

Terraform-Associate-004資格模擬 & Terraform-Associate-004参考書内容



BONUS!!! ShikenPASS Terraform-Associate-004ダンプの一部を無料でダウンロード：https://drive.google.com/open?id=1JlqJns2XrhLeX36_Y3qeF6IONVJhYsIm

あなたはIT職員ですか。成功したいのですか。成功したいのならShikenPASSのHashiCorpのTerraform-Associate-004試験トレーニング資料を利用してください。当社の資料は実践の検証に合格したもので、あなたが首尾よくIT認証試験に合格することを助けます。ShikenPASSのHashiCorpのTerraform-Associate-004トレーニング資料を手に入れたらあなたはIT業種でもっとよい昇進を持つようになり、高レベルのホワイトカラーのトリートメントを楽しむこともできます。あなたはまだ何を心配しているのですか。ShikenPASSのHashiCorpのTerraform-Associate-004トレーニング資料はあなたのニーズを満たすことができますから、躊躇わずにShikenPASSを選んでください。ShikenPASSはあなたと苦楽を共にして、一緒に挑戦に直面します。

HashiCorp Terraform-Associate-004 認定試験の出題範囲：

トピック	出題範囲
トピック 1	Terraform state management: This domain focuses on managing Terraform's state file, understanding local and remote backends, implementing state locking, and handling resource drift.
トピック 2	Terraform modules: This domain explains organizing and reusing code through modules, understanding variable scope between modules, implementing modules in configurations, and managing module versions.
トピック 3	Core Terraform workflow: This domain focuses on the essential workflow steps: initializing directories, validating configurations, generating execution plans, applying changes, destroying infrastructure, and formatting code.
トピック 4	Terraform fundamentals: This domain addresses installing and managing provider plugins, understanding Terraform's provider architecture, and how Terraform tracks infrastructure state.
トピック 5	Maintain infrastructure with Terraform: This domain addresses importing existing infrastructure into Terraform, inspecting state using CLI commands, and using verbose logging for troubleshooting.
トピック 6	HCP Terraform: This domain covers using HashiCorp Cloud Platform Terraform for infrastructure provisioning, collaboration and governance features, organizing workspaces and projects, and configuring integrations.

>> Terraform-Associate-004資格模擬 <<

最高Terraform-Associate-004 | 効果的なTerraform-Associate-004資格模擬試験 | 試験の準備方法HashiCorp Certified: Terraform Associate (004)

(HCTA0-004)参考書内容

クライアントがTerraform-Associate-004試験トレントの料金を支払うと、5～10分でシステムから送信されたメールを受け取ります。その後、クライアントはリンクをたいてダウンロードすることができ、HashiCorpその後、Terraform-Associate-004質問トレントを使用して学習できます。試験の準備をする人にとって時間は非常に重要であるため、クライアントは支払い後すぐにダウンロードできるため、Terraform-Associate-004ガイド急流の大きな利点です。したがって、クライアントがTerraform-Associate-004試験問題を使用して学習することは非常に便利です。

HashiCorp Certified: Terraform Associate (004) (HCTA0-004) 認定 Terraform-Associate-004 試験問題 (Q68-Q73):

質問 # 68

Which of the following should you add in the `required_providers` block to define a provider version constraint?

- A. `version: 3.1`
- B. `version - 3.1`
- C. `version`
- D. `version = "3.1"`

正解: D

解説:

Rationale for Correct Answer (B):

Provider version constraints in Terraform are specified using the `version` argument in the `required_providers` block, e.g.:

```
terraform {  
  required_providers {  
    aws = {  
      source = "hashicorp/aws"  
      version = "3.1"  
    }  
  }  
}
```

This ensures Terraform always uses a specific provider version (or constraint expression).

Analysis of Incorrect Options:

- A). `version`: Incomplete, no value specified.
- C). `version: 3.1`: Incorrect syntax, Terraform uses `=` not `:`.
- D). `version - 3.1`: Incorrect syntax, `-` is invalid here.

Key Concept:

Defining provider version constraints ensures consistent provider behavior across environments and avoids breaking changes.

Reference:Terraform Exam Objective - Manage Terraform Resources and Providers.

質問 # 69

Which of the following can you do with `terraform plan`? (Pick 2 correct responses)

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

正解: B、D

解説:

Rationale for Correct Answers:

B). View the execution plan: Correct - `terraform plan` shows what Terraform intends to do (create, update, or destroy).

C). Save a generated execution plan: Correct - using `terraform plan -out=planfile` lets you save the plan to apply later with `terraform apply planfile`.

Analysis of Incorrect Options:

A). Schedule runs in the future: Incorrect, Terraform does not provide scheduling; external tools (like cron, CI

/CD) are required.

D). Execute a plan in a different workspace: Incorrect, execution happens in the current workspace; switching workspaces must be done before running the plan.

Key Concept:

terraform plan is used for previewing and optionally saving an execution plan for controlled and predictable infrastructure changes.

Reference: Terraform Exam Objective - Understand Terraform Basics and CLI.

質問 # 70

Which two steps are required to provision new infrastructure in the Terraform workflow? Choose two correct answers.

- A. Plan
- B. Alidate
- **C. Init**
- D. Import
- **E. apply**

正解: C、 E

解説:

The two steps that are required to provision new infrastructure in the Terraform workflow are init and apply.

The terraform init command initializes a working directory containing Terraform configuration files. It downloads and installs the provider plugins that are needed for the configuration, and prepares the backend for storing the state. The terraform apply command applies the changes required to reach the desired state of the configuration, as described by the resource definitions in the configuration files. It shows a plan of the proposed changes and asks for confirmation before making any changes to the infrastructure. References =

[The Core Terraform Workflow], [Initialize a Terraform working directory with init], [Apply Terraform Configuration with apply]

質問 # 71

Which of the following module source paths does not specify a remote module?

- A. Source = "github.com:microp/example"
- B. Source = "git@github.com:hasicrop/example.git"
- **C. Source = "module/consul"**
- D. Source = "hasicrop/consul/aws"

正解: C

解説:

The module source path that does not specify a remote module is source = "module/consul". This specifies a local module, which is a module that is stored in a subdirectory of the current working directory. The other options are all examples of remote modules, which are modules that are stored outside of the current working directory and can be accessed by various protocols, such as Git, HTTP, or the Terraform Registry. Remote modules are useful for sharing and reusing code across different configurations and environments. References = [Module Sources], [Local Paths], [Terraform Registry], [Generic Git Repository], [GitHub]

質問 # 72

What does terraform destroy do?

- A. Destroys the Terraform state file, leaving the infrastructure intact.
- **B. Destroys all infrastructure in the Terraform state file.**
- C. Destroys all Terraform code files in the current directory, leaving the state file intact.
- D. Destroys all infrastructure in the configured Terraform provider.

正解: B

解説:

* Rationale for Correct answer: terraform destroy removes all resources currently tracked in the Terraform state file by issuing destroy requests to the configured providers. It does not delete configuration files or providers themselves.

さらに、ShikenPASS Terraform-Associate-004ダンプの一部が現在無料で提供されています：https://drive.google.com/open?id=1JlqJns2XrhLeX36_Y3qeF6lONVJhYsIm