 * [new branch]      main -> master

```

- When the deployment finishes, in the Azure portal, go to the web app's Overview page, and then select URL. You can paste it into a browser or double-click to open the URL in a new tab.

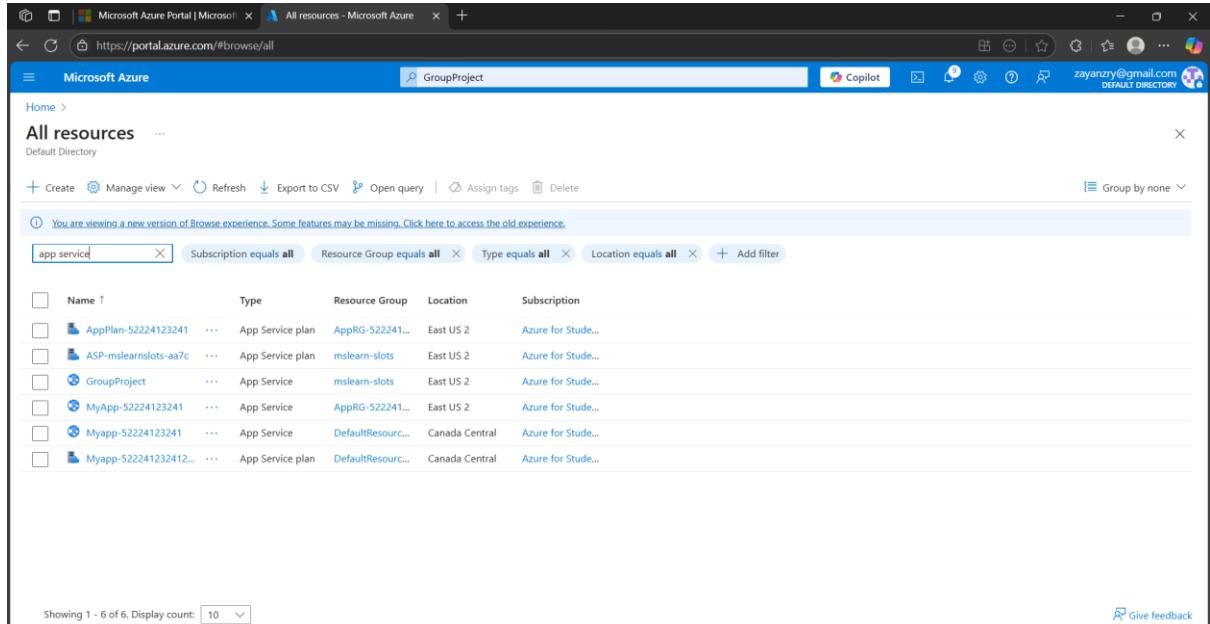
The screenshot shows the Microsoft Azure Portal interface. The left sidebar shows navigation options like Home, Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Microsoft Defender for Cloud, Events (preview), Recommended services (preview), Log stream, Resource visualizer, Deployment slots, Deployment Center, Settings, Performance, App Service plan, and Development Tools. The main content area is titled 'GroupProject' and shows the 'Overview' tab selected. It displays the following details:

- Essentials**
  - Resource group (move) : mslearn-slots
  - Status : Running
  - Location (move) : East US 2
  - Subscription (move) : Azure for Students
  - Subscription ID : 8d71b2f0-81f1-4131-9ba9-a125e27464f2
  - Tags (edit) : Add tags
- Properties** tab is selected, showing:
  - Web app**: Name - GroupProject, Publishing model - Code, Runtime Stack - Dotnet - v4.0
  - Deployment Center**: Deployment logs, Last deployment, Deployment provider
  - Domains**: Default domain - groupproject-eqcjagaeb8c7e7fx.eastus2-01.azurewebsites.net
  - Application Insights**: Name, Enable Application Insights
- Monitoring**, **Logs**, **Capabilities**, **Notifications (1)**, and **Recommendations** tabs are also present.

- Close the browser tab that displays the web app.

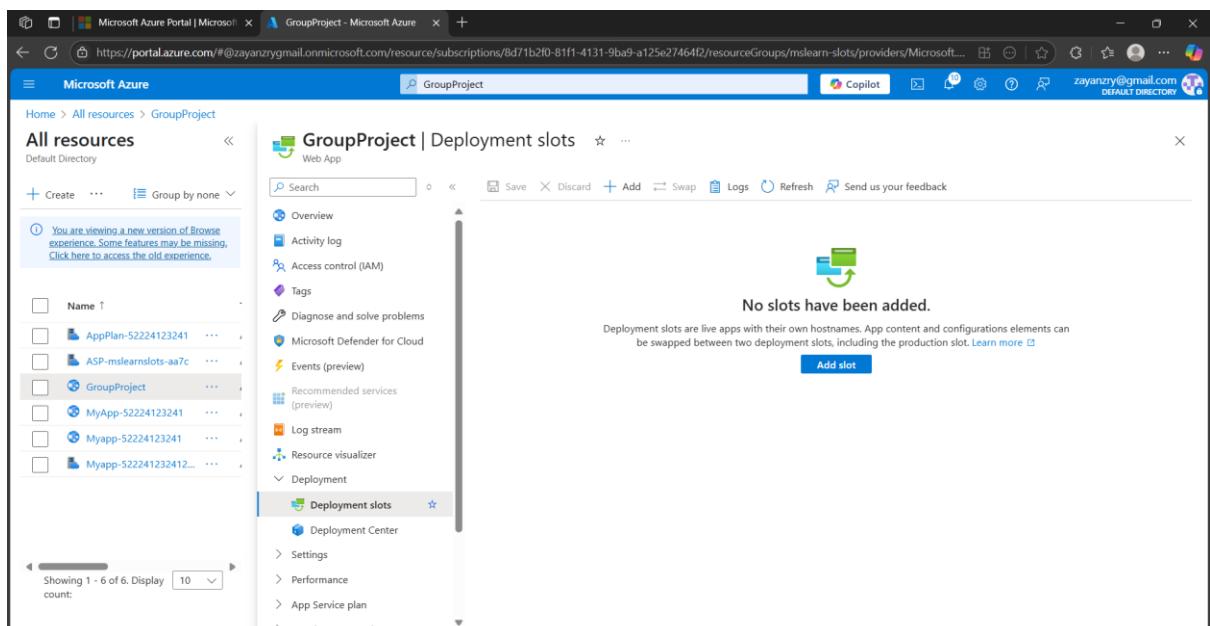
## Create a new staging slot

1. On the Azure portal menu or from the Home page, select All resources, and then filter by Type == App Service.



The screenshot shows the Microsoft Azure Portal's 'All resources' blade. The search bar at the top contains the text 'GroupProject'. Below the search bar is a filter bar with several dropdowns: 'Subscription equals all', 'Resource Group equals all', 'Type equals all' (which is currently selected), and 'Location equals all'. There is also a '+ Add filter' button. The main area displays a table of resources, each with a checkbox, name, type, resource group, location, and subscription information. The 'Group by none' button is located in the top right corner of the table area.

2. Select your web app. The web app App Service pane appears.
3. In the menu pane, under Deployment, select Deployment slots. The Deployment slots pane appears.



The screenshot shows the 'Deployment slots' pane for the 'GroupProject' web app. The title bar says 'GroupProject | Deployment slots'. The left sidebar has a tree view with nodes like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Microsoft Defender for Cloud, Events (preview), Recommended services (preview), Log stream, Resource visualizer, Deployment, Deployment slots (which is expanded), Deployment Center, Settings, Performance, App Service plan, and Development Tools. A message in the center says 'No slots have been added.' with a 'Add slot' button. The 'Deployment slots' node has a small icon with a green square and a blue arrow pointing up.

- From the top menu bar, select Add Slot. The Add a slot pane appears.
- In the Name field, enter Staging, accept the default for Clone settings from, and then select Add.

## Add Slot

X

Name

staging

groupproject-staging-a0gshagnfhf7f5en.eastus2-01.azurewebsites.net

Clone settings from:

Do not clone settings

- After the deployment slot is successfully created, select Close

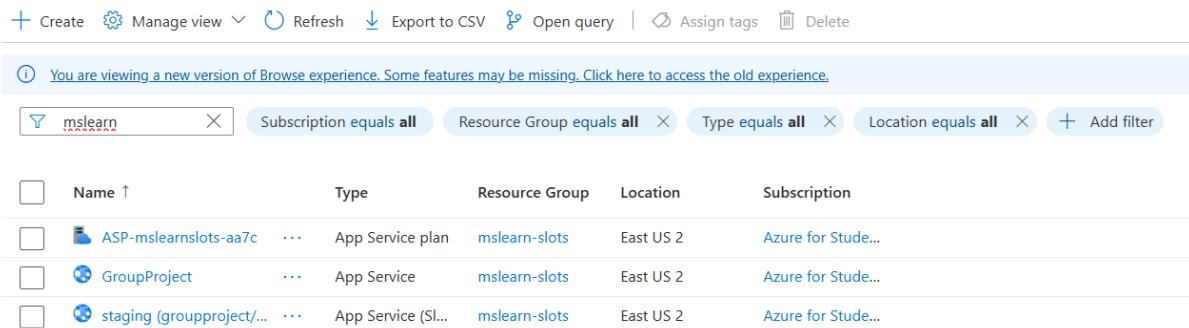
Name	Status	App service plan	Traffic %
groupproject <span style="background-color: green; color: white;">PRODUCTION</span>	Running	ASP-mslearnslots-aa7c	100
groupproject-staging	Running	ASP-mslearnslots-aa7c	0

## Set up git deployment for the staging slot

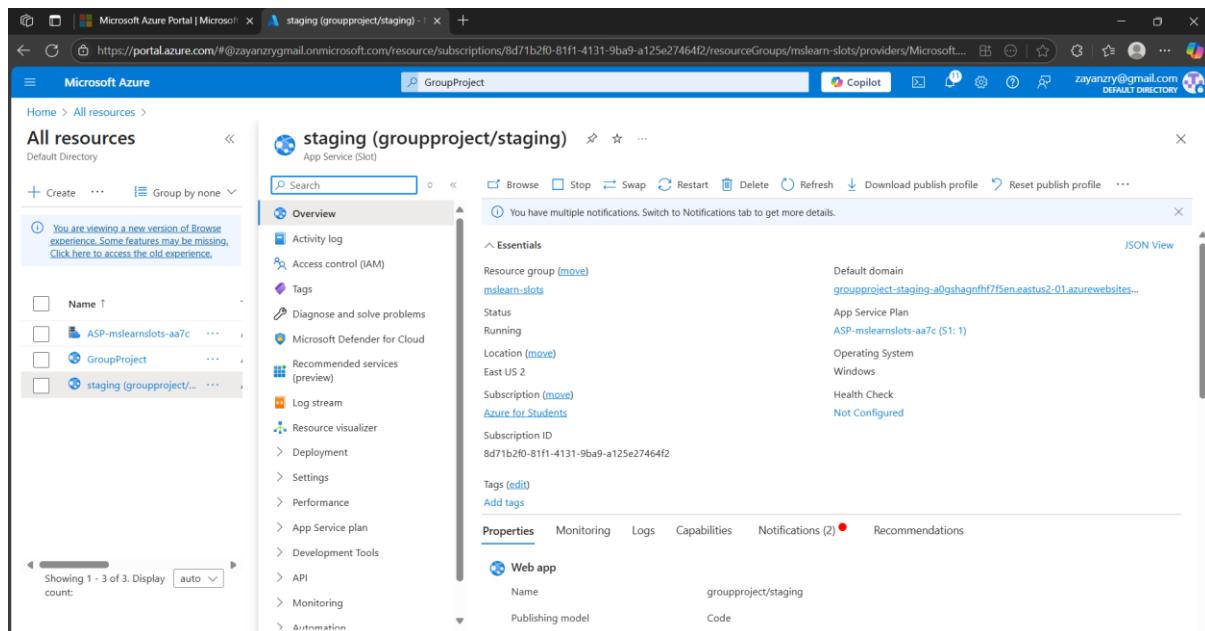
1. On the Azure portal menu or from the Home page, select All resources. In the list of all resources, you can filter on *Resource group == mslearn-slots*. You will see two App Service entries. Deployment slots are represented as separate apps in the portal. Select the entry representing the staging slot to go to its Overview pane.

### All resources

Default Directory



All resources					
Default Directory					
<a href="#">Create</a> <a href="#">Manage view</a> <a href="#">Refresh</a> <a href="#">Export to CSV</a> <a href="#">Open query</a> <a href="#">Assign tags</a> <a href="#">Delete</a>					
<a href="#">(i) You are viewing a new version of Browse experience. Some features may be missing. Click here to access the old experience.</a>					
<input type="text" value="mslearn"/> <a href="#">Subscription equals all</a> <a href="#">Resource Group equals all</a> <a href="#">Type equals all</a> <a href="#">Location equals all</a> <a href="#">Add filter</a>					
<input type="checkbox"/>	Name ↑	Type	Resource Group	Location	Subscription
<input type="checkbox"/>	ASP-mslearnslots-aa7c	App Service plan	mslearn-slots	East US 2	Azure for Stude...
<input type="checkbox"/>	GroupProject	App Service	mslearn-slots	East US 2	Azure for Stude...
<input type="checkbox"/>	staging (groupproject...)	App Service (Sl...	mslearn-slots	East US 2	Azure for Stude...



Microsoft Azure

Home > All resources >

All resources

Default Directory

[Create](#) [Group by none](#)

(i) You are viewing a new version of Browse experience. Some features may be missing. Click here to access the old experience.

Search: GroupProject

staging (groupproject/staging) - Microsoft Azure

Overview

You have multiple notifications. Switch to Notifications tab to get more details.

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Microsoft Defender for Cloud

Recommended services (preview)

Log stream

Resource visualizer

Deployment

Settings

Performance

App Service plan

Development Tools

API

Monitoring

Automation

Essentials

Resource group ([move](#))

mslearn-slots

Status

Running

Location ([move](#))

East US 2

Subscription ID

8d71b2f0-81f1-4131-9ba9-a125e27464f2

Tags ([edit](#))

Add tags

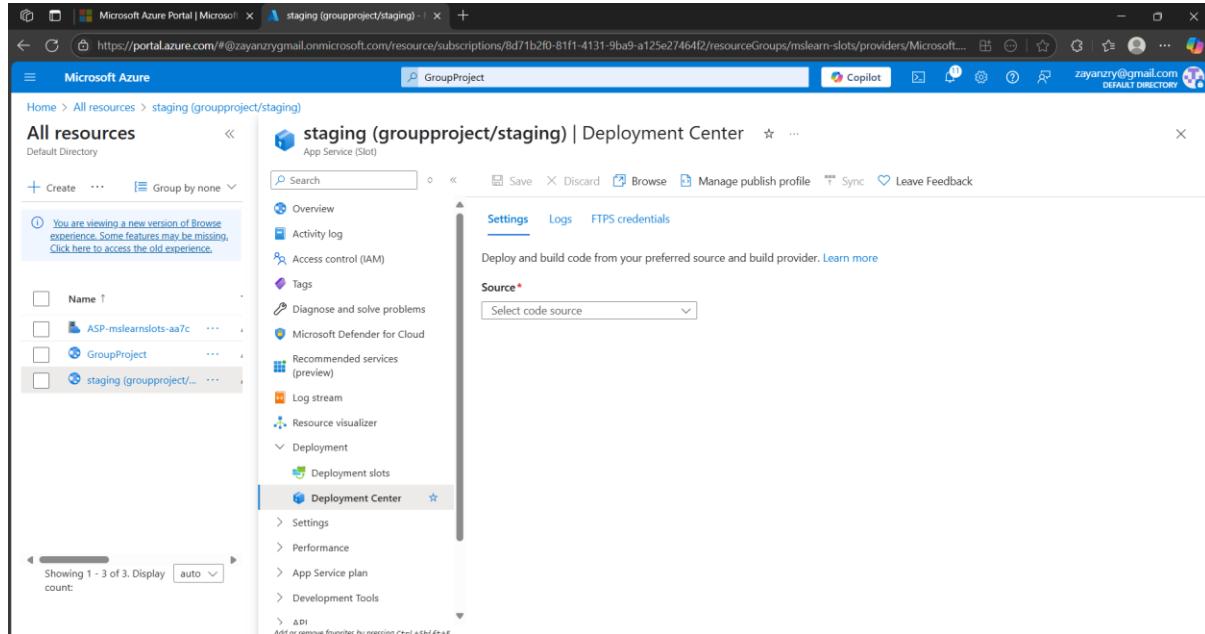
Properties Monitoring Logs Capabilities Notifications (2) Recommendations

Web app

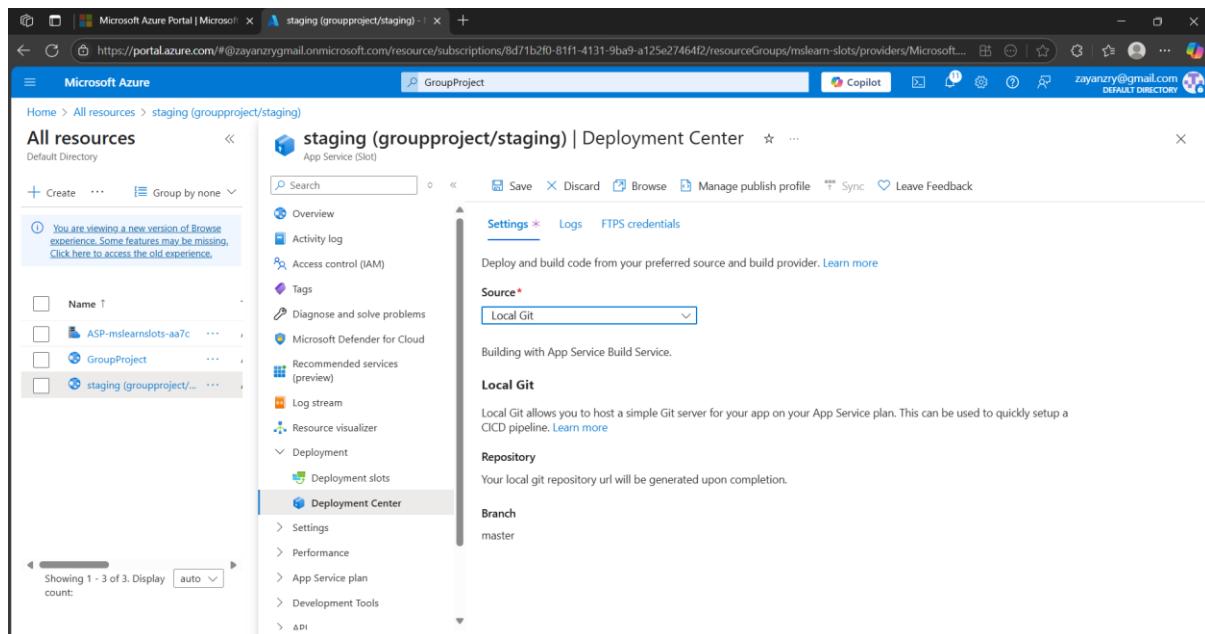
Name: groupproject/staging

Publishing model: Code

2. On the Overview pane, in the left menu, under Deployment, select Deployment Center.



3. On the Settings tab, for Source, select Local Git. In the top menu bar, select \*Save.



4. On the resulting Deployment Center pane, select the Local Git/FTPS credentials tab.

5. Under User scope, enter a new username and password of your choice, and from the top menu bar, select Save. Make a note of the username and password for later.

### User scope

User scope credentials are defined by you, the user, and can be used with all the apps to which you have access. These credentials can be used with FTPS, Local Git and WebDeploy. Authenticating to an FTPS endpoint using user-level credentials requires a username in the following format: 'groupproject-staging\ProjectCloudComputing'. Authenticating with Git requires only the username 'ProjectCloudComputing' defined below. [Learn more](#)

#### Username

#### Password

Reset

#### Confirm Password

## Set up git to deploy the app to the staging slot

1. In the Azure portal, on the staging web app's Overview page, in the Essentials section, select the *Copy to clipboard* icon for the Git clone url. Note that the URL contains your deployment username.
2. To add the remote for the staging slot, run the following command in Cloud Shell. Replace *git-clone-uri* with the URI from the previous step.

```
nur [ ~/demoapp/app-service-web-dotnet-get-started ]$ git remote add staging nur https://ProjectStagingCloudComputing@groupproject-staging-a0gshagnfhf7f5en.scm.eastus2-01.azurewebsites.net/groupproject.git
usage: git remote add [<options>] <name> <url>

-f, --[no-]fetch      fetch the remote branches
--[no-]tags          import all tags and associated objects when fetching
                     or do not fetch any tag at all (--no-tags)
-t, --[no-]track <branch>
                     branch(es) to track
-m, --[no-]master <branch>
                     master branch
--[no-]mirror[=(push|fetch)]
                     set up remote as a mirror to push to or fetch from
```

## Modify the app source code and deploy the app to the staging slot

1. In Cloud Shell, run the following command.

```
nur [ ~/demoapp/app-service-web-dotnet-get-started ]$ code .
```

2. In the list of Files, expand demoapp > app-service-web-dotnet-get-started > aspnet-get-started > Views > Home.

```
▲ demoapp
  ▲ app-service-web-dotnet-get-started
    ▶ .git
  ▲ aspnet-get-started
    ▶ App_Start
    ▶ Content
    ▶ Controllers
    ▶ fonts
    ▶ Properties
    ▶ Scripts
  ▲ Views
    ▲ Home
      About.cshtml
      Contact.cshtml
      Index.cshtml
```

3. Select Index.cshtml.
4. Locate the following code.

```
<h1>ASP.NET</h1>
```

5. Replace that code with this code.

```
<h1>Web App Version 2</h1>
```

6. To save your changes, press Ctrl+S, and then press Ctrl+Q to close the editor.

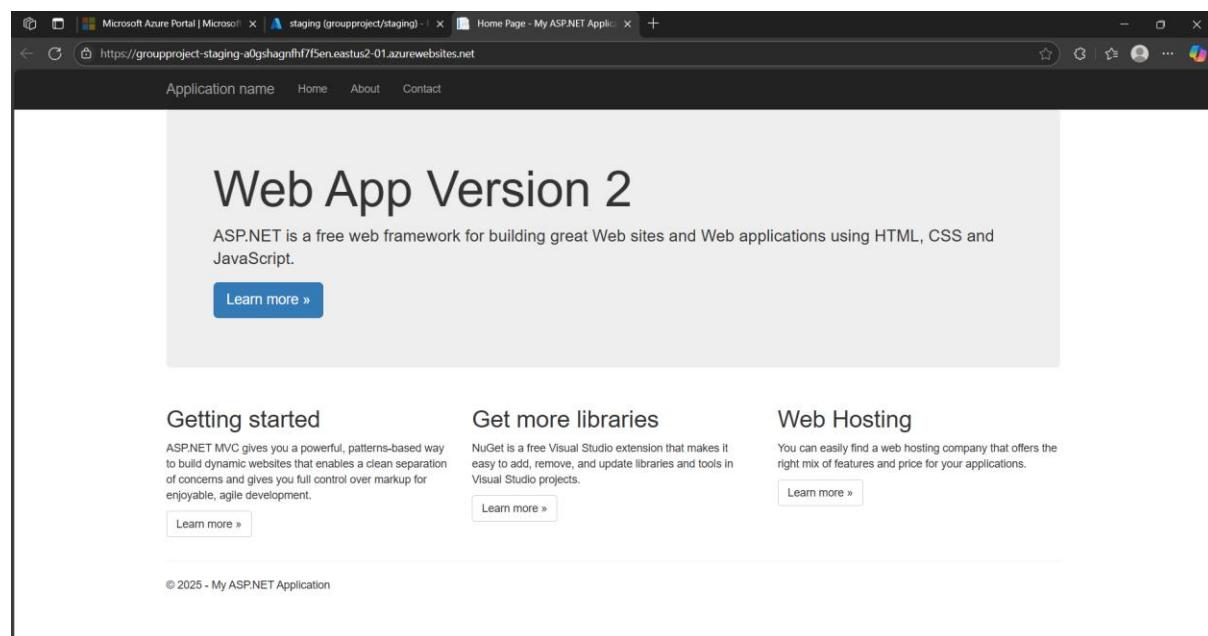
In Cloud Shell, run the following commands to commit the new version of the app to git, and deploy it to the staging slot. When prompted, enter your deployment password.

```
nur [ ~/demoapp/app-service-web-dotnet-get-started ]$ git config --global user.name "ProjectStagingCloudComputing"
nur [ ~/demoapp/app-service-web-dotnet-get-started ]$ git config --global user.email "zayanzy@gmail.com"
nur [ ~/demoapp/app-service-web-dotnet-get-started ]$ git commit -m "New version of web app."
[main 8551e0d] New version of web app.
 1 file changed, 2 insertions(+), 2 deletions(-)
nur [ ~/demoapp/app-service-web-dotnet-get-started ]$ git push staging main:master
Password for 'https://ProjectStagingCloudComputing@groupproject-staging-a0gshagnfhf7f5en.scm.eastus2-01.azurewebsites.net':
```

## Browse the staging slot

Now you can view the new version of the web app by browsing to the staging deployment slot's URL.

In the Azure portal, go to the Overview page for the staging slot. In the top menu bar, select Browse. The new version of the web app appears in a browser tab.



## Configure a slot setting

1. From the All resources view in the Azure portal, navigate to the Overview page of the production slot of the web app (this is the main webapp).
2. Navigate to the Configuration page.
3. Select New application setting. Add a new setting with the name ENVIRONMENT\_NAME and a value of production. Check the deployment slot setting box to make this a slot setting.
4. Add another setting called APP\_VERSION, and enter the value 1. *Don't* make this a slot setting.
5. Near the top of the page, select Save.

The screenshot shows the 'App settings' tab selected in the Azure portal. At the top, there's a search bar and several action buttons: '+ Add', 'Refresh', 'Hide values', 'Advanced edit', and 'Pull reference values'. Below the header is a table with columns: Name, Value, Deployment slot setting, Source, and Delete. Two rows are present:

Name	Value	Deployment slot setting	Source	Delete
APP_VERSION	1		App Service	
ENVIRONMENT_NAME	production	✓	App Service	

6. Repeat the preceding steps on the Staging slot, but use the following values:

The screenshot shows the 'App settings' tab selected in the Azure portal. At the top, there's a search bar and several action buttons: '+ Add', 'Refresh', 'Show values', 'Advanced edit', and 'Pull reference values'. Below the header is a table with columns: Name, Value, Deployment slot setting, Source, and Delete. Three rows are present:

Name	Value	Deployment slot setting	Source	Delete
APP_VERSION	2		App Service	
ENVIRONMENT_NAME	staging	✓	App Service	
WEBSITE_NODE_DEFAULT_VER...			App Service	

## Swap the slots

1. To make sure you're configuring the production slot, select All resources, and then select the production slot of the web app.
2. In the left menu pane, under Deployment, select Deployment slots > Swap.

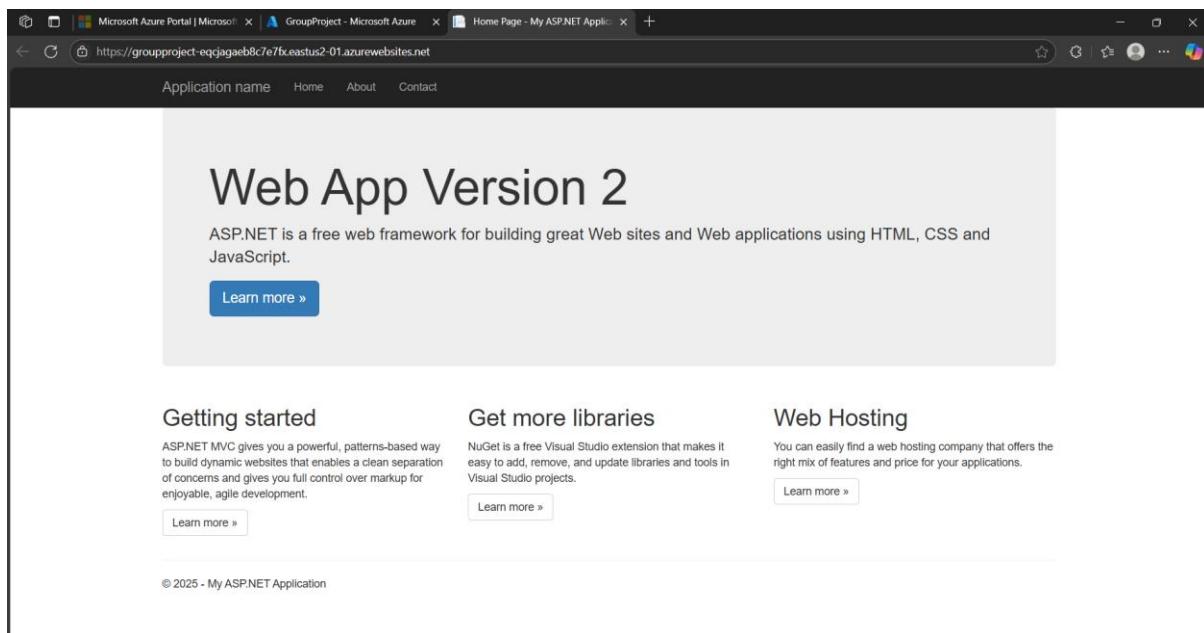
The screenshot shows the Azure portal interface for managing deployment slots. At the top, there are buttons for Save, Discard, Add, Swap (which is highlighted in blue), Logs, and more. Below this, a descriptive text states: "Deployment slots are live apps with their own hostnames. App content and configurations elements can be swapped between two deployment slots, including the production slot." A table lists two slots: "groupproject" (Status: Running, App service plan: ASP-mslearnsl) and "groupproject-staging" (Status: Running, App service plan: ASP-mslearnsl). At the bottom of the table are navigation arrows.

3. Make sure you're about to swap the staging and production slots. Notice how the swap will affect settings. The value of the APP\_VERSION setting will be exchanged between the slots, but the value of the ENVIRONMENT slot setting won't be swapped. Select Swap.

The screenshot shows the "Swap" dialog box. It has fields for "Source" (set to "groupproject-staging") and "Target" (set to "PRODUCTION" which is highlighted in green). There is also a checkbox for "Perform swap with preview". Below this, a section titled "Config Changes" provides a summary of configuration changes. It includes tabs for "Source slot changes" (selected) and "Target slot changes". A table at the bottom details the changes:

Setting	Type	Old Value	New Value
PhpVersion	General	5.6	
APP_VERSION	AppSetting	2	1
WEBSITE_NODE_DEFAULT_...	AppSetting	6.9.1	Not set

- When the swap is complete, go to the Overview page of the production slot's web app, and select Browse. The web app appears on a new browser tab. Notice that version 2 of the web app is now in production.



- Close the browser tab.

## Configure auto swap for the staging slot

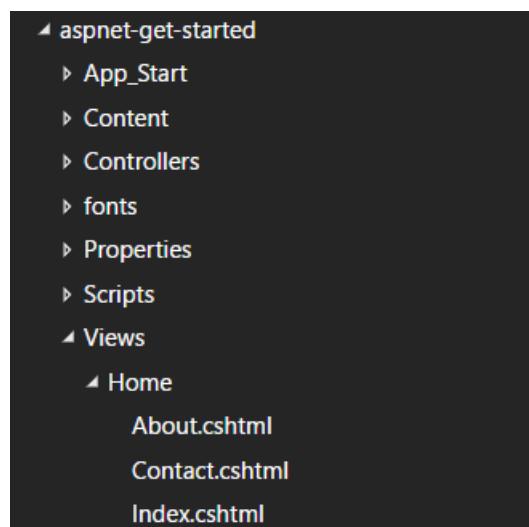
- Go to the Configuration page of the staging slot's web app, and navigate to the General settings tab.
- Set Auto swap enabled to On.
- In the Auto swap deployment slot list, select production, and then select Save.

## Deploy new code and auto swap it into production

1. On the right side of the Cloud Shell window, restart the editor if it's not already running.

```
nur [ ~ ]$ cd ~/demoapp/app-service-web-dotnet-get-started/
nur [ ~/demoapp/app-service-web-dotnet-get-started ]$ code .
```

2. In the code editor, in the File list on the left, expand aspnet-get-started > Views > Home, and then select Index.cshtml.



3. Locate the following code

```
<h1>Web App Version 2</h1>
```

4. Replace that code with this code.

```
<h1>Web App Version 3</h1>
```

5. To save your changes, press Ctrl+S.
6. In Cloud Shell, enter the following commands. Enter your deployment password when you're prompted.

```
nur [ ~/demoapp/app-service-web-dotnet-get-started ]$ git add .
nur [ ~/demoapp/app-service-web-dotnet-get-started ]$ git commit -m "Third version of web app."
On branch main
Your branch is ahead of 'origin/main' by 2 commits.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
nur [ ~/demoapp/app-service-web-dotnet-get-started ]$ git push staging
Password for 'https://ProjectStagingCloudComputing@groupproject-staging-a0gshagnfhf7f5en.scm.eastus2-01.azurewebsites.net':
```

7. Wait for the deployment to finish. Near the end of the text output, you'll see a message that indicates that the deployment has requested an auto swap to the production slot.

```
remote: Deployment successful.  
remote: Requesting auto swap to 'production' slot with 'AUTOSWAPf9caacd6-bd36-409e-859e-5e26ebbe4061' id successful.  
To https://groupproject-staging-a0gshagnfhf7f5en.scm.azurewebsites.net/groupproject.git  
 212a655..0c660b6  main -> main
```

8. In the Azure portal, navigate to the Overview page for the production slot's web app, and select Browse. The third version of the web app appears on a new browser tab. If the old version is shown, you may need to wait briefly and then refresh the page - the swap operation is atomic and occurs instantly, but it takes App Service a few moments to prepare the swap operation before it's execute.

Application name Home About Contact

# Web App Version 3

ASP.NET is a free web framework for building great Web sites and Web applications using HTML, CSS and JavaScript.

[Learn more »](#)

## Getting started

ASP.NET MVC gives you a powerful, patterns-based way to build dynamic websites that enables a clean separation of concerns and gives you full control over markup for enjoyable, agile development.

[Learn more »](#)

## Get more libraries

NuGet is a free Visual Studio extension that makes it easy to add, remove, and update libraries and tools in Visual Studio projects.

[Learn more »](#)

## Web Hosting

You can easily find a web hosting company that offers the right mix of features and price for your applications.

[Learn more »](#)

© 2025 - My ASP.NET Application

Application name Home About Contact

# Web App Version 3

ASP.NET is a free web framework for building great Web sites and Web applications using HTML, CSS and JavaScript.

[Learn more »](#)

## Getting started

ASP.NET MVC gives you a powerful, patterns-based way to build dynamic websites that enables a clean separation of concerns and gives you full control over markup for enjoyable, agile development.

[Learn more »](#)

## Get more libraries

NuGet is a free Visual Studio extension that makes it easy to add, remove, and update libraries and tools in Visual Studio projects.

[Learn more »](#)

## Web Hosting

You can easily find a web hosting company that offers the right mix of features and price for your applications.

[Learn more »](#)

© 2025 - My ASP.NET Application

## Roll back the new version

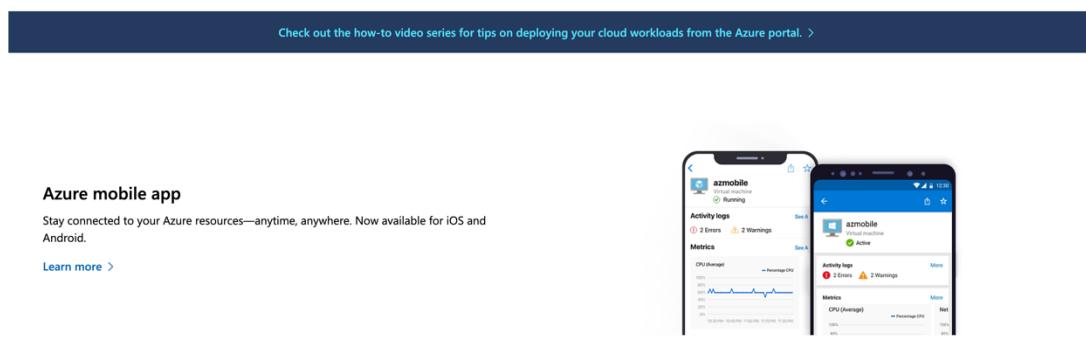
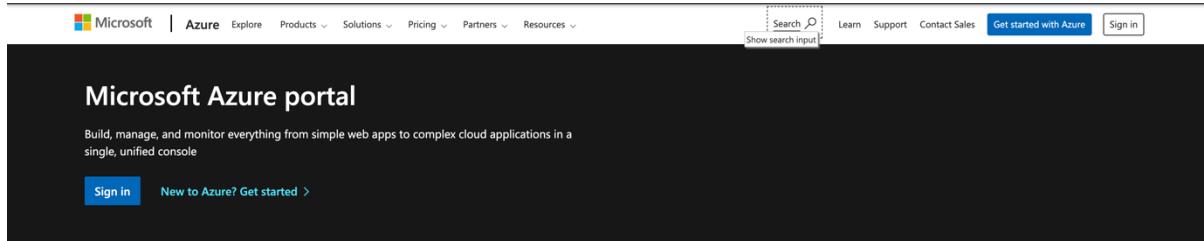
1. Go to the Deployment slots page of the production slot's web app.
2. Swap the staging and production slots.
3. When the swap finishes, on the Overview page, select Browse to view the app one last time. You'll see that version 2 has been redeployed to production.

The screenshot shows a web application interface. At the top, there is a dark navigation bar with the text "Application name" followed by "Home", "About", and "Contact". Below this, the main content area has a light gray background. In the center, the text "Web App Version 2" is displayed in a large, bold, black font. Below this, a smaller text block reads: "ASP.NET is a free web framework for building great Web sites and Web applications using HTML, CSS and JavaScript." To the right of this text is a blue rectangular button with the white text "Learn more »". Further down the page, there are three sections: "Getting started", "Get more libraries", and "Web Hosting". Each section contains a brief description and a "Learn more »" button. At the very bottom of the page, there is a small copyright notice: "© 2025 - My ASP.NET Application".

## 9-1 Scale an App Service Web App

### Step 1: Create an App Service Plan and Web App

1. Sign in to Azure Portal and log in with your Azure account.



2. Create a new Web App

- Click Create a resource from the Home Page.

The screenshot shows the Microsoft Azure home page. At the top, there's a search bar and a Copilot button. Below the search bar is a section titled "Azure services" with a "Create a resource" button highlighted with a red box. Other buttons in this section include "Resource groups", "All resources", "Subscriptions", "Virtual machines", "Quickstart Center", "Azure AI Foundry", "Kubernetes services", "App Services", and "More services". Below this is a "Resources" section with tabs for "Recent" and "Favorite". It shows a table with columns for "Name", "Type", and "Last Viewed". A message "No resources have been viewed recently" is displayed, along with a "View all resources" button. Further down are sections for "Navigate" (with links to Subscriptions, Resource groups, All resources, and Dashboard) and "Tools" (with links to Microsoft Learn, Azure Monitor, Microsoft Defender for Cloud, and Cost Management). At the bottom left, there's a URL: "https://portal.azure.com/#createBlurb".

- In the Categories section , select Web then choose Web App and click Create.

The screenshot shows the Microsoft Azure portal's 'Create a resource' interface. On the left, there's a sidebar with various service categories like AI + Machine Learning, Analytics, Blockchain, Compute, etc. The 'Web' category is at the bottom. In the main area, there's a search bar and a 'Popular Azure services' section. Under 'Popular Azure services', the 'Web App' icon is circled in black. To its right is a 'Popular Marketplace products' section listing various cloud services with their icons and names.

### 3. Fill out the Basics Tab

The screenshot shows the 'Create Web App' configuration page. The 'Project Details' tab is selected. It includes fields for 'Subscription' (set to 'Azure for Students'), 'Resource Group' (set to 'mslearn-scale'), 'Name' (set to 'myhotelsystem.azurewebsites.net'), 'Publish' (set to 'Code'), 'Runtime stack' (set to '.NET 8 (LTS)'), 'Operating System' (set to 'Windows'), 'Region' (set to 'Canada Central'), and 'Pricing plan' (set to 'Standard S1'). At the bottom, there are 'Review + create' and 'Next : Database >' buttons.

### 4. Click Review + Create, then Create once validation completes.

## Step 2: Build and Deploy the Web App

1. Open Azure Cloud Shell ( top bar in Home page , ‘>\_’ icon ).

The screenshot shows the Microsoft Azure portal's home page. At the top, there's a navigation bar with 'Microsoft Azure', a search bar, and various icons like Copilot, Shell, and settings. Below the header, there's a section for 'Azure services' with links for 'Create a resource', 'App Services', 'Resource groups', 'All resources', 'Subscriptions', 'Virtual machines', 'Quickstart Center', 'Azure AI Foundry', 'Kubernetes services', and 'More services'. Under 'Resources', it shows 'Recent' and 'Favorite' items: 'MyHotelSystem' (App Service) and 'mslearn-scale' (Resource group), both viewed 'a few seconds ago'. There's also a 'See all' link. In the 'Navigate' section, there are links for 'Subscriptions', 'Resource groups', 'All resources', and 'Dashboard'. The 'Tools' section includes links for 'Microsoft Learn', 'Azure Monitor', 'Microsoft Defender for Cloud', and 'Cost Management'. The 'Useful links' section has links for 'Technical Documentation', 'Azure Services', 'Recent Azure Updates', and 'Azure mobile app' download links for the App Store and Google Play. At the bottom left, there's a URL 'https://portal.azure.com/#'.

2. In the Cloud Shell, clone the Sample Project Repository.

```
nur [ ~ ]$ git clone https://github.com/MicrosoftDocs/mslearn-hotel-reservation-system.git
```

3. Navigate to the Source Folder.

```
nur [ ~ ]$ cd mslearn-hotel-reservation-system/src  
nur [ ~/mslearn-hotel-reservation-system/src ]$ ls
```

4. Build the Project.

```
nur [ ~/mslearn-hotel-reservation-system/src ]$ dotnet build
```

5. Publish the Web App.

```
nur [ ~/mslearn-hotel-reservation-system/src ]$ cd HotelReservationSystem  
dotnet publish -o website
```

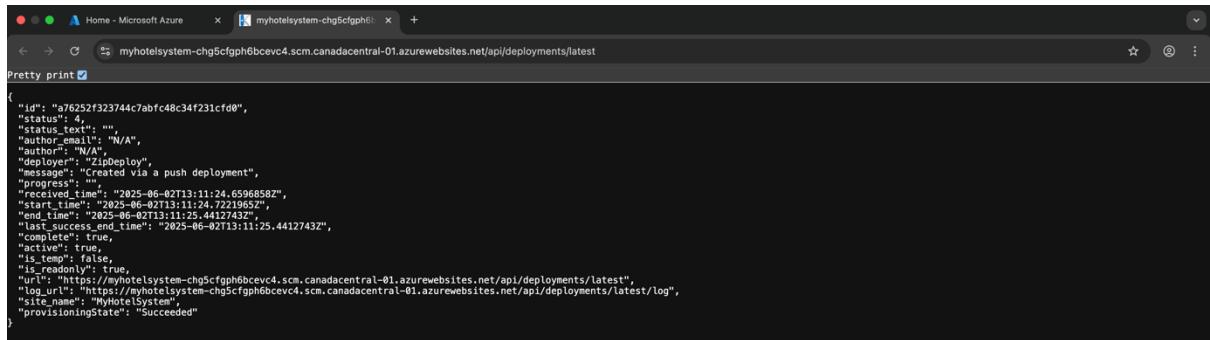
6. Zip and Deploy the Files to Azure.

```
nur [ ~/mslearn-hotel-reservation-system/src/HotelReservationSystem ]$ cd website  
zip website.zip *  
az webapp deployment source config-zip --src website.zip --name myhotelsystem --resource-group mslearn-scale
```

## 7. Test the Web App

In your browser, go to :

<https://myhotelsystem-chg5cfgph6bcevc4.scm.canadacentral01.azurewebsites.net/api/deployments/latest>



```
{
  "id": "a76252f323744c7abfc48c34f231cf0",
  "status": "Succeeded",
  "status_text": "",
  "author_email": "N/A",
  "author": "N/A",
  "deployer": "ZipDeploy",
  "was_created_via": "a push deployment",
  "progress": "",
  "received_time": "2025-06-02T13:11:24.6596858Z",
  "start_time": "2025-06-02T13:11:24.7221965Z",
  "end_time": "2025-06-02T13:11:25.4412743Z",
  "last_success_end_time": "2025-06-02T13:11:25.4412743Z",
  "complete": true,
  "active": true,
  "is_temp": false,
  "is_sticky": true,
  "url": "https://myhotelsystem-chg5cfgph6bcevc4.scm.canadacentral-01.azurewebsites.net/api/deployments/latest",
  "log_url": "https://myhotelsystem-chg5cfgph6bcevc4.scm.canadacentral-01.azurewebsites.net/api/deployments/latest/log",
  "site_name": "MyHotelSystem",
  "provisioningState": "Succeeded"
}
```

### Step 3: Monitor Performance Before Scaling

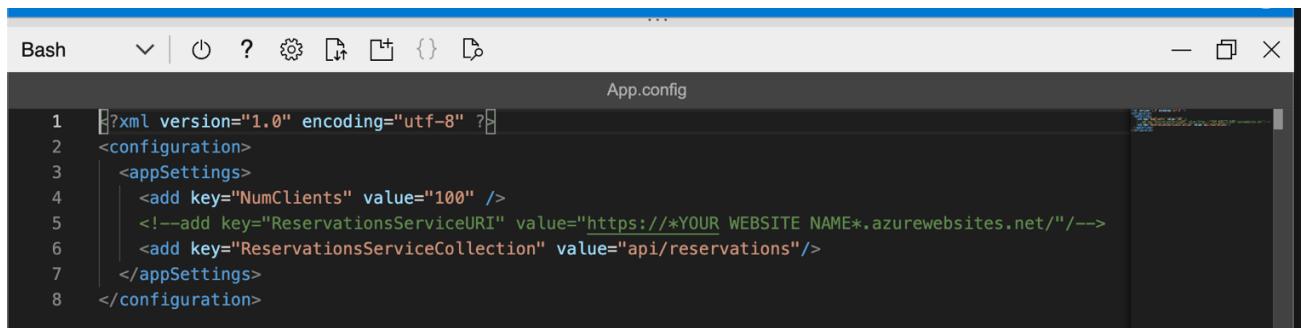
1. In Cloud Shell, Navigate to the Test Client Folder.

```
nur [ ~ ]$ cd ~/mslearn-hotel-reservation-system/src/HotelReservationSystemTestClient
```

2. Edit the Configuration File.

```
nur [ ~ ]$ code App.config
```

- Replace \*YOUR WEBSITE NAME\* with your web app name:



```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <appSettings>
    <add key="NumClients" value="100" />
    <!--add key="ReservationsServiceURI" value="https://*YOUR WEBSITE NAME*.azurewebsites.net/" /-->
    <add key="ReservationsServiceCollection" value="api/reservations"/>
  </appSettings>
</configuration>
```

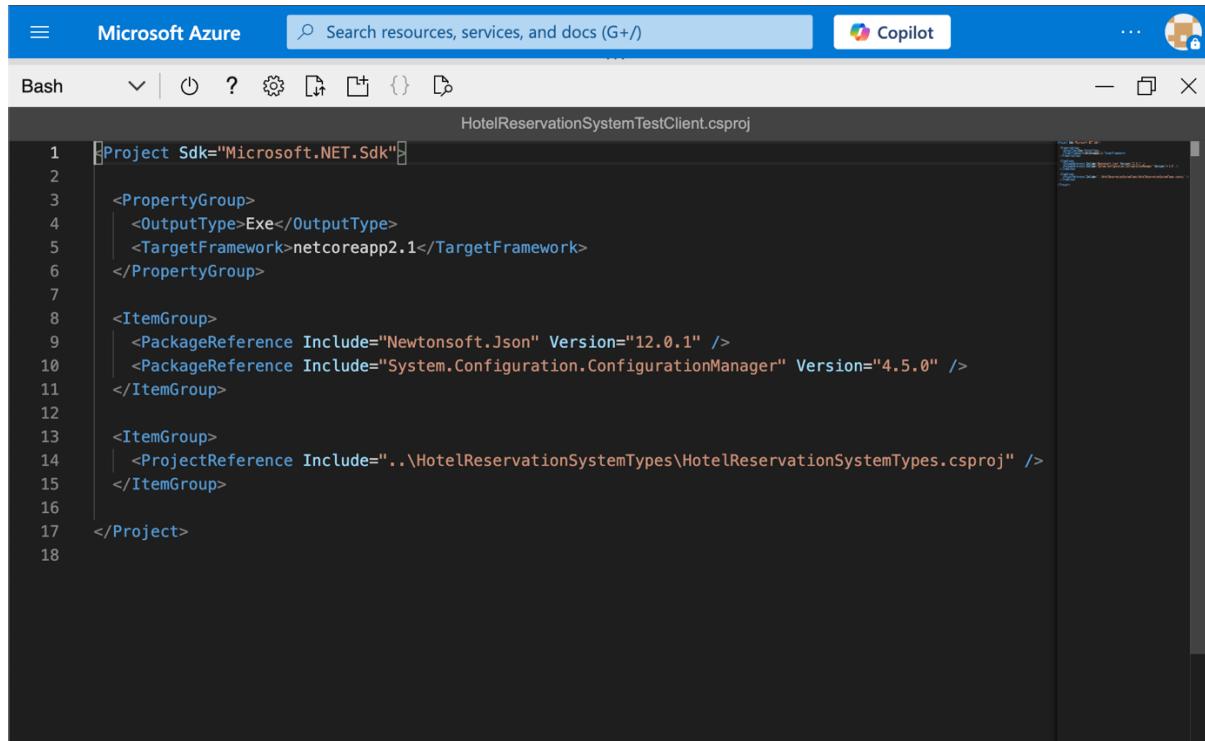
3. Save and Close the Editor.
4. Rebuild the Test Client App.

```
nur [ ~ ]$ dotnet build
```

## 5. Update the Target Framework.

```
nur [ ~/mslearn-hotel-reservation-system/src/HotelReservationSystemTestClient ]$ code HotelReservationSystemTestClient.csproj
```

- Replace the netcore to 8.0.



```
1 <Project Sdk="Microsoft.NET.Sdk">
2
3   <PropertyGroup>
4     <OutputType>Exe</OutputType>
5     <TargetFramework>netcoreapp2.1</TargetFramework>
6   </PropertyGroup>
7
8   <ItemGroup>
9     <PackageReference Include="Newtonsoft.Json" Version="12.0.1" />
10    <PackageReference Include="System.Configuration.ConfigurationManager" Version="4.5.0" />
11  </ItemGroup>
12
13  <ItemGroup>
14    <ProjectReference Include="..\HotelReservationSystemTypes\HotelReservationSystemTypes.csproj" />
15  </ItemGroup>
16
17</Project>
18
```

## 6. Run the Load Test.

```
nur [ ~/mslearn-hotel-reservation-system/src/HotelReservationSystemTestClient ]$ dotnet run
```

## 7. Open Metrics in Azure Portal

- Go to your web app → Monitoring → Metrics

The screenshot shows the Azure Portal's Metrics blade for the 'MyHotelSystem' web app. The left sidebar lists various monitoring and diagnostic tools. The main area displays a line chart titled 'MyHotelSystem | Metrics'. The chart configuration panel includes fields for Scope (set to 'MyHotelSystem'), Metric Namespace (set to 'App Service standard...'), Metric (set to 'Select metric'), and Aggregation (set to 'Select aggregation'). Below the chart, there are three interactive cards: 'Filter & Split', 'Plot multiple metrics', and 'Build custom dashboards'. The chart itself shows data points from Monday 02 at 6 AM to 6 PM, with UTC+08:00 indicated.

- Change the time range to Last 30 minutes and click Apply

The screenshot shows the same Metrics blade for 'MyHotelSystem', but with the time range changed to 'Last 30 minutes'. A modal dialog box is open in the top right corner, allowing the user to change the time range and time granularity. The 'Time range' section has 'Last 30 minutes' selected. The 'Time granularity' section has 'Automatic' selected. Other time range options like 'Last 48 hours', 'Last 24 hours', and 'Last hour' are also listed. The 'Apply' button is highlighted in blue.

## 8. Add Metrics to Chart

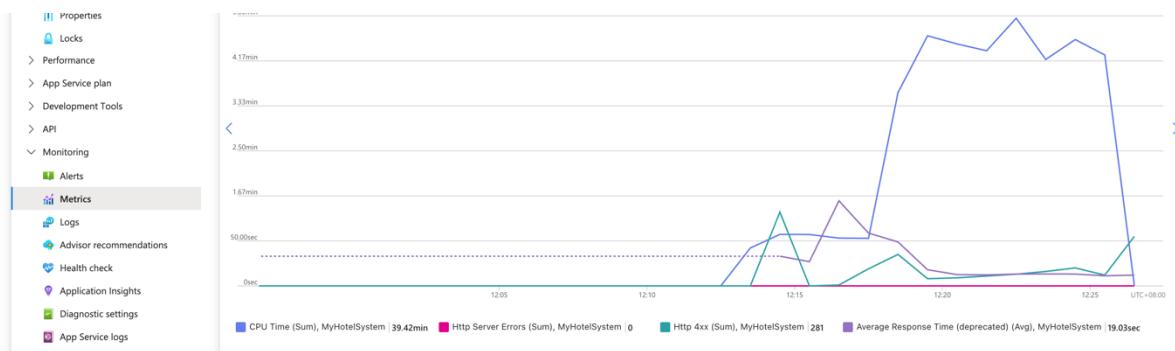
- CPU Time (Aggregation: Sum)
- Http Server Errors (Sum)
- Http 4xx (Sum)
- Average Response Time (Avg)

The screenshot shows the Microsoft Azure Metrics interface for the 'MyHotelSystem' app service. The left sidebar lists other services like 'AlpineSkiHouse-maya' and 'MyHotelSystem'. The main area shows a chart titled 'Sum CPU Time, Sum Http Server Errors, and 2 other metrics for MyHotelSystem'. The chart displays four data series: 'CPU Time (Sum)', 'Http Server Errors (Sum)', 'Http 4xx (Sum)', and 'Average Response Time (Avg)'. The 'CPU Time (Sum)' series has a prominent peak at approximately 11:28. The 'Average Response Time (Avg)' series shows a smaller peak at approximately 11:32.

## 9. Pin the Chart to Your Dashboard

The screenshot shows the 'Pin to dashboard' dialog box. On the left is a preview of the chart from the previous step. On the right, under 'Existing', the 'Private' radio button is selected. Under 'Dashboard', the dropdown menu shows 'My Dashboard'. At the bottom right of the dialog are 'Pin' and 'Cancel' buttons.

## 10. Observe the Metrics



11. Leave the client app running while you perform the next task.

## Step 4: Scale Out the Web App

### 1. Go to App Service > Scale-out (App Service Plan)

The screenshot shows the Azure App Services - Scale out (App Service plan) settings page for the MyHotelSystem web app. The 'Scale' tab is selected. Under 'Settings', the 'Scale out (App Service plan)' section is active, showing a Standard S1 pricing plan. The 'Scaling' section provides information about scaling features and methods. The 'Scaling' method is set to 'Manual', with a current instance count of 1. The 'Instance count' slider is set to 1. At the bottom, there are 'Save' and 'Discard' buttons.

### 2. Set Instance Count

- Click Configure
- Set Instance count to 5
- Click Save

A screenshot of a configuration dialog for setting the instance count. It shows a slider labeled 'Instance count' with a value of 5. Below the slider are 'Save' and 'Discard' buttons. At the bottom, there is a note: 'Showing 1 - 2 of 2. Display auto count: Add or remove favorites by pressing Cmd+Shift+F'.

3. Switch back to the Cloud Shell running the client app. You should observe a decrease in failed requests.
4. Allow the client app to run for another five minutes.
5. Go to the metrics chart for your app on the dashboard in the Azure portal. You should notice a significant increase in CPU time, a drop in average response time, and a decrease in HTTP 4xx errors.
6. Return to the Cloud Shell that's running the client app. Select Enter to stop the app.
7. In the Azure portal, set the instance count back to 1. Select your web app, and in the menu, select Scale out (App Service plan). On the Configure tab, set the instance count to 1, and then select Save in the menu bar at the top of the App Service pane.

