

Semester	T.E. Semester V – Information Technology
Subject	Advance DevOps Lab
Subject Professor In-charge	Prof. Indu Anoop
Laboratory	(Leave blank for now)

Student Name	Khot Mohammed Siddique	
Roll Number	20101A0048	
Grade and Subject Teacher's Signature		

Experiment	5	
Problem Statement	To understand terraform lifecycle, core concepts/terminologies and install it on a Linux Machine	
Resources / Apparatus Required	Hardware: Computer System	Software: Web Browser
Details	<p><b>Terraform</b></p> <p>Terraform is an infrastructure as code (IaC) tool that allows you to build, change, and version infrastructure safely and efficiently. This includes low-level components such as compute instances, storage, and networking, as well as high-level components such as DNS entries, SaaS features, etc. Terraform can manage both existing service providers and custom in-house solutions.</p> <p><b>Key Features</b></p> <p><b>Infrastructure as Code:</b></p> <p>You describe your infrastructure using Terraform's high-level configuration language in human-readable, declarative configuration files. This allows you to create a blueprint that you can version, share, and reuse.</p>	

## Resource Graph

Terraform builds a resource graph and creates or modifies non-dependent resources in parallel. This allows Terraform to build resources as efficiently as possible and gives you greater insight into your infrastructure.

## Change Automation

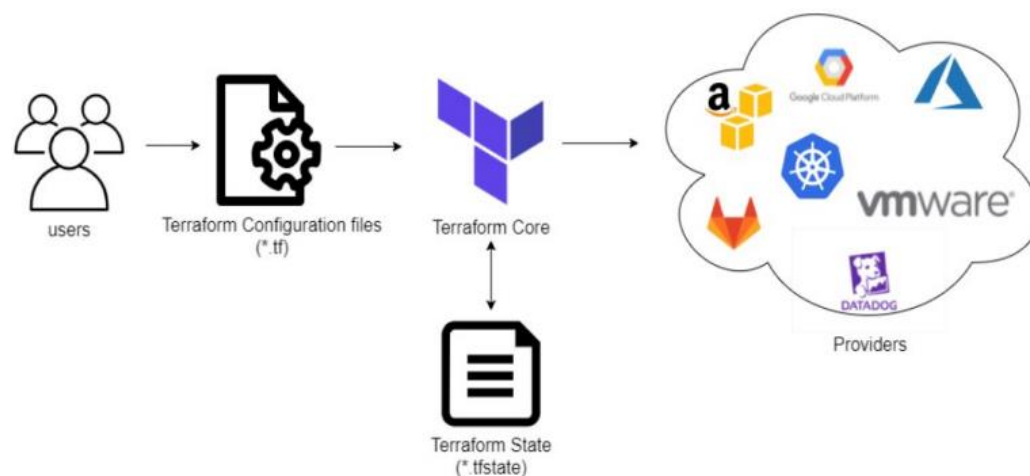
Terraform can apply complex changesets to your infrastructure with minimal human interaction. When you update configuration files, Terraform determines what changed and creates incremental execution plans that respect dependencies.

## How does Terraform work?

Terraform works with two major components:

one is the **terraform core**: it takes the terraform configuration which is being provided by the user and then takes the terraform state which is managed by terraform itself. As such, this gets fed into the core that is responsible for figuring out what is that graph of our different resources for example how these different pieces relate to each other or what needs to be created/updated/destroyed, it does all the essential lifecycle management.

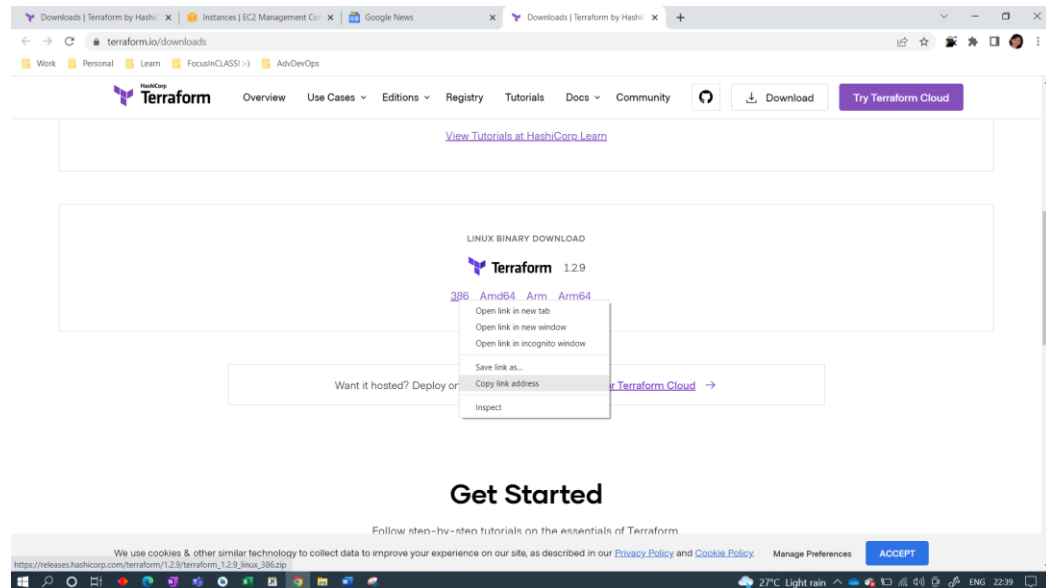
On the backside, terraform supports many different **providers**, such as: cloud providers (AWS,GCP,AZURE) and they also could be on-premise infrastructure (VMware, OpenStack.) But this support is not restricted or limited only to Infrastructure As A Service , terraform can also manage higher level like Platform As A Service(Kubernetes, Lambdas..)or even Software As A Service (DataDog, GitHub..)



All of these are important pieces of the infrastructure, they are all part of the logical end-to-end delivery.

	<p>Terraform has over a hundred providers for different technologies, and each provider gives terraform users access to their resources. It also gives you the ability to create infrastructure at different levels.</p> <p><b>Terraform Core Concepts:</b></p> <p>Below are the core concepts/terminologies used in Terraform:</p> <ul style="list-style-type: none"> <li>•<b>Variables:</b> Also used as input-variables, it is a key-value pair used by Terraform modules to allow customization.</li> <li>•<b>Provider:</b> It is a plugin to interact with APIs of service and access its related resources.</li> <li>•<b>Module:</b> It is a folder with Terraform templates where all the configurations are defined</li> <li>•<b>State:</b> It consists of cached information about the infrastructure managed by Terraform and its related configurations.</li> <li>•<b>Resources:</b> It refers to a block of one or more infrastructure objects (compute instances, virtual networks, etc.), which are used in configuring and managing the infrastructure.</li> <li>•<b>Data Source:</b> It is implemented by providers to return information on external objects to terraform.</li> <li>•<b>Output Values:</b> These are return values of a terraform module that can be used by other configurations.</li> <li>•<b>Plan:</b> It is one of the stages where it determines what needs to be created, updated, or destroyed to move from the real/current state of the infrastructure to the desired state.</li> <li>•<b>Apply:</b> It is one of the stages where it applies the changes in the real/current state of the infrastructure to move to the desired state.</li> </ul>
Code	<p>Ec2 instance bana ubuntu 20  Port default 22 hi chahiye  Key pairs bana ek terraformke liye  Instance ko connect kar  Sudo su karke niche given steps ko use karke terraform install kar  Install hone ke baad ussi jagah terraform commands use kar</p> <p>Terraform Installation Steps on Ubuntu20</p> <p><b>Step: 1</b> (Ensure root user privileges )Terraform uses HashiCorp Configuration Language (HCL) to manage environments of Operators and Infrastructure teams. To <b>download</b> go to site <a href="https://www.terraform.io/downloads.html">https://www.terraform.io/downloads.html</a></p>

Select the appropriate package for your operating system and architecture. Copy link address



Use wget command to download from the link

wget [https://releases.hashicorp.com/terraform/1.2.9/terraform\\_1.2.9\\_linux\\_386.zip](https://releases.hashicorp.com/terraform/1.2.9/terraform_1.2.9_linux_386.zip)

**Step:2 unzip the archive by using below command**

```
unzip terraform_1.2.9_linux_386.zip //unzip hai nhi to apt install unzip
```

The archive will extract a single binary called **terraform** and Move the terraform binary to a directory included in your system's PATH in my case usr/local/bin/

```
mv terraform /usr/local/bin/
```


**Step 4: To check whether Terraform is installed, run:**

```
terraform -v
```

or  
type

```
terraform
```

to see options list for terraform

	<div><div><div><div><div></div><div>Services</div><div><input type="text" value="Search for services, features, blogs, docs, and more"/></div><div>[Alt+5]</div></div></div><div><div>Cloud9</div><div>IAM</div><div>Elastic Beanstalk</div><div>CodePipeline</div><div>EC2</div></div><div>root@ip-172-31-82-25:/home/ubuntu# terraform -v</div><div>Terraform v1.2.9</div><div>on linux_386</div><div>Your version of Terraform is out of date! The latest version</div><div>is 1.3.0. You can update by downloading from <a href="https://www.terraform.io/downloads.html">https://www.terraform.io/downloads.html</a></div><div>root@ip-172-31-82-25:/home/ubuntu# terraform</div><div>Usage: terraform [global options] &lt;subcommand&gt; [args]</div><div>The available commands for execution are listed below.</div><div>The primary workflow commands are given first, followed by</div><div>less common or more advanced commands.</div><div>Main commands:</div><div>init Prepare your working directory for other commands</div><div>validate Check whether the configuration is valid</div><div>plan Show changes required by the current configuration</div><div>apply Create or update infrastructure</div><div>destroy Destroy previously-created infrastructure</div><div>All other commands:</div><div>console Try Terraform expressions at an interactive command prompt</div><div>fmt Reformat your configuration in the standard style</div><div>force-unlock Release a stuck lock on the current workspace</div><div>get Install or upgrade remote Terraform modules</div><div>graph Generate a Graphviz graph of the steps in an operation</div><div>import Associate existing infrastructure with a Terraform resource</div><div>login Obtain and save credentials for a remote host</div><div>logout Remove locally-stored credentials for a remote host</div><div>output Show output values from your root module</div><div>providers Show the providers required for this configuration</div><div>refresh Update the state to match remote systems</div><div>show Show the current state or a saved plan</div><div>state Advanced state management</div><div>taint Mark a resource instance as not fully functional</div><div>test Experimental support for module integration testing</div><div>untaint Remove the 'tainted' state from a resource instance</div><div>version Show the current Terraform version</div><div>workspace Workspace management</div><div>Global options (use these before the subcommand, if any):</div><div>-chdir=DIR Switch to a different working directory before executing the</div><div>given subcommand.</div><div>-help Show this help output, or the help for a specified subcommand.</div><div>-version An alias for the "version" subcommand.</div><div>root@ip-172-31-82-25:/home/ubuntu#</div></div></div>
Conclusi on	Successfully understood terraform and it's installation