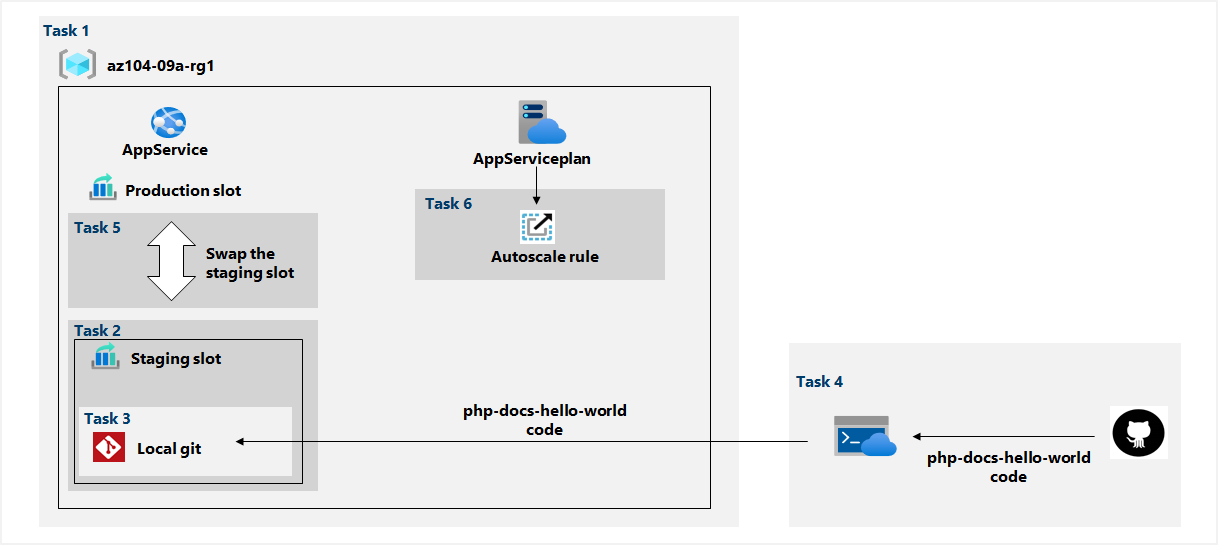
Objectives:

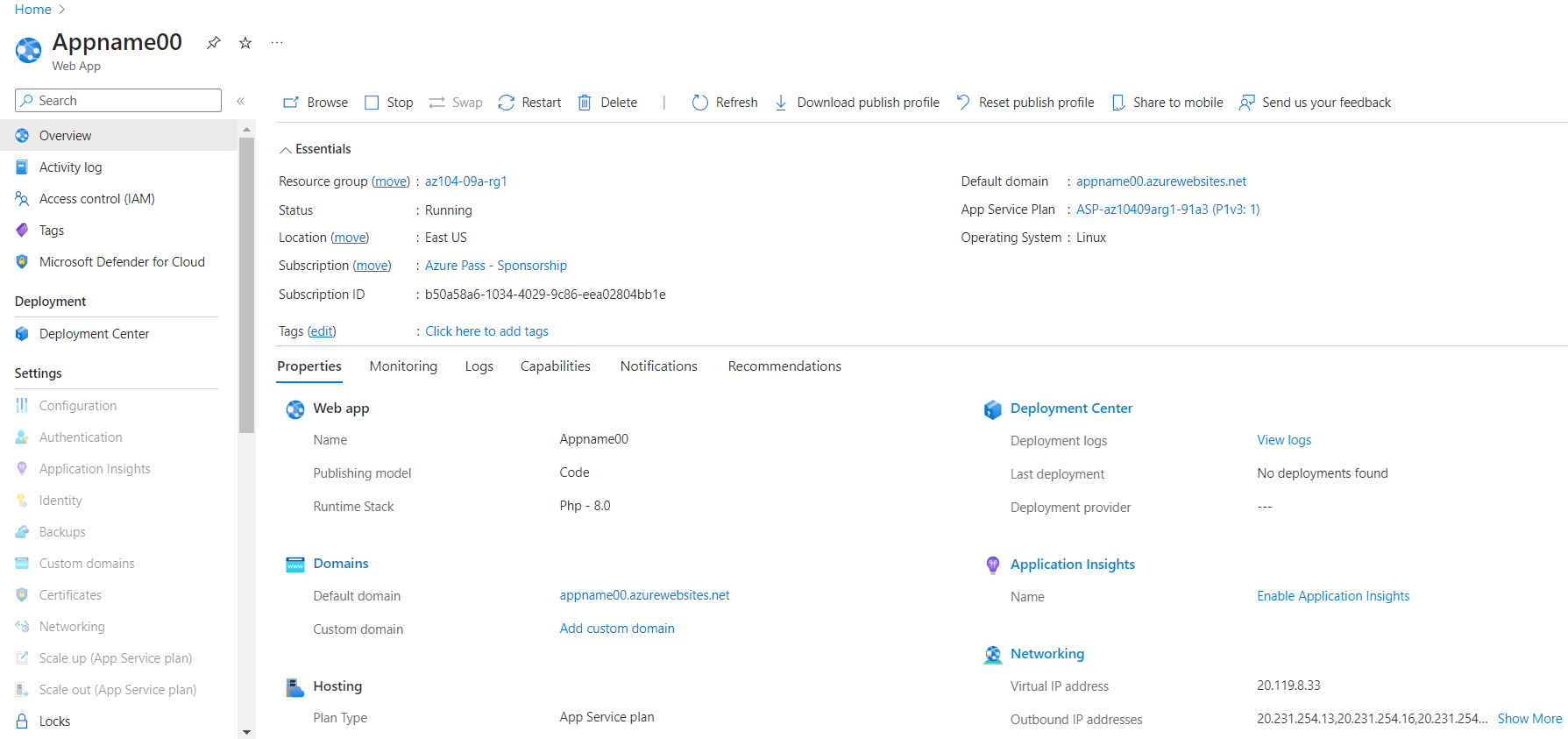
* Task 1: Create an Azure web app
* Task 2: Create a staging deployment slot
* Task 3: Configure web app deployment settings
* Task 4: Deploy code to the staging deployment slot
* Task 5: Swap the staging slots
* Task 6: Configure and test autoscaling of the Azure web app

Architecture diagram:



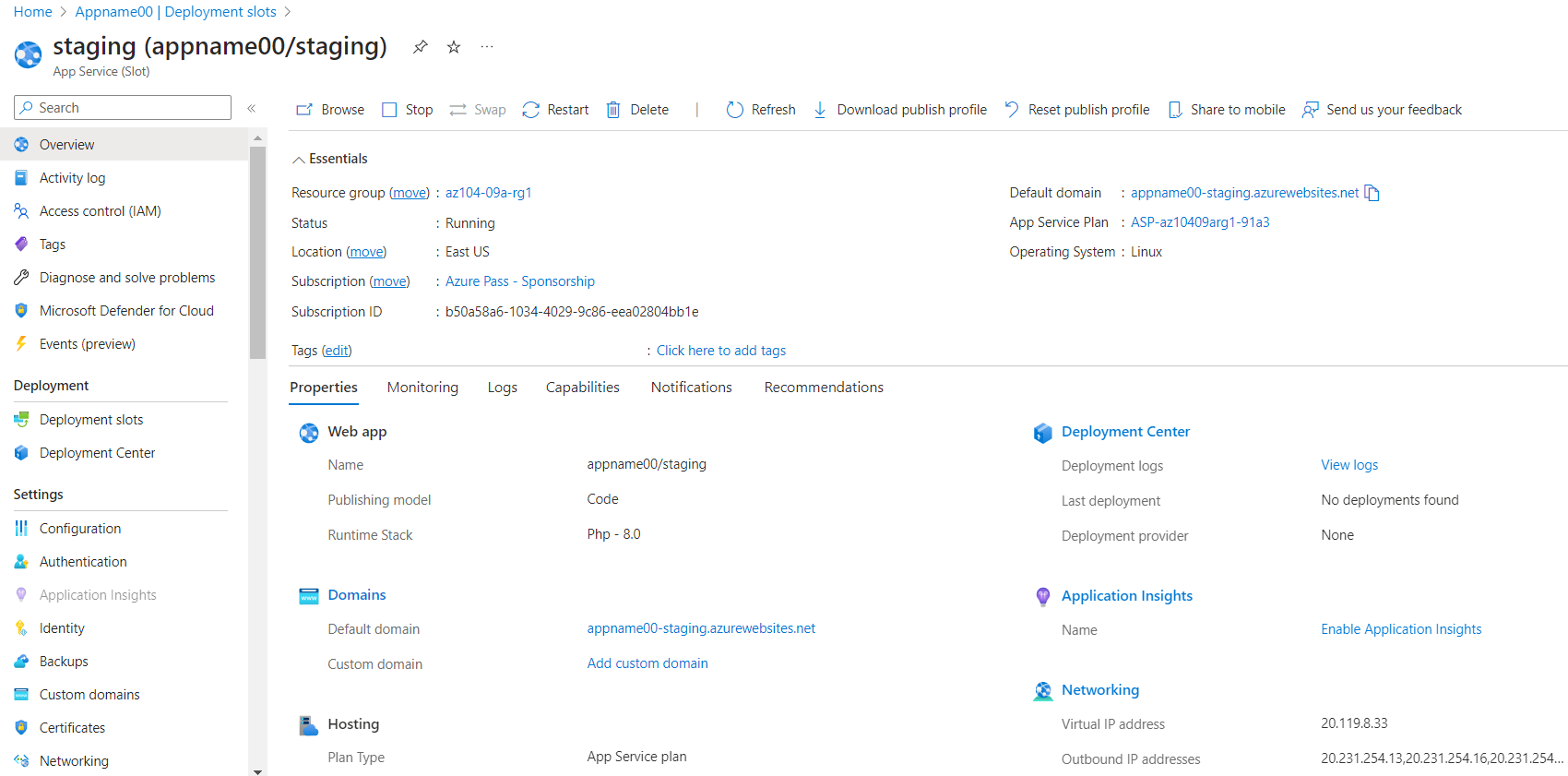
Task 1: Create an Azure web app.

In this task, we will be creating an Azure web app with the required settings:



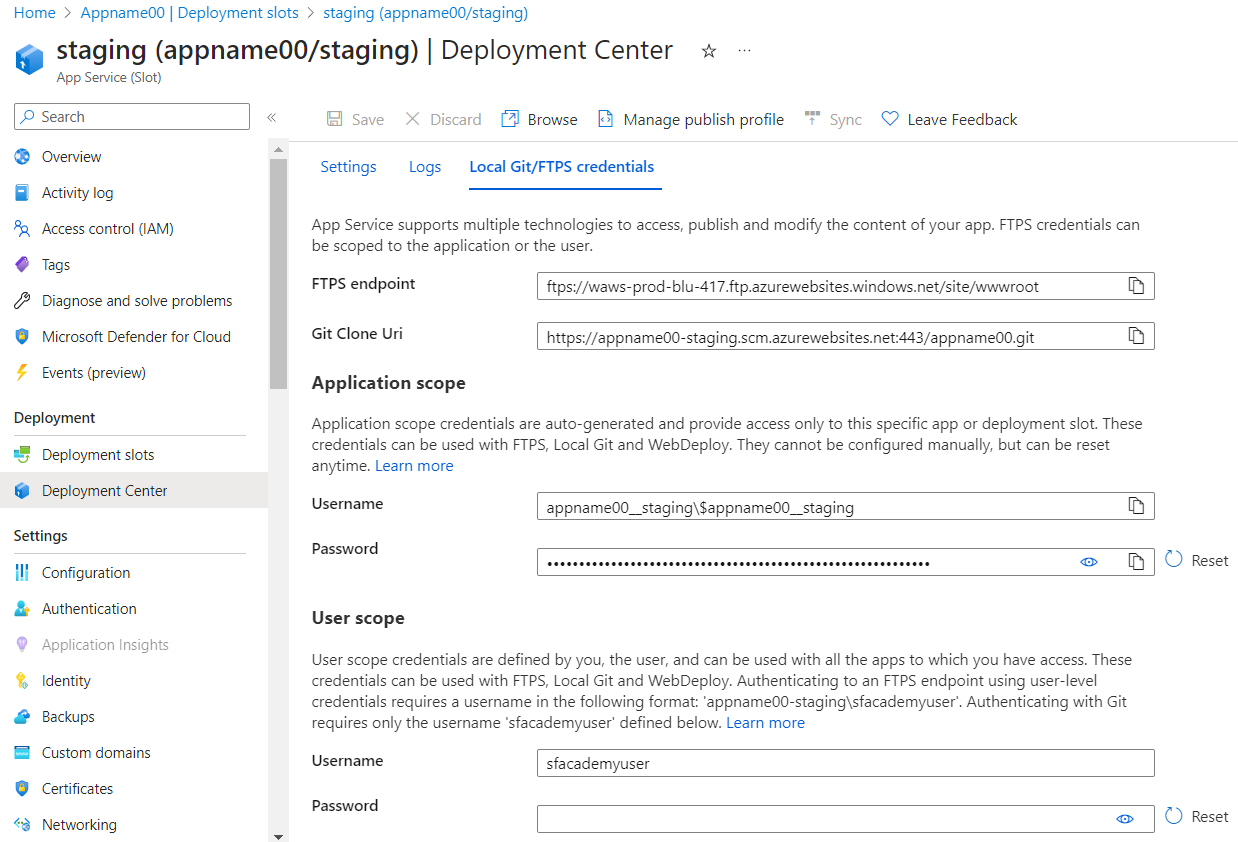
Task 2: Create a staging deployment slot.

In this task, we will be creating a staging deployment slot:



Task 3: Configure web app deployment settings.

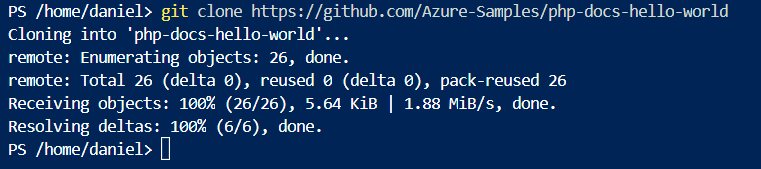
In this task, we will configure web app deployment settings:



Task 4: Deploy code to the staging deployment slot.

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

Here, we will be running some PowerShell commands to first clone the remote repository containing the code for the web app:



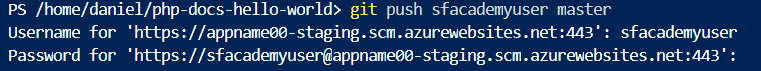
Then we will set the current location to the newly created clone of the local repository containing the sample web app code:



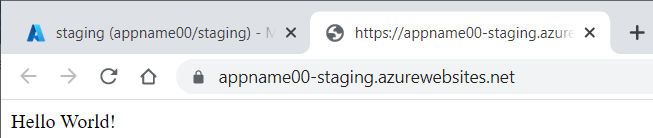
After that we add the remote git using the username and git clone url from the previous task:



Next, we push the sample web app code from the local repository to the Azure web app staging deployment slot:



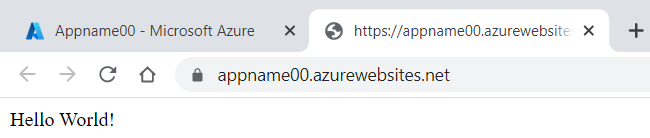
Using the staging slot blade URL, we can see the correct message showing:



Task 5: Swap the staging slots.

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

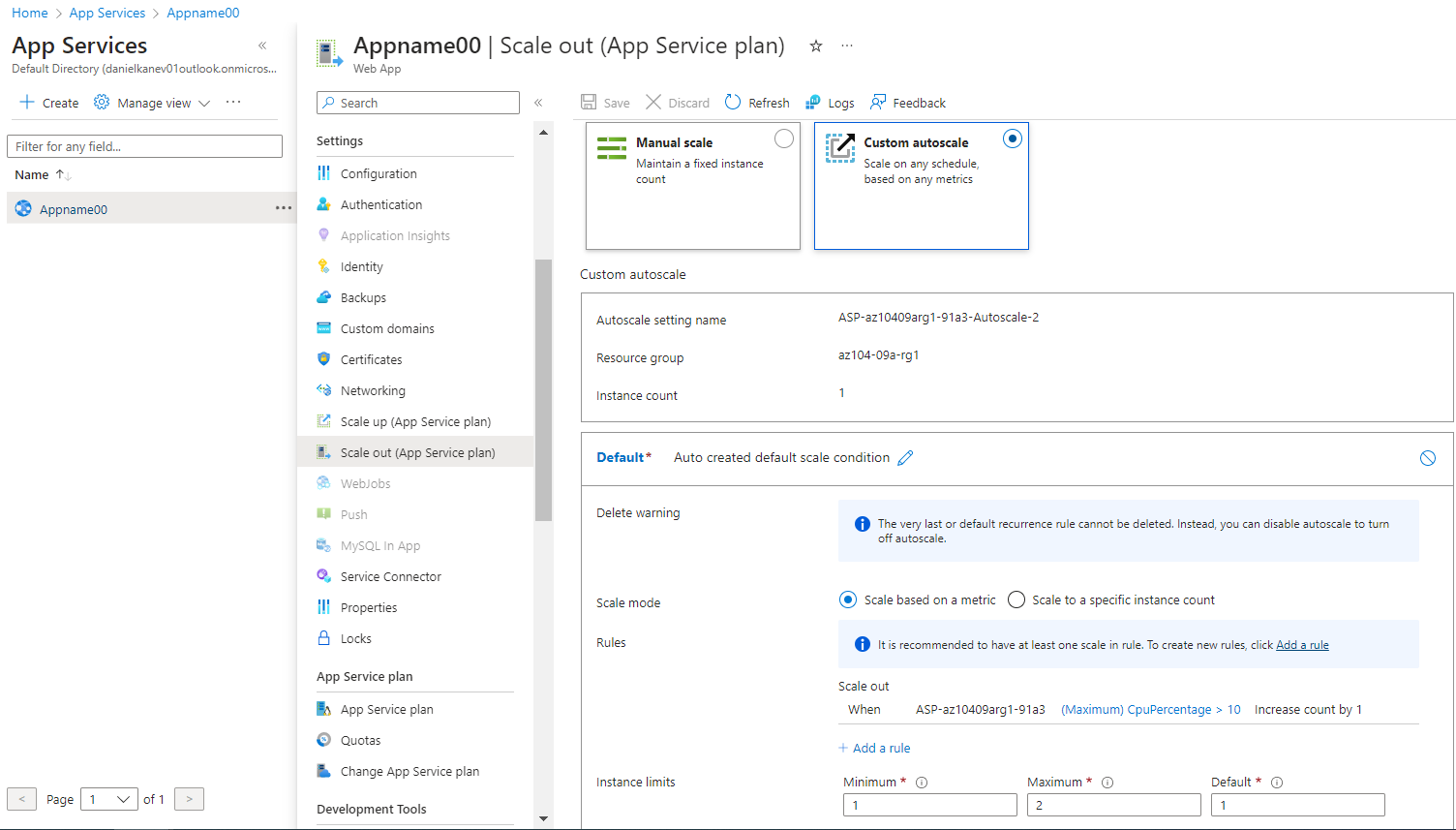
After swapping the staging slot with the production slot, we check the url of the production slot blade and we can see the correct message showing up again:



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

In this task, we will configure and test autoscaling of Azure web app:

First, in the Settings section -> Scale out -> we configure the custom autoscaling setting



After the configuration step above, we run a few shell commands to start the infinite loop that sends the HTTP requests to the web app. I then opened a second shell, ran the infinite loop again there to load the processor a bit. As a result, we can see the instance count has increased to 2:

