**Create a Cosmos DB Instance in Azure with Terraform**

**Introduction**

In this lab, we will cover these 4 objectives:

1. First, we will log into the Azure Portal, configure the Cloud Shell, and then download and run the lab setup script to setup our lab.
2. Second, we will import the resource group.
3. Third, we will create the configuration and deploy the Azure Cosmos DB instance.
4. And for the fourth objective, we will update our configuration with an application container instance that will use the database created in the previous step as a simple database-driven application that we will test in the browser.

**Solution**

**Set Up Cloud Shell and the Lab Environment**

1. In the Azure Portal, click the Cloud Shell icon (>\_) in the upper right. Take note of the **Location** for the lab environment.
2. Select **Bash**.
3. Click **Show advanced settings**.
4. If needed, change the **Cloud Shell region** to match the **Location** of the lab environment.
5. For **Storage account**, select **Use existing**.
6. For **File share**, select **Create new** and give it a name of *terraform*.
7. Click **Create storage**.

**Import the Resource Group into Terraform**

1. Using az group list, find and copy everything within the quotes for the subscription id:

az group list

1. Using the resource name and label from the cosmosDB.tf file, as well as the subscription id that was just copied, import the Azure resource group:

terraform import azurerm\_resource\_group.super-vote <SUBSCRIPTION\_ID>

**Define the Cosmos DB Instance and Apply the Configuration**

1. Open the cosmosDB.tf file:

vim cosmosDB.tf

1. In the Azure Portal, copy the **Resource group** name.
2. In the Cloud Shell, on line 2 of the cosmosDB.tf file, uncomment the line by deleting the # and replace <RESOURCE\_GROUP\_NAME> with the **Resource group** name that was just copied.
3. On line 3, uncomment the line by deleting the # and replace <RESOURCE\_GROUP\_LOCATION> with the location for the lab environment, using westus for West US, centralus for Central US, and so on.
4. Starting on line 11, add a new azurerm\_cosmosdb\_account resource using the following configuration:

resource "azurerm\_cosmosdb\_account" "super-vote" {

name = "tfex-cosmos-db-${random\_integer.ri.result}"

location = azurerm\_resource\_group.super-vote.location

resource\_group\_name = azurerm\_resource\_group.super-vote.name

offer\_type = "Standard"

kind = "GlobalDocumentDB"

consistency\_policy {

consistency\_level = "BoundedStaleness"

max\_interval\_in\_seconds = 10

max\_staleness\_prefix = 200

}

geo\_location {

location = "eastus"

failover\_priority = 0

}

}

1. Save and exit the file:

ESC :wq!

1. Format the file:

terraform fmt

1. Validate the syntax:

terraform validate

1. Do a dry run of the configuration:

terraform plan

1. Apply the configuration:

terraform apply

1. Use terraform state list to verify that the deployment was successful.

**Update the Configuration with the Container Instance and Apply the Configuration**

1. Open the cosmosDB.tf file:

vim cosmosDB.tf

1. Starting on line 29, add a new azurerm\_container\_group resource using the following configuration:

resource "azurerm\_container\_group" "super-vote" {

name = "super-vote"

location = azurerm\_resource\_group.super-vote.location

resource\_group\_name = azurerm\_resource\_group.super-vote.name

ip\_address\_type = "public"

dns\_name\_label = "super-vote"

os\_type = "linux"

container {

name = "super-vote"

image = "mcr.microsoft.com/azuredocs/azure-vote-front:cosmosdb"

cpu = "0.5"

memory = "1.5"

ports {

port = 80

protocol = "TCP"

}

secure\_environment\_variables = {

"COSMOS\_DB\_ENDPOINT" = azurerm\_cosmosdb\_account.super-vote.endpoint

"COSMOS\_DB\_MASTERKEY" = azurerm\_cosmosdb\_account.super-vote.primary\_master\_key

"TITLE" = "Best Superhero!"

"VOTE1VALUE" = "Batman"

"VOTE2VALUE" = "Superman"

}

}

}

1. Save and exit the file:

ESC :wq!

1. Create a new file named output.tf:

vim output.tf

1. In the file, paste the following:

output "application\_endpoint" {

value = azurerm\_container\_group.super-vote.fqdn

}

1. Save and exit the file:

ESC :wq!

1. Format the files:

terraform fmt

1. Validate the syntax:

terraform validate

1. Do a dry run of the updated configuration:

terraform plan

1. Apply the updated configuration:

terraform apply

1. When the deployment is complete, copy the output within the quotes for the application\_endpoint.
2. Use terraform state list to verify that the deployment was successful.

**Test Your Application**

1. In a new browser tab, navigate to the application\_endpoint output URL copied earlier.
2. Test the application by casting your vote.