

# [Actual4Exams]



## Try before you buy

Download a free sample of any of our exam questions and answers

- ✓ 24/7 customer support, Secure shopping site
- ✓ Free One year updates to match real exam scenarios
- ✓ If you failed your exam after buying our products we will refund the full amount back to you.

## Choose an exam to sample

Select a vendor...

Select an exam...

Your email address

[Download Now](#)



Neil

" As a busy-working man I have no time and heart to prepare so I purchase braindumps for 117-010. I pass exam just one day's preparation. Great! "



Jodie

" The introduction of my friend said actual4exams is a good choice. The PDF &SOFT dumps on it are very good. So I came here and found that your guys are very kind. Then I decided to buy 117-010 exam dpf from you. I eventually passed the exam. Thanks "



Abraham

" I passed 117-010 easily. After using actual4exams pdf materials, I can say without any doubt that actual4exams is a very professional website that provides all of candidates with the excellent exam materials. Thank you guys "

## Why Choose Actual4Exams



### Quality and Value

Actual4Exams Practice Exams are written to the highest standards of technical accuracy, using only certified subject matter experts and published authors for development - no all vce.



### Tested and Approved

We are committed to the process of vendor and third party approvals. We believe professionals and executives alike deserve the confidence of quality coverage these authorizations provide.



### Easy to Pass

If you prepare for the exams using our Actual4Exams testing engine, It is easy to succeed for all certifications in the first attempt. You don't have to deal with all dumps or any free torrent / rapidshare all stuff.



### Try Before Buy

Actual4Exams offers free demo of each product. You can check out the interface, question quality and usability of our practice exams before you decide to buy.

<http://www.actual4exams.com>

Actual & valid exam test dumps for your successful pass

**Exam** : **TA-002-P**

**Title** : HashiCorp Certified:  
Terraform Associate

**Vendor** : HashiCorp

**Version** : DEMO

**NO.1** Which of the following state management command allow you to retrieve a list of resources that are part of the state file?

- A. terraform state list
- B. terraform state view
- C. terraform view
- D. terraform list

**Answer:** A

Explanation

The terraform state list command is used to list resources within a Terraform state.

Usage: terraform state list [options] [address...]

The command will list all resources in the state file matching the given addresses (if any). If no addresses are given, all resources are listed.

<https://www.terraform.io/docs/commands/state/list.html>

**NO.2** When TF\_LOG\_PATH is set, TF\_LOG must be set in order for any logging to be enabled.

- A. False
- B. True

**Answer:** B

Explanation

TF\_LOG\_PATH specifies where the log should persist its output to. Note that even when TF\_LOG\_PATH is set, TF\_LOG must be set in order for any logging to be enabled.

For example, to always write the log to the directory you're currently running terraform from:

```
export TF_LOG_PATH=./terraform.log
```

```
export TF_LOG=TRACE
```

**NO.3** Once a resource is marked as tainted, the next plan will show that the resource will be \_\_\_\_\_ and \_\_\_\_\_ and the next apply will implement this change.

- A. recreated and tainted
- B. destroyed and not recreated
- C. tainted and not destroyed
- D. destroyed and recreated

**Answer:** D

**NO.4** Which of the following variable definition files will terraform load automatically?

- A. terraform.tfvar
- B. Any files with names ending in .auto.tfvars.json
- C. terraform.tfvars
- D. terraform.tfvars.json

**Answer:** B C D

Explanation

Terraform also automatically loads a number of variable definitions files if they are present:

Files named exactly terraform.tfvars or terraform.tfvars.json.

Any files with names ending in .auto.tfvars or .auto.tfvars.json.

<https://www.terraform.io/docs/configuration/variables.html>

<https://www.terraform.io/docs/configuration/variables.html#variable-definitions-tfvars-files>

**NO.5** You have provisioned some aws resources in your test environment through Terraform for a POC work. After the POC, now you want to destroy the resources but before destroying them you want to check what resources will be getting destroyed through terraform. what are the options of doing that? (Select TWO)

- A. Use terraform destroy command
- B. This is not possible
- C. Use terraform plan command
- D. Use terraform plan -destroy command.

**Answer:** A D

Explanation

<https://learn.hashicorp.com/terraform/getting-started/destroy>

**NO.6** Which of the following represents a feature of Terraform Cloud that is NOT free to customers?

- A. Roles and Team Management
- B. WorkSpace Management
- C. Private Module Registry
- D. VCS Integration

**Answer:** A

Explanation

Role Based Access Controls (RBAC) for controlling permissions for who has access to what configurations within an organization and it is not free to customers.

<https://www.hashicorp.com/products/terraform/pricing/>

**NO.7** lookup retrieves the value of a single element from which of the below data type?

- A. map
- B. set
- C. string
- D. list

**Answer:** A

Explanation

<https://www.terraform.io/docs/configuration/functions/lookup.html>

**NO.8** After creating a new workspace "PROD" you need to run the command terraform select PROD to switch to it.

- A. False
- B. True

**Answer:** A

Explanation

By default, when you create a new workspace you are automatically switched to it To create a new workspace and switch to it, you can use terraform workspace new <new\_workspace\_name>; to switch to a existing workspace you can use terraform workspace select <existing\_workspace\_name>;

Example:

```
$ terraform workspace new example
```

Created and switched to workspace "example"!

You're now on a new, empty workspace. Workspaces isolate their state, so if you run "terraform plan" Terraform will not see any existing state for this configuration.

**NO.9** When using parent/child modules to deploy infrastructure, how would you export a value from one module to import into another module.

For example, a module dynamically deploys an application instance or virtual machine, and you need the IP address in another module to configure a related DNS record in order to reach the newly deployed application.

- A.** Export the value using terraform export and input the value using terraform input.
- B.** Configure the pertinent provider's configuration with a list of possible IP addresses to use.
- C.** Configure an output value in the application module in order to use that value for the DNS module.
- D.** Preconfigure the IP address as a parameter in the DNS module.

**Answer:** C

Explanation

Output values are like the return values of a Terraform module, and have several uses:

- \* A child module can use outputs to expose a subset of its resource attributes to a parent module.
- \* A root module can use outputs to print certain values in the CLI output after running terraform apply.
- \* When using remote state, root module outputs can be accessed by other configurations via a terraform\_remote\_state data source.

<https://www.terraform.io/docs/configuration/outputs.html>

**NO.10** What happens when a terraform apply command is executed?

- A.** Creates the execution plan for the deployment of resources.
- B.** Applies the changes required in the target infrastructure in order to reach the desired configuration.
- C.** The backend is initialized and the working directory is prepped.
- D.** Reconciles the state Terraform knows about with the real-world infrastructure.

**Answer:** B

Explanation

The terraform apply command is used to apply the changes required to reach the desired state of the configuration, or the pre-determined set of actions generated by a terraform plan execution plan.

<https://www.terraform.io/docs/commands/apply.html>

**NO.11** When using Terraform in a team it is important for everyone to be working with the same state so that operations will be applied to the same remote objects. Which of the below option is a recommended solution for this?

- A.** Remote State
- B.** Module
- C.** Use the cached state and treat this as the record of truth.
- D.** Workspace

**Answer:** A

Explanation

<https://www.terraform.io/docs/state/remote.html>

**NO.12** Which of the following challenges would Terraform be a candidate for solving? (Select THREE)

- A.** Enable self-service infrastructure to allocate resources on your proprietary private cloud.
- B.** Reduce the number of workflows needed for managing infrastructure across each of the companies public and private clouds.
- C.** Utilize a single tool for all of the infrastructure and configuration management needs.
- D.** Have a single interoperable tool to manage the variety of services including GitHub repositories, MySQL database, and Kubernetes clusters.

**Answer:** A B D

**NO.13** Valarie has created a database instance in AWS and for ease of use is outputting the value of the database password with the following code. Valarie wants to hide the output value in the CLI after terraform apply that's why she has used sensitive parameter.

```
1. output "db_password" {  
2. value = local.db_password  
3. sensitive = true  
4. }
```

Since sensitive is set to true, will the value associated with db password be available in plain-text in the state file for everyone to read?

**A.** Yes

**B.** No

**Answer:** A

Explanation

Outputs can be marked as containing sensitive material by setting the sensitive attribute to true, like this:

```
output "sensitive" {  
sensitive = true  
value = VALUE  
}
```

When outputs are displayed on-screen following a terraform apply or terraform refresh, sensitive outputs are redacted, with <sensitive> displayed in place of their value.

Limitations of Sensitive Outputs

The values of sensitive outputs are still stored in the Terraform state, and available using the terraform output command, so cannot be relied on as a sole means of protecting values.

Sensitivity is not tracked internally, so if the output is interpolated in another module into a resource, the value will be displayed.

**NO.14** What does the command terraform fmt do?

- A.** Rewrite Terraform configuration files to a canonical format and style.
- B.** Deletes the existing configuration file.
- C.** Updates the font of the configuration file to the official font supported by HashiCorp.
- D.** Formats the state file in order to ensure the latest state of resources can be obtained.

**Answer: A**

Explanation

The terraform fmt command is used to rewrite Terraform configuration files to a canonical format and style.

This command applies a subset of the Terraform language style conventions, along with other minor adjustments for readability.

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.

<https://www.terraform.io/docs/commands/fmt.html>

**NO.15** You have created a custom variable definition file testing.tfvars. How will you use it for provisioning infrastructure?

**A.** terraform apply -var-state-file="testing.tfvars"

**B.** terraform plan -var-file="testing.tfvar"

**C.** terraform apply -var-file="testing.tfvars"

**D.** terraform apply var-file="testing.tfvars"

**Answer: C**

Explanation

<https://www.terraform.io/docs/configuration/variables.html>

**NO.16** Your manager has instructed you to start using terraform for the entire infra provisioning of the application stack. There are 4 environments - DEV , QA , UAT , and PROD. The application team has asked for complete segregation between these environments including the backend , state , and also configurations ,since there will be unique resources in different environments . What is the possible way to structure the terraform code to facilitate that.

**A.** Completely separate the working directories , keep one for each environment . For each working directory , maintain a separate configuration file , variables file , and map to a different backend.

**B.** Completely separate the working directories , keep one for each environment . For each working directory , maintain a separate configuration file , variables file , and map to the same backend.

**C.** Implement terraform workspaces , and map each environment with one workspace.

**D.** Enable remote backend storage . Configure 4 different backend storages , one for each environment.

**Answer: A**

Explanation

In particular, organizations commonly want to create a strong separation between multiple deployments of the same infrastructure serving different development stages (e.g. staging vs. production) or different internal teams. In this case, the backend used for each deployment often belongs to that deployment, with different credentials and access controls. Named workspaces are not a suitable isolation mechanism for this scenario.

<https://www.terraform.io/docs/state/workspaces.html>

**NO.17** You are using a terraform operation that writes state. Unfortunately automatic state unlocking has failed for that operation. Which of the below commands can be used to remove the already acquired lock on the state?



- A. terraform unlock
- B. terraform force-unlock
- C. terraform state unlock
- D. None of the above

**Answer:** B

Explanation

Command: force-unlock

Manually unlock the state for the defined configuration.

This will not modify your infrastructure. This command removes the lock on the state for the current configuration. The behavior of this lock is dependent on the backend being used. Local state files cannot be unlocked by another process.

<https://www.terraform.io/docs/commands/force-unlock.html>

<https://www.terraform.io/docs/state/locking.html>

Terraform has a force-unlock command to manually unlock the state if unlocking failed.

If you unlock the state when someone else is holding the lock it could cause multiple writers. Force unlock should only be used to unlock your own lock in the situation where automatic unlocking failed.

**NO.18** You wanted to destroy some of the dependent resources from real infrastructure. You choose to delete those resources from your configuration file and run terraform plan and then apply. Which of the following way your resources would be destroyed?

- A. Terraform can still determine the correct order for destruction from the state even when you delete one or more items from the configuration.
- B. Those would be destroyed in the order in which they were written in the configuration file previously before you have deleted them from configuration file.
- C. The resource will be destructed in random order as you have already deleted them from configuration.
- D. You can not destroy resources by deleting them from configuration file and running plan and apply.

**Answer:** A

Explanation

Terraform typically uses the configuration to determine dependency order. However, when you delete a resource from a Terraform configuration, Terraform must know how to delete that resource. Terraform can see that a mapping exists for a resource not in your configuration and plan to destroy. However, since the configuration no longer exists, the order cannot be determined from the configuration alone.

To ensure correct operation, Terraform retains a copy of the most recent set of dependencies within the state.

Now Terraform can still determine the correct order for destruction from the state when you delete one or more items from the configuration.