

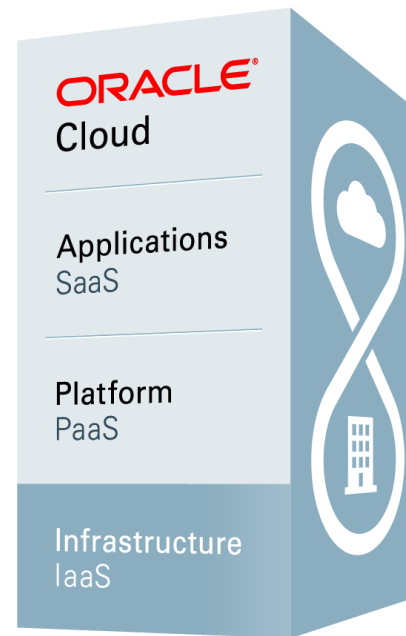
# GIT OCI Training

Vagrant Demo – Terraform

Rich Lewis

Oracle GIT ENS  
January 16, 2018

# Oracle Cloud Infrastructure - OCI





Search



Overview

My Wall

Flags

Favorites

**Conversations**

Collections

People

Groups

**GIT OCI Training****Messages**

Documents

Referring



16



0



Find:

0

0

0

**Richard Lewis**

2017/11/27 at 03:10 PM

Creating OSN Group to discuss GIT OCI Training covering the following suggested skills:

**Code Training -**

1. Shell Script ([#bash](#), linux, vagrant, git)
2. [#python](#) (terraform, linux)
3. [#ruby](#) (chef, linux)

**Tool Training -**

1. [#vagrant](#) / [#virtualbox](#)
2. [#git](#)
3. [#chef](#)
4. [#jenkins](#)
5. [#terraform](#)

**Process Training -**

1. Software Development Lifecycle

Post a message to the Conversation.

**Recent Activity**

**Stuart Dairiki** The EPTool DB successfully upgraded to 12c.  
1h ago [EP Tool Production](#)



**Timothy Daly** Hello, I really need help with my customer who runs ATG. The customer has been running it for over 10 years, and need help identifying...

**Trending**[#wellness](#)[#Terraform](#)**Contacts**

**Edward Mason**  
Offline



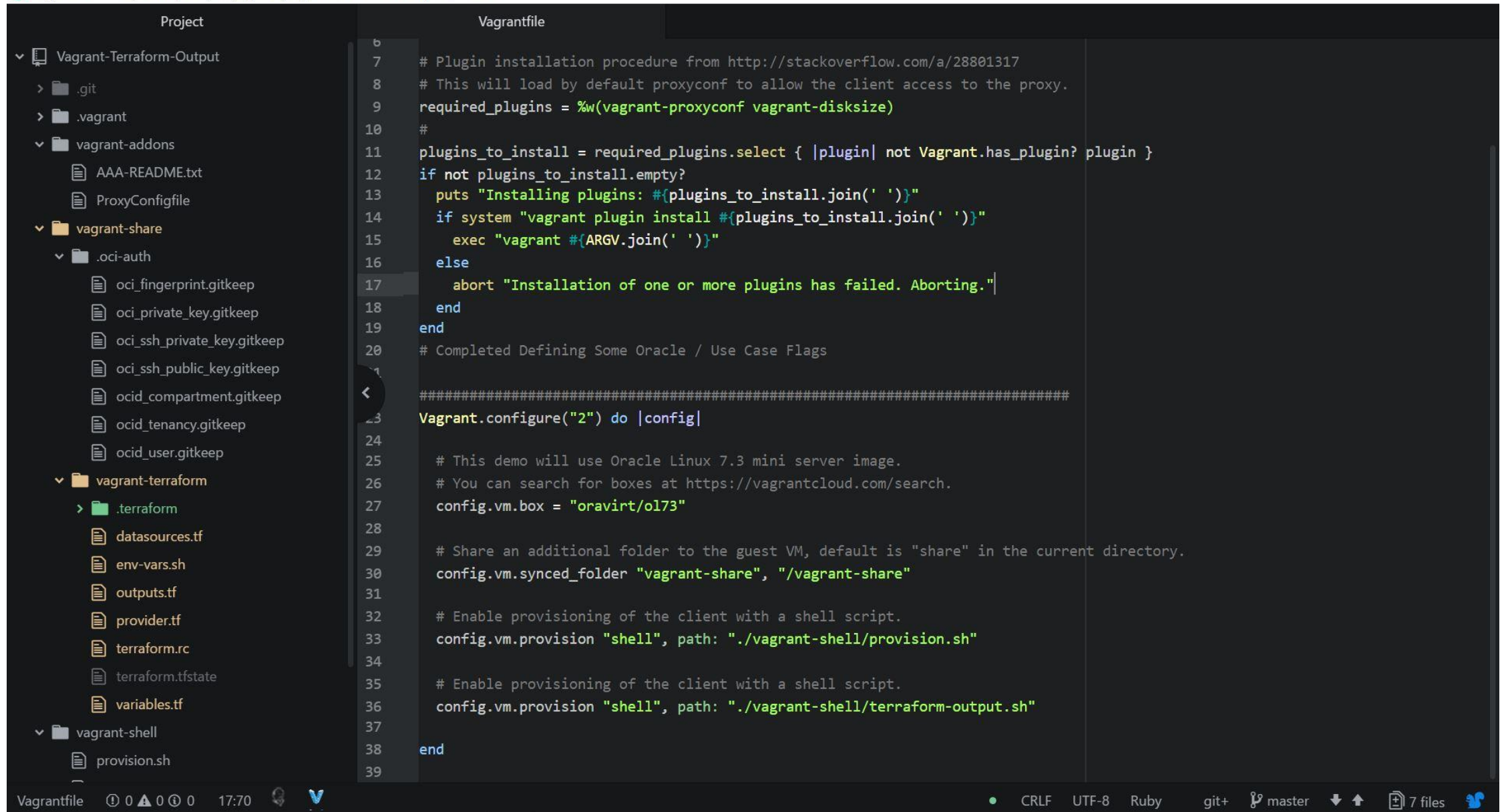
**Greg Green**  
Offline



**kevin.reynolds@oracle.com**  
Offline

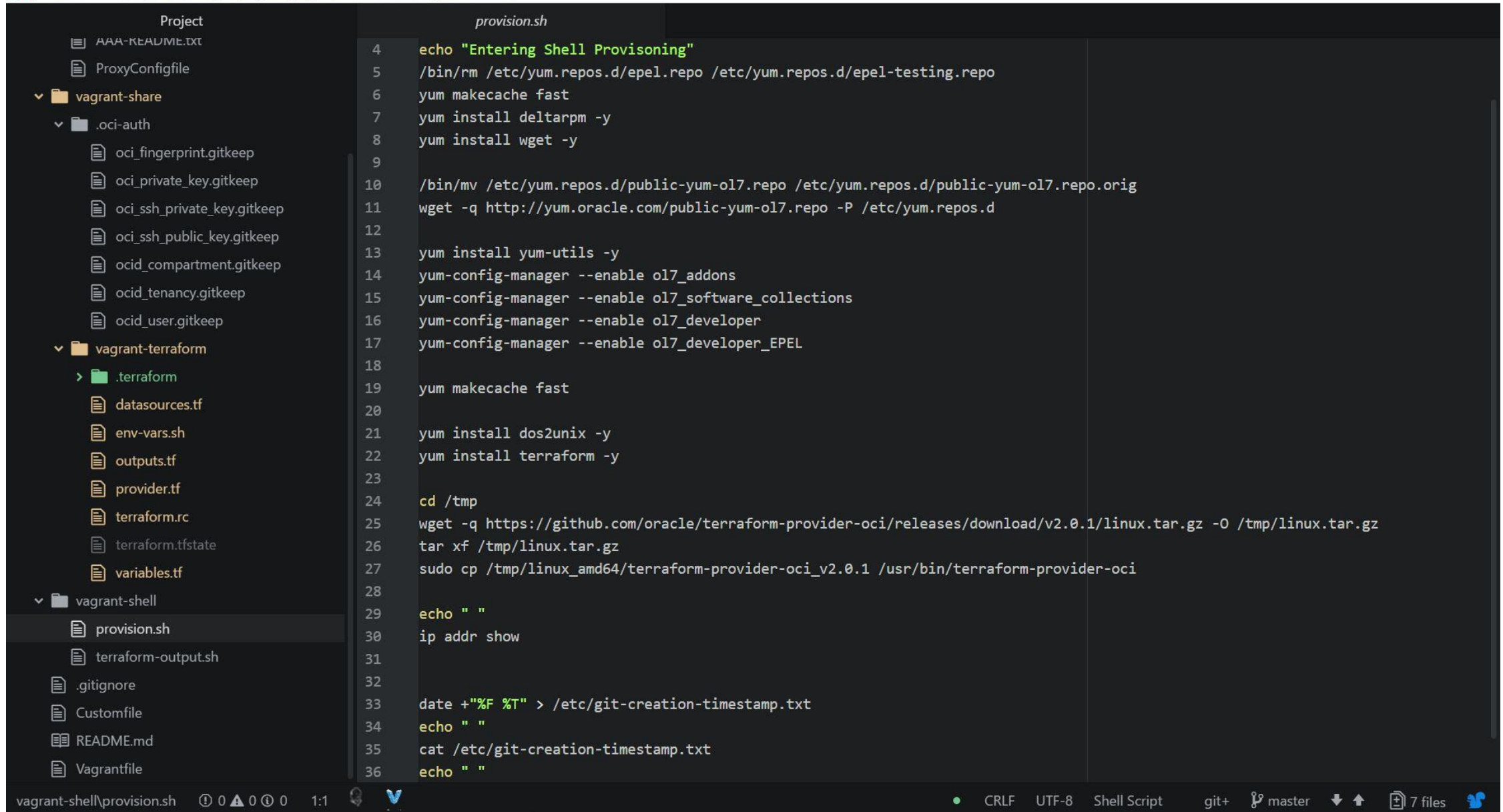
# Vagrant Terraform

- Configuration via VagrantConfigFile
  - Proxy Configuration
  - Download the Oracle Linux 7 Client
  - Share folder ./vagrant-share
  - Provision instance with provision.sh script
- Execute Terraform
  - Plan is in /vagrant-share/vagrant-terraform/terraform.rc
  - Authentication Samples requiring config are in /vagrant-share/.oci-auth/\*.gitkeep



```
6
7 # Plugin installation procedure from http://stackoverflow.com/a/28801317
8 # This will load by default proxyconf to allow the client access to the proxy.
9 required_plugins = %w(vagrant-proxyconf vagrant-disksize)
10 #
11 plugins_to_install = required_plugins.select { |plugin| not Vagrant.has_plugin? plugin }
12 if not plugins_to_install.empty?
13   puts "Installing plugins: #{plugins_to_install.join(' ')}"
14   if system "vagrant plugin install #{plugins_to_install.join(' ')}"
15     exec "vagrant #{ARGV.join(' ')}"
16   else
17     abort "Installation of one or more plugins has failed. Aborting."
18   end
19 end
20 # Completed Defining Some Oracle / Use Case Flags
21
22 #####
23 Vagrant.configure("2") do |config|
24
25   # This demo will use Oracle Linux 7.3 mini server image.
26   # You can search for boxes at https://vagrantcloud.com/search.
27   config.vm.box = "oravirt/ol73"
28
29   # Share an additional folder to the guest VM, default is "share" in the current directory.
30   config.vm.synced_folder "vagrant-share", "/vagrant-share"
31
32   # Enable provisioning of the client with a shell script.
33   config.vm.provision "shell", path: "./vagrant-shell/provision.sh"
34
35   # Enable provisioning of the client with a shell script.
36   config.vm.provision "shell", path: "./vagrant-shell/terraform-output.sh"
37
38 end
39
```





The image shows the Atom text editor interface. On the left is a file explorer pane showing a project structure. The main editor pane displays the content of the file `provision.sh`. The script performs the following actions:

- Line 4: Prints "Entering Shell Provisioning"
- Line 5: Removes the file `/etc/yum.repos.d/epel.repo`
- Line 6: Runs `yum makecache fast`
- Line 7: Installs `deltarpm` with `yum install deltarpm -y`
- Line 8: Installs `wget` with `yum install wget -y`
- Line 9: Blank line
- Line 10: Moves the file `/etc/yum.repos.d/public-yum-ol7.repo` to `/etc/yum.repos.d/public-yum-ol7.repo.orig`
- Line 11: Downloads a repository file from `http://yum.oracle.com/public-yum-ol7.repo` to `/etc/yum.repos.d` using `wget`
- Line 12: Blank line
- Line 13: Installs `yum-utils` with `yum install yum-utils -y`
- Line 14: Enables `ol7_addons` in yum-config-manager
- Line 15: Enables `ol7_software_collections` in yum-config-manager
- Line 16: Enables `ol7_developer` in yum-config-manager
- Line 17: Enables `ol7_developer_EPEL` in yum-config-manager
- Line 18: Blank line
- Line 19: Runs `yum makecache fast`
- Line 20: Blank line
- Line 21: Installs `dos2unix` with `yum install dos2unix -y`
- Line 22: Installs `terraform` with `yum install terraform -y`
- Line 23: Blank line
- Line 24: Changes directory to `/tmp` with `cd /tmp`
- Line 25: Downloads the terraform provider OCI release from GitHub
- Line 26: Extracts the tarball with `tar xf /tmp/linux.tar.gz`
- Line 27: Copies the provider binary to `/usr/bin/terraform-provider-oci` with `sudo cp`
- Line 28: Blank line
- Line 29: Prints a blank line with `echo " "`
- Line 30: Shows IP address with `ip addr show`
- Line 31: Blank line
- Line 32: Blank line
- Line 33: Writes the current date and time to `/etc/git-creation-timestamp.txt`
- Line 34: Prints a blank line with `echo " "`
- Line 35: Displays the contents of `/etc/git-creation-timestamp.txt` with `cat`
- Line 36: Prints a blank line with `echo " "`

The status bar at the bottom indicates the file is a Shell Script in UTF-8 encoding, with CRLF line endings. It also shows the git repository is on the master branch and contains 7 files.

Project

AAA-README.TXT

ProxyConfigfile

vagrant-share

.oci-auth

oci\_fingerprint.gitkeep

oci\_private\_key.gitkeep

oci\_ssh\_private\_key.gitkeep

oci\_ssh\_public\_key.gitkeep

ocid\_compartment.gitkeep

ocid\_tenancy.gitkeep

ocid\_user.gitkeep

vagrant-terraform

.terraform

datasources.tf

env-vars.sh

outputs.tf

provider.tf

terraform.rc

terraform.tfstate

variables.tf

vagrant-shell

provision.sh

terraform-output.sh

.gitignore

Customfile

README.md

Vagrantfile

terraform-output.sh

1

#!/usr/bin/env bash

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

```

echo " "
echo "Executing /vagrant-share/vagrant-terraform/Terraform"

echo "--Remove any trailing \r dos characters--"
cd /vagrant-share
for i in `find . -type f -print | file -f - | grep -v binary | grep -v gitkeep | grep CRLF | sed -e "s/\:.*$/\""; do
  dos2unix ${i}
done

echo "--Setup the environment variables--"
cd /vagrant-share/vagrant-terraform
source /vagrant-share/vagrant-terraform/env-vars.sh

printenv | \
grep -i TF_VAR | \
sed '/ocid=/s/\:.\.(\.\\{10\\}).*/\:.\.\\1\|*****/' | \
sed '/fingerprint=/s/=\\(\.\\{15\\}).*/=\\1*:*:*:*:*:*:*:*:*:*:*/' | \
sort

echo "--Run terraform init--"
terraform init
echo "--Run terraform plan--"
terraform plan
echo "--Run terraform apply--"
terraform apply

echo "Terminating /vagrant-share/vagrant-terraform/Terraform"

```

vagrant-shell\terraform-output.sh

0

0

0

0

1:1

CRLF

UTF-8

Shell Script

git+

master

7 files

ORACLE®

Copyright © 2017, Oracle and/or its affiliates. All rights reserved. | Confidential – Oracle Internal/Restricted/Highly Restricted

7

# Pre-Requisites

1. Git - <https://git-scm.com/downloads>
2. Virtualbox - <https://www.virtualbox.org/wiki/Downloads>
3. Vagrant - <https://www.vagrantup.com/downloads.html>



# Setup

1. Create Training Directory
  - mkdir Training
  - cd Training
2. Clone the Demo from GitHub
  - git clone <https://github.com/richlewis469/Vagrant-Terraform-Output.git>
  - cd Vagrant-Terraform-Output
3. Configure vagrant-addons/ProxyConfigFile if necessary
4. Configure Authentication and OCID's in vagrant-share/.oci-auth/\*.gitkeep

Project	oci_fingerprint.gitkeep
1	fd:21:0a:c9:d3:**:**:**:**:**:**:**:**:**:**
2	

oci_private_key.gitkeep	
1	-----BEGIN RSA PRIVATE KEY-----
2	MIIEpAIBA*****
3	*****

oci_ssh_private_key.gitkeep	
1	-----BEGIN RSA PRIVATE KEY-----
2	MIIEpgIBA*****
3	*****

oci_ssh_public_key.gitkeep	
1	ssh-rsa AAAAB3NzaC*****
2	

ocid_compartment.gitkeep	
1	ocid1.tenancy.oc1..aaaaaaatj*****
2	

ocid_tenancy.gitkeep	
1	ocid1.tenancy.oc1..aaaaaaatj*****
2	

ocid_user.gitkeep	
1	ocid1.user.oc1..aaaaaaaai*****
2	

vagrant-share\oci-auth\ocid\_user.gitkeep

0 0 0 0

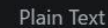
1:1



CRLF



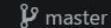
UTF-8



Plain Text



git+




master



7 files



# Demo

- 
4. This is an Automated Demo
  5. Bring up the Vagrant Environment
    - vagrant up
  7. Note the data output
  9. Destroy the Vagrant Environment and free up space.
    - vagrant destroy --force

# Integrated Cloud

## Applications & Platform Services

ORACLE®