Module 14 - Lab 2: Azure App Services (Optional) - Advanced Settings

Task 1: Create a static HTML web app in Azure CLI

> ❓ Azure Web Apps provides a highly scalable, self-patching web hosting service. This lab shows how to deploy a basic HTML+CSS site to Azure Web Apps. You'll complete this task in Cloud Shell but you can also run these commands locally with Azure CLI.

☐ 1. Navigate to 📄 **portal.azure.com** and login using username 📄 **sheikhnasir3LZ3U@gdcs4.com** and password 📄 **6pz0UfTcC8pgiKPR**.

☐ 2. Open the **Cloud Shell** in **Bash** mode.

☐ 3. Click **Show advanced settings**.

☐ 4. Select the **East US** region. Select **Use existing** Resource group and select the pre-provisioned resource group for the lab.

☐ 5. Enter a name for the storage account (this must be unique) and type 📄 **cloudshell** as the name of the File share then click **Create Storage.**

☐ 6. Your Cloud Shell will now launch.

☐ 7. To install the webapp extension, run the following command:

```
az extension add --name webapp
```

☐ 8. When the extension has been installed, the Cloud Shell shows the following message:

The installed extension 'webapp' is in preview.

☐ 9. Download the sample website. In the Cloud Shell, create a quickstart directory and then change to it using the following commands.

```
mkdir quickstart
```

```
cd quickstart
```

```
Bash                    ⏻  ?  ⚙  ⊡  ⊞  {}  ⊡

Initializing your account for Cloud Shell...\
Requesting a Cloud Shell.Succeeded.
Connecting terminal...

Welcome to Azure Cloud Shell

Type "az" to use Azure CLI
Type "help" to learn about Cloud Shell

gareth_demo1aegkc@Azure:~$ az extension add --name w
The installed extension 'webapp' is in preview.
gareth_demo1aegkc@Azure:~$ mkdir quickstart
gareth_demo1aegkc@Azure:~$ cd quickstart
gareth_demo1aegkc@Azure:~/quickstart$ ▮
```

☐ 10. Next, run the following command to clone the sample app repository to your quickstart directory.

```
git clone https://github.com/Azure-Samples/html-docs-hello-world.git
```

```
Bash                    ⏻  ?  ⚙  ⊡  ⊞  {}  ⊡                              —  ☐

gareth_demo1aegkc@Azure:~/quickstart$ git clone https://github.com/Azure-Samples/html-docs-hello-world.git
Cloning into 'html-docs-hello-world'...
remote: Enumerating objects: 46, done.
remote: Total 46 (delta 0), reused 0 (delta 0), pack-reused 46
Unpacking objects: 100% (46/46), done.
Checking connectivity... done.
gareth_demo1aegkc@Azure:~/quickstart$
```

☐ 11. Change to the directory that contains the sample code and run the az webapp up command.

```
cd html-docs-hello-world
```

☐ 12. Run the following command replacing <app_name> with a unique app name:

```
az webapp up --location eastus --resource-group myRG-UP9MFWS0AE --name <app_name> --html
```

☐ 13. The az webapp up command does the following actions:

- Create a default app service plan.

- Create an app with the specified name.

- [Zip deploy](#) files from the current working directory to the web app.

  This command may take a few minutes to run. While running, it displays information similar to the following example:

  ```
  {
      "app_url": "https://<app_name>.azurewebsites.net",
      "location": "eastus",
      "name": "<app_name>",
      "os": "Windows",
      "resourcegroup": "appsvc_rg_Windows_westeurope",
      "serverfarm": "appsvc_asp_Windows_westeurope",
      "sku": "FREE",
      "src_path": "/home/<username>/quickstart/html-docs-hello-world ",
      < JSON data removed for brevity. >
  }
  ```

☐ 14. Browse to the app. In a browser, go to the Azure web app URL: **http://<app_name>.azurewebsites.net**.

The page is running as an Azure App Service web app.

Azure App Service - Sample Static HTML Site
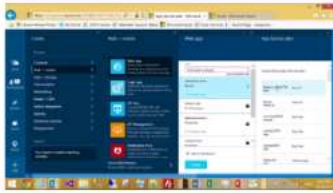
Azure App Service Web Apps

App Service Web Apps is a fully managed compute platform that is optimized for hosting websites and web applications. This platform-as-a-service (PaaS) offering of Microsoft Azure lets you focus on your business logic while Azure takes care of the infrastructure to run and scale your apps.

Azure Content Delivery Network (CDN)

The Azure Content Delivery Network (CDN) caches static web content at strategically placed locations to provide maximum throughput for delivering content to users. The CDN offers developers a global solution for delivering high-bandwidth content by caching the content at physical nodes across the world.

Task 2: Update and redeploy the app

☐ 1. In the Cloud Shell, type the following command and press enter to open the nano text editor.

```
nano index.html
```
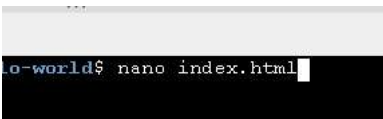
☐ 2. In the `<h1>` heading tag, change "Azure App Service - Sample Static HTML Site" to "Azure App Service", as shown below.
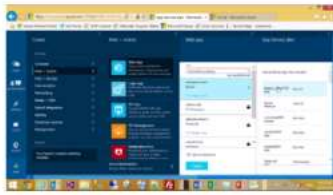
☐ 3. Save your changes Use the command ^O *(Ctrl + O)* then **Enter** and ^X *(Ctrl + X)* to exit nano.

☐ 4. You'll now redeploy the app with the same az webapp up command:

```
az webapp up --location eastus --resource-group myRG-UP9MFWS0AE --name <app_name> --html
```

☐ 5. Once deployment has completed, switch back to the browser window that opened in the **Browse to the app** step, and refresh the page.

Azure App Service

Azure App Service Web Apps

App Service Web Apps is a fully managed compute platform that is optimized for hosting websites and web applications. This platform-as-a-service (PaaS) offering of Microsoft Azure lets you focus on your business logic while Azure takes care of the infrastructure to run and scale your apps.

Azure Content Delivery Network (CDN)

The Azure Content Delivery Network (CDN) caches static web content at strategically placed locations to provide maximum throughput for delivering content to users. The CDN offers developers a global solution for delivering high-bandwidth content by caching the content at physical nodes across the world.

☐   6. Exit the Bash Shell in the Azure portal.

Task 3: Scale up a Web App.

> ❓ In the previous task you automatically created an App Service Plan. However, this was a basic tiwe and required it to be scaled up in order to use some of the additional features in the following exercises. This task will demonstrate how to Scale up an App Service.

☐   1. In the Azure Portal, navigate to **myRG-UP9MFWS0AE** and click on the App Service Plan that was created in the previous task.

☐   2. Click on **Scale up (App Service plan)**.

☐   3. Select the Production, then select **P1V2** tier and click **Apply**.

☐ 4. Wait for the deployment to complete.

Task 4: Add a custom domain and certificate to a Web App.

> ❓ In this task you will add a Public Domain name (Custom Domain) and Public Certificate (SSL) to your web app.

☐ 1. Open **File Explorer** and navigate to the **E:** drive. Note the Certificate file.

☐ 2. In the Azure Portal select your **App Service**.

☐ 3. Click **Custom domains**.

☐ 4. Click **Add custom domain**.

godeployazurelab | Custom domains
App Service

- Identity
- Backups
- Custom domains
- TLS/SSL settings
- Networking
- Scale up (App Service plan)
- Scale out (App Service plan)
- WebJobs
- Push
- MySQL In App

Custom Domains

Configure and manage custom domains assig

IP address: ⓘ
40.71.250.191

Custom Domain Verification ID: ⓘ
9E192D4EC9E2011046041727E8BE2CB8B40B65

HTTPS Only: ⓘ
On

+ Add custom domain

Status Filter

☐ 5. In the lab environment, click the **DNS tab** and identify your **Lab Domain Name** that you have been allocated in the format **labxxxxxx.godeploylabs.com**.

Home   Lab Guide   DNS

Lab Domain Name lab356666.godeploylabs.com

➕ A Records

Name	IPv4 Address
	No A records

➕ AAAA Records

Name	IPv6 Address

☐ 6. In the Hostname field enter your domain name 📖 **unrecognised token ($gd.com(labDomain).fqdn)** and click **Validate**.

Add custom domain ✕
godeployazurelab

Custom domain *
lab356666.godeploylabs.com

Validate

☐ 7. In the Hostname record type select **A record (example.com)**.

Add custom domain ✕
godeployazurelab

Validate

Hostname record type

A record (example.com) ▼

🌐 A record configuration

☐ 8. Scroll down to the bottom of the Add custom domain blade and note the two DNS records that are required to verify Domain ownership.

! Domain ownership

To verify domain ownership create TXT and A records with your
DNS provider using the configuration below. Learn more

Type	Host	Value
TXT	asuid	9E192D4EC9E2011046041727E8BE2CB8B40B657F7B1E
A	@	40.71.250.191

☐   9. Add the TXT and A records to your DNS Tab of the lab environment. *(Note you only need to add the name asuid for the TXT record and nothing for the A record)*

☐ 10. Note the External IP Address (Public IP) and in the DNS tab of the lab environment on the A records click the + sign to add a new record. Do **NOT** enter anything in the name field just enter the **External IP Address** from the portal into the **IPv4 Address** field then click **Save**.

☐ 11. Once the records have been added, click **Validate** at the top of the blade.

☐ 12. Once validated, click **Add custom domain**

Add custom domain   ✕

godeployazurelab

Custom domain *

lab356666.godeploylabs.com   ✓

Validate

Hostname record type

CNAME (www.example.com or any subdomain) ⌄

🌐 **CNAME configuration**

A CNAME record is used to specify that a domain name is an alias
for another domain. In your scenario, that would be mapping
lab356666.godeploylabs.com to custom domain verification id
below. Learn More

Custom Domain Verification ID: ⓘ

9E192D4EC9E2011046041727E8BE2CB8B40B657F7B1E197B7... 📋

CNAME

godeployazurelab.azurewebsites.net

Add custom domain

✅ Hostname availability

✅ Domain ownership

☐ 13. Navigate back to the Azure Portal and click **Add Binding**.

☐ 14. Click **Upload PFX certificate**.

☐ 15. Select the PFX file from the **E:** of the your lab machine entering the password 🖼 **Pa55w.rd** and then click **Upload**.

☐ 16. Select the **Private Certificate .godeploylabs.com** and then select **IP Based SSL type** and click **Add binding**.

17. Wait for the deployment to complete then open another tab on the browser and check to see if the SSL binding has been successful by new **HTTPS://** URL of your web app (ie 📊 **unrecognised token ($gd.com(labDomain).fqdn)**)
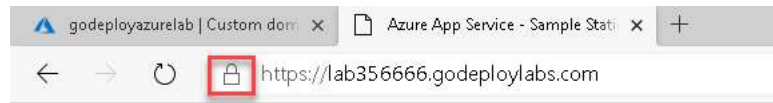
Task 5: So this is PaaS, where are my settings?

❓ When PaaS, such as Web Apps, are deployed you have no access to the underlying Virtual Machine. This is all done using settings in Azure. This task will show you how to configure the different settings for the Web App PaaS service.

1. To configure the settings for the PaaS service click **Configuration** and note all the available options to turn on/off features. Also note the similarities to the IIS Manager on IaaS.

✓ **Results** : You have now completed this Lab.