PACKETFABRIC DOCUMENTATION

Filter

packetfabric provider
- Guides
  - Early Termination Liability (ETL) Fees
  - Getting Started
  - Importing Resources
  - Service Labels
- Resources
- Data Sources

PacketFabric Provider

You can use the PacketFabric Terraform provider to create and manage PacketFabric services.

For information on how to get started, see Getting Started or Terraform with PacketFabric in the PacketFabric knowledge base.

Before you begin

You will need a PacketFabric user account with appropriate permissions. See User Permissions in the PacketFabric documentation.

You will also need an API key. See Getting Started.

Using with other Terraform providers

If you are using Terraform to create cloud connections, there are typically actions that you need to take from the cloud side (for example, accepting connections or retrieving service keys).

To do this, you will need appropriate access to that cloud environment.

You can use their Terraform provider:

- https://registry.terraform.io/providers/hashicorp/aws/latest
- https://registry.terraform.io/providers/hashicorp/google/latest
- https://registry.terraform.io/providers/hashicorp/azurerm/latest
- https://registry.terraform.io/providers/oracle/oci/latest
- https://registry.terraform.io/providers/IBM-Cloud/ibm/latest

For examples on how to use a cloud's Terraform provider alongside PacketFabric, see examples/use-cases.

Additionally, you can simplify the configuration and management of PacketFabric Cloud Routers by utilizing the PacketFabric Terraform Cloud Router Module. This module provides pre-defined configurations and workflows for provisioning cloud routers on the PacketFabric platform.

Example Usage

```
terraform {
  required_providers {
    packetfabric = {
      source  = "PacketFabric/packetfabric"
      version = ">= 1.6.0"
    }
  }
}

provider "packetfabric" {}
```

Environment Variables

> **Warning:**
>
> Hard-coded credentials are not recommended in any Terraform configuration. Using environment variables is highly recommended.

Credentials can be provided by using the `PF_TOKEN`, and optionally `PF_USER` and `PF_PASSWORD` environment variables.

For example:

```
$ export PF_TOKEN="secret"
$ terraform init
$ terraform plan
```

Account ID can be provided by using the `PF_ACCOUNT_ID` environment variable.

For example:

```
$ export PF_ACCOUNT_ID="123456789"
```

Schema

Optional

ON THIS PAGE

- Before you begin
- Using with other Terraform providers
- Example Usage
- Environment Variables
- Schema

Report an issue

- `host` (String) PacketFabric API endpoint. Can also be set with the PF_HOST environment variable. Defaults to https://api.packetfabric.com
- `password` (String, Sensitive) PacketFabric username. Can also be set with the PF_PASSWORD environment variable.
- `token` (String, Sensitive) PacketFabric API access token. Can also be set with the PF_TOKEN environment variable.
- `username` (String, Sensitive) PacketFabric username. Can also be set with the PF_USER environment variable.