Python And Terraform

**1 Objective**

This document outlines the steps to create a Terraform module that facilitates the creation of a temporary environment tailored for testing purposes. By combining Terraform's infrastructure provisioning capabilities with Python scripting, this module streamlines the setup process for temporary environments.

**2 Directory Structure**

**│**

**├── main.tf**

**├── variables.tf**

**├── outputs.tf**

**├── scripts/**

**│ └── setup\_script.py**

**└── README.md**

## 

## **3 Files Description**

## 

1. **main.tf:** This file contains the primary Terraform configuration responsible for provisioning the temporary environment.
2. **variables.tf:** Defines input variables utilized within the Terraform module.
3. **outputs.tf:** Specifies the output values of the Terraform module.
4. **scripts/setup\_script.py:** A Python script designed to set up the environment after provisioning.
5. **README.md**: Documentation encompassing instructions, usage guidelines, and pertinent information regarding the module's functionality.

## **3 Steps to Implement**

1. **Directory Setup:**
   1. Create a folder named terraform-ephemeral-environment.
   2. This folder includes the mentioned files and directories adhering to the provided structure.
2. **File Configuration:**
   1. Populate **main.tf** with the necessary Terraform configurations for environment provisioning.
   2. Define required variables in **variables.tf**.
   3. Specify output values in **outputs.tf**.
   4. Add Python setup logic to **scripts/setup\_script.py**.
3. **Documentation:**
   1. Update **README.md** to furnish comprehensive documentation, including module usage, instructions for customization, and any additional relevant details.
4. **Testing and Deployment:**
   1. Execute the module to provision an ephemeral environment using Terraform.
   2. Validate the setup using the provided Python script for environment configuration.

## 

## 