Lab: Terraform Cloud - Private Module Registry

Terraform Cloud's Private Module Registry allows you to store and version Terraform modules which are re-usable snippets of Terraform code. It is very similar to the Terraform Public Module registry including support for module versioning along with a searchable and filterable list of available modules for quickly deploying common infrastructure configurations.

The Terraform Private Module Registry is an index of private modules that you don't want to share with publicly. This lab demonstrates how to register a module with your Private Module Registry then reference it in a workspace.

- Task 1: Move terraform module code to a private GitHub repository
- Task 2: Publish module to Private Module Registry
- Task 3: Review module block reference to Private Module Registry

Task 1: Move terraform module code to a private GitHub repository

Instead of writing a terraform module from scratch we will copy and existing S3 module from the public Terraform registry. Visit this URL to view the AWS S3 module:

https://github.com/terraform-aws-modules/terraform-aws-s3-bucket

Step 1.1 Fork the Module Repository to your GitHub account

You are going to fork the following repository into your own GitHub account:

- https://github.com/terraform-aws-modules/terraform-aws-s3-bucket

This repository represents a module that can be developed and versioned independently. Note the **Source:** link that points at the github repository for this module. Click on the Source URL and create your own fork of this repository with the **Fork** button.

Task 2: Publish module to Private Module Registry

We need to add this repository into the Private Module Registry. Navigate back to Terraform Cloud and click the "Modules" menu at the top of the page. From there click the "+ Add Module" button.

Add Module

This module will be created under the current organization, **example-org-6cde13**. Modules can be added from all supported VCS providers.

Connect to VCS | 2 Choose a repository | 3 Confirm selection

Choose a repository

Choose the repository that hosts your module source code. We'll watch this for commits and tags. The format of your repository name should be `terraform-<PROVIDER>-<NAME>`.

12 repositories — Filter: acme-corp/infrastructure

- gmaentz/terraform-aws-s3-bucket
- gmaentz/terraform-aws-server
- gmaentz/terraform-example-module

Select the S3 repository you forked earlier.

Add Module

This module will be created under the current organization, **example-org-6cde13**. Modules can be added from all supported VCS providers.

Connect to VCS | Choose a repository | 3 Confirm selection

Confirm selection

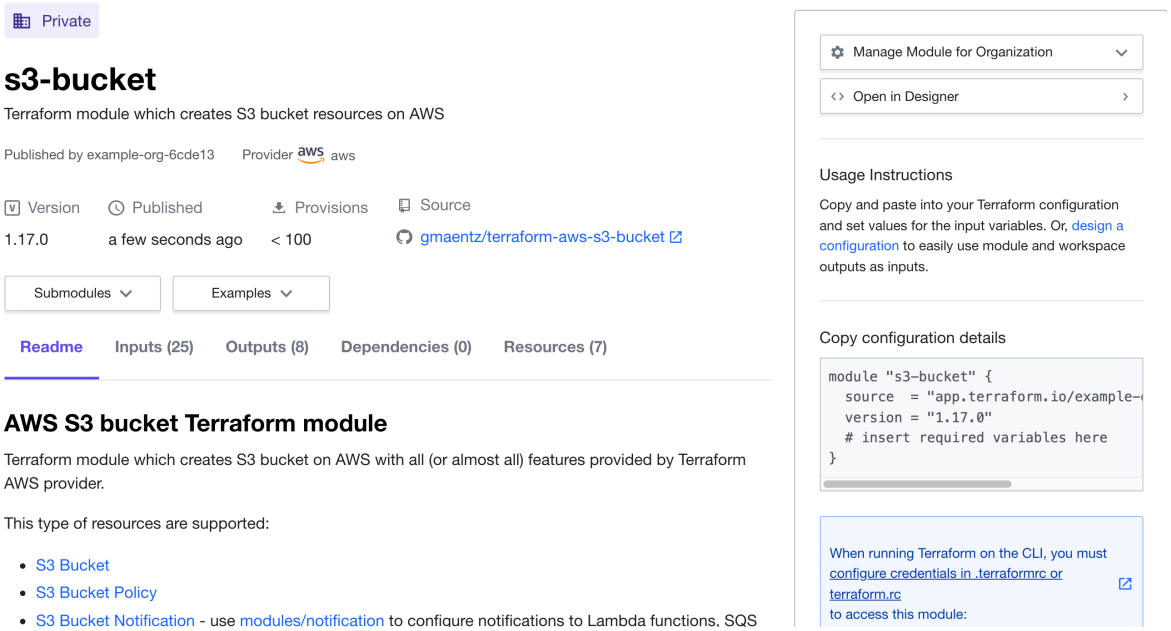
Provider	GitHub (GitHub.com)
Repository	gmaentz/terraform-aws-s3-bucket

Publish module | Cancel

> Note: You will see your github user name since you forked this repo.

Click “Publish Module”.

This will query the repository for necessary files and tags used for versioning.



Task 3: Review module block reference to Private Module Registry

Now that the module has been published to the Private Module registry, we can utilize it within any module block in our terraform configuration. Note the `source` and `version` of the newly published private module.

```
module "s3-bucket" {
  source  = "app.terraform.io/example-org-6cde13/s3-bucket/aws"
  version = "1.17.0"
  # insert required variables here
}
```

Resources

- Private Registries
- Publishing Modules