**Terraform**

**1.What is Terraform ?**

* It’s an Open source Infra as a code tool, which is used to automate infra and service build in AWS.

**2. Define Terraform init**

1. Installing Plugins
2. Installation of a Child Module
3. Initialization of the backend

**3.Define Terraform provider?**

Terraform is a tool for managing and informing infrastructure resources such as physical machines, virtual machines (VMs), network switches, containers, and more. A provider is responsible for API interactions that are thoughtful and reveal resources. Terraform is compatible with a wide range of cloud providers.

**4. Define IAC?**

IaC is a short form of the term “Infrastructure as Code”. IaC refers to a scheme whereby developers can run and provision the computer data center’s mechanically instead of getting into a physical process. [Terraform IAC](https://k21academy.com/terraform-iac/day1-training/), for example, is a case tool of IAC

**5.How to check the installed version of Terraform?**

We can use the terraform -version of the command to identify the version which we are running.

**6.Module :**

Terraform module is a set of terraform configuration files in a single directory (module) / module/eks.

And every terraform configurations is a part of module folder

While creating a module it should be a centralized configuration and mostly use variables inside. Don't hardcode any values in module folder

**7.What is State File Locking?**

Answer: State file locking is a Terraform mechanism that prevents operations on a specific state file from being performed by multiple users at the same time.

**8.What is a Remote Backend in Terraform?**

Answer: [Terraform](https://k21academy.com/cloud-infrastructure-automation-certification-terraform-associate-self-study-training/) remote backend is used to store Terraform’s state and can also run operations in Terraform Cloud

**9.What are the most useful Terraform commands ?**

**Here are some useful Terraform Commands**

* **Fmt ( To Format the code correctly  )**
* **init**
* **validate**
* **plan**
* **apply**
* **destroy**
* **output**
* **show**
* **state**
* **version**

**10.How to destroy particular resources :**

-terraform state list

-terraform destroy -target aws\_instance.webservers[0]

**auto approving the commit:**

terraform apply -auto-approve

**Terraform:**

* Infra as a code tool
* It is an Open source tool
* It is a declarative type of language
* It is written in Golang and it is having its own syntax in HCL (Hashicorp configuration language)
* company Hashicorp
* It supports multi cloud platform (AWS,Azure,GCP)
* 28 July 2014
* It is designed on immutable infrastructure principle (Idempotency)

eg: 2 ec2 instances

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

Scenario based questions:

**setup:**

Dev

QA

Pre-prod

prod

1. We keep all the codes(tf files) in github -> Jenkins -> Deployments

1. **If a person is updating something manually in AWS but you want to adopt the changes in your state file ?**

terraform import

1. **How do you maintain your remote state file ?**

We maintain our state file in S3 and maintains the lock using dynamo DB

1. Did  you have experience in coding the full AWS configuration using terraform ?

EKS setup using terraform ?

1 . we use modules for this and we add :

-> in main.tf we will point the eks module

module :

1. create IAM role for eks cluster policy
2. Then create VPC
3. Create EKS and bind the IAM and VPC
4. then directly prepare worker node setup :

* create 3 iam roles eg: eks work node policy, CNI policy , container policy
* add the worker  node group to these 3 roles and connect it to the subnets
* create the autoscaling setup for this node group

1. How to create more number of service in AWS eg: parameter store , IAM

Terraform looping

1. How you will validate the code in terraform

* terraform plan
* terraform validate

1. how you will apply  same configuration to different environment ?

using modules