**The laboratory work 5**

**Create an ASP.NET Core web app**

Create an ASP.NET Core web app by following these steps:

1. Open Visual Studio and then select **Create a new project**.
2. In **Create a new project**, find and choose **ASP.NET Core Web Application** for C#, then select **Next**.
3. In **Configure your new project**, name the application *myFirstAzureWebApp*, and then select **Create**.



1. For this quickstart, choose the **Web Application** template. Make sure authentication is set to **No Authentication** and no other option is selected. Select **Create**.



You can deploy any type of ASP.NET Core web app to Azure.

1. From the Visual Studio menu, select **Debug** > **Start Without Debugging** to run the web app locally.



**Publish your web app**

1. In **Solution Explorer**, right-click the **myFirstAzureWebApp** project and select **Publish**.
2. Choose **App Service** and then select **Publish**.



1. In **App Service Create new**, your options depend on whether you're signed in to Azure already and whether you have a Visual Studio account linked to an Azure account. Select either **Add an account** or **Sign in** to sign in to your Azure subscription. If you're already signed in, select the account you want.

Note

If you're already signed in, don't select **Create** yet.



A [resource group](https://docs.microsoft.com/en-us/azure/azure-resource-manager/management/overview#terminology) is a logical container into which Azure resources like web apps, databases, and storage accounts are deployed and managed. For example, you can choose to delete the entire resource group in one simple step later.

1. For **Resource group**, select **New**.
2. In **New resource group name**, enter *myResourceGroup* and select **OK**.

An [App Service plan](https://docs.microsoft.com/en-us/azure/app-service/overview-hosting-plans) specifies the location, size, and features of the web server farm that hosts your app. You can save money when hosting multiple apps by configuring the web apps to share a single App Service plan.

App Service plans define:

* + Region (for example: North Europe, East US, or Southeast Asia)
	+ Instance size (small, medium, or large)
	+ Scale count (1 to 20 instances)
	+ SKU (Free, Shared, Basic, Standard, or Premium)
1. For the **Hosting Plan**, select **New**.
2. In the **Configure Hosting Plan** dialog, enter the values from the following table, and then select **OK**.

|  |  |  |
| --- | --- | --- |
| **Setting** | **Suggested Value** | **Description** |
| App Service Plan | myAppServicePlan | Name of the App Service plan. |
| Location | West Europe | The datacenter where the web app is hosted. |
| Size | Free | [Pricing tier](https://azure.microsoft.com/pricing/details/app-service/?ref=microsoft.com&utm_source=microsoft.com&utm_medium=docs&utm_campaign=visualstudio) determines hosting features. |



1. In **Name**, enter a unique app name that includes only the valid characters are a-z, A-Z, 0-9, and -. You can accept the automatically generated unique name. The URL of the web app is http://<app\_name>.azurewebsites.net, where <app\_name> is your app name.



1. Select **Create** to start creating the Azure resources.

Once the wizard completes, it publishes the ASP.NET Core web app to Azure, and then launches the app in the default browser.



The app name specified in the **App Service Create new** page is used as the URL prefix in the format http://<app\_name>.azurewebsites.net.

**Congratulations!** Your ASP.NET Core web app is running live in Azure App Service.

**Update the app and redeploy**

1. In **Solution Explorer**, under your project, open **Pages** > **Index.cshtml**.
2. Replace the two <div> tags with the following code:

HTML

1. <div class="jumbotron">
2. <h1>ASP.NET in Azure!</h1>
3. <p class="lead">This is a simple app that we’ve built that demonstrates how to deploy a .NET app to Azure App Service.</p>
4. </div>
5. To redeploy to Azure, right-click the **myFirstAzureWebApp** project in **Solution Explorer** and select **Publish**.
6. In the **Publish** summary page, select **Publish**.



When publishing completes, Visual Studio launches a browser to the URL of the web app.



**Manage the Azure app**

To manage the web app, go to the [Azure portal](https://portal.azure.com), and search for and select **App Services**.



On the **App Services** page, select the name of your web app.



You see your web app's Overview page. Here, you can do basic management like browse, stop, start, restart, and delete.



The left menu provides different pages for configuring your app.

**Clean up resources**

In the preceding steps, you created Azure resources in a resource group. If you don't expect to need these resources in the future, you can delete them by deleting the resource group.

From the Azure portal menu or **Home** page, select **Resource groups**, and on the **Resource groups** page, select **myResourceGroup**.

On the **myResourceGroup** page, make sure that the listed resources are the ones you want to delete.

Select **Delete**, type **myResourceGroup** in the text box, and then select **Delete**.