



Free Questions for [VA-002-P](#) by [certsinside](#)

Shared by [Sheppard](#) on [29-01-2024](#)

For More Free Questions and Preparation Resources

[Check the Links on Last Page](#)

Question 1

Question Type: MultipleChoice

Your organization has moved to AWS and has manually deployed infrastructure using the console. Recently, a decision has been made to standardize on Terraform for all deployments moving forward.

What can you do to ensure that all existing is managed by Terraform moving forward without interruption to existing services?

Options:

- A- resources that are manually deployed in the AWS console cannot be imported by Terraform
- B- using terraform import, import the existing infrastructure into your Terraform state
- C- delete the existing resources and recreate them using new a Terraform configuration so Terraform can manage them moving forward
- D- submit a ticket to AWS and ask them to export the state of all existing resources and use terraform import to import them into the state file

Answer:

B

Explanation:

Terraform is able to import existing infrastructure. This allows you to take resources you've created by some other means and bring it under Terraform management.

This is a great way to slowly transition infrastructure to Terraform or to be sure you're confident that you can use Terraform in the future if it currently doesn't support every AWS service or feature you need today.

Question 2

Question Type: MultipleChoice

When multiple engineers start deploying infrastructure using the same state file, what is a feature of remote state storage that is critical to ensure the state does not become corrupt?

Options:

- A- state locking
- B- object storage
- C- encryption

D- workspaces

Answer:

A

Explanation:

If supported by your backend, Terraform will lock your state for all operations that could write state. This prevents others from acquiring the lock and potentially corrupting your state.

State locking happens automatically on all operations that could write state. You won't see any message that it is happening. If state locking fails, Terraform will not continue. You can disable state locking for most commands with the `-lock` flag but it is not recommended.

Question 3

Question Type: MultipleChoice

Which of the following allows Terraform users to apply policy as code to enforce standardized configurations for resources being deployed via infrastructure as code?

Options:

- A- functions
- B- workspaces
- C- module registry
- D- sentinel

Answer:

D

Explanation:

Sentinel is an embedded policy-as-code framework integrated with the HashiCorp Enterprise products. It enables fine-grained, logic-based policy decisions, and can be extended to use information from external sources.

Question 4

Question Type: MultipleChoice

Which of the following actions are performed during a terraform init? (select three)

Options:

- A- provisions the declared resources in your configuration
- B- download the declared providers which are supported by HashiCorp
- C- initializes the backend configuration
- D- initializes downloaded and/or installed providers

Answer:

B, C, D

Explanation:

The terraform init command is used to initialize a working directory containing Terraform configuration files. This is the first command that should be run after writing a new Terraform configuration or cloning an existing one from version control. It is safe to run this command multiple times.

Question 5

Question Type: MultipleChoice

True or False: You can migrate the Terraform backend but only if there are no resources currently being managed.

Options:

A- False

B- True

Answer:

A

Explanation:

If you are already using Terraform to manage infrastructure, you probably want to transfer to another backend, such as Terraform Cloud, so you can continue managing it. By migrating your Terraform state, you can hand off infrastructure without de-provisioning anything.

Question 6

Question Type: MultipleChoice

Which of the following connection types are supported by the remote-exec provisioner? (select two)

Options:

A- rdp

B- smb

C- ssh

D- winrm

Answer:

C, D

Explanation:

The remote-exec provisioner invokes a script on a remote resource after it is created. The remote-exec provisioner supports both ssh and winrm type connections.

Question 7

Question Type: MultipleChoice

What is the best and easiest way for Terraform to read and write secrets from HashiCorp Vault?

Options:

- A- CLI access from the same machine running Terraform
- B- API access using the AppRole auth method
- C- Vault provider
- D- Integration with a tool like Jenkins

Answer:

C

Explanation:

The Vault provider allows Terraform to read from, write to, and configure Hashicorp Vault.

Question 8

Question Type: MultipleChoice

What Terraform command can be used to inspect the current state file?

```
# aws_instance.example:
resource "aws_instance" "example" {
  ami                = "ami-2757f631"
  arn                = "arn:aws:ec2:us-east-1:130490850807:instance/i-0
  associate_public_ip_address = true
  availability_zone  = "us-east-1c"
  cpu_core_count    = 1
  cpu_threads_per_core = 1
  disable_api_termination = false
  ebs_optimized     = false
  get_password_data  = false
  id                = "i-0bbf06244e44211d1"
  instance_state    = "running"
  instance_type     = "t2.micro"
```

Options:

A- terraform inspect

B- terraform show

C- terraform read

D- terraform state

Answer:

B

Explanation:

The terraform show command is used to provide human-readable output from a state or plan file. This can be used to inspect a plan to ensure that the planned operations are expected, or to inspect the current state as Terraform sees it.

Machine-readable output can be generated by adding the -json command-line flag.

Note: When using the -json command-line flag, any sensitive values in Terraform state will be displayed in plain text.

To Get Premium Files for VA-002-P Visit

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

For More Free Questions Visit

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

