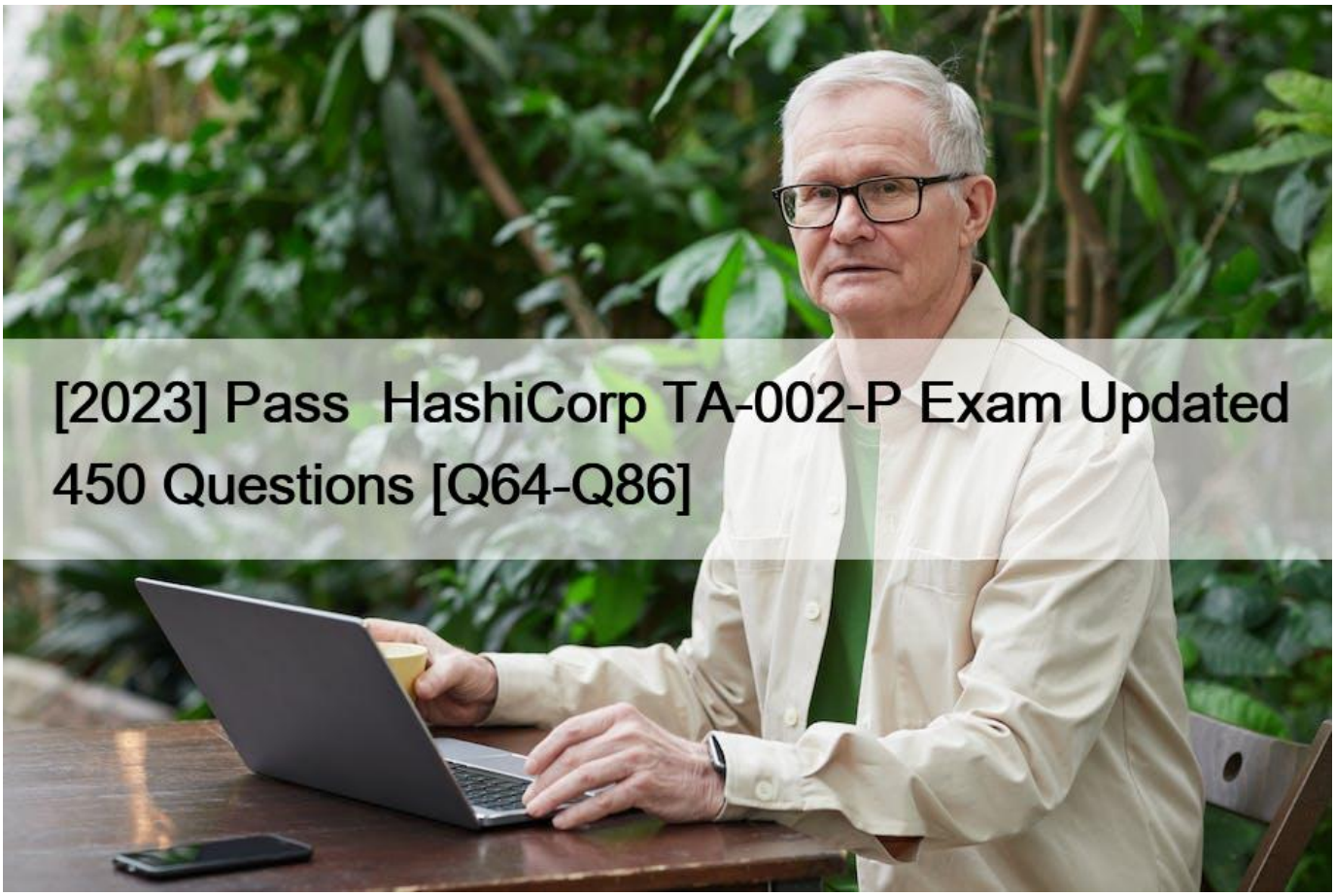
[2023 Pass HashiCorp TA-002-P Exam Updated 450 Questions [Q64-Q86



[2023] Pass HashiCorp TA-002-P Exam Updated 450 Questions
Get 2023 Updated Free HashiCorp TA-002-P Exam Questions and Answer

HashiCorp Certified: Terraform Associate (TA-002-P) exam is designed to test your knowledge of Terraform's core concepts, best practices, and configuration syntax. TA-002-P exam is intended for individuals who are new to Terraform or have some experience with the tool and want to demonstrate their proficiency. TA-002-P exam is not intended for experts with years of experience using Terraform, but for those who are just starting out or have limited experience with the tool.

HashiCorp TA-002-P certification exam covers a range of topics related to Terraform, including understanding the core concepts of Terraform, using Terraform to manage infrastructure resources, understanding Terraform modules, working with Terraform providers, and managing Terraform state. TA-002-P exam is designed for individuals who have experience working with Terraform and want to demonstrate their knowledge and skills to potential employers. HashiCorp Certified: Terraform Associate certification exam is a valuable credential for professionals working in DevOps, cloud infrastructure management, and other related fields.

NO.64 Which of the following state management command allow you to retrieve a list of resources that are part of the state file?

* terraform state list
* terraform state view
* terraform view
* terraform list

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

NO.65 Select all Operating Systems that Terraform is available for. (select five)

* Linux
* macOS
* Unix
* Solaris
* Windows
* FreeBSD

Terraform is available for macOS, FreeBSD, OpenBSD, Linux, Solaris, Windows https://www.terraform.io/downloads.html

NO.66 When should you use the force-unlock command?

* You see a status message that you cannot acquire the lock
* You have a high priority change
* Automatic unlocking failed
* Your apply failed due to a state lock

Explanation

Be very careful with this command. 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. Source: https://www.terraform.io/language/state/locking

https://www.terraform.io/cli/commands/force-unlock

NO.67 terraform init retrieves the source code tot all referenced modules

* True
* False

NO.68 A data block requests that Terraform read from a given data source and export the result under the given local name.

* False
* True

NO.69 What value does the Terraform Cloud/Terraform Enterprise private module registry provide over the public

Terraform Module Registry?

* The ability to share modules with public Terraform users and members of Terraform Enterprise

Organizations
* The ability to tag modules by version or release
* The ability to restrict modules to members of Terraform Cloud or Enterprise organizations
* The ability to share modules publicly with any user of Terraform

Explanation

Terraform Cloud’s private registry works similarly to the public Terraform Registry and helps you share

Terraform providers and Terraform modules across your organization. It includes support for versioning and a

searchable list of available providers and modules.

NO.70 Which of the following is allowed as a Terraform variable name?
* count
* name
* source
* version

NO.71 Using multi-cloud and provider-agnostic tools provides which of the following benefits?
* Operations teams only need to learn and manage a single tool to manage infrastructure, regardless of

where the infrastructure is deployed.
* Increased risk due to all infrastructure relying on a single tool for management.
* Can be used across major cloud providers and VM hypervisors.
* Slower provisioning speed allows the operations team to catch mistakes before they are applied.

Explanation

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.

NO.72 Terraform providers are always installed from the Internet.
* True
* False

Explanation

Terraform configurations must declare which providers they require, so that Terraform can install and use

them.

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

NO.73 Why might a user opt to include the following snippet in their configuration file?
* Terraform 0.12 introduced substantial changes to the syntax used to write Terraform configuration
* The user wants to ensure that the application being deployed is a minimum version of 0.12
* this ensures that all Terraform providers are above a certain version to match the application being

deployed
* versions before Terraform 0.12 were not approved by HashiCorp to be used in production

NO.74 The terraform.tfstate file always matches your currently built infrastructure.
* True
* False
Reference: https://www.terraform.io/docs/language/state/index.html

NO.75 Which of the following statements best describes the Terraform list(…) type?
* a collection of values where each is identified by a string label.
* a sequence of values identified by consecutive whole numbers starting with zero.
* a collection of unique values that do not have any secondary identifiers or ordering.
* 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

NO.76 Which of the following Terraform files should be ignored by Git when committing code to a repo? (select Three)
* Files named exactly terraform.tfvars or terraform.tfvars.json.
* Any files with names ending in .auto.tfvars or .auto.tfvars.json.
* input.tf
* terraform.tfstate
* output.tf
Explanation

The .gitignore file should be configured to ignore Terraform files that either contain sensitive data or are not required to save.

Terraform state (terraform.tfstate) 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.

The terraform.tfvars file may contain sensitive data, such as passwords or IP addresses of an environment that you may not want to share with others.

NO.77 Given the Terraform configuration below, in which order will the resources be created?
* Larger image
* resources will be created simultaneously
* aws_eip will be created first aws_instance will be created second
* aws_instance will be created first aws_eip will be created second
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

NO.78 When running the command terraform taint against a managed resource you want to force recreation upon, Terraform will immediately destroy and recreate the resource.
* True
* False

NO.79 You have been given requirements to create a security group for a new application. Since your organization

standardizes on Terraform, you want to add this new security group with the fewest number of lines of code.

What feature could you use to iterate over a list of required tcp ports to add to the new security group?
* dynamic backend
* splat expression
* terraform import
* dynamic block

Explanation

A dynamic block acts much like a for expression, but produces nested blocks instead of a complex typed

value. It iterates over a given complex value and generates a nested block for each element of that complex

value.

https://www.terraform.io/docs/configuration/expressions.html#dynamic-blocks

NO.80 Once a resource is marked as tainted, the next plan will show that the resource will be _____ and

_____ and the next apply will implement this change.
* recreated and tainted
* destroyed and not recreated
* tainted and not destroyed
* destroyed and recreated

NO.81 HashiCorp Configuration Language (HCL) supports user-defined functions.
* True
* False

Explanation

https://www.terraform.io/language/functions

The Terraform language does not support user-defined functions, and so only the functions built into the

language are available for use

NO.82 What allows you to conveniently switch between multiple instances of a single configuration within its single

backend?
* Local backends
* Providers
* Remote backends
* Workspaces

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

NO.83 How can terraform plan aid in the development process?
* Validates your expectations against the execution plan without permanently modifying state
* Initializes your working directory containing your Terraform configuration files
* Formats your Terraform configuration files
* Reconciles Terraform’s state against deployed resources and permanently modifies state using the current status of deployed resources

NO.84 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?
* Local Backends
* This can be done using any of the local or remote backends
* Remote Backends
* Terraform Backends

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

NO.85 lookup retrieves the value of a single element from which of the below data type?
* map
* set
* string
* list

Explanation

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

NO.86 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’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’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’s unsupported (e.g. variable, depends_on, module.source, provider) missing value for a variable (none of -var foo=… 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