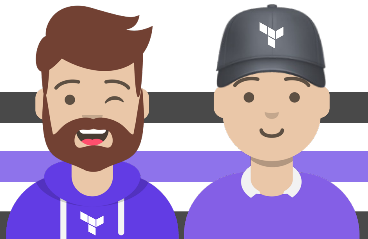
Terraform Cloud Private Registry

What is the Private Registry?

Private/Custom Providers

Modules

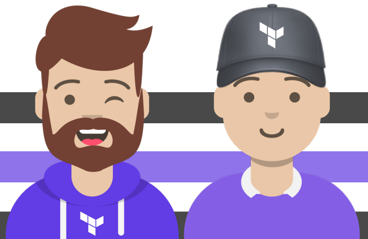
Policies



A Quick Note

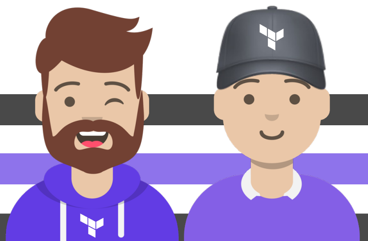
The **Private Registry** feature was previously known as the *Private Module Registry*.

It now supports custom providers and an active exploration to include policies, so it's no longer just for modules, hence the name change. We will refer to it as the **Private Registry**, but you may still see it referred to as the *Private Module Registry* in HashiCorp documentation.

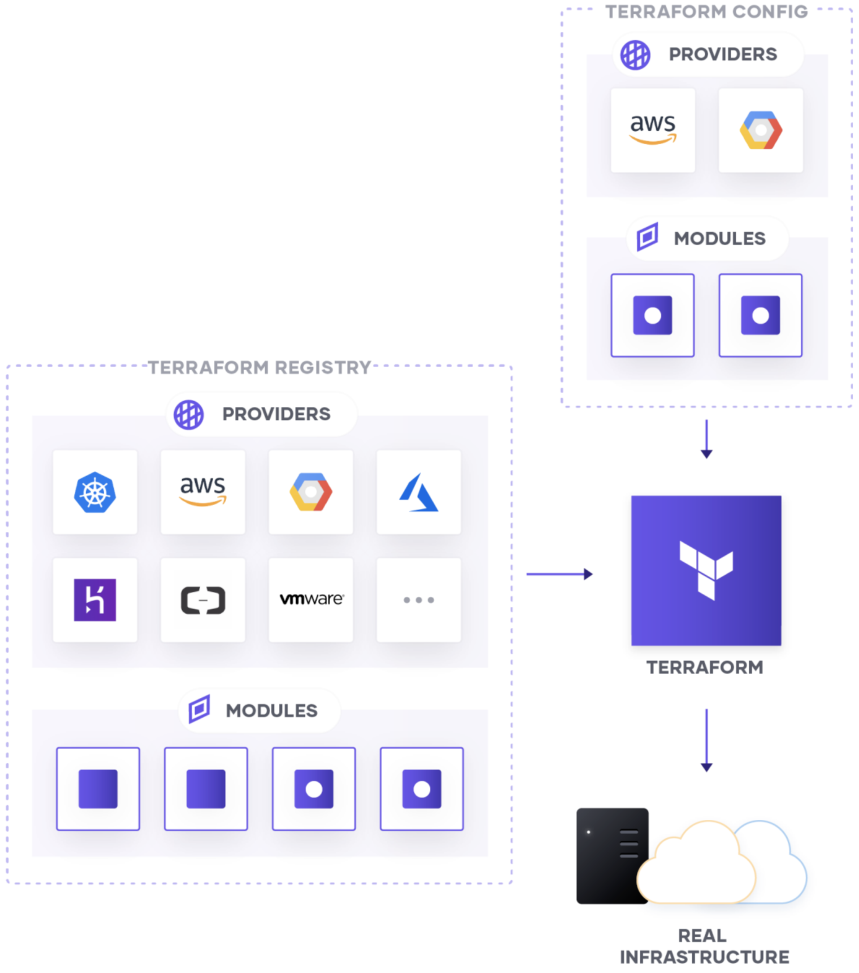


What is the Private Registry?

- Terraform Cloud's private registry allows you to curate the modules and providers your organization uses, which eases discoverability.
- Like the public registry, you can store and share custom modules, providers, and sentinel policies within your organization
- These modules are NOT public
- Supports versioning and users can easily search for providers and modules
- Enables your organization to publish approved modules for consumption and standardization



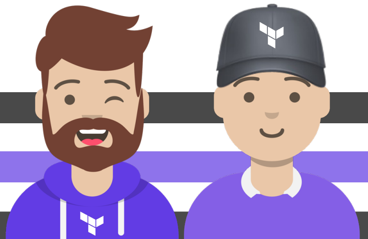
What is the Private Registry?



Providers are the plugins that Terraform uses to manage infrastructure resources.

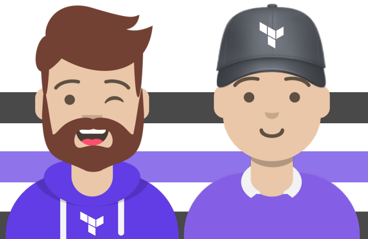
Modules are reusable Terraform configurations that can be called and configured by other configurations.

Terraform Registry makes it easy to use any provider or module.

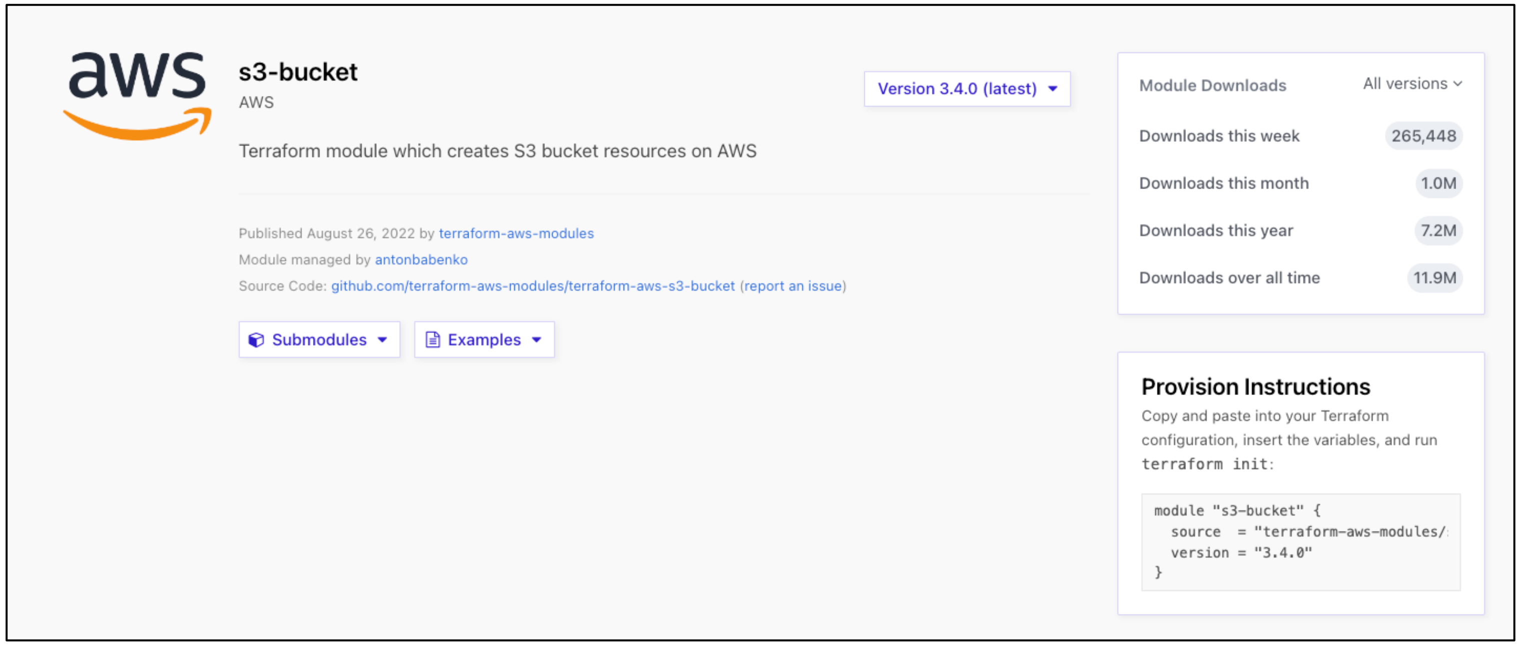


Private Registry: Providers

- Clearly designate which public providers are recommended for the organization and makes their supporting documentation and examples centrally accessible.
- Providers hosted on the public Terraform Registry can automatically synchronize to the Terraform Cloud organization's private registry.
- Supports the ability to publish private providers that are unique to your organization.



What is a Terraform Module


s3-bucket
AWS

Version 3.4.0 (latest)

Terraform module which creates S3 bucket resources on AWS

Published August 26, 2022 by terraform-aws-modules
Module managed by antonbabenko
Source Code: github.com/terraform-aws-modules/terraform-aws-s3-bucket (report an issue)

Module Downloads — All versions

Downloads this week	265,448
Downloads this month	1.0M
Downloads this year	7.2M
Downloads over all time	11.9M

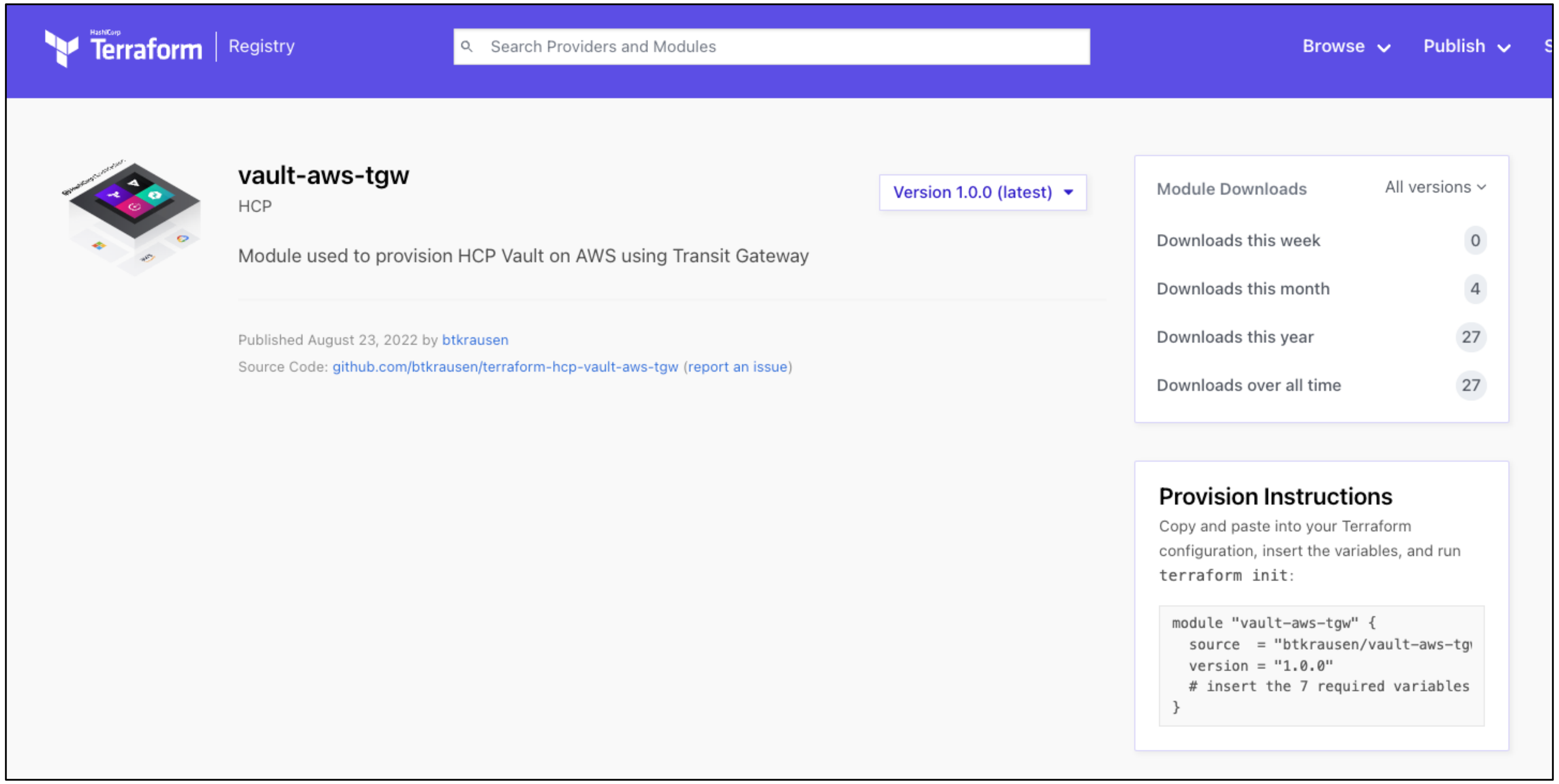
Provision Instructions

Copy and paste into your Terraform configuration, insert the variables, and run `terraform init`:

```
module "s3-bucket" {
  source  = "terraform-aws-modules/
  version = "3.4.0"
}
```

Modules are reusable units of Terraform code that hide unnecessary complexity from the user. This one creates a standard VPC configuration with only 8 variables.

What is a Terraform Module



Terraform | Registry

Search Providers and Modules

Browse Publish

vault-aws-tgw

HCP

Version 1.0.0 (latest)

Module used to provision HCP Vault on AWS using Transit Gateway

Published August 23, 2022 by btkrausen

Source Code: github.com/btkrausen/terraform-hcp-vault-aws-tgw (report an issue)

Module Downloads	All versions
Downloads this week	0
Downloads this month	4
Downloads this year	27
Downloads over all time	27

Provision Instructions

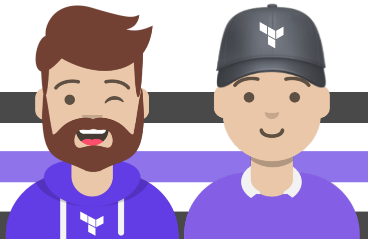
Copy and paste into your Terraform configuration, insert the variables, and run `terraform init`:

```
module "vault-aws-tgw" {
  source  = "btkrausen/vault-aws-tg
  version = "1.0.0"
  # insert the 7 required variables
}
```

Managing Modules

Common Questions about Modules

- How do you manage dozens or hundreds of modules?
- How do you manage module versions?
- Where do people within the organization go to consume the correct module?



Managing Modules

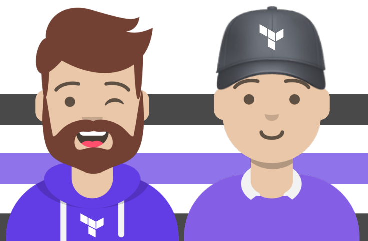
How are Terraform Modules Configured?

Creating Terraform Modules in 3 easy steps:

1. Write some Terraform code, configuring inputs and outputs.
2. Store the Terraform code somewhere your workstation can access it.
3. Reference your modules by file path or source URL.

Sounds easy right?

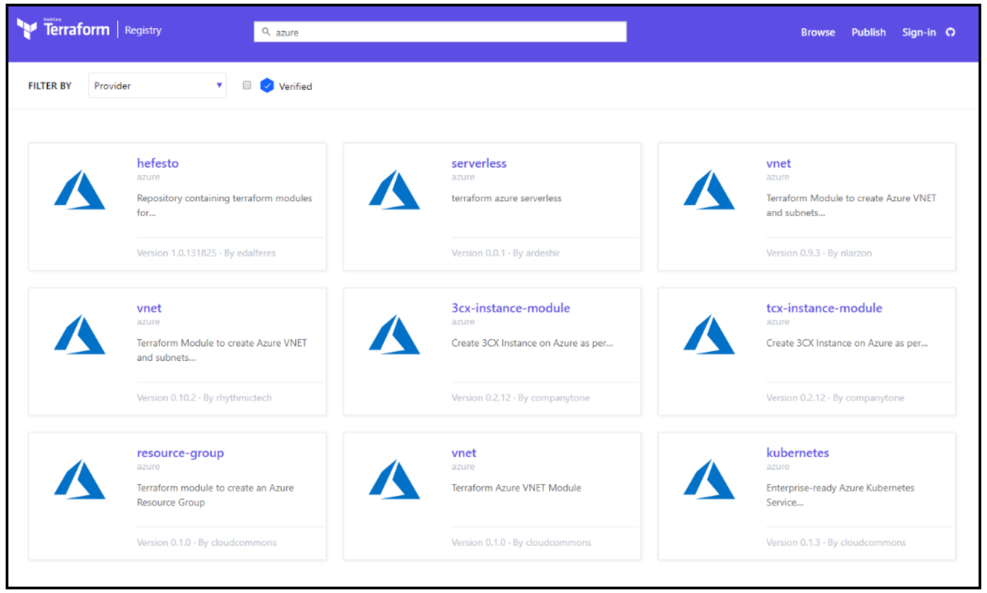
What if you had to manage dozens or hundreds of modules, with different versions of each?



Private Registry: Modules

Terraform modules are reusable packages of Terraform code that you can use to build your infrastructure.

Terraform Cloud includes a Private Module Registry where you can store, version, and distribute modules to your organizations and teams.



hefesto
azure
Repository containing terraform modules for...
Version 1.0.131825 · By edalferes

serverless
azure
terraform azure serverless
Version 0.0.1 · By ardeshir

vnet
azure
Terraform Module to create Azure VNET and subnets...
Version 0.9.3 · By nlarzon

vnet
azure
Terraform Module to create Azure VNET and subnets...
Version 0.10.2 · By rhythmictech

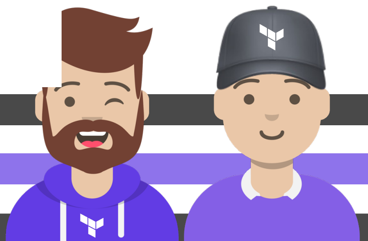
3cx-instance-module
azure
Create 3CX Instance on Azure as per...
Version 0.2.12 · By companytone

tcx-instance-module
azure
Create 3CX Instance on Azure as per...
Version 0.2.12 · By companytone

resource-group
azure
Terraform module to create an Azure Resource Group
Version 0.1.0 · By cloudcommons

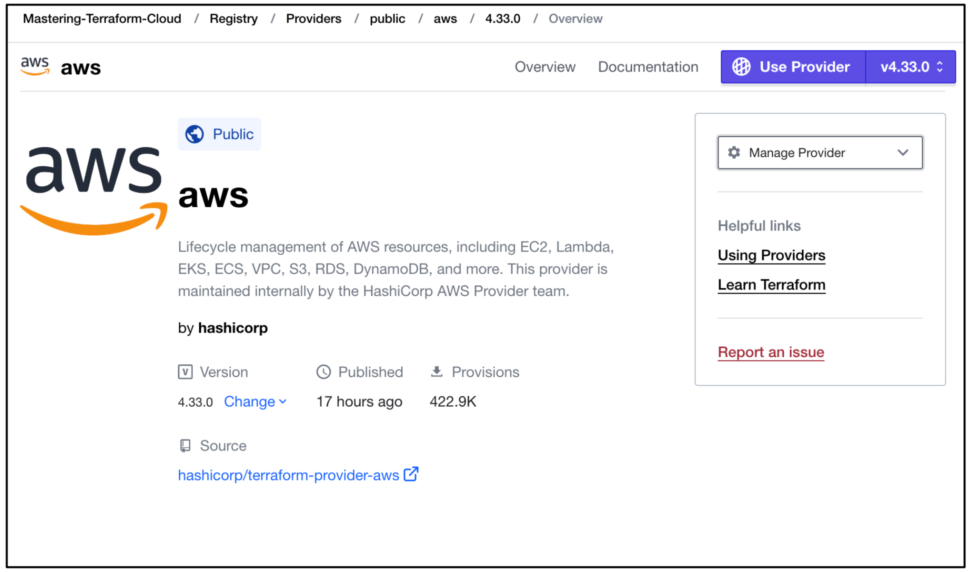
vnet
azure
Terraform Azure VNET Module
Version 0.1.0 · By cloudcommons

kubernetes
azure
Enterprise-ready Azure Kubernetes Service...
Version 0.1.3 · By cloudcommons



Publishing to the Private Registry

Search the Public Registry and add **providers** and/or **modules** to your organization's infrastructure for use in your workspaces



Mastering-Terraform-Cloud / Registry / Providers / public / aws / 4.33.0 / Overview

aws aws — Overview — Documentation — Use Provider — v4.33.0

Public

aws

Lifecycle management of AWS resources, including EC2, Lambda, EKS, ECS, VPC, S3, RDS, DynamoDB, and more. This provider is maintained internally by the HashiCorp AWS Provider team.

by **hashicorp**

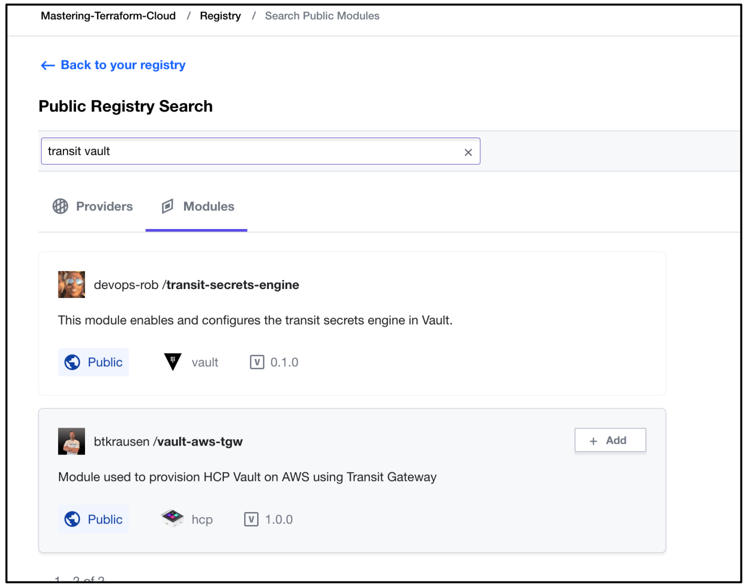
Version	Published	Provisions
4.33.0 Change	17 hours ago	422.9K

Source
hashicorp/terraform-provider-aws

Manage Provider

Helpful links
Using Providers
Learn Terraform

Report an issue



Mastering-Terraform-Cloud / Registry / Search Public Modules

← Back to your registry

Public Registry Search

transit vault

Providers — Modules

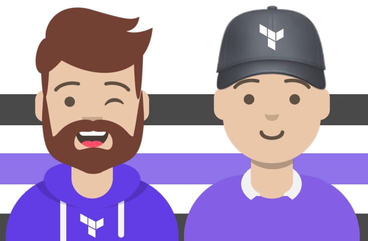
devops-rob /**transit-secrets-engine**
This module enables and configures the transit secrets engine in Vault.
Public — vault — 0.1.0

btkrausen /**vault-aws-tgw** — + Add
Module used to provision HCP Vault on AWS using Transit Gateway
Public — hcp — 1.0.0

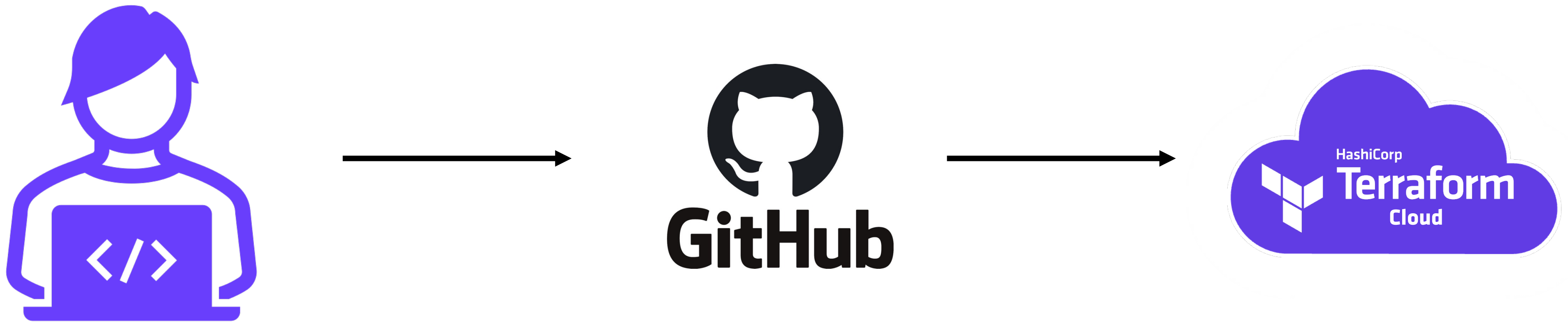
1 – 2 of 2

Providers

Modules



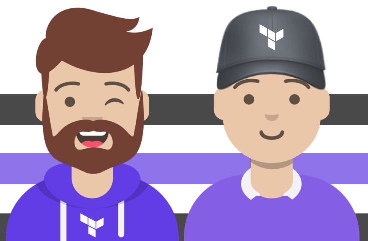
Publishing Modules to Registry



Develop a Module

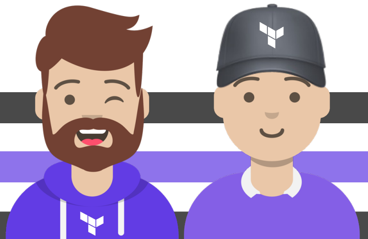
Code Repo
(private)

Publish and Share to Private Registry



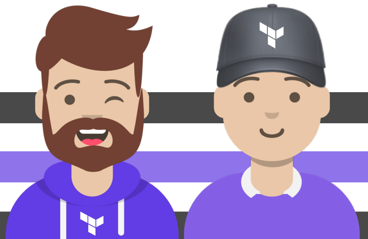
Publishing to the Public Registry

- **GitHub:** The module must be on GitHub and must be a public repo. This is only a requirement for the public registry. If you're using a private registry, you may ignore this requirement.
- **Named *`terraform-<PROVIDER>-<NAME>`*:** Module repositories must use this three-part name format, where <NAME> reflects the type of infrastructure the module manages and <PROVIDER> is the main provider where it creates that infrastructure. The <NAME> segment can contain additional hyphens. Examples: terraform-google-vault or terraform-aws-ec2-instance.
- **Repository description:** The GitHub repository description is used to populate the short description of the module. This should be a simple one sentence description of the module.
- **Standard module structure:** The module must adhere to the standard module structure. This allows the registry to inspect your module and generate documentation, track resource usage, parse submodules and examples, and more.
- **x.y.z tags for releases:** The registry uses tags to identify module versions. Release tag names must be a semantic version, which can optionally be prefixed with a v. For example, v1.0.4 and 0.9.2. To publish a module initially, at least one release tag must be present. Tags that don't look like version numbers are ignored.

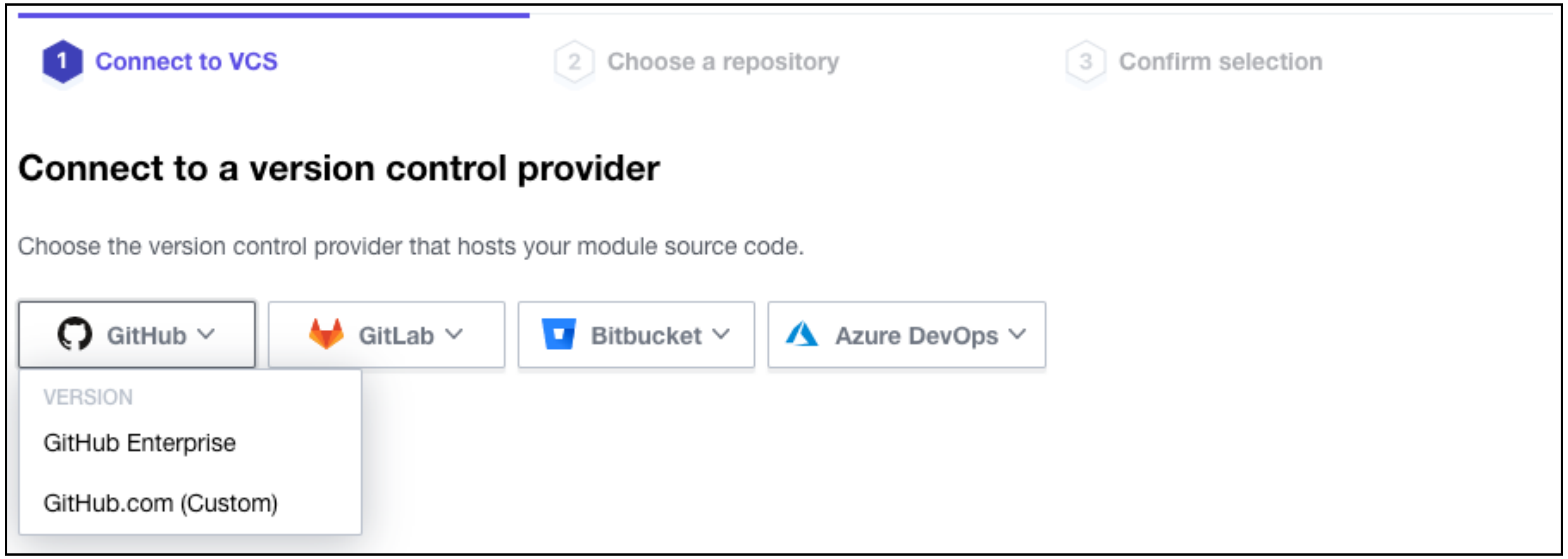


Publishing to the Private Registry

- GitHub
  Terraform Cloud supported VCS Provider / Private Repos
- Named *terraform-<PROVIDER>-<NAME>*
- Repository description
- Standard module structure
- x.y.z tags for releases



Private Registry and Version Control

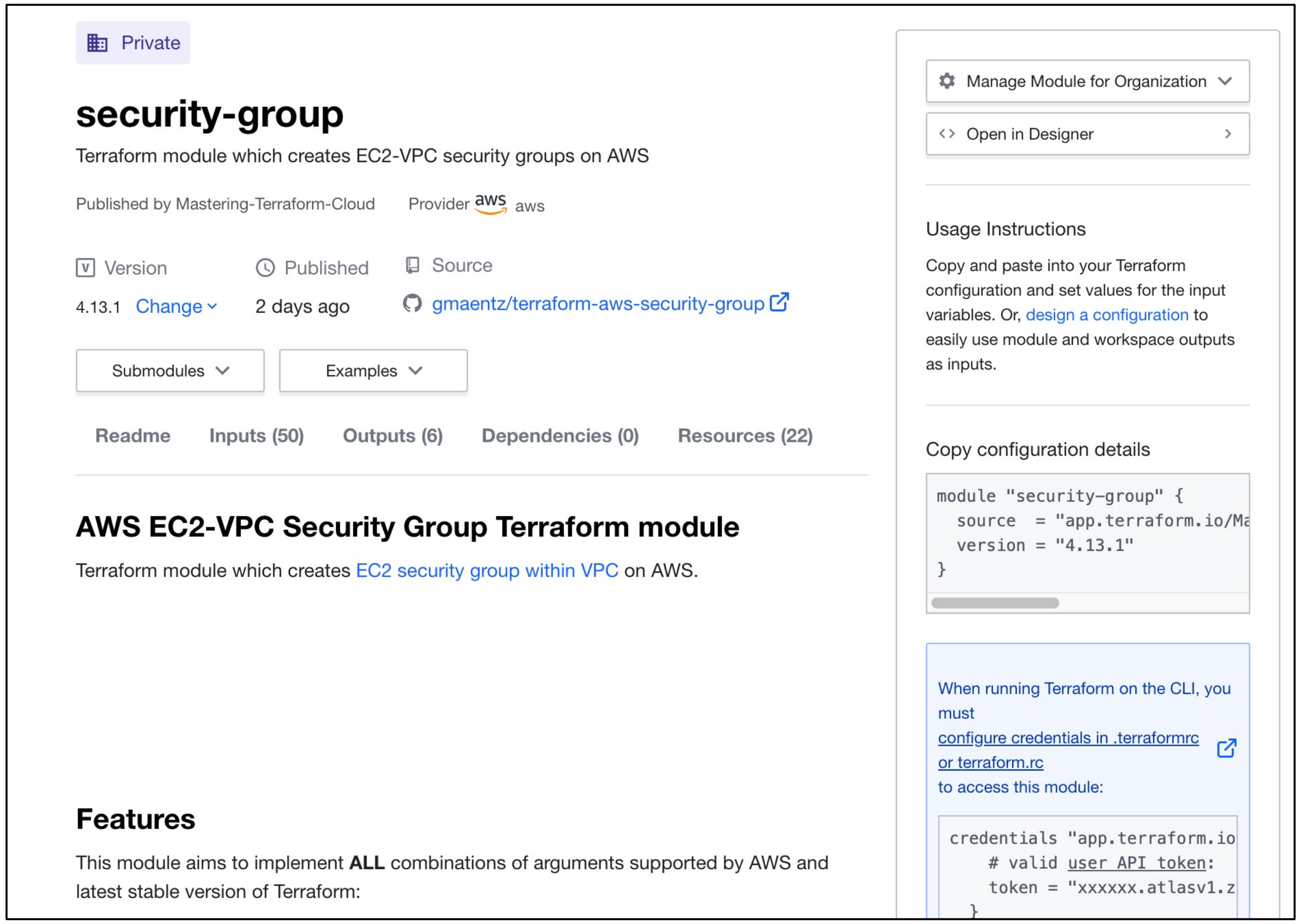


1 Connect to VCS
2 Choose a repository
3 Confirm selection

Connect to a version control provider

Choose the version control provider that hosts your module source code.

Using the Private Registry


Private

security-group

Terraform module which creates EC2-VPC security groups on AWS

Published by Mastering-Terraform-Cloud   Provider aws aws

Version	Published	Source
4.13.1 Change	2 days ago	gmaentz/terraform-aws-security-group

AWS EC2-VPC Security Group Terraform module

Terraform module which creates EC2 security group within VPC on AWS.

Features

This module aims to implement **ALL** combinations of arguments supported by AWS and latest stable version of Terraform:

Usage Instructions

Copy and paste into your Terraform configuration and set values for the input variables. Or, design a configuration to easily use module and workspace outputs as inputs.

Copy configuration details

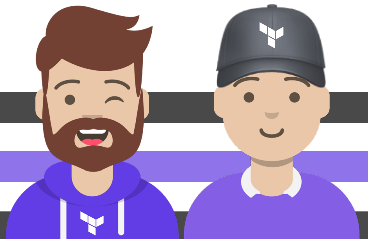
```
module "security-group" {
  source  = "app.terraform.io/Ma
  version = "4.13.1"
}
```

When running Terraform on the CLI, you must configure credentials in .terraformrc or terraform.rc to access this module:

```
credentials "app.terraform.io
    # valid user API token:
    token = "xxxxxx.atlasv1.z
  }
```

Releasing New Versions

- The Terraform Registry uses tags to detect releases.
- Tag names must be a valid semantic version, optionally prefixed with a v. Example of valid tags are: v1.0.1 and 0.9.4. To publish a new module, you must already have at least one tag created.
- To release a new version, create and push a new tag with the proper format. The webhook will notify the registry of the new version and it will appear on the registry usually in less than a minute.



END OF SECTION