



**Free Questions for TA-002-P by ebraindumps**

**Shared by Sears on 15-04-2024**

**For More Free Questions and Preparation Resources**

**Check the Links on Last Page**

# Question 1

---

## Question Type: MultipleChoice

---

You have a Terraform configuration that defines a single virtual machine with no references to it, You have run terraform apply to create the resource, and then removed the resource definition from your Terraform configuration file.

What will happen you run terraform apply in the working directory again?

### Options:

---

- A- Terraform will remove the virtual machine from the state file, but the resource will still exist
- B- Nothing
- C- Terraform will error
- D- Terraform will destroy the virtual machine

This is what will happen if you run terraform apply in the working directory again, after removing the resource definition from your Terraform configuration file. Terraform will detect that there is a resource in the state file that is not present in the configuration file, and will assume that you want to delete it.

### Answer:

---

D

## Question 2

---

**Question Type:** MultipleChoice

---

How can a ticket-based system slow down infrastructure provisioning and limit the ability to scale? Choose two correct answers.

### Options:

---

- A-** End-users have to request infrastructure changes
- B-** Ticket based systems generate a full audit trail of the request and fulfillment process
- C-** Users can access catalog of approved resources from drop down list in a request form
- D-** The more resources your organization needs, the more tickets your infrastructure team has to process

These are some of the ways that a ticket-based system can slow down infrastructure provisioning and limit the ability to scale, as they introduce delays, bottlenecks, and manual interventions in the process of creating and modifying infrastructure.

### Answer:

---

A

## Question 3

---

**Question Type: MultipleChoice**

---

You have created a main.tf Terraform configuration consisting of an application server, a database and a load balanced. You ran terraform apply and Terraform created all of the resources successfully.

Now you realize that you do not actually need the load balancer, so you run terraform destroy without any flags. What will happen?

**Options:**

---

- A- Terraform will prompt you to pick which resource you want to destroy
- B- Terraform will destroy the application server because it is listed first in the code
- C- Terraform will prompt you to confirm that you want to destroy all the infrastructure
- D- Terraform will destroy the main, tf file
- E- Terraform will immediately destroy all the infrastructure

This is what will happen if you run terraform destroy without any flags, as it will attempt to delete all the resources that are associated with your current working directory or workspace. You can use the -target flag to specify a particular resource that you want to destroy.

**Answer:**

---

C

## Question 4

---

**Question Type:** MultipleChoice

---

Terraform variable names are saved in the state file.

### Options:

---

**A-** True

**B-** False

Terraform variable names are not saved in the state file, only their values are. The state file only stores the attributes of the resources and data sources that are managed by Terraform, not the variables that are used to configure them.

### Answer:

---

B

## Question 5

---

**Question Type:** MultipleChoice

---

When using multiple configuration of the same Terraform provider, what meta-argument must you include in any non-default provider configurations?

**Options:**

---

A- Alias

B- Id

C- Depends\_on

D- name

This is the meta-argument that you must include in any non-default provider configurations, as it allows you to give a friendly name to the configuration and reference it in other parts of your code. The other options are either invalid or irrelevant for this purpose.

**Answer:**

---

A

## Question 6

---

**Question Type:** MultipleChoice

---

A Terraform provider is NOT responsible for:

### Options:

---

- A- Exposing resources and data sources based on an APUI
- B- Managing actions to take based on resources differences
- C- Understanding API interactions with some service
- D- Provisioning infrastructure in multiple

This is not a responsibility of a Terraform provider, as it does not make sense grammatically or logically. A Terraform provider is responsible for exposing resources and data sources based on an API, managing actions to take based on resource differences, and understanding API interactions with some service.

### Answer:

---

D

## Question 7

---

### Question Type: MultipleChoice

---

In a Terraform Cloud workspace linked to a version control repository speculative plan run start automatically commit changes to version control.

**Options:**

---

A- True

B- False

When you use a remote backend that needs authentication, HashiCorp recommends that you:

**Answer:**

---

A

## Question 8

---

**Question Type:** MultipleChoice

---

Which command add existing resources into Terraform state?

**Options:**

---

A- Terraform init

B- Terraform plan

C- Terraform refresh



**D-** Terraform import

**E-** All of these

This is the command that can add existing resources into Terraform state, by matching them with the corresponding configuration blocks in your files.

**Answer:**

---

D

## Question 9

---

**Question Type:** MultipleChoice

---

How does the Terraform cloud integration differ from other state backends such as S3, Consul,etc?

**Options:**

---

**A-** It can execute Terraform runs on dedicated infrastructure in Terraform Cloud

**B-** It doesn't show the output of a terraform apply locally

**C-** It is only arable lo paying customers

**D-** All of the above

This is how the Terraform Cloud integration differs from other state backends such as S3, Consul, etc., as it allows you to perform remote operations on Terraform Cloud's servers instead of your local machine. The other options are either incorrect or irrelevant.

**Answer:**

---

A

## Question 10

---

**Question Type:** MultipleChoice

---

Setting the TF\_LOG environment variable to DEBUG causes debug messages to be logged into stdout.

**Options:**

---

**A-** True

**B-** False

Setting the TF\_LOG environment variable to DEBUG causes debug messages to be logged into stdout, along with other log levels such as TRACE, INFO, WARN, and ERROR. This can be useful for troubleshooting or debugging purposes.

**Answer:**

---

A

**To Get Premium Files for TA-002-P Visit**

<https://www.p2pexams.com/products/ta-002-p>

**For More Free Questions Visit**

<https://www.p2pexams.com/hashicorp/pdf/ta-002-p>

