Top Terraform Interview Questions in 2023

Description

Introduction

Since the emphasis on the cloud technologies have risen over the recent times, a wide variety of tools, technologies, languages have entered in the cloud space. One of the is Terraform. Terraform has become one of the top choices for dev ops engineers as far as infrastructure level code is concerned. In this post you will come across all types of important interview questions being asked when you are interviewed on Terraform. Lets get started.

What is Infrastructure as Code or IAC

As the name says, Infrastructure as code, it means creating your infrastructure with the help of code. Instead of creating the infra by web UI, people have started creating the infrastructure via code. Searching through so many options and creating an S3 bucket or an EC2 instance through UI is more complex than writing code which is even simpler. I have written an elaborate blogpost on Infrastructure as code.

What is your understanding on Terraform

Terraform is used to automate and manage your infrastructure, your platform, and services that run on the platform. Terraform is Infrastructure as a code tool that you can use to work with any of the the current available cloud providers platforms. Terraform is open source and uses Hashicorp configuration language (HCL) to define resources. Terraform makes sure that whatever code you write it will make sure that it will call the corresponding cloud provider API's and provision those resources. Sometimes the properties can be different because each cloud provider probably expects is own unique set of values to provision a source. We write code for multiple cloud providers in the same file and Terraform is responsible to create the resources with the right cloud providers.

What are the features of Terraform

- Terraform is open source.
- Terraform is used to describe your infrastructure in declarative configuration files that are human-readable.
- Implementing multi-cloud deployment environment is possible with less efforts.
- We can automate the infrastructure management using Terraform.
- Terraform's execution plan will ask for the final user consent before making any change to the infrastructure. This enables the user to evaluate the change one last time.
- Terraform has the ability to translate the Hashicorp configuration language (HCL) into Json.

Explain the workflow of the core terraform
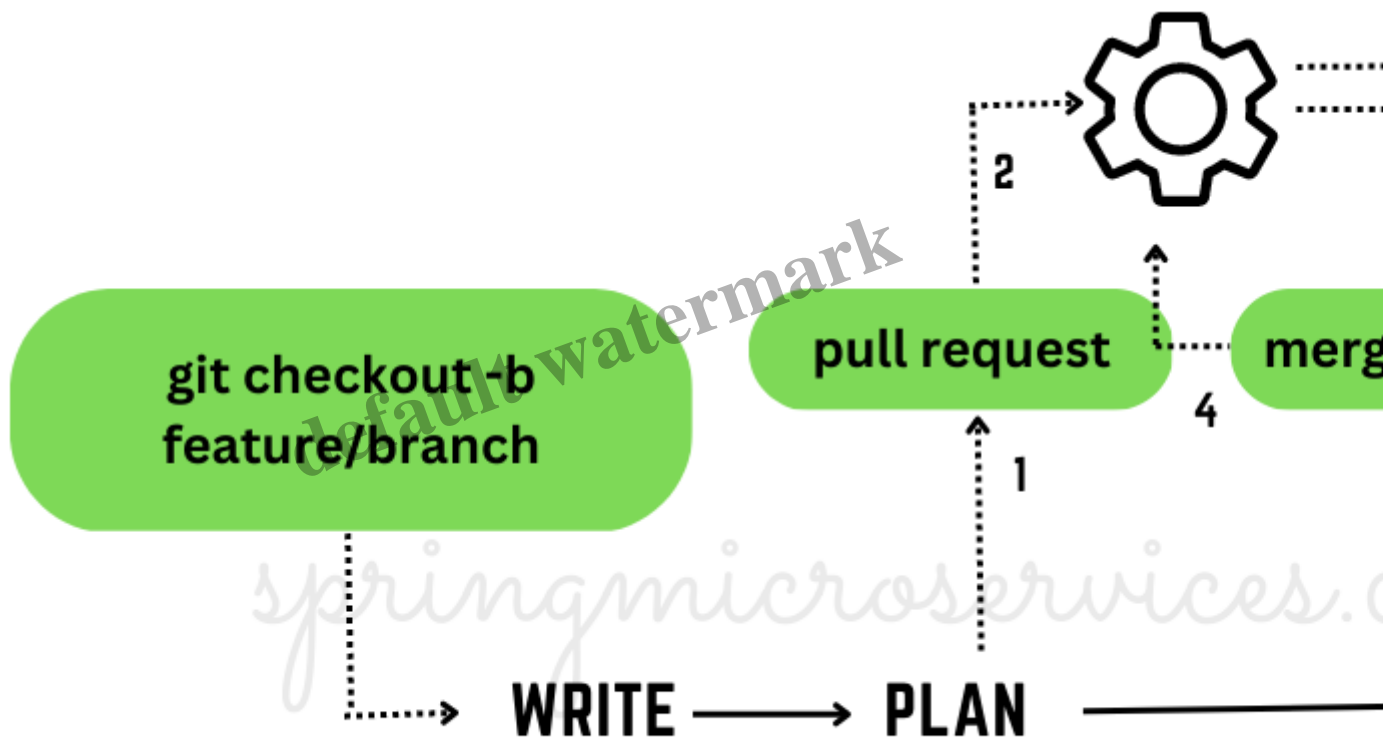
- **Write** – Create infrastructure in the form of code.
- **Plan** – This step will plan ahead of time to see how the changes look before they are implemented.
- **Apply** – Create a repeatable infrastructure.

What are the components present in Terraform architecture

Following are the components present in Terraform architecture.

- CLI (Command Line interface)
- Graph Walk
- Graph Builder

- State Manager
- Sub-graphs
- Expression Evaluation
- Vertex Evaluation
- Configuration Loader
- Backend

What are the most frequently used Terraform commands

Below are some of the most frequently used Terraform commands

- version
- show
- fmt
- init
- validate
- plan
- apply
- destroy
- output
- state

What is the purpose of Terraform in Devops

Terraform helps DevOps teams to define and enforce infrastructure configurations across multiple clouds and on-premises data centers.

What is Terraform init

Terraform init is used to initialize the current directory as a working directory where all the Terraform configurations files are present.

Syntax :

```
terraform init [options]
```

How to define a null resource in Terraform

Sometimes it€™s useful to have a resource that effectively creates nothing. Terraform provides such a resource called a null resource. A null resource is particularly useful for debugging modules. By adding a null resource, we can quickly echo out variables or runs.

The null_resource resource implements the standard resource lifecycle but takes no further action. The

triggers argument allows specifying an arbitrary set of values that, when changed, will cause the resource to be replaced. A null resource behaves like any other resource, you can configure provisioners, connection details, and other meta-parameters just like any other resource. This gives you more precise control over when provisioners execute in the dependency graph.

What is Terraform cloud

Terraform Cloud is software that enables teams to work together on Terraform. It provides features such as easy access to shared state and secret data, access controls for approving infrastructure changes, a private registry for sharing Terraform modules, detailed policy controls for governing the contents of Terraform configurations, and more to ensure that Terraform runs in a consistent and reliable environment.

Terraform Cloud is a hosted service that can be found at https://app.terraform.io. Terraform allows small teams to connect to version control, share variables, run Terraform in a reliable remote environment, and securely save remote state for free. Paid tiers provide you with the ability to add more than five people, establish teams with varying levels of access, enforce policies before building infrastructure, and work more efficiently.

What is Terraform backend

A backend defines where Terraform stores its state data files. By default, Terraform uses a backend called local which stores state as a local file on disk. Terraform keeps track of all the resources created in a state file

What are the alternatives of Terraform available in the market

- Ansible
- Kubernetes
- Azure Management Tools.
- Morpheus.
- CloudHealth.
- Turbonomic.
- CloudBolt.
- Apptio Cloudability
- Platform9 Managed Kubernetes.

What do you mean by Terraform CLI and some of the basic Terraform commands

The Terraform Command-Line Interface (CLI) is used to manage infrastructure and interact with Terraform state, configuration files, providers, etc.

What are Terraform modules

Modules can be thought of as a container consisting of various resources that work in collaboration. The root module includes resources mentioned in the .tf files and is required for every Terraform.

What is private module registry in Terraform

Private module registry in Terraform helps you to share Terraform providers and Terraform modules across the organization. It supports versioning and supportable list of available providers and modules. We can also apply rules on the registry which specifies how many members in the organization can use the modules.

Can we use Terraform for on premise infrastructure

Yes. we can use Terraform for on premise infrastructure

Name some of the built-in provisioners available in Terraform

Below are some of the built-in provisioners available in Terraform

- Salt-masterless Provisioner
- Puppet Provisioner
- File Provisioner
- Chef Provisioner
- Remote-exec Provisioner
- Local-exec Provisioner
- Habitat Provisioner

What is destroy command in the context of Terraform.

Terraform destroy command destroys the previously-created infrastructure or this command is used to quickly clean up all of those temporary objects.

```
terraform destroy [options]
```

Explain the usage of terraform validate command in the context of Terraform.

Terraform validate command is used to verify whether the configuration is valid or not. Validate checks if a configuration is correct syntactically. It is used for general verification of reusable modules, such as ensuring that attribute names and value types are correct. This command can be executed

Footer Tagline

automatically, for example as a post-save check in a text editor or as a test step for a reusable module in a continuous integration system.

Syntax:

```
terraform validate [options]
```

Explain the command terraform apply in the context of Terraform.

Terraform apply is used to create or update the infrastructure. This command is used in conjunction with the plan file.

Syntax:

```
terraform apply [options] [plan file]
```

Explain the command terraform version in the context of Terraform.

Terraform version command is used to see the current Terraform version as well as any installed plugins.

Syntax:

```
terraform version [options]
```

Explain the command terraform taint in the context of Terraform.

The terraform taint command informs Terraform that a particular object has become degraded or damaged. Terraform represents this by marking the object as "tainted" in the Terraform state, and Terraform will propose to replace it in the next plan you create.

Since this command is deprecated, it is recommended to use `-replace` option with terraform apply.

Syntax:

Footer Tagline

```
terraform taint [options] address
```

Explain the command terraform fmt in the context of Terraform.

The terraform fmt is used to rewrite Terraform configuration files to a canonical format and style. Other Terraform commands that generate Terraform configuration will produce configuration files that conform to the style imposed by terraform fmt, so using this style in your own files will ensure consistency.

Syntax:

```
terraform fmt [options] DIR
```

What is the command to destroys Terraform managed infrastructure

Below command is used to destroys Terraform managed infrastructure.

Syntax:

```
terraform destroy [options]
```

What is resource graph in Terraform

Resource graph in Terraform is used to provide visual representation of the resources. It helps in creating or modifying independent resources. Terraform generates plans and refreshes the state by creating a plan for the graphâ€™s configuration. It promptly and successfully builds structure to assist us in understanding the drawbacks.

What do you mean by Sentinel in Terraform

Sentinels are a powerful way to implement a variety of policies in Terraform. Some of the examples are as below.

- Restrict roles the cloud provider can assume
- Forbid only certain resources, providers, or data sources
- Enforce explicit ownership in resources
- Review an audit trail for Terraform Cloud operations
- Enforce mandatory tagging on resources
- Restrict how modules are used in the Private Module Registry

What is the way to store sensitive data in Terraform

Saving sensitive information like credentials in plain text on local drive or in the code repo can be very risky. We should use encrypted storage in order to store all your passwords, TLS certificates, SSH keys, etc.

What is State File Locking

State File Locking is used in order to prevent state file corruption. State file corruption happens when multiple users are doing the same task by blocking activities on a given state file. When a particular user releases the lock, only then the other one can operate on that state.

Explain Terragrunt and its uses

Terragrunt is a lightweight wrapper that adds extra features for maintaining DRY (Donâ€™t Repeat Yourself) configurations, dealing with many Terraform modules, and managing remote state.

Terragrunt is used for

- Keeping the Terraform code DRY
- Keeping our CLI flags DRY
- Working with multiple AWS accounts
- Executing Terraform commands on multiple modules
- Keeping the remote state configuration DRY

What is Tainted Resource in Terraform

When an object is degraded or damaged and when you want to notify Terraform about it, then we run below command.

```
terraform taint [options] address
```

Terraform represents it by marking the object as tainted in the Terraform state, and Terraform will replace it in the next plan whenever it is created. Complete and latest information about this command can be found in this link

Explain what is Terraform Core

Terraform Core is a statically compiled binary written in Go programming language. The compiled binary is the command line tool (CLI), the entrypoint for anyone using Terraform. The code is open source and hosted at github.com/hashicorp/terraform.

The primary responsibilities of Terraform Core are:

- Infrastructure as code: reading and interpolating configuration files and modules
- Resource state management.
- Construction of the Resource graph.
- Plan execution.
- Communication with plugins over RPC.

How to upgrade plugins in Terraform

```
terraform init -upgrade
```

-upgrade option is used to upgrade modules and plugins throughout the installation process. This command rechecks the releases.hashicorp.com to find new acceptable provider versions. It also downloads available provider versions.

Differentiate between Terraform and Cloudformation.

	Terraform	Cloudformation
Platform Support	Terraform supports various cloud providers such as AWS, GCP, Azure etc	CloudFormation is limited to AWS services
Language support	Terraform makes use of HCL language (Hashicorp Configuration Language). This language is also JSON compatible.	CloudFormation uses either JSON or YAML as a language.
Cost	Open source and completely free. Also has premium enterprise version.	Completely free. The only expense that consumers pay is for the AWS service that CloudFormation provides.

Footer Tagline

Differentiate between Terraform and Ansible.

Terraform	Ansible
Terraform is a tool that is designed to help with the provisioning and deprovisioning of cloud infrastructure using an infrastructure as code method.	Ansible is a more general configuration tool.
Terraform is a tool used for creating and managing IT infrastructure.	Ansible automates provisioning, deployment, and other IT processes.
It uses the declarative approach.	**Ansible automates provisioning, deployment, and other IT processes.** It uses the procedural approach.
It's ideal for orchestrating cloud services and building cloud infrastructure from the ground up.	It is mostly used to configure servers with the appropriate software and to update resources that have previously been configured.

Category

1. Cloud
2. Interview

Date Created
April 8, 2023
Author
kk-ravi144gmail-com