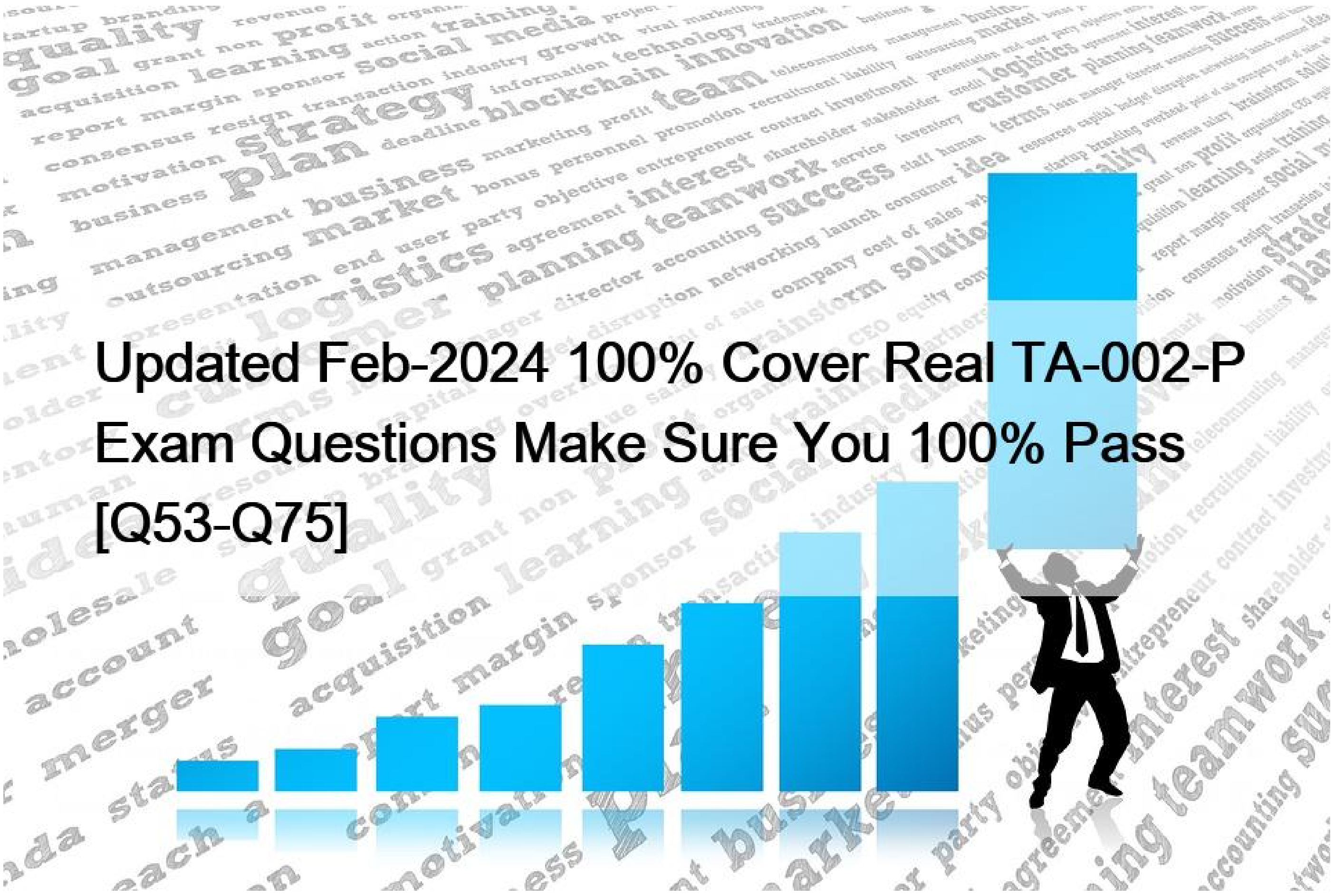
Updated Feb-2024 100% Cover Real TA-002-P Exam Questions Make Sure You 100% Pass [Q53-Q75



Updated Feb-2024 100% Cover Real TA-002-P Exam Questions Make Sure You 100% Pass TA-002-P dumps Accurate Questions and Answers with Free and Fast Updates

Earning the HashiCorp Certified: Terraform Associate certification can open up new career opportunities for IT professionals, as it is recognized by top employers in the industry. It also validates a candidate's skills and knowledge in using Terraform to automate infrastructure, making them a valuable asset to any organization looking to improve their infrastructure management processes.

The Terraform Associate certification exam is created by HashiCorp, a leading provider of infrastructure automation software. TA-002-P exam is targeted towards professionals who are involved in cloud infrastructure management, DevOps, and automation. TA-002-P exam is designed to test the candidate's ability to use Terraform to manage infrastructure as code, and it covers topics such as Terraform basics, resource management, configuration, and state management.

NEW QUESTION 53

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.
* False
* True
Explanation

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

NEW QUESTION 54

Which of the following actions are performed during a terraform init?
* Initializes downloaded and/or installed providers
* Initializes the backend configuration
* Provisions the declared resources in your configuration
* Download the declared providers which are supported by HashiCorp
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 &#8211;

* Copy a Source Module

* Backend Initialization

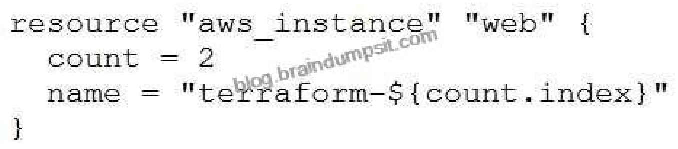
* Child Module Installation

* Plugin Installation

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

NEW QUESTION 55

How would you reference the &#8220;name&#8221; value of the second instance of this fictitious resource?


```
resource "aws_instance" "web" {
  count = 2
  name = "terraform-${count.index}"
}
```

* element(aws_instance.web, 2)
* aws_instance.web[1].name
* aws_instance.web[1]
* aws_instance.web[2].name
* aws_instance.web.*.name

Explanation/Reference: https://www.terraform.io/docs/configuration-0-11/interpolation.html

NEW QUESTION 56

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

* True
* False

Reference: https://blog.gruntwork.io/a-comprehensive-guide-to-managing-secrets-in-your-terraform-code-

1d586955ace1

NEW QUESTION 57

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

* provider
* terraform
* resource
* data

Explanation

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 = &#8220;> 0.12.0&#8221;

}

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

NEW QUESTION 58

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

* The terraform block
* The resource block
* The provider block
* The datasource block

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

NEW QUESTION 59

When does Sentinel enforce policy logic during a Terraform Enterprise run?
* Before the plan phase
* During the plan phase
* Before the a apply phase
* After the apply phase

NEW QUESTION 60

You have created 2 workspaces PROD and RQA. You have switched to RQA and provisioned RQA

infrastructure from this workspace. Where is your state file stored?
* terraform.tfstate.d
* terraform.d
* terraform.tfstate.RQA
* terraform.tfstate

NEW QUESTION 61

Which of these ate features of Terraform Cloud? Choose two correct answers.
* Automated infrastructure deployment visualization
* Automatic backups
* A web-based user interface (Ul)
* Remote state storage
These are features of Terraform Cloud, which is a hosted service that provides a web-based UI, remote state storage, remote operations, collaboration features, and more for managing your Terraform infrastructure.

NEW QUESTION 62

Terraform validate reports syntax check errors from which of the following scenarios?
* Code contains tabs indentation instead of spaces
* There is missing value for a variable
* The state files does not match the current infrastructure
* None of the above
Explanation

The terraform validate command is used to validate the syntax of the terraform files. Terraform performs a

syntax check on all the terraform files in the directory, and will display an error if any of the files doesn&#8217;t

validate. This command does not check formatting (e.g. tabs vs spaces, newlines, comments etc.). The

following can be reported: invalid HCL syntax (e.g. missing trailing quote or equal sign) invalid HCL

references (e.g. variable name or attribute which doesn&#8217;t exist) same provider declared multiple times same

module declared multiple times same resource declared multiple times invalid module name interpolation used

in places where it&#8217;s unsupported (e.g. variable, depends_on, module.source, provider) missing value for a variable (none of -var foo=&#8230; flag, -var-file=foo.vars flag, TF_VAR_foo environment variable,

terraform.tfvars, or default value in the configuration)

https://www.typeerror.org/docs/terraform/commands/validate

https://learning-ocean.com/tutorials/terraform/terraform-validate

NEW QUESTION 63

What is the standard workflow that a developer follows while working with terraform open source version?
* Run terraform refresh to update the terraform state , then write the terraform code , and finally run terraform apply.
* Run terraform destroy first since you need to start from fresh every time , before running terraform apply.
* Write terraform code , and run terraform push , to update the terraform state to the remote repo , which in turn will take care of the next steps.
* Write the terraform code on the developer machine , run terraform plan to check the changes , and run terraform apply to provision the infra.
Explanation

You do not need to run terraform refresh as terraform plan implicitly will run terraform refresh.

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

NEW QUESTION 64

What is the workflow for deploying new infrastructure with Terraform?
* terraform plan to import the current infrastructure to the state file, make code changes, and terraform

apply to update the infrastructure
* Write a Terraform configuration, run terraform show to view proposed changes, and terraform apply to

create new infrastructure.
* terraform plan to import the current infrastructure to the state file, make code changes, and terraform

apply to update the infrastructure
* Write a Terraform configuration, run terraform init, run terraform plan to view planned infrastructure

changes, and terraform apply to create new infrastructure.
Reference:

https://www.google.com/search?q=Write+a+Terraform+configuration%2C+run+terraform+init%2C

+run+terraform+plan+to+view+planned+infrastructure+changes%2C+and+terraform+apply+to+create+new

+infrastructure.&oq=Write+a+Terraform+configuration%2C+run+terraform+init%2C+run+terraform+plan+to

+view+planned+infrastructure+changes%2C+and+terraform+apply+to+create+new

+infrastructure.&aqs=chrome..69i57.556j0j7&sourceid=chrome&ie=UTF-8

NEW QUESTION 65

Terraform Cloud is available only as a paid offering from HashiCorp.
* True
* False
Explanation

Many of Terraform Cloud features are free for small teams, including remote state storage, remote runs, and

VCS connections.

&#8220;Terraform Cloud is a commercial SaaS product developed by HashiCorp. Many of its features are free for

small teams, including remote state storage, remote runs, and VCS connections. We also offer paid plans for

larger teams that include additional collaboration and governance features.&#8221;

NEW QUESTION 66

Multiple providers can be declared within a single Terraform configuration file.
* True
* False
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 &#8220;aws&#8221; {

region = &#8220;us-east-1&#8221;

}

# Additional provider configuration for west coast region

provider &#8220;aws&#8221; {

alias = &#8220;west&#8221;

region = &#8220;us-west-2&#8221;

}

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.

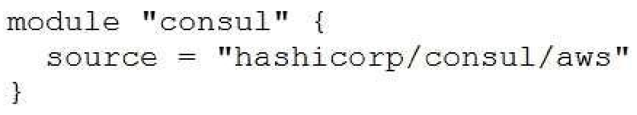
https://www.terraform.io/docs/configuration/providers.html

NEW QUESTION 67

Which one of the following command will rewrite Terraform configuration files to a canonical format and style.
* terraform graph -h
* terraform init
* terraform graph
* terraform fmt
The terraform fmt command is used to rewrite Terraform configuration files to a canonical format and style. This command applies a subset of the Terrra Terraform language style conventions, along with other minor adjustments for readability.

NEW QUESTION 68

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



```
module "consul" {
  source = "hashicorp/consul/aws"
}
```

How do you specify version 1.0.0?
* Modules stored on the public Terraform Module Registry do not support versioning
* Append ?ref=v1.0.0 argument to the source path
* Add version = &#8220;1.0.0&#8221; attribute to module block
* Nothing &#8211; modules stored on the public Terraform Module Registry always default to version 1.0.0
Explanation/Reference: https://www.terraform.io/docs/language/modules/sources.html

NEW QUESTION 69

After creating a new workspace &#8220;PROD&#8221; you need to run the command terraform select PROD to switch to it.
* False
* True
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 &#8220;example&#8221;!

You&#8217;re now on a new, empty workspace. Workspaces isolate their state, so if you run &#8220;terraform plan&#8221; Terraform will not see any existing state for this configuration.

NEW QUESTION 70

Terraform has detailed logs which can be enabled by setting the _________ environmental variable.
* TF_TRACE
* TF_DEBUG
* TF_LOG
* TF_INFO
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 71

In terraform, most resource dependencies are handled automatically. Which of the following statements

describes best how terraform resource dependencies are handled?
* Resource dependencies are identified and maintained in a file called resource.dependencies. Each

terraform provider is required to maintain a list of all resource dependencies for the provider and it&#8217;s

included with the plugin during initialization when terraform init is executed. The file is located in the

terraform.d folder.
* The terraform binary contains a built-in reference map of all defined Terraform resource dependencies.

Updates to this dependency map are reflected in terraform versions. To ensure you are working with the

latest resource dependency map you much be running the latest version of Terraform.
* Resource dependencies are handled automatically by the depends_on meta_argument, which is set to

true by default.
* Terraform analyses any expressions within a resource block to find references to other objects, and

treats those references as implicit ordering requirements when creating, updating, or destroying

resources.
Explanation

https://www.terraform.io/docs/configuration/resources.html

NEW QUESTION 72

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 and ran terraform plan.

After running terraform plan you check the output and saw one instance will be updated from t2.micro &#8211;> 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?
* No resource will be updated and you will see the message : Apply Complete ! Resources : 0 added, 0 changed, 0 destroyed.
* The instance type will be changed to t2.micro and again will be changed to t2.medium
* terraform apply will through an error.
* 1 resource will be updated and you will see the message : Apply Complete ! Resources : 0 added, 1 changed, 0 destroyed.

NEW QUESTION 73

Terraform must track metadata such as resource dependencies. Where is this data stored?
* workspace
* backend
* state file
* metadata store
Explanation

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

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

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

https://www.terraform.io/docs/state/purpose.html#metadata

NEW QUESTION 74

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.
* False
* True
https://www.terraform.io/docs/commands/refresh.html

NEW QUESTION 75

Which of the following statements about local modules is incorrect:
* Local modules are not cached by terraform init command
* Local modules are sourced from a directory on disk
* Local modules support versions
* All of the above (all statements above are incorrect
* None of the above (all statements above are correct)

Understanding functional and technical aspects of HashiCorp Certified: Terraform Associate TA-002-P Professional Exam Object Management

The following will be discussed in **HASHICORP TA-002 exam dumps**:

- Design and distinguish resource and data configuration- Learn the application of the collection and structural types- Exhibit use of variables and outputs- Use Terraform built-in functions to write configuration- Configure resource using a dynamic block **Real TA-002-P Quesions Pass Certification Exams Easily:**

https://www.braindumpsit.com/TA-002-P_real-exam.html]