

# [By OnlineInterviewQuestions.com](https://www.onlineinterviewquestions.com)

## Terraform Interview Questions

### What is Terraform in DevOps?

**Terraform** is a tool from the toolchain of DevOps. This tool is used for development and change in the infrastructure while ensuring safety. Terraform aids organizations in automating and also help them with infrastructure as code. The extraordinary features of terraforming include its quickness and the operations it performs. The tool has been attracting popularity all over the world and has taken an essential stake in DevOps tools.

Here ask some important questions that have been carefully picked for your success in the Terraform interview. Do prepare these **Terraform Interview Questions** very dedicatedly in order to get ready for **cracking the Terraform Interview**. You can download **Terraform Interview Questions PDF** from the given link among the questions.

#### Q1. [What is Terraform?](#)

**Terraform** is a tool of the DevOps toolchain which is used for development and change in the infrastructure with safety. It is an efficient tool for the versioning of the infrastructure. This tool has been gaining popularity worldwide and has taken its stake in the chain of essential DevOps tools by many professionals. It is capable of managing solutions to custom in-house problems. It can also manage the famous service provider that still exists. Professionals prefer using Terraform due to a significant reason that is for the management of infrastructure as code. This is because of the extraordinary features of this tool which stands high in the race of other available options.

#### Q2. [Why should Terraform be preferred for DevOps?](#)

There are a bunch of reasons for giving preference to Terraform to be used as one of the significant tools of DevOps. The main motive of DevOps is to bring efficiency and quality in software delivery. For this, some tools are required for making the delivery smoother, faster and efficient. Here, terraform comes to the limelight where it aids organizations in automating and also aids with infrastructure as code. Terraform is helpful with its availability of implementing every type of coding principle. The extraordinary feature of terraforming includes its quickness and the operations performed by it. These are some of the important reasons for which Terraform is gaining popularity and attention in the organizations.

#### Q3. [What are the features of Terraform?](#)

Main key features of Terraform include :

- **Graphing** - Its features of graphing that are built-in are helpful in visualizing the infrastructure.
- **Custom Syntax** - It's custom syntax is very friendly which aids in enhancing efficiency.
- **Resource Relationships** - A very beneficial feature of terraforming is that it is able to understand resource relationships.
- **Updates** - The updates and features are added by the Open Source Project. It does so with a group of lots of contributors.
- **Improved Maintenance** - It is capable of breaking down the configuration into small parts or chunks for improving the organization and the maintenance.

#### **Q4. What steps should be followed for making an object of one module to be available for the other module at a high level?**

Following are the steps which should be followed for **making an object of one module** to be available for the other module at a high level: First, an output variable to be defined in a resource configuration. Till you do not declare resource configuration details, the scope of local and to a module. Now, you have to declare the output variable of module\_A to be used in another module's configuration. A brand new and latest key name should be created by you and the value should be kept equivalent to the module\_A's output variable.

Now, for module\_B you have to create a file variable.tf. Establish an input variable inside this file having exactly the same name as was in the key defined by you in module\_B. In a module, this particular variable enables the resource's dynamic configuration. For making this variable available to some other module also, replicate the process. This is because the particular variable established here has its scope restricted to module\_B.

#### **Q5. What are the new factors in the latest v1.24.0 & v1.25.0 Terraform Azure Provider?**

There are many new resources in the latest versions v1.24.0 & v1.25.0. Also, it involves new data resources. One of the the latest version is Azurerm\_batch\_certificate. It helps in managing the certificate in the Azure batch. This resource is used for managing public IP, the prefix in networking. New data resource in the latest version includes Azurerm\_firewall which allows access to the data for some already existing firewall. The latest release has also done work in the fixing of bugs. The azurerm\_app\_service has been enhanced which is a good sign of improvement.

#### **Q6. Are themes supported by terraforming?**

Yes. **Gtk themes** are efficiently supported by the 0.3.1 version of terraforming. If your system contains a gtk theme, then it can be enabled by performing this: cp/usr/wherever/THEMENAME/gtk/gtkrc \$HOME/.gtkrc If it shows error on opening the theme files and theme files fail to open, then .gtkrc should be edited and after that attach a line like the below one at the starting of the file. pixmap\_path"/usr/wherever/THEMENAME/gtk" Now the theme should be able to load on startup.

### Q7. Give a configuration of for creating a single E2C instance in Amazon Web Services ( AWS ).

```
provider "aws" {
  region = "ap-south-1"
}
resource "aws_instance"
"example" {
  ami = "ami-4fc58420"
  instance_type = "t2.micro"
  tags {
    Name = "terraform-example"
  }
}
```

### Q8. Why POVRay renders fields and sometimes just doesn't get displayed?

The reason for the failure of the default export into POVRay might be that the variant of your POVRay is of 3.0 version. The same could be the reason for POVRay not getting displayed without any report of an error. Terraform functions with 3.1 version and if you have a 3.0 version of POVRay, then you should use --pov30 switch for informing about this issue to terraform.

For checking your version of POVRay, you have to just type POVRay and look into output's first line. It will be having the version number in the first line itself.

### Q9. How will you check whether your POVRay install is ok with the terraform?

For checking whether my POVRay install is ok with terraforming, I will try this: **povray+I tf\_land.pov**

I may get either of the two outcomes from this. Either I will observe an error of this type: tf\_land.pov:26:error: Error opening TGA image.

**tf\_land.pov** refers to the file that is distributed in the root directory of terraforming. If I see the above type error, then it means that it is ok and the included file can be found by my POV. The second error type that I may observe will be about 'colors .inc'. This error means that the included files can't be found by my POV. This will help me in verifying if it is ok with terraforming or not.

**Q10. In case something went wrong, how will you handle and control rollbacks?**

I will recommit the previous version of the code to be the new and current version in my VCS. With this a terraform run will be triggered which will be responsible for running the old code. Don't forget that terraform is more declarative. Make sure that all the things provisioned in the code for rollback are contained in the old code. Ensure that it does not get destroyed on running the old code because of the lack of these. In case the state file got corrupted from a terraform run recently, then I will take the help of the State Rollback Feature of Terraform Enterprise for roll back to the previous latest state which was in a good state. This could be done due to the reason that every state change is versioned.

**Q11. Can policies be added to the Open Source or to the Pro Version of Enterprise?**

No. Policies cannot be added to the Open Source. Policies also cannot be added to the Enterprise Pro Version. Sentinel policies can only be accessed through the Premium Version of Terraform Enterprise.

**Q12. Mention the version controls that are supported besides GitHub?**

This is simple terraform interview questions yet should be taken into note significantly. GitLab CE and GitLab EE are supported. Bucket cloud is also supported.

**Q13. Can the Terraform Module Versions be locked anyhow?**

Yes, there is a way to lock it. if terraform module registry is being used as a source, then the 'version' attribute can be used in the module in a configuration file of Terraform. If as a source, a repository of Github is being used then in the query string, branch, and versions to be specified by '? ref '.

**Q14. On Azure, Can callbacks be used with terraforming?**

Yes, callbacks can be used on Azure with terraforming with the help of the Azure Event Hubs. The AzureRM provider of Terraform is today making this functionality easily available. Now the terraform actually comes already pre-installed itself in the portal of Azure Cloud Shell of Microsoft.

**Q15. Can terraform be used for on-prem infrastructure?**

Yes, Terraform can be used for on-prem infrastructure. There are many providers which are available. you can choose any one of them which suits you best. Many also build customer terraform providers for themselves, all needed is just an API.

**Q16. Is the management of Azure Availability Zones alike the management done by distinct cloud providers available?**

These zones are situated within an area and each zone has its separate source of power and network. Any region where availability zone is enabled has at least three such availability zones. The AzureRM Terraform provider have no resource accessible to manage the Azure Availability Zones at present

**Q17. What do you mean by IaC?**

IaC is an abbreviation to the term “Infrastructure as Code”. IaC refers to a system whereby developers can manage and provision the computer data centers automatically instead of getting into a manual process. Terraform, for instance, is an example tool of IaC.

**Q18. With the help of an example, explain the significance of 'read permission' on a workspace.**

For remote state functionality of terraforming, the workspace’s outputs can be 'read' which allows distinct workspaces to function and work jointly in collaboration. For instance, let's look into the management teams. There are two different teams. The first one is the networking team whose job is to manage the resources of networking in one single workspace. The second team is the development team.

This team manages resources of development in a different workspace. The development team had to provision resources in the VPC of the networking team. But the development team should focus on the management of development resources rather than modifying the infrastructure of networking. It provides the development team with 'read' permissions on the workspace of the networking team for reading the outputs exposed by the networking team for the provision of resources of development.

**Q19. While using TFE's API for the provision of resources, will history be the same as it is on the web?**

Yes. The history will be the same as it is on the web since the UI is developed keeping API as the base. So everything that is on the UI could be availed through the API and the other way around.

Please Visit [OnlineInterviewquestions.com](https://www.onlineinterviewquestions.com) to download more pdfs