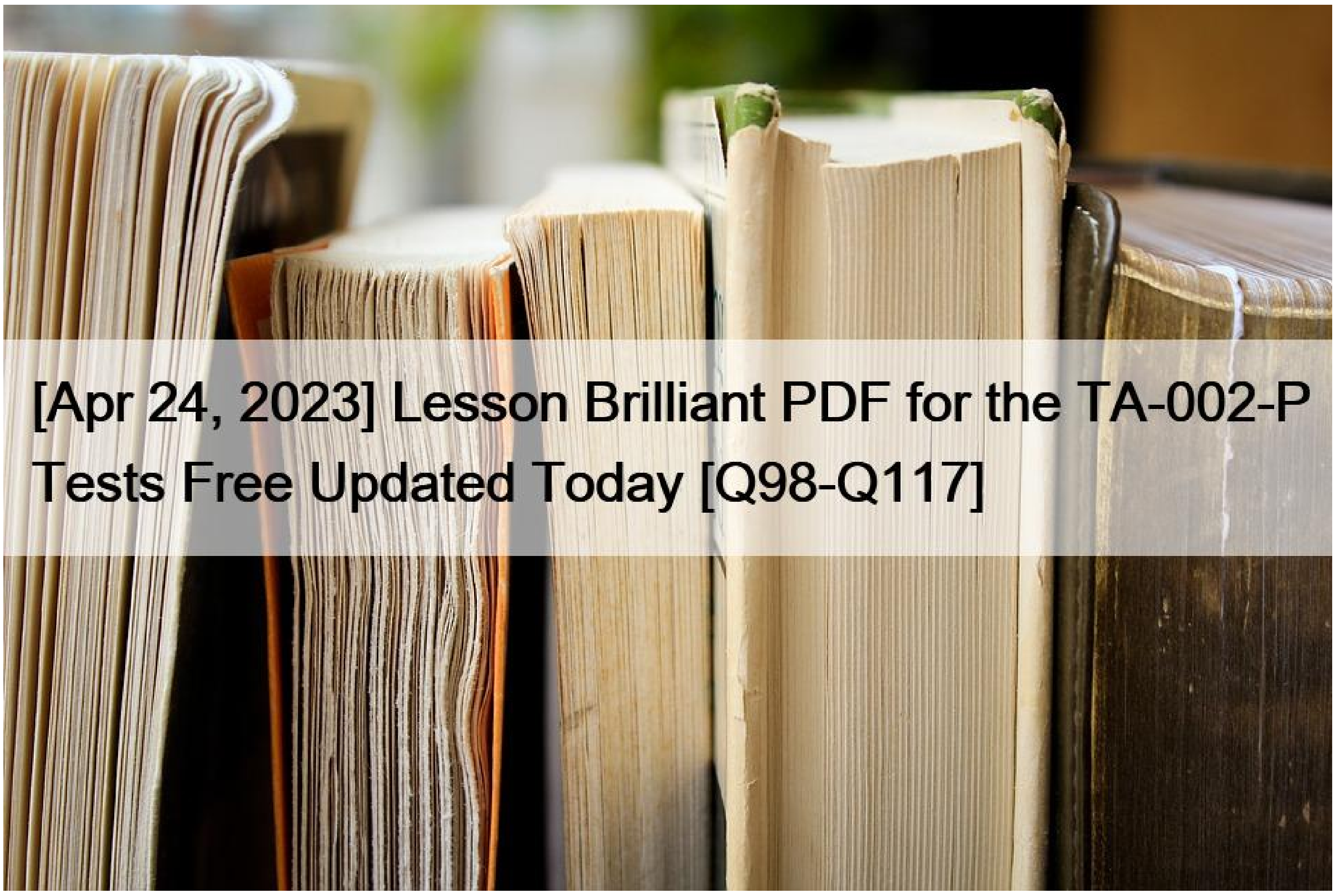
[Apr 24, 2023 Lesson Brilliant PDF for the TA-002-P Tests Free Updated Today [Q98-Q117



[Apr 24, 2023] Lesson Brilliant PDF for the TA-002-P Tests Free Updated Today
Get New 2023 Valid Practice HashiCorp Infrastructure Automation TA-002-P Q&A - Testing Engine

Exam Topics for HashiCorp Certified: Terraform Associate TA-002-P Professional Exam

The following will be practiced in **HASHICORP TA-002 practice exam** and **HASHICORP TA-002 practice exams**:

- Terraform's purpose (vs other IaC)- Terraform Cloud capabilities- Terraform workflow- Terraform basics

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**:

- Specify remote state storage mechanisms and supported regular backends.- Interact with module inputs and outputs- Learn concealed management in state files- Destroy Terraform controlled infrastructure- Verify a Terraform arrangement- Define variable scope inside modules- Contrast module source alternatives- Explore modules from the public Terraform Module Registry- Outline state locking- Control backend authentication processes

The benefit in Obtaining the HashiCorp Certified: Terraform Associate TA-002-P Professional Exam

This certification is an industry-recognized credential from HashiCorp that assesses the applicant's understanding of fundamental concepts and skills on Terraform OSS and the characteristics that exist on Terraform Cloud & Terraform Enterprise packages. When it comes to employment, this certification is a career game-changer that will advance you closer to achieving your dream profession. Some more benefits are:

- Generate new leads and gain new projects- Better avenues for improving professional expertise- Better Salary- Achieve recognition for your hard work- Planning for a better future- Opportunities to grow your professional network **Q98.** 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

Q99. 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

Explanation

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

Q100. Which of the following is not a valid Terraform string function?

* replace
* format
* join
* tostring

Explanation

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

Q101. You have already set TF_LOG = DEBUG to enable debug log. Now you want to always write the log to the directory you&#8217;re currently running terraform from. what should you do to achieve this.

* Run the command export TF_LOG_FILE=./terraform.log.
* Run the command export TF_LOG_PATH=./terraform.log.
* Run the command export TF_DEBUG_PATH=./terraform.log.
* No explicit action required. Terraform will take care of this as you have enable TF_LOG.

Explanation

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

Q102. True or False? Each Terraform workspace uses its own state file to manage the infrastructure associated with that particular

workspace.
* False
* True

Explanation

The persistent data stored in the backend belongs to a workspace. Initially, the backend has only one workspace, called &#8220;default&#8221;, and thus there is only one Terraform state associated with that configuration.

Q103. Complete the following sentence:

For local state, the workspaces are stored directly in a ___________.
* a file called terraform.tfstate.backup
* directory called terraform.workspaces.tfstate
* a file called terraform.tfstate
* directory called terraform.tfstate.d

Explanation

For local state, Terraform stores the workspace states in a directory called terraform.tfstate.d.

https://www.terraform.io/docs/state/workspaces.html#workspace-internals

Q104. 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.
* Export the value using terraform export and input the value using terraform input.
* Configure the pertinent provider&#8217;s configuration with a list of possible IP addresses to use.
* Configure an output value in the application module in order to use that value for the DNS module.
* Preconfigure the IP address as a parameter in the DNS module.

Explanation

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

* A child module can use outputs to expose a subset of its resource attributes to a parent module.

* A root module can use outputs to print certain values in the CLI output after running terraform apply.

* When using remote state, root module outputs can be accessed by other configurations via a

terraform_remote_state data source.

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

Q105. You have declared a variable called var.list which is a list of objects that all have an attribute id.

Which options will produce a list of the IDs? (Choose two.)

* { for o in var.list : o => o.id }
* var.list[*].id
* [var.list[*].id]
* [for o in var.list : o.id]

Explanation

https://www.terraform.io/language/expressions/splat

A splat expression provides a more concise way to express a common operation that could otherwise be

performed with a for expression.

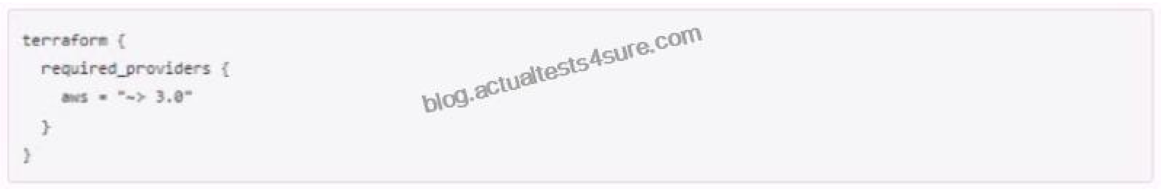
Q106. Which of the following Terraform commands will automatically refresh the state unless supplied with additional flags or arguments? Choose TWO correct answers.
* terraform state
* terraform apply
* terraform plan
* terraform validate
* terraform output

Q107. Which of the following does terraform apply change after you approve the execution plan? Choose two correct

answers.
* The execution plan
* Terraform code
* Cloud infrastructure
* State file
* The .terraform directory

Q108. Which of the below backends support state locking?
* S3
* consul
* azurerm
* artifactory

Q109. What does this code do?



```
terraform {
  required_providers {
    aws = "~> 3.0"
  }
}
```

* Requires any version of the AWS provider > = 3.0 and < 4.0
* Requires any version of the AWS provider > = 3.0
* Requires any version of the AWS provider after the 3.0 major release like 4.1
* Requires any version of the AWS provider > 3.0

Explanation

https://www.terraform.io/language/expressions/version-constraints#-3

Allows only the rightmost version component to increment. For example, to allow new patch releases within a

specific minor release, use the full version number: ~> 1.0.4 will allow installation of 1.0.5 and 1.0.10 but not

1.1.0

Q110. resource &#8220;aws_s3_bucket&#8221; &#8220;example&#8221; { bucket = &#8220;my-test-s3-terraform-bucket&#8221; &#8230;} resource &#8220;aws_iam_role&#8221;

&#8220;test_role&#8221; { name = &#8220;test_role&#8221; &#8230;}

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?
* This will already be taken care of by terraform native implicit dependency. Nothing else needs to be done from your end.
* Add explicit dependency using depends_on . This will ensure the correct order of resource creation.
* 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.
* 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.
Explanation

Use the depends_on meta-argument to handle hidden resource dependencies that Terraform can&#8217;t automatically infer.

Explicitly specifying a dependency is only necessary when a resource relies on some other resource&#8217;s behavior but doesn&#8217;t access any of that resource&#8217;s data in its arguments.

Q111. John is writing a module and within the module, there are multiple places where he has to use the same conditional expression but he wants to avoid repeating the same values or expressions multiple times in a configuration,. What is a better approach to dealing with this?
* Local Values
* Expressions
* Functions
* Variables
Explanation

A local value assigns a name to an expression, allowing it to be used multiple times within a module without repeating it.

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

Q112. Which of the following best describes the default local backend?
* The local backend is where Terraform Enterprise stores logs to be processed by an log collector.
* The local backend stores state on the local filesystem, locks the state using system APIs, and performs

operations locally.
* The local backend is the directory where resources deployed by Terraform have direct access to in order

to update their current state.
* The local backend is how Terraform connects to public cloud services, such as AWS, Azure, or GCP.

Explanation

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

operations locally.

terraform {

backend &#8220;local&#8221; {

path = &#8220;relative/path/to/terraform.tfstate&#8221;

}

}

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

Q113. What does terraform refresh command do?
* terraform refresh can be used to selectively update sections of the state file, using terraform resource

level addressing.
* terraform refresh command basically updates the configuration file with the current state of the actual

infrastructure
* terraform refresh is use to change/modify the infrastructure based on the existing state file, at that

moment.
* terraform refresh can be used to selectively update sections of the state file, using terraform resource

level addressing.
* terraform refresh syncs the state file with the real world infrastructure.

Q114. After creating a new workspace &#8220;PROD&#8221; you need to run the command terraform select PROD to switch to it.
* False
* True

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 &#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.

Q115. What are some of the features of Terraform state? (select three)
* inspection of cloud resources
* determining the correct order to destroy resources
* mapping configuration to real-world resources
* increased performance

Q116. 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?
* Remote State
* Module
* Use the cached state and treat this as the record of truth.
* Workspace

Explanation

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

Q117. When TF_LOG_PATH is set, TF_LOG must be set in order for any logging to be enabled.
* False
* True

Explanation

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

For example, to always write the log to the directory you&#8217;re currently running terraform from:

```
export TF_LOG_PATH=./terraform.log

export TF_LOG=TRACE
```