

Infrastructure as Code (IaC)

Hands on Lab with Terraform on Oracle Cloud Infrastructure

Version 2.6 | February 23, 2018

Table of Contents

[Introduction 1](#_Toc507011536)

[**Infrastructure as Code** 1](#_Toc507011537)

[**Infrastructure Lifecycle** 1](#_Toc507011538)

[**Broad categories of IaC** 1](#_Toc507011539)

[**Terraform** 1](#_Toc507011540)

[**Lab Overview** 2](#_Toc507011541)

[**Individual Exercises** 2](#_Toc507011542)

[**Disclaimer** 2](#_Toc507011543)

[Clone Hands-on Lab Repository 4](#_Toc507011544)

[**Cloning Process** 4](#_Toc507011545)

[Getting Started with Terraform Configurations for OCI 5](#_Toc507011546)

# **Introduction**

## **Infrastructure as Code**

“Infrastructure as Code's goal is to create and manage cloud infrastructure and deployments predictably and repeatedly - It makes use of templates and automation for just about everything.”

## **Infrastructure Lifecycle**

* Provision
* Update
* Destroy

## **Broad categories of IaC**

* Ad hoc scripts
* Configuration management tools (chef, puppet etc.)
* Server templating tools (Packer, Vagrant, Docker, etc.)
* Orchestration - Infrastructure Automation tools (Terraform, CloudFormation, Heat)

## **Terraform**

* Built by HashiCorp
* Terraform is a tool that acts like a Makefile for cloud, it’s a multi cloud multi provider templating tool, good at deploying infrastructure. Terraform is not really code but more of a markup language for infrastructure.
* Terraform is a tool for building, changing, managing and versioning infrastructure across different providers efficiently, reliably & at scale.
* Written in Go
* Runtimes available for OS X, FreeBSD, Linux, OpenBSD, Solaris, Windows
* IA32, x64 and ARM
* HCL – (Hashi Configuration Language) which is a simple markup format JSON interoperable
* Works well with existing tools - puppet, chef, ansible, etc

## **Lab Overview**

This lab book is comprised of individual exercises. These exercises allow you to get first hands-on exposure working with Oracle Cloud Infrastructure (OCI) product using a demo environment, where you will see how key features and functionality are deployed in the software. Using what you learn in the presentations and individual exercises working with the software, you will collaborate as a team in developing and delivering practice presentations.

## **Individual Exercises**

In most cases, demo environments are used for this training. Separate instructions will be provided to virtual participants.

A single environment has been assigned and provisioned to each person. Each person will be able to do the hands-on individual exercises by logging in as a different OCI user. For each product, there is an exercise on how to load OCI content, so everyone should have an opportunity to do this activity first hand. The steps may vary by product but the basic concepts apply.

## **Disclaimer**

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle’s products remains at the sole discretion of Oracle. This HOL is based on a HOL on BMCS provided by the Oracle PM team.

**Setup GIT/ GIT Bash**

Before we begin, we need to install git as main dependency for our lab session. This will provide most of the dependencies to continue with our hands on lab

* Please navigate to <https://git-scm.com/downloads> and download suitable installer for your operating system.

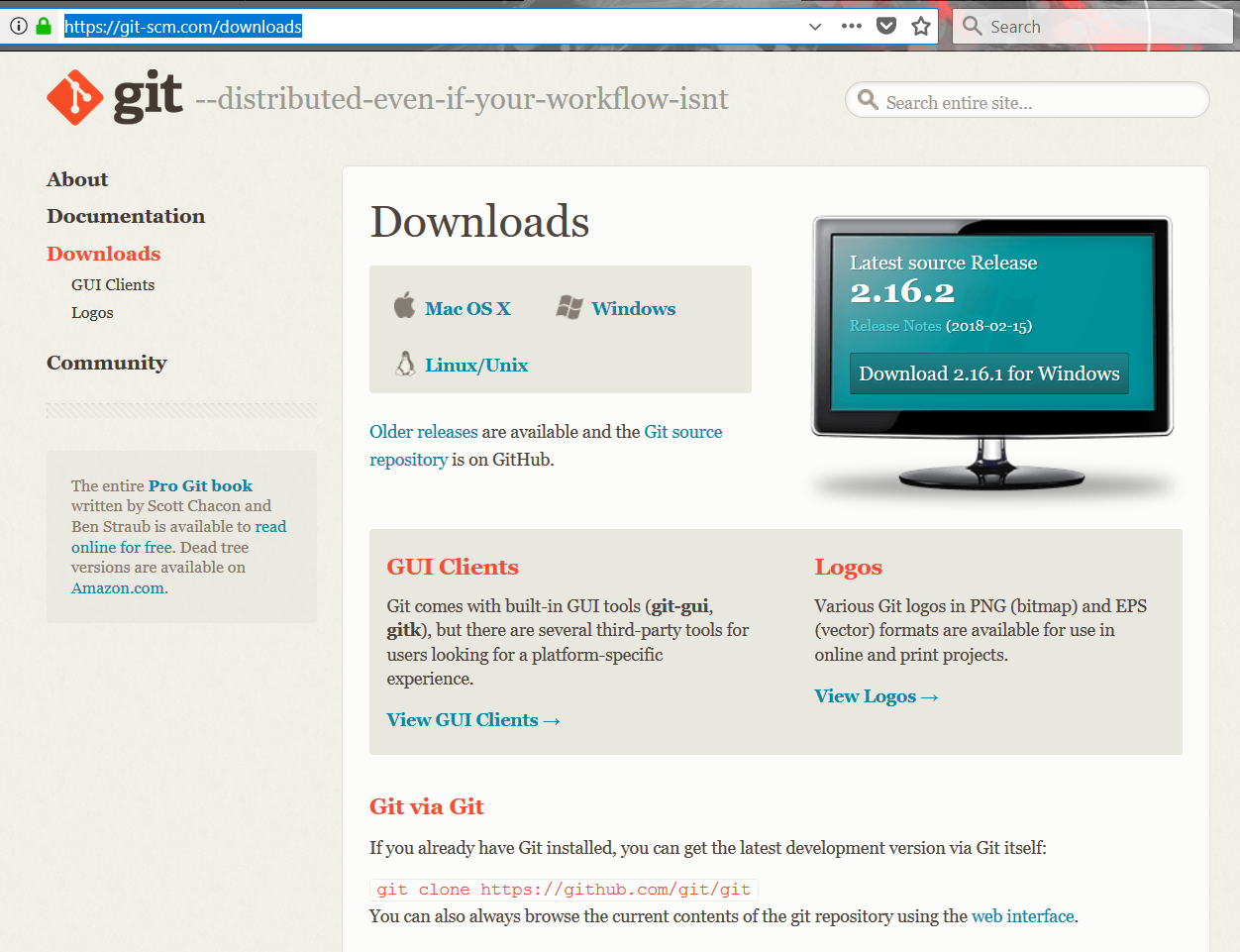


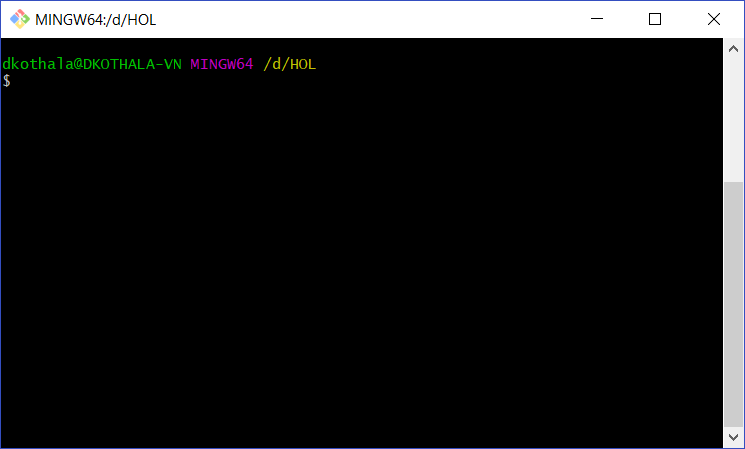
Figure 1. git-scm, download page

# **Clone Hands-on Lab Repository**

We have create pre-configured terraform setup to do this hand-on lab in-order to save time on dependency download time and focus more on IaC capabilities on versioned enterprise cloud infrastructure.

## **Cloning Process**

1. Select one of your desired folder/location and right click on that location
2. Click on *“Git Bash Here*” option

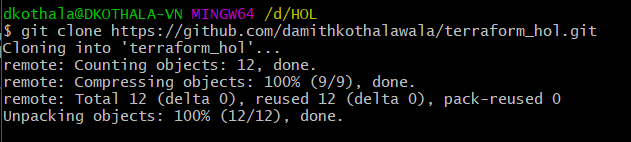


# 

🡨 Once you selected, you will get a window like this.

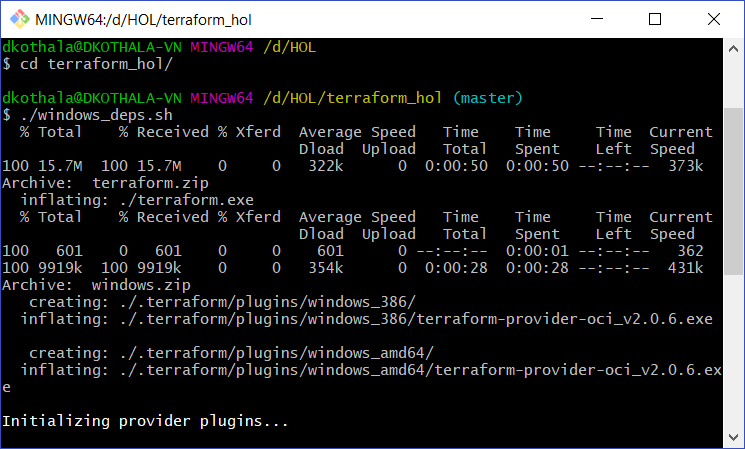
1. Then enter following command on your git bash window. And press “Enter”

|  |
| --- |
| git clone https://github.com/damithkothalawala/terraform\_hol.git |

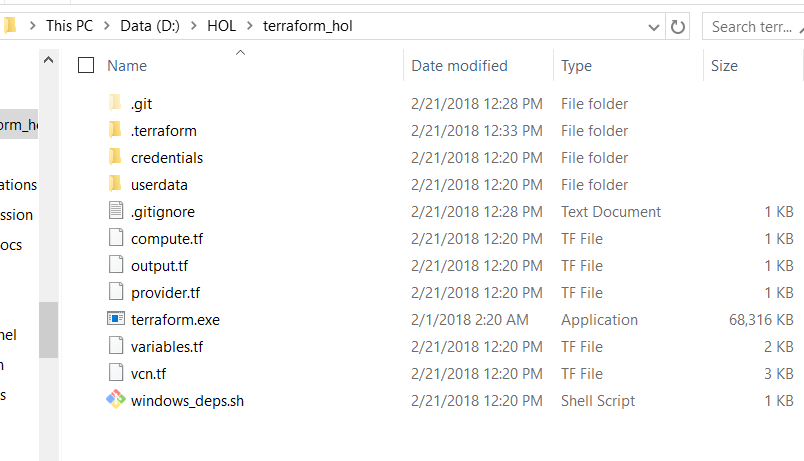
 🡨 You will see this kind of output

1. Then enter following two commands to download dependencies finalize our repository setup

|  |
| --- |
| cd terraform\_hol  ./windows\_deps.sh |

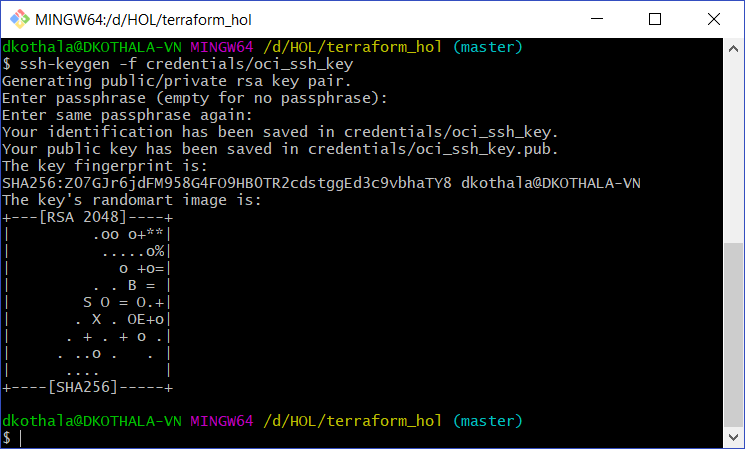
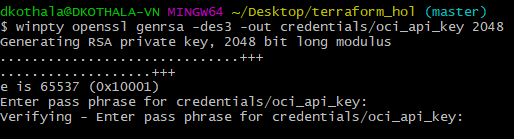


# **Getting Started with Terraform Configurations for OCI**

1. Please minimize your “git bash” window and check for a folder with a name of **“terraform\_hol”** within your directory
2. Within there you should be able to see following folders and files

File and Folder Description

* **variables.tf** 🡪 contains configuration variables which needs to be update with your cloud account
* **provider.tf** 🡪 contains provider (Oracle Cloud Infrastructure) connection details
* **credentials folder** 🡪 contains blank template files for your SSH and Signing public/private key pairs
* **vcn.tf** 🡪 contains your virtual cloud network setup including your network/routes/firewall rules and subnets
* **compute.tf** 🡪 contains information about shape and details about cloud compute instance
* **userdata/bootstrap** 🡪 You can use this file to execute commands at the 1st boot of an instance

1. Please execute following command to generate SSH keys for your cloud instances  
     
   ssh-keygen -f credentials/oci\_ssh\_key  
     
    **Please press enter for the passphrase**
2. Please execute following command to create your API Private Keys  
     
   winpty openssl genrsa -des3 -out credentials/oci\_api\_key 2048  
     
   \*\*\*\* You have to provide a password \*\*\*\*
3. Finally extract API Public key using following command  
     
   winpty openssl rsa -in credentials/oci\_api\_key -outform PEM -pubout -out credentials/oci\_api\_key.pub