

HashiCorp.TA-002-P.v2022-08-10.q213

Exam Code:	TA-002-P
Exam Name:	HashiCorp Certified: Terraform Associate
Certification Provider:	HashiCorp
Free Question Number:	213
Version:	v2022-08-10
# of views:	3028
# of Questions views:	2130
https://www.freeqas.com/qa/HashiCorp/TA-002-P/HashiCorp.TA-002-P.v2022-08-10.q213.html	

NEW QUESTION: 1

During a terraform apply, a resource is successfully created but eventually fails during provisioning. What happens to the resource?

- A. The resource will be planned for destruction and recreation upon the next terraform apply
- B. Terraform will retry to provision again.
- C. The failure of provisioner will be ignored and it will not cause a failure to terraform apply
- D. The resource will be automatically destroyed.

Answer: A (LEAVE A REPLY)

If a creation-time provisioner fails, the resource is marked as tainted. A tainted resource will be planned for destruction and recreation upon the next terraform apply. Terraform does this because a failed provisioner can leave a resource in a semi-configured state. Because Terraform cannot reason about what the provisioner does, the only way to ensure proper creation of a resource is to recreate it. This is tainting.

You can change this behavior by setting the `on_failure` attribute, which is covered in detail below.

<https://www.terraform.io/docs/provisioners/index.html#creation-time-provisioners>

<https://www.terraform.io/docs/provisioners/index.html#destroy-time-provisioners>

<https://www.terraform.io/docs/provisioners/index.html#failure-behavior>

NEW QUESTION: 2

You have provisioned some virtual machines (VMs) on Google Cloud Platform (GCP) using the `gcloud` command line tool. However, you are standardizing with Terraform and want to manage these VMs using Terraform instead.

What are the two things you must do to achieve this? (Choose two.)

- A. Provision new VMs using Terraform with the same VM names
- B. Use the terraform import command for the existing VMs
- C. Write Terraform configuration for the existing VMs
- D. Run the terraform import-gcp command

Answer: B,D ([LEAVE A REPLY](#))

The terraform import command is used to import existing infrastructure. Import existing Google Cloud resources into Terraform with Terraformer.

NEW QUESTION: 3

When using remote state, state is only ever held in memory when used by Terraform.

- A. False
- B. True

Answer: B ([LEAVE A REPLY](#))

NEW QUESTION: 4

When using a module block to reference a module stored on the public Terraform Module Registry such as:

How do you specify version 1.0.0?

- A. Modules stored on the public Terraform Module Registry do not support versioning
- B. Append ?ref=v1.0.0 argument to the source path
- C. Add version = "1.0.0" attribute to module block
- D. Nothing - modules stored on the public Terraform Module Registry always default to version 1.0.0

Answer: A ([LEAVE A REPLY](#))

NEW QUESTION: 5

Which of the following is not a valid Terraform string function?

- A. replace
- B. format
- C. join
- D. tostring

Answer: D ([LEAVE A REPLY](#))

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

NEW QUESTION: 6

True or False? Each Terraform workspace uses its own state file to manage the infrastructure associated with that particular workspace.

- A. False
- B. True

Answer: ([SHOW ANSWER](#))

The persistent data stored in the backend belongs to a workspace. Initially, the backend has only one workspace, called "default", and thus there is only one Terraform state associated with that configuration.

NEW QUESTION: 7

You want to define multiple data disks as nested blocks inside the resource block for a virtual machine. What Terraform feature would help you define the blocks using the values in a variable?

- A. Dynamic blocks
- B. Collection functions
- C. Local values
- D. Count arguments

Answer: A ([LEAVE A REPLY](#))

NEW QUESTION: 8

Eric needs to make use of module within his terraform code. Should the module always be public and open-source to be able to be used?

- A. False
- B. True

Answer: A ([LEAVE A REPLY](#))

Terraform module need not be public and open-source. Module can be placed in -

- * Local paths
- * Terraform Registry
- * GitHub
- * Bitbucket
- * Generic Git, Mercurial repositories
- * HTTP URLs
- * S3 buckets
- * GCS buckets

<https://www.terraform.io/docs/modules/sources.html>

NEW QUESTION: 9

A user has created three workspaces using the command line - prod, dev, and test. The user wants to create a fourth workspace named stage. Which command will the user execute to accomplish this?

- A. terraform workspace new stage
- B. terraform workspace -new stage
- C. terraform workspace -create stage
- D. terraform workspace create stage

Answer: A ([LEAVE A REPLY](#))

The terraform workspace new command is used to create a new workspace.

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

NEW QUESTION: 10

You have created an AWS EC2 instance of type t2.micro through your terraform configuration file ec2.tf .

Now you want to change the instance type from t2.micro to t2.medium. Accordingly you have changed your configuration file and ran terraform plan. After running terraform plan you check the output and saw one instance will be updated from t2.micro --> t2.medium. After this you went to grab a coffee without running terraform apply and meanwhile a member of your team changed the instance type of that EC2 instance to t2.medium from aws console. After coming to your desk you run terraform apply. What will happen?

- A. terraform apply will through an error.
- B. No resource will be updated and you will see the message : Apply Complete ! Resources : 0 added, 0 changed, 0 destroyed.
- C. 1 resource will be updated and you will see the message : Apply Complete ! Resources : 0 added, 1 changed, 0 destroyed.
- D. The instance type will be changed to t2.micro and again will be changed to t2.medium

Answer: B ([LEAVE A REPLY](#))

NEW QUESTION: 11

The Terraform language does not support user-defined functions, and so only the functions built in to the language are available for use.

- A. False
- B. True

Answer: ([SHOW ANSWER](#)**)**

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

NEW QUESTION: 12

Which of these is the best practice to protect sensitive values in state files?

- A. Blockchain
- B. Secure Sockets Layer (SSL)
- C. Enhanced remote backends
- D. Signed Terraform providers

Answer: C ([LEAVE A REPLY](#))

Explanation

Use of remote backends and especially the availability of Terraform Cloud, there are now a variety of backends that will encrypt state at rest and will not store the state in cleartext on machines running. Reference:

<https://www.terraform.io/docs/extend/best-practices/sensitive-state.html>

NEW QUESTION: 13

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: (SHOW ANSWER)

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>

NEW QUESTION: 14

Your company has been using Terraform Cloud for a some time now . But every team is creating their own modules , and there is no standardization of the modules , with each team creating the resources in their own unique way . You want to enforce a standardization of the modules across the enterprise . What should be your approach.

- A. Create individual workspaces for each team , and ask them to share modules across workspaces.
- B. Implement a Private module registry in Terraform cloud , and ask teams to reference them.
- C. Upgrade to Terraform enterprise , since this is not possible in terraform cloud.
- D. Upload the modules in the terraform public module registry , and ask teams to reference them

Answer: B (LEAVE A REPLY)

Terraform Cloud's private module registry helps you share Terraform modules across your organization. It includes support for module versioning, a searchable and filterable list of available modules, and a configuration designer to help you build new workspaces faster.

By design, the private module registry works much like the public Terraform Registry. If you're already used the public registry, Terraform Cloud's registry will feel familiar.

Understand the different offerings in Terraform OS, Terraform Cloud and Terraform Enterprise.

Terraform Cloud's private module registry helps you share Terraform modules across your organization.

<https://www.terraform.io/docs/cloud/registry/index.html>

<https://www.terraform.io/docs/cloud/registry/publish.html>

NEW QUESTION: 15

Which of the following is not true of Terraform providers?

- A. Providers can be maintained by a community of users
- B. None of the above
- C. Some providers are maintained by HashiCorp
- D. Major cloud vendors and non-cloud vendors can write, maintain, or collaborate on Terraform providers
- E. Providers can be written by individuals

Answer: D (LEAVE A REPLY)

NEW QUESTION: 16

Which of the below configuration file formats are supported by Terraform? (Select TWO)

- A. Node
- B. JSON
- C. Go
- D. YAML
- E. HCL

Answer: ([SHOW ANSWER](#))

Terraform supports both HashiCorp Configuration Language (HCL) and JSON formats for configurations.

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

Valid TA-002-P Dumps shared by PrepPdf.com for Helping Passing TA-002-P Exam!

PrepPdf.com now offer the **newest TA-002-P exam dumps**, the PrepPdf.com TA-002-P exam **questions have been updated** and **answers have been corrected** get the **newest** PrepPdf.com TA-002-P dumps with Test Engine here:

<https://www.preppdf.com/HashiCorp/TA-002-P-prepaway-exam-dumps.html> (94 Q&As Dumps,

40%OFF Special Discount: Exam-Tests)

NEW QUESTION: 17

terraform validate validate validates that your infrastructure matches the Terraform state file.

- A. False
- B. True

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 18

You want to get involved in the development of Terraform. As this is an open source project, you would like to contribute a fix for an open issue of Terraform. What programming language will need to use to write the fix?

- A. It depends on which command issue related to.
- B. Python
- C. Go
- D. Java

Answer: C ([LEAVE A REPLY](#))

Explanation

Basic programming knowledge. Terraform and Terraform Plugins are written in the Go programming language, but even if you've never written a line of Go before, you're still welcome to take a dive into the code and submit patches. The community is happy to assist with code reviews and offer guidance specific to Go.

NEW QUESTION: 19

Terraform Cloud is more powerful when you integrate it with your version control system (VCS) provider. Select all the supported VCS providers from the answers below. (select four)

- A. GitHub
- B. CVS Version Control
- C. Azure DevOps Server
- D. Bitbucket Cloud
- E. GitHub Enterprise

Answer: A,C,D,E ([LEAVE A REPLY](#))

Explanation

Terraform Cloud supports the following VCS providers:

- <https://www.terraform.io/docs/cloud/vcs/github.html>
- <https://www.terraform.io/docs/cloud/vcs/github.html>
- <https://www.terraform.io/docs/cloud/vcs/github-enterprise.html>
- <https://www.terraform.io/docs/cloud/vcs/gitlab-com.html>
- <https://www.terraform.io/docs/cloud/vcs/gitlab-eece.html>
- <https://www.terraform.io/docs/cloud/vcs/bitbucket-cloud.html>
- <https://www.terraform.io/docs/cloud/vcs/bitbucket-server.html>
- <https://www.terraform.io/docs/cloud/vcs/azure-devops-server.html>
- <https://www.terraform.io/docs/cloud/vcs/azure-devops-services.html>

<https://www.terraform.io/docs/cloud/vcs/index.html#supported-vcs-providers>

NEW QUESTION: 20

Jim has created several AWS resources from a single terraform configuration file. Someone from his team has manually modified one of the EC2 instance.

Now to discard the manual change, Jim wants to destroy and recreate the EC2 instance. What is the best way to do it?

- A. terraform recreate
- B. terraform taint
- C. terraform destroy
- D. terraform refresh

Answer: B ([LEAVE A REPLY](#))

The terraform taint command manually marks a Terraform-managed resource as tainted, forcing it to be destroyed and recreated on the next apply.

This command will not modify infrastructure, but does modify the state file in order to mark a resource as tainted. Once a resource is marked as tainted, the next plan will show that the resource will be destroyed and recreated and the next apply will implement this change.

Forcing the recreation of a resource is useful when you want a certain side effect of recreation that is not visible in the attributes of a resource. For example: re-running provisioners will cause

the node to be different or rebooting the machine from a base image will cause new startup scripts to run.

Note that tainting a resource for recreation may affect resources that depend on the newly tainted resource. For example, a DNS resource that uses the IP address of a server may need to be modified to reflect the potentially new IP address of a tainted server. The plan command will show this if this is the case.

This example will taint a single resource:

```
$ terraform taint aws_security_group.allow_all
```

The resource `aws_security_group.allow_all` in the module root has been marked as tainted.

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

NEW QUESTION: 21

Select all features which are exclusive to Terraform Enterprise. (Select Three)

- A. Sentinel
- B. Cost Estimation
- C. Audit Logs
- D. Clustering
- E. SAML/SSO

Answer: C,D,E ([LEAVE A REPLY](#))

Explanation

Sentinel and Cost Estimation are also available in Terraform Cloud

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

NEW QUESTION: 22

The `terraform.tfstate` file always matches your currently built infrastructure.

- A. True
- B. False

Answer: B ([LEAVE A REPLY](#))

Reference: <https://www.terraform.io/docs/language/state/index.html>

NEW QUESTION: 23

Terraform installs its providers during which phase?

- A. Plan
- B. Refresh
- C. All of the above
- D. Init

Answer: D ([LEAVE A REPLY](#))

NEW QUESTION: 24

Why might a user opt to include the following snippet in their configuration file?

- A. Terraform 0.12 introduced substantial changes to the syntax used to write Terraform configuration
- B. versions before Terraform 0.12 were not approved by HashiCorp to be used in production
- C. The user wants to ensure that the application being deployed is a minimum version of 0.12
- D. this ensures that all Terraform providers are above a certain version to match the application being deployed

Answer: A ([LEAVE A REPLY](#))

NEW QUESTION: 25

Consider the following Terraform 0.12 configuration snippet:

```
1. variable "vpc_cidrs" {  
2. type = map  
3. default = {  
4. us-east-1 = "10.0.0.0/16"  
5. us-east-2 = "10.1.0.0/16"  
6. us-west-1 = "10.2.0.0/16"  
7. us-west-2 = "10.3.0.0/16"  
8. }  
9. }  
10.  
11. resource "aws_vpc" "shared" {  
12. cidr_block = _____  
13. }
```

How would you define the cidr_block for us-east-1 in the aws_vpc resource using a variable?

- A. var.vpc_cidrs.0
- B. var.vpc_cidrs[0]
- C. vpc_cidrs["us-east-1"]
- D. var.vpc_cidrs["us-east-1"]

Answer: D ([LEAVE A REPLY](#))

NEW QUESTION: 26

Which of the following is considered a Terraform plugin?

- A. Terraform language
- B. Terraform tooling
- C. Terraform logic
- D. Terraform provider

Answer: (SHOW ANSWER)

Terraform is built on a plugin-based architecture. All providers and provisioners that are used in Terraform configurations are plugins, even the core types such as AWS and Heroku. Users of Terraform are able to write new plugins in order to support new functionality in Terraform.

<https://www.terraform.io/docs/plugins/basics.html>

NEW QUESTION: 27

Where does the Terraform local backend store its state?

- A. In the /tmp directory
- B. In the terraform.tfvars file
- C. In the terraform.tfstate file
- D. In the user's .terraformrc file

Answer: ([SHOW ANSWER](#))

The local backend stores state on the local filesystem, locks that state using system APIs, and performs operations locally.

NEW QUESTION: 28

How would you be able to reference an attribute from the vsphere_datacenter data source for use with the argument within the vsphere_folder resource in the following configuration?

- A. data.vsphere_datacenter.dc.id
- B. data.vsphere_datacenter.dc
- C. data.dc.id
- D. vsphere_datacenter.dc.id

Answer: A ([LEAVE A REPLY](#))

NEW QUESTION: 29

Which of the below options is a valid interpolation syntax for retrieving a data source?

- A. \${google_storage_bucket.backend}
- B. \${azurerm_resource_group.test.data}
- C. \${aws_instance.web.id.data}
- D. \${data.google_dns_keys.foo_dns_keys.key_signing_keys[0].ds_record}

Answer: D ([LEAVE A REPLY](#))

Data source attributes are interpolated with the general syntax data.TYPE.NAME.ATTRIBUTE.

The interpolation for a resource is the same but without the data. prefix

(TYPE.NAME.ATTRIBUTE).

<https://www.terraform.io/docs/configuration-0-11/interpolation.html#attributes-of-a-data-source>

NEW QUESTION: 30

Refer below code where pessimistic constraint operator has been used to specify a version of a provider.

```
terraform { required_providers { aws = "~> 1.1.0" }}
```

Which of the following options are valid provider versions that satisfy the above constraint. (select two)

- A. 1.1.1
- B. 1.2.9
- C. 1.1.8

D. 1.2.0

Answer: A,C ([LEAVE A REPLY](#))

Pessimistic constraint operator, constraining both the oldest and newest version allowed. For example, `~> 0.9` is equivalent to `>= 0.9, < 1.0`, and `~> 0.8.4`, is equivalent to `>= 0.8.4, < 0.9`

NEW QUESTION: 31

Which of the following terraform subcommands could be used to remove the lock on the state for the current configuration?

- A. Unlock
- B. force-unlock
- C. Removing the lock on a state file is not possible
- D. state-unlock

Answer: C ([LEAVE A REPLY](#))

Explanation

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

Valid TA-002-P Dumps shared by PrepPdf.com for Helping Passing TA-002-P Exam!

PrepPdf.com now offer the **newest TA-002-P exam dumps**, the PrepPdf.com TA-002-P exam **questions have been updated** and **answers have been corrected** get the **newest** PrepPdf.com TA-002-P dumps with Test Engine here:

<https://www.preppdf.com/HashiCorp/TA-002-P-prepaway-exam-dumps.html> (94 Q&As Dumps,

40%OFF Special Discount: Exam-Tests)

NEW QUESTION: 32

Ric wants to enable detail logging and he wants highest verbosity of logs. Which of the following environment variable settings is correct option for him to select.

- A. Set `TF_LOG = DEBUG`
- B. Set `VAR_TF = TRACE`
- C. Set `TF_LOG = TRACE`
- D. Set `VAR_TF_LOG = TRACE`

Answer: C ([LEAVE A REPLY](#))

<https://www.terraform.io/docs/internals/debugging.html>

NEW QUESTION: 33

Which of the following best describes the default local backend?

- A. The local backend is where Terraform Enterprise stores logs to be processed by an log collector.
- B. The local backend stores state on the local filesystem, locks the state using system APIs, and performs operations locally.

- C.** The local backend is the directory where resources deployed by Terraform have direct access to in order to update their current state.
- D.** The local backend is how Terraform connects to public cloud services, such as AWS, Azure, or GCP.

Answer: B ([LEAVE A REPLY](#))

The local backend stores state on the local filesystem, locks that state using system APIs, and performs operations locally.

```
terraform {  
  backend "local" {  
    path = "relative/path/to/terraform.tfstate"  
  }  
}
```

<https://www.terraform.io/docs/backends/types/local.html>

NEW QUESTION: 34

You want to use different AMI images for different regions and for the purpose you have defined following code block.

1. variable "images"
2. {
3. type = "map"
- 4.
5. default = {
6. us-east-1 = "image-1234"
7. us-west-2 = "image-4567"
8. us-west-1 = "image-4589"
9. }
10. }

What of the following approaches needs to be followed in order to select image-4589?

- A.** var.images[2]
- B.** var.images[3]
- C.** lookup(var.images["us-west-1"]
- D.** var.images["us-west-1"]

Answer: D ([LEAVE A REPLY](#))

NEW QUESTION: 35

What is the provider for this fictitious resource?

- A.** vpc
- B.** main
- C.** aws
- D.** test

Answer: C ([LEAVE A REPLY](#))

Reference: <https://docs.aws.amazon.com/cloudformation-cli/latest/userguide/resource-types.html>

NEW QUESTION: 36

Environment variables can be used to set variables. The environment variables must be in the format

"____"<variablename>. Select the correct prefix string from the following list.

- A. TF_CLI_ARGS
- B. TF_VAR
- C. TF_VAR_
- D. TF_VAR_ENV

Answer: C ([LEAVE A REPLY](#))

Explanation

Environment variables can be used to set variables. The environment variables must be in the format TF_VAR_name and this will be checked last for a value. For example:

```
export TF_VAR_region=us-west-1
```

```
export TF_VAR_ami=ami-049d8641
```

```
export TF_VAR_alist='[1,2,3]'
```

```
export TF_VAR_amap='{ foo = "bar", baz = "qux" }'
```

<https://www.terraform.io/docs/commands/environment-variables.html>

NEW QUESTION: 37

What is the provider for this fictitious resource?

- A. vpc
- B. aws
- C. main
- D. test

Answer: B ([LEAVE A REPLY](#))

NEW QUESTION: 38

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 ([LEAVE A REPLY](#))

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.

NEW QUESTION: 39

Which of the following value will be accepted for var1?

```
variable "var1" {  
  type = string  
}
```

- A. None of the above
- B. Both A and B
- C. "5"
- D. 5

Answer: ([SHOW ANSWER](#))

Terraform automatically converts number and bool values to strings when needed.

NEW QUESTION: 40

You have written a terraform IaC script which was working till yesterday , but is giving some vague error from today , which you are unable to understand . You want more detailed logs that could potentially help you troubleshoot the issue , and understand the root cause. What can you do to enable this setting? Please note , you are using terraform OSS.

- A. Terraform OSS can push all its logs to a syslog endpoint. As such, you have to set up the syslog sink, and enable TF_LOG_PATH env variable to the syslog endpoint and all logs will automatically start streaming.
- B. Detailed logs are not available in terraform OSS, except the crash message. You need to upgrade to terraform enterprise for this point.
- C. Enable the TF_LOG_PATH to the log sink file location, and logging output will automatically be stored there.
- D. Enable TF_LOG to the log level DEBUG, and then set TF_LOG_PATH to the log sink file location. Terraform debug logs will be dumped to the sink path, even in terraform OSS.

Answer: D ([LEAVE A REPLY](#))

Terraform has detailed logs which can be enabled by setting the TF_LOG environment variable to any value. This will cause detailed logs to appear on stderr.

You can set TF_LOG to one of the log levels TRACE, DEBUG, INFO, WARN or ERROR to change the verbosity of the logs. TRACE is the most verbose and it is the default if TF_LOG is set to something other than a log level name.

To persist logged output you can set TF_LOG_PATH in order to force the log to always be appended to a specific file when logging is enabled. Note that even when TF_LOG_PATH is set, TF_LOG must be set in order for any logging to be enabled.

NEW QUESTION: 41

In regards to Terraform state file, select all the statements below which are correct?

- A. When using local state, the state file is stored in plain-text.
- B. The state file is always encrypted at rest.
- C. Storing state remotely can provide better security.
- D. Using the mask feature, you can instruct Terraform to mask sensitive data in the state file.
- E. The Terraform state can contain sensitive data, therefore the state file should be protected from unauthorized access.
- F. Terraform Cloud always encrypts state at rest.

Answer: A,C,E,F ([LEAVE A REPLY](#))

Explanation

Terraform state can contain sensitive data, depending on the resources in use and your definition of

"sensitive." The state contains resource IDs and all resource attributes. For resources such as databases, this may contain initial passwords.

When using local state, state is stored in plain-text JSON files.

When using remote state, state is only ever held in memory when used by Terraform. It may be encrypted at rest, but this depends on the specific remote state backend.

Storing Terraform state remotely can provide better security. As of Terraform 0.9, Terraform does not persist state to the local disk when remote state is in use, and some backends can be configured to encrypt the state data at rest.

Recommendations

If you manage any sensitive data with Terraform (like database passwords, user passwords, or private keys), treat the state itself as sensitive data.

Storing state remotely can provide better security. As of Terraform 0.9, Terraform does not persist state to the local disk when remote state is in use, and some backends can be configured to encrypt the state data at rest.

For example:

* Terraform Cloud always encrypts state at rest and protects it with TLS in transit. Terraform Cloud also knows the identity of the user requesting state and maintains a history of state changes. This can be used to control access and track activity. Terraform Enterprise also supports detailed audit logging.

* The S3 backend supports encryption at rest when the encrypt option is enabled. IAM policies and logging can be used to identify any invalid access. Requests for the state go over a TLS connection.

NEW QUESTION: 42

Using multi-cloud and provider-agnostic tools provides which of the following benefits?

- A. Operations teams only need to learn and manage a single tool to manage infrastructure, regardless of where the infrastructure is deployed.
- B. Increased risk due to all infrastructure relying on a single tool for management.
- C. Can be used across major cloud providers and VM hypervisors.
- D. Slower provisioning speed allows the operations team to catch mistakes before they are applied.

Answer: A,C ([LEAVE A REPLY](#))

Using a tool like Terraform can be advantageous for organizations deploying workloads across multiple public and private cloud environments. Operations teams only need to learn a single tool, single language, and can use the same tooling to enable a DevOps-like experience and workflows.

NEW QUESTION: 43

After executing a terraform apply, you notice that a resource has a tilde (~) next to it. What does this infer?

- A. The resource will be updated in place.
- B. The resource will be created.
- C. Terraform can't determine how to proceed due to a problem with the state file.
- D. The resource will be destroyed and recreated.

Answer: A ([LEAVE A REPLY](#))

Explanation

The prefix -/+ means that Terraform will destroy and recreate the resource, rather than updating it in-place.

The prefix ~ means that some attributes and resources can be updated in-place.

\$ terraform apply

aws_instance.example: Refreshing state... [id=i-0bbf06244e44211d1]

An execution plan has been generated and is shown below.

Resource actions are indicated with the following symbols:

-/+ destroy and then create replacement

Terraform will perform the following actions:

aws_instance.example must be replaced

-/+ resource "aws_instance" "example" {

~ ami = "ami-2757f631" -> "ami-b374d5a5" # forces replacement

~ arn = "arn:aws:ec2:us-east-1:130490850807:instance/i-0bbf06244e44211d1" -> (known after apply)

~ associate_public_ip_address = true -> (known after apply)
~ availability_zone = "us-east-1c" -> (known after apply)
~ cpu_core_count = 1 -> (known after apply)
~ cpu_threads_per_core = 1 -> (known after apply)
- disable_api_termination = false -> null
- ebs_optimized = false -> null
get_password_data = false
+ host_id = (known after apply)
~ id = "i-0bbf06244e44211d1" -> (known after apply)
~ instance_state = "running" -> (known after apply)
instance_type = "t2.micro"
~ ipv6_address_count = 0 -> (known after apply)
~ ipv6_addresses = [] -> (known after apply)
+ key_name = (known after apply)
- monitoring = false -> null
+ network_interface_id = (known after apply)
+ password_data = (known after apply)
+ placement_group = (known after apply)
~ primary_network_interface_id = "eni-0f1ce5bdae258b015" -> (known after apply)
~ private_dns = "ip-172-31-61-141.ec2.internal" -> (known after apply)
~ private_ip = "172.31.61.141" -> (known after apply)
~ public_dns = "ec2-54-166-19-244.compute-1.amazonaws.com" -> (known after apply)
~ public_ip = "54.166.19.244" -> (known after apply)
~ security_groups = [
- "default",
] -> (known after apply)
source_dest_check = true
~ subnet_id = "subnet-1facdf35" -> (known after apply)
~ tenancy = "default" -> (known after apply)
~ volume_tags = {} -> (known after apply)
~ vpc_security_group_ids = [
- "sg-5255f429",
] -> (known after apply)
- credit_specification {
- cpu_credits = "standard" -> null
}
+ ebs_block_device {
+ delete_on_termination = (known after apply)
+ device_name = (known after apply)
+ encrypted = (known after apply)
+ iops = (known after apply)

```

+ snapshot_id = (known after apply)
+ volume_id = (known after apply)
+ volume_size = (known after apply)
+ volume_type = (known after apply)
}
+ ephemeral_block_device {
+ device_name = (known after apply)
+ no_device = (known after apply)
+ virtual_name = (known after apply)
}
+ network_interface {
+ delete_on_termination = (known after apply)
+ device_index = (known after apply)
+ network_interface_id = (known after apply)
}
~ root_block_device {
~ delete_on_termination = true -> (known after apply)
~ iops = 100 -> (known after apply)
~ volume_id = "vol-0079e485d9e28a8e5" -> (known after apply)
~ volume_size = 8 -> (known after apply)
~ volume_type = "gp2" -> (known after apply)
}
}

```

Plan: 1 to add, 0 to change, 1 to destroy.

NEW QUESTION: 44

A user creates three workspaces from the command line - prod, dev, and test. Which of the following commands will the user run to switch to the dev workspace?

- A. terraform workspace dev
- B. terraform workspace select dev
- C. terraform workspace -switch dev
- D. terraform workspace switch dev

Answer: B ([LEAVE A REPLY](#))

Explanation

The terraform workspace select command is used to choose a different workspace to use for further operations. <https://www.terraform.io/docs/commands/workspace/select.html>

NEW QUESTION: 45

One remote backend configuration always maps to a single remote workspace.

- A. True
- B. False

Answer: A ([LEAVE A REPLY](#))

NEW QUESTION: 46

Which task does terraform init not perform?

- A. Sources all providers present in the configuration and ensures they are downloaded and available locally
- B. Connects to the backend
- C. Sources any modules and copies the configuration locally
- D. Validates all required variables are present

Answer: ([SHOW ANSWER](#)**)**

Reference: <https://www.terraform.io/docs/cli/commands/init.html>

Valid TA-002-P Dumps shared by PrepPdf.com for Helping Passing TA-002-P Exam! PrepPdf.com now offer the **newest TA-002-P exam dumps**, the PrepPdf.com TA-002-P exam **questions have been updated** and **answers have been corrected** get the **newest** PrepPdf.com TA-002-P dumps with Test Engine here:
<https://www.preppdf.com/HashiCorp/TA-002-P-prepaway-exam-dumps.html> (94 Q&As Dumps, **40%OFF** Special Discount: **Exam-Tests**)

NEW QUESTION: 47

True or False? When using the Terraform provider for Vault, the tight integration between these HashiCorp tools provides the ability to mask secrets in the terraform plan and state files.

- A. True

Explanation

Currently, Terraform has no mechanism to redact or protect secrets that are returned via data sources, so secrets read via this provider will be persisted into the Terraform state, into any plan files, and in some cases in the console output produced while planning and applying. These artifacts must, therefore, all be protected accordingly.

- B. False

Answer: B ([LEAVE A REPLY](#))

NEW QUESTION: 48

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?

- A. 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.

- B.** Delete the existing resources and recreate them using new a Terraform configuration so Terraform can manage them moving forward.
- C.** Resources that are manually deployed in the AWS console cannot be imported by Terraform.
- D.** Using terraform import, import the existing infrastructure into your Terraform state.

Answer: D ([LEAVE A REPLY](#))

Explanation

Terraform is able to import existing infrastructure. This allows us take resources we've created by some other means (i.e. via console) and bring it under Terraform management.

This is a great way to slowly transition infrastructure to Terraform.

The terraform import command is used to import existing infrastructure.

To import a resource, first write a resource block for it in our configuration, establishing the name by which it will be known to Terraform.

Example:

```
resource "aws_instance" "import_example" {  
  # ...instance configuration...  
}
```

Now terraform import can be run to attach an existing instance to this resource configuration.

```
$ terraform import aws_instance.import_example i-03efafa258104165f
```

```
aws_instance.import_example: Importing from ID "i-03efafa258104165f"...
```

```
aws_instance.import_example: Import complete!
```

```
Imported aws_instance (ID: i-03efafa258104165f)
```

```
aws_instance.import_example: Refreshing state... (ID: i-03efafa258104165f) Import successful!
```

The resources that were imported are shown above. These resources are now in your Terraform state and will henceforth be managed by Terraform.

This command locates the AWS instance with ID i-03efafa258104165f (which has been created outside Terraform) and attaches its existing settings, as described by the EC2 API, to the name aws_instance.import_example in the Terraform state.

NEW QUESTION: 49

Your developers are facing a lot of problem while writing complex expressions involving difficult interpolations . They have to run the terraform plan every time and check whether there are errors , and also check terraform apply to print the value as a temporary output for debugging purposes. What should be done to avoid this?

- A.** Use terraform console command to have an interactive UI with full access to the underlying terraform state to run your interpolations , and debug at real-time.
- B.** Add a breakpoint in your code, using the watch keyword , and output the value to console for temporary debugging.
- C.** Use terraform zipmap function , it will be able to easily do the interpolations without complex code.
- D.** Use terraform console command to have an interactive UI , but you can only use it with local state , and it does not work with remote state.

Answer: (SHOW ANSWER)

The terraform console command provides an interactive console for evaluating expressions. This is useful for testing interpolations before using them in configurations, and for interacting with any values currently saved in state.

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

NEW QUESTION: 50

Terraform validate reports syntax check errors from which of the following scenarios?

- A. Code contains tabs indentation instead of spaces
- B. None of the above
- C. There is missing value for a variable
- D. The state files does not match the current infrastructure

Answer: (SHOW ANSWER)

NEW QUESTION: 51

Environment variables can be used to set variables. The environment variables must be in the format "`____`"_<variablename>. Select the correct prefix string from the following list.

- A. TF_CLI_ARGS
- B. TF_VAR
- C. TF_VAR_
- D. TF_VAR_ENV

Answer: C (LEAVE A REPLY)

Environment variables can be used to set variables. The environment variables must be in the format `TF_VAR_name` and this will be checked last for a value. For example:

```
export TF_VAR_region=us-west-1
export TF_VAR_ami=ami-049d8641
export TF_VAR_alist='[1,2,3]'
export TF_VAR_amap='{ foo = "bar", baz = "qux" }'
```

<https://www.terraform.io/docs/commands/environment-variables.html>

NEW QUESTION: 52

Which of the following terraform subcommands could be used to remove the lock on the state for the current configuration?

- A. Removing the lock on a state file is not possible
- B. force-unlock
- C. Unlock
- D. state-unlock

Explanation

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

Answer: A (LEAVE A REPLY)

NEW QUESTION: 53

If a module uses a local variable, you can expose that value with a terraform output.

- A. True
- B. False

Answer: A ([LEAVE A REPLY](#))

Explanation

Output values are like function return values.

Reference: <https://www.terraform.io/docs/language/values/locals.html>

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

NEW QUESTION: 54

When using providers that require the retrieval of data, such as the HashiCorp Vault provider, in what phase does Terraform actually retrieve the data required?

- A. terraform plan
- B. terraform delete
- C. terraform init
- D. terraform apply

Answer: C ([LEAVE A REPLY](#))

NEW QUESTION: 55

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

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

Answer: C ([LEAVE A REPLY](#))

NEW QUESTION: 56

Setting the TF_LOG environment variable to DEBUG causes debug messages to be logged into syslog.

- A. True
- B. False

Answer: ([SHOW ANSWER](#))

Reference: <https://www.terraform.io/docs/internals/debugging.html>

NEW QUESTION: 57

When writing Terraform code, HashiCorp recommends that you use how many spaces between each nesting level?

- A. 0
- B. 1
- C. 2

D. 4

Answer: C ([LEAVE A REPLY](#))

The Terraform parser allows you some flexibility in how you lay out the elements in your configuration files, but the Terraform language also has some idiomatic style conventions which we recommend users always follow for consistency between files and modules written by different teams. Automatic source code formatting tools may apply these conventions automatically.

Indent two spaces for each nesting level.

When multiple arguments with single-line values appear on consecutive lines at the same nesting level, align their equals signs:

```
ami = "abc123"
```

```
instance_type = "t2.micro"
```

When both arguments and blocks appear together inside a block body, place all of the arguments together at the top and then place nested blocks below them. Use one blank line to separate the arguments from the blocks.

Use empty lines to separate logical groups of arguments within a block.

For blocks that contain both arguments and "meta-arguments" (as defined by the Terraform language semantics), list meta-arguments first and separate them from other arguments with one blank line. Place meta-argument blocks last and separate them from other blocks with one blank line.

```
resource "aws_instance" "example" {
```

```
  count = 2 # meta-argument first
```

```
  ami = "abc123"
```

```
  instance_type = "t2.micro"
```

```
  network_interface {
```

```
    # ...
```

```
  }
```

```
  lifecycle { # meta-argument block last
```

```
    create_before_destroy = true
```

```
  }
```

```
}
```

Top-level blocks should always be separated from one another by one blank line. Nested blocks should also be separated by blank lines, except when grouping together related blocks of the same type (like multiple provisioner blocks in a resource).

Avoid separating multiple blocks of the same type with other blocks of a different type, unless the block types are defined by semantics to form a family. (For example: `root_block_device`, `ebs_block_device` and `ephemeral_block_device` on `aws_instance` form a family of block types describing AWS block devices, and can therefore be grouped together and mixed.)

NEW QUESTION: 58

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 ([LEAVE A REPLY](#))

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>

NEW QUESTION: 59

Examine the following Terraform configuration, which uses the data source for an AWS AMI. What value should you enter for the ami argument in the AWS instance resource?

A. aws_ami.ubuntu

B. data.aws_ami.ubuntu

C. data.aws_ami.ubuntu.id

D. aws_ami.ubuntu.id

Answer: C ([LEAVE A REPLY](#))

```
resource "aws_instance" "web" {  
  ami = data.aws_ami.ubuntu.id
```

Reference: <https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/instance>

NEW QUESTION: 60

What is the command you can use to set an environment variable named "var1" of type String?

A. export TF_VAR_VAR1

B. set TF_VAR_var1

C. variable "var1" { type = "string" }

D. export TF_VAR_var1

Answer: D ([LEAVE A REPLY](#))

The environment variable must be in the format TF_VAR_name, so for the

https://www.terraform.io/docs/commands/environment-variables.html#tf_var_name

NEW QUESTION: 61

Complete the following sentence:

The terraform state command can be used to _____

A. modify state

B. view state

C. refresh state

D. There is no such command

Answer: (SHOW ANSWER)

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

Valid TA-002-P Dumps shared by PrepPdf.com for Helping Passing TA-002-P Exam!

PrepPdf.com now offer the **newest TA-002-P exam dumps**, the PrepPdf.com TA-002-P exam **questions have been updated** and **answers have been corrected** get the **newest** PrepPdf.com TA-002-P dumps with Test Engine here:

<https://www.preppdf.com/HashiCorp/TA-002-P-prepaway-exam-dumps.html> (94 Q&As Dumps, **40%OFF Special Discount: Exam-Tests**)

NEW QUESTION: 62

When using constraint expressions to signify a version of a provider, which of the following are valid provider versions that satisfy the expression found in the following code snippet: (select two)

1. terraform
 2. {
 3. required_providers
 4. {
 5. aws = "~> 1.2.0"
 6. }
 7. }
- A.** 1.3.1
B. 1.2.3
C. 1.2.9
D. 1.3.0

Answer: ([SHOW ANSWER](#))

Explanation

As your Terraform usage becomes more advanced, there are some cases where you may need to modify the Terraform state. Rather than modify the state directly, the terraform state commands can be used in many cases instead. This command is a nested subcommand, meaning that it has further subcommands.

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

NEW QUESTION: 63

The terraform init command is always safe to run multiple times, to bring the working directory up to date with changes in the configuration. Though subsequent runs may give errors, this command will never delete your existing configuration or state.

- A.** False
B. True

Answer: B ([LEAVE A REPLY](#))

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

NEW QUESTION: 64

Which Terraform command will check and report errors within modules, attribute names, and value types to make sure they are syntactically valid and internally consistent?

- A. terraform validate
- B. terraform format
- C. terraform fmt
- D. terraform show

Answer: A ([LEAVE A REPLY](#))

Explanation

The terraform validate command validates the configuration files in a directory, referring only to the configuration and not accessing any remote services such as remote state, provider APIs, etc. Validate runs checks that verify whether a configuration is syntactically valid and internally consistent, regardless of any provided variables or existing state. It is thus primarily useful for general verification of reusable modules, including the correctness of attribute names and value types.

It is safe to run this command automatically, for example as a post-save check in a text editor or as a test step for a re-usable module in a CI system.

NEW QUESTION: 65

How does Terraform handle working with so many providers?

- A. Terraform ships with all of the plugins embedded in the Terraform binary.
- B. Terraform uses a plugin architecture for providers and only installs the provider plugins required by your configuration in the configuration's working directory.
- C. Terraform uses a plugin architecture for providers and only installs the provider plugins required by your configuration in a shared, system-wide plugins directory.
- D. Terraform allows you to select the providers you want to support during the Terraform installation process.

Answer: (SHOW ANSWER)

Explanation

Terraform is built on a plugin-based architecture. All providers and provisioners that are used in Terraform configurations are plugins, even the core types such as AWS and Heroku. Users of Terraform are able to write new plugins in order to support new functionality in Terraform.

NEW QUESTION: 66

Which of the following are string functions? Select three

- A. tostring
- B. tonumber
- C. Chomp
- D. format
- E. join

Answer: C,D,E ([LEAVE A REPLY](#))

tonumber and toString are Type Conversion function

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

NEW QUESTION: 67

Which of the following clouds does not have a provider maintained HashiCorp?

- A. IBM Cloud
- B. DigitalOcean
- C. OpenStack
- D. AWS

Answer: A ([LEAVE A REPLY](#))

Explanation

IBM Cloud does not have a provider maintained by HashiCorp, although IBM Cloud does maintain their own Terraform provider.

<https://www.terraform.io/docs/providers/index.html>

NEW QUESTION: 68

Ric wants to enable detail logging and he wants highest verbosity of logs. Which of the following environment variable settings is correct option for him to select.

- A. Set TF_LOG = DEBUG
- B. Set VAR_TF = TRACE
- C. Set TF_LOG = TRACE
- D. Set VAR_TF_LOG = TRACE

Answer: C ([LEAVE A REPLY](#))

Explanation

<https://www.terraform.io/docs/internals/debugging.html>

NEW QUESTION: 69

FILL BLANK

You need to specify a dependency manually.

What resource meta-parameter can you use to make sure Terraform respects the dependency?

Type your answer in the field provided. The text field is not case-sensitive and all variations of the correct answer are accepted.

Answer:

the local_file data source

Reference: <https://www.terraform.io/docs/language/functions/file.html>

NEW QUESTION: 70

True or False? Each Terraform workspace uses its own state file to manage the infrastructure associated with that particular workspace.

- A. True

Explanation

The persistent data stored in the backend belongs to a workspace. Initially, the backend has only one workspace, called "default", and thus there is only one Terraform state associated with that configuration.

B. False

Answer: A ([LEAVE A REPLY](#))

NEW QUESTION: 71

How is terraform import run?

A. As a part of terraform init

B. By an explicit call

C. As a part of terraform plan

D. As a part of terraform refresh

E. All of the above

Answer: C ([LEAVE A REPLY](#))

NEW QUESTION: 72

Which of the below features of Terraform can be used for managing small differences between different environments which can act more like completely separate working directories.

A. Repositories

B. Workspaces

C. Environment Variables

D. Backends

Answer: B ([LEAVE A REPLY](#))

workspaces allow conveniently switching between multiple instances of a single configuration within its single backend. They are convenient in a number of situations, but cannot solve all problems.

A common use for multiple workspaces is 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. For example, a developer working on a complex set of infrastructure changes might create a new temporary workspace in order to freely experiment with changes without affecting the default workspace.

Non-default workspaces are often related to feature branches in version control. The default workspace might correspond to the "master" or "trunk" branch, which describes the intended state of production infrastructure. When a feature branch is created to develop a change, the developer of that feature might create a corresponding workspace and deploy into it a temporary "copy" of the main infrastructure so that changes can be tested without affecting the production infrastructure. Once the change is merged and deployed to the default workspace, the test infrastructure can be destroyed and the temporary workspace deleted.

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

<https://www.terraform.io/docs/state/workspaces.html#when-to-use-multiple-workspaces>

NEW QUESTION: 73

As a developer, you want to ensure your plugins are up to date with the latest versions. Which Terraform command should you use?

- A. terraform init -upgrade
- B. terraform refresh -upgrade
- C. terraform providers- upgrade
- D. terraform apply -upgrade

Answer: A ([LEAVE A REPLY](#))

NEW QUESTION: 74

Which of the below configuration file formats are supported by Terraform? (Select TWO)

- A. Node
- B. JSON
- C. Go
- D. YAML
- E. HCL

Answer: B,E ([LEAVE A REPLY](#))

Explanation

Terraform supports both HashiCorp Configuration Language (HCL) and JSON formats for configurations.

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

NEW QUESTION: 75

Terraform init can indeed be run only a few times, because, every time terraform init will initialize the project

, and download all plugins from the internet repository , regardless of whether they were present or not , and this increases the waiting time

- A. True
- B. False

Answer: B ([LEAVE A REPLY](#))

Explanation

Re-running init with modules already installed will install the sources for any modules that were added to configuration since the last init, but will not change any already-installed modules. Use -upgrade to override this behavior, updating all modules to the latest available source code.

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

NEW QUESTION: 76

Your manager has instructed you to start using terraform for your day-to-day operations, but your security team is concerned about the terraform state files. They have heard it contains

confidential information, and are worried that it will not be securely protected. What should be your response to the security team in this regard?

A. Inform the security team that using terraform state is optional . There are ways to avoid it , and you will do the same.

B. Ensure that the state is managed in a remote backend , preferably an enterprise grade state management system like Terraform Cloud.

C. Mask the confidential entries in the terraform state file , using Vault Enterprise, another Hashicorp product , while keeping it locally.

D. Keep the state file locally on each developer machine , and ensure that there is a local protection software like KeyPass protecting it.

Answer: B (LEAVE A REPLY)

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

State is very important topic for exam. Please read all of the below subtopics Purpose Import Existing Resources Locking Workspaces Remote State Sensitive Data

Valid TA-002-P Dumps shared by PrepPdf.com for Helping Passing TA-002-P Exam!

PrepPdf.com now offer the **newest TA-002-P exam dumps**, the PrepPdf.com TA-002-P exam **questions have been updated** and **answers have been corrected** get the **newest** PrepPdf.com TA-002-P dumps with Test Engine here:

<https://www.preppdf.com/HashiCorp/TA-002-P-prepaway-exam-dumps.html> (94 Q&As Dumps, **40%OFF Special Discount: Exam-Tests**)

NEW QUESTION: 77

178

correct?

A. When using local state, the state file is stored in plain-text.

B. The state file is always encrypted at rest.

C. Storing state remotely can provide better security.

D. Using the mask feature, you can instruct Terraform to mask sensitive data in the state file.

E. The Terraform state can contain sensitive data, therefore the state file should be protected from unauthorized access.

F. Terraform Cloud always encrypts state at rest.

Answer: A,C,E,F (LEAVE A REPLY)

Terraform state can contain sensitive data, depending on the resources in use and your definition of "sensitive." The state contains resource IDs and all resource attributes. For resources such as databases, this may contain initial passwords.

When using local state, state is stored in plain-text JSON files.

When using remote state, state is only ever held in memory when used by Terraform. It may be encrypted at rest, but this depends on the specific remote state backend.

Storing Terraform state remotely can provide better security. As of Terraform 0.9, Terraform does not persist state to the local disk when remote state is in use, and some backends can be configured to encrypt the state data at rest.

Recommendations

If you manage any sensitive data with Terraform (like database passwords, user passwords, or private keys), treat the state itself as sensitive data.

Storing state remotely can provide better security. As of Terraform 0.9, Terraform does not persist state to the local disk when remote state is in use, and some backends can be configured to encrypt the state data at rest.

For example:

* Terraform Cloud always encrypts state at rest and protects it with TLS in transit. Terraform Cloud also knows the identity of the user requesting state and maintains a history of state changes. This can be used to control access and track activity. Terraform Enterprise also supports detailed audit logging.

* The S3 backend supports encryption at rest when the encrypt option is enabled. IAM policies and logging can be used to identify any invalid access. Requests for the state go over a TLS connection.

NEW QUESTION: 78

You have recently started a new job at a retailer as an engineer. As part of this new role, you have been tasked with evaluating multiple outages that occurred during peak shopping time during the holiday season. Your investigation found that the team is manually deploying new compute instances and configuring each compute instance manually. This has led to inconsistent configuration between each compute instance.

How would you solve this using infrastructure as code?

- A. Replace the compute instance type with a larger version to reduce the number of required deployments
- B. Implement a provisioning pipeline that deploys infrastructure configurations committed to your version control system following code reviews
- C. Implement a ticketing workflow that makes engineers submit a ticket before manually provisioning and configuring a resource
- D. Implement a checklist that engineers can follow when configuring compute instances

Answer: C ([LEAVE A REPLY](#))

NEW QUESTION: 79

In Terraform Enterprise, a workspace can be mapped to how many VCS repos?

- A. 5
- B. 2
- C. 3
- D. 1

Answer: ([SHOW ANSWER](#)**)**

A workspace can only be configured to a single VCS repo, however, multiple workspaces can use the same repo.

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

NEW QUESTION: 80

Dawn has created the below child module. Without changing the module, can she override the instance_type from t2.micro to t2.large from her code while calling this module?

1. resource "aws_instance" "myec2"
2. {
3. ami = "ami-082b5a644766e0e6f"
4. instance_type = "t2.micro"
5. }

A. YES

B. No

Answer: ([SHOW ANSWER](#))

As the instance_type is hard-coded in source module, you will not be able to change its value from destination module. Instead of hard-coding you should use variable with default values.

NEW QUESTION: 81

Select all Operating Systems that Terraform is available for. (select five)

- A. Windows
- B. Unix
- C. Solaris
- D. Linux
- E. FreeBSD

Explanation

Terraform is available for macOS, FreeBSD, OpenBSD, Linux, Solaris, Windows

<https://www.terraform.io/downloads.html>

F. macOS

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 82

What does terraform refresh modify?

- A. Your cloud infrastructure
- B. Your Terraform configuration
- C. Your Terraform plan
- D. Your state file

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 83

State is a requirement for Terraform to function

A. True

B. False

Answer: A ([LEAVE A REPLY](#))

Explanation

State is a necessary requirement for Terraform to function. It is often asked if it is possible for Terraform to work without state, or for Terraform to not use state and just inspect cloud resources on every run.

Purpose of Terraform State

State is a necessary requirement for Terraform to function. It is often asked if it is possible for Terraform to work without state, or for Terraform to not use state and just inspect cloud resources on every run. This page will help explain why Terraform state is required.

As you'll see from the reasons below, state is required. And in the scenarios where Terraform may be able to get away without state, doing so would require shifting massive amounts of complexity from one place (state) to another place (the replacement concept).

1. Mapping to the Real World

Terraform requires some sort of database to map Terraform config to the real world. When you have a resource resource "aws_instance" "foo" in your configuration, Terraform uses this map to know that instance i- abcd1234 is represented by that resource.

For some providers like AWS, Terraform could theoretically use something like AWS tags. Early prototypes of Terraform actually had no state files and used this method. However, we quickly ran into problems. The first major issue was a simple one: not all resources support tags, and not all cloud providers support tags.

Therefore, for mapping configuration to resources in the real world, Terraform uses its own state structure.

2. Metadata

Alongside the mappings between resources and remote objects, Terraform must also track metadata such as resource dependencies.

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.

One way to avoid this would be for Terraform to know a required ordering between resource types. For example, Terraform could know that servers must be deleted before the subnets they are a part of. The complexity for this approach quickly explodes, however: in addition to Terraform having to understand the ordering semantics of every resource for every cloud, Terraform must also understand the ordering across providers.

Terraform also stores other metadata for similar reasons, such as a pointer to the provider configuration that was most recently used with the resource in situations where multiple aliased providers are present.

3. Performance

In addition to basic mapping, Terraform stores a cache of the attribute values for all resources in the state. This is the most optional feature of Terraform state and is done only as a performance improvement.

When running a terraform plan, Terraform must know the current state of resources in order to effectively determine the changes that it needs to make to reach your desired configuration.

For small infrastructures, Terraform can query your providers and sync the latest attributes from all your resources. This is the default behavior of Terraform: for every plan and apply, Terraform will sync all resources in your state.

For larger infrastructures, querying every resource is too slow. Many cloud providers do not provide APIs to query multiple resources at once, and the round trip time for each resource is hundreds of milliseconds. On top of this, cloud providers almost always have API rate limiting so Terraform can only request a certain number of resources in a period of time. Larger users of Terraform make heavy use of the `-refresh=false` flag as well as the `-target` flag in order to work around this. In these scenarios, the cached state is treated as the record of truth.

4. Syncing

In the default configuration, Terraform stores the state in a file in the current working directory where Terraform was run. This is okay for getting started, but 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.

Remote state is the recommended solution to this problem. With a fully-featured state backend, Terraform can use remote locking as a measure to avoid two or more different users accidentally running Terraform at the same time, and thus ensure that each Terraform run begins with the most recent updated state.

NEW QUESTION: 84

Your team has started using terraform OSS in a big way , and now wants to deploy multi region deployments (DR) in aws using the same terraform files . You want to deploy the same infra (VPC,EC2 ...) in both us-east-1 ,and us-west-2 using the same script , and then peer the VPCs across both the regions to enable DR traffic. But , when you run your script , all resources are getting created in only the default provider region.

What should you do? Your provider setting is as below -

```
# The default provider configuration provider "aws" { region = "us-east-1" }
```

A. No way to enable this via a single script . Write 2 different scripts with different default providers in the

2 scripts , one for us-east , another for us-west.

B. Create a list of regions , and then use a for-each to iterate over the regions , and create the same resources ,one after the one , over the loop.

C. Use provider alias functionality , and add another provider for us-west region . While creating the resources using the tf script , reference the appropriate provider (using the alias).

D. Manually create the DR region , once the Primary has been created , since you are using terraform OSS , and multi region deployment is only available in Terraform Enterprise.

Answer: (SHOW ANSWER)

Explanation

You can optionally define multiple configurations for the same provider, and select which one to use on a per-resource or per-module basis. The primary reason for this is to support multiple regions for a cloud platform; other examples include targeting multiple Docker hosts, multiple Consul hosts, etc.

To include multiple configurations for a given provider, include multiple provider blocks with the same provider name, but set the alias meta-argument to an alias name to use for each additional configuration. For example:

The default provider configuration

```
provider "aws" {  
  region = "us-east-1"  
}
```

Additional provider configuration for west coast region

```
provider "aws" {  
  alias = "west"  
  region = "us-west-2"  
}
```

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

NEW QUESTION: 85

When configuring a remote backend in Terraform, it might be a good idea to purposely omit some of the required arguments to ensure secrets and other important data aren't inadvertently shared with others. What are the ways the remaining configuration can be added to Terraform so it can initialize and communicate with the backend? (select three)

A. directly querying HashiCorp Vault for the secrets

B. command-line key/value pairs

C. use the -backend-config=PATH to specify a separate config file

D. interactively on the command line

Answer: B,C,D (LEAVE A REPLY)

Explanation

You do not need to specify every required argument in the backend configuration. Omitting certain arguments may be desirable to avoid storing secrets, such as access keys, within the main configuration. When some or all of the arguments are omitted, we call this a partial configuration.

With a partial configuration, the remaining configuration arguments must be provided as part of the initialization process. There are several ways to supply the remaining arguments:

<https://www.terraform.io/docs/backends/init.html#backend-initialization>

NEW QUESTION: 86

Which option can not be used to keep secrets out of Terraform configuration files?

- A. secure string
- B. Environment variables
- C. A -var flag
- D. A Terraform provider

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 87

From the code below, identify the implicit dependency:

- A. The AMI used for the EC2 instance
- B. The S3 bucket labeled company_data
- C. The EC2 instance labeled web_server
- D. The EIP with an id of ami-2757f631

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 88

Which of the below options is the equivalent Terraform 0.12 version of the snippet which is written in Terraform 0.11?

`"${var.instance_id}"`

- A. `var.instance_ids`
- B. `var.instance_id`
- C. `variable.instance_id`
- D. None of the above

Answer: B ([LEAVE A REPLY](#))

NEW QUESTION: 89

What does the default "local" Terraform backend store?

- A. tfplan files
- B. Terraform binary
- C. Provider plugins
- D. State file

Answer: D ([LEAVE A REPLY](#))

The local backend stores state on the local filesystem, locks that state using system APIs, and performs operations locally.

NEW QUESTION: 90

You want terraform plan and apply to be executed in Terraform Cloud's run environment but the output is to be streamed locally. Which one of the below you will choose?

- A. Local Backends
- B. This can be done using any of the local or remote backends
- C. Remote Backends
- D. Terraform Backends

Answer: ([SHOW ANSWER](#))

The remote backend stores Terraform state and may be used to run operations in Terraform Cloud.

When using full remote operations, operations like terraform plan or terraform apply can be executed in Terraform Cloud's run environment, with log output streaming to the local terminal. Remote plans and applies use variable values from the associated Terraform Cloud workspace.
<https://www.terraform.io/docs/backends/types/remote.html>

NEW QUESTION: 91

Terraform-specific settings and behaviors are declared in which configuration block type?

- A. provider
- B. terraform
- C. resource
- D. data

Answer: B ([LEAVE A REPLY](#))

The special terraform configuration block type is used to configure some behaviors of Terraform itself, such as requiring a minimum Terraform version to apply your configuration.

Example

```
terraform {  
  required_version = "> 0.12.0"  
}
```

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

Valid TA-002-P Dumps shared by PrepPdf.com for Helping Passing TA-002-P Exam!

PrepPdf.com now offer the **newest TA-002-P exam dumps**, the PrepPdf.com TA-002-P exam **questions have been updated** and **answers have been corrected** get the **newest** PrepPdf.com TA-002-P dumps with Test Engine here:

<https://www.preppdf.com/HashiCorp/TA-002-P-prepaway-exam-dumps.html> (94 Q&As Dumps, **40%OFF Special Discount: Exam-Tests**)

NEW QUESTION: 92

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: (SHOW ANSWER)

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>

NEW QUESTION: 93

You have a Terraform configuration file where a variable itemNum is defined as follows:

```
variable "itemNum" { default = 3 }
```

You also have a defined the following environment variables in your shell: TF_itemNum =6, TF_VAR_itemNum =9. You also have a terraform.tfvars file with the following contents itemNum = 7 When you run the following apply command, what is the value assigned to the itemNum variable?

```
terraform apply -var itemNum =4
```

- A. 10
- B. 6
- C. 1
- D. 4
- E. 3

Answer: D (LEAVE A REPLY)

The -var and -var-file methods of assigning variables have the highest precedence.

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

NEW QUESTION: 94

You need to deploy resources into two different cloud regions in the same Terraform configuration. To do that, you declare multiple provider configurations as follows:

What meta-argument do you need to configure in a resource block to deploy the resource to the "us-west-2" AWS region?

- A. alias = west
- B. provider = aws.west

C. alias = aws.west

D. provider = west

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 95

Which backend does the Terraform CLI use by default?

A. Local

B. Terraform Cloud

C. Consul

D. Remote

Answer: A ([LEAVE A REPLY](#))

NEW QUESTION: 96

While attempting to deploy resources into your cloud provider using Terraform, you begin to see some odd behavior and experience sluggish responses. In order to troubleshoot you decide to turn on Terraform debugging. Which environment variables must be configured to make Terraform's logging more verbose?

A. TF_10G_PATM

B. TF.LOG.FUE

C. TF_LOG

D. TF_10G_LEVEL

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 97

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 ([LEAVE A REPLY](#))

Explanation

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

NEW QUESTION: 98

By default, where does Terraform store its state file?

A. current working directory

Explanation

By default, the state file is stored in a local file named "terraform.tfstate", but it can also be stored remotely, which works better in a team environment.

- B. shared directory
- C. remotely using Terraform Cloud
- D. Amazon S3 bucket

Answer: A ([LEAVE A REPLY](#))

NEW QUESTION: 99

Which of the following is not a valid string function in Terraform?

- A. split
- B. join
- C. slice
- D. chomp

Answer: ([SHOW ANSWER](#)**)**

Reference: <https://www.terraform.io/docs/language/functions/chomp.html>

NEW QUESTION: 100

Select two answers to complete the following sentence: Before a new provider can be used, it must be _____ and _____.

- A. approved by HashiCorp
- B. uploaded to source control
- C. declared in the configuration
- D. initialized

Answer: ([SHOW ANSWER](#)**)**

Each time a new provider is added to configuration -- either explicitly via a provider block or by adding a resource from that provider -- Terraform must initialize the provider before it can be used. Initialization downloads and installs the provider's plugin so that it can later be executed.

NEW QUESTION: 101

During a terraform plan, a resource is successfully created but eventually fails during provisioning. What happens to the resource?

- A. Terraform attempts to provision the resource up to three times before exiting with an error
- B. the terraform plan is rolled back and all provisioned resources are removed
- C. it is automatically deleted
- D. the resource is marked as tainted

Answer: D ([LEAVE A REPLY](#))

Explanation

If a resource successfully creates but fails during provisioning, Terraform will error and mark the resource as

"tainted". A resource that is tainted has been physically created, but can't be considered safe to use since provisioning failed. Terraform also does not automatically roll back and destroy the resource during the apply when the failure happens, because that would go against the execution

plan: the execution plan would've said a resource will be created, but does not say it will ever be deleted.

NEW QUESTION: 102

A user creates three workspaces from the command line - prod, dev, and test. Which of the following commands will the user run to switch to the dev workspace?

- A. terraform workspace -switch dev
- B. terraform workspace select dev
- C. terraform workspace switch dev

Explanation

The terraform workspace select command is used to choose a different workspace to use for further operations. <https://www.terraform.io/docs/commands/workspace/select.html>

- D. terraform workspace dev

Answer: B ([LEAVE A REPLY](#))

NEW QUESTION: 103

You have never used Terraform before and would like to test it out using a shared team account for a cloud provider. The shared team account already contains 15 virtual machines (VM). You develop a Terraform configuration containing one VM, perform terraform apply, and see that your VM was created successfully.

What should you do to delete the newly-created VM with Terraform?

- A. The Terraform state file only contains the one new VM. Execute terraform destroy.
- B. Delete the Terraform state file and execute Terraform apply.
- C. The Terraform state file contains all 16 VMs in the team account. Execute terraform destroy and select the newly-created VM.
- D. Delete the VM using the cloud provider console and terraform apply to apply the changes to the Terraform state file.

Answer: ([SHOW ANSWER](#)**)**

NEW QUESTION: 104

Only the user that generated a plan may apply it.

- A. True
- B. False

Answer: A ([LEAVE A REPLY](#))

Explanation

The optional -out argument can be used to save the generated plan to a file for later execution with terraform apply, which can be useful when running Terraform in automation.

Reference: <https://learn.hashicorp.com/tutorials/terraform/automate-terraform>

NEW QUESTION: 105

State is a requirement for Terraform to function

A. True

B. False

Answer: A ([LEAVE A REPLY](#))

State is a necessary requirement for Terraform to function. It is often asked if it is possible for Terraform to work without state, or for Terraform to not use state and just inspect cloud resources on every run.

Purpose of Terraform State

State is a necessary requirement for Terraform to function. It is often asked if it is possible for Terraform to work without state, or for Terraform to not use state and just inspect cloud resources on every run. This page will help explain why Terraform state is required.

As you'll see from the reasons below, state is required. And in the scenarios where Terraform may be able to get away without state, doing so would require shifting massive amounts of complexity from one place (state) to another place (the replacement concept).

1. Mapping to the Real World

Terraform requires some sort of database to map Terraform config to the real world. When you have a resource resource "aws_instance" "foo" in your configuration, Terraform uses this map to know that instance i- abcd1234 is represented by that resource.

For some providers like AWS, Terraform could theoretically use something like AWS tags. Early prototypes of Terraform actually had no state files and used this method. However, we quickly ran into problems. The first major issue was a simple one: not all resources support tags, and not all cloud providers support tags.

Therefore, for mapping configuration to resources in the real world, Terraform uses its own state structure.

2. Metadata

Alongside the mappings between resources and remote objects, Terraform must also track metadata such as resource dependencies.

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.

One way to avoid this would be for Terraform to know a required ordering between resource types. For example, Terraform could know that servers must be deleted before the subnets they are a part of. The complexity for this approach quickly explodes, however: in addition to Terraform having to understand the ordering semantics of every resource for every cloud, Terraform must also understand the ordering across providers.

Terraform also stores other metadata for similar reasons, such as a pointer to the provider configuration that was most recently used with the resource in situations where multiple aliased providers are present.

3. Performance

In addition to basic mapping, Terraform stores a cache of the attribute values for all resources in the state. This is the most optional feature of Terraform state and is done only as a performance improvement.

When running a terraform plan, Terraform must know the current state of resources in order to effectively determine the changes that it needs to make to reach your desired configuration.

For small infrastructures, Terraform can query your providers and sync the latest attributes from all your resources. This is the default behavior of Terraform: for every plan and apply, Terraform will sync all resources in your state.

For larger infrastructures, querying every resource is too slow. Many cloud providers do not provide APIs to query multiple resources at once, and the round trip time for each resource is hundreds of milliseconds. On top of this, cloud providers almost always have API rate limiting so Terraform can only request a certain number of resources in a period of time. Larger users of Terraform make heavy use of the `-refresh=false` flag as well as the `-target` flag in order to work around this. In these scenarios, the cached state is treated as the record of truth.

4. Syncing

In the default configuration, Terraform stores the state in a file in the current working directory where Terraform was run. This is okay for getting started, but 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.

Remote state is the recommended solution to this problem. With a fully-featured state backend, Terraform can use remote locking as a measure to avoid two or more different users accidentally running Terraform at the same time, and thus ensure that each Terraform run begins with the most recent updated state.

NEW QUESTION: 106

The Terraform language does not support user-defined functions, and so only the functions built in to the language are available for use.

A. False

B. True

Answer: B ([LEAVE A REPLY](#))

Explanation

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

Valid TA-002-P Dumps shared by PrepPdf.com for Helping Passing TA-002-P Exam!

PrepPdf.com now offer the **newest TA-002-P exam dumps**, the PrepPdf.com TA-002-P exam

questions have been updated and answers have been corrected get the **newest**

PrepPdf.com TA-002-P dumps with Test Engine here:

<https://www.preppdf.com/HashiCorp/TA-002-P-prepaway-exam-dumps.html> (94 Q&As Dumps,

40%OFF Special Discount: Exam-Tests)

NEW QUESTION: 107

What command should you run to display all workspaces for the current configuration?

- A. terraform workspace
- B. terraform workspace show
- C. terraform workspace list
- D. terraform show workspace

Answer: (SHOW ANSWER)

terraform workspace list

The command will list all existing workspaces.

Reference: <https://www.terraform.io/docs/cli/commands/workspace/list.html>

NEW QUESTION: 108

Named workspaces are not a suitable isolation mechanism for strong separation between staging and production?

- A. True
- B. False

Answer: A (LEAVE A REPLY)

Explanation

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#when-to-use-multiple-workspaces>

NEW QUESTION: 109

Terraform will sync all resources in state by default for every plan and apply, hence for larger infrastructures this can slow down terraform plan and terraform apply commands?

- A. False
- B. True

Answer: B (LEAVE A REPLY)

Explanation

For small infrastructures, Terraform can query your providers and sync the latest attributes from all your resources. This is the default behavior of Terraform: for every plan and apply, Terraform will sync all resources in your state.

For larger infrastructures, querying every resource is too slow. Many cloud providers do not provide APIs to query multiple resources at once, and the round trip time for each resource is hundreds of milliseconds. On top of this, cloud providers almost always have API rate limiting so Terraform can only request a certain number of resources in a period of time. Larger users of Terraform make heavy use of the `-refresh=false` flag as well as the `-target` flag in order to work around this. In these scenarios, the cached state is treated as the record of truth.

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

NEW QUESTION: 110

Which task does terraform init not perform?

- A. Validates all required variables are present
- B. Connects to the backend
- C. Sources all providers present in the configuration and ensures they are downloaded and available locally
- D. Sources any modules and copies the configuration locally

Answer: A ([LEAVE A REPLY](#))

NEW QUESTION: 111

Choose the answer that correctly completes the sentence: _____backends support state locking.

- A. Only local
- B. All
- C. Some
- D. No

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 112

Anyone can publish and share modules on the Terraform Public Module Registry, and meeting the requirements for publishing a module is extremely easy. Select from the following list all valid requirements. (select three)

- A. The module must be PCI/HIPPA compliant.
- B. Module repositories must use this three-part name format, terraform-- .
- C. The registry uses tags to identify module versions.
- D. Release tag names must be for the format x.y.z, and can optionally be prefixed with a v .
- E. The module must be on GitHub and must be a public repo.

Answer: C,D,E ([LEAVE A REPLY](#))

Explanation

<https://www.terraform.io/docs/registry/modules/publish.html#requirements>

NEW QUESTION: 113

You want to use terraform import to start managing infrastructure that was not originally provisioned through infrastructure as code. Before you can import the resource's current state, what must you do in order to prepare to manage these resources using Terraform?

- A.** Run terraform refresh to ensure that the state file has the latest information for existing resources.
- B.** Update the configuration file to include the new resources.
- C.** Shut down or stop using the resources being imported so no changes are inadvertently missed.
- D.** Modify the Terraform state file to add the new resources.

Answer: B ([LEAVE A REPLY](#))

The current implementation of Terraform import can only import resources into the state. It does not generate configuration. A future version of Terraform will also generate configuration. Because of this, prior to running terraform import it is necessary to write manually a resource configuration block for the resource, to which the imported object will be mapped.

The terraform import command is used to import existing infrastructure.

To import a resource, first write a resource block for it in our configuration, establishing the name by which it will be known to Terraform.

Example:

```
resource "aws_instance" "import_example" {  
  # ...instance configuration...  
}
```

Now terraform import can be run to attach an existing instance to this resource configuration.

```
$ terraform import aws_instance.import_example i-03efafa258104165f  
aws_instance.import_example: Importing from ID "i-03efafa258104165f"...  
aws_instance.import_example: Import complete!
```

```
Imported aws_instance (ID: i-03efafa258104165f)
```

```
aws_instance.import_example: Refreshing state... (ID: i-03efafa258104165f) Import successful!
```

The resources that were imported are shown above. These resources are now in your Terraform state and will henceforth be managed by Terraform.

This command locates the AWS instance with ID i-03efafa258104165f (which has been created outside Terraform) and attaches its existing settings, as described by the EC2 API, to the name aws_instance.import_example in the Terraform state.

NEW QUESTION: 114

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 ([LEAVE A REPLY](#))

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>

NEW QUESTION: 115

What is one disadvantage of using dynamic blocks in Terraform?

- A.** They make configuration harder to read and understand
- B.** Terraform will run more slowly
- C.** Dynamic blocks can construct repeatable nested blocks
- D.** They cannot be used to loop through a list of values

Answer: D ([LEAVE A REPLY](#))

NEW QUESTION: 116

1. resource "aws_s3_bucket" "example" {
2. bucket = "my-test-s3-terraform-bucket"
3. ...} resource "aws_iam_role" "test_role" {
4. name = "test_role"
5. ...}

Due to the way that the application code is written, the s3 bucket must be created before the test role is created, otherwise there will be a problem. How can you ensure that?

- A.** Add explicit dependency using depends_on . This will ensure the correct order of resource creation.
- B.** This will already be taken care of by terraform native implicit dependency. Nothing else needs to be done from your end.
- C.** This is not possible to control in terraform . Terraform will take care of it in a native way , and create a dependency graph that is best suited for the parallel resource creation.
- D.** Create 2 separate terraform config scripts , and run them one by one , 1 for s3 bucket , and another for IAM role , run the S3 bucket script first.

Answer: A ([LEAVE A REPLY](#))

Implicit dependency works only if there is some reference of one resource to another. Explicit dependency is the option here.

NEW QUESTION: 117

You want to get involved in the development of Terraform. As this is an open source project, you would like to contribute a fix for an open issue of Terraform. What programming language will need to use to write the fix?

- A. It depends on which command issue related to.
- B. Python
- C. Go
- D. Java

Answer: C ([LEAVE A REPLY](#))

Basic programming knowledge. Terraform and Terraform Plugins are written in the Go programming language, but even if you've never written a line of Go before, you're still welcome to take a dive into the code and submit patches. The community is happy to assist with code reviews and offer guidance specific to Go.

NEW QUESTION: 118

Which of the following is not a benefit of adopting infrastructure as code?

- A. Reusability of code
- B. Automation
- C. Interpolation
- D. Versioning

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 119

A terraform apply can not _____ infrastructure.

- A. provision
- B. change
- C. import
- D. destroy

Answer: B ([LEAVE A REPLY](#))

NEW QUESTION: 120

If a Terraform creation-time provisioner fails, what will occur by default?

- A. The resource will be destroyed
- B. The resource will be marked as "tainted"
- C. The resource will not be affected, but the provisioner will need to be applied again
- D. Nothing, provisioners will not show errors in the command line

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 121

Which of the following can you do with terraform plan? Choose two correct answers.

- A. Save a generated execution plan to apply later
- B. Schedule Terraform to run at a planned time in the future
- C. View the execution plan and check if the changes match your expectations
- D. Execute a plan in a different workspace

Answer: A,C ([LEAVE A REPLY](#))

Valid TA-002-P Dumps shared by PrepPdf.com for Helping Passing TA-002-P Exam!

PrepPdf.com now offer the **newest TA-002-P exam dumps**, the PrepPdf.com TA-002-P exam **questions have been updated** and **answers have been corrected** get the **newest** PrepPdf.com TA-002-P dumps with Test Engine here:

<https://www.preppdf.com/HashiCorp/TA-002-P-prepaway-exam-dumps.html> (94 Q&As Dumps, **40%OFF** Special Discount: **Exam-Tests**)

NEW QUESTION: 122

One remote backend configuration always maps to a single remote workspace.

- A. True
- B. False

Answer: A ([LEAVE A REPLY](#))

Explanation/Reference: <https://www.terraform.io/docs/language/settings/backends/remote.html>

NEW QUESTION: 123

Which argument(s) are required when declaring a Terraform variable?

- A. description
- B. type
- C. default
- D. All of the above
- E. None of the above

Answer: B ([LEAVE A REPLY](#))

NEW QUESTION: 124

When using constraint expressions to signify a version of a provider, which of the following are valid provider versions that satisfy the expression found in the following code snippet: (select two)

1. terraform
2. {
3. required_providers
4. {

5. aws = "~> 1.2.0"

6. }

7. }

A. 1.3.1

B. 1.2.3

C. 1.2.9

D. 1.3.0

Answer: B,C (LEAVE A REPLY)

As your Terraform usage becomes more advanced, there are some cases where you may need to modify the Terraform state. Rather than modify the state directly, the terraform state commands can be used in many cases instead. This command is a nested subcommand, meaning that it has further subcommands.

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

NEW QUESTION: 125

Given the Terraform configuration below, in which order will the resources be created?

A. Larger image

B. resources will be created simultaneously

C. aws_eip will be created first aws_instance will be created second

D. aws_instance will be created first aws_eip will be created second

Answer: (SHOW ANSWER)

Explanation

The aws_instance will be created first, and then aws_eip will be created second due to the aws_eip's resource dependency of the aws_instance id

NEW QUESTION: 126

terraform state subcommands such as list are read-only commands, do read-only commands create state backup files?

A. Yes

B. No

Answer: (SHOW ANSWER)

Explanation

Subcommands that are read-only (such as list) do not write any backup files since they aren't modifying the state.

All terraform state subcommands that modify the state write backup files. The path of these backup file can be controlled with -backup.

<https://www.terraform.io/docs/commands/state/index.html#backups>

NEW QUESTION: 127

Which Terraform command will force a marked resource to be destroyed and recreated on the next apply?

- A. terraform fmt
- B. terraform destroy
- C. terraform taint
- D. terraform refresh

Answer: ([SHOW ANSWER](#))

The terraform taint command manually marks a Terraform-managed resource as tainted, forcing it to be destroyed and recreated on the next apply.

This command will not modify infrastructure, but does modify the state file in order to mark a resource as tainted. Once a resource is marked as tainted, the next plan will show that the resource will be destroyed and recreated and the next apply will implement this change.

Forcing the recreation of a resource is useful when you want a certain side effect of recreation that is not visible in the attributes of a resource. For example: re-running provisioners will cause the node to be different or rebooting the machine from a base image will cause new startup scripts to run.

Note that tainting a resource for recreation may affect resources that depend on the newly tainted resource. For example, a DNS resource that uses the IP address of a server may need to be modified to reflect the potentially new IP address of a tainted server. The plan command will show this if this is the case.

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

NEW QUESTION: 128

Refer to the following terraform variable definition

```
variable "track_tag" { type = list default = ["data_ec2","integration_ec2","digital_ec2"]} track_tag = { Name = element(var.track_tag,count.index)}
```

If count.index is set to 2, which of the following values will be assigned to the name attribute of track_tag variable?

- A. data_ec2
- B. track_tag
- C. digital_ec2
- D. integration_ec2

Answer: C ([LEAVE A REPLY](#))

NEW QUESTION: 129

Terraform Cloud is more powerful when you integrate it with your version control system (VCS) provider. Select all the supported VCS providers from the answers below. (select four)

- A. GitHub
- B. Azure DevOps Server
- C. GitHub Enterprise

Explanation

Terraform Cloud supports the following VCS providers:

- <https://www.terraform.io/docs/cloud/vcs/github.html>
- <https://www.terraform.io/docs/cloud/vcs/github.html>

- <https://www.terraform.io/docs/cloud/vcs/github-enterprise.html>
- <https://www.terraform.io/docs/cloud/vcs/gitlab-com.html>
- <https://www.terraform.io/docs/cloud/vcs/gitlab-eece.html>
- <https://www.terraform.io/docs/cloud/vcs/bitbucket-cloud.html>
- <https://www.terraform.io/docs/cloud/vcs/bitbucket-server.html>
- <https://www.terraform.io/docs/cloud/vcs/azure-devops-server.html>
- <https://www.terraform.io/docs/cloud/vcs/azure-devops-services.html>

<https://www.terraform.io/docs/cloud/vcs/index.html#supported-vcs-providers>

D. Bitbucket Cloud

E. CVS Version Control

Answer: A,B,C,D ([LEAVE A REPLY](#))

NEW QUESTION: 130

Terraform provisioners can be added to any resource block.

A. True

B. False

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 131

Multiple configurations for the same provider can be used in a single configuration file.

A. False

B. True

Answer: B ([LEAVE A REPLY](#))

Explanation

You can optionally define multiple configurations for the same provider, and select which one to use on a per-resource or per-module basis. The primary reason for this is to support multiple regions for a cloud platform; other examples include targeting multiple Docker hosts, multiple Consul hosts, etc.

To include multiple configurations for a given provider, include multiple provider blocks with the same provider name, but set the alias meta-argument to an alias name to use for each additional configuration. For example:

```
# The default provider configuration
```

```
provider "aws" {  
  region = "us-east-1"  
}
```

```
# Additional provider configuration for west coast region
```

```
provider "aws" {  
  alias = "west"  
  region = "us-west-2"  
}
```

The provider block without alias set is known as the default provider configuration. When alias is set, it creates an additional provider configuration. For providers that have no required configuration arguments, the implied empty configuration is considered to be the default provider configuration.

<https://www.terraform.io/docs/configuration/providers.html#alias-multiple-provider-instances>

NEW QUESTION: 132

In Terraform 0.13 and above, outside of the required_providers block, Terraform configurations always refer to providers by their local names.

A. True

B. False

Answer: (SHOW ANSWER)

Outside of the required_providers block, Terraform configurations always refer to providers by their local names.

NEW QUESTION: 133

What allows you to conveniently switch between multiple instances of a single configuration within its single backend?

A. Local backends

B. Providers

C. Remote backends

D. Workspaces

Answer: D (LEAVE A REPLY)

Explanation

Named workspaces allow conveniently switching between multiple instances of a single configuration within its single backend. ... A common use for multiple workspaces is 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.

Workspaces, allowing multiple states to be associated with a single configuration. The configuration still has only one backend, but multiple distinct instances of that configuration to be deployed without configuring a new backend or changing authentication credentials.

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

NEW QUESTION: 134

Module variable assignments are inherited from the parent module and do not need to be explicitly set.

A. False

B. True

Answer: A (LEAVE A REPLY)

NEW QUESTION: 135

You have recently started a new job at a retailer as an engineer. As part of this new role, you have been tasked with evaluating multiple outages that occurred during peak shopping time during the holiday season.

Your investigation found that the team is manually deploying new compute instances and configuring each compute instance manually. This has led to inconsistent configuration between each compute instance.

How would you solve this using infrastructure as code?

- A.** Implement a provisioning pipeline that deploys infrastructure configurations committed to your version control system following code reviews
- B.** Implement a ticketing workflow that makes engineers submit a ticket before manually provisioning and configuring a resource
- C.** Replace the compute instance type with a larger version to reduce the number of required deployments
- D.** Implement a checklist that engineers can follow when configuring compute instances

Answer: B ([LEAVE A REPLY](#))

NEW QUESTION: 136

What feature of Terraform Cloud and/or Terraform Enterprise can you publish and maintain a set of custom modules which can be used within your organization?

- A.** private module registry
- B.** custom VCS integration
- C.** remote runs
- D.** Terraform registry

Answer: A ([LEAVE A REPLY](#))

Valid TA-002-P Dumps shared by PrepPdf.com for Helping Passing TA-002-P Exam!

PrepPdf.com now offer the **newest TA-002-P exam dumps**, the PrepPdf.com TA-002-P exam **questions have been updated** and **answers have been corrected** get the **newest** PrepPdf.com TA-002-P dumps with Test Engine here:

<https://www.preppdf.com/HashiCorp/TA-002-P-prepaway-exam-dumps.html> (94 Q&As Dumps,

40%OFF Special Discount: Exam-Tests)

NEW QUESTION: 137

True or False. The terraform refresh command is used to reconcile the state Terraform knows about (via its state file) with the real-world infrastructure. If drift is detected between the real-world infrastructure and the last known-state, it will modify the infrastructure to correct the drift.

- A.** False
- B.** True

Answer: A ([LEAVE A REPLY](#))

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

NEW QUESTION: 138

Command terraform refresh will update state file?

- A. False
- B. True

Answer: (SHOW ANSWER)

Explanation

The terraform refresh command is used to reconcile the state Terraform knows about (via its state file) with the real-world infrastructure. This can be used to detect any drift from the last-known state, and to update the state file.

This does not modify infrastructure, but does modify the state file. If the state is changed, this may cause changes to occur during the next plan or apply.

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

NEW QUESTION: 139

After running into issues with Terraform, you need to enable verbose logging to assist with troubleshooting the error. Which of the following values provides the MOST verbose logging?

- A. ERROR
- B. INFO
- C. WARN
- D. TRACE
- E. DEBUG

Answer: D (LEAVE A REPLY)

Terraform has detailed logs that can be enabled by setting the TF_LOG environment variable to any value. This will cause detailed logs to appear on stderr.

You can set TF_LOG to one of the log levels TRACE, DEBUG, INFO, WARN or ERROR to change the verbosity of the logs. TRACE is the most verbose and it is the default if TF_LOG is set to something other than a log level name.

Examples:

```
export TF_LOG=DEBUG
```

```
export TF_LOG=TRACE
```

NEW QUESTION: 140

Complete the following sentence:

The terraform state command can be used to _____

- A. view state
- B. refresh state
- C. modify state
- D. There is no such command

Explanation

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

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 141

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

A. False

B. True

Answer: A ([LEAVE A REPLY](#))

If you need to migrate 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.

<https://www.terraform.io/docs/cloud/migrate/index.html>

NEW QUESTION: 142

What features does the hosted service Terraform Cloud provide? (Choose two.)

A. Automated infrastructure deployment visualization

B. Automatic backups

C. Remote state storage

D. A web-based user interface (UI)

Answer: ([SHOW ANSWER](#))

Reference:

<https://www.terraform.io/docs/enterprise/admin/automated-recovery.html>

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

NEW QUESTION: 143

A user has created a module called "my_test_module" and committed it to GitHub. Over time, several commits have been made with updates to the module, each tagged in GitHub with an incremental version number. Which of the following lines would be required in a module configuration block in terraform to select tagged version v1.0.4?

A. source = "git::https://example.com/my_test_module.git?ref=v1.0.4"

Explanation

<https://www.terraform.io/docs/modules/sources.html#selecting-a-revision>

B. source = "git::https://example.com/my_test_module.git&ref=v1.0.4"

C. source = "git::https://example.com/my_test_module.git@tag=v1.0.4"

D. source = "git::https://example.com/my_test_module.git#tag=v1.0.4"

Answer: A ([LEAVE A REPLY](#))

NEW QUESTION: 144

does not require GO language to be installed as a prerequisite and it does not require a Windows Server as well.

- A. True
- B. False

Answer: B ([LEAVE A REPLY](#))

NEW QUESTION: 145

Which one of the following will run echo 0 and echo 1 on a newly created host?

A. provisioner "local-exec" {
 command = "echo 0"
 command = "echo 1"
}

B. provisioner "remote-exec" {
 inline = [
 echo 0,
 echo 1
]
}

C. provisioner "remote-exec" {
 command = "\${echo 0}"
 command = "\${echo 1}"
}

D. provisioner "remote-exec" {
 inline = [
 "echo 0",
 "echo 1"
]
}

Answer: D ([LEAVE A REPLY](#))

remote-exec Provisioner

Example usage

```
resource "aws_instance" "web" {  
  # ...  
  provisioner "remote-exec" {  
    inline = [  
      "puppet apply",  
      "consul join ${aws_instance.web.private_ip}",  
    ]  
  }  
}
```

NEW QUESTION: 146

State locking does not happen automatically and must be specified at run

- A. False

B. True

Answer: ([SHOW ANSWER](#))

Explanation

State locking happens automatically on all operations that could write state.

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

NEW QUESTION: 147

Which of the following is not a valid string function in Terraform?

A. split

B. join

C. slice

D. chomp

Answer: C ([LEAVE A REPLY](#))

NEW QUESTION: 148

What is the command you can use to set an environment variable named "var1" of type String?

A. export TF_VAR_VAR1

B. set TF_VAR_var1

C. variable "var1" { type = "string" }

D. export TF_VAR_var1

Answer: D ([LEAVE A REPLY](#))

Explanation

The environment variable must be in the format TF_VAR_name, so for the question:

TF_VAR_var1 is the correct choice.

https://www.terraform.io/docs/commands/environment-variables.html#tf_var_name

NEW QUESTION: 149

Which of the following value will be accepted for var1?

```
variable "var1" {
```

```
type = string
```

```
}
```

A. None of the above

B. Both A and B

C. "5"

D. 5

Answer: B ([LEAVE A REPLY](#))

Explanation

Terraform automatically converts number and bool values to strings when needed.

NEW QUESTION: 150

When using a module block to reference a module stored on the public Terraform Module Registry such as:

How do you specify version 1.0.0?

- A. Modules stored on the public Terraform Module Registry do not support versioning
- B. Append ?ref=v1.0.0 argument to the source path
- C. Add version = "1.0.0" attribute to module block
- D. Nothing - modules stored on the public Terraform Module Registry always default to version 1.0.0

Answer: A ([LEAVE A REPLY](#))

Explanation/Reference: <https://www.terraform.io/docs/language/modules/sources.html>

NEW QUESTION: 151

Which statement describes a goal of infrastructure as code?

- A. Write once, run anywhere
- B. The programmatic configuration of resources
- C. A pipeline process to test and deliver software
- D. An abstraction from vendor specific APIs

Answer: B ([LEAVE A REPLY](#))

Valid TA-002-P Dumps shared by PrepPdf.com for Helping Passing TA-002-P Exam!

PrepPdf.com now offer the **newest TA-002-P exam dumps**, the PrepPdf.com TA-002-P exam **questions have been updated** and **answers have been corrected** get the **newest** PrepPdf.com TA-002-P dumps with Test Engine here:

<https://www.preppdf.com/HashiCorp/TA-002-P-prepaway-exam-dumps.html> (94 Q&As Dumps,

40%OFF Special Discount: Exam-Tests)

NEW QUESTION: 152

You have provisioned some virtual machines (VMs) on Google Cloud Platform (GCP) using the gcloud command line tool. However, you are standardizing with Terraform and want to manage these VMs using Terraform instead.

What are the two things you must do to achieve this? (Choose two.)

- A. Provision new VMs using Terraform with the same VM names
- B. Use the terraform import command for the existing VMs
- C. Write Terraform configuration for the existing VMs
- D. Run the terraform import-gcp command

Answer: ([SHOW ANSWER](#)**)**

Explanation

The terraform import command is used to import existing infrastructure. Import existing Google Cloud resources into Terraform with Terraformer.

Reference: <https://www.terraform.io/docs/cli/import/usage.html>
<https://cloud.google.com/docs/terraform>

NEW QUESTION: 153

Terraform Cloud always encrypts state at rest and protects it with TLS in transit. Terraform Cloud also knows the identity of the user requesting state and maintains a history of state changes.

A. False

B. True

Answer: (SHOW ANSWER)

Explanation

Terraform Cloud always encrypts state at rest and protects it with TLS in transit. Terraform Cloud also knows the identity of the user requesting state and maintains a history of state changes. This can be used to control access and track activity. Terraform Enterprise also supports detailed audit logging.

<https://www.terraform.io/docs/state/sensitive-data.html#recommendations>

NEW QUESTION: 154

A provider configuration block is required in every Terraform configuration.

Example:

A. True

B. False

Answer: A (LEAVE A REPLY)

NEW QUESTION: 155

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 (LEAVE A REPLY)

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.

NEW QUESTION: 156

Which argument(s) is (are) required when declaring a Terraform variable?

- A. type
- B. default
- C. description
- D. All of the above
- E. None of the above

Answer: (SHOW ANSWER)

Explanation

The variable declaration can also include a default argument.

Reference: <https://www.terraform.io/docs/language/values/variables.html>

NEW QUESTION: 157

You write a new Terraform configuration and immediately run terraform apply in the CLI using the local backend.

Why will the apply fail?

- A. The Terraform CLI needs you to log into Terraform cloud first
- B. Terraform needs you to format your code according to best practices first
- C. Terraform needs to install the necessary plugins first
- D. Terraform requires you to manually run terraform plan first

Answer: A (LEAVE A REPLY)

NEW QUESTION: 158

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 (LEAVE A REPLY)

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.

NEW QUESTION: 159

You are reviewing Terraform configurations for a big project in your company. You noticed that there are several identical sets of resources that appear in multiple configurations. What feature of Terraform would you recommend to use to reduce the amount of cloned configuration between the different configurations?

- A. Packages
- B. Backends
- C. Provisioners
- D. Modules

Answer: ([SHOW ANSWER](#))

Explanation

Modules are reusable configuration packages that Terraform can share through a variety of sources including Terraform Registries, GitHub, and Amazon S3 buckets.

A module is a container for multiple resources that are used together. Modules can be used to create lightweight abstractions, so that you can describe your infrastructure in terms of its architecture, rather than directly in terms of physical objects.

Modules are reusable configuration packages that Terraform can share through a variety of sources including Terraform Registries, GitHub, and Amazon S3 buckets.

<https://www.terraform.io/docs/modules/index.html>

NEW QUESTION: 160

You should store secret data in the same version control repository as your Terraform configuration.

- A. False
- B. True

Answer: A ([LEAVE A REPLY](#))

NEW QUESTION: 161

What is one disadvantage of using dynamic blocks in Terraform?

- A. They cannot be used to loop through a list of values
- B. Dynamic blocks can construct repeatable nested blocks
- C. They make configuration harder to read and understand
- D. Terraform will run more slowly

Answer: ([SHOW ANSWER](#))

Reference: <https://github.com/hashicorp/terraform/issues/19291>

NEW QUESTION: 162

Which option can not be used to keep secrets out of Terraform configuration files?

- A. A Terraform provider
- B. Environment variables
- C. A -var flag
- D. secure string

Answer: C ([LEAVE A REPLY](#))

Reference: <https://secrethub.io/blog/secret-management-for-terraform/>

NEW QUESTION: 163

Which of the following statements about local modules is incorrect:

- A. Local modules support versions
- B. Local modules are not cached by terraform init command
- C. Local modules are sourced from a directory on disk
- D. None of the above (all statements above are correct)
- E. All of the above (all statements above are incorrect)

Answer: A ([LEAVE A REPLY](#))

NEW QUESTION: 164

What kind of configuration block will create an infrastructure object with settings specified in the block?

- A. provider
- B. data
- C. state
- D. resource

Answer: D ([LEAVE A REPLY](#))

NEW QUESTION: 165

You want terraform plan and apply to be executed in Terraform Cloud's run environment but the output is to be streamed locally. Which one of the below you will choose?

- A. Local Backends
- B. This can be done using any of the local or remote backends
- C. Remote Backends
- D. Terraform Backends

Answer: (SHOW ANSWER)

Explanation

The remote backend stores Terraform state and may be used to run operations in Terraform Cloud.

When using full remote operations, operations like terraform plan or terraform apply can be executed in Terraform Cloud's run environment, with log output streaming to the local terminal. Remote plans and applies use variable values from the associated Terraform Cloud workspace.
<https://www.terraform.io/docs/backends/types/remote.html>

NEW QUESTION: 166

What does terraform refresh command do?

- A. terraform refresh can be used to selectively update sections of the state file, using terraform resource level addressing.
- B. terraform refresh is use to change/modify the infrastructure based on the existing state file, at that moment.
- C. terraform refresh command basically updates the configuration file with the current state of the actual infrastructure
- D. terraform refresh can be used to selectively update sections of the state file, using terraform resource level addressing.
- E. terraform refresh syncs the state file with the real world infrastructure.

Answer: E ([LEAVE A REPLY](#))

Valid TA-002-P Dumps shared by PrepPdf.com for Helping Passing TA-002-P Exam!

PrepPdf.com now offer the **newest TA-002-P exam dumps**, the PrepPdf.com TA-002-P exam **questions have been updated** and **answers have been corrected** get the **newest** PrepPdf.com TA-002-P dumps with Test Engine here:

<https://www.preppdf.com/HashiCorp/TA-002-P-prepaway-exam-dumps.html> (94 Q&As Dumps, **40%OFF Special Discount: Exam-Tests**)

NEW QUESTION: 167

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 ([LEAVE A REPLY](#))

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>

NEW QUESTION: 168

Taint the resource "aws_instance" "baz" resource that lives in module bar which lives in module foo.

- A. terraform taint module.foo.module.bar.baz
- B. terraform taint module.foo.bar.aws_instance.baz
- C. terraform taint module.foo.module.bar.aws_instance.baz
- D. terraform taint foo.bar.aws_instance.baz

Answer: C ([LEAVE A REPLY](#))

Check resource addressing

<https://www.terraform.io/docs/internals/resource-addressing.html>

NEW QUESTION: 169

Which of the following is considered a Terraform plugin?

- A. Terraform language
- B. Terraform tooling
- C. Terraform logic
- D. Terraform provider

Answer: D ([LEAVE A REPLY](#))

Explanation

Terraform is built on a plugin-based architecture. All providers and provisioners that are used in Terraform configurations are plugins, even the core types such as AWS and Heroku. Users of Terraform are able to write new plugins in order to support new functionality in Terraform.

<https://www.terraform.io/docs/plugins/basics.html>

NEW QUESTION: 170

Which of the following is an invalid variable name?

- A. count
- B. web
- C. var1
- D. instance_name

Answer: A ([LEAVE A REPLY](#))

<https://www.terraform.io/intro/examples/count.html>

NEW QUESTION: 171

Select the operating systems which are supported for a clustered Terraform Enterprise: (select four)

- A. Unix
- B. Red Hat
- C. CentOS
- D. Amazon Linux
- E. Ubuntu

Answer: B,C,D,E ([LEAVE A REPLY](#))

Explanation

<https://www.terraform.io/docs/enterprise/before-installing/index.html#operating-systemrequirements>

NEW QUESTION: 172

What command does Terraform require the first time you run it within a configuration directory?

- A. terraform import
- B. terraform init
- C. terraform plan
- D. terraform workspace

Answer: ([SHOW ANSWER](#))

terraform init command is used to initialize a working directory containing Terraform configuration files. Reference: <https://www.terraform.io/docs/cli/commands/init.html>

NEW QUESTION: 173

While Terraform is generally written using the HashiCorp Configuration Language (HCL), what other syntax can Terraform are expressed in?

- A. JSON
- B. YAML
- C. TypeScript
- D. XML

Answer: ([SHOW ANSWER](#))

The constructs in the Terraform language can also be expressed in JSON syntax, which is harder for humans to read and edit but easier to generate and parse programmatically.

NEW QUESTION: 174

Which of the following Terraform commands will automatically refresh the state unless supplied with additional flags or arguments? Choose TWO correct answers.

- A. terraform state
- B. terraform plan
- C. terraform output
- D. terraform validate
- E. terraform apply

Answer: B,E ([LEAVE A REPLY](#))

NEW QUESTION: 175

What is the command you can use to set an environment variable named "var1" of type String?

- A. export TF_VAR_VAR1
- B. set TF_VAR_var1
- C. variable "var1" { type = "string"}

D. export TF_VAR_var1

Answer: D ([LEAVE A REPLY](#))

Explanation

The environment variable must be in the format TF_VAR_name, so for the question TF_VAR_var1 is the correct choice.

https://www.terraform.io/docs/commands/environment-variables.html#tf_var_name

NEW QUESTION: 176

In the example below, the depends_on argument creates what type of dependency?

A. non-dependency resource

B. implicit dependency

C. explicit dependency

D. internal dependency

Answer: C ([LEAVE A REPLY](#))

NEW QUESTION: 177

Select the answer below that completes the following statement: Terraform Cloud can be managed from the CLI but requires _____?

A. an API token

B. a TOTP token

C. a username and password

D. authentication using MFA

Answer: A ([LEAVE A REPLY](#))

Explanation

API and CLI access are managed with API tokens, which can be generated in the Terraform Cloud UI. Each user can generate any number of personal API tokens, which allow access with their own identity and permissions. Organizations and teams can also generate tokens for automating tasks that aren't tied to an individual user.

NEW QUESTION: 178

A "backend" in Terraform determines how state is loaded and how an operation such as apply is executed. Which of the following is not a supported backend type?

A. Terraform enterprise

B. Consul

C. Github

D. S3

E. Artifactory

Answer: C ([LEAVE A REPLY](#))

Github is not a supported backend type.

<https://www.terraform.io/docs/backends/types/index.html>

NEW QUESTION: 179

You have already set `TF_LOG = DEBUG` to enable debug log. Now you want to always write the log to the directory you're currently running terraform from. what should you do to achieve this.

- A. Run the command `export TF_LOG_FILE=./terraform.log`.
- B. Run the command `export TF_LOG_PATH=./terraform.log`.
- C. Run the command `export TF_DEBUG_PATH=./terraform.log`.
- D. No explicit action required. Terraform will take care of this as you have enable `TF_LOG`.

Answer: (SHOW ANSWER)

<https://www.terraform.io/docs/commands/environment-variables.html>

NEW QUESTION: 180

Where in your Terraform configuration do you specify a state backend?

- A. The terraform block
- B. The resource block
- C. The provider block
- D. The datasource block

Answer: A (LEAVE A REPLY)

Backends are configured with a nested backend block within the top-level terraform block.

NEW QUESTION: 181

Which Terraform collection type should you use to store key/value pairs?

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

Answer: A (LEAVE A REPLY)

Valid TA-002-P Dumps shared by PrepPdf.com for Helping Passing TA-002-P Exam!

PrepPdf.com now offer the **newest TA-002-P exam dumps**, the PrepPdf.com TA-002-P exam **questions have been updated** and **answers have been corrected** get the **newest**

PrepPdf.com TA-002-P dumps with Test Engine here:

<https://www.preppdf.com/HashiCorp/TA-002-P-prepaway-exam-dumps.html> (94 Q&As Dumps,

40%OFF Special Discount: Exam-Tests)

NEW QUESTION: 182

Every region in AWS has a different AMI ID for Linux and these are keep on changing. What is the best approach to create the EC2 instances that can deal with different AMI IDs based on regions?

- A. Use data source `aws_ami`.

- B. Create a map of region to ami id.
- C. Create different configuration file for different region.
- D. None of the above

Answer: A ([LEAVE A REPLY](#))

<https://www.terraform.io/docs/configuration/data-sources.html>

NEW QUESTION: 183

Terraform has detailed logs which can be enabled by setting the _____ environmental variable.

- A. TF_TRACE
- B. TF_DEBUG
- C. TF_LOG
- D. TF_INFO

Answer: ([SHOW ANSWER](#)**)**

Explanation

Terraform has detailed logs that can be enabled by setting the TF_LOG environment variable to any value.

This will cause detailed logs to appear on stderr.

You can set TF_LOG to one of the log levels TRACE, DEBUG, INFO, WARN or ERROR to change the verbosity of the logs. TRACE is the most verbose and it is the default if TF_LOG is set to something other than a log level name.

<https://www.terraform.io/docs/internals/debugging.html>

NEW QUESTION: 184

You do not need to specify every required argument in the backend configuration. Omitting certain arguments may be desirable to avoid storing secrets, such as access keys, within the main configuration. When some or all of the arguments are omitted, we call this a _____.

- A. First Time Configuration
- B. Default Configuration
- C. Changing Configuration
- D. Partial Configuration
- E. Incomplete Configuration

Answer: D ([LEAVE A REPLY](#))

Explanation

You do not need to specify every required argument in the backend configuration. Omitting certain arguments may be desirable to avoid storing secrets, such as access keys, within the main configuration. When some or all of the arguments are omitted, we call this a partial configuration.

With a partial configuration, the remaining configuration arguments must be provided as part of the initialization process. There are several ways to supply the remaining arguments:

* Interactively: Terraform will interactively ask you for the required values, unless interactive input is disabled. Terraform will not prompt for optional values.

* File: A configuration file may be specified via the init command line. To specify a file, use the -backend-config=PATH option when running terraform init. If the file contains secrets it may be kept in a secure data store, such as Vault, in which case it must be downloaded to the local disk before running Terraform.

* Command-line key/value pairs: Key/value pairs can be specified via the init command line. Note that many shells retain command-line flags in a history file, so this isn't recommended for secrets. To specify a single key/value pair, use the -backend-config="KEY=VALUE" option when running terraform init.

<https://www.terraform.io/docs/backends/config.html#partial-configuration>

NEW QUESTION: 185

Workspaces in Terraform provides similar functionality in the open-source, Terraform Cloud, and Enterprise versions of Terraform.

A. True

B. False

Answer: B ([LEAVE A REPLY](#))

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

Workspaces, managed with the terraform workspace command, aren't the same thing as Terraform Cloud's workspaces. Terraform Cloud workspaces act more like completely separate working directories; CLI workspaces are just alternate state files.

NEW QUESTION: 186

terraform destroy is the only way to remove infrastructure.

A. True

B. False

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 187

What does this code do?

A. Requires any version of the AWS provider > = 3.0

B. Requires any version of the AWS provider after the 3.0 major release like 4.1

C. Requires any version of the AWS provider > = 3.0 and < 4.0

D. Requires any version of the AWS provider > 3.0

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 188

The following is a snippet from a Terraform configuration file:

Which, when validated, results in the following error:

Fill in the blank in the error message with the correct string from the list below.

A. alias

Explanation

<https://www.terraform.io/docs/configuration/providers.html#alias-multiple-providerinstances>

B. label

C. multi

D. version

Answer: A ([LEAVE A REPLY](#))

NEW QUESTION: 189

Select the most accurate statement to describe the Terraform language from the following list.

A. Terraform is an immutable, declarative, Infrastructure as Code provisioning language based on Hashicorp Configuration Language, or optionally JSON.

B. Terraform is a mutable, declarative, Infrastructure as Code configuration management language based on Hashicorp Configuration Language, or optionally JSON.

C. Terraform is an immutable, procedural, Infrastructure as Code configuration management language based on Hashicorp Configuration Language, or optionally JSON.

D. Terraform is a mutable, procedural, Infrastructure as Code provisioning language based on Hashicorp Configuration Language, or optionally YAML.

Answer: ([SHOW ANSWER](#)**)**

Terraform is not a configuration management tool -

<https://www.terraform.io/intro/vs/chefpuppet.html> Terraform is a declarative language -

<https://www.terraform.io/docs/configuration/index.html> Terraform supports a syntax that is JSON compatible - <https://www.terraform.io/docs/configuration/syntax-json.html> Terraform is primarily designed on immutable infrastructure principles - <https://www.hashicorp.com/resources/what-is-mutable-vs-immutable-infrastructure>

NEW QUESTION: 190

You have created an AWS EC2 instance of type t2.micro through your terraform configuration file ec2.tf . Now you want to change the instance type from t2.micro to t2.medium. Accordingly you have changed your configuration file and ran terraform plan. After running terraform plan you check the output and saw one instance will be updated from t2.micro --> t2.medium. After this you went to grab a coffee without running terraform apply and meanwhile a member of your team changed the instance type of that EC2 instance to t2.medium from aws console. After coming to your desk you run terraform apply. What will happen?

A. No resource will be updated and you will see the message : Apply Complete ! Resources : 0 added, 0 changed, 0 destroyed.

B. terraform apply will through an error.

C. 1 resource will be updated and you will see the message : Apply Complete ! Resources : 0 added, 1 changed, 0 destroyed.

D. The instance type will be changed to t2.micro and again will be changed to t2.medium

Answer: A ([LEAVE A REPLY](#))

NEW QUESTION: 191

Which of the below backends support state locking?

- A. consul
- B. azurearm
- C. artifactory
- D. S3

Answer: A,B,D ([LEAVE A REPLY](#))

NEW QUESTION: 192

Provider dependencies are created in several different ways. Select the valid provider dependencies from the following list: (select three)

- A. Existence of any resource instance belonging to a particular provider in the current state.
- B. Explicit use of a provider block in configuration, optionally including a version constraint.
- C. Use of any resource belonging to a particular provider in a resource or data block in configuration.
- D. Existence of any provider plugins found locally in the working directory.

Explanation

The existence of a provider plugin found locally in the working directory does not itself create a provider dependency. The plugin can exist without any reference to it in the terraform configuration. <https://www.terraform.io/docs/commands/providers.html>

Answer: A,B,C ([LEAVE A REPLY](#))

NEW QUESTION: 193

How would you reference the "name" value of the second instance of this fictitious resource?

- A. element(aws_instance.web, 2)
- B. aws_instance.web[1].name
- C. aws_instance.web[1]
- D. aws_instance.web[2].name
- E. aws_instance.web.*.name

Answer: A ([LEAVE A REPLY](#))

Explanation/Reference: <https://www.terraform.io/docs/configuration-0-11/interpolation.html>

NEW QUESTION: 194

Jim has created several AWS resources from a single terraform configuration file. Someone from his team has manually modified one of the EC2 instance.

Now to discard the manual change, Jim wants to destroy and recreate the EC2 instance. What is the best way to do it?

- A. terraform recreate
- B. terraform taint
- C. terraform destroy

D. terraform refresh

Answer: B ([LEAVE A REPLY](#))

Explanation

The terraform taint command manually marks a Terraform-managed resource as tainted, forcing it to be destroyed and recreated on the next apply.

This command will not modify infrastructure, but does modify the state file in order to mark a resource as tainted. Once a resource is marked as tainted, the next plan will show that the resource will be destroyed and recreated and the next apply will implement this change.

Forcing the recreation of a resource is useful when you want a certain side effect of recreation that is not visible in the attributes of a resource. For example: re-running provisioners will cause the node to be different or rebooting the machine from a base image will cause new startup scripts to run.

Note that tainting a resource for recreation may affect resources that depend on the newly tainted resource. For example, a DNS resource that uses the IP address of a server may need to be modified to reflect the potentially new IP address of a tainted server. The plan command will show this if this is the case.

This example will taint a single resource:

```
$ terraform taint aws_security_group.allow_all
```

The resource aws_security_group.allow_all in the module root has been marked as tainted.

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

NEW QUESTION: 195

Which of the below datatype is not supported by Terraform.

- A. List
- B. Array
- C. Map
- D. Object

Answer: ([SHOW ANSWER](#)**)**

NEW QUESTION: 196

What does terraform plan do ?

- A. Create an execution plan by evaluating the difference between configuration file and state file.
- B. Create an execution plan by evaluating the difference between configuration file and actual infrastructure.
- C. Checks whether the execution plan for a set of changes matches your expectations by making changes to real resources or to the state.
- D. Performs a refresh, unless explicitly disabled, and then apply the changes that are necessary to achieve the desired state specified in the configuration files.

Answer: A ([LEAVE A REPLY](#))

Valid TA-002-P Dumps shared by PrepPdf.com for Helping Passing TA-002-P Exam!
PrepPdf.com now offer the **newest TA-002-P exam dumps**, the PrepPdf.com TA-002-P exam **questions have been updated** and **answers have been corrected** get the **newest** PrepPdf.com TA-002-P dumps with Test Engine here:
<https://www.preppdf.com/HashiCorp/TA-002-P-prepaway-exam-dumps.html> (94 Q&As Dumps, **40%OFF Special Discount: Exam-Tests**)

NEW QUESTION: 197

If a module uses a local variable, you can expose that value with a terraform output.

- A. True
- B. False

Answer: A ([LEAVE A REPLY](#))

Output values are like function return values.

Reference: <https://www.terraform.io/docs/language/values/locals.html>

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

NEW QUESTION: 198

What are some of the problems of how infrastructure was traditionally managed before Infrastructure as Code?

(select three)

- A. Requests for infrastructure or hardware required a ticket, increasing the time required to deploy applications
- B. Traditional deployment methods are not able to meet the demands of the modern business where resources tend to live days to weeks, rather than months to years
- C. Traditionally managed infrastructure can't keep up with cyclic or elastic applications
- D. Pointing and clicking in a management console is a scalable approach and reduces human error as businesses are moving to a multi-cloud deployment model

Answer: A,B,C ([LEAVE A REPLY](#))

Explanation

Businesses are making a transition where traditionally-managed infrastructure can no longer meet the demands of today's businesses. IT organizations are quickly adopting the public cloud, which is predominantly API-driven. To meet customer demands and save costs, application teams are architecting their applications to support a much higher level of elasticity, supporting technology like containers and public cloud resources.

These resources may only live for a matter of hours; therefore the traditional method of raising a ticket to request resources is no longer a viable option Pointing and clicking in a management console is NOT scale and increases the change of human error.

NEW QUESTION: 199

Which of the following actions are performed during a terraform init?

- A. Initializes downloaded and/or installed providers
- B. Initializes the backend configuration
- C. Provisions the declared resources in your configuration
- D. Download the declared providers which are supported by HashiCorp

Answer: (SHOW ANSWER)

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.

This command is always safe to run multiple times, to bring the working directory up to date with changes in the configuration. Though subsequent runs may give errors, this command will never delete your existing configuration or state.

terraform init command does -

- * Copy a Source Module
- * Backend Initialization
- * Child Module Installation
- * Plugin Installation

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

NEW QUESTION: 200

Which of the following statements best describes the Terraform list(...) type?

- A. a collection of values where each is identified by a string label.
- B. a collection of unique values that do not have any secondary identifiers or ordering.
- C. a sequence of values identified by consecutive whole numbers starting with zero.
- D. a collection of named attributes that each have their own type.

Explanation

A terraform list is a sequence of values identified by consecutive whole numbers starting with zero. <https://www.terraform.io/docs/configuration/types.html#structural-types>

Answer: C (LEAVE A REPLY)

NEW QUESTION: 201

When should you use the force-unlock command?

- A. You see a status message that you cannot acquire the lock
- B. You have a high priority change
- C. Automatic unlocking failed
- D. Your apply failed due to a state lock

Answer: C (LEAVE A REPLY)

Manually unlock the state for the defined configuration.

NEW QUESTION: 202

From the answers below, select the advantages of using Infrastructure as Code.

- A. Provide a codified workflow to develop customer-facing applications.
- B. Safely test modifications using a "dry run" before applying any actual changes.
- C. Easily integrate with application workflows (GitLab Actions, Azure DevOps, CI/CD tools).
- D. Easily change and update existing infrastructure.
- E. Provide reusable modules for easy sharing and collaboration.

Answer: B,C,D,E ([LEAVE A REPLY](#))

Explanation

Infrastructure as Code is not used to develop applications, but it can be used to help deploy or provision those applications to a public cloud provider or on-premises infrastructure.

All of the others are benefits to using Infrastructure as Code over the traditional way of managing infrastructure, regardless if it's public cloud or on-premises.

NEW QUESTION: 203

Only the user that generated a plan may apply it.

- A. True
- B. False

Answer: A ([LEAVE A REPLY](#))

The optional -out argument can be used to save the generated plan to a file for later execution with terraform apply, which can be useful when running Terraform in automation.

NEW QUESTION: 204

From the answers below, select the advantages of using Infrastructure as Code.

- A. Provide a codified workflow to develop customer-facing applications.
- B. Safely test modifications using a "dry run" before applying any actual changes.
- C. Easily integrate with application workflows (GitLab Actions, Azure DevOps, CI/CD tools).
- D. Easily change and update existing infrastructure.
- E. Provide reusable modules for easy sharing and collaboration.

Answer: ([SHOW ANSWER](#))

Infrastructure as Code is not used to develop applications, but it can be used to help deploy or provision those applications to a public cloud provider or on-premises infrastructure.

All of the others are benefits to using Infrastructure as Code over the traditional way of managing infrastructure, regardless if it's public cloud or on-premises.

NEW QUESTION: 205

A variable az has the following default value. What will be the datatype of the variable?

az=["us-west-1a","us-east-1a"]

- A. Map
- B. String
- C. Object
- D. List

Answer: D ([LEAVE A REPLY](#))

NEW QUESTION: 206

Which one of the following will run echo 0 and echo 1 on a newly created host?

A. provisioner "local-exec" { command = "echo 0"

command = "echo 1"

}

B. provisioner "remote-exec" {

inline = [

echo 0,

echo 1

]

}

C. provisioner "remote-exec" {

command = "\${echo 0}"

command = "\${echo 1}"

}

D. provisioner "remote-exec" {

inline = [

"echo 0",

"echo 1"

]

}

Answer: D ([LEAVE A REPLY](#))

Explanation

remote-exec Provisioner

Example usage

```
resource "aws_instance" "web" {
```

```
# ...
```

```
provisioner "remote-exec" {
```

```
inline = [
```

```
"puppet apply",
```

```
"consul join ${aws_instance.web.private_ip}",
```

```
]
```

```
}
```

```
}
```

NEW QUESTION: 207

Which of the below terraform commands do not run terraform refresh implicitly before taking actual action of the command?

A. terraform apply

- B. terraform destroy
- C. terraform init
- D. terraform import
- E. terraform plan

Answer: C,D ([LEAVE A REPLY](#))

Explanation

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

NEW QUESTION: 208

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 ([LEAVE A REPLY](#))

Explanation

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

NEW QUESTION: 209

What kind of resource dependency is stored in terraform.tfstate file?

- A. Both implicit and explicit dependencies are stored in state file.
- B. Only explicit dependencies are stored in state file.
- C. Only implicit dependencies are stored in state file.
- D. No dependency information is stored in state file.

Answer: ([SHOW ANSWER](#)**)**

Explanation

Terraform state captures all dependency information, both implicit and explicit. One purpose for state is to determine the proper order to destroy resources. When resources are created all of their dependency information is stored in the state. If you destroy a resource with dependencies, Terraform can still determine the correct destroy order for all other resources because the dependencies are stored in the state.

<https://www.terraform.io/docs/state/purpose.html#metadata>

NEW QUESTION: 210

Which of the following statements best describes the Terraform list(...) type?

- A. a collection of values where each is identified by a string label.
- B. a sequence of values identified by consecutive whole numbers starting with zero.
- C. a collection of unique values that do not have any secondary identifiers or ordering.

D. a collection of named attributes that each have their own type.

Answer: B ([LEAVE A REPLY](#))

Explanation

A terraform list is a sequence of values identified by consecutive whole numbers starting with zero. <https://www.terraform.io/docs/configuration/types.html#structural-types>

NEW QUESTION: 211

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

- A. WinRM
- B. UDP
- C. SMB
- D. RDP
- E. ssh

Answer: ([SHOW ANSWER](#)**)**

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.

remote-exec connection types -

- * ssh on Linux
- * winrm on Windows

<https://www.terraform.io/docs/provisioners/remote-exec.html>

Valid TA-002-P Dumps shared by PrepPdf.com for Helping Passing TA-002-P Exam!

PrepPdf.com now offer the **newest TA-002-P exam dumps**, the PrepPdf.com TA-002-P exam **questions have been updated** and **answers have been corrected** get the **newest** PrepPdf.com TA-002-P dumps with Test Engine here:

<https://www.preppdf.com/HashiCorp/TA-002-P-prepaway-exam-dumps.html> (94 Q&As Dumps,

40%OFF Special Discount: Exam-Tests)

NEW QUESTION: 212

What is the default backend for Terraform?

- A. consul
- B. gcs
- C. local
- D. etcd

Answer: ([SHOW ANSWER](#)**)**

By default, Terraform uses the "local" backend, which is the normal behavior of Terraform you're used to.

<https://www.terraform.io/docs/backends/index.html>

NEW QUESTION: 213

Eric needs to make use of module within his terraform code. Should the module always be public and open-source to be able to be used?

A. False

B. True

Answer: A ([LEAVE A REPLY](#))

Explanation

Terraform module need not be public and open-source. Module can be placed in -

- * Local paths
- * Terraform Registry
- * GitHub
- * Bitbucket
- * Generic Git, Mercurial repositories
- * HTTP URLs
- * S3 buckets
- * GCS buckets

<https://www.terraform.io/docs/modules/sources.html>

Valid TA-002-P Dumps shared by PrepPdf.com for Helping Passing TA-002-P Exam!

PrepPdf.com now offer the **newest TA-002-P exam dumps**, the PrepPdf.com TA-002-P exam **questions have been updated** and **answers have been corrected** get the **newest** PrepPdf.com TA-002-P dumps with Test Engine here:

<https://www.preppdf.com/HashiCorp/TA-002-P-prepaway-exam-dumps.html> (94 Q&As Dumps,

40%OFF Special Discount: Exam-Tests)