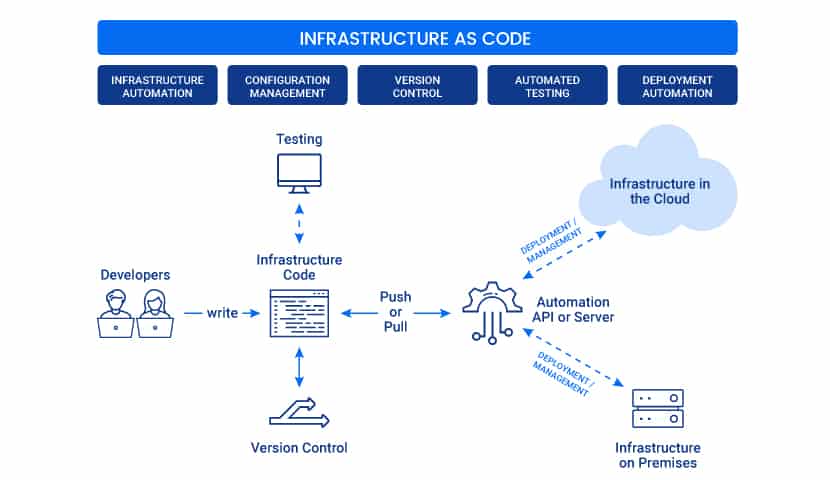
Terraform

Iac



**Harshicrop Terraform**

Terraform is an open source infrastructure as code ([IaC](https://www.techtarget.com/searchitoperations/definition/Infrastructure-as-Code-IAC)) software tool that allows DevOps engineers to programmatically provision the physical resources an application requires to run.Terraform is a tool for infrastructure provisioning

**Keypoints of Terraform:**

1. **Improve speed**
2. **Improve reliability**
3. **Prevent configuration drift**
4. **Support experimentation, testing, and optimization**

**Terraform commands for different stages:**

**1.Terraform will know what is the current state of the infrastructure**

**2.Terraform will determines what actions are necessary to achieve the desired state**

**3.just a preview ,no changes to real resources**

**4.Refresh-query infrastructure provider to get current state**

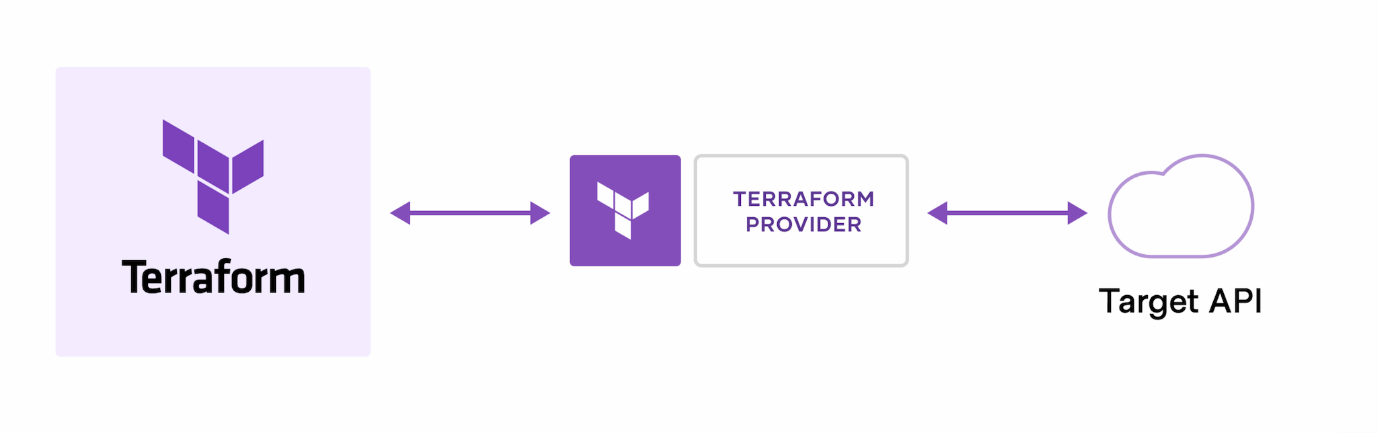
**5.plan- create an execution plan**

**6.apply-execute the plan**

**7.destroy –destroy the resources/infrastructure**

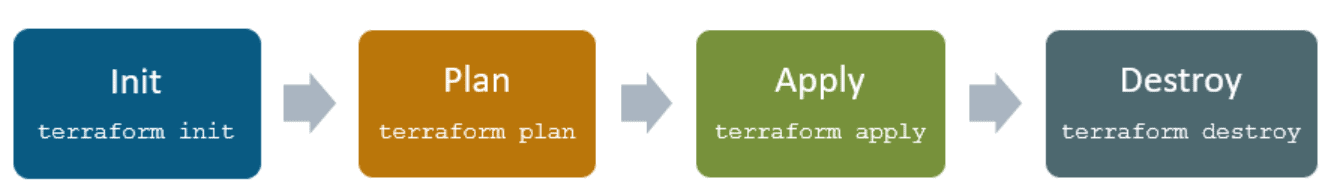
**How does Terraform work?**

Terraform creates and manages resources on cloud platforms and other services through their application programming interfaces (APIs). Providers enable Terraform to work with virtually any platform or service with an accessible API.



## ****Terraform Lifecycle****

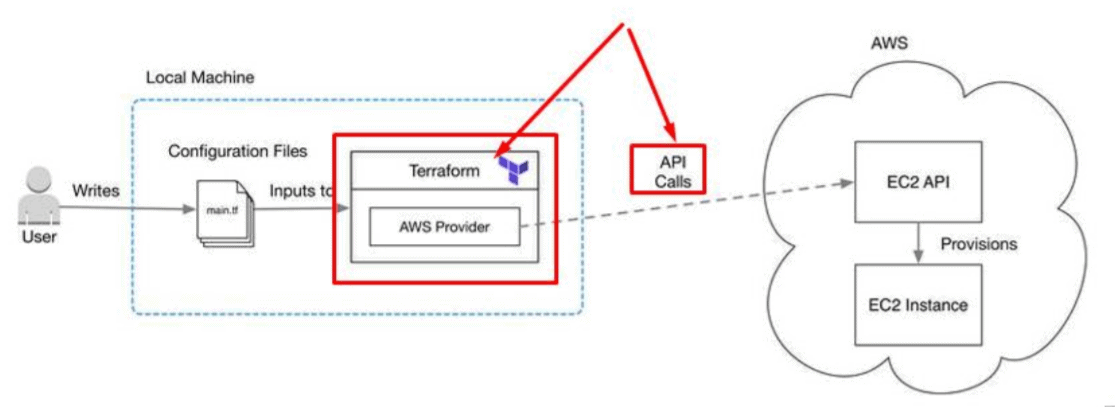
Terraform lifecycle consists of – **init**, **plan**, **apply**, and **destroy**.



**1.Terraform init** initializes the (local) Terraform environment. Usually executed only once per session.  
2. **Terraform plan** compares the Terraform state with the as-is state in the cloud, build and display an  
execution plan. This does not change the deployment (read-only).  
3. **Terraform apply** executes the plan. This potentially changes the deployment.  
4. **Terraform destroy** deletes all resources that are governed by this specific terraform environment.

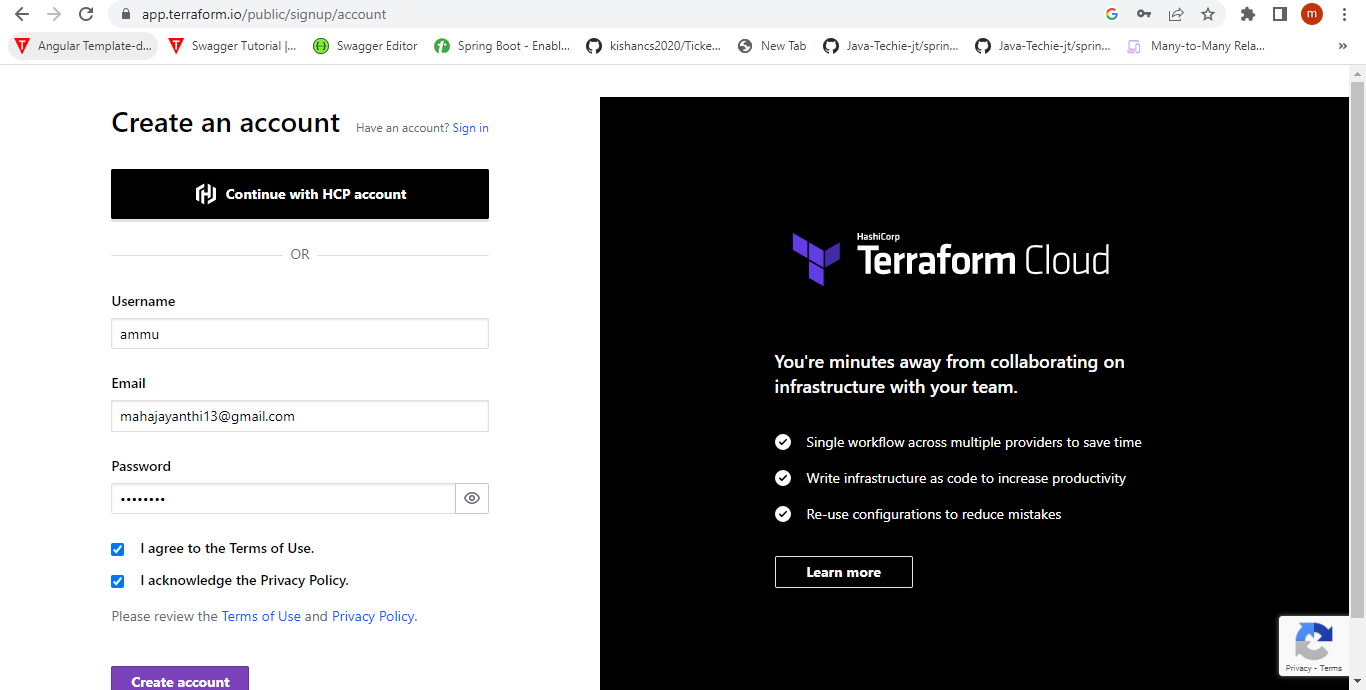
## ****Terraform Providers****

A provider is responsible for understanding API interactions and exposing resources. It is an executable plug-in that contains code necessary to interact with the API of the service. Terraform configurations must declare which providers they require so that Terraform can install and use them

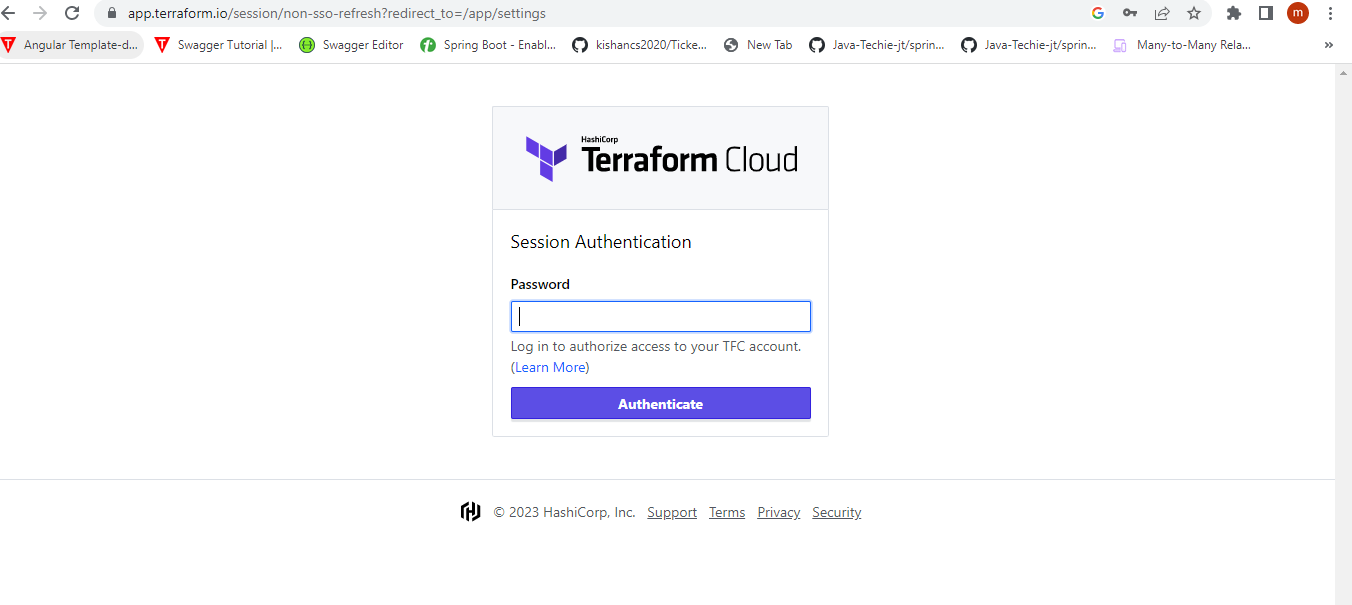


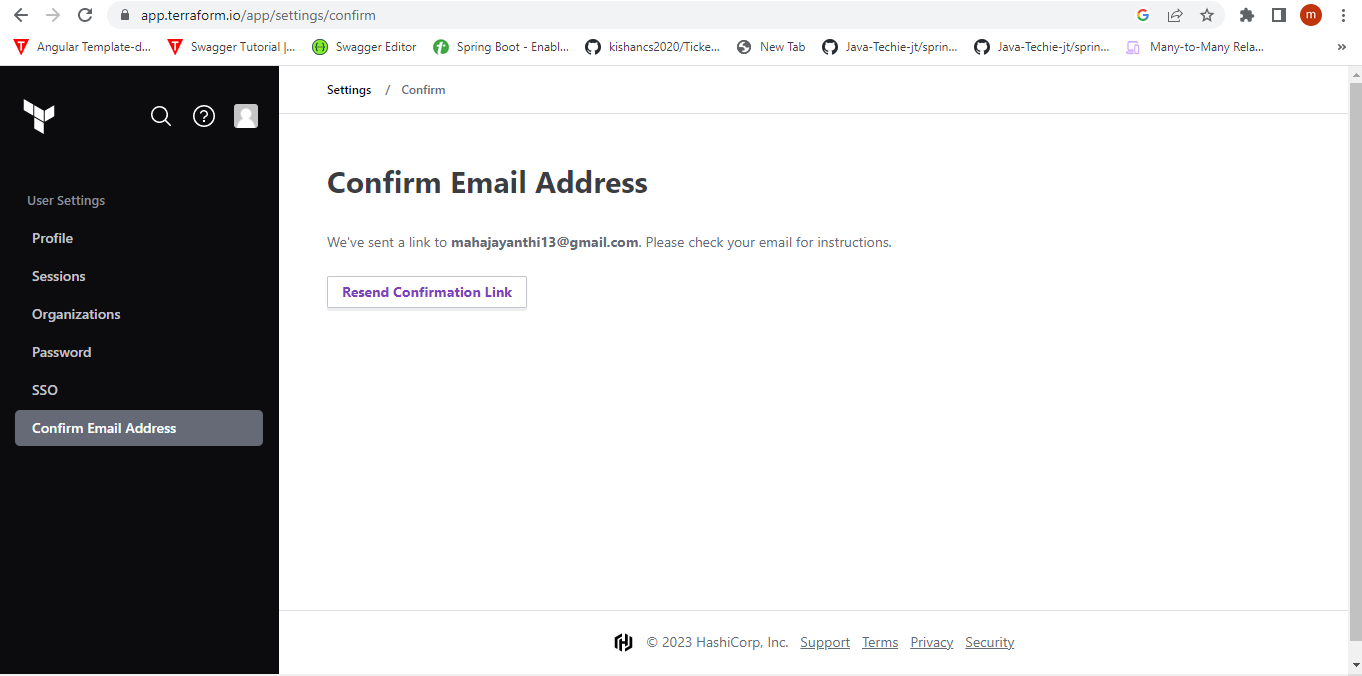
**Steps for working in Terraform:**

**Step1: How to create a account in Terraform Cloud**

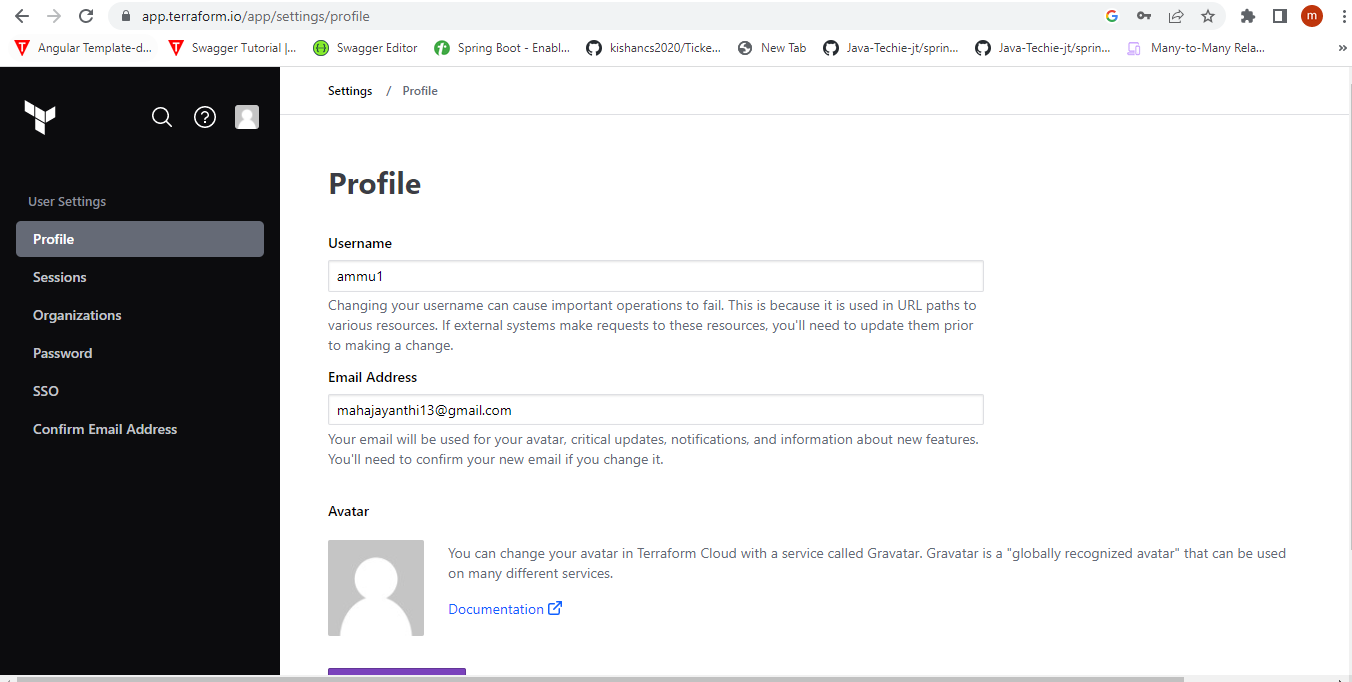


**Step2: Using the username and password**





**Step3: Finally a profile has been created based on our username and password**



**Benefits of using Terraform:**

Using Terraform has several advantages over manually managing your infrastructure: Terraform can manage infrastructure on multiple cloud platforms. The human-readable configuration language helps you write infrastructure code quickly. Terraform's state allows you to track resource changes throughout your deployments.

**Advantages:**

### Improve multi-cloud infrastructure deployment

### Automated infrastructure management

### Infrastructure as code

### Reduced development costs

### 5.Reduced time to provision

**DisAdvantages:**

1. Currently under development. Each month, we release a beta version.
2. Terraform does not support script generation from the state.
3. Terraform acknowledges that specific versions may include bugs.
4. There is no way to roll back. As a result, we must delete everything and re-run if necessary.

