This document outlines the steps to deploy a virtual network with subnets, a web server, and an Azure SQL database using Terraform. The code is modularized for reusability and simplicity.

Directory Structure

The Terraform code is organized as follows:

```
AzureTaskResources/
├── main.tf
├── outputs.tf
├── providers.tf
├── variables.tf
├── terraform.tfvars
├── modules/
│   ├── network/
│   │   ├── main.tf
│   │   ├── variables.tf
│   │   └── outputs.tf
│   ├── vm/
│   │   ├── main.tf
│   │   ├── variables.tf
│   └── sql/
│       ├── main.tf
│       ├── variables.tf
```

Steps to Use

Step 1: Clone the Repository

Clone the repository containing the Terraform files.

```
git clone <repository-url>
cd AzureTaskResources
```

Step 2: Initialize Terraform

Run the following command to initialize Terraform and download the required providers:

```
terraform init
```

Step 3: Update Variables (Optional)

Edit the `terraform.tfvars` file to customize the deployment parameters:

```
region              = "East US"
resource_group_name = "example-resource-group"
web_subnet_name     = "web-subnet"
db_subnet_name      = "db-subnet"
vm_name             = "web-server"
admin_username      = "adminuser"
admin_password      = "StrongPassword123!"
vm_size             = "Standard_DS1_v2"
sql_server_name     = "example-sql-server"
sql_database_name   = "example-database"
sql_sku             = "S1"
```

Step 4: Plan the Deployment

Run the following command to view the resources that will be created:

```
terraform plan
```

Step 5: Apply the Deployment

Run the following command to create the resources in Azure:

```
terraform apply
```

Confirm the deployment by typing `yes` when prompted.

Step 6: Verify the Deployment

Once the deployment is complete, Terraform will output the following:

- Public IP of the web server.
- SQL Server name.

You can verify the resources in the Azure portal.

Outputs

After running `terraform apply`, the following outputs are available:

- **Web Server Public IP**: Use this to access the web server.
- **SQL Server Name**: Use this to connect to the SQL database.

Modules Breakdown

Network Module

- **Purpose**: Creates a virtual network with two subnets (web and database) and a network security group.
- **Path**: `modules/network/`

Web VM Module

- **Purpose**: Deploys a Linux-based virtual machine in the web subnet.
- **Path**: `modules/vm/`

SQL Module

- **Purpose**: Deploys an Azure SQL server and database in the database subnet.
- **Path**: `modules/sql/`

Cleaning Up Resources

To destroy all the resources created by Terraform, run:
`terraform destory`