Lab: Configure Terraform Provider Registry

Table of Contents

[Lab Introduction 1](#_Toc99109173)

[Lab Information Sheets 1](#_Toc99109174)

[Getting started with Cisco Intersight and Terraform 2](#_Toc99109175)

[Prerequisites 2](#_Toc99109176)

[Task 1: Generate Cisco Intersight API-Key 2](#_Toc99109177)

[Task 2: Configure Policies 3](#_Toc99109178)

[Task 3: Plan the resource deployment 7](#_Toc99109179)

[Task 4: Apply the resource deployment 8](#_Toc99109180)

[Task 5: Update the Policy using Terraform 8](#_Toc99109181)

[Task 6: Execute a Rollback 10](#_Toc99109182)

Lab Introduction

Lab Information Sheets

Learners will need the access and additional information provided by the instructor to complete the labs for this course. Learners may optionally bookmark pages in the Cisco eReader for reference while observing the presentation and performing the labs.

Please refer to the Lab Information Sheet for lab access procedures, IP addresses, Pod#, and credentials.

* Use the Linux Student Desktop to complete this lab.

**Note:** Due to the rapidly updated nature of Software as a Service (SaaS), some screen layouts may differ from the example illustrations shown.

# Getting started with Cisco Intersight and Terraform

Terraform is an open-source infrastructure as code platform that allows you to provision infrastructure resources via providers. Codification enables version control and automation, reducing human error and increasing productivity.

Prerequisites

Prior to starting this learning lab, it would be helpful to have an understanding of basic Cisco Intersight functionality.

An understanding of REST API and a basic understanding of Terraform would also be helpful.

A workstation with Terraform CLI installed. You can download the latest Terraform CLI from [https://www.terraform.io/downloads](about:blank)

Requirements:

* Ubuntu
* Git – If not installed run the command “sudo apt install git”

## Task 1: Generate Cisco Intersight API-Key

Connect to your student Linux host. Terraform should already installed there. If not, please install.

For ubuntu: **snap install terraform --classic**

Get the Rest API Key and Secret Key from Cisco Intersight

1. If you already generated a key in a previous lab and have the secret.txt file you can skip to task #2.
2. Log on to Intersight
   1. Click the Service Selector
   2. Choose **System**

Graphical user interface, application

Description automatically generated

1. Under Settings Select API Keys (towards bottom of list)

Graphical user interface, application, Teams

Description automatically generated

1. Click Generate API Key

Graphical user interface, text, application

Description automatically generated

On Generate API Key screen, Enter following

Description: POD**X** (Where **X** as POD number, for example: POD01)

Select **API Key for OpenAPI schema version 2.**

Click **Generate.**

Graphical user interface, text, application, chat or text message

Description automatically generated

1. Secret Key can be viewed or downloaded only one time. The API Key can be viewed any time. Click the **Download** and secret key will be downloaded as SecretKey.txt.

You may also want to copy and paste the API Key into Linux Nano for easy access later.

**Note:** This SecretKey.txt file and the API Key ID will be used in the upcoming Labs

1. We will create two file in Linux one called secret.txt and one called apikey.txt
2. Type **nano secretkey.txt** and paste the secret key. Control+X to exit and Save.
3. Type **nano apikey.txt** and paste the api key. Control+X to exit and Save.
4. You can close the API key dialog box

## Task 2: Configure Policies

1. Connect to the remote NterOne Lab Linux machine
2. Open Terminal

Background pattern

Description automatically generated

1. Run the following command to clone the Git repository.

**git clone https://github.com/intersightlab/imm2**

root@ICO:~# git clone https://github.com/intersightlab/intersight-terraform

Cloning into 'intersight-terraform'...

remote: Enumerating objects: 8, done.

remote: Counting objects: 100% (8/8), done.

remote: Compressing objects: 100% (8/8), done.

remote: Total 8 (delta 2), reused 0 (delta 0), pack-reused 0

Receiving objects: 100% (8/8), done.

Resolving deltas: 100% (2/2), done.

root@ICO:~#

Type the following command to change your directory to intersight-terraform.

**cd intersight-terraform**

root@ICO:~# cd intersight-terraform

root@ICO:~/intersight-terraform#

1. Note that the folder contains 3 terraform files: main.tf, variables.tf and configure\_ntp\_policy.tf

root@ICO:~/intersight-terraform# ls -Fla

total 24

drwxr-xr-x 3 root root 4096 Apr 5 11:11 ./

drwx------ 7 root root 4096 Apr 5 11:11 ../

-rw-r--r-- 1 root root 652 Apr 5 11:11 config\_ntp\_policy.tf

drwxr-xr-x 8 root root 4096 Apr 5 11:11 .git/

-rw-r--r-- 1 root root 312 Apr 5 11:11 main.tf

-rw-r--r-- 1 root root 359 Apr 5 11:11 variables.tf

root@ICO:~/intersight-terraform#

m

For the next step, note that here are two keys:

1. The Secret Key which is stored in the secretkey.txt file, and has been configured.
2. The API key is stored in the apikey.txt
   1. For the file in the next step, you will need to open up apikey.txt and copy the API key.
   2. Then paste the API key where you see "PUT\_YOUR\_API\_KEY\_HERE"
3. Review and edit **variables.tf**

Open **variables.tf** using Linux Nano editor, by typing the command below in the bash shell.

**nano variables.tf**

There are two keys:

1. The Secret Key which is stored in the secretkey.txt file, and has been configured.
2. The API key is stored in the apikey.txt
   1. For the file below, you will need to open up apikey.txt and copy the API key.
   2. The paste the API key where you see "PUT\_YOUR\_API\_KEY\_HERE"

This file contains all the variables used by Terraform. The file should be as follows:

# Location of the Intersight API Secretkey file

variable "secretkey" {

default = "/root/secretkey.txt"

}

# Intersight API-Key-ID

variable "apikey" {

default = "PUT\_YOUR\_API\_KEY\_HERE"

}

variable "endpoint" {

default = "https://www.intersight.com"

}

# Replace X with your POD number

variable "organization\_name" {

default = "default"

}

Change the following items:

* **apikey:** Update the API key ID previously copied from Cisco Intersight

**Note:** This is the API Key which was copied from Cisco Intersight in Task 1.

Control+X and Save.

1. Review **main.tf**

Go back to the The bash shell command line. Open.**main.tf** using Linux Nano editor, by typing the command below in the bash shell.

**cat main.tf**

1. Review main.tf and note the apikey and secretkey will be placed by the values we provided.

root@ICO:~/intersight-terraform# cat main.tf

# Intersight Provider Information

terraform {

required\_providers {

intersight = {

source = "ciscodevnet/intersight"

version = ">= 1.0.28"

}

}

}

# Intersight Authentication details

provider "intersight" {

apikey = var.apikey

secretkey = var.secretkey

endpoint = var.endpoint

}

root@ICO:~/intersight-terraform#

1. Review **config\_ntp\_policy.tf**

Go back to the The bash shell command line. Openmian.**main.tf** using Linux Nano editor, by typing the command below in the bash shell.

**nano config\_ntp\_plicy.tf**

root@ICO:~/intersight-terraform# cat config\_ntp\_policy.tf

#Get the Organization Moid by Organization Name

data "intersight\_organization\_organization" "organization\_moid" {

name = var.organization\_name

}

output "organization\_moid" {

value = data.intersight\_organization\_organization.organization\_moid.results[0].moid

}

# NTP Policy details

resource "intersight\_ntp\_policy" "NTP" {

name = "${var.organization\_name}-ntp-policy"

description = "NTP policy from Terraform"

enabled = true

ntp\_servers = [

"pool.ntp.org"

]

organization {

object\_type = "organization.Organization"

moid = data.intersight\_organization\_organization.organization\_moid.results[0].moid

}

}

root@ICO:~/intersight-terraform#

.

This file is used to create the NTP Policy in Cisco Intersight. Using *Data sources* [intersight\_organization\_organization], we query the Organizations in Cisco Intersight by Organization name and get its Moid (Managed Object ID). Since we are using Multiple Organization, we need to create a NTP policy under a specific Organization. We are passing NTP policy details with Organization Moid in this file to create a NTP policy.

1. Go back to your open Command Line

The terraform init command will initialize the working directory containing Terraform configuration files and install any required plugins.

Note: The terraform init command is safe to run multiple times, to bring the working directory up to date with changes in the configuration.

Type the following command:

**terraform init**

Text

Description automatically generated

Now the backend and provider plugins are initialized.

Terraform validate Operation

1. Run terraform validation

The terraform validate command is used to validate the syntax of the terraform files and the configuration. Terraform performs a syntax check on all the terraform files in the directory and will display an error if any of the files are not valid.

Type the following command.

**terraform validate**

Text

Description automatically generated

Now the configurations are validated.

## Task 3: Plan the resource deployment

Terraform plan Operation

1. Run terraform plan

The terraform plan command creates an execution plan, which allows you preview the changes that terraform plans to make to your infrastructure.

**terraform plan**

If you get a message like:

Text

Description automatically generated

It did not find your organization, check the spelling and it is case sensitive.

This screen will explain what terraform is going to do in the infrastructure.

Text

Description automatically generated

<Output Truncated>

Terraform apply Operation

## Task 4: Apply the resource deployment

1. Run terraform apply

The terraform apply command is used to apply the changes required to reach the desired state of the configuration, or the pre-determined set of actions generated by a terraform plan execution plan.

**terraform apply**

**or**

**terraform -auto-approve**

Enter **yes** when asked “Do you want to perform these actions?”

Result shows that one resource added to the system.

Text

Description automatically generated

1. To View the new NTP Policy in Cisco Intersight, on left-hand navigation menu, under **CONFIGURE**, select **Policies** and click on the POD**X**-ntp-policy (Where **X** as POD number, for example: POD01).

Graphical user interface, application

Description automatically generated

## Task 5: Update the Policy using Terraform

1. Edit config\_ntp.policy.tf using nano (you may already have the file open in nano)

**nano config\_ntp.policy.tf**

In the ntp\_servers section of the NTP Policy details, add a comma after the existing ntp server and then add a new ntp server **10.1.1.1** in the next line.

Text

Description automatically generated

**Click File > Save**

1. Type the following commands to validate the configuration

**terraform validate**

yes

If validation is successful, then run following command:

**terraform plan**

The output shows what terraform is going to change in Cisco Intersight.

**terraform apply**

Enter **yes** When asked “Do you want to perform these actions?”

Text

Description automatically generated

<Output Truncated>

One resource changed with the NTP servers.

1. To View the updated NTP Policy in Cisco Intersight, on left-hand navigation menu on Cisco Intersight, under **CONFIGURE**, select **Policies** and click on the POD**X**-ntp-policy (Where **X** as POD number, for example: POD01).

## Task 6: Execute a Rollback

Terraform destroy Operation

1. Run terraform destroy to delete the resource created by config\_ntp\_policy.tf file.

The terraform destroy command terminates resources managed by your Terraform project. This command is the inverse of terraform apply in that it terminates all the resources specified in your Terraform state

**terraform destroy**

Enter **yes** when asked “Do you really want to destroy all resources?”

Text

Description automatically generated

<Output Truncated>

The result shows that one resource destroyed from Cisco Intersight.

1. To verify that POD**X**-ntp-policy is no longer in Cisco Intersight, on left-hand navigation menu in Cisco Intersight, under **CONFIGURE**, select **Policies**, and ensure that POD**X**-ntp-policy (Where **X** as POD number, for example: POD01) has been removed.

This completes the Cisco Intersight Terraform Provider lab.