Terraform and AWS

Note 1: Terraform ( IaC – Infrastructure as Code)

Terraform is used for the configuration purpose (IaC) and similarly Ansible is used for the integration purpose

Use case : it will be used to deploy very complex infrastructure to the cloud ( AWS, Azure, GCP)

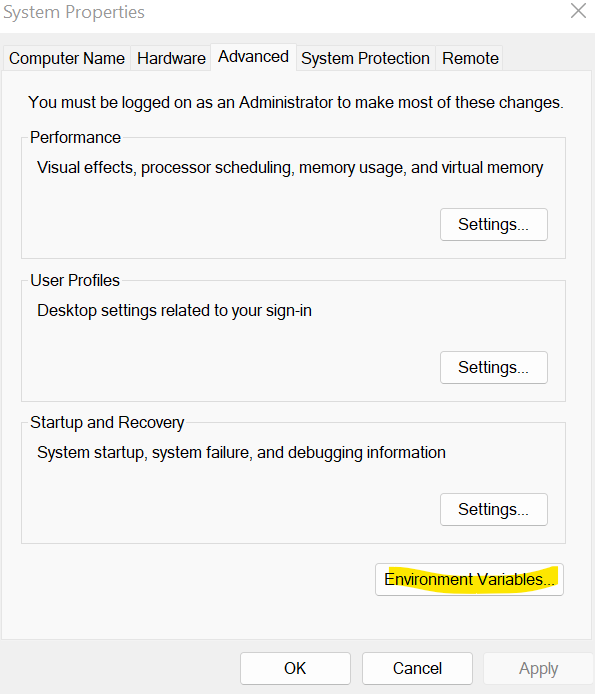
Reference link to all related Document: **https://registry.terraform.io/browse/providers**

**How to Install Terraform**

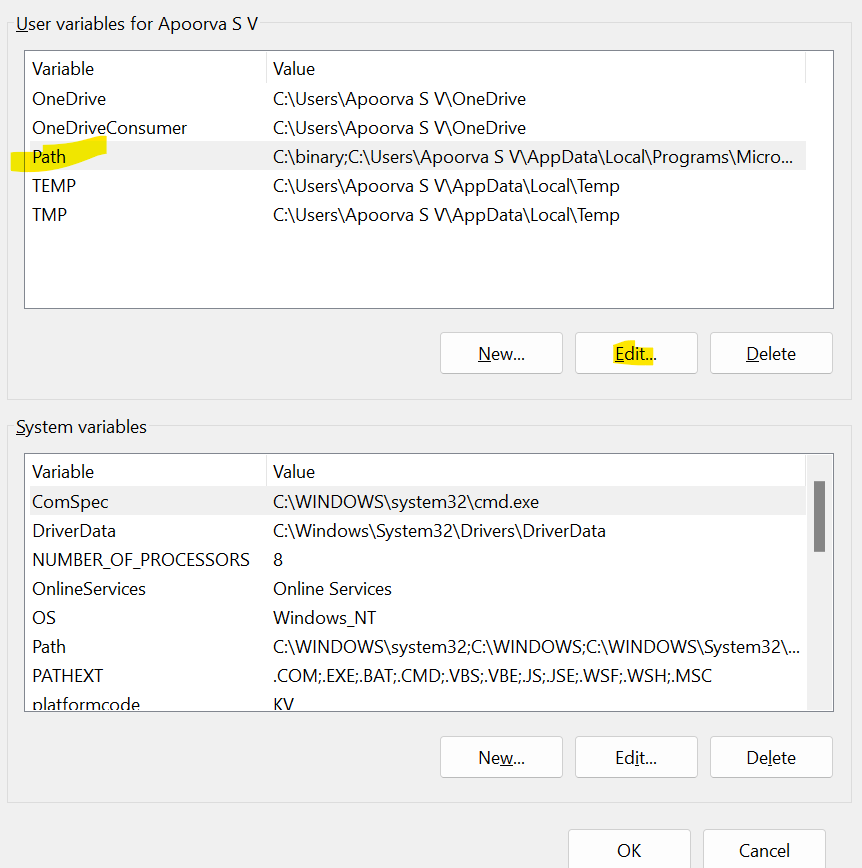
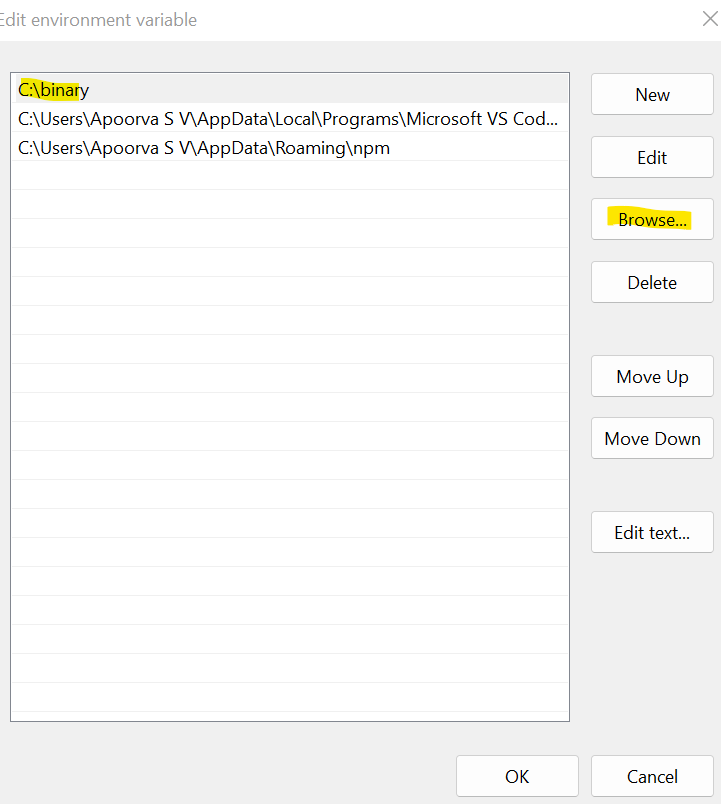
Step 1: Download binary from the google ( AMD64)

Step 2: Once it is installed Extract jar file

Step 3: In your system search for “advance system setting”

Step 4: Click on Environment tab  


Step 5: Click on path and edit then browse you terraform path.

Step 5 : To verify terraform installed go to CMD prompt and enter Terraform (You should able to see terraform related information’s)

**AWS**

1. **After account creation**
2. **Need to create user in IAM (Identity and Access Management) services**

**Terraform scripting**

**Terraform Primary function create , modify, destroy infrastructure to match the desired state**

**Important Consideration to launch the AWS resources**

1. How to Authenticate in AWS
2. Which region source need to be launched
3. Which resource you want to launch

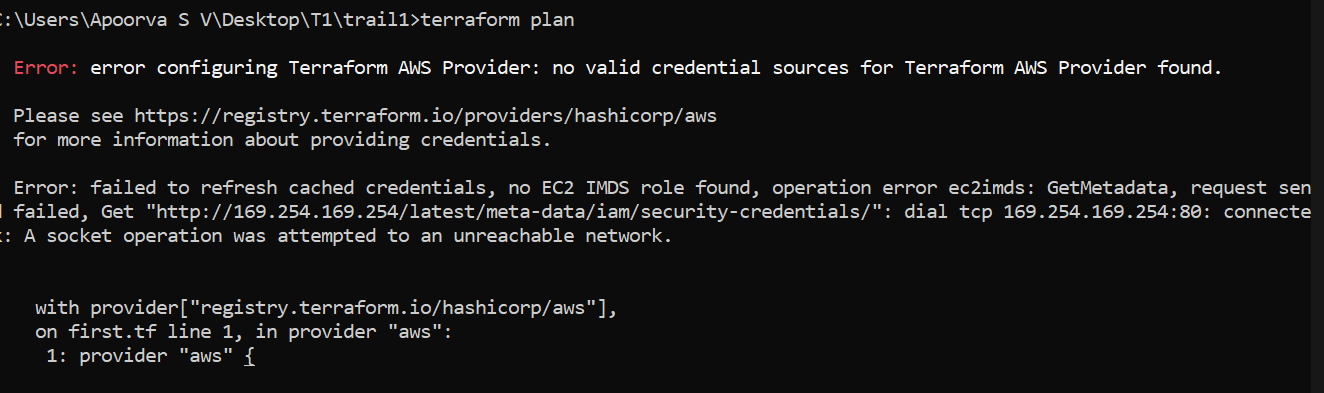
**Must Provider navigate to respective provide Azure,AWS, etc..**

provider "aws" {

region = "us-east-1"

}

Need to assigne the access key to initiate the aws provider or we will get below issue



Create new User in AWS and Take Access key and Secrete Key

provider "aws" {

    region = "us-east-1"

      access\_key = "AKIATZEFJKUJ2FRVM7F6"

     secret\_key = "GeppuOPef4jMdKK83YCr+Y9jgFOdMcHlH4qsBvw8"

}

We have C:\Users\Apoorva S V\Desktop\T1\trail1\.terraform\providers\registry.terraform.io\hashicorp\aws\4.22.0\windows\_386

Where it will download all Provider plugin

How to destroy the resources ?

* **Command to use to destroy - Terraform destroy**
* **To delete the specific target – terraform destroy -target aws\_instance.myEc2**
* **Target format – resource type + local resource name**

**Git Files**

**To Create New Git repo**

terraform {

  required\_providers {

   github = {

    source  = "integrations/github"

      version = "~> 4.0"

   }

  }

  }

provider "github" {

    token = "?" // We need to create token

}

resource "github\_repository" "git-repo" {

  name        = "example"

  visibility = "private"

}

**Current state and Desired State**

**Current state -** Resource and their configuration available in the UI

**Desired State** – It is the infrastructure defined in the Terraform file

In the real world both Current and Desired state should match if we find out any difference in them Terraform automatically identify the changes when the Terraform apply is initiated.

