* 10 minutes

Next, let’s use the Azure CLI to create a resource group, and then to deploy a web app into this resource group.

### Using a resource group

When you’re working with your own machine and Azure subscription, you’ll need to first sign in to Azure using the az login command. However, signing in is unnecessary when you are using the browser-based Cloud Shell environment.

Next, you would normally create a resource group for all your related Azure resources with an az group create command, but for this exercise the following resource group has been created for you: **[sandbox resource group name]**.

1. Your first step in this exercise will be to create several variables that you will use in later commands.

* export RESOURCE\_GROUP=[sandbox resource group name]  
  export AZURE\_REGION=centralus  
  export AZURE\_APP\_PLAN=popupappplan-$RANDOM  
  export AZURE\_WEB\_APP=popupwebapp-$RANDOM

1. You can ask the Azure CLI to list all your resource groups in a table. There should just be one while you are in the free Azure sandbox.

* az group list --output table
* Tip
* You can use the **Copy** button to copy commands to the clipboard. To paste, right-click on a new line in the Cloud Shell window and select **Paste**, or use the Shift+Insert keyboard shortcut (⌘+V on macOS).

1. As you do more Azure development, you can end up with several resource groups. If you have several items in the group list, you can filter the return values by adding a --query option. Try the following command:

* az group list --query "[?name == '$RESOURCE\_GROUP']"
* The query is formatted using **JMESPath**, which is a standard query language for JSON requests. You can learn more about this powerful filter language at <http://jmespath.org/>. We also cover queries in more depth in the **Manage VMs with the Azure CLI** module.

### Steps to create a service plan

When you run Web Apps using the Azure App Service, you pay for the Azure compute resources that are used by the app, and the resource costs depend on the App Service plan associated with your Web Apps. Service plans determine the region used for the app datacenter, number of VMs used, and pricing tier.

1. Create an App Service plan to run your app. The following command specifies the free pricing tier, but you can run az appservice plan create --help to see the other pricing tiers.

* Note
* The name of the app and plan must be *unique* in all of Azure. The variables that you created earlier will assign random values as suffixes to make sure they’re unique. However, if you receive an error when you are creating any resources, you should run the commands listed earlier to reset all of the variables with new random values.
* If you receive an error about the resource group, run the commands listed earlier with a different resource group value.
* az appservice plan create --name $AZURE\_APP\_PLAN --resource-group $RESOURCE\_GROUP --location $AZURE\_REGION --sku FREE
* This command can take several minutes to complete.

1. Verify that the service plan was created successfully by listing all your plans in a table.

* az appservice plan list --output table
* You’ll see a response like the following example.
* Kind Location MaximumNumberOfWorkers Name NumberOfSites ResourceGroup Status  
  ------ ---------- ------------------------ ------------------ --------------- ------------------------------------------ --------  
  app Central US 3 popupappplan-54321 0 Learn-12345678-1234-1234-1234-123456789abc Ready

### Steps to create a web app

Next, you’ll create the web app in your service plan. You can deploy the code at the same time, but for our example, we’ll create the web app and deploy the code as separate steps.

1. To create the web app, you’ll supply web app name and the name of the app plan you created above. Just like the app plan name, the web app name must be unique, and the variables that you created earlier will assign random values that should be sufficient for this exercise.

* az webapp create --name $AZURE\_WEB\_APP --resource-group $RESOURCE\_GROUP --plan $AZURE\_APP\_PLAN

1. Verify that the app was created successfully by listing all your apps in a table.

* az webapp list --output table
* You’ll see a response like the following example.
* Name Location State ResourceGroup DefaultHostName AppServicePlan  
  ----------------- ---------- ------- ------------------------------------------ ----------------------------------- ------------------  
  popupwebapp-12345 Central US Running Learn-12345678-1234-1234-1234-123456789abc popupwebapp-12345.azurewebsites.net popupappplan-54321
* Make a note of the **DefaultHostName** listed in the table; this address is the URL for the new website. Azure will make your website available through the unique app name in the azurewebsites.net domain. For example, if my app name was “popupwebapp-mslearn123”, then my website URL would be: http://popupwebapp-mslearn123.azurewebsites.net.

1. Your site has a “quickstart” page created by Azure that you can see either in a browser, or with CURL, just use the **DefaultHostName**:

* curl $AZURE\_WEB\_APP.azurewebsites.net
* You’ll see the default HTML for the sample app returned.

### Steps to deploy code from GitHub

1. The final step is to deploy code from a GitHub repository to the web app. Let’s use a simple PHP page available in the Azure Samples GitHub repository that displays “Hello World!” when it executes. Make sure to use the web app name you created.

* az webapp deployment source config --name $AZURE\_WEB\_APP --resource-group $RESOURCE\_GROUP --repo-url "https://github.com/Azure-Samples/php-docs-hello-world" --branch master --manual-integration

1. Once it’s deployed, hit your site again with a browser or CURL.

* curl $AZURE\_WEB\_APP.azurewebsites.net
* The page displays “Hello World!”
* Hello World!

This exercise demonstrated a typical pattern for an interactive Azure CLI session. You first used a standard command to create a new resource group. You then used a set of commands to deploy a resource (in this example, a web app) into this resource group. This set of commands could easily be combined into a shell script, and executed every time you need to create the same resource.

Need help? See our [troubleshooting guide](https://docs.microsoft.com/en-us/learn/support/troubleshooting?uid=learn.control-azure-services-with-cli.5-exercise-create-website-using-the-cli&documentId=d333a4fc-657a-9cf6-d02d-69e5a41783a5&versionIndependentDocumentId=66655574-4b27-61d2-2e7c-97b97836449e&contentPath=%2FMicrosoftDocs%2Flearn-pr%2Fblob%2Flive%2Flearn-pr%2Fazure%2Fcontrol-azure-services-with-cli%2F5-exercise-create-website-using-the-cli.yml&url=https%3A%2F%2Fdocs.microsoft.com%2Fen-us%2Flearn%2Fmodules%2Fcontrol-azure-services-with-cli%2F5-exercise-create-website-using-the-cli&author=dbradish) or provide specific feedback by [reporting an issue](https://docs.microsoft.com/en-us/learn/support/troubleshooting?uid=learn.control-azure-services-with-cli.5-exercise-create-website-using-the-cli&documentId=d333a4fc-657a-9cf6-d02d-69e5a41783a5&versionIndependentDocumentId=66655574-4b27-61d2-2e7c-97b97836449e&contentPath=%2FMicrosoftDocs%2Flearn-pr%2Fblob%2Flive%2Flearn-pr%2Fazure%2Fcontrol-azure-services-with-cli%2F5-exercise-create-website-using-the-cli.yml&url=https%3A%2F%2Fdocs.microsoft.com%2Fen-us%2Flearn%2Fmodules%2Fcontrol-azure-services-with-cli%2F5-exercise-create-website-using-the-cli&author=dbradish#report-feedback).