# 2 Getting Started-Copy Break the bilingual relationship

| **Issue** | 01 |
| **Date** | 2023-12-22 |



**HUAWEI TECHNOLOGIES CO., LTD.**

# Huawei Technologies Co., Ltd.

Address: Huawei Industrial Base
Bantian, Longgang
Shenzhen 518129
People's Republic of China

Website: https://www.huawei.com

Email: support@huawei.com

# Security Declaration

## Vulnerability

Huawei's regulations on product vulnerability management are subject to the *Vul. Response Process.* For details about this process, visit the following web page:
https://www.huawei.com/en/psirt/vul-response-process
For vulnerability information, enterprise customers can visit the following web page:
https://securitybulletin.huawei.com/enterprise/en/security-advisory

2 Getting Started-Copy Break the bilingual relationship
2 Getting Started-Copy Break the bilingual relationship
Contents

# Contents

2 Getting Started-Copy Break the bilingual
relationship
2 Getting Started-Copy Break the bilingual
relationship
1 Getting Started

<div align="right">

# 1 Getting Started

</div>

This topic walks you through the process of installing Terraform and using Terraform to create a virtual private cloud (VPC).

## Installing Terraform

Terraform is distributed as a single binary. Download a Terraform package and decompress it to a directory included in your system's **PATH**.

**Step 1** Go to **Terraform** and download a Terraform package that matches your operating system.

**Step 2** Decompress the package and add the directory where the package is located to the system's **PATH**.

**Step 3** Run the following command in the command-line interface (CLI) to check whether the path is correctly configured:

**terraform**

If the following information is displayed, the configuration is correct and Terraform is ready to run.

```
Usage: terraform [-version] [-help] <command> [args]

....
```

**----End**

## Authentication

Terraform supports orchestration of multiple cloud resources. Before using Terraform to manage cloud resources, you need to obtain the AK/SK and configure them on Terraform for authentication.

You can configure authentication by using either static credentials or environment variables.

- **Static credentials**

  Configure the following parameters to add AK/SK in the Terraform configuration file:

```
provider "huaweicloud" {
  region     = "eu-west-101"
```

2 Getting Started-Copy Break the bilingual
relationship
2 Getting Started-Copy Break the bilingual
relationship

1 Getting Started

```
access_key = "my-access-key"
secret_key = "my-secret-key"
}
```

- **region**: region where the resources are to be created and managed. Such as: "eu-west-101".

- **access_key**: access secret ID (AK). For details about how to query the AK/SK, see **Access Keys**.

- **secret_key**: secret access key (SK). For details about how to query the AK/SK, see **Access Keys**.

- **Environment variables**

  Configure the region, AK, and SK as environment variables. For example:

  ```
  $ export HW_REGION_NAME="eu-west-101"
  $ export HW_ACCESS_KEY="my-access-key"
  $ export HW_SECRET_KEY="my-secret-key"
  ```

  - **HW_REGION_NAME**: region where the resources are to be created and managed. Such as: "eu-west-101".

  - **HW_ACCESS_KEY**: access secret ID (AK). For details about how to query the AK/SK, see **Access Keys**.

  - **HW_SECRET_KEY**: secret access key (SK). For details about how to query the AK/SK, see **Access Keys**.

  For details about more parameters, see **https://registry.terraform.io/providers/ huaweicloud/huaweicloud/latest/docs**.

## Creating a VPC Using Terraform

This example shows how to create a VPC using Terraform. Terraform version 0.13 is used, and the HUAWEI CLOUD provider version is 1.20.0.

**Step 1** Create the **versions.tf** file in the working directory and specify the registry source and HUAWEI CLOUD provider version.

```
terraform {
  required_providers {
    huaweicloud = {
      source = "huaweicloud/huaweicloud"
      version = ">= 1.20.0"
    }
  }
}
```

**Step 2** Create the **main.tf** file, configure the HUAWEI CLOUD provider, and create a VPC.

```
# Configure the HuaweiCloud Provider
provider "huaweicloud" {
  region     = "eu-west-101"
  access_key = "my-access-key"
  secret_key = "my-secret-key"
}

# Create a VPC.
resource "huaweicloud_vpc" "example" {
  name = "terraform_vpc"
  cidr = "192.168.0.0/16"
}
```

The first part configures the provider and provides AK/SK for authentication. For details on how to configure these parameters, see **Authentication**. If you provide credentials using environment variables, skip this part.

2 Getting Started-Copy Break the bilingual
relationship
2 Getting Started-Copy Break the bilingual
relationship

1 Getting Started

The second part describes a VPC resource named **example**. The VPC name is
**terraform_vpc** and the CIDR block is **192.168.0.0/16**.

**Step 3** Run the following command to perform initialization:

**terraform init**

The following command output is displayed. The provider will be downloaded and
installed when you run this command for the first time.

```
$ terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of huaweicloud/huaweicloud
- Installing huaweicloud/huaweicloud v1.20.0...
...
Terraform has been successfully initialized!
```

**Step 4** Run the following command to view the resources to be created:

**terraform plan**

The following command output is displayed. Terraform prints the resources to be
created.

```
...
An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # huaweicloud_vpc.example will be created
  + resource "huaweicloud_vpc" "example" {
     + cidr   = "192.168.0.0/16"
     + id     = (known after apply)
     + name   = "terraform_vpc"
     + region = (known after apply)
     + routes = (known after apply)
     + shared = (known after apply)
     + status = (known after apply)
   }

Plan: 1 to add, 0 to change, 0 to destroy.
...
```

**Step 5** Run the following command to create the resources:

**terraform apply**

Enter **yes** as prompted. The following information is displayed. The VPC named
**terraform_vpc** has been created. You can check the VPC on the console.

```
An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # huaweicloud_vpc.example will be created
  + resource "huaweicloud_vpc" "example" {
     + cidr   = "192.168.0.0/16"
     + id     = (known after apply)
     + name   = "terraform_vpc"
     + region = (known after apply)
     + routes = (known after apply)
```

2 Getting Started-Copy Break the bilingual
relationship
2 Getting Started-Copy Break the bilingual
relationship

1 Getting Started

```
        + shared = (known after apply)
        + status = (known after apply)
      }

Plan: 1 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

    Enter a value: yes

huaweicloud_vpc.example: Creating...
huaweicloud_vpc.example: Creation complete after 7s [id=ceab8267-38e5-4a4c-8065-12967ad9eb31]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
```

**----End**