## **Overview**

[Terraform](https://www.terraform.io/intro/index.html) is a tool for building, changing and versioning infrastructure safely and efficiently. Terraform can manage existing and popular cloud service providers as well as custom in-house solutions.

Configuration files describe to **Terraform** the components needed to run a single application or your entire datacenter. Terraform generates an execution plan describing what it will do to reach the desired state, and then executes it to build the described infrastructure. As the configuration changes, Terraform is able to determine what changed and create incremental execution plans which can be applied.

**How to create Azure Resources using Terraform in Azure Cloud | Automate Infrastructure setup using Terraform in Azure Cloud | Create Azure WebApp using Terraform**

WebApp is an App service provided by Azure Cloud for hosting your application. It supports .NET, Java, PHP,  Python and NodeJS. It is a fully managed Platform as a Serice (PaaS) where developers can deploy mobile or web applications in Azure Cloud.

With App Service, you pay for the Azure compute resources you use. The compute resources you use are determined by the *App Service plan* that you run your apps on.

[Hashicorp's Terraform](https://www.terraform.io/) is an open-source tool for provisioning and managing cloud infrastructure. Terraform can provision resources on any cloud platform.

[Diagram

Description automatically generated](https://1.bp.blogspot.com/-rukBdutuutE/YIc_B7CoBZI/AAAAAAAADZQ/Pwyxy3sXBKkCeFYWVdvfJR-qupq1QlU8ACLcBGAsYHQ/s996/tf%2Bcloud.png)

Terraform allows you to create infrastructure in configuration files(tf files) that describe the topology of cloud resources. These resources include virtual machines, storage accounts, and networking interfaces. The Terraform CLI provides a simple mechanism to deploy and version the configuration files to Azure.

#### Advantages of using Terraform:

* Reduce manual human errors while deploying and managing infrastructure.
* Deploys the same template multiple times to create identical development, test, and production environments.
* Reduces the cost of development and test environments by creating them on-demand.

#### How to Authenticate with Azure?

Terraform can authenticate with Azure in many ways, in this example we will use Azure CLI to authenticate with Azure and then we will create resources using Terraform.

**Pre-requistes:**

### Install Azure CLI in Ubuntu 18.0.4 | How to setup Azure CLI in Ubuntu 18.0.4 | How to Install Azure CLI in Ubuntu

The **Azure command-line** interface (**Azure CLI**) is a set of commands used to create and manage **Azure** resources. The **Azure CLI** is available across **Azure** services and is designed to get you working quickly with **Azure**, with an emphasis on automation. **Azure CLI** is Microsoft's cross-platform command-line experience for managing Azure resources.

Azure CLI can be installed by executing the below command:

curl -sL https://aka.ms/InstallAzureCLIDeb | sudo bash

Once Azure CLI is installed, you can verify it by executing below command:

az version

Text

Description automatically generated

**Terraform needs to be installed.**

**How to install Terraform on Ubuntu 18.0.4 | TerraForm Installation on Ubuntu 18.0.4 | Setup Terraform on Ubuntu**

Please find steps for installing Terraform on Ubuntu Machine.

Terraform is used for provisioning infrastructure on Cloud. you don't need to create manually any resource in AWS.

**Create a working directory**

sudo mkdir -p /opt/terraform

cd /opt/terraform  
  
**Download Terraform from Hasicorp website**

sudo wget https://releases.hashicorp.com/terraform/1.2.9/terraform\_1.2.9\_linux\_386.zip

**Install unzip utility**

sudo apt-get install unzip -y

**Unzip Terraform Zip file**  
sudo unzip terraform\_1.2.9\_linux\_386.zip

**Add terraform to PATH**  
sudo mv /opt/terraform/terraform /usr/bin/

**Verify Terraform version**

terraform -version

this should show version of Terraform

#### Logging into the Azure CLI

Login to the Azure CLI using:

az login

The above command will open the browser and will ask your Microsoft account details. Once you logged in, you can see the account info by executing below command:

az account list

Now create a directory to store Terraform files.

mkdir azure-terraform

cd azure-terraform

Let's create a terraform file to use azure provider. To configure Terraform to use the Default Subscription defined in the Azure CLI, use the below cod.

sudo vi azure.tf

terraform {  
  required\_providers {  
    azurerm = {  
      source  = "hashicorp/azurerm"  
      version = "=2.46.0"  
    }  
  }  
}  
  
# Configure the Microsoft Azure Provider  
provider "azurerm" {  
  features {}  
}

**Now initialize the working directory**

Perform the below command:

terraform init

Once directory is initialized, you can start writing code for setting up the infrastructure.

**Create WebApp(App Service) using Terraform script**

sudo vi create-app-svc.tf

# Create a resource group  
resource "azurerm\_resource\_group" "dev-rg" {  
  name     = "dev-environment-rg"  
  location = "South Central US"  
}  
  
# Create app service plan  
resource "azurerm\_app\_service\_plan" "service-plan" {  
  name = "simple-service-plan"  
  location = azurerm\_resource\_group.dev-rg.location  
  resource\_group\_name = azurerm\_resource\_group.dev-rg.name  
  kind = "Linux"  
  reserved = true  
  sku {  
    tier = "Standard"  
    size = "S1"  
  }  
  tags = {  
    environment = "dev"  
  }  
}  
  
# Create JAVA app service  
resource "azurerm\_app\_service" "app-service" {  
  name = "my-awesome-app-svc"  
  location = azurerm\_resource\_group.dev-rg.location  
  resource\_group\_name = azurerm\_resource\_group.dev-rg.name  
  app\_service\_plan\_id = azurerm\_app\_service\_plan.service-plan.id  
  
site\_config {  
    linux\_fx\_version = "TOMCAT|8.5-java11"  
  }  
tags = {  
    environment = "dev"  
  }  
}

Now Perform the below command to validate terraform files.

terraform validate

[](https://1.bp.blogspot.com/-NsdXmfjhehc/YIc4il8_cxI/AAAAAAAADYw/l7iN8UHsNNomeAhFByuKDE-TPNC7y-KZQCLcBGAsYHQ/s1074/tf%2Bvalidate.png)

perform plan command to see how many resources will be created.

terraform plan

[Text

Description automatically generated](https://1.bp.blogspot.com/-Wx4HhlWKMug/YIc5GdSkoDI/AAAAAAAADY4/49jknR3xWMoO6nyqX6XzsPdjq-b-pTxfACLcBGAsYHQ/s1054/tf%2Bplan.png)

terraform apply

Do you want to perform these actions?

type yes

[Text

Description automatically generated](https://1.bp.blogspot.com/-SzojULzrIqE/YIc5_qHM7AI/AAAAAAAADZA/3ijHWdvxtxo3B_17q5UGX3u2-VonTDUTwCLcBGAsYHQ/s1564/tf%2Bapply.png)

Now Login into Azure Cloud to see all the resources created.

[Graphical user interface, application, Teams

Description automatically generated](https://1.bp.blogspot.com/-Sjh_mAjfqwc/YIc66Fo0TvI/AAAAAAAADZI/JRhO5LSN4cw1Vm6bWku7U5QfOOx17nZCQCLcBGAsYHQ/s1638/az%2Blist.png)