

Infra As Code on Oracle Cloud Infrastructure Hands on Lab (Terraform on OCI)

V2.5

ORACLE LAB BOOK | FEB 2018





Table of Contents

Lab: Install, Configure and Run Terraform with Oracle OCI	4
Exercise 1: Sign into the Console	6
Exercise 2: Create a Virtual Cloud Network (VCN)	6
Exercise 3: Provision Windows Instance	16
Exercise 4: Prepare Windows for Terraforming	21
Exercise 5: Initialize, Plan, Apply and Destroy the single instance terraform template.	32

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

4 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.

The Oracle logo, consisting of the word "ORACLE" in white, uppercase letters on a red rectangular background.



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.



Lab: Install, Configure and Run Terraform with Oracle OCI

Oracle Cloud Infrastructure (OCI) platform can run both Oracle workloads and cloud native applications. We will be using Terraform to showcase Infra As a code capabilities in OCI.

In this hands-on lab the process of getting Terraform installed will be covered along with configuring the OCI Terraform plugin and performing some live test runs with Terraform using sample scripts.

At the end of this lab, you will be familiar with using Terraform with Oracle OCI.

Key Goals for this lab are :

1. Get a terraform environment up and running on a staging Windows VM (on OCI)
2. Run a basic terraform apply against Oracle OCI

Pre-Requisites:

1. User, Password, and Tenant for OCI
2. Account credentials for OCI , API Key, ssh keys and the value of the below (sample provided):

```
$env:TF_VAR_tenancy_ocid="PASTE THE TENANCY OCID HERE"
$env:TF_VAR_user_ocid="PASTE THE USER OCID HERE"
$env:TF_VAR_fingerprint="PASTE THE FINGERPRINT OF THE API KEY HERE"
$env:TF_VAR_private_key_path="C:\Users\opc\.oracleoci\oci_api_key.pem"
$env:TF_VAR_region="PASTE THE REGION HERE"
$env:TF_VAR_compartment_ocid="PASTE THE OCID OF THE COMPARTMENT"
$env:TF_VAR_ssh_public_key=Get-Content C:\Users\opc\.ssh\id_rsa.pub -Raw
$env:TF_VAR_ssh_private_key=Get-Content C:\Users\opc\.ssh\id_rsa -Raw
```

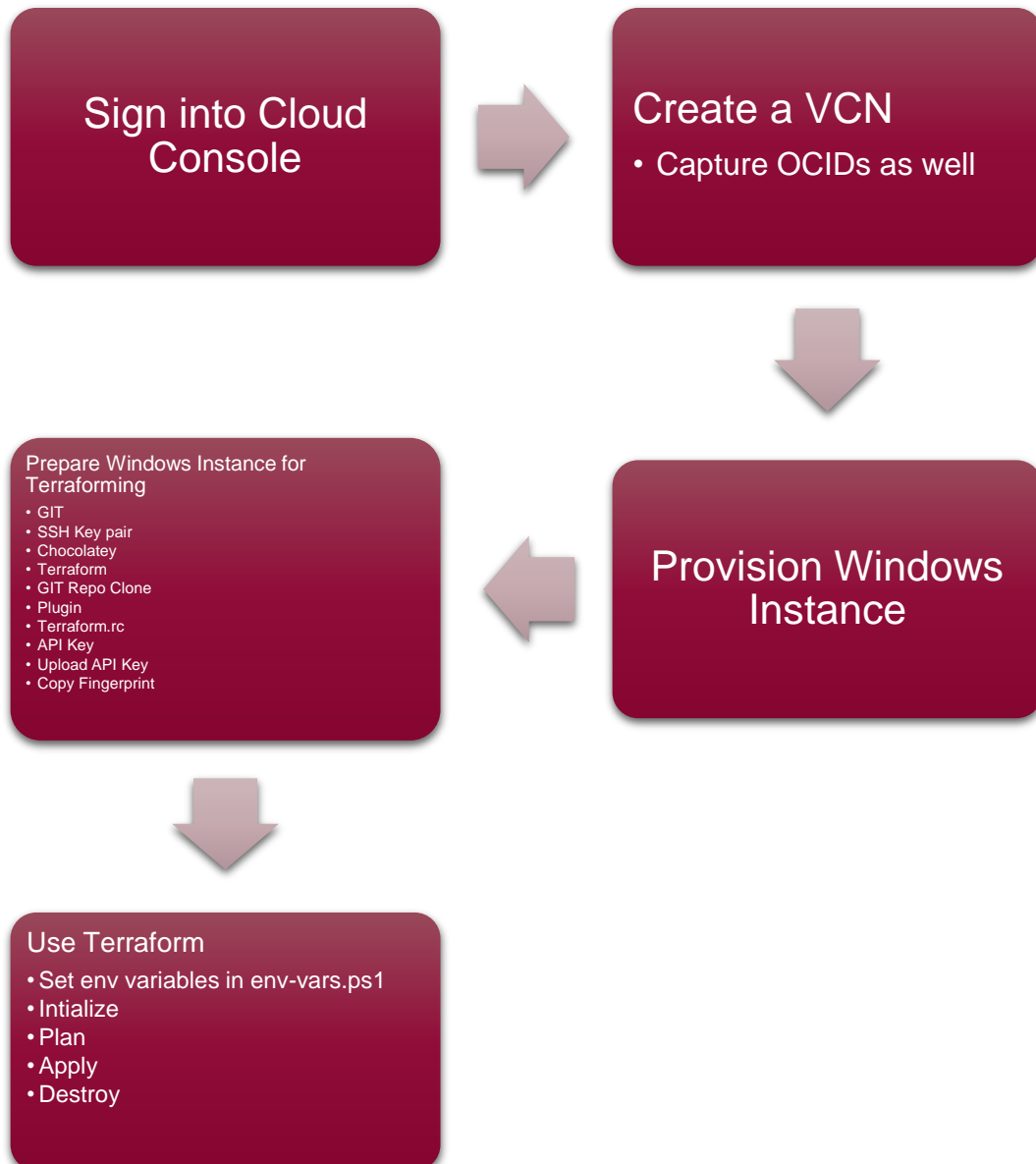
3. ssh keys generated and available
 1. Refer here for key generation tutorials (http://www.oracle.com/webfolder/technetwork/tutorials/obe/cloud/javaservice/JCS/JCS_SSH/create_sshkey.html)

Downloads (DO NOT start downloading these upfront, the HOL will guide you when to download these):

1. Git for Windows / Git-bash: <https://github.com/git-for-windows/git/releases/download/v2.13.0.windows.1/Git-2.13.0-64-bit.exe>
2. Chocolatey (install by command line): iex ((New-Object System.Net.WebClient).DownloadString('https://chocolatey.org/install.ps1'))
3. Terraform to be installed through chocolatey
4. Terraform on OCI repo: <https://github.com/oracle/terraform-provider-oci/> (to be git cloned)
5. Terraform Plugin for OCI : <https://github.com/oracle/terraform-provider-oci/releases/download/v2.0.6/windows.zip>



Process Flow:



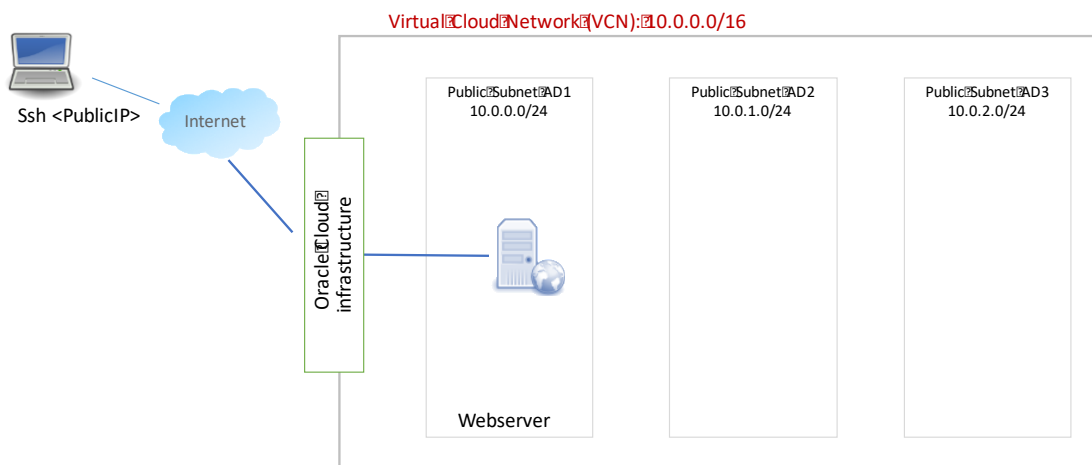
Exercise 1: Sign into the Console

1. Sign in to the console before you begin the lab. You must obtain user / password / tenant credentials.
2. Navigate to <https://console.us-phoenix-1.oraclecloud.com>
3. Enter your credentials to sign-in:
4. User: <>
5. Password: <>
6. Tenant: <>

Exercise 2: Create a Virtual Cloud Network (VCN)

A Virtual Cloud Network (VCN) is a virtual version of a traditional network—including subnets, route tables, and gateways—on which your compute instances run. Customers can bring their network topology to the cloud with VCN. Creating a VCN involves a few key aspects such as:

- Allocate a private IP block for the cloud (CIDR range for the VCN). Customers can bring their own RFC1918 IP addresses.
- Create Subnets by partitioning the CIDR range into smaller networks (sub networks for front end, back end, database)
- Create an optional Internet Gateway to connect VCN subnet with Internet. Instances created in this subnet will have a public IP address.
- Create Route table with route rules for Internet access
- Create Security Group to allow relevant ports for ingress and egress access



Creating a VCN involves allocating a CIDR range (IP address block) for the network, creating a Route Table with custom route rules and path to Internet, carving out a subnet from the IP address block allocated to the VCN.

ORACLE

1. After you login, navigate to the networking tab and select Virtual Cloud Networks.

The screenshot shows the Oracle Cloud Infrastructure (OCI) Networking console. The top navigation bar includes the Oracle logo, Tenancy (gse00014601), Region (us-ashburn-1), and various service links. The left sidebar shows the 'Networking' section with 'Virtual Cloud Networks' selected. The main content area displays 'Virtual Cloud Networks in gse00014601 (root) Compartment' with a 'Create Virtual Cloud Network' button. Below this, a message states: 'There are no Virtual Cloud Networks in gse00014601 (root) that match the filter criteria.' A dropdown menu on the right is open, showing a list of networking services, with 'Virtual Cloud Networks' highlighted. The 'List Scope' section shows the compartment as 'gse00014601 (root)' and the state as 'Any state'.

Tenancy OCID: ocid1.tenancy.oc1..aaaaaaaoykiamqzgfayqirz5cpnuob4dkwikfjijxtxos2l5xcqyhog4sa
[About Oracle](#) [Contact Us](#) [Service Health Dashboard](#) [Legal Notices](#) [Terms of Use](#) [Privacy](#)

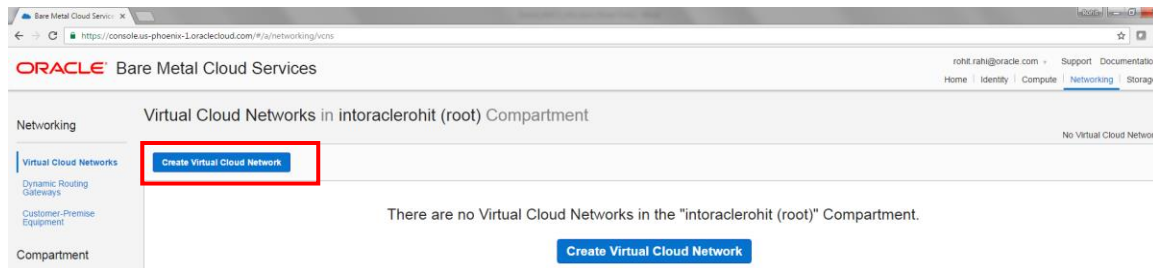
STOP and make sure you're in the correct compartment

E.g. for a user with a username such as **demo.user48**, the correct compartment is **c48** and so on. If you don't select the correct compartment, none of the steps below will work. You might need to scroll down under the compartment heading to locate your specific compartment number. Also note the Tenancy OCID.
(tenancy_ocid)

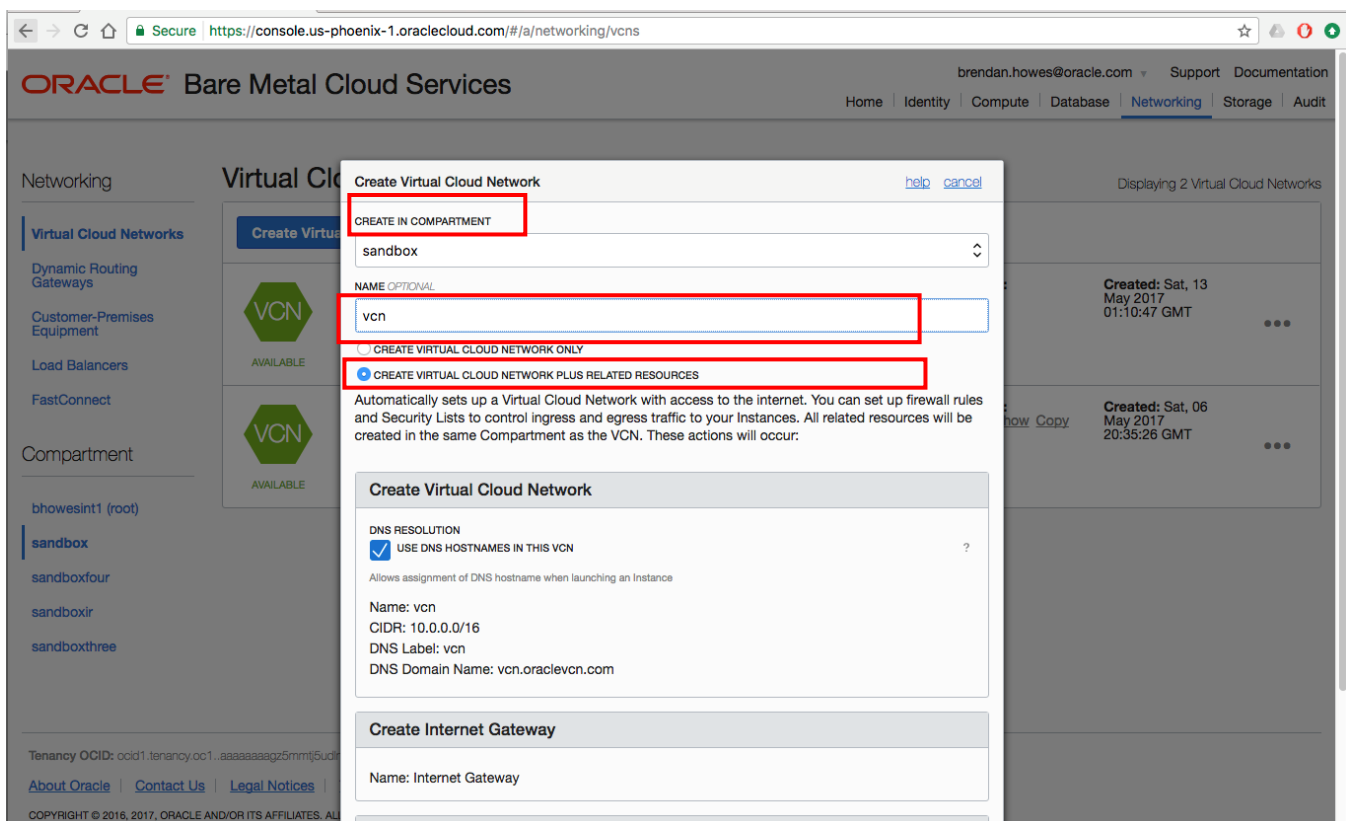
The screenshot shows the Oracle Cloud Infrastructure (OCI) Networking console for a different user, 'demo.user48'. The top navigation bar shows 'Bare Metal Cloud Services' and the user's name. The left sidebar shows the 'Networking' section with 'Virtual Cloud Networks' selected. The main content area displays 'Virtual Cloud Networks in c48 Compartment' with a 'Create Virtual Cloud Network' button. Below this, a message states: 'There are no Virtual Cloud Networks in the "c48" Compartment.' A dropdown menu on the left is open, showing a list of compartments, with 'Compartment' highlighted. The list includes 'gsebmcs00001 (root)', '33', '39', and '44'.

ORACLE

2. Click on the create Virtual Cloud Network button, assuming you're in the correct compartment number. The steps below shows the root compartment, but you should select your specific compartment as per above.



3. Create a Cloud Network by specifying a name for your VCN and selecting the "Create VIRTUAL CLOUD NETWORK PLUS RELATED RESOURCES" option. This will create a VCN, Subnets, Routing Table, Security Groups and Internet Gateway using a 10.0.0.0/16 CIDR range. Scroll to the bottom of the screen and click "create Virtual Cloud Network" button. Be sure to check off use DNS.



ORACLE

Oracle Cloud Infrastructure console showing the 'Create Virtual Cloud Network' dialog. The dialog includes tabs for 'Name: Internet Gateway', 'Update Default Route Table', 'Add Route Rule: 0.0.0.0/0 - Internet Gateway', and 'Create Subnet'. The 'Create Subnet' tab is active, showing details for three subnets. At the bottom, the 'Create Virtual Cloud Network' button is highlighted with a red box, and a checkbox 'View detail page after this resource is created' is checked.

- Once the VCN is created, navigating to list of VCN's, you can see the "vcn" or whatever name was used, which you just created in the step above.

Oracle Cloud Infrastructure console showing the 'Virtual Cloud Networks in sandbox Compartment' page. The page displays a table of VCNs. The second row, representing the newly created VCN, is highlighted with a red box. The table columns include VCN icon, Name (CompleteVCN), CIDR Block (10.0.0.0/16), Default Route Table (Default Route Table for CompleteVCN), DNS Domain Name (cvcn...), and Created date (Sat, 13 May 2017 01:10:47 GMT).

ORACLE


- Click on the vcn link above and it displays the three subnets within this network.

ORACLE Bare Metal Cloud Services

brendan.howes@oracle.com ▾ Support Documentation

Home | Identity | Compute | Database | **Networking** | Storage | Audi

Networking » Virtual Cloud Networks » Virtual Cloud Network Details

**VCN**

Terminate

CIDR Block: 10.0.0.0/16

Compartment: sandbox

Created: Sat, 13 May 2017 02:56:12 GMT

OCID: ocid1.vcn.oc1.phx.aaaaaaa06lgxbmknng6w34jlh4hdgggkcxccowwftwu7cx2crq27ocip7k
[Hide Copy](#)

Default Route Table: [Default Route Table for vcn](#)

DNS Domain Name: vcn.oraclevcn.com [Hide Copy](#)

AVAILABLE

Resources

Subnets (3)

Route Tables (1)

Internet Gateways (1)

Dynamic Routing Gateways (0)

Security Lists (1)

DHCP Options (1)

Compartment

bhowesint1 (root)

sandbox

sandboxfour




sandboxir

sandboxthree

Subnets in sandbox Compartment

Displaying 3 Subnets

Create Subnet

 AVAILABLE	Public Subnet GOFA:PHX-AD-1 OCID: ocid1.subnet.oc1.phx.aaaaaaatw6ahd434fygp24ftrrsytqt52nutmrynftfseg6jezdshyxf73a Hide Copy	CIDR Block: 10.0.0.0/24 Virtual Router MAC Address: 00:00:17:72:37:46	Availability Domain: GOFA:PHX-AD-1 DNS Domain Name: sub05130256480... Show Copy Subnet Access: Public Subnet	Route Table: Default Route Table for vcn Security Lists: Default Security List for vcn	DHCP Options: Default DHCP Options for vcn	Terminate
 AVAILABLE	Public Subnet GOFA:PHX-AD-2 OCID: ...qoucsg Show Copy	CIDR Block: 10.0.1.0/24 Virtual Router MAC Address: 00:00:17:72:37:46	Availability Domain: GOFA:PHX-AD-2 DNS Domain Name: sub05130256481... Show Copy Subnet Access: Public Subnet	Route Table: Default Route Table for vcn Security Lists: Default Security List for vcn	DHCP Options: Default DHCP Options for vcn	Terminate
 AVAILABLE	Public Subnet GOFA:PHX-AD-3 OCID: ...l6g5ca Show Copy	CIDR Block: 10.0.2.0/24 Virtual Router MAC Address: 00:00:17:72:37:46	Availability Domain: GOFA:PHX-AD-3 DNS Domain Name: sub05130256482... Show Copy Subnet Access: Public Subnet	Route Table: Default Route Table for vcn Security Lists: Default Security List for vcn	DHCP Options: Default DHCP Options for vcn	Terminate

- Next, we are going to edit the security Lists in this VCN, since we need to RDP to the instance and will need access to port 3389, also enable ICMP for ingress traffic. Click on the Security Lists tab on left.

You may skip adding the port 3389 right now as this will be added as part of the Windows VM provisioning/creation.

- Click on the Security List under Resources on the left hand pane and it displays the default security rules for this VCN.

ORACLE

Networking » Virtual Cloud Networks » Virtual Cloud Network Details » Security Lists



VCN

Terminate

CIDR Block: 10.0.0.0/16
Compartment: sandbox
Created: Sat, 13 May 2017 02:56:12 GMT

OCID: ...cip7lq [Show](#) [Copy](#)
Default Route Table: [Default Route Table for vcn](#)
DNS Domain Name: vcn... [Show](#) [Copy](#)

AVAILABLE

Resources

- Subnets (3)
- Route Tables (1)
- Internet Gateways (1)
- Dynamic Routing Gateways (0)
- Security Lists (1)**
- DHCP Options (1)

Security Lists in sandbox Compartment

Displaying 1 Security Lists

Create Security List



AVAILABLE

[Default Security List for vcn](#)
OCID: ocid1.securitylist.oc1.phx.aaaaaaa7zdturhsx6krc16adjgk3lp7xfktcdbeho72tlrw6dgormla [Hide](#) [Copy](#)

Created: GMT [View Security List Details](#)
Terminate

...

- Add port RDP 3389 as shown below by clicking on “Edit All Rules”.

Networking » Virtual Cloud Networks » Virtual Cloud Network Details » Security Lists » Security List Details



AVAILABLE

Default Security List for vcn

Edit All Rules Terminate Apply Tag(s)

Security List Information Tags

OCID: ...pnbzq [Show](#) [Copy](#)
Created: Fri, 09 Feb 2018 08:54:36 GMT

Instance traffic is controlled by firewall rules on each instance in addition to this Security List

Resources

- Ingress Rules (3)**
- Egress Rules (1)

Ingress Rules

Stateless Rules

No Ingress Rules

There are no stateless Ingress Rules for this Security List.

Stateful Rules

Source: 0.0.0.0/0	IP Protocol: TCP	Source Port Range: All	Destination Port Range: 22	Allows: TCP traffic for ports: 22 SSH Remote Login Protocol
Source: 0.0.0.0/0	IP Protocol: ICMP	Type and Code: 3, 4		Allows: ICMP traffic for: 3, 4 Destination Unreachable: Fragmentation Needed and Don't Fragment was Set
Source: 10.0.0.0/16	IP Protocol: ICMP	Type and Code: 3		Allows: ICMP traffic for: 3 Destination Unreachable

ORACLE

Oracle recommends adding an ingress rule to receive Path MTU Discovery fragmentation messages. Without it, you may experience connectivity issues for traffic going outside the VCN. For more information, see [Hanging Connection](#).

×

SOURCE CIDR

10.0.0.0/16

IP PROTOCOL

ICMP

[\(more information\)](#)

TYPE AND CODE *(OPTIONAL)*

3

Examples: '0', '3', '5' or 'All'

[\(more information\)](#)

STATELESS

[\(more information\)](#)

Allows ICMP traffic for: 3 Destination Unreachable

[+ Add Rule](#)

<input checked="" type="checkbox"/>	<input type="text" value="0.0.0.0/0"/>	<input type="text" value="All Protocols"/>
STATELESS		(more information)
<i>Allows all traffic for all ports</i>		



SECURITY LIST NAME

Default Security List for vcn

Allow Rules for Ingress

Oracle recommends adding an ingress rule to receive Path MTU Discovery fragmentation messages. Without it, you may experience connectivity issues for traffic going outside the VCN. For more information, see [Handling Connection](#).

<input type="checkbox"/>	<input type="checkbox"/>	SOURCE CIDR 0.0.0.0/0	IP PROTOCOL TCP	SOURCE PORT RANGE (OPTIONAL) All	DESTINATION PORT RANGE (OPTIONAL) 22
STATELESS more information		<small>Allows TCP traffic for ports: 22 SSH Remote Login Protocol</small>			

<input type="checkbox"/>	<input type="checkbox"/>	SOURCE CIDR 0.0.0.0/0	IP PROTOCOL ICMP	TYPE AND CODE (OPTIONAL) 3, 4	
STATELESS more information		<small>Allows ICMP traffic for: 3, 4 Destination Unreachable: Fragmentation Needed and Don't Fragment was Set</small>			

<input type="checkbox"/>	<input type="checkbox"/>	SOURCE CIDR 10.0.0.0/16	IP PROTOCOL ICMP	TYPE AND CODE (OPTIONAL) 3	
STATELESS more information		<small>Allows ICMP traffic for: 3, 4 Destination Unreachable: Fragmentation Needed and Don't Fragment was Set</small>			

<input type="checkbox"/>	<input type="checkbox"/>	SOURCE CIDR 0.0.0.0/0	IP PROTOCOL TCP	SOURCE PORT RANGE (OPTIONAL) All	DESTINATION PORT RANGE (OPTIONAL) 3389
STATELESS more information		<small>Specified IP addresses: 0.0.0.0-255.255.255.255 (4,294,967,296 IP addresses)</small>			

+ Add Rule

Allow Rules for Egress

<input type="checkbox"/>	<input type="checkbox"/>	DESTINATION CIDR 0.0.0.0/0	IP PROTOCOL All Protocols
STATELESS more information		<small>Allows all traffic for all ports</small>	

+ Add Rule

Save Security List Rules

9. Navigate to Identity, Compartments and note the OCID for the compartment. (compartment_ocid)

ORACLE
Oracle Cloud Infrastructure

TENANCY
gse00014601

REGION
us-ashburn-1

oracleidentitycloudservice/cloud.admin

Support

Documentation

Home

Identity

Compute

Database

Networking

Storage

Audit

Identity

Users

Groups

Dynamic Groups

Policies

Service Limits

Compartments

Federation

Tag Namespaces

Compartments

Displaying 3 Compartments

Create Compartment

	gse00014601 (root) OCID: ...hog4sa Show Copy	Description: The root Compartment of the tenancy	Created: --
	Demo OCID: ocid1.compartment.oc1..aaaaaaaasq4tlo277m4kav5kxqgn64je6nt76cftpahnaeciv4whudiaca Hide	Description: Demo Compartment	Created: Thu, 01 Feb 2018 04:40:16 GMT
	ManagedCompartmentForPaaS OCID: ...a3e3oq Show Copy	Description: Idcs-1de17bd56802459489c3cc0c700757f1120436032jgse00014601	Created: Sat, 27 Jan 2018 22:17:45 GMT


10. Navigate to Identity → Users and note the OCID for the login you are using. (user_ocid)

The screenshot shows the Oracle Cloud console interface for 'Bare Metal Cloud Services'. The 'Identity' section is selected, and the 'Users' page is displayed. A table lists users, with the first user 'a.user@oracle.com' highlighted. The 'OCID' field for this user is highlighted in a red box. The OCID is: ocid1.user.oc1...aaaaaaa7gg3hg2sadh7hue57i6i4omav3ak2ointpvloqybc2dylqkrcta. The user is described as 'auser' and was created on 'Sat, 13 May 2017 03:13:04 GMT'.

If you do not see any users listed, then go to User Settings under the User name in the top right hand corner of the portal as shown in the screenshot below, and copy the OCID for the user:

The screenshot shows the Oracle Cloud console interface for 'Oracle Cloud Infrastructure'. The 'demo.user26' dropdown menu is open, and the 'User Settings' option is highlighted in a red box. The user is identified as 'gsebmcs00008' in the 'us-phoenix-1' region. The page also features several informational cards: 'New to Oracle Cloud Infrastructure?' with a 'Getting Started Guide' button, 'Using Compartments' with an 'About Compartments' button, 'Want to Know More?' with a 'Documentation' button, and 'Need Assistance?' with a 'Contact Support' button.

ORACLE



Oracle Cloud Infrastructure

TENANCY
gsebmcs00008

REGION
us-phoenix-1

demo.user26

Support

Documentation

Home

Identity

Compute


Database

Networking

Storage

Audit

Identity > Users > User Details



ACTIVE

demo.user26

Description: Demo user 26

Create/Reset Password Unblock Delete Apply Tag(s)

User Information Tags

OCID:
ocid1.user.oc1..aaaaaaaarhq435pn6vakcwq6ixitcw2vqh56jr7smzfml2xl22hibi7khoa
Hide Copy

Status: Active

Created: Sat, 10 Dec 2016 00:39:51 GMT

Resources

API Keys (0)

Swift Passwords (0)

Amazon S3 Compatibility API Keys (0)

Groups

API Keys

No API Keys

Add Public Key

There are no API Keys for this User.

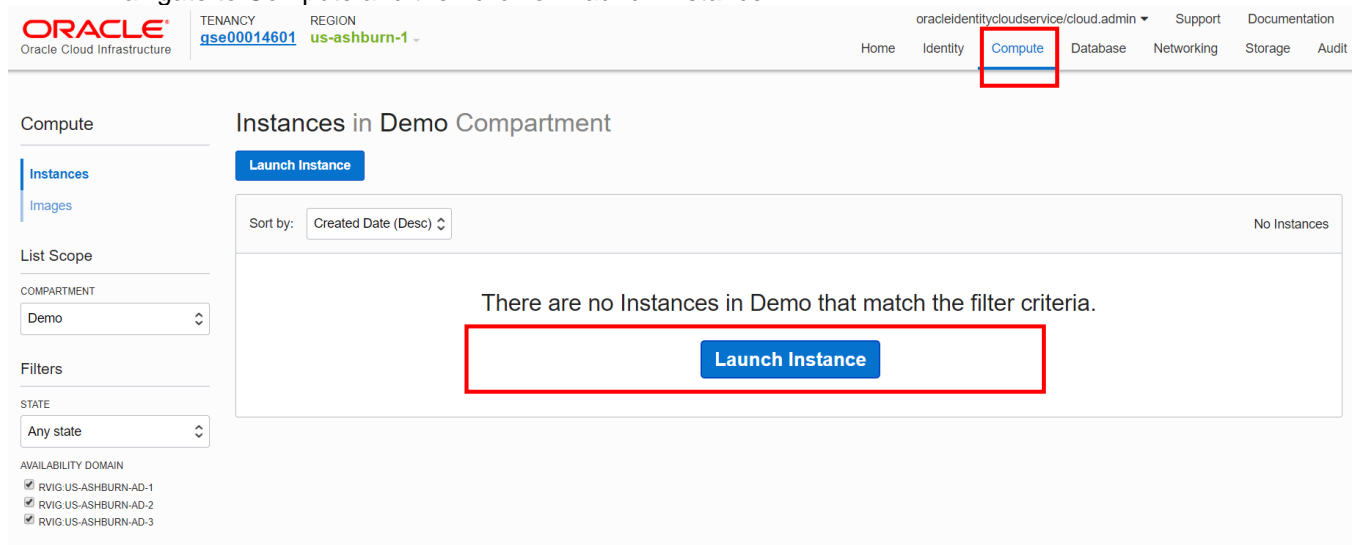
Add Public Key

In this exercise we logged in to OCI, entered our compartment and proceeded to create a VCN and its associated resources and open up the security ACLs for RDP and ICMP. We also collected the user, tenancy, compartment OCIDs.

Exercise 3: Provision Windows Instance

In this exercise we will be creating a Windows Instance wherein Terraform and other related tools will be installed and configured. This will be our staging environment from where we will run the Terraform scripts to provision new Instances, VCNs, attach block storage etc.

1. Navigate to Compute and then click on Launch Instance.



The screenshot displays the Oracle Cloud Infrastructure (OCI) console interface. At the top, the navigation bar includes the Oracle logo, account details (TENANCY: gse00014601, REGION: us-ashburn-1), and a top-level menu with 'Compute' highlighted. The main content area is titled 'Instances in Demo Compartment' and features a 'Launch Instance' button. Below this, a message states 'There are no Instances in Demo that match the filter criteria.' with another 'Launch Instance' button. The left sidebar contains filters for 'COMPARTMENT' (set to Demo), 'STATE' (set to Any state), and 'AVAILABILITY DOMAIN' (with three options: RVIG-US-ASHBURN-AD-1, RVIG-US-ASHBURN-AD-2, and RVIG-US-ASHBURN-AD-3).

ORACLE®

2. Follow the screenshot below to create a Windows Instance.

Launch Instance

[help](#) [cancel](#)

Launching an Instance will take several minutes. You'll need to wait another minute for the OS to boot before you can Remote Desktop to the Instance.

Traffic on this Instance is controlled by its firewall rules in addition to the selected Subnet's Security Lists.

If the image, Virtual Cloud Network, or Subnet is in a different Compartment than the Instance, [click here](#) to enable Compartment selection for those resources.

NAME

stagingwindows

AVAILABILITY DOMAIN

RViG:US-ASHBURN-AD-2

IMAGE SOURCE

☒ ORACLE-PROVIDED OS IMAGE ☐ CUSTOM IMAGE ☐ BOOT VOLUME ☐ IMAGE OCID

IMAGE OPERATING SYSTEM

Windows Server 2012 R2 Standard

The image will be booted using native mode.

SHAPE TYPE

☒ VIRTUAL MACHINE ☐ BARE METAL MACHINE

SHAPE

VM.Standard1.2

Shape compatibility based on selected operating system.

IMAGE BUILD

2018.01.13-0 (latest)

[Release Notes](#)

VIRTUAL CLOUD NETWORK

vcn

SUBNET

Public Subnet RViG:US-ASHBURN-AD-2

ORACLE

To connect to your Windows instance using Remote Desktop, make sure the security list associated with this subnet has an ingress rule that allows TCP traffic to port 3389. For more information about connecting to your Windows instance, see [Connecting to an Instance](#)

Update Security Lists

Your security list may be blocking access to the Remote Desktop Protocol (RDP) needed to log into the instance

☐ Add an ingress rule to the security list to allow access to the RDP port (if necessary)

Note: Updates will be applied to all instances within the chosen subnet, not only the instance being launched

LOGIN CREDENTIALS

Upon creating this Instance, both a user name and an initial password will be generated for you. They will be available on the details screen for the newly launched Instance. You must create a new password upon logging into the instance for the first time.

PRIVATE IP ADDRESS (Optional)

Must be within 10.0.0.2 to 10.0.0.254. Cannot be in current use.

☒ Assign public IP address

HOSTNAME (Optional)

stagingwindows

No spaces. Only letters, numbers, and hyphens. 63 characters max.

FULLY QUALIFIED DOMAIN NAME (Read-only)

stagingwindows.sub02060443380.vcn.oraclevcn.com

☒ I accept the [Partner Terms of Use](#)

TAGS

Tagging is a metadata system that allows you to organize and track resources within your tenancy. Tags are composed of keys and values which can be attached to resources.

[Learn more about tagging](#)

TAG NAMESPACE

None (apply a free-form tag)

TAG KEY

VALUE

+

Launch Instance

☒ View details page after this instance is launched

ORACLE

- Once the Instances provisioning is completed, copy the Public IP Address & note the Windows Username and initial password.

The screenshot shows the Oracle Cloud Infrastructure console. At the top, the Oracle logo and navigation links are visible. The main content area displays the details for an instance named 'stagingwindows'. The instance is in a 'RUNNING' state, indicated by a green box. The 'Instance Information' section shows the following details:

- Availability Domain: RVIG:US-ASHBURN-AD-1
- OCID: ...baytmq [Show](#) [Copy](#)
- Launched: Tue, 06 Feb 2018 05:04:37 GMT
- Compartment: Demo
- Virtual Cloud Network: [vcn](#)
- Launch Mode: NATIVE

The 'Primary VNIC Information' section shows the following details:

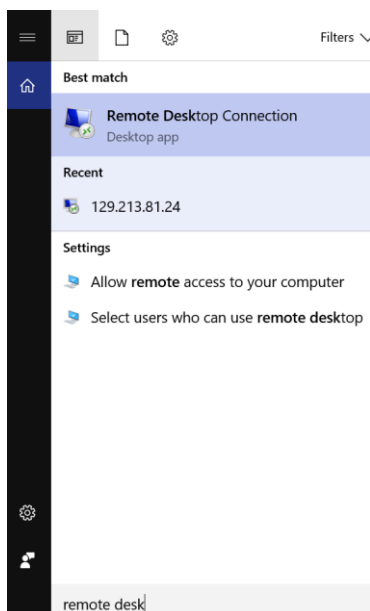
- Private IP Address: 10.0.0.2
- Public IP Address: 129.213.51.252

The 'Image' section shows the following details:

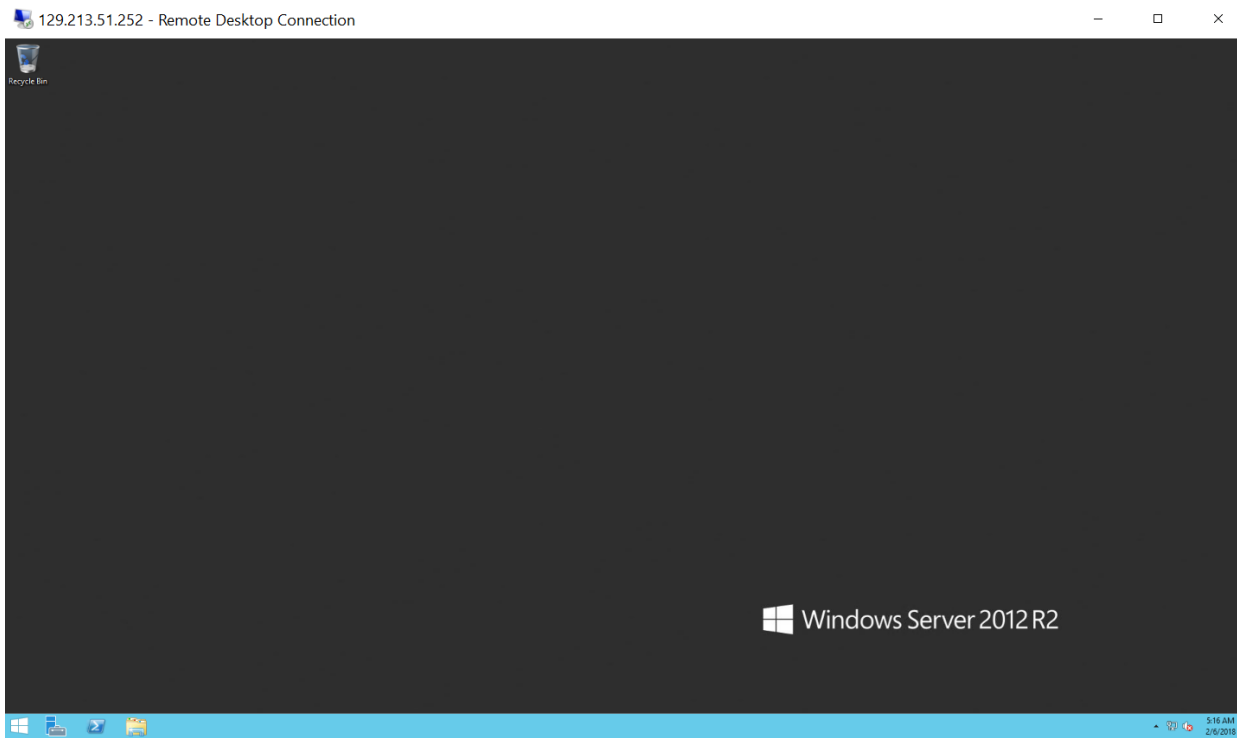
- Image: [Windows-Server-2012-R2-Standard-Edition-VM-2018.01.13-0](#)
- Region: iad
- Shape: VM.Standard1.2
- Windows Username: opc
- Initial Windows Password: Cjz*6.Zv5c,WkZ~cFpY [Hide](#) [Copy](#)

The 'Internal FQDN' is stagingwindows... [Show](#) [Copy](#). The 'Subnet' is [Public Subnet RVIG-US-ASHBURN-AD-1](#).

- Open remote desktop connection from the local laptop and connect to the Public IP Address of the Windows Instance.



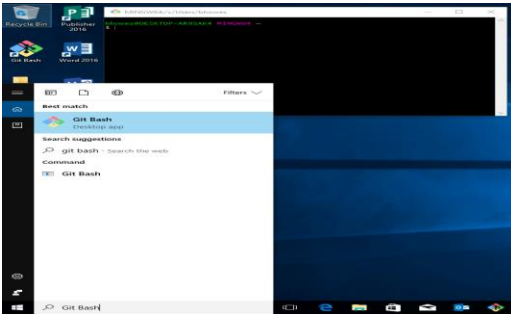
ORACLE



Exercise 4: Prepare Windows for Terraforming

In this section we prepare the terraform runtime environment for use with Windows. This is geared towards PowerShell users. Logon to the Windows Instance created in the previous section and do the below:

1. Install Git for windows by choosing all the default options. Download <https://github.com/git-for-windows/git/releases/download/v2.13.0.windows.1/Git-2.13.0-64-bit.exe> and install.
2. Enter Git-bash:



3. Generate ssh-keys by running this command in the Git-Bash:

```
$ ssh-keygen
```

(Hit enter for every prompt, do not enter anything for simplicity of this exercise)

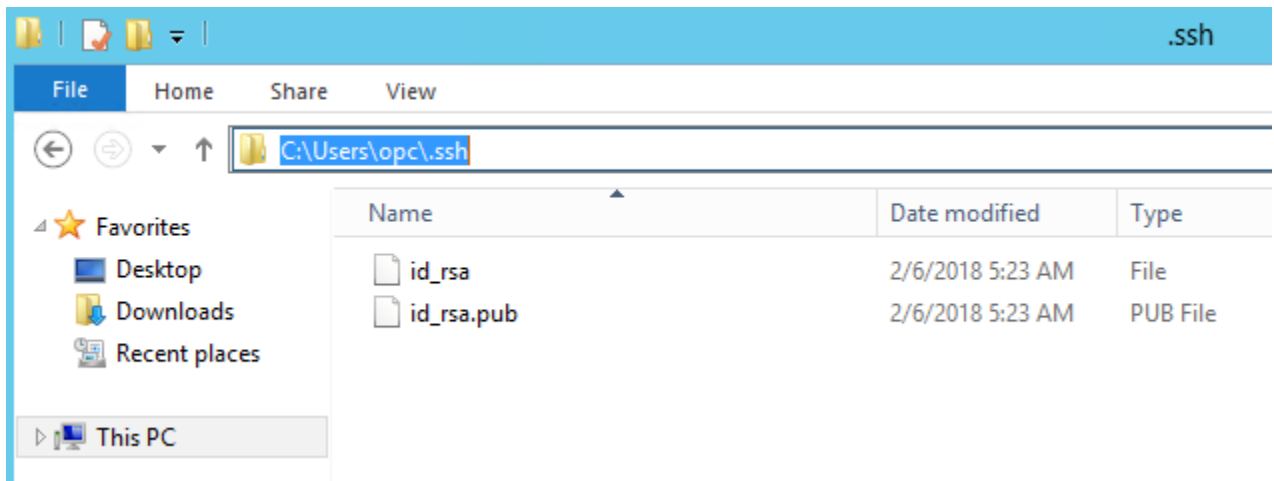
```
MINGW64:/c/Users/opc
opc@stagingwindows MINGW64 ~
$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/opc/.ssh/id_rsa):
Created directory '/c/Users/opc/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/opc/.ssh/id_rsa.
Your public key has been saved in /c/Users/opc/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:tTuUy0bxg0/tzCiYfFIFeyxHpbVfqGaNbtylignr8RI opc@stagingwindows
The key's randomart image is:
+----[RSA 2048]-----+
  .o .
  .o.+ + .
  . =B + . .
  oo==O = .
  E S=B B O
  . += B
  .o+. B .
  =.o. *
  o .. +
+----[SHA256]-----+
```

NOTE: In Git-Bash, C:\Users\username\ is shown as /c/Users/username/

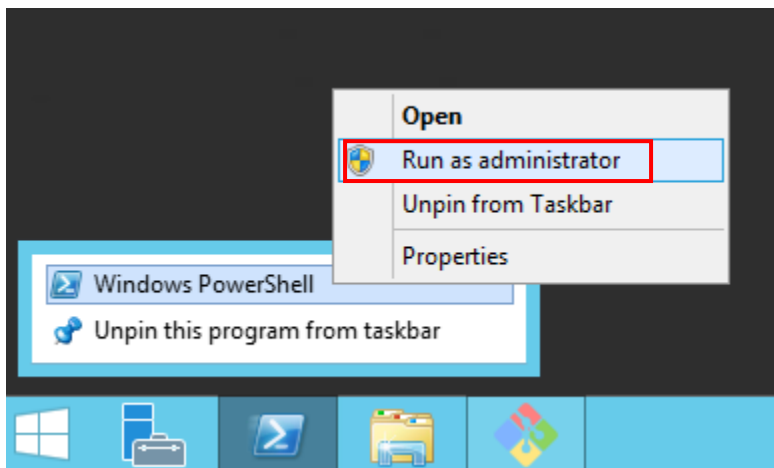
ORACLE®

NOTE: Alternatively, you can download a shell script from [here](#). You may run this shell script directly inside Git-Bash to create the key pair.

4. The Public and Private keys will be created under: C:\Users\opc\.ssh



5. Start an Admin PowerShell Window:



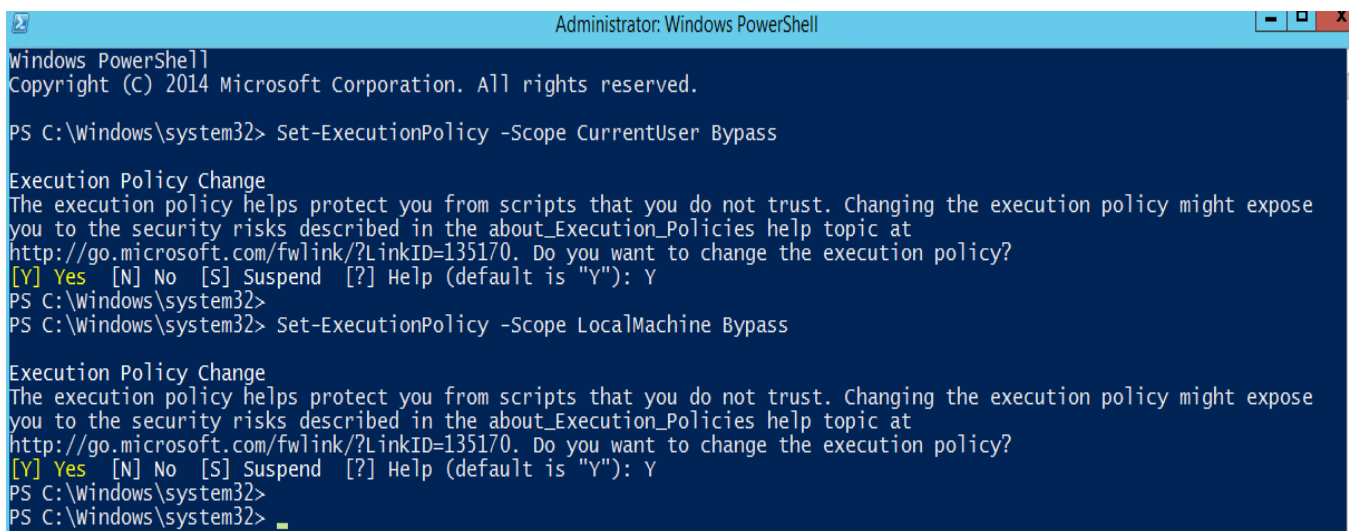
6. Now we need to run chocolatey installer to install Chocolatey from PowerShell. Install completely with indicated defaults.

NOTE:

- a. Chocolatey is useful when installing packages, particularly things like wget or GraphViz or other tools that are commonly packaged in *nix/Linux.
- b. By default, all new program files managed by Chocolatey go into the path and are located here:
C:\ProgramData\Chocolatey\bin
- c. For this exercise, chocolatey will be installing terraform.exe into C:\ProgramData\Chocolatey\bin\ and adding that to the path. If chocolatey is causing issues just place terraform.exe and the terraform-provider-oci.exe in the path.
- d. ProgramData is by default hidden in windows. Enter "C:\ProgramData" into File Explorer in the address bar and it will show.
- e. PowerShell Execution policy after installation should be set back to their previous values, the default is Undefined and a safer setting which still allows signed scripts is AllSigned.

```
PS > Set-ExecutionPolicy -Scope CurrentUser Bypass
```

```
PS > Set-ExecutionPolicy -Scope LocalMachine Bypass
```



```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) 2014 Microsoft Corporation. All rights reserved.

PS C:\Windows\system32> Set-ExecutionPolicy -Scope CurrentUser Bypass

Execution Policy Change
The execution policy helps protect you from scripts that you do not trust. Changing the execution policy might expose
you to the security risks described in the about_Execution_Policies help topic at
http://go.microsoft.com/fwlink/?LinkID=135170. Do you want to change the execution policy?
[Y] Yes [N] No [S] Suspend [?] Help (default is "Y"): Y
PS C:\Windows\system32>
PS C:\Windows\system32> Set-ExecutionPolicy -Scope LocalMachine Bypass

Execution Policy Change
The execution policy helps protect you from scripts that you do not trust. Changing the execution policy might expose
you to the security risks described in the about_Execution_Policies help topic at
http://go.microsoft.com/fwlink/?LinkID=135170. Do you want to change the execution policy?
[Y] Yes [N] No [S] Suspend [?] Help (default is "Y"): Y
PS C:\Windows\system32>
PS C:\Windows\system32> _
```


PS > iex ((New-Object System.Net.WebClient).DownloadString('https://chocolatey.org/install.ps1'))

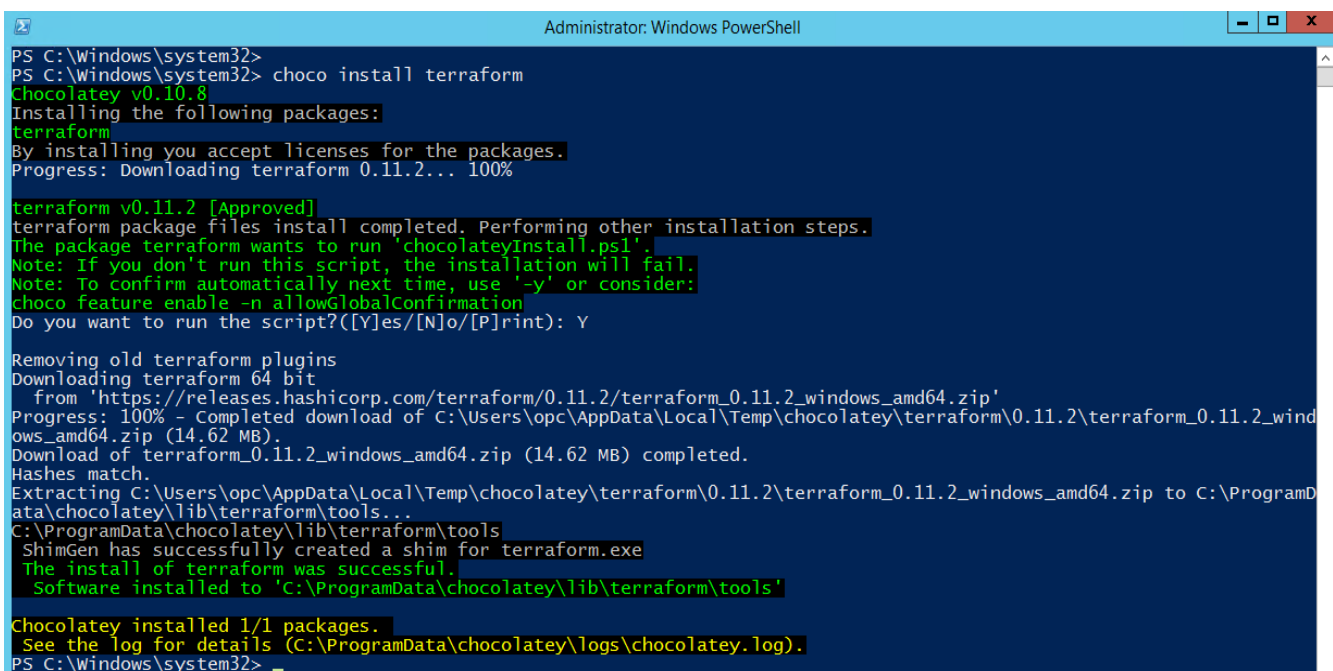
```
PS C:\Windows\system32> iex ((New-Object System.Net.WebClient).DownloadString('https://chocolatey.org/install.ps1'))
Getting latest version of the Chocolatey package for download.
Getting Chocolatey from https://chocolatey.org/api/v2/package/chocolatey/0.10.8.
Downloading 7-Zip commandline tool prior to extraction.
Extracting C:\Users\opc\AppData\Local\Temp\chocolatey\chocInstall\chocolatey.zip to C:\Users\opc\AppData\Local\Temp\cho
chocolatey\chocInstall...
Installing chocolatey on this machine
Creating ChocolateyInstall as an environment variable (targeting 'Machine')
Setting ChocolateyInstall to 'C:\ProgramData\chocolatey'
WARNING: It's very likely you will need to close and reopen your shell
before you can use choco.
Restricting write permissions to Administrators
We are setting up the Chocolatey package repository.
The packages themselves go to 'C:\ProgramData\chocolatey\lib'
(i.e. C:\ProgramData\chocolatey\lib\yourPackageName).
A shim file for the command line goes to 'C:\ProgramData\chocolatey\bin'
and points to an executable in 'C:\ProgramData\chocolatey\lib\yourPackageName'.

Creating Chocolatey folders if they do not already exist.

WARNING: You can safely ignore errors related to missing log files when
upgrading from a version of Chocolatey less than 0.9.9.
'Batch file could not be found' is also safe to ignore.
'The system cannot find the file specified' - also safe.
chocolatey.nupkg file not installed in lib.
Attempting to locate it from bootstrapper.
PATH environment variable does not have C:\ProgramData\chocolatey\bin in it. Adding...
WARNING: Not setting tab completion: Profile file does not exist at
'C:\Users\opc\Documents\WindowsPowerShell\Microsoft.PowerShell_profile.ps1'.
Chocolatey (choco.exe) is now ready.
You can call choco from anywhere, command line or powershell by typing choco.
Run choco /? for a list of functions.
You may need to shut down and restart powershell and/or consoles
first prior to using choco.
Ensuring chocolatey commands are on the path
Ensuring chocolatey.nupkg is in the lib folder
PS C:\Windows\system32>
```

7. Once chocolatey is installed, install the following packages:

PS> choco install terraform



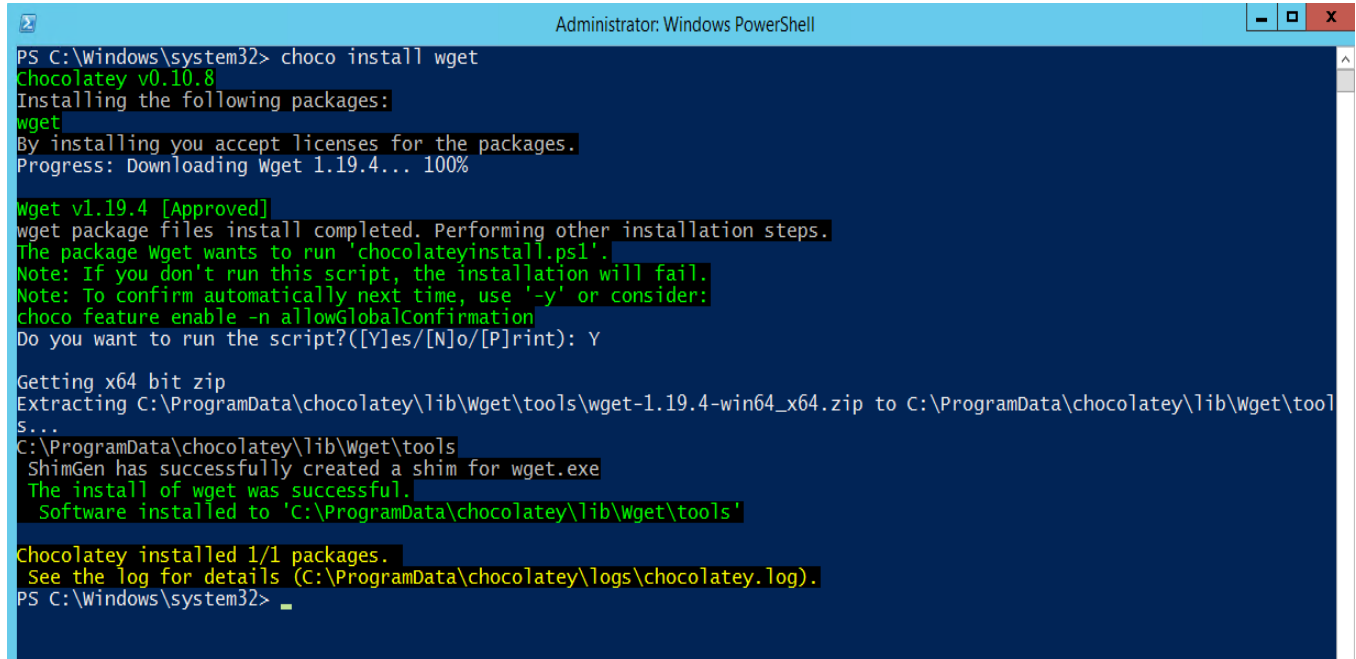
```
Administrator: Windows PowerShell
PS C:\Windows\system32>
PS C:\Windows\system32> choco install terraform
Chocolatey v0.10.8
Installing the following packages:
terraform
By installing you accept licenses for the packages.
Progress: Downloading terraform 0.11.2... 100%

terraform v0.11.2 [Approved]
terraform package files install completed. Performing other installation steps.
The package terraform wants to run 'chocolateyinstall.ps1'.
Note: If you don't run this script, the installation will fail.
Note: To confirm automatically next time, use '-y' or consider:
choco feature enable -n allowGlobalConfirmation
Do you want to run the script?([Y]es/[N]o/[P]rint): y

Removing old terraform plugins
Downloading terraform 64 bit
from 'https://releases.hashicorp.com/terraform/0.11.2/terraform_0.11.2_windows_amd64.zip'
Progress: 100% - Completed download of C:\Users\opc\AppData\Local\Temp\chocolatey\terraform\0.11.2\terraform_0.11.2_wind
ows_amd64.zip (14.62 MB).
Download of terraform_0.11.2_windows_amd64.zip (14.62 MB) completed.
Hashes match.
Extracting C:\Users\opc\AppData\Local\Temp\chocolatey\terraform\0.11.2\terraform_0.11.2_windows_amd64.zip to C:\ProgramD
ata\chocolatey\lib\terraform\tools...
C:\ProgramData\chocolatey\lib\terraform\tools
ShimGen has successfully created a shim for terraform.exe
The install of terraform was successful.
Software installed to 'C:\ProgramData\chocolatey\lib\terraform\tools'

Chocolatey installed 1/1 packages.
See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).
PS C:\Windows\system32>
```

PS> choco install wget



```
Administrator: Windows PowerShell
PS C:\Windows\system32> choco install wget
Chocolatey v0.10.8
Installing the following packages:
wget
By installing you accept licenses for the packages.
Progress: Downloading Wget 1.19.4... 100%

Wget v1.19.4 [Approved]
Wget package files install completed. Performing other installation steps.
The package Wget wants to run 'chocolateyinstall.ps1'.
Note: If you don't run this script, the installation will fail.
Note: To confirm automatically next time, use '-y' or consider:
choco feature enable -n allowGlobalConfirmation
Do you want to run the script?([Y]es/[N]o/[P]rint): Y

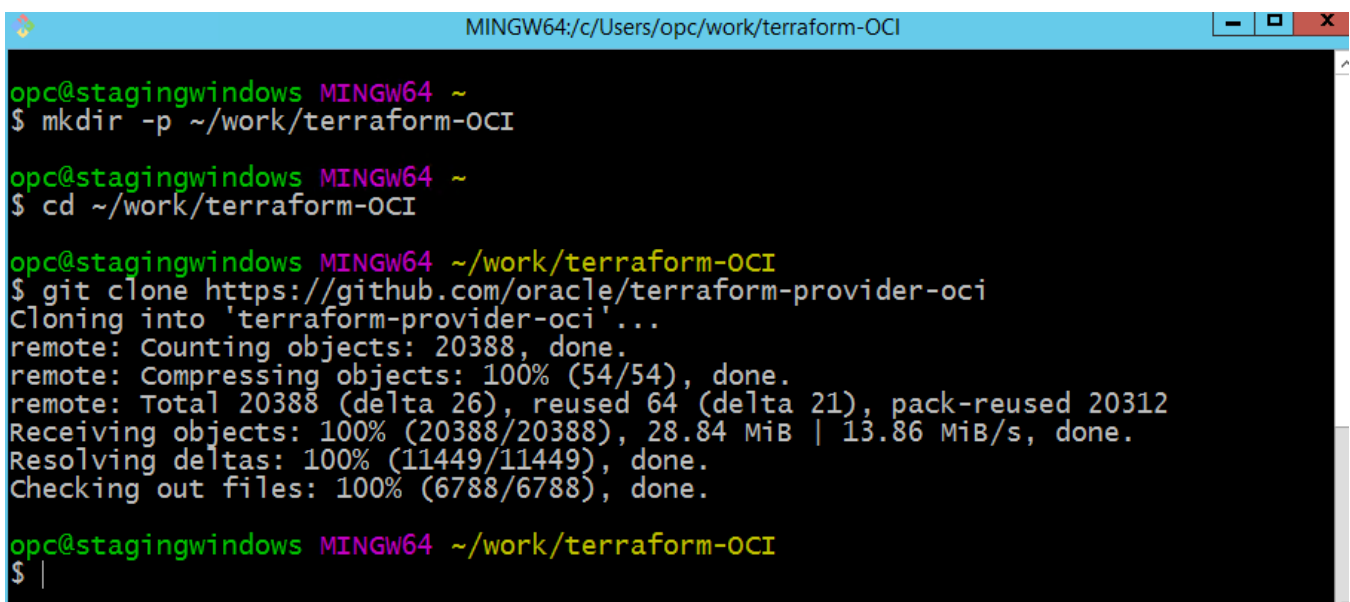
Getting x64 bit zip
Extracting C:\ProgramData\chocolatey\lib\Wget\tools\wget-1.19.4-win64_x64.zip to C:\ProgramData\chocolatey\lib\Wget\tools\...
C:\ProgramData\chocolatey\lib\Wget\tools
ShimGen has successfully created a shim for wget.exe
The install of wget was successful.
Software installed to 'C:\ProgramData\chocolatey\lib\Wget\tools'

Chocolatey installed 1/1 packages.
See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).
PS C:\Windows\system32>
```

8. Enter Git-Bash and create a working directory and **clone** our repo:

```
$ mkdir -p ~/work/terraform-OCI
$ cd ~/work/terraform-OCI
$ git clone https://github.com/oracle/terraform-provider-oci
```

NOTE: Alternatively, you can download a shell script from [here](#). Create a shell script (.sh) and run this shell script directly inside Git-Bash to clone the repo.

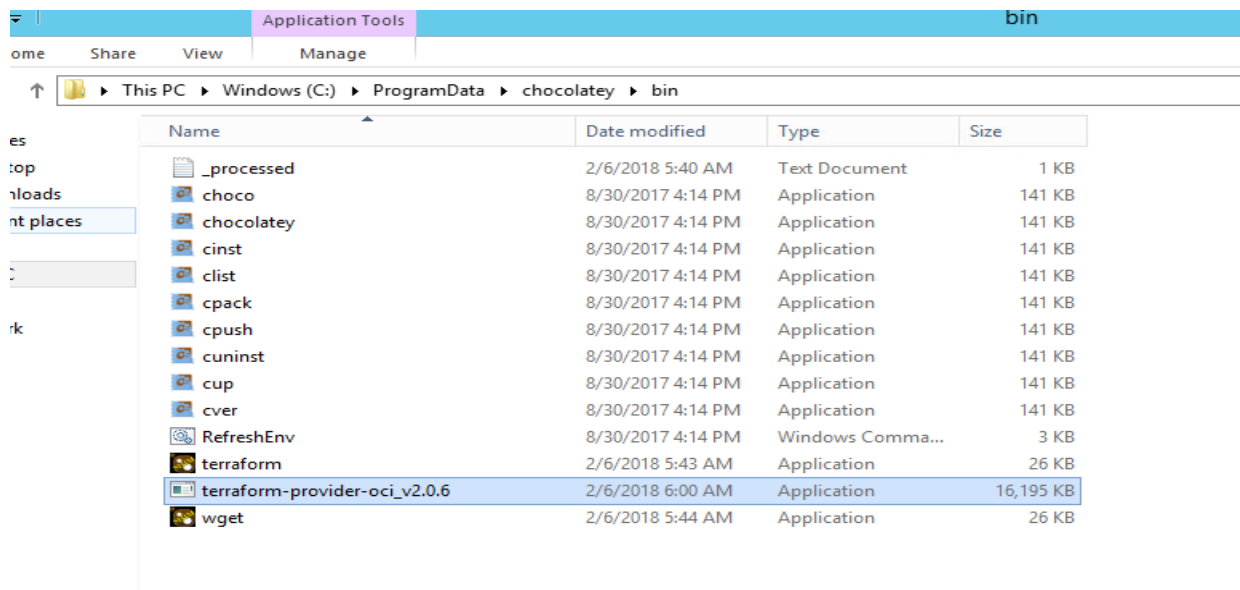


```
MINGW64:/c:/Users/opc/work/terraform-OCI
opc@stagingwindows MINGW64 ~
$ mkdir -p ~/work/terraform-OCI
opc@stagingwindows MINGW64 ~
$ cd ~/work/terraform-OCI
opc@stagingwindows MINGW64 ~/work/terraform-OCI
$ git clone https://github.com/oracle/terraform-provider-oci
Cloning into 'terraform-provider-oci'...
remote: Counting objects: 20388, done.
remote: Compressing objects: 100% (54/54), done.
remote: Total 20388 (delta 26), reused 64 (delta 21), pack-reused 20312
Receiving objects: 100% (20388/20388), 28.84 MiB | 13.86 MiB/s, done.
Resolving deltas: 100% (11449/11449), done.
Checking out files: 100% (6788/6788), done.
opc@stagingwindows MINGW64 ~/work/terraform-OCI
$ |
```

9. Get the OCI terraform package for Windows:
<https://github.com/oracle/terraform-provider-oci/releases/download/v2.0.6/windows.zip> & unzip the archive.

Place the file `C:\Users\opc\Downloads\windows\windows_386\terraform-provider-oci_v2.0.6.exe` into the same directory as `terraform.exe` (`C:\ProgramData\Chocolatey\bin`)

Note: ProgramData folder under C:\ is a hidden folder!



NOTE: Whatever CPU type terraform.exe is built for should be matched by the terraform-provider-oci.exe. By default they should both be set to machine type 32-bit even on x64 systems.

10. Create a terraform.rc file to add the OCI terraform provider. Use notepad to create the following and save in a terraform.rc file name.

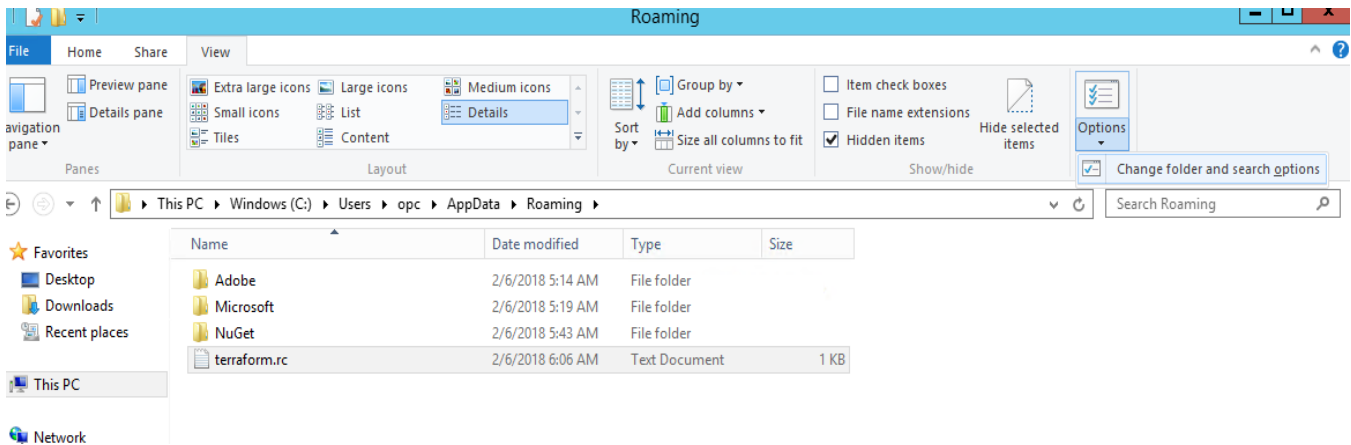
This file needs to be placed under `C:\Users\opc\AppData\Roaming` folder. (AppData is a hidden folder!)

```
providers {  
  oci = "C:\\ProgramData\\Chocolatey\\bin\\terraform-provider-oci_v2.0.6.exe"  
}
```

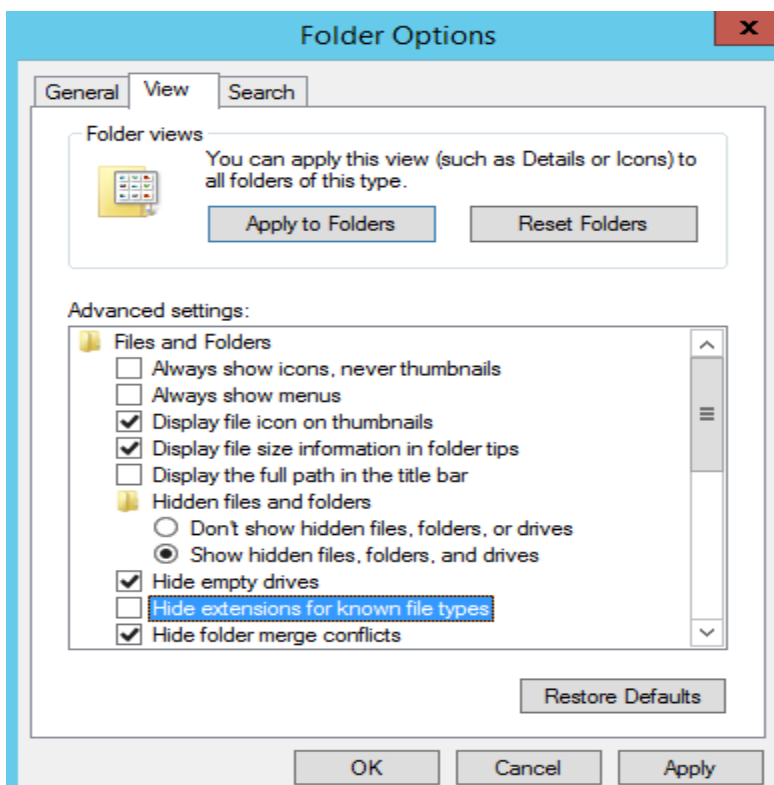
NOTE: "\\" must be escaped here with "\\\" and make sure that the file is saved as terraform.rc.

If the file type shows as 'Text Document', then go to View->Options->Change folder and search options

ORACLE







In the pop up screen, navigate to View and uncheck the option Hide extensions for known file types, and click on Apply & Ok



The screenshot shows a Windows Explorer window. The address bar displays the path: This PC > Windows (C:) > Users > opc > A. The main pane shows a list of files and folders. The files are: Adobe, Microsoft, NuGet, and terraform.rc.txt. The file terraform.rc.txt is selected and highlighted in blue.

▶ This PC ▶ Windows (C:) ▶ Users ▶ opc ▶ AppData ▶ Roaming ▶

Name	Date modified	Type	Size
 Adobe	2/6/2018 5:14 AM	File folder	
 Microsoft	2/6/2018 5:19 AM	File folder	
 NuGet	2/6/2018 5:43 AM	File folder	
 terraform.rc	2/6/2018 6:06 AM	RC File	1 KB

- NOTE:** Alternatively, you can download a shell script from [here](#) to automate the entire step. Create a shell script (.sh) and run this shell script directly inside Git-Bash to create the API keys. Also, for your information, the folder `~/oracleoci` is replaced by `~/apikeyoci` in the automation script.

```
opc@stagingwindows MINGW64 ~/work/terraform-OCI
$ mkdir ~/.oracleoci

opc@stagingwindows MINGW64 ~/work/terraform-OCI
$ openssl genrsa -out ~/.oracleoci/oci_api_key.pem 2048
Generating RSA private key, 2048 bit long modulus
.....+++
.....+++
e is 65537 (0x10001)
```

```
$ chmod 0700 ~/.oracleoci
$ chmod 0600 ~/.oracleoci/oci_api_key.pem
$ openssl rsa -pubout -in ~/.oracleoci/oci_api_key.pem -out ~/.oracleoci/oci_api_key_public.pem
```

```
opc@stagingwindows MINGW64 ~/work/terraform-OCI
$ chmod 0700 ~/.oracleoci

opc@stagingwindows MINGW64 ~/work/terraform-OCI
$ chmod 0600 ~/.oracleoci/oci_api_key.pem

opc@stagingwindows MINGW64 ~/work/terraform-OCI
$ openssl rsa -pubout -in ~/.oracleoci/oci_api_key.pem -out ~/.oracleoci/oci_api_key_public.pem
writing RSA key

opc@stagingwindows MINGW64 ~/work/terraform-OCI
$
```

```
$ cat ~/.oracleoci/oci_api_key_public.pem
$ openssl rsa -pubout -outform DER -in ~/.oracleoci/oci_api_key.pem | openssl md5
$ openssl rsa -pubout -outform DER -in ~/.oracleoci/oci_api_key.pem | openssl md5 >
~/.oracleoci/oci_api_key_fingerprint
```

```
opc@stagingwindows MINGW64 ~/work/terraform-OCI
$ cat ~/.oracleoci/oci_api_key_public.pem
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA2nw3I0gz62xrSteOh400
YZ0SMK/2txU4jet8w7BNVCJNyMSyFRSAAwxK9YG0J/241ftAtGCUdmtYfkuUMCik
I3orMVi1k0ktOZ74vjxH9Ys2bS15M6YI+E97+usxMnCVhv8f+o88GldC+4vw3WUc
mSRqly/EEoDCaV+LTfZfNvGcXRLGEsTRTgc6IFyXq22dWPxpB5epDVi/z1AMlqvz
GsZJI/C4NiHp7LUEiWBN8E2zraAEFKpCBWxnjTvQv9bfVxeLVw/WezXUA754IxUB
LQ53r3gVctnnKWWu2roXe26DGVjvoJn/UjRlowa/opFE0dzCoegWVnDJxRth4ija
VwIDAQAB
-----END PUBLIC KEY-----

opc@stagingwindows MINGW64 ~/work/terraform-OCI
$ openssl rsa -pubout -outform DER -in ~/.oracleoci/oci_api_key.pem | openssl md5
writing RSA key
(stdin)= 3d8c044931124ed112596b37f011e802

opc@stagingwindows MINGW64 ~/work/terraform-OCI
$ openssl rsa -pubout -outform DER -in ~/.oracleoci/oci_api_key.pem | openssl md5
> ~/.oracleoci/oci_api_key_fingerprint
writing RSA key
```

Save the Public Key:

```
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA2nw3I0gz62xrSteOh400
YZ0SMK/2txU4jet8w7BNVCJNyMSyFRSAAwxK9YG0J/241ftAtGCUdmtYfkuUMCik
I3orMVi1k0ktOZ74vjxH9Ys2bS15M6YI+E97+usxMnCVhv8f+o88GldC+4vw3WUc
mSRqly/EEoDCaV+LTfZfNvGcXRLGEsTRTgc6IFyXq22dWPxpB5epDVi/z1AMlqvz
GsZJI/C4NiHp7LUEiWBN8E2zraAEFKpCBWxnjTvQv9bfVxeLVw/WezXUA754IxUB
LQ53r3gVctnnKWWu2roXe26DGVjvoJn/UjRlowa/opFE0dzCoegWVnDJxRth4ija
VwIDAQAB
-----END PUBLIC KEY-----
```



12. Go to OCI Portal, Identity->Users->Choose the user that you are using or logged into with-> View User Details

ORACLE

Oracle Cloud Infrastructure

TENANCY

gse00014601

REGION

us-ashburn-1

Home

Identity

Compute

Database

Networking

Storage

AI

Support

Documentation

Identity

Users

Displaying 2 Users

Create User

PLEASE NOTE: This tenancy uses federated user accounts. Federated users cannot be managed by this console.

<div> <div>D</div> <div>demo.user</div> <div>OCID: ...mechyq</div> <div>Show</div> <div>Copy</div> <div>ACTIVE</div> </div>	<div>Description: demo.user</div>	<div>Created: Mon, 05 Feb 2018 11:58:42 GMT</div>	<div>...</div>
<div> <div>G</div> <div>gse-admin_ww@oracle.com</div> <div>OCID: ...4l2vvq</div> <div>Show</div> <div>Copy</div> <div>ACTIVE</div> </div>	<div>Description: gse-admin_ww gse-admin_ww</div>	<div>Created: Sat, 27 Jan 2018 22:14:44 GMT</div>	<div> <div>View User Details</div> <div>Apply Tag(s)</div> <div>Delete</div> <div>...</div> </div>

- Click on “Add Public Key” & paste the `OCI_api_key_public.pem` (the key which you have copied in the previous page) into the pop up window.

Ensure there are NO trailing spaces after pasting the Key and then click on Add

ORACLE®
Oracle Cloud Infrastructure

TENANCY
gse00014601

REGION
us-ashburn-1

oracleidentitycloudservice/cloud.admin

Support

Home

Identity

Compute

Database

Networking

G

ACTIVE

Resources

API Keys (0)

Swift Passwords (0)

Amazon S3 Compatibility API Keys (0)

Groups (1)

gse-admin_ww@oracle.com

Describe

Create

User Info

OCID:

Create

API Keys

Add

Add Public Key

help cancel

Note: Public Keys must be in the PEM format.

PUBLIC KEY

-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA2nw3I0gz62xrSteoh400
Y205MK/2txU4jet8w7BHVCJNyMSyFRSAAwXK9YG0J/z41fAtGCudmTYFkuUMC1k
I3orMv11koktOZ74vJxH9Ys2b515M6YI+E97+usxMnCvHv8f+o886iDC+4vw3WUC
mSRq1V/EEoDcav+LTfZfNvGcXRLGESRTTgc6IFyXq22dwPxpB5ep0V1/z1AM1qvz
GSZJr/C4NiHp7LUEiNB8E2zraAEfKpCBWxnjTVQv9bfVxeLWw/WezXUA754IXuB
LQ53r3gVctnnKWMu2roXe26DGVjvoJn/UjR1owa/opFE0dzCoegwWnDJXRth4iJa
VwIDAQAB
-----END PUBLIC KEY-----

Add

There are no API Keys for this User.

Add Public Key



14. Copy the Fingerprint after uploading the API Key. This is the API private PEM key's fingerprint which we will use later on along with the OCIDs that we had captured in Exercise 2.

ORACLE
Oracle Cloud Infrastructure


TENANCY
gsebmcs00008

REGION
us-phoenix-1

demo.user26 ▾ Support Documentation

Home Identity Compute Database Networking Storage Audit

Identity » Users » User Details


ACTIVE

demo.user26

Description: Demo user 26

Create/Reset Password Unblock Delete Apply Tag(s)

User Information Tags

OCID: ocid1.user.oc1..aaaaaaarhq435pn6vakcwq6ixltcw2vqh56jr7smzfmil2xl22h1bi7khoa [Hide](#) [Copy](#) Status: Active

Created: Sat, 10 Dec 2016 00:39:51 GMT

Resources

API Keys (1)

Swift Passwords (0)


Amazon S3 Compatibility API Keys (0)

Groups

API Keys

Displaying 1 API Keys

Add Public Key

 Fingerprint: 17:e8:fb:93:17:29:5c:a7:80:e6:2a:54:3fa8:93:48 Time Created: Fri, 09 Feb 2018 10:50:20 GMT [Delete](#)

ORACLE

Exercise 5: Initialize, Plan, Apply and Destroy the single instance terraform template.

In this exercise the single instance terraform script will be examined and run. The single instance compute exemplifies a number of terraform properties, capabilities and idiosyncrasies in use. Remember that with terraform, all the .tf files in the directory are treated as one file. The separation is to keep things organized.

NOTE: The details about the various files located under the ..\work\...\examples\compute folder are mentioned at the end of the exercise.

1. The variables for which the values need to be provided are listed below, please go to your cloud portal and copy the respective values from the portal:

```
$env:TF_VAR_tenancy_ocid="PASTE THE TENANCY OCID HERE"
$env:TF_VAR_user_ocid="PASTE THE USER OCID HERE"
$env:TF_VAR_fingerprint="PASTE THE FINGERPRINT OF THE API KEY HERE"
$env:TF_VAR_private_key_path="C:\Users\opc\oracleoci\oci_api_key.pem"
$env:TF_VAR_region="PASTE THE REGION HERE"
$env:TF_VAR_compartment_ocid="PASTE THE OCID OF THE COMPARTMENT"
$env:TF_VAR_ssh_public_key=Get-Content C:\Users\opc\ssh\id_rsa.pub -Raw
$env:TF_VAR_ssh_private_key=Get-Content C:\Users\opc\ssh\id_rsa -Raw
```

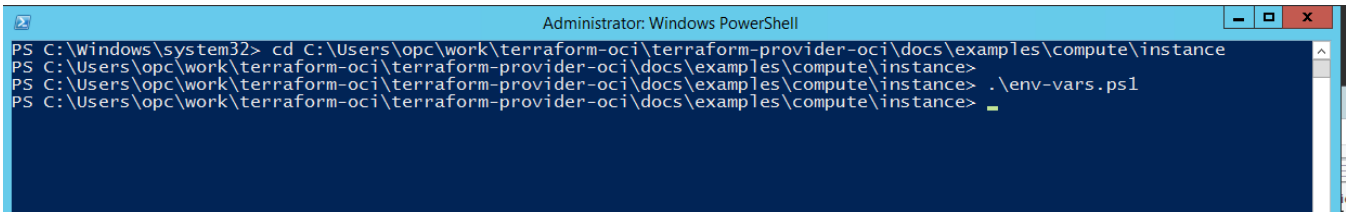
2. Go to the work directory (C:\Users\opc\work\terraform-OCI\terraform-provider-oci\docs\examples\compute) that contains our sample Terraform script and setup our environment file.
3. Edit **env-vars.ps1** (if this file exists as env-vars, save it as env-vars.ps1) Populate with the appropriate environment variables, key locations and OCIDs for your environment as I have done below:

```
### Authentication details
$env:TF_VAR_tenancy_ocid="ocid1.tenancy.oc1..aaaaaaaoykjamqzgafayqjrz5cpnuob4dkwjfkjxtzos2l5xcgyhog4sa"
$env:TF_VAR_user_ocid="ocid1.user.oc1..aaaaaaa6sg3vx6rtan4tprro723nzy3z4jfm3ehipcdo7rb2u224l2vvq"
$env:TF_VAR_fingerprint="a5:58:84:aa:97:a5:c2:1b:5a:ed:01:f9:fd:74:1a:bb"
$env:TF_VAR_private_key_path="C:\Users\opc\oracleoci\oci_api_key.pem"
### Region
$env:TF_VAR_region="us-ashburn-1"
### Compartment
$env:TF_VAR_compartment_ocid="ocid1.tenancy.oc1..aaaaaaaoykjamqzgafayqjrz5cpnuob4dkwjfkjxtzos2l5xcgyhog4sa"
### Public/private keys used on the instance
$env:TF_VAR_ssh_public_key=Get-Content C:\Users\opc\ssh\id_rsa.pub -Raw
$env:TF_VAR_ssh_private_key=Get-Content C:\Users\opc\ssh\id_rsa -Raw
```

Note: The above env-vars.ps1 is just a sample from my environment for your ease of reference, please do not use the same values as shown above

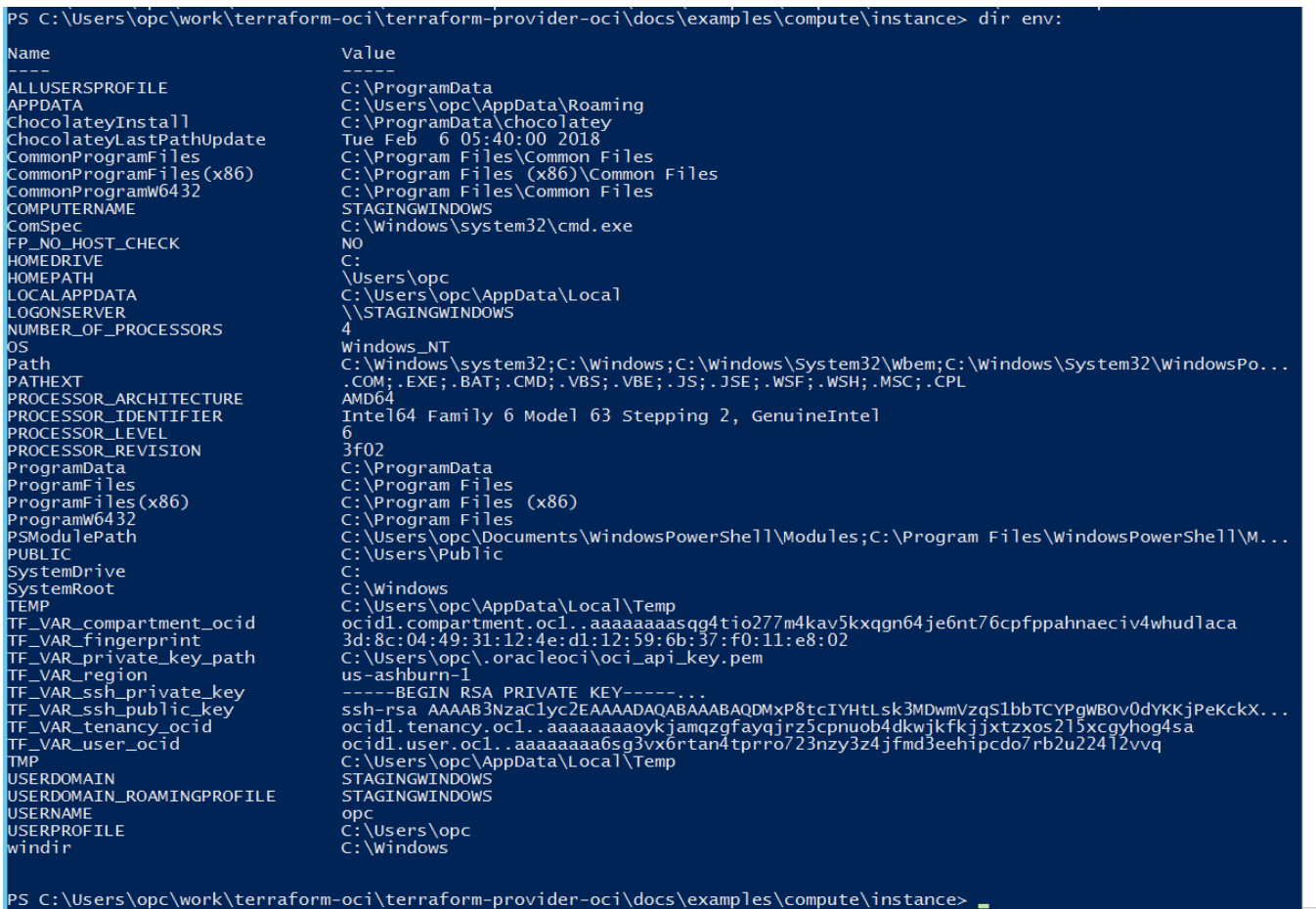
4. Source the variables into PowerShell's environment and check they are set correctly. The public and private keys will be read into their respective variables. Execute the PS file to make sure the environment variables are set.

```
PS > cd C:\Users\opc\work\terraform-oci\terraform-provider-oci\docs\examples\compute\instance
.\env-vars.ps1
```



```
Administrator: Windows PowerShell
PS C:\Windows\system32> cd C:\Users\opc\work\terraform-oci\terraform-provider-oci\docs\examples\compute\instance
PS C:\Users\opc\work\terraform-oci\terraform-provider-oci\docs\examples\compute\instance> .\env-vars.ps1
PS C:\Users\opc\work\terraform-oci\terraform-provider-oci\docs\examples\compute\instance>
```

```
PS > dir env:
```



```
PS C:\Users\opc\work\terraform-oci\terraform-provider-oci\docs\examples\compute\instance> dir env:
Name
----
Value
-----
ALLUSERSPROFILE      C:\ProgramData
APPDATA               C:\Users\opc\AppData\Roaming
ChocolateyInstall     C:\ProgramData\chocolatey
ChocolateyLastPathUpdate  Tue Feb  6 05:40:00 2018
CommonProgramFiles   C:\Program Files\Common Files
CommonProgramFiles(x86) C:\Program Files (x86)\Common Files
CommonProgramW6432   C:\Program Files\Common Files
COMPUTERNAME          STAGINGWINDOWS
ComSpec               C:\Windows\system32\cmd.exe
FP_NO_HOST_CHECK      NO
HOMEDRIVE             C:
HOMEPATH              \Users\opc
LOCAL APPDATA          C:\Users\opc\AppData\Local
LOGONSERVER            \\STAGINGWINDOWS
NUMBER_OF_PROCESSORS  4
OS                    Windows_NT
Path                  C:\Windows\system32;C:\Windows;C:\Windows\System32\wbem;C:\Windows\System32\WindowsPo...
PATHEXT               .COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC;.CPL
PROCESSOR_ARCHITECTURE AMD64
PROCESSOR_IDENTIFIER  Intel64 Family 6 Model 63 Stepping 2, GenuineIntel
PROCESSOR_LEVEL        6
PROCESSOR_REVISION    3f02
ProgramData           C:\ProgramData
ProgramFiles           C:\Program Files
ProgramFiles(x86)      C:\Program Files (x86)
ProgramW6432           C:\Program Files
PSModulePath           C:\Users\opc\Documents\WindowsPowerShell\Modules;C:\Program Files\WindowsPowerShell\M...
PUBLIC                 C:\Users\Public
SystemDrive            C:
SystemRoot             C:\Windows
TEMP                  C:\Users\opc\AppData\Local\Temp
TF_VAR_compartment_ocid ocid1.compartment.oc1..aaaaaaaasqg4tio277m4kav5kxqgn64je6nt76cpfpahnaeciv4whudlaca
TF_VAR_fingerprint    3d:8c:04:49:31:12:4e:d1:12:59:6b:37:f0:11:e8:02
TF_VAR_private_key_path C:\Users\opc\.oracleoci\oci_api_key.pem
TF_VAR_region         us-ashburn-1
TF_VAR_ssh_private_key -----BEGIN RSA PRIVATE KEY-----...
TF_VAR_ssh_public_key ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDMxP8tcIYHtLsk3MDwmVzqS1bbTCYPgWBOv0dYKKjPeKckX...
TF_VAR_tenancy_ocid   ocid1.tenancy.oc1..aaaaaaaoykjamqzgfayqjr25cpnuob4dkwjfkjxztzxs215xcgyhog4sa
TF_VAR_user_ocid      ocid1.user.oc1..aaaaaaa6sg3vx6rtan4tprro723nzy3z4jfm3eehipcdo7rb2u22412vvq
TMP                   C:\Users\opc\AppData\Local\Temp
USERDOMAIN             STAGINGWINDOWS
USERDOMAIN_ROAMINGPROFILE STAGINGWINDOWS
USERNAME               opc
USERPROFILE            C:\Users\opc
windir                 C:\Windows
PS C:\Users\opc\work\terraform-oci\terraform-provider-oci\docs\examples\compute\instance>
```

5. Initialize Terraform:

```
PS > terraform init
```



```
PS C:\Users\opc\work\terraform-oci\terraform-provider-oci\docs\examples\compute\instance> terraform init
```

```
Initializing provider plugins...
```

```
- Checking for available provider plugins on https://releases.hashicorp.com...
```

```
- Downloading plugin for provider "null" (1.0.0)...
```

```
The following providers do not have any version constraints in configuration,  
so the latest version was installed.
```

```
To prevent automatic upgrades to new major versions that may contain breaking  
changes, it is recommended to add version = "..." constraints to the  
corresponding provider blocks in configuration, with the constraint strings  
suggested below.
```

```
* provider.null: version = "~> 1.0"
```

```
Terraform has been successfully initialized!
```

```
You may now begin working with Terraform. Try running "terraform plan" to see  
any changes that are required for your infrastructure. All Terraform commands  
should now work.
```

```
If you ever set or change modules or backend configuration for Terraform,  
rerun this command to reinitialize your working directory. If you forget, other  
commands will detect it and remind you to do so if necessary.
```

```
PS C:\Users\opc\work\terraform-oci\terraform-provider-oci\docs\examples\compute\instance> _
```

6. Give terraform a test from PowerShell and verify "terraform plan" is working

```
PS > terraform plan
```

The Oracle logo, consisting of the word "ORACLE" in white capital letters on a red rectangular background.

```

PS C:\Users\opc\work\terraform-oci\terraform-provider-oci\docs\examples\compute\instance> terraform plan
Refreshing Terraform state in-memory prior to plan...
The refreshed state will be used to calculate this plan, but will not be
persisted to local or remote state storage.
data.oci_identity_availability_domains.ADs: Refreshing state...

-----
An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
+ create
~ update in-place
<= read (data resources)
Terraform will perform the following actions:

~ data.oci_core_vnic.InstanceVnic
  id: "" => <computed>
  availability_domain: "" => <computed>
  compartment_id: "" => <computed>
  display_name: "" => <computed>
  hostname_label: "" => <computed>
  is_primary: "" => <computed>
  mac_address: "" => <computed>
  private_ip_address: "" => <computed>
  public_ip_address: "" => <computed>
  skip_source_dest_check: "" => <computed>
  state: "" => <computed>
  subnet_id: "" => <computed>
  time_created: "" => <computed>
  vnic_id: "" => <computed>
  attachments[0].\vnic_id\": "${lookup(data.oci_core_vnic_attachments.InstanceVnics.vnic_at
hancements[0],\vnic_id\")}"

<= data.oci_core_vnic_attachments.InstanceVnics
  id: <computed>
  availability_domain: "RViG:US-ASHBURN-AD-1"
  compartment_id: "ocidl.compartment.oc1..aaaaaaaasqg4tio277m4kav5kxqgn64je6nt76cpfppa
hnaeciv4whudlaca"
  instance_id: "${oci_core_instance.TFInstance.id}"
  vnic_attachments.#: <computed>

+ null_resource.remote-exec
  id: <computed>

+ oci_core_instance.TFInstance
  id: <computed>
  availability_domain: "RViG:US-ASHBURN-AD-1"
  compartment_id: "ocidl.compartment.oc1..aaaaaaaasqg4tio277m4kav5kxqgn64je6nt76cpfppa
hnaeciv4whudlaca"
  create_vnic_details.#: "1"
  create_vnic_details.0.assign_public_ip: "true"
  create_vnic_details.0.display_name: "primaryvnic"
  create_vnic_details.0.hostname_label: "tfexampleinstance"

```

At this point the system is ready to terraform.

- Once the plan looks good, run apply. Lots of output will scroll by. The result of all this is captured in tfstate at the end of the run. This will actually provision the resources mentioned in the template/scripts.

PS > terraform apply

NOTE: Terraform does not do rollbacks however the state file is updated so a failure mid run must be remedied and re-applied to complete. Note that if any variable that is needed is not set and does not have a default value you will be prompted here for any variables needed to complete the run.

```
PS C:\Users\opc\work\terraform-oci\terraform-provider-oci\docs\examples\compute\instance> terraform apply
data.oci_identity_availability_domains.ADs: Refreshing state...

An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
  + create
  ~ update in-place
  <= read (data resources)

Terraform will perform the following actions:

~ data.oci_core_vnic.InstanceVnic
  id: "" => <computed>
  availability_domain: "" => <computed>
  compartment_id: "" => <computed>
  display_name: "" => <computed>
  hostname_label: "" => <computed>
  is_primary: "" => <computed>
  mac_address: "" => <computed>
  private_ip_address: "" => <computed>
  public_ip_address: "" => <computed>
  skip_source_dest_check: "" => <computed>
  state: "" => <computed>
  subnet_id: "" => <computed>
  time_created: "" => <computed>
  vnic_id: "" => "${lookup(data.oci_core_vnic_attachments.InstanceVnics.vnic_
attachments[0],\"vnic_id\")}"

<= data.oci_core_vnic_attachments.InstanceVnics
  id: <computed>
  availability_domain: "RVIG:US-ASHBURN-AD-1"
  compartment_id: "ocidl.compartment.oc1..aaaaaaaasqg4tio277m4kav5kxqgn64je6nt76cpfp
hnaeciy4whudlaca"
  instance_id: "${oci_core_instance.TFInstance.id}"
  vnic_attachments.#: <computed>

null_resource.remote-exec (remote-exec): Connecting to remote host via SSH...
null_resource.remote-exec (remote-exec): Host: 129.213.101.156
null_resource.remote-exec (remote-exec): User: opc
null_resource.remote-exec (remote-exec): Password: false
null_resource.remote-exec (remote-exec): Private key: true
null_resource.remote-exec (remote-exec): SSH Agent: false
null_resource.remote-exec: Still creating... (1m20s elapsed)
null_resource.remote-exec (remote-exec): Connecting to remote host via SSH...
null_resource.remote-exec (remote-exec): Host: 129.213.101.156
null_resource.remote-exec (remote-exec): User: opc
null_resource.remote-exec (remote-exec): Password: false
null_resource.remote-exec (remote-exec): Private key: true
null_resource.remote-exec (remote-exec): SSH Agent: false
null_resource.remote-exec: Still creating... (1m30s elapsed)
null_resource.remote-exec (remote-exec): Connecting to remote host via SSH...
null_resource.remote-exec (remote-exec): Host: 129.213.101.156
null_resource.remote-exec (remote-exec): User: opc
null_resource.remote-exec (remote-exec): Password: false
null_resource.remote-exec (remote-exec): Private key: true
null_resource.remote-exec (remote-exec): SSH Agent: false
null_resource.remote-exec (remote-exec): Connected!
null_resource.remote-exec (remote-exec): New iSCSI node [tcp:[hw=,ip=,net_if=,iscsi_if=default] 169.254.2.2,3260,-1 iqn
2015-12.com.oracleiaas:00d31e5a-526f-4116-8c68-a4297c1e3bfc] added
null_resource.remote-exec: Creation complete after 1m38s (ID: 875036295631838689)

Apply complete! Resources: 5 added, 0 changed, 2 destroyed.

Outputs:
InstancePrivateIP = [
  10.1.20.2
]
InstancePublicIP = [
  129.213.101.156
]
```

8. Once the run has completed we can check out the contents of the tfstate file:

PS > cat terraform.tfstate

ORACLE

Here there will be everything terraform needs to understand the current state of the infrastructure.

```
PS C:\Users\opc\work\terraform-oci\terraform-provider-oci\docs\examples\compute\instance> cat terraform.tfstate
{
  "version": 3,
  "terraform_version": "0.11.2",
  "serial": 2,
  "lineage": "0303634f-57aa-4cd1-8ba3-f06786bb7c06",
  "modules": [
    {
      "path": [
        "root"
      ],
      "outputs": {
        "InstancePrivateIP": {
          "sensitive": false,
          "type": "list",
          "value": [
            "10.1.20.2"
          ]
        },
        "InstancePublicIP": {
          "sensitive": false,
          "type": "list",
          "value": [
            "129.213.101.156"
          ]
        }
      },
      "resources": {
        "data.oci_core_vnic.InstanceVnic": {
          "type": "oci_core_vnic",
          "depends_on": [
            "data.oci_core_vnic_attachments.InstanceVnics"
          ]
        }
      }
    }
  ]
}
```

9. To destroy the resources created:

PS > terraform destroy

```
oci_core_subnet.ExampleSubnet: Still destroying... (ID: ocid1.subnet.oc1.iad.aaaaaaaas2hk3gxcg2j...mykh5dg6atieks5uwyuzc2hvbqq4g7djvhhxcpa, 10s elapsed)
oci_core_subnet.ExampleSubnet: Still destroying... (ID: ocid1.subnet.oc1.iad.aaaaaaaas2hk3gxcg2j...mykh5dg6atieks5uwyuzc2hvbqq4g7djvhhxcpa, 20s elapsed)
oci_core_subnet.ExampleSubnet: Destruction complete after 26s
oci_core_route_table.ExampleRT: Destroying... (ID: ocid1.routetable.oc1.iad.aaaaaaaq6ea3w...se65cxzztmhbxys1kvoxfgc65fr26cq7f6jppq)
oci_core_route_table.ExampleRT: Still destroying... (ID: ocid1.routetable.oc1.iad.aaaaaaaq6ea3w...se65cxzztmhbxys1kvoxfgc65fr26cq7f6jppq, 10s elapsed)
oci_core_route_table.ExampleRT: Destruction complete after 17s
oci_core_internet_gateway.ExampleIG: Destroying... (ID: ocid1.internetgateway.oc1.iad.aaaaaaa...isuFhuk227cfkb5hvbe23ococljwhi6v7rfheq)
oci_core_internet_gateway.ExampleIG: Destruction complete after 1s
oci_core_virtual_network.ExampleVCN: Destroying... (ID: ocid1.vcn.oc1.iad.aaaaaaaaxyvdf5sbyumeschydujk6kjceqkps6ivppxagjrbkjlymiz5icqg)
oci_core_virtual_network.ExampleVCN: Destruction complete after 2s
Destroy complete! Resources: 8 destroyed.
```

ORACLE

The files included are:

- **env-vars** - if environment variables are not used to set each of the listed variables these then sourcing this file (or the work-alike env-vars.ps1 shown in exercise 3A) can be used to set them. Most of these variables are authentication related. These typically are not checked in with the rest of the code.
- **variables.tf** - Any variables that are used must be declared before use. Where the variables are declared a default value can be defined. These values are always over-rideable with environment, passing from the root module to modules or with a tfvar file.
- **provider.tf** - This file is where the Oracle OCI provider is used with the basic required authentication (tenancy ocid, user ocid, API private key, and API private key fingerprint). This enables OCI-specific capabilities and an understanding of OCI dependencies within terraform. Providers implement resources. Resources are "pieces of infrastructure" and their dependencies.
- **compute.tf** - This is the file which templates some of the basic resources for the instance. This includes the availability domain, compartment ID, display name, hostname, subnet, image and shape for the instance. Within compute.tf is also passing in our public key. For most images this key is passed to root's authorized keys, along with the user "opc". For Ubuntu, rather than using the opc user, the username ubuntu is used. In addition to the public key injection, there is a bootstrap script which is passed to the image as a uuencoded string. The plain text of this file is in .userdata/bootstrap. There is a timeout defined here as well.
- **./userdata/bootstrap** - The boot strap file is a startup script used in this example (compute.tf):

```
#!/bin/bash
yum update -y
```

This is a very basic bootstrap. One item that could be added that is handy in a script like this is "yum -y makecache fast" before the update line - this would rebuild the package cache. The script by default runs as root from the root directory.

- **datasources.tf** - defines the data sources used in the configuration - availability domains, images, network interfaces, etc. Providers are responsible for defining and implementing data sources. A resource causes Terraform to create and manage a new infrastructure component, data sources are read-only views into pre-existing data.
- **outputs.tf** - The simplest terraform tf would be a single output statement, eg,


```
"hello.tf"
output "hw" {value = "Hello World"}.
```

In this file there is:

```
output "InstancePrivateIP" { value = ["${data.oci_core_vnic.InstanceVnic.private_ip_address}"] }
output "InstancePublicIP" { value = ["${data.oci_core_vnic.InstanceVnic.public_ip_address}"] }
```

This will show the private and public IPs at the end of a run.

- **remote-exec.tf** - a "remote-exec" in terraform is a provisioner that invokes a script on a remote resource after it is created. This can be used to run a configuration management tool, bootstrap into a cluster, etc. The remote-exec provisioner supports both SSH and Winrm type connections. From this file there is a directive to connect to the instance created in compute.tf and attempt an SSH login over the course of 10 minutes. Once the login is completed a file is made in opc's home directory, /home/opc/IMadeAFileRightHere. Another example of a provisioner (vs the remote-exec used here) is chef. **chef can be invoked in a similar fashion to configure instances and software after the infrastructure is created by terraform.**
- **block.tf** - An example of creating a block volume and attaching it to an instance via iscsi.

- 
- **terraform.tfstate** - After a terraform apply (or refresh) the state of the provisioned infrastructure is held in a tfstate file. This is the amalgamation of all the files and the created state all in one shot. These state files can grow to be quite large and complex for a complex infrastructure. They are also the way of "locking" the infrastructure so if there are multiple users of a terraformed infrastructure care must be taken to prevent collisions. HashiCorp has enterprise services such as Atlas and Nomad to help manage these state files and maintenance of infrastructure by teams. There are also tools such as Terragrunt (<https://github.com/gruntwork-io/terragrunt>) which facilitate team built infrastructure on terraform. Given the nature of the data within terraform.tfstate care should be taken to secure the state. One thing to keep in mind about terraform is it does not import well / at all, so losing the state file means the infrastructure will likely need to be rebuilt.