

HashiCorp Certified Terraform Associate 003

The background features abstract, glowing purple geometric shapes, including a large cube on the left and various lines and arcs extending across the frame, set against a dark, textured purple background.

4. Use Terraform outside of core workflow

The background features abstract, glowing blue and white geometric shapes, including a large cube-like structure on the left and various lines and arcs extending across the frame, creating a technical or architectural feel.

**Describe when to use
terraform import to import
existing infrastructure into
your Terraform state**

O que é o Terraform Import?

O **Terraform Import** é um comando do Terraform que permite importar recursos existentes de provedores de infraestrutura para o estado do Terraform. Ele é usado para trazer recursos já existentes, criados manualmente ou por outros meios, para dentro do controle do Terraform, permitindo que você gerencie esses recursos como código.

- Ao importar um recurso com o **Terraform Import**, você está trazendo as informações desse recurso para o estado do Terraform, o que permite ao Terraform ciente da existência desse recurso e também de suas configurações atuais.
- Essa funcionalidade é útil quando você já possui infraestrutura existente e deseja começar a gerenciá-la com o Terraform. Ao importar recursos, você pode aproveitar todos os benefícios do Terraform, como automação, controle de versão e colaboração.

Como funciona a importação no Terraform?

Quando você realiza a importação de um recurso, o Terraform cria uma entrada correspondente no estado do Terraform para representar esse recurso. Essa entrada no estado é utilizada para rastrear o estado atual do recurso e gerenciar as alterações subsequentes.

Ao importar um recurso, você está essencialmente dizendo ao Terraform:

“Ei, este recurso já existe, e eu quero gerenciá-lo usando o Terraform”.

Ao executar o comando de importação, o Terraform consulta o provedor de infraestrutura, localiza o recurso correspondente com base no identificador fornecido e obtém as informações necessárias para criar o estado correspondente no arquivo **terraform.tfstate**.

Como funcionava a importação antes da v1.5?

O fluxo de trabalho consistia primeiro em executar o comando `terraform import` com o endereço desejado do recurso em sua configuração e o identificador exclusivo do recurso no ambiente existente.

O comando de importação pesquisaria o recurso existente e reuniria os atributos desse recurso e, em seguida, criaria uma entrada de dados de estado para ele usando o endereço desejado especificado. Isso não criaria o bloco de configuração de recursos em seu código Terraform, isso dependia de você.

- Antes da versão 1.5, o `terraform import` era mais limitado em relação aos recursos que poderiam ser importados.
- A sintaxe para importação de recursos era mais complexa e dependia da implementação de um código de recurso correspondente no arquivo de configuração do Terraform.

Como funciona a importação na v1.5?

O novo bloco de importação, introduzido no Terraform 1.5, é uma evolução do comando de importação anterior. Sua utilização segue o padrão dos fluxos de trabalho típicos do Terraform.

1. Adicionando Blocos de Importação:

- A sintaxe é simples: a palavra-chave **import** seguida do argumento **to** para especificar o endereço do recurso na configuração e o argumento **id** para o identificador exclusivo do recurso existente.

2. Executando um Plano do Terraform:

- Após adicionar os blocos de importação, um plano do Terraform é executado para visualizar as alterações propostas.

3. Aplicando as Alterações:

- Se satisfeito com o plano, você pode aplicá-lo para gerenciar os recursos. Ao finalizar, os blocos de importação podem ser removidos da configuração.

4. Gerando Blocos de Recursos (Experimental):

- Um recurso experimental permite a geração automática de blocos de recursos usando o sinalizador **-generate-config-out** com o comando **plan**.

5. Revisão e Ajustes:

- Os blocos gerados podem ser revisados e ajustados conforme necessário antes da execução do plano final.

6. Pontos de Atenção:

- Este recurso experimental pode apresentar algumas configurações iniciais incorretas, mas serve como um ponto de partida sólido para a construção da configuração.

The background features a dark, textured purple field. On the left side, there are several glowing purple wireframe shapes, including a large cube and a smaller rectangular prism, along with some circular arcs. These shapes are composed of thin, bright purple lines.

Hands-On

Terraform Import



The background features several glowing purple geometric shapes, including a large cube on the left and various intersecting lines and arcs, creating a technical or architectural feel.

**Use terraform state to view
Terraform state**

O que é Terraform State?

O Terraform State é um registro do estado atual da sua infraestrutura provisionada.

Ele mantém informações sobre os recursos criados, suas configurações e dependências.

Isso permite ao Terraform determinar qual é o estado atual da infraestrutura e quais ações precisam ser executadas para atingir o estado desejado.



TFTEC CLOUD

Por que o Terraform State é Importante?

Consistência: O Terraform State garante que a configuração declarativa do Terraform corresponda à realidade atual.

Controle de Alterações: Ele rastreia as alterações realizadas na infraestrutura e ajuda a prever os impactos de futuras alterações.

Colaboração: Facilita a colaboração entre membros da equipe, garantindo que todos tenham uma visão consistente da infraestrutura.

Recurso de Bloqueio: Evita que várias instâncias do Terraform modifiquem o estado simultaneamente, evitando conflitos.

Comandos Básicos do Terraform State

1. **Terraform state list:** Lista todos os recursos gerenciados pelo estado do Terraform.
2. **terraform state show <resource_name>:** Exibe detalhes sobre um recurso específico no estado, incluindo seus atributos e metadados.
3. **terraform state pull:** Obtém e exibe o estado atual em formato JSON. Isso é útil para automação e integração com outras ferramentas.

```
1  terraform state show module.storage_module.random_string.random
2  # module.storage_module.random_string.random:
3  resource "random_string" "random" {
4      id          = "ggqy0h"
5      length      = 5
6      lower       = true
7      min_lower    = 0
8      min_numeric  = 0
9      min_special  = 0
10     min_upper    = 0
11     number       = true
12     numeric      = true
13     result       = "ggqy0h"
14     special      = false
15     upper        = false
16 }
```


The background features abstract, glowing purple geometric shapes, including a large cube-like structure on the left and various intersecting lines and arcs, creating a futuristic, technical aesthetic.

Hands-On

Terraform State CLI

The background features abstract, glowing purple geometric shapes, including a large cube-like structure on the left and various lines and arcs extending across the frame. The overall aesthetic is futuristic and digital.

**Describe when to enable
verbose logging and what
the outcome/value is**

Quando habilitar o log detalhado no Terraform?

1. **Solução de Problemas:** O log detalhado é particularmente útil ao solucionar problemas em suas configurações do Terraform. Se algo não está funcionando como esperado, habilitar o log detalhado pode fornecer informações valiosas sobre o que está acontecendo nos bastidores.
2. **Entendendo o Fluxo de Trabalho:** Se você está aprendendo como o Terraform funciona internamente ou deseja entender melhor o fluxo de trabalho durante a execução do plano e da aplicação, o log detalhado pode ser uma ferramenta educacional.
3. **Depuração de Recursos Específicos:** Às vezes, você pode enfrentar problemas com recursos específicos. O log detalhado pode ajudar a rastrear a interação do Terraform com esses recursos e identificar problemas potenciais.

Como habilitar o log detalhado

Para habilitar o log detalhado no Terraform, você pode usar a variável de ambiente TF_LOG e definir seu valor para um dos seguintes níveis:

- **TRACE:** O nível mais detalhado de log, fornece informações extremamente detalhadas sobre cada ação do Terraform.
- **DEBUG:** Um nível menos detalhado que o TRACE, mas ainda fornece muitas informações úteis para depuração.
- **INFO:** O nível padrão do log, que fornece informações gerais sobre o progresso do Terraform.
- **WARN:** Apenas informações de aviso e erros são registradas.
- **ERROR:** Apenas mensagens de erro são registradas.

Resultado de valor

Habilitar o log detalhado oferece os seguintes resultados e valor:

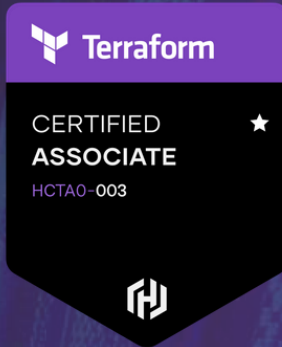
- **Maior Visibilidade:** Você obtém uma visão detalhada do que o Terraform está fazendo internamente, o que pode ajudar a identificar problemas e entender o fluxo de trabalho.
- **Informações de Depuração:** O log detalhado fornece informações úteis para depuração de problemas específicos, incluindo interações com APIs de provedores de nuvem e recursos.
- **Educação:** Se você está aprendendo sobre o Terraform, o log detalhado pode ser uma ferramenta educacional valiosa para entender como as ações são executadas.
- **Cautela:** Lembre-se de que, devido à quantidade de informações registradas no log detalhado, ele pode ser volumoso. Use-o com moderação e desative-o quando não for mais necessário.

The background features abstract, glowing purple geometric shapes, including a large cube-like structure on the left and various intersecting lines and arcs, creating a technical or architectural feel.

Hands-On

Debugging Terraform





Obrigado