## Oefening: een Azure-website maken met de CLI

* 10 minuten

Laten we vervolgens de Azure CLI gebruiken om een resourcegroep te maken en vervolgens een web-app in deze resourcegroep te implementeren.

### Een resourcegroep gebruiken

Wanneer u met uw eigen computer en Azure-abonnement werkt, moet u zich eerst aanmelden bij Azure met de opdracht. Aanmelden is echter niet nodig wanneer u de browsergebaseerde Cloud Shell-omgeving gebruikt.az login

Vervolgens zou u normaal gesproken een resourcegroep maken voor al uw gerelateerde Azure-resources met een opdracht, maar voor deze oefening is de volgende resourcegroep voor u gemaakt: **[naam sandbox-resourcegroep]**.az group create

1. Uw eerste stap in deze oefening is het maken van verschillende variabelen die u in latere opdrachten zult gebruiken.

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

1. U kunt de Azure CLI vragen om al uw resourcegroepen in een tabel weer te geven. Er moet er slechts één zijn terwijl u zich in de gratis Azure-sandbox bevindt.

* az group list --output table
* Fooi
* U kunt de knop **Kopiëren** gebruiken om opdrachten naar het klembord te kopiëren. Als u wilt plakken, klikt u met de rechtermuisknop op een nieuwe regel in het Cloud Shell-venster en selecteert u **Plakken** of gebruikt u de sneltoets (op macOS).Shift+Insert⌘+V

1. Naarmate u meer Azure-ontwikkeling uitvoert, kunt u eindigen met verschillende resourcegroepen. Als u meerdere items in de groepslijst hebt, kunt u de retourwaarden filteren door een optie toe te voegen. Probeer de volgende opdracht:--query

* az group list --query "[?name == '$RESOURCE\_GROUP']"
* De query wordt opgemaakt met **JMESPath**, een standaardquerytaal voor JSON-aanvragen. U kunt meer te weten komen over deze krachtige filtertaal op <http://jmespath.org/>. We behandelen query’s ook uitgebreider in de **VM’s beheren met de Azure CLI-module**.

### Stappen voor het maken van een serviceplan

Wanneer u Web Apps uitvoert met de Azure App Service, betaalt u voor de Azure-rekenresources die door de app worden gebruikt en zijn de resourcekosten afhankelijk van het App Service-abonnement dat aan uw Web Apps is gekoppeld. Serviceplannen bepalen de regio die wordt gebruikt voor het app-datacenter, het aantal gebruikte VM’s en de prijscategorie.

1. Maak een App Service-abonnement om uw app uit te voeren. Met de volgende opdracht wordt de gratis prijscategorie opgegeven, maar u kunt deze uitvoeren om de andere prijsniveaus te bekijken.az appservice plan create --help

* Notitie
* De naam van de app en het abonnement moeten *uniek* zijn in heel Azure. De variabelen die u eerder hebt gemaakt, wijzen willekeurige waarden toe als achtervoegsels om ervoor te zorgen dat ze uniek zijn. Als er echter een foutbericht wordt weergegeven wanneer u bronnen maakt, moet u de eerder vermelde opdrachten uitvoeren om alle variabelen opnieuw in te stellen met nieuwe willekeurige waarden.
* Als er een fout over de resourcegroep wordt weergegeven, voert u de eerder vermelde opdrachten uit met een andere resourcegroepwaarde.
* 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 domain. For example, if my app name was “popupwebapp-mslearn123”, then my website URL would be: .azurewebsites.net``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).