Lab 09a - Implement Web Apps

Objectives

In this lab, we will:

* Task 1: Create an Azure web app
* Task 2: Create a staging deployment slot
* Task 3: Configure web app deployment settings
* Task 4: Deploy code to the staging deployment slot
* Task 5: Swap the staging slots
* Task 6: Configure and test autoscaling of the Azure web app

## Task 1: Create an Azure web app

In this task, we will create an Azure web app.

In the Azure portal we go to App services, and on the App services blade we click + Create. On the Basics tab of Create Web App we specify the following settings:

| Setting | Value |
| --- | --- |
| Subscription | the name of the Azure subscription you are using in this lab |
| Resource group | the name of a new resource group **az104-09a-rg1** |
| Web app name | any globally unique name |
| Publish | **Code** |
| Runtime stack | **PHP 8.0** |
| Operating system | **Linux** |
| Region | the name of an Azure region where you can provision Azure web apps |
| App service plan | accept the default configuration |

After that we click on Review + create, ensure that the validation passed and click Create. When the deployment is done, we click Go to resource.

## Task 2: Create a staging deployment slot

In this task, we will create a staging deployment slot.

On the blade of the newly deployed web app, we click the URL link to display the default web page in a new browser tab. The page looks like this:

Graphical user interface, text, application, chat or text message

Description automatically generated

We close the new browser tab, and we go back in the Azure portal. In the Deployment section of the web app blade, we click Deployment slots. In Deployment slots we add a new slot with the following settings:

| Setting | Value |
| --- | --- |
| Name | **staging** |
| Clone settings from | **Do not clone settings** |

Back on the Deployment slots blade of the web app, we click the entry representing the newly created staging slot. We notice that its URL differs from the one assigned to the production slot - academydevops-staging.azurewebsites.net

## Task 3: Configure web app deployment settings

In this task, we will configure web app deployment settings.

On the staging deployment slot blade, in the Deployment section, we click Deployment Center and then select the Settings tab. There, in the Source drop-down list we select Local Git and save it. On the Deployment Center blade, we copy the Git Clone Url: https://academydevops-staging.scm.azurewebsites.net:443/academydevops.git

In the Deployment Center blade, we select the Local Git/FTPS credentials tab. In the User Scope section, we specify the following settings and then we save it.

| Setting | Value |
| --- | --- |
| Username | devopsacademy (must not contain @ character) |
| Password | any password that satisfies complexity requirements |

We will need these credentials later.

## Task 4: Deploy code to the staging deployment slot

In this task, we will deploy code to the staging deployment slot.

We open the Azure Cloud Shell, and run the following commands:

* To clone the remote repository containing the code for the web app:

git clone <https://github.com/Azure-Samples/php-docs-hello-world>

* To set the current location to the newly created clone of the local repository containing the sample web app code:

Set-Location -Path $HOME/php-docs-hello-world/

* To add the remote git:

git remote add devopsacademy <https://academydevops-staging.scm.azurewebsites.net:443/academydevops.git>

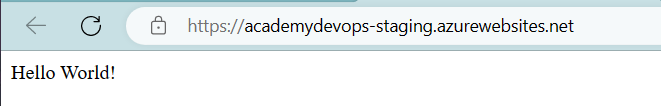
* To push the sample web app code from the local repository to the Azure web app staging deployment slot:

git push devopsacademy master

We type the deployment user name and password.

When we are done with the commands, we close the Cloud Shell.

On the staging slot blade, we click Overview and then we click the URL link to display the default web page in a new browser tab. We verify that the browser page displays the Hello World! message and close the new tab.



## Task 5: Swap the staging slots

In this task, we will swap the staging slot with the production slot.

In the Deployment section, we click Deployment slots and then we click Swap. We review the settings and click Swap. From Overview on the production slot blade of the web app we then click the URL link to display the web site home page in a new browser tab. We can see that the page has been replaced.

Graphical user interface, application, Word

Description automatically generated

## Task 6: Configure and test autoscaling of the Azure web app

In this task, you will configure and test autoscaling of Azure web app.

On the blade displaying the production slot of the web app, in the Settings section, we click Scale out (App Service plan) and Custom autoscale. We select Scale based on a metric and click + Add a rule. On the Scale rule blade, we specify the following settings and then click Add:

| Setting | Value |
| --- | --- |
| Metric source | **Current resource** |
| Metric namespace | **standard metrics** |
| Metric name | **CPU Percentage** |
| Operator | **Greater than** |
| Metric threshold to trigger scale action | **10** |
| Duration (in minutes) | **1** |
| Time grain statistic | **Maximum** |
| Time aggregation | **Maximum** |
| Operation | **Increase count by** |
| Instance count | **1** |
| Cool down (minutes) | **5** |

On the App Service plan scaling blade, we specify the following settings and Save it:

| Setting | Value |
| --- | --- |
| Instance limits Minimum | **1** |
| Instance limits Maximum | **2** |
| Instance limits Default | **1** |

Next, we open the Azure Cloud Shell – PowerShell. We run the following code to identify the URL of the Azure web app.

* $rgName = 'az104-09a-rg1'
* $webapp = Get-AzWebApp -ResourceGroupName $rgName

After these we run the following to start and infinite loop that sands HTTP:

* while ($true) { Invoke-WebRequest -Uri $webapp.DefaultHostName }

We minimize the pane, and on the web app blade in settings, we click Scale out (App Service plan). From Run history tab se check Observed resource count. We monitor the number of instaces. If is increased to 2 we close the Cloud shell pane.

Show data for last 
6 hours 12 hours 
Pin to dashboard 
I day 
7 days 
Custom 
Observed resource instance count 
PM 
12:30 PM 
12:45 PM 
2 
Autoscale events for this time range 
View more details in the Activity [ng 
Operation name 
Autoscale scale up initiated 
Status 
Succeeded 
Time 
7 minutes ago 