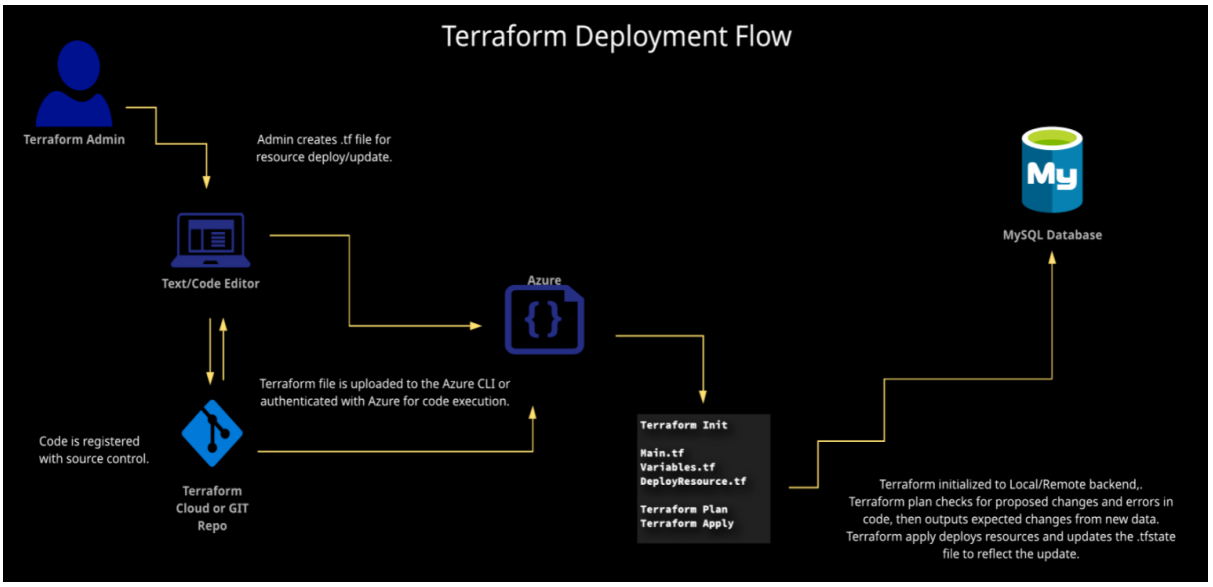
Deploy a MySQL Database with Terraform

In this lab, we will deploy a MySQL v5.7 database instance with 5GB of storage with a Gen 5, basic tier database SKU. To do this, we will be using the Azure CLI to create storage to deploy the database using Terraform.



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 select Show Advanced Settings. Leave both the Resource Group and Storage Account as the provided defaults. In the File share section, enter a name for the account (for this example, we are using Console) and click the Attach Storage button. Once the command prompt completes, we can continue.

Deploy a MySQL Database Instance

Using the following code block, create a **.tf** file named **lab.tf**. Replace the name with a unique name (we'll use"**tflab-mysqlserver-1**" for our example) and the "Enter

Resource Group Name" with the name of the resource group provided by the lab, which should be preceded by three numbers on the Azure portal's main page.

```
provider "azurerm" {
    version = 1.38
    }

resource "azurerm_mysql_server" "example" {
   name                          = "tflab-mysqlserver-1-[some unique
characters]"
  location             = "The location of your lab provided resource
group"
  resource_group_name = "Enter Resource Group Name"

  sku {
    name     = "B_Gen5_2"
    capacity = 2
    tier     = "Basic"
    family   = "Gen5"
  }

  storage_profile {
    storage_mb            = 5120
    backup_retention_days = 7
    geo_redundant_backup  = "Disabled"
  }

  administrator_login          = "mysqladminun"
  administrator_login_password = "easytologin4once!"
  version                      = "5.7"
  ssl_enforcement              = "Enabled"
}

resource "azurerm_mysql_database" "example" {
  name                = "exampledb"
  resource_group_name = "Enter Resource Group Name"
```

```
  server_name         = azurerm_mysql_server.example.name
  charset             = "utf8"
  collation           = "utf8_unicode_ci"
}
```

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.

To make sure that this worked correctly, go back to the Resource groups page and select Refresh. The database that we created appears on our resource list.

Conclusion

Upon completing the lab, we are now able to use the Azure CLI to create a MySQL Database and the type of code needed to create the database.