

**Deploy a Web Application with Terraform**

In this lab, we will deploy an Azure Application Service plan with a .NET Framework site and a Local Git content management setting. To do this, we will be using an Azure CLI and the code provided.

**Before We Begin**

Before we get started, we need to log into the Azure portal using the provided credentials.

**Set Up the Azure CLI**

In the Azure portal, select the Command Line button at the top of the screen. Open the CLI. Here, select Bash when prompted. We then want to choose Show Advanced Settings. Choose the same Cloud Shell region as your lab provisioned Storage Account. Leave both the Resource Group and Storage Account as using the existing resources. In the File share section, enter a name for the account (for this example, we are using cloudcli ) and click the Attach Storage button. Once finished, we can continue.

**Deploy a Web Application**

The first thing we need to do now that we have a storage account is to create a file using the code we've been provided. In this code, replace the "Enter App Service Plan name" and "Enter Web App Service Name" with a unique name (we'll use"newweb-appserviceplan" and "custom-tf-webapp-for-thestudent" for our example). For each instance of "Enter Resource Group Name", use the name of the resource group provided by the lab, which should be preceded by three numbers on the Azure portal's main page. For our video example, this title starts with 191. We will be naming this file lab.tf.

provider "azurerm" {

version = 1.38

}

resource "azurerm\_app\_service\_plan" "svcplan" {

name = "Enter App Service Plan name"

location = "eastus"

resource\_group\_name = "Enter Resource Group Name"

sku {

tier = "Standard"

size = "S1"

}

}

resource "azurerm\_app\_service" "appsvc" {

name = "Enter Web App Service Name"

location = "eastus"

resource\_group\_name = "Enter Resource Group Name"

app\_service\_plan\_id = azurerm\_app\_service\_plan.svcplan.id

site\_config {

dotnet\_framework\_version = "v4.0"

scm\_type = "LocalGit"

}

}

Once we've saved the file, go to the **Upload/Download** button above the Azure CLI and upload the file.

With that uploaded, we need to run the command terraform init. Next, run terraform plan and review the output to confirm that our changes are reflected. The green plus signs indicate the resources that will be added.

Lastly, run terraform apply to confirm these changes, answering **yes** to the prompt to continue.

Once Terraform completes the deployment, go back to the *Resource groups* page and select **Refresh**. The two items we created via the code appear in the resource list. By clicking on them, we can review the content of the applications.

**Conclusion**

Upon completing the lab, we are now able to use the