# Terraform Modules Publishing to Terraform registry

[Terraform Modules Publishing to Terraform registry 1](#_Toc125980571)

[Prerequisites 1](#_Toc125980572)

[Terraform cli 1](#_Toc125980573)

[Step 1 Terraform Module 1](#_Toc125980574)

[Step 2 OAuth Verification for Git/Bitbucket 3](#_Toc125980575)

[Git 3](#_Toc125980576)

[Bit Bucket 5](#_Toc125980577)

[Step 3 On Terraform Cloud, Set up Your Provider 7](#_Toc125980578)

[Step 4 Publish Module 8](#_Toc125980579)

[Publish Terraform Modules to registry via API 10](#_Toc125980580)

[Create an OAuth Client - API 10](#_Toc125980581)

[List modules - API 13](#_Toc125980582)

This Document illustrate how to create Terraform modules and publish them to Registry

## Prerequisites

1. GitHub/Bitbucket Account
2. Access to create OAuth (GIT: client Id, Secretes or Bitbucket: Key, Secret)
3. Terraform Registry Account
4. Setup Terraform cli in your local

## Terraform cli

Linux/Mac:

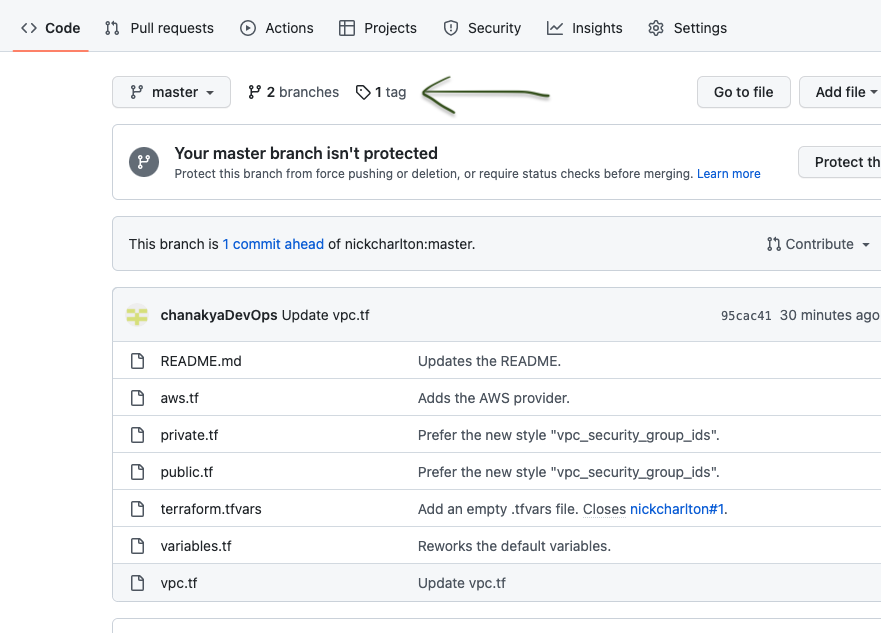
Windows:

### Step 1 Terraform Module

Terraform module is a set of Terraform configuration files (\*.tf) in a directory. Write your first terraform code following [module-create](https://developer.hashicorp.com/terraform/tutorials/modules/module-create)

*Note:* To publish modules to registry, TF module names must have the format **terraform-<PROVIDER>-<NAME>, where <NAME>**can contain extra hyphens.

* Create a repository in GitHub/Bitbucket Account and upload the terraform module to repository
* Tag a release, click on the tag icon, and create a new tag e.g v1.0.0 as below and then publish release



Graphical user interface, text, application, email

Description automatically generated

* As we have versioned terraform module, now we need to add VCS connection to our Terraform cloud organization.

## Step 2 OAuth Verification for Git/Bitbucket

## Git

* Private Module Registry requires OAuth verification
* Login to Git and navigate to <https://github.com/settings/applications/new>

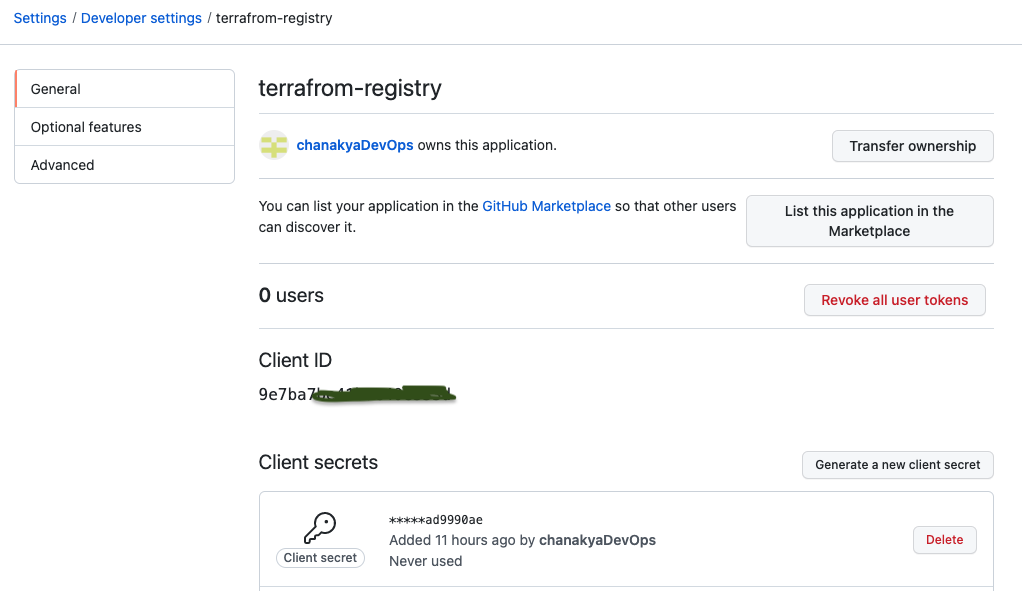
Graphical user interface, text, application, email

Description automatically generated

* Fill out the text fields as follows:

| **Field name** | **Value** |
| --- | --- |
| Application Name | Terraform Cloud (<YOUR ORGANIZATION NAME>) |
| Homepage URL | https://app.terraform.io (or the URL of your Terraform Enterprise instance) |
| Application Description | Any description of your choice. |
| Authorization callback URL | https://app.terraform.io/<YOUR CALLBACK URL> |

* Click the "Register application" button, which creates the application and takes you to its page.
* After registration

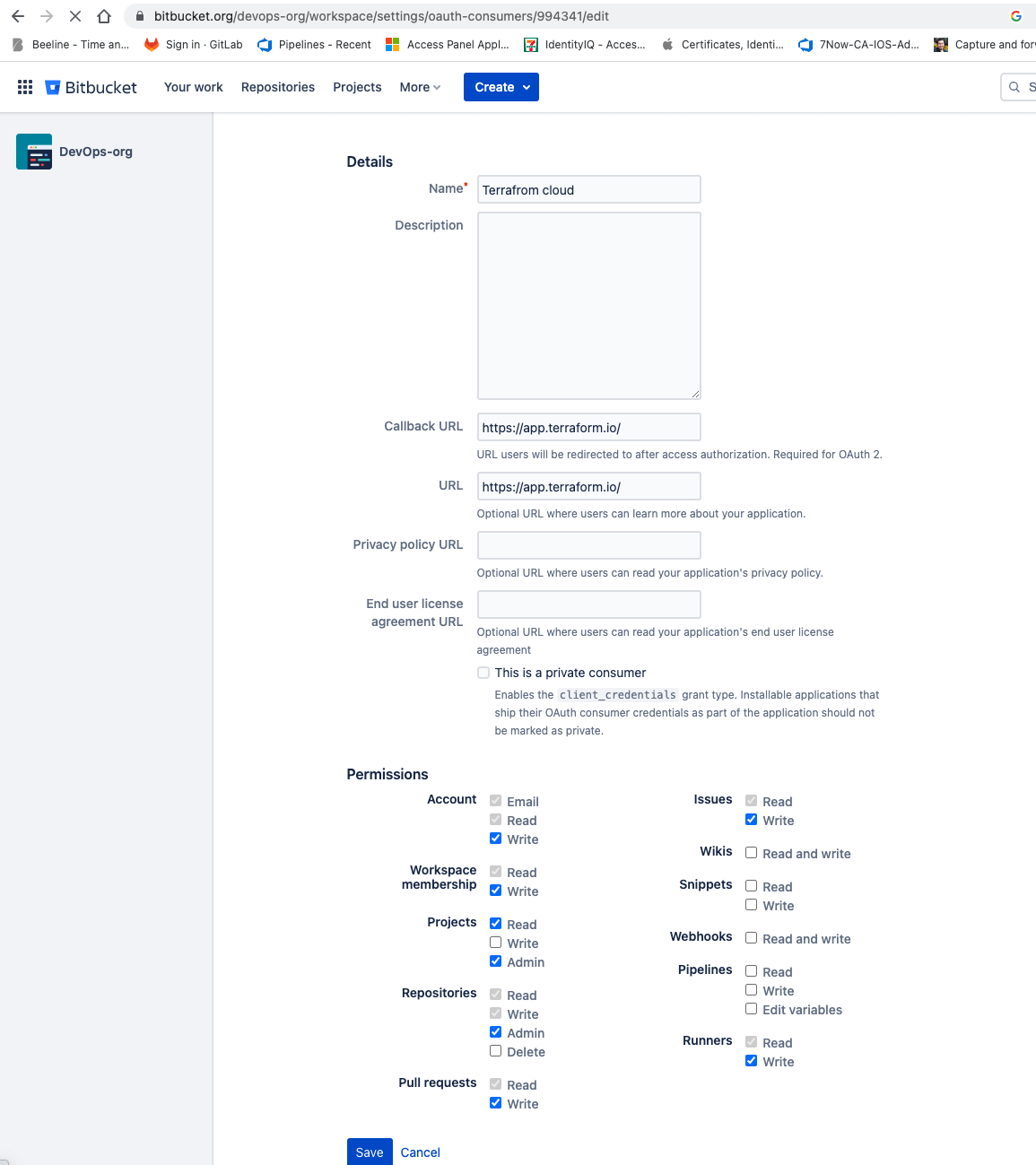


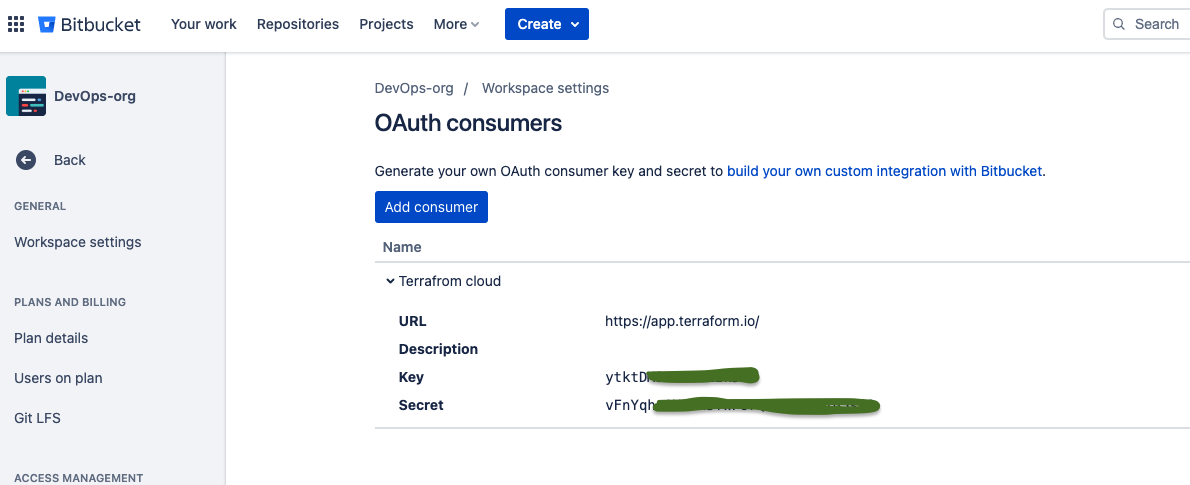
## Bit Bucket

* Click your profile picture and choose the workspace you want to access.
* Click "Settings".
* Click "OAuth consumers," which is in the "Apps and Features" section.
* On the OAuth settings page, click the "Add consumer" button

Ensure that the "This is a private consumer" option is checked. Then, activate the following permissions checkboxes:

| **Permission type** | **Permission level** |
| --- | --- |
| Account | Write |
| Repositories | Admin |
| Pull requests | Write |
| Webhooks | Read and write |

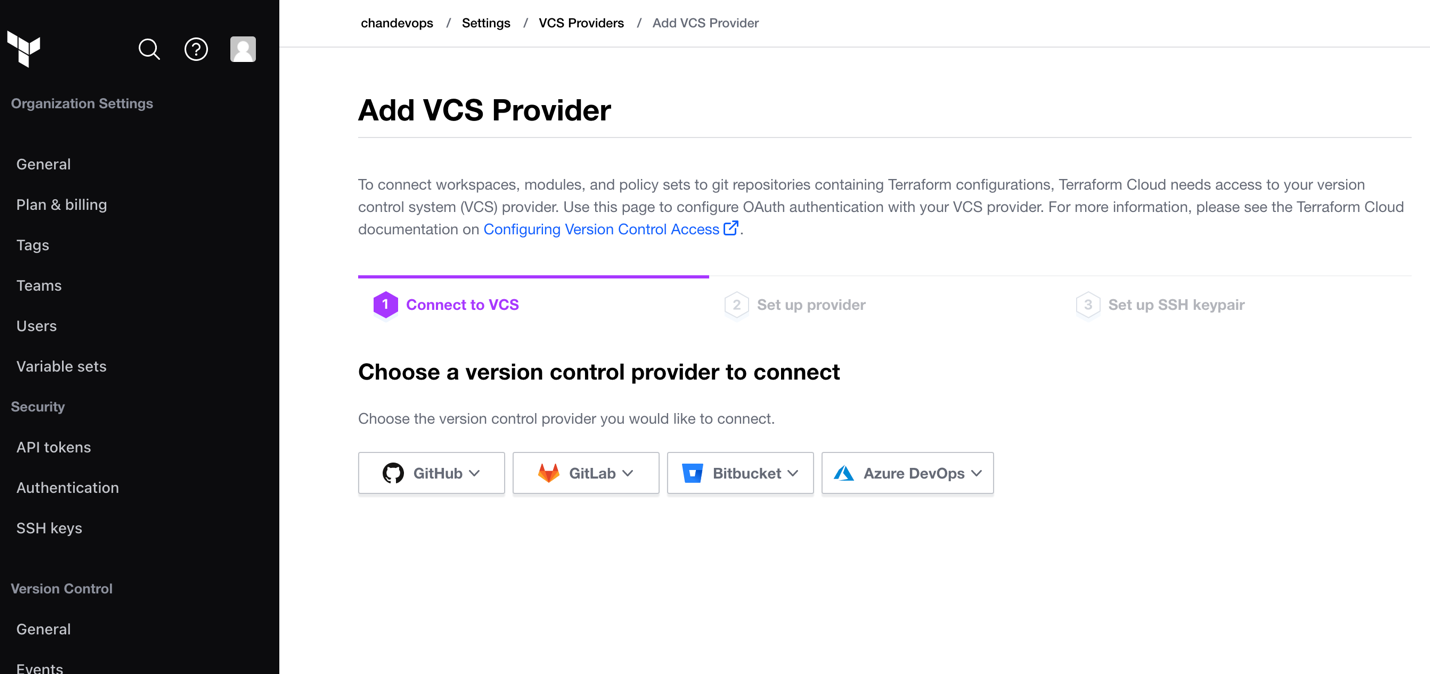




## 

## Step 3 On Terraform Cloud, Set up Your Provider

* Choose a version control provider to connect
* Go to your organization's settings and then click **Providers**. The **VCS Providers** page appears.
* Click **Add VCS Provider**. The **VCS Providers** page appears.



* Select the provider GitHub or Bitbucket
* Enter the **Client ID** and **Client Secret** from the above step got GitHub, and **Name** for the new VCS connection.
* Enter the Key and Secret from the above step got Bitbucket, and **Name** for the new VCS connection.
* Click "Connect and continue." This takes you to a page on GitHub.com, asking whether you want to authorize the app.
* Click the green "Authorize <USER>" button

## Step 4 Publish Module

* To publish your module into your “private module registry”. First Login to terraform cloud account. Select “Registry” → “Modules” → “Publish” from the upper right corner.

Graphical user interface, text, application, email

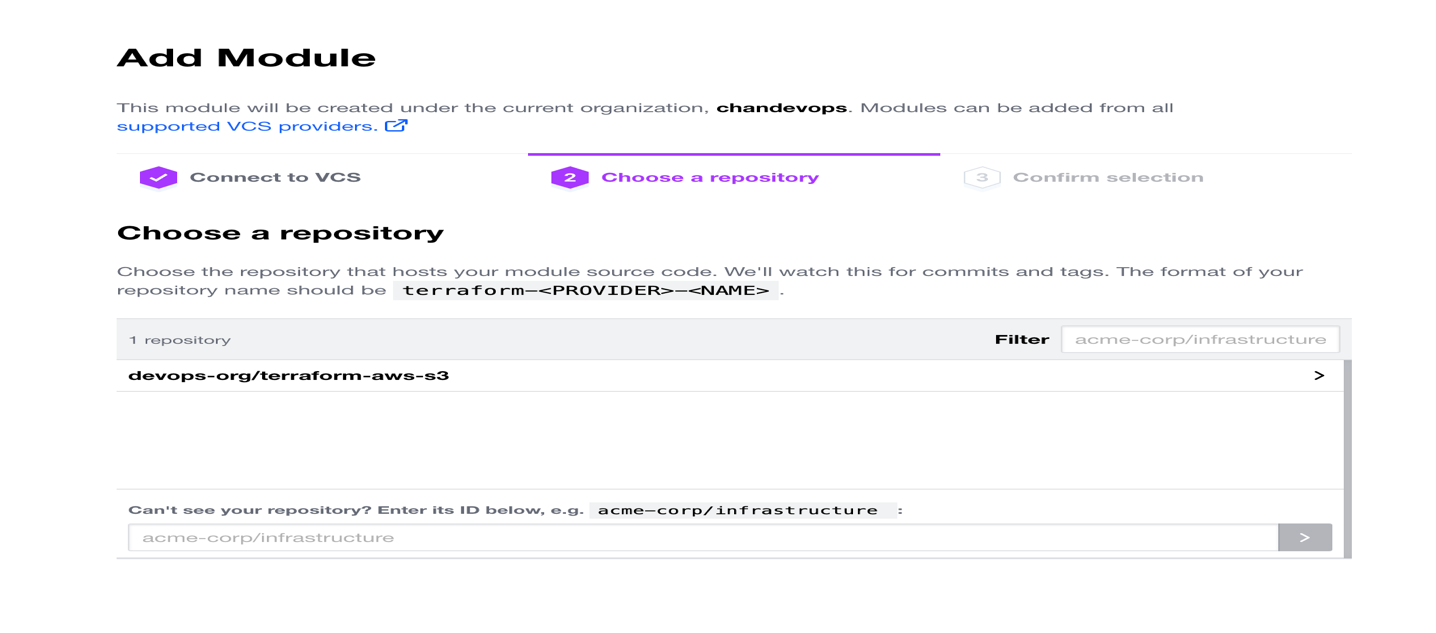
Description automatically generated

* Select Module

Graphical user interface, text, application, Teams

Description automatically generated

* Choose repository



* Publish Module

Graphical user interface, application, Teams

Description automatically generated

* Select the module and click the “Publish module” button.
* Click on the Module just added and read Usage instructions

Graphical user interface, text, application, email

Description automatically generated

## Publish Terraform Modules to registry via API

Once the modules are pushed to git repository with release tag, Pipeline should create an OAuth Token

(OAuth Client represents the connection between Git and a VCS provider.)

## Create an OAuth Client - API

POST /organizations/:organization\_name/oauth-clients

i.e.

POST [/github.com/pipeline-automation/oauth-clients](https://github.com/pipeline-automation/oauth-clients)

POST call allows you to create a VCS connection between an organization and a VCS provider (GitHub)

**Note:** This endpoint does not currently support creation of a Bitbucket Cloud, Bitbucket Server, or Azure DevOps Services OAuth Client.

Creating the OAuth Token is one time activity for organization (git, bitbucket), to allow permission on VCS Provider – we can do it manually for the above unsupported services.

Which cover Manual [Step 2 OAuth Verification for Git/Bitbucket](#_Step_2_OAuth) and [Step 3 On Terraform Cloud, Set up Your Provider](#_Step_3_On)

Sample Payload

{

"data": {

"type": "oauth-clients",

"attributes": {

"service-provider": "github",

"http-url": "https://github.com",

"api-url": "https://api.github.com",

"oauth-token-string": "4306823352f0009d0ed81f1b654ac17a"

}

}

}

Sample Request

curl \

--header "Authorization: Bearer $TOKEN" \

--header "Content-Type: application/vnd.api+json" \

--request POST \

--data @payload.json \

https://app.terraform.io/api/v2/organizations/my-organization/oauth-clients

curl \

--header "Authorization: Bearer $TOKEN" \

--header "Content-Type: application/vnd.api+json" \

--request POST \

--data @payload.json \

<https://app.terraform.io/api/v2/organizations/my-organization/oauth-clients>

Sample Response

{

"data": {

"id": "oc-XKFwG6ggfA9n7t1K",

"type": "oauth-clients",

"attributes": {

"created-at": "2018-04-16T20:42:53.771Z",

"callback-url": "https://app.terraform.io/auth/35936d44-842c-4ddd-b4d4-7c741383dc3a/callback",

"connect-path": "/auth/35936d44-842c-4ddd-b4d4-7c741383dc3a?organization\_id=1",

"service-provider": "github",

"service-provider-display-name": "GitHub",

"name": null,

"http-url": "https://github.com",

"api-url": "https://api.github.com",

"key": null,

"rsa-public-key": null

},

"relationships": {

"organization": {

"data": {

"id": "my-organization",

"type": "organizations"

},

"links": {

"related": "/api/v2/organizations/my-organization"

}

},

"oauth-tokens": {

"data": [],

"links": {

"related": "/api/v2/oauth-tokens/ot-KaeqH4cy72VPXFQT"

}

}

}

}

}

## List modules - API

Sample Request

curl 'https://registry.terraform.io/v1/modules?limit=2&verified=true'

Sample Response

{

"meta": {

"limit": 2,

"current\_offset": 0,

"next\_offset": 2,

"next\_url": "/v1/modules?limit=2&offset=2&verified=true"

},

"modules": [

{

"id": "terraform-aws-modules/vpc/aws/1.5.1",

"owner": "",

"namespace": "terraform-aws-modules",

"name": "vpc",

"version": "1.5.1",

"provider": "aws",

"description": "Terraform module which creates VPC resources on AWS",

"source": "https://github.com/terraform-aws-modules/terraform-aws-vpc",

"published\_at": "2017-11-23T10:48:09.400166Z",

"downloads": 29714,

"verified": true

}

]

}

List Available Version for Specific Modules:

 returning the available versions for a given fully qualified module.

| **Method** | **Path** | **Produces** |
| --- | --- | --- |
| GET | <base\_url>/:namespace/:name/:provider/versions | application/json |

**Parameters**

* [namespace](https://developer.hashicorp.com/terraform/registry/api-docs#namespace-2) (string: <required>) - The user or organization the module is owned by. This is required and is specified as part of the URL path.
* [name](https://developer.hashicorp.com/terraform/registry/api-docs#name) (string: <required>) - The name of the module. This is required and is specified as part of the URL path.
* [provider](https://developer.hashicorp.com/terraform/registry/api-docs#provider-2) (string: <required>) - The name of the provider. This is required and is specified as part of the URL path.

curl https://registry.terraform.io/v1/modules/hashicorp/consul/aws/versions

curl https://registry.terraform.io/v1/modules/hashicorp/consul/aws/versions

Sample Request

curl \

https://registry.terraform.io/v1/modules/hashicorp/consul/aws/0.0.1

You can check for version in payload in as "version": "0.0.1",