250 Practice Questions For Terraform Associate Certification

Read and Practice these questions before your exam

Bhargav Bachina

Terraform Certification

The Terraform Associate certification is for Cloud Engineers specializing in operations, IT, or developers who know the basic concepts and skills associated with open source HashiCorp Terraform. Candidates will be best prepared for this exam if they have professional experience using Terraform in production, but performing the exam objectives in a personal demo environment may also be sufficient.

Since this exam is multiple-choice, multiple-answer, and fill in the banks' questions, we need a lot of practice before the exam. This article helps you understand, practice, and get you ready for the exam. ***All the questions and answers are taken straight from their documentation. These are only practice questions.***

We are not going to discuss any concepts here, rather, I just want to create a bunch of practice questions for this exam based on the curriculum provided here.

- ***Understand infrastructure as code (IaC) concepts***
- ***Understand Terraform's purpose (vs other IaC)***
- ***Understand Terraform basics***
- ***Use the Terraform CLI (outside of core workflow)***
- ***Interact with Terraform modules***
- ***Navigate Terraform workflow***
- ***Implement and maintain state***
- ***Read, generate, and modify the configuration***
- ***Understand Terraform Cloud and Enterprise capabilities***

Understand infrastructure as code (IaC) concepts

Practice questions based on these concepts

- Explain what IaC is
- Describe the advantages of IaC patterns

1. What is Infrastructure as Code?

```
You write and execute the code to define, deploy, update,
```

2. What are the benefits of IaC?

```
a. Automation
We can bring up the servers with one script and scale up 
We can reuse the same codec. Versioning
We can check it into version control and we get versionin
```

3. How using IaC make it easy to provision infrastructure?

```
IaC makes it easy to provision and apply infrastructure c
```

4. What is Ideompodent in terms of IaC?

```
The idempotent characteristic provided by IaC tools ensur
```

5. What are Day 0 and Day 1 activities?

```
IaC can be applied throughout the lifecycle, both on the 
```

6. What are the use cases of Terraform?

```
Heroku App Setup
Multi-Tier Applications
Self-Service Clusters
```

```
Software Demos
Disposable Environments
Software Defined Networking
Resource Schedulers
Multi-Cloud Deploymenthttps://www.terraform.io/intro/use-
```

7. What are the advantages of Terraform?

```
Platform Agnostic
State Management
Operator Confidencehttps://learn.hashicorp.com/terraform/
```

8. Where do you describe all the components or your entire datacenter so that Terraform provision those?

```
Configuration files ends with *.tf
```

9. How can Terraform build infrastructure so efficiently?

```
Terraform builds a graph of all your resources, and paral
```

Understand Terraform's purpose (vs other IaC)

Practice questions based on these concepts

- Explain multi-cloud and provider-agnostic benefits
- Explain the benefits of state

10. What is multi-cloud deployment?

```
Provisoning your infrastrcutire into multiple cloud provi
```

11. How multi-cloud deployment is useful?

```
By using only a single region or cloud provider, fault to
```

12. What is cloud-agnostic in terms of provisioning tools?

```
cloud-agnostic and allows a single configuration to be us
```

13. Is Terraform cloud-agostic?

```
Yes
```

14. What is the use of terraform being cloud-agnostic?

```
It simplifies management and orchestration, helping opera
```

15. What is the Terraform State?

```
Every time you run Terraform, it records information abou
```

16. What is the purpose of the Terraform State?

```
Mapping to the Real World
Terraform requires some sort of database to map Terraform
Terraform must also track metadata such as resource depen
When running a terraform plan, Terraform must know the cu
When two people works on the same file and doing some cha
```

17. What is the name of the terraform state file?

```
terraform.tfstate
```

Understand Terraform basics

Practice questions based on these concepts

- Handle Terraform and provider installation and versioning
- Describe the plug-in based architecture
- Demonstrate using multiple providers
- Describe how Terraform finds and fetches providers
- Explain when to use and not use provisioners and when to use local-exec or remote-exec

18. How do you install terraform on different OS?

```
// Mac OS
brew install terraform// Windows
choco install terraformhttps://learn.hashicorp.com/terrafo
```

19. How do you manually install terraform?

```
step 1: Download the zip fille
step 2: mv ~/Downloads/terraform /usr/local/bin/terraform
```

20. Where do you put terraform configurations so that you can configure some behaviors of Terraform itself?

```
The special terraform configuration block type is used to
  # ...
```

```
}
```

21. Only constants are allowed inside the terraform block. Is this correct?

```
YesWithin a terraform block, only constant values can be 
```

22. What are the Providers?

```
A provider is a plugin that Terraform uses to translate t
```

23. How do you configure a Provider?

```
provider "google" {
  project = "acme-app"
  region  = "us-central1"
}The name given in the block header ("google" in this exa
```

24. What are the meta-arguments that are defined by Terraform itself and available for all `provider` blocks?

version: Constraining the allowed provider versions**alias:**

25. What is Provider initialization and why do we need?

```
Each time a new provider is added to configuration -- eit
```

26. How do you initialize any Provider?

Provider initialization is one of the actions of ***terrafor***

27. When you run `terraform init` command, all the providers are installed in the current working directory. Is this true?

```
Providers downloaded by terraform init are only installed
```

28. How do you constrain the provider version?

```
To constrain the provider version as suggested, add a requ
  required_providers {
    aws = "~> 1.0"
  }
}
```

29. How do you upgrade to the latest acceptable version of the provider?

```
terraform init --upgradeIt upgrade to the latest acceptab
This command also upgrades to the latest versions of all
```

30. How many ways you can configure provider versions?

```
1. With required_providers blocks under terraform blockte
  required_providers {
    aws = "~> 1.0"
  }
}2. Provider version constraints can also be specified us
  version= "1.0"
}
```

31. How do you configure Multiple Provider Instances?

aliasYou can optionally define multiple configurations fo

32. Why do we need Multiple Provider instances?

```
Some of the example scenarios:a. multiple regions for a c
b. targeting multiple Docker hosts
c. multiple Consul hosts, etc.
```

33. How do we define multiple Provider configurations?

```
To include multiple configurations for a given provider, 
provider "aws" {
  region = "us-east-1"
}

# Additional provider configuration for west coast region
provider "aws" {
  alias  = "west"
  region = "us-west-2"
}
```

34. How do you select alternate providers?

```
By default, resources use a default provider configuratio
  provider = aws.west

  # ...
}
```

35. What is the location of the user plugins directory?

```
Windows                            %APPDATA%\terraform.d\plugins
```

```
All other systems          ~/.terraform.d/plugins
```

36. Third-party plugins should be manually installed. Is that true?

```
True
```

37. The command `terraform init` cannot install third-party plugins? True or false?

```
TrueInstall third-party providers by placing their plugin
```

38. What is the naming scheme for provider plugins?

```
terraform-provider-<NAME>_vX.Y.Z
```

39. What is the CLI configuration File?

```
The CLI configuration file configures per-user settings fo
```

40. Where is the location of the CLI configuration File?

```
On Windows, the file must be named named terraform.rc and
```

41. What is Provider Plugin Cache?

```
By default, terraform init downloads plugins into a subdi
```

42. How do you enable Provider Plugin Cache?

```
To enable the plugin cache, use the plugin_cache_dir sett
```

43. When you are using plugin cache you end up growing cache directory with different versions. Whose responsibility to clean it?

```
UserTerraform will never itself delete a plugin from the 
```

44. Why do we need to initialize the directory?

```
When you create a new configuration — or check out an exi
  profile = "default"
  region  = "us-east-1"
}

resource "aws_instance" "example" {
  ami           = "ami-2757f631"
  instance_type = "t2.micro"
}Initializing a configuration directory downloads and ins
```

45. What is the command to initialize the directory?

```
terraform init
```

46. If different teams are working on the same configuration. How do you make files to have consistent formatting?

```
terraform fmtThis command automatically updates configura
```

47. If different teams are working on the same configuration. How do you make files to have syntactically valid and internally

consistent?

```
terraform validateThis command will check and report erro
```

48. What is the command to create infrastructure?

```
terraform apply
```

49. What is the command to show the execution plan and not apply?

```
terraform plan
```

50. How do you inspect the current state of the infrastructure applied?

```
terraform showWhen you applied your configuration, Terrafo
```

51. If your state file is too big and you want to list the resources from your state. What is the command?

```
terraform state listhttps://learn.hashicorp.com/terraform/
```

52. What is plug-in based architecture?

```
Defining additional features as plugins to your core plat
```

53. What are Provisioners?

If you need to do some initial setup on your instances, t

54. How do you define provisioners?

```
resource "aws_instance" "example" {
  ami           = "ami-b374d5a5"
  instance_type = "t2.micro"

  provisioner "local-exec" {
    command = "echo hello > hello.txt"
  }
}Provisioner block within the resource block. Multiple pr
```

55. What are the types of provisioners?

```
local-exec
remote-exec
```

56. What is a local-exec provisioner and when do we use it?

The local-exec provisioner executing a command locally on

57. What is a remote-exec provisioner and when do we use it?

Another useful provisioner is remote-exec which invokes a

58. Are provisioners runs only when the resource is created or destroyed?

Provisioners are only run when a resource is *created or d*

59. What do we need to use a remote-exec?

```
In order to use a remote-exec provisioner, you must choos
  profile = "default"
  region  = "us-west-2"
}resource "aws_key_pair" "example" {
  key_name   = "examplekey"
  public_key = file("~/.ssh/terraform.pub")
}resource "aws_instance" "example" {
  key_name      = aws_key_pair.example.key_name
  ami           = "ami-04590e7389a6e577c"
  instance_type = "t2.micro"connection {
    type        = "ssh"
    user        = "ec2-user"
    private_key = file("~/.ssh/terraform")
    host        = self.public_ip
  }provisioner "remote-exec" {
    inline = [
      "sudo amazon-linux-extras enable nginx1.12",
      "sudo yum -y install nginx",
      "sudo systemctl start nginx"
    ]
  }
}
```

60. When terraform mark the resources are tainted?

```
If a resource successfully creates but fails during provi
```

61. You applied the infrastructure with terraform apply and you have some tainted resources. You run an execution plan now what happens to those tainted resources?

```
When you generate your next execution plan, Terraform wil
```

62. Terraform also does not automatically roll back and destroy the resource during the apply when the failure happens. Why?

```
Terraform also does not automatically roll back and destr
```

63. How do you manually taint a resource?

```
terraform taint resource.id
```

64. Does the taint command modify the infrastructure?

```
terraform taint resource.idThis command will not modify i
```

65. By default, provisioners that fail will also cause the Terraform apply itself to fail. Is this true?

```
True
```

66. By default, provisioners that fail will also cause the Terraform apply itself to fail. How do you change this?

```
The on_failure setting can be used to change this. The al
  # ...

  provisioner "local-exec" {
    command  = "echo The server's IP address is ${self.pr
    on_failure = "continue"
  }
}
```

67. How do you define destroy provisioner and give an example?

```
You can define destroy provisioner with the parameter whe
    when = "destroy"

    # <...snip...>

}
```

68. How do you apply constraints for the provider versions?

```
The required_providers setting is a map specifying a vers
  required_providers {
    aws = ">= 2.7.0"
  }
}
```

69. What should you use to set both a lower and upper bound on versions for each provider?

```
~>terraform {
  required_providers {
    aws = "~> 2.7.0"
  }
}
```

70. How do you try experimental features?

```
In releases where experimental features are available, yo
  experiments = [example]
}
```

71. When does the terraform does not recommend using provisions?

```
Passing data into virtual machines and other compute reso
```

72. Expressions in `provisioner` blocks cannot refer to their parent resource by name. Is this true?

```
TrueThe self object represents the provisioner's parent r
```

73. What does this symbol version = "~> 1.0" mean when defining versions?

```
Any version more than 1.0 and less than 2.0
```

74. Terraform supports both cloud and on-premises infrastructure platforms. Is this true?

```
True
```

75. Terraform assumes an empty default configuration for any provider that is not explicitly configured. A provider block can be empty. Is this true?

```
True
```

76. How do you configure the required version of Terraform CLI can be used with your configuration?

```
The required_version setting can be used to constrain whi
```

77. Terraform CLI versions and provider versions are independent of each other. Is this true?

```
True
```

78. You are configuring aws provider and it is always recommended to hard code aws credentials in *.tf files. Is this true?

```
FalseHashiCorp recommends that you never hard-code creden
```

79. You are provisioning the infrastructure with the command terraform apply and you noticed one of the resources failed. How do you remove that resource without affecting the whole infrastructure?

```
You can taint the resource ans the next apply will destro
```

Use the Terraform CLI (outside of core workflow)

Practice questions based on these concepts

- Given a scenario: choose when to use `terraform fmt` to format code
- Given a scenario: choose when to use `terraform taint` to taint Terraform resources
- Given a scenario: choose when to use `terraform import` to import existing infrastructure into your Terraform state

- Given a scenario: choose when to use `terraform workspace` to create workspaces
- Given a scenario: choose when to use `terraform state` to view Terraform state
- Given a scenario: choose when to enable verbose logging and what the outcome/value is

80. What is command fmt?

```
The terraform fmt command is used to rewrite Terraform co
```

81. What is the recommended approach after upgrading terraform?

```
The canonical format may change in minor ways between Ter
```

82. What is the command usage?

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

83. By default, `fmt` scans the current directory for configuration files. Is this true?

```
TrueBy default, fmt scans the current directory for confi
```

84. You are formatting the configuration files and what is the flag you should use to see the differences?

```
terraform fmt -diff
```

85. You are formatting the configuration files and what is the flag you should use to process the subdirectories as well?

```
terraform fmt -recursive
```

86. You are formatting configuration files in a lot of directories and you don't want to see the list of file changes. What is the flag that you should use?

```
terraform fmt -list=false
```

87. What is the command taint?

```
The terraform taint command manually marks a Terraform-ma
```

88. What is the command usage?

```
terraform taint [options] addressThe address argument is
```

89. When you are tainting a resource terraform reads the default state file terraform.tfstate. What is the flag you should use to read from a different path?

```
terraform taint -state=path
```

90. Give an example of tainting a single resource?

```
terraform taint aws_security_group.allow_allThe resource
```

91. Give an example of tainting a resource within a module?

```
terraform taint "module.couchbase.aws_instance.cb_node[9]'
```

92. What is the command import?

```
The terraform import command is used to import existing r
```

93. What is the command import usage?

```
terraform import [options] ADDRESS ID
```

94. What is the default workspace name?

```
default
```

95. What are workspaces?

```
Each Terraform configuration has an associated backend th
```

96. What is the command to list the workspaces?

```
terraform workspace list
```

97. What is the command to create a new workspace?

```
terraform workspace new <name>
```

98. What is the command to show the current workspace?

```
terraform workspace show
```

99. What is the command to switch the workspace?

```
terraform workspace select <workspace name>
```

100. What is the command to delete the workspace?

```
terraform workspace delete <workspace name>
```

101. Can you delete the default workspace?

```
No. You can't ever delete default workspace
```

102. You are working on the different workspaces and you want to use a different number of instances based on the workspace. How do you achieve that?

```
resource "aws_instance" "example" {
  count = "${terraform.workspace == "default" ? 5 : 1}"

  # ... other arguments
}
```

103. You are working on the different workspaces and you want to use tags based on the workspace. How do you achieve that?

```
resource "aws_instance" "example" {
  tags = {
    Name = "web - ${terraform.workspace}"
  }

  # ... other arguments
}
```

104. You want to create a parallel, distinct copy of a set of infrastructure in order to test a set of changes before modifying the main production infrastructure. How do you achieve that?

```
Workspaces
```

105. What is the command state?

```
The terraform state command is used for advanced state ma
```

106. What is the command usage?

```
terraform state <subcommand> [options] [args]
```

107. You are working on terraform files and you want to list all the resources. What is the command you should use?

```
terraform state list
```

108. How do you list the resources for the given name?

```
terraform state list <resource name>
```

109. What is the command that shows the attributes of a single resource in the state file?

```
terraform state show 'resource name'
```

110. How do you do debugging terraform?

```
Terraform has detailed logs which can be enabled by setti
```

111. If terraform crashes where should you see the logs?

```
crash.logIf Terraform ever crashes (a "panic" in the Go r
```

112. What is the first thing you should do when the terraform crashes?

```
panic messageThe most interesting part of a crash log is
```

113. You are building infrastructure for different environments for example test and dev. How do you maintain separate states?

```
There are two primary methods to separate state between e
```
workspaces

114. What is the difference between directory-separated and workspace-separated environments?

```
Directory separated environments rely on duplicate Terraf
```

115. What is the command to pull the remote state?

```
terraform state pullThis command will download the state
```

116. What is the command is used manually to upload a local state file to a remote state

```
terraform state pushThe terraform state push command is u
```

117. The command terraform taint modifies the state file and doesn't modify the infrastructure. Is this true?

```
TrueThis command will not modify infrastructure, but does
```

118. Your team has decided to use terraform in your company and you have existing infrastructure. How do you migrate your existing resources to terraform and start using it?

```
You should use terraform import and modify the infrastrcu
```

119. When you are working with the workspaces how do you access the current workspace in the configuration files?

```
${terraform.workspace}
```

120. When you are using workspaces where does the Terraform save the state file for the local state?

```
terraform.tfstate.dFor local state, Terraform stores the
```

121. When you are using workspaces where does the Terraform save the state file for the remote state?

```
For remote state, the workspaces are stored directly in tl
```

122. How do you remove items from the Terraform state?

```
terraform state rm 'packet_device.worker'The terraform sta
```

123. How do you move the state from one source to another?

```
terraform state mv 'module.app' 'module.parent.module.app
```

124. How do you rename a resource in the terraform state file?

```
terraform state mv 'packet_device.worker' 'packet_device.l
```

Interact with Terraform modules

Practice questions based on these concepts

- Contrast module source options
- Interact with module inputs and outputs
- Describe variable scope within modules/child modules
- Discover modules from the public Terraform Module Registry
- Defining module version

125. Where do you find and explore terraform Modules?

```
The Terraform Registry makes it simple to find and use mo
```

126. How do you make sure that modules have stability and compatibility?

By default, only verified modules are shown in search resu

127. How do you download any modules?

You need to add any module in the configuration file like
```
  source = "hashicorp/consul/aws"
  version = "0.1.0"
}terraform init command will download and cache any module
```

128. What is the syntax for referencing a registry module?

```
<NAMESPACE>/<NAME>/<PROVIDER>// for example
module "consul" {
  source = "hashicorp/consul/aws"
  version = "0.1.0"
}
```

129. What is the syntax for referencing a private registry module?

```
<HOSTNAME>/<NAMESPACE>/<NAME>/<PROVIDER>// for example
module "vpc" {
  source = "app.terraform.io/example_corp/vpc/aws"
  version = "0.9.3"
}
```

130. The terraform recommends that all modules must follow semantic versioning. Is this true?

True

131. What is a Terraform Module?

A Terraform module is a set of Terraform configuration fi

132. Why do we use modules for?

∗ Organize configuration
∗ Encapsulate configuration
∗ Re-use configuration
∗ Provide consistency and ensure best practiceshttps://le

133. How do you call modules in your configuration?

Your configuration can use module blocks to call modules

134. How many ways you can load modules?

Local and remote modulesModules can either be loaded from

135. What are the best practices for using Modules?

1. Start writing your configuration with modules in mind.

136. What are the different source types for calling modules?

Local paths
Terraform Registry
GitHub
Generic Git, Mercurial repositories
Bitbucket
HTTP URLs

```
S3 buckets
GCS bucketshttps://www.terraform.io/docs/modules/sources.
```

137. What are the arguments you need for using modules in your configuration?

```
source and version// example
module "consul" {
  source = "hashicorp/consul/aws"
  version = "0.1.0"
}
```

138. How do you set input variables for the modules?

```
The configuration that calls a module is responsible for
```

For example, we have defined a lot of input variables for the modules such as ads, cidr, name, etc

main.tf

139. How do you access output variables from the modules?

```
You can access them by referringmodule.<MODULE NAME>.<OUT
```

140. Where do you put output variables in the configuration?

```
Module outputs are usually either passed to other parts o
```

outputs.tf

141. How do you pass input variables in the configuration?

```
You can define variables.tf in the root foldervariable "v
  description = "Name of VPC"
  type        = string
  default     = "example-vpc"
}Then you can access these varibles in the configuration
  source  = "terraform-aws-modules/vpc/aws"
  version = "2.21.0"

  name = var.vpc_name
  cidr = var.vpc_cidr

  azs             = var.vpc_azs
  private_subnets = var.vpc_private_subnets
  public_subnets  = var.vpc_public_subnets

  enable_nat_gateway = var.vpc_enable_nat_gateway

  tags = var.vpc_tags
}
```

142. What is the child module?

```
A module that is called by another configuration is somet
```

143. When you use local modules you don't have to do the command init or get every time there is a change in the local module. why?

```
When installing a local module, Terraform will instead re
```

144. When you use remote modules what should you do if there is a change in the module?

```
When installing a remote module, Terraform will download
```

145. A simple configuration consisting of a single directory with one or more .tf files is a module. Is this true?

```
True
```

146. When using a new module for the first time, you must run either `terraform init` or `terraform get` to install the module. Is this true?

```
True
```

147. When installing the modules and where does the terraform save these modules?

```
.terraform/modules// Example.terraform/modules
├── ec2_instances
│   └── terraform-aws-modules-terraform-aws-ec2-instance-
├── modules.json
└── vpc
    └── terraform-aws-modules-terraform-aws-vpc-2417f60
```

148. What is the required argument for the module?

```
sourceAll modules require a source argument, which is a m
```

149. What are the other optional meta-arguments along with the source when defining modules

```
version - (Optional) A version constraint string that spe
```

Navigate Terraform workflow

Practice questions based on these concepts

- Describe Terraform workflow (Write -> Plan -> Create)
- Initialize a Terraform working directory (terraform init)
- Validate a Terraform configuration (terraform validate)
- Generate and review an execution plan for Terraform (terraform plan)
- Execute changes to infrastructure with Terraform (terraform apply)
- Destroy Terraform managed infrastructure (terraform destroy)

150. What is the Core Terraform workflow?

```
The core Terraform workflow has three steps:1. Write - Au
2. Plan - Preview changes before applying.
3. Apply - Provision reproducible infrastructure.
```

151. What is the workflow when you work as an Individual Practitioner?

```
https://www.terraform.io/guides/core-workflow.html#workin
```

152. What is the workflow when you work as a team?

```
https://www.terraform.io/guides/core-workflow.html#workin
```

153. What is the workflow when you work as a large organization?

https://www.terraform.io/guides/core-workflow.html#the-co

154. What is the command init?

```
The terraform init command is used to initialize a workin
```

155. You recently joined a team and you cloned a terraform configuration files from the version control system. What is the first command you should use?

```
terraform initThis command performs several different ini
```

156. What is the flag you should use to upgrade modules and plugins a part of their respective installation steps?

```
upgradeterraform init -upgrade
```

157. When you are doing initialization with terraform init, you want to skip backend initialization. What should you do?

```
terraform init -backend=false
```

158. When you are doing initialization with terraform init, you want to skip child module installation. What should you do?

```
terraform init -get=false
```

159. When you are doing initialization where do all the plugins stored?

```
On most operationg systems :         ~/.terraform.d/plugi
on Windows                 :         %APPDATA%\terraform.
```

160. When you are doing initialization with terraform init, you want to skip plugin installation. What should you do?

```
terraform init -get-plugins=falseSkips plugin installatio
```

161. What does the command terraform validate does?

The ***terraform validate*** command validates the configuratio

162. What does the command plan do?

The ***terraform plan*** command is used to create an execution

163. What does the command apply do?

The ***terraform apply*** command is used to apply the changes

164. You are applying the infrastructure with the command apply and you don't want to do interactive approval. Which flag should you use?

terraform apply -auto-approvehttps://www.terraform.io/doc

165. What does the command destroy do?

The ***terraform destroy*** command is used to destroy the Terr

166. How do you preview the behavior of the command terraform destroy?

```
terraform plan -destroy
```

167. What are implicit and explicit dependencies?

Implicit dependency:
By studying the resource attributes used in interpolation
Sometimes there are dependencies between resources that a

168. Give an example of implicit dependency?

In the example below, the reference to **aws_instance.examp**

```
  profile    = "default"
  region     = "us-east-1"
}

resource "aws_instance" "example" {
  ami           = "ami-b374d5a5"
  instance_type = "t2.micro"
}resource "aws_eip" "ip" {
    vpc = true
    instance = aws_instance.example.id
}
```

169. Give an example of explicit dependency?

In the example below, an application we will run on our E

```
  bucket = "some_bucket"
  acl    = "private"
}resource "aws_instance" "example" {
  ami           = "ami-2757f631"
  instance_type = "t2.micro"

  depends_on = [aws_s3_bucket.example]
}
```

170. How do you save the execution plan?

```
terraform plan -out=tfplanyou can use that file with appl
```

171. You have started writing terraform configuration and you are using some sample configuration as a basis. How do you copy the example configuration into your working directory?

```
terraform init -from-module=MODULE-SOURCEhttps://www.terr
```

172. What is the flag you should use with the terraform plan to get detailed on the exit codes?

```
terraform plan -detailed-exitcodeReturn a detailed exit c
* 1 = Error
* 2 = Succeeded with non-empty diff (changes present)
```

173. How do you target only specific resources when you run a terraform plan?

```
-target=resource - A Resource Address to target. This fla
```

174. How do you update the state prior to checking differences

when you run a terraform plan?

```
terraform plan -refresh=true
```

175. The behavior of any `terraform destroy` command can be previewed at any time with an equivalent `terraform plan -destroy` command. Is this true?

```
True
```

176. You have the following file and created two resources docker_image and docker_container with the command `terraform apply` and you go to the terminal and delete the container with the command `docker rm`. You come back to your configuration and run the command again. Does terraform recreates the resource?

main.tf

```
Yes. Terrsform creates the resource again since the execu
```

177. You created a VM instance on AWS cloud provider with the terraform configuration and you log in AWS console and removed the instance. What does the next apply do?

```
It creates the instance again
```

178. You have the following file and created two resources docker_image and docker_container with the command `terraform plan`and you go to the terminal and delete the container with the command `docker rm`. You come back to your

configuration and run the command again. What is the output of the command plan?

```
Terraform will perform the following actions:# docker_con
```

Implement and maintain state

Practice questions based on these concepts

- Describe default local backend
- Outline state locking
- Handle backend authentication methods
- Describe remote state storage mechanisms and supported standard backends
- Describe the effect of Terraform refresh on state
- Describe backend block in configuration and best practices for partial configurations
- Understand secret management in state files

179. What are Backends?

```
A "backend" in Terraform determines how state is loaded a
```

180. What is local Backend?

```
The local backend stores state on the local filesystem, l
  backend "local" {
    path = "relative/path/to/terraform.tfstate"
  }
}
```

181. What is the default path for the local backend?

This defaults to "terraform.tfstate" relative to the root

182. What is State Locking?

If supported by your backend, Terraform will lock your st

183. Does Terraform continue if state locking fails?

No. If state locking fails, Terraform will not continue.

184. Can you disable state locking?

Yes. You can disable state locking for most commands with

185. What are the types of Backend?

Standard: State management, functionality covered in Stat

186. What are remote Backends?

Remote backends allow Terraform to use a shared storage s

187. What is the benefit of using remote backend?

Remote state storage makes collaboration easier and keeps

188. If you want to switch from using remote backend to local

backend. What should you do?

```
If you want to move back to local state, you can remove tl
```

189. What does the command refresh do?

```
The terraform refresh command is used to reconcile the sta
```

190. Does the command refresh modify the infrastructure?

The command ***refresh*** does not modify infrastructure, but do

191. How do you backup the state to the remote backend?

```
1. When configuring a backend for the first time (moving
```

192. What is a partial configuration in terms of configuring Backends?

```
You do not need to specify every required argument in the
```

193. What are the ways to provide remaining arguments when using partial configuration?

Interactively: Terraform will interactively ask you for tl

194. What is the basic requirement when using partial configuration?

```
When using partial configuration, Terraform requires at a
terraform {
  backend "consul" {}
}
```

195. Give an example of passing partial configuration with Command-line Key/Value pairs?

```
terraform init \
    -backend-config="address=demo.consul.io" \
    -backend-config="path=example_app/terraform_state" \
    -backend-config="scheme=https"
```

196. How to unconfigure a backend?

```
If you no longer want to use any backend, you can simply
```

197. How do you encrypt sensitive data in the state?

`Terraform Cloud always encrypts state at rest and protect`

198. Backends are completely optional. Is this true?

`Backends are completely optional.` `You can successfully us`

199. What are the benefits of Backends?

`Working in a team:` `Backends can store their state remotel`

200. Why should you be very careful with the Force unlocking

the state?

```
Terraform has a force-unlock command to manually unlock tl
```

201. You should only use force unlock command when automatic unlocking fails. Is this true?

```
True
```

Read, generate, and modify the configuration

Practice questions based on these concepts

- Demonstrate the use of variables and outputs
- Describe secure secret injection best practice
- Understand the use of the collection and structural types
- Create and differentiate resource and data configuration
- Use resource addressing and resource parameters to connect resources together
- Use Terraform built-in functions to write configuration
- Configure resource using a dynamic block
- Describe built-in dependency management (order of execution based)

202. How do you define a variable?

```
variable "region" {
  default = "us-east-1"
}This defines the region variable within your Terraform c
```

203. How do you access the variable in the configuration?

```
// accessing a variableprovider "aws" {
  region = var.region
}
```

204. How many ways you can assign variables in the configuration?

```
Command-line flagsterraform apply -var 'region=us-east-1'
  -var-file="secret.tfvars" \
  -var-file="production.tfvars"From environment varibles
If you execute terraform apply with any variable unspecif
```

205. Does environment variables support List and map types?

```
NoEnvironment variables can only populate string-type var
```

206. How do you provision infrastructure in a staging environment or a production environment using the same Terraform configuration?

```
You can use different varible files with the same configu
terraform apply -var-file="dev.tfvars"// For test
terraform apply -var-file="test.tfvars"
```

207. How do you assign default values to variables?

```
If no value is assigned to a variable via any of these me
  default = "us-east-1"
}
```

208. What are the data types for the variables?

```
string
number
boollist(<TYPE>)
set(<TYPE>)
map(<TYPE>)
object({<ATTR NAME> = <TYPE>, ... })
tuple([<TYPE>, ...])
```

209. Give an example of data type List variables?

```
Lists are defined either explicitly or implicitly.variabl
  type    = list(string)
  default = ["us-west-1a"]
}
```

210. Give an example of data type Map variables?

```
variable "region" {}
variable "amis" {
  type = map(string)
}amis = {
  "us-east-1" = "ami-abc123"
  "us-west-2" = "ami-def456"
}// accessing
resource "aws_instance" "example" {
  ami           = var.amis[var.region]
  instance_type = "t2.micro"
}
```

211. What is the Variable Definition Precedence?

The above mechanisms for setting variables can be used to

212. What are the output variables?

output variables as a way to organize data to be easily q

213. Hoe do you define an output variable?

```
output "ip" {
  value = aws_eip.ip.public_ip
}Multiple output blocks can be defined to specify multipl
```

214. How do you view outputs and queries them?

You will see the output when you run the following comman
terraform apply You can query the output with the followin
terraform output ip

215. What are the dynamic blocks?

some resource types include repeatable *nested blocks* in t

example using dynamic blocks

216. What are the best practices for dynamic blocks?

Overuse of dynamic blocks can make configuration hard to

217. What are the Built-in Functions?

```
The Terraform language includes a number of built-in func
```

218. Does Terraform language support user-defined functions?

```
NoThe Terraform language does not support user-defined fu
```

219. What is the built-in function to change string to a number?

`parseint` `parses the given string as a representation of a`
`100More Number Functions here`
https://www.terraform.io/docs/configuration/functions/abs

220. What is the built-in function to evaluates given expression and returns a boolean whether the expression produced a result without any errors?

```
cancondition     = can(formatdate("", var.timestamp))http
```

221. What is the built-in function to evaluates all of its argument expressions in turn and returns the result of the first one that does not produce any errors?

```
trylocals {
  example = try(
    [tostring(var.example)],
    tolist(var.example),
  )
}
```

222. What is Resource Address?

A **Resource Address** is a string that references a specific

223. What is the Module path?

A module path addresses a module within the tree of modul

224. What is the Resource spec?

```
A resource spec addresses a specific resource in the conf
  # ...
  count = 4
}aws_instance.web[3]  // Refers to only last instance
aws_instance.web     // Refers to all four "web" instance
  # ...
  for_each = {
    "terraform": "value1",
    "resource":  "value2",
    "indexing":  "value3",
    "example":   "value4",
  }
}aws_instance.web["example"] // Refers to only the "examp
```

225. What are complex types and what are the collection types Terraform supports?

A *complex* type is a type that groups multiple values into
collection types (for grouping similar values)* list(...)

226. What are the named values available and how do we refer to?

Terraform makes several kinds of named values available.

227. What is the built-in function that reads the contents of a file at the given path and returns them as a base64-encoded string?

```
filebase64(path)https://www.terraform.io/docs/configurati
```

228. What is the built-in function that converts a timestamp into a different time format?

```
formatdate(spec, timestamp)https://www.terraform.io/docs/
```

229. What is the built-in function encodes a given value to a string using JSON syntax?

```
jsonencode({"hello"="world"})https://www.terraform.io/doc
```

230. What is the built-in function that calculates a full host IP address for a given host number within a given IP network address prefix?

```
> cidrhost("10.12.127.0/20", 16)
10.12.112.16https://www.terraform.io/docs/configuration/f
```

Understand Terraform Cloud and Enterprise capabilities

Practice questions based on these concepts

- Describe the benefits of Sentinel, registry, and workspaces
- Differentiate OSS and Terraform Cloud workspaces

- Summarize features of Terraform Cloud

231. What is Sentinel?

```
Sentinel is an embedded policy-as-code framework integrat
```

232. What is the benefit of Sentinel?

```
Codifying policy removes the need for ticketing queues, w
```

233. What is the Private Module Registry?

```
Terraform Cloud's private module registry helps you share
```

234. What is the difference between public and private module registries when defined source?

```
The public registry uses a three-part <NAMESPACE>/<MODULE
  source  = "app.terraform.io/example_corp/vpc/aws"
  version = "1.0.4"
}
```

235. Where is the Terraform Module Registry available at?

https://registry.terraform.io/

236. What is a workspace?

```
A workspace contains everything Terraform needs to manage
```

237. What are the benefits of workspaces?

```
https://www.hashicorp.com/resources/terraform-enterprise-
```

238. You are configuring a remote backend in the terraform cloud. You didn't create an organization before you do terraform init. Does it work?

```
While the organization defined in the backend stanza must
```

239. You are configuring a remote backend in the terraform cloud. You didn't create a workspace before you do terraform init. Does it work?

```
Terraform Cloud will create it if necessary. If you opt t
```

240. Terraform workspaces when you are working with CLI and Terraform workspaces in the Terraform cloud. Is this correct?

```
If you are familiar with running Terraform using the CLI,
```

241. How do you authenticate the CLI with the terraform cloud?

```
Newer Versions:1. terraform login
2. it will open the terraform cloud and generate the toke
3. paste that token back in the CLIhttps://learn.hashicor
  token = "xxxxxx.atlasv1.zzzzzzzzzzzz"
}https://www.terraform.io/docs/commands/cli-config.html#c
```

242. You are building infrastructure on your local machine and

you changed your backend to remote backend with the Terraform cloud. What should you do to migrate the state to the remote backend?

```
terraform initOnce you have authenticated the remote back
```

243. How do you configure remote backend with the terraform cloud?

```
You need to configure in the terraform blockterraform {
  backend "remote" {
    hostname      = "app.terraform.io"
    organization  = "<YOUR-ORG-NAME>"

    workspaces {
      name = "state-migration"
    }
  }
}
```

244. What is Run Triggers?

```
Terraform Cloud’s run triggers allow you to link workspac
```

245. What is the benefit of Run Triggers?

```
When managing complex infrastructure with Terraform Cloud
```

246. What are the available permissions that terraform clouds can have?

```
Terraform Cloud teams can have read, plan, write, or admi
```

247. Who can grant permissions on the workspaces?

```
Organization owners grant permissions by grouping users i
```

248. Which plan do you need to manage teams on Terraform cloud?

```
Team Plan
```

249. How can you add users to an organization?

```
You can add users to an organization by inviting them usi
```

250. The Terraform Cloud Team plan charges you on a per-user basis. Is this true?

```
Yes. The Terraform Cloud Team plan is charged on a per-us
```

Conclusion

The Terraform associate exam is multiple-choice, multiple answers, text-based, exam. These sample questions definitely help you prepare for the certification. I would recommend you go through the documentation first and then refer to this afterward or right before the exam.