Terraform

What is Terraform?

* Tool that allows you to automate the provisioning and management of infrastructure on any cloud platform or service using declarative configuration files.

* Terraform is Infrastructure as a code

* To automate the resources of the cloud we use the terraform.

* Basically, we can do a process in the cloud, using 3 ways, consider them manual or automation.

1) Start and maintain resources using console

a) In this approach, maintenance overhead is high, as we need to assign all the properties to the resources manually.

2) Use Cloud CLI, and start and maintain resources (Need to execute commands in Cloud CLI)

3) We can use the help of SDKs of other programming languages like python, which can execute azure from its scripts, thus we can automate.

4) We can use ARM templates, to automate the resources. ARM => Azure resource manager (Service which is offered by azure, for managing resources)

If we try to maintain, these resources manually, we need to configure all manually, we need to configure the servers, networks, storage etc, and if any new user want to replicate this virtual machine, it is difficult to understand the configurations, because as said these are done manually, there may or may not be documentation. So, when coming to ansible, we will be having yaml files and for terraform we will have HCL and state files or JSON files, so it is easy to figure out configurations, and it is easy to re-use for other VM's. If any other team, want to create a resource with same configuration, they can simply use already existing configuration files, instead of creating them manually.

We can also use ARM's (Azure resource manager) to structure the resource management process, but ARM is only confined to Azure, so if we want something, which is cross platform supported and useful to create and manage the resources and develop the infrastructure, we can use Terraform. Terraform supports many cloud providers, like Amazon Web Services, IBM Cloud, Google Cloud Platform, Linode , Microsoft Azure , Oracle Cloud Infrastructure , or VMware vSphere as well as open Stack.

Terraform basically supports high-level configuration language called Hashicorp Configuration language, but it also supports JSON as a language through we can accept the requests.

Terraform supports AWS, IBM Cloud (Bluemix), GCP, Linode , Azure , Oracle Cloud Infrastructure , VMWare vSphere , Open Stack.

Ansible is configuration management tool, terraform is infrastructure management as a code.

Ansible is mainly used for configuration management and automation. It helps you to install, update and configure software on your servers or other devices. It also lets you automate tasks like provisioning, deployment and security. Ansible uses simple YAML syntax to write playbooks that define the desired state of your infrastructure. Ansible is agentless, which means you don't need to install any software on the target systems to manage them. Ansible is imperative, which means you specify the steps to achieve the desired state.

Terraform is mainly used for infrastructure as code and orchestration. It helps you to create, modify and destroy cloud resources like servers, networks, storage, etc. It also lets you manage dependencies and logical relationships between resources.

Terraform uses HCL language to write configuration files that define the desired state of your infrastructure.

Terraform is agent-based, we need to install a software called Terraform CLI on system to interact with cloud providers.

Terraform uses a declarative approach, meaning you define the desired state of your infrastructure, and terraform figures out the steps to achieve that state. This is different from Ansible, which is more procedural, requiring you to define the steps to reach the desired state.

Terraform is declarative and figures out the steps to reach the desired state on its own, while Ansible is procedural and requires you to define the steps.

Both Terraform and Ansible have their own strengths and are often used together in the industry to manage different aspects of infrastructure.

Terraform is excellent for provisioning and managing the state of infrastructure, particularly in a cloud environment. It’s often used to set up the infrastructure from scratch.

Ansible, on the other hand, excels at configuration management and application deployment. Once the infrastructure is set up (possibly by Terraform), Ansible can ensure that all systems are configured correctly and maintain this state over time.

Terraform have state files, which have the current status information of their already created resources, and configuration files, which have the entire build details, which is used to build the infrastructure.

Main advantage of terraform comes at place of destroying the resources, if we try to delete a resource like VM, all the underlying resources won't be deleted automatically, but terraform deletes the resource and its underlying dependent resources. And even while creating a resource, it helps us to create network groups, security groups and the rest.

Resource Groups:

In general, to create a resource group, we need to create a subscription. Resource Group is created in the context of the subscription.

Usage of Resource group: In general, in any company, we will have set of resources depending on the type of the server, we are on, either it is dev, prod, test etc ..., so for every type of server, we can create a resource group and add the required resources in respective resource groups. So, whenever we want to deploy an infrastructure on any server, we can directly use this resource group and deploy it directly as we are having some template of resources.

Resource Group is created for an application for every environment.

Resource Group is a container that holds related resources for an Azure solution. It stores metadata about the resources. It also helps us in sharing access to other team members using RBAC (Role-Based Access Control). So only defined users can access the resources of the resource groups. Resource Groups can contain storage, database cache and API's etc as its resources. We have to select subscription while creating the resource group, and we can also create the region of where the resource group going to create. After creating the resource group, we can add individual resources into the group.

If we want to create a resource group through terraform, basically, we need details of subscription, resource group name and the region of the cloud provider where we want to create the resource group. With these details we can create a resource group through Azure GUI & Terraform & Azure CLI (Cloud Shell), VSCode (with help of (Azure Terraform - extension), but we need the credentials of the cloud), Windows PowerShell.

How to check version of terraform:

```
terraform --version
```

To check list of terraform commands available:

```
terraform
```