边缘安全加速平台 EO

Terraform



腾讯云

版权所有：腾讯云计算（北京）有限责任公司

【版权声明】

©2013-2026 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分内容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

【商标声明】



及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

【服务声明】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

【联系我们】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100或95716。

文档目录

Terraform
Terraform 简介
安装和配置 Terraform

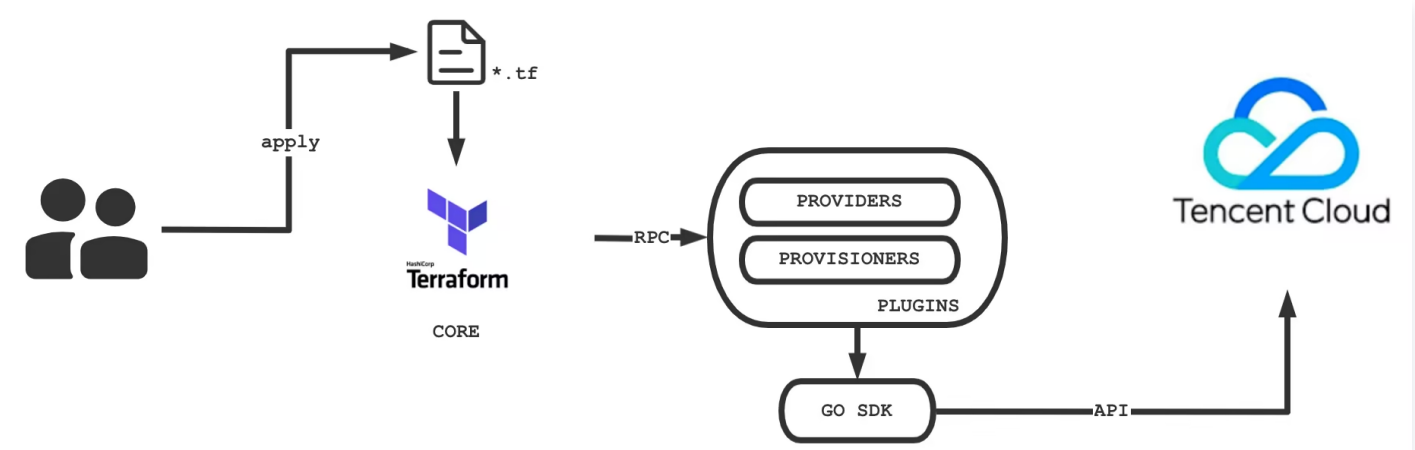
版权所有：腾讯云计算（北京）有限责任公司

Terraform
Terraform 简介

最近更新时间：2024-04-18 09:47:31

Terraform 是什么?

Terraform 是一款使用 Go 编写、运行在客户端、开源的资源编排工具。基于 HashiCorp Plugin 的架构设计，赋予 Terraform 高度可扩展的特性。目前腾讯云也基于 Terraform Plugin 实现了 TencentCloud Provider，支持通过 Terraform 管理腾讯云上资源。示意图如下：



TencentCloud Provider 基于 tencentcloud-sdk-go 实现，目前已经提供了超过183个 Resource 和158个 Data Source，覆盖计算、存储、网络、容器服务、负载均衡、中间件、数据库等超过30款产品，已满足众多用户的基本上云需求。
您可通过 EdgeOne 提供的 Terraform 文档 及 Terraform 编写样例，来快速了解并开始使用 Terraform。

Terraform 优势

多云编排

Terraform 适用于多云方案，您可将相类似的基础结构部署到腾讯云、其他云提供商或本地数据中心。开发人员能够使用相同的工具和相似的配置文件同时管理不同云提供商的资源。

基础设施及代码

基础设施可以使用高级配置语法 HCL 进行描述，使得基础设施能够被代码化和版本化，从而可以进行共享和重复使用。示例如下：

```
# 以 NS 接入方式创建站点 example.com
resource "tencentcloud_teo_zone" "zone" {
  zone_name      = "example.com"
```

```
  # 通过 zone_available_plans 查询您可用的套餐信息
  plan_type      = "<your-plan-type>"
  type           = "full"
  paused         = false
  cname_speed_up = "enabled"
}

# 创建 example.com 的 DNS 记录
resource "tencentcloud_teo_dns_record" "dns_record" {
  zone_id     = tencentcloud_teo_zone.zone.id
  type        = "A"
  name        = "example.com"
  # 开启 CDN 加速服务
  mode        = "proxied"
  content     = "<your-backend-ip>"
  ttl         = 60
}
```

执行计划

Terraform 具备 “Planning” 步骤，执行 terraform plan 命令后 Terraform 会生成执行计划。执行计划会显示当调用 apply 时 Terraform 的状态，您可通过该能力在 Terraform 操作基础设施时避免任何意外。示例如下：

```
Terraform used the selected providers to generate the following
execution plan. Resource actions are indicated with the following
symbols:
  + create

Terraform will perform the following actions:

  # tencentcloud_teo_dns_record.dns_record will be created
  + resource "tencentcloud_teo_dns_record" "dns_record" {
      + cname          = (known after apply)
      + content        = "<your-backend-ip>"
      + created_on     = (known after apply)
      + domain_status  = (known after apply)
      + id             = (known after apply)
      + locked         = (known after apply)
      + mode           = "proxied"
```

```
      + modified_on    = (known after apply)
      + name           = "example.com"
      + priority       = (known after apply)
      + type           = "A"
      + status         = (known after apply)
      + ttl            = 60
      + zone_id        = (known after apply)
    }

  # tencentcloud_teo_zone.zone will be created
  + resource "tencentcloud_teo_zone" "zone" {
      + area                    = (known after apply)
      + cname_speed_up          = "enabled"
      + cname_status            = (known after apply)
      + created_on              = (known after apply)
      + id                      = (known after apply)
      + modified_on             = (known after apply)
      + name                    = "example.com"
      + name_servers            = (known after apply)
      + original_name_servers   = (known after apply)
      + paused                  = false
      + plan_type               = "sta"
      + status                  = (known after apply)
      + type                    = "full"
      + vanity_name_servers_ips = (known after apply)

      + vanity_name_servers {
          + servers = (known after apply)
          + switch  = (known after apply)
        }
    }

Plan: 2 to add, 0 to change, 0 to destroy.

------------------------------------------------------------------------
-------
```

```
Note: You didn't use the -out option to save this plan, so Terraform
can't guarantee to take exactly these actions if you run "terraform
apply" now.
```

自动变更

您可在基础设施上通过最少的人工干预工作，应用复杂的变更集。通过前文中的执行计划及资源拓扑，您可准确获取 Terraform 动态，避免可能的人为错误。

远程状态管理

Terraform 引入了远程状态存储机制 Backend，目前腾讯云可通过 对象存储 COS 来管理用户的 tfState 文件，避免将文件保存在本地，造成丢失。同时，远程存储使多人同时对 Terraform 资源进行管理成为可能。

安装和配置 Terraform

最近更新时间：2026-01-06 14:47:42

本文介绍如何安装和配置 Terraform。

步骤1：安装 Terraform

1. 前往 Terraform 官网，使用命令行直接安装 Terraform 或下载二进制安装文件。
2. 解压并配置全局路径。
   若您通过命令行安装，则请跳过此步骤。若您通过下载二进制安装文件，则请配置全局路径。

Linux 及 macOS

1. 执行以下命令进行解压，请将 1.x.x 替换为您实际安装 Terraform 版本号。

```
unzip terraform_1.x.x_linux_amd64.zip
```

2. 执行以下命令，将当前目录添加至 `~/.profile` 文件中。

```
echo $"export PATH=\$PATH:$(pwd)" >> ~/.bash_profile
```

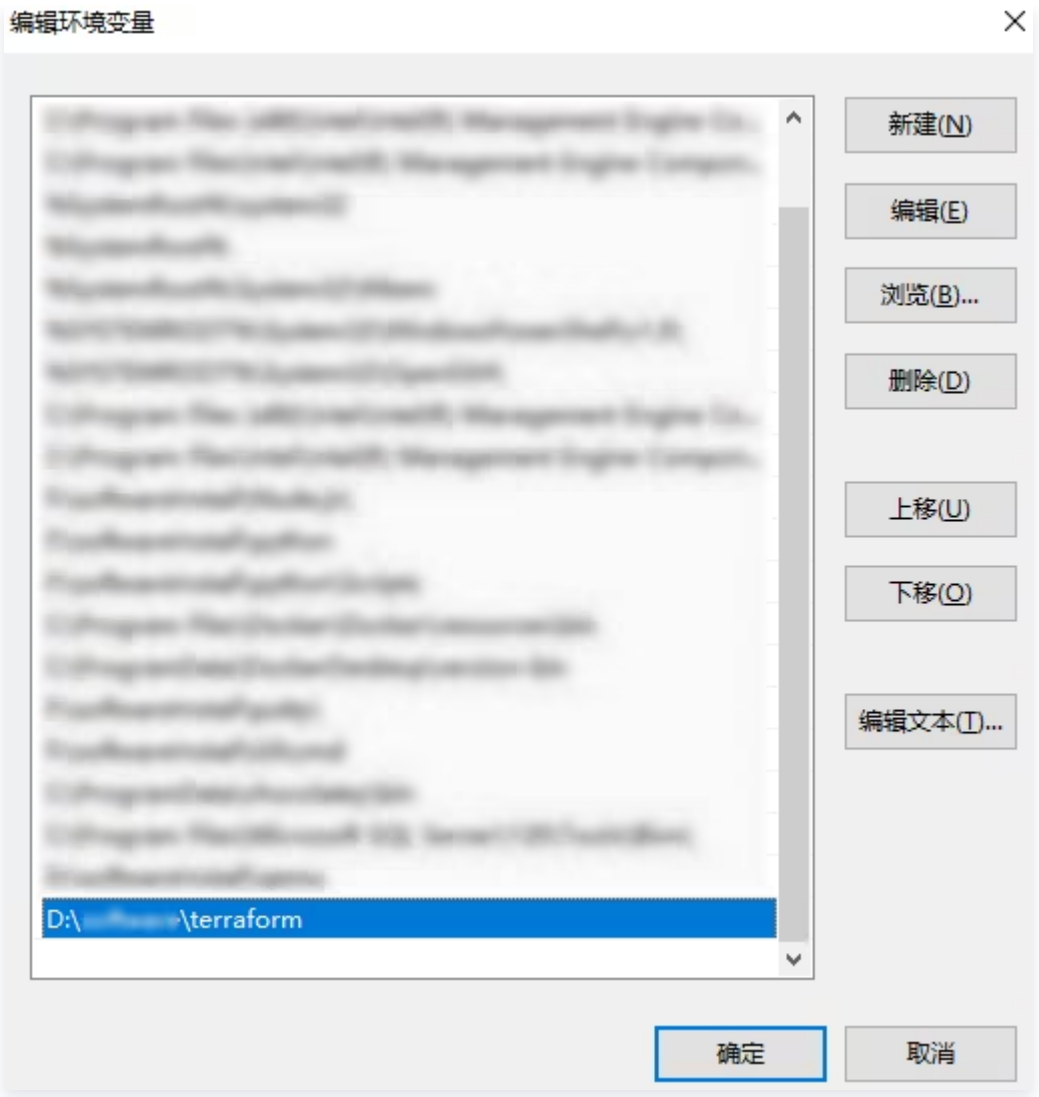
3. 执行以下命令，使全局路径配置生效。

```
source ~/.bash_profile
```

Windows

1. 在桌面中，右键选择“计算机”，并在弹出菜单中选择**属性**。
2. 在弹出窗口中，找到并单击**高级系统设置**。
3. 在弹出的“系统属性”窗口中，单击**环境变量**。
4. 在“系统变量”中，找到 `Path`，将 `terraform.exe` 的绝对路径加入其中。如下图所示：
   本文以 Windows 10 操作系统为例，`terraform.exe` 位于 `D:\xxxx\terraform` 文件夹下。

版权所有：腾讯云计算（北京）有限责任公司



5. 单击**确定**即可。

3. 执行以下命令，查看是否安装成功。

```
terraform  -version
```

返回信息如下所示（版本号可能存在差异），则表示安装成功。

```
> Terraform v1.0.10
> on darwin_amd64
> Your version of Terraform is out of date! The latest version
> is 1.1.0. You can update by downloading from
https://www.terraform.io/downloads.html
```

步骤2：配置安全凭证

在首次使用 Terraform 之前，请前往 云 API 密钥页面 申请安全凭证 SecretId 和 SecretKey。若已有可使用的安全凭证，则跳至第3步。

1. 登录 访问管理控制台，在左侧导航栏，选择**访问密钥 > API 密钥管理**。
2. 在 API 密钥管理页面，单击 **新建密钥**，即可创建一对 SecretId/SecretKey。



3. 您可通过以下两种方式进行鉴权：

环境变量鉴权

请将如下信息添加至环境变量配置：
`<your-secret-id>` 及 `<your-secret-key>` 请替换为 获取凭证 中的 `SecretId` 和 `SecretKey` 。

```
export TENCENTCLOUD_SECRET_ID=<your-secret-id>
export TENCENTCLOUD_SECRET_KEY=<your-secret-key>
```

静态凭证鉴权

在用户目录下创建 `provider.tf` 文件，输入如下内容：
`<your-secret-id>` 及 `<your-secret-key>` 请替换为 获取凭证 中的 `SecretId` 和 `SecretKey` 。

```
provider "tencentcloud" {
    secret_id = "<your-secret-id>"
    secret_key = "<your-secret-key>"
}
```

4. 到此已完成 Terraform 安装和环境变量配置，可进行下一步：通过 Terraform 创建站点 。