

Practice: Install Terraform and Create an Instance by Using Terraform

Try this hands-on lab with the **Oracle Cloud 30 days Free Trial** account or your own tenancy. If you do not have a free account, click [here](#) to get one.


Tasks

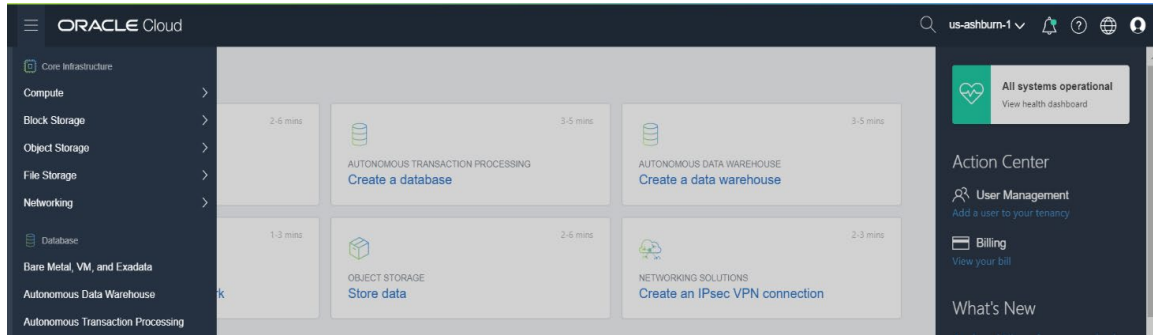
1. Log in to your [Oracle Cloud Free Tier Account](#).
2. Select the **Oracle Cloud Infrastructure Direct Sign-In** option, enter the cloud account **User Name** and **Password** assigned to you, and click **Sign In**.
 - **Important:** Do NOT use the “Single Sign-On (SSO)” option.

The screenshot shows the Oracle Cloud login page for tenant 'ocucictrng28'. It features two main login options:

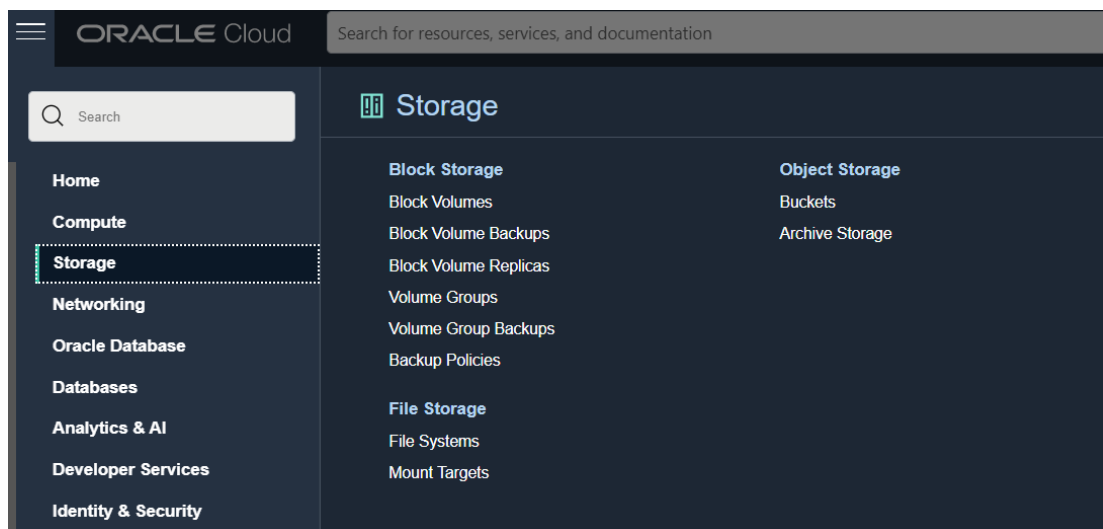
- Single Sign-On (SSO):** This option is highlighted with a red box. It includes a message: "We have detected that your tenancy has been federated to another Identity Provider. Select your Identity Provider below." Below this is a dropdown menu showing 'oracleidentitycloudservice' and a 'Continue' button. A red arrow points to this section with the text "Don't use this".
- Oracle Cloud Infrastructure Direct Sign-In:** This option is highlighted with a green box. It includes a message: "This login is uncommon for federated accounts. If you have questions, please review the FAQ or contact your tenancy administrator." Below this are input fields for 'User Name' and 'Password', and a 'Sign In' button. A green arrow points to this section with the text "Use this to Sign-In".

3. At this point, you should be logged in to **Oracle Cloud Infrastructure (OCI) Console**, also called the OCI home page or OCI web console.

4. Click **Menu**  in the top-left corner and explore the options available. You will use this navigation path through the practice.



5. To navigate to the page that displays networking resources, you need to click **Networking** and likewise for compute resources, the **Compute** menu option, and so on.
6. In preparation for upcoming practices, create a storage bucket and upload the lab files.
 - a. From the Main menu, select **Storage**, and then select **Buckets** in Object Storage:



- b. Select **Create Bucket** and provide a meaningful **bucket name**. Click **Create**.

Create Bucket

Bucket Name
Terraform_Binary

Default Storage Tier
☒ Standard
☐ Archive

The default storage tier for a bucket can only be specified during creation. Once set, you cannot change the storage tier in which a bucket resides. [Learn more](#)

☐ Enable Auto-Tiering
Automatically move infrequently accessed objects from the Standard tier to less expensive storage. [Learn more](#)

☐ Enable Object Versioning
Create an object version when a new object is uploaded, an existing object is overwritten, or when an object is deleted. [Learn more](#)

☐ Emit Object Events
Create automation based on object state changes using the [Events Service](#).

- c. Click the three dots in front of the bucket name. Click **Edit Visibility**.

Bucket Name	Storage Tier	Visibility	Created	Actions
Terraform_Binary	Standard	Private	Wed, Mar 11, 2020 10:10 AM	<div><div>View Bucket Details</div><div>Create Pre-Authenticated Request</div><div>Move Resource</div><div>Edit Visibility</div><div>Add Tags</div><div>View Tags</div><div>Delete</div></div>

- d. Select **Public** and click **Save Changes**.

Edit Visibility

Enabling public visibility will let anonymous and unauthenticated users access data stored in the bucket.

Visibility

☐ Private

☒ Public

☒ Allow users to list objects from this bucket

Consider using pre-authenticated requests instead

We recommend using pre-authenticated requests instead of public buckets. Pre-authenticated requests support additional authorization, expiry, and scoping capabilities not possible with public buckets. [Learn more](#)

Save Changes [Cancel](#)

- e. Lab files (that is, lab scripts) are available for download along with the Student Guide and Activity Guide as part of the Subscription course.
- f. Download the **lab.zip** lab file on your local system.
- g. Click the bucket name and click **Upload** in the Objects section.

Objects

<input type="checkbox"/>	Name	Last Modified

- h. Select files and upload the **terraform_0.15.3_linux_amd64.zip** and **compute.tf** files one after the other into the bucket. You downloaded these files as part of **lab.zip**.

Upload Objects

Object Name Prefix *Optional*

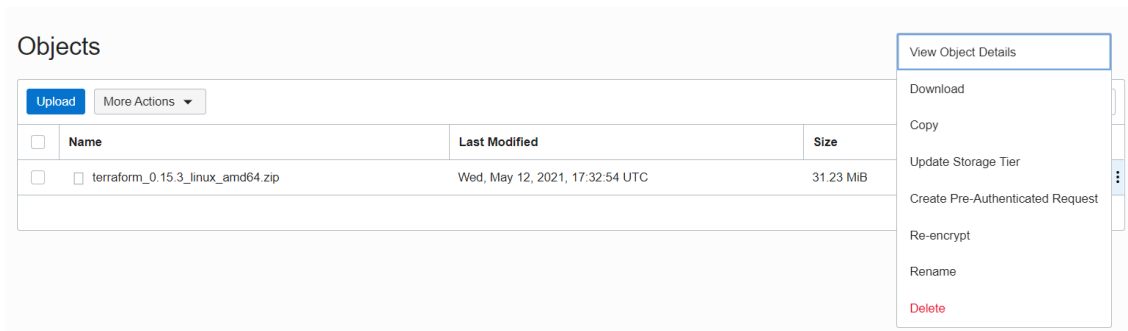
Storage Tier
Standard

Choose Files from your Computer

Drop files here or [select files](#)

terraform_0.15.3_linux_amd64.zip 31.23 MiB
1 files, 31.23 MiB total
[Show Optional Response Headers and Metadata](#)

- i. Click the three dots next to the uploaded file and select **View Object Details** from the menu.



- j. Copy the URL path shown in a Notepad for your reference in the next practice.

This completes the pre-requisite for the next practice.

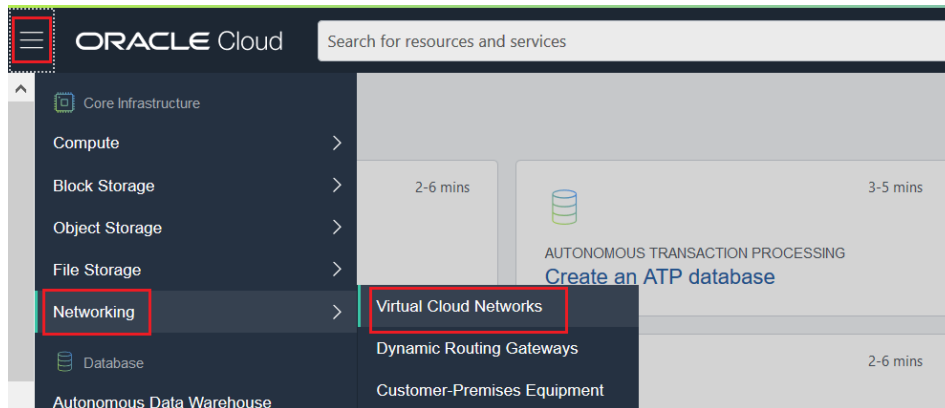
Practice: Creating a Virtual Cloud Network

Overview

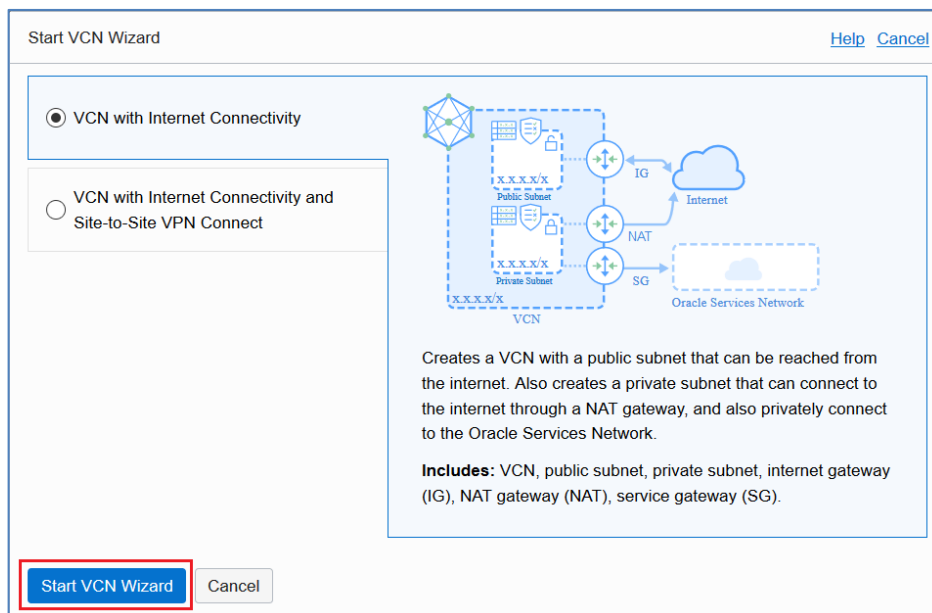
In this practice, we are going to create a VCN and its required resources.

Tasks

1. Log in to your [Oracle Account](#).
2. On the OCI console page, click **Menu, Networking, and Virtual Cloud Networks**.



3. On the Virtual Cloud Networks page, click **Start VCN Wizard**.
4. In the dialog box, select **VCN with Internet Connectivity**, and click **Start VCN Wizard**.



5. Provide the configuration information:

Create a VCN with Internet Connectivity

[Help](#)

1 Configuration

2 Review and Create

Configuration

Basic Information

VCN NAME ⓘ

COMPARTMENT ⓘ

99619031-C01

ocucicdmg28 (root)/99619031-C01

Configure VCN and Subnets

VCN CIDR BLOCK ⓘ

Example: 10.0.0.0/16
If you plan to peer this VCN with another VCN, the VCNs must not have overlapping CIDRs. [Learn more.](#)

PUBLIC SUBNET CIDR BLOCK ⓘ

Example: 10.0.0.0/24
The subnet CIDR blocks must not overlap.

PRIVATE SUBNET CIDR BLOCK ⓘ

Example: 10.0.1.0/24
The subnet CIDR blocks must not overlap.

DNS RESOLUTION

☒ USE DNS HOSTNAMES IN THIS VCN
Required for instance hostname assignment if you plan to use VCN DNS or a third-party DNS. This choice cannot be changed after the VCN is created. [Learn more.](#)

[Show Tagging Options](#)

VCN with Internet Connectivity

Includes:

- VCN
- Public subnet
- Private subnet
- Internet gateway (IG)
- NAT gateway (NAT)
- Service gateway (SG)

Previous

Next

Cancel

- VCN NAME – **OpsVCN**
- COMPARTMENT – Confirm that the correct compartment has been selected.
- VCN CIDR BLOCK – Enter **10.0.0.0/16**.
- PUBLIC SUBNET CIDR BLOCK – Enter **10.0.1.0/24**.
- PRIVATE SUBNET CIDR BLOCK – Enter **10.0.2.0/24**.

Click **Next**.

- Review and click **Create**. Many useful information is available on this page; review the information.

- 
7. The VCN is created along with resources: VCN, Public subnet, Private subnet, Internet gateway (IG).

Note: This option is the quickest way to get a working cloud network in the fewest steps.

Practice: Generate SSH Keys

Overview

Instances use an SSH key pair instead of a password to authenticate a remote user. A key pair file contains a private key and public key. You keep the private key on your computer and provide the public key every time you launch an instance. In this practice, you will generate SSH keys to be used later while launching an instance.

Tasks

1. Launch the Cloud Shell session.
2. Generate ssh-keys for your instance if you don't have one. If `id_rsa` and `id_rsa.pub` key pairs are present, they can be reused. By default, these are stored in `~/.ssh/`:

```
$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key
(/home/username/.ssh/id_rsa): <Press Enter>
Enter passphrase (empty for no passphrase): <Press Enter>
Enter same passphrase again: <Press Enter>
Your identification has been saved in
/home/username/.ssh/id_rsa.
Your public key has been saved in
/home/username/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:tAn6nKkcZDTXl/vXRAD/pfLzdmF5rQ2948MQgz5CWe8
The key's randomart image is:
+---[RSA 2048]---+
|                ...  |
|                .  o  |
|             o o o = o |
|          . + o * o + + |
|            +   S o o + = |
|         o o o . + E.*+ |
|          . =   . o B+ = |
|         . o           ..B+ |
|          o             .o = |
+-----[SHA256]-----+
```

3. Make sure permissions are restricted, else SSH fails if private keys have permissive permissions. Cloud Shell takes care of this. Verify using the following command, change the permission if required, and continue.

a. This is how the file permissions should look:

```
$ ls -l ~/.ssh
total 12

-rw-----. 1 x_99520482 oci 1679 Feb  4 09:13 id_rsa
-rw-r--r--. 1 x_99520482 oci  405 Feb  4 09:13 id_rsa.pub
-rw-r--r--. 1 x_99520482 oci  176 Feb  4 09:09 known_hosts
```

b. Use the following commands if you need to change the file permissions:

```
$ chmod 0700 ~/.ssh
$ chmod 0400 ~/.ssh/id_rsa
$ chmod 0644 ~/.ssh/id_rsa.pub
```

4. Copy the contents of `~/.ssh/id_rsa.pub`, that is, the SSH Public key value in a Notepad / the **LabFile1.txt** file. This is your Public SSH key to connect to the Compute instances. You will need this through the course; keep it handy.

```
$ cd ~/.ssh

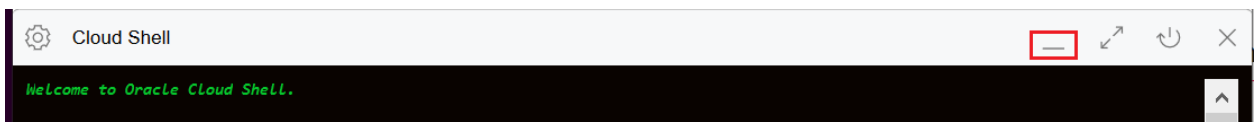
$ cat id_rsa.pub

ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQDYJdfGjh5/mZFGQFaN0z+2crXr4hCBjsZIQl
iHincmCxvAWQtBP6pMjox6nVk5Tzuc9Crcy+95PpqtjXCCs1BirZeSxjKK1vVWicSV
3isyAKGns3OdcttsqDpliF7gniplMk1a9q/oSubSl7bcQ6qMWWpaphRqGNJQN4DTwr
agLMYBKprPrK67fTxGUpBmsvvjfx1WHN4B5EWE0gl/LOecJN7W0jSbXgWoTEcHyp5M
HeXOA+nfwYqbl26rsTiMpsyzQb5kvUgfeTvV3eiQj1OSOcXYNF7R4nqs9KzUTZEs68
GQUA3TNKEfmzptuqX04o/fVcwp7chtV8DEUsWc5jUz lab_user16@90f4f8d1e20e
```

5. Record the availability domain name

```
$ oci iam availability-domain list
{
  ... ..
  "name": "AMVH:US-ASHBURN-AD-3"
  ... ..
}
```

6. Minimize the Cloud Shell terminal by clicking the minimize icon:



While you are still logged in to the OCI Console, you can bring this Cloud Shell terminal back any time you need it. Even if you log out, or if your session gets disconnected, you can launch Cloud Shell any time and your files will be intact for the duration of this course.

This completes the task of creating the SSH key.

Practice: Create a Terraform Server

Overview

In this practice, you will set up the Terraform Server.

Tasks

1. In the OCI Console, navigate to **MENU > Compute > Instances**. Click **Create Instance**.
2. Provide details:
 - **Name your instance:** **opsterraform**
 - **Compartment:** Ensure that the Compartment assigned to you is selected.
 - **Availability Domain:** Select any Availability Domain.
 - **Image Operating System:** Click Change Image.
 - In the Browse All Images window, click Platform Images.
 - Select Oracle Linux Cloud Developer Image.
 - **Shape:** Leave the default value.
 - **Configure Networking:**
 - **Network:** Choose Select existing cloud network.
 - **VCN Compartment:** The same that has been allocated to you
 - For **VCN:** Select **OpsVCN** created earlier.
 - **Subnet:** Select the **Public Subnet** of your VNC.
 - **Select Assign a public IP address.**
 - **SSH Key:** Select the **Paste public keys** option and paste the public SSH key you had created earlier using Cloud Shell.
 - **Boot Volume:** Use the default; leave this section as is.
3. Click **Create**.
4. Once the instance state changes to **Running**, you can SSH to the instance from Cloud Shell. Take a note of the Public IP address of your Compute instance.

Wait for 5 mins before trying to ssh after the instance is running since the connection requires sometime for developer image.

5. Bring up the Cloud Shell terminal and connect to the instance by using the following command:
 - Notice the output and the change in the prompt.
 - If this is your first connection, enter **yes** when prompted as shown in the following.
 - Ensure to substitute the Public IP of your Compute VM, that is, Terraform Server.

```
$ ssh -i ~/.ssh/id_rsa opc@168.138.115.123
The authenticity of host '158.101.21.107 (158.101.21.107)' can't
be established.
ECDSA key fingerprint is
SHA256:Urvo0TAL759of2awpOQdEn+t3lAlnD3mUcy7ffCVZME.
ECDSA key fingerprint is
MD5:56:6c:6b:08:b1:bf:a4:01:f3:49:a8:5b:4a:09:49:dd.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '158.101.21.107' (ECDSA) to the list
of known hosts.
Last login: Thu Feb  4 09:45:23 2021
```

If you get this error,

```
ssh: connect to host <public_ip> port 22: Connection refused
```

Then kindly wait for some more minutes, as Developer Image will have few auto installations after build, and until these are completed, SSH connection might be refused by the instance.

This completes the task of creating and connecting to the Terraform Server.

Practice: Install Terraform and Create an Instance

Overview

In this practice, you will install Terraform, and create and destroy OCI resources. But first we will configure OCI CLI, which is required for Terraform to work as expected on our Terraform server.

Tasks

1. Download the file from the object storage using the URL copied in the previous practice.
 - a) Ensure to substitute your object storage URL in the following command before executing it.

```
[opc@opsterraform ~]$ wget https://objectstorage.us-ashburn-1.oraclecloud.com/n/ocuocictrng6/b/Terraform_Binary/o/terraform_0.15.3_linux_amd64.zip
--2021-05-13 03:13:13-- https://objectstorage.us-ashburn-1.oraclecloud.com/n/ocuocictrng6/b/Terraform_Binary/o/terraform_0.15.3_linux_amd64.zip
Resolving objectstorage.us-ashburn-1.oraclecloud.com
(objectstorage.us-ashburn-1.oraclecloud.com) ... 134.70.32.1, 134.70.28.1, 134.70.24.1
Connecting to objectstorage.us-ashburn-1.oraclecloud.com
(objectstorage.us-ashburn-1.oraclecloud.com) |134.70.32.1|:443...
connected.
HTTP request sent, awaiting response... 200 OK
Length: 32743141 (31M) [application/x-zip-compressed]
Saving to: 'terraform_0.15.3_linux_amd64.zip'

100%[=====
=====
=====
======>] 32,743,141  89.9MB/s
in 0.3s

2021-05-13 03:13:14 (89.9 MB/s) -
'terraform_0.15.3_linux_amd64.zip' saved [32743141/32743141]
```

b. Unzip the downloaded Terraform file:

```
$ unzip terraform_0.15.3_linux_amd64.zip
Archive:  terraform_0.11.13_linux_amd64.zip
  inflating: terraform

$ sudo mv terraform /usr/local/bin/
$ cd /usr/local/bin/
$ ls -l terraform
-rwxrwxr-x. 1 root root 79991413 Aug 12 18:37 terraform
```

2. Confirm if Terraform is ready for use.

```
$ cd /home/opc
$ pwd
/home/opc
$ terraform
Usage: terraform [-version] [-help] <command> [args]

The available commands for execution are listed below.
The most common, useful commands are shown first, followed by
less common or more advanced commands. If you're just getting
started with Terraform, stick with the common commands. For the
other commands, please read the help and docs before usage.

Common commands:
  apply                Builds or changes infrastructure
  console              Interactive console for Terraform interpolations
  destroy              Destroy Terraform-managed infrastructure
  env                  Workspace management
  fmt                  Rewrites config files to canonical format
  get                  Download and install modules for the
configuration
  graph                Create a visual graph of Terraform resources
  import               Import existing infrastructure into Terraform
  init                 Initialize a Terraform working directory
  output               Read an output from a state file
  plan                 Generate and show an execution plan
  providers             Prints a tree of the providers used in the
configuration
  push                 Upload this Terraform module to Atlas to run
  refresh              Update local state file against real resources
  show                 Inspect Terraform state or plan
  taint                Manually mark a resource for recreation
  untaint              Manually unmark a resource as tainted
```


validate	Validates the Terraform files
version	Prints the Terraform version
workspace	Workspace management

All other commands:

debug	Debug output management (experimental)
force-unlock	Manually unlock the terraform state
state	Advanced state management

```
[opc@opsterraform~]$ terraform -version
Terraform v0.15.3
on linux_amd64
[opc@opsterraform~]$ cd
```

3. Configure the SDK with your Oracle Cloud Infrastructure credentials. When working with OCI modules, you need to specify the basic configuration information, such as user credentials and tenancy OCID. The default configuration file name and location is `~/.oci/config`. If `.oci` is not available, create it.

```
[opc@opsterraform ~]$ mkdir ~/.oci
[opc@opsterraform ~]$ cd ~/.oci
[opc@opsterraform .oci]$ openssl genrsa -out
~/.oci/oci_api_key.pem 2048
Generating RSA private key, 2048 bit long modulus
.....+++
.....+++
e is 65537 (0x10001)
[opc@opsterraform .oci]$ cd
[opc@opsterraform ~]$ chmod go-rwx ~/.oci/oci_api_key.pem
[opc@opsterraform ~]$ openssl rsa -pubout -in
~/.oci/oci_api_key.pem -out ~/.oci/oci_api_key_public.pem
writing RSA key
[opc@opsterraform ~]$ cd .oci
[opc@opsterraform .oci]$ ll
total 8
-rw-----. 1 root root 1675 Apr 23 09:58 oci_api_key.pem
-rw-r--r--. 1 root root 451 Apr 23 09:59 oci_api_key_public.pem
[opc@opsterraform .oci]$ openssl rsa -pubout -outform DER -in
~/.oci/oci_api_key.pem | openssl md5 -c
writing RSA key
(stdin)= 5b:ff:9a:51:e3:8b:a4:50:08:af:d0:31:60:73:9c:35
```


Copy the generated key to Notepad.

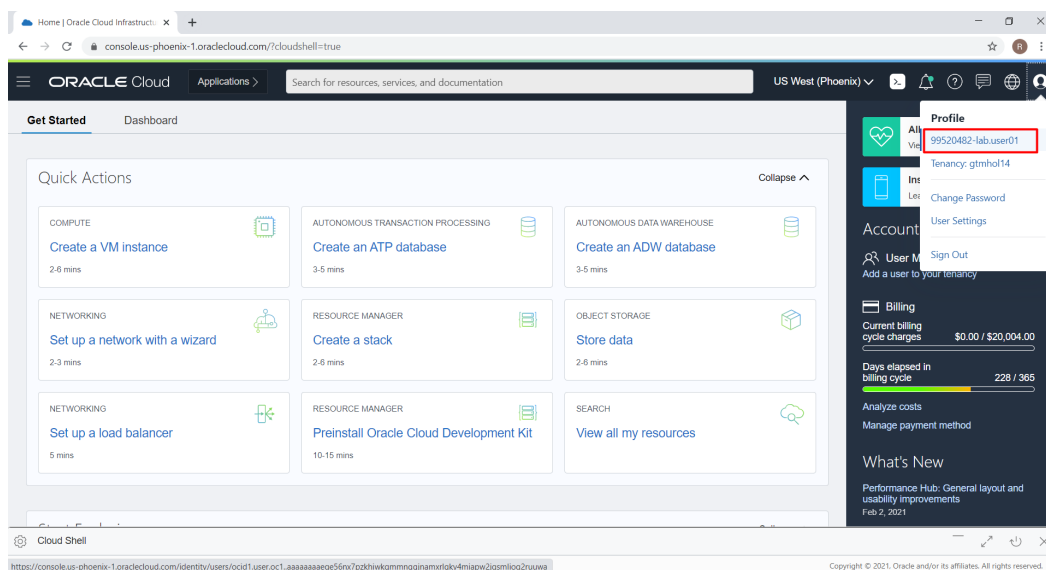
```
[opc@opsterraform .oci]$ cat oci_api_key_public.pem
```

```
-----BEGIN PUBLIC KEY-----
```

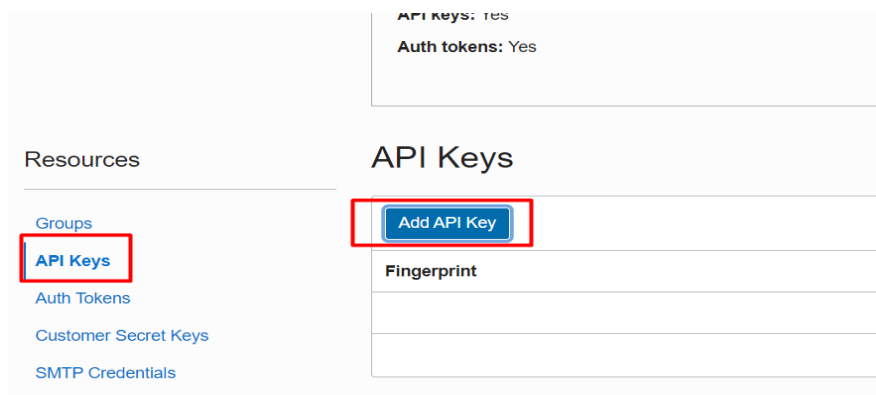
```
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAvqM69hiSf0MElk9tR4Af  
CYT+cOexiZVhvTWPEBYcA8fR01/9lU4hwsOrdp4zC/WhqFILWgkF+BJIzNsUb+eV  
KpOrsvg8h6+myKmlCdsDkTjvC7nQ+Auhxc/zcvqvSbS8MslUBgZ5+46RA4ssE/uQ  
Bpys55VaHPT2Qs2PM/y4G+Pus/At8lKpzplxY2YY+YNUFaQ/9wxLNxhbf5PphG9R  
u8X9DPxkmx7f54b7jqEX45TrWLvHVSuj4OgkOzKZE5lza+200DpUdHEGJ2tLtIsX  
D5TDjZofPQCydKld6yLn9EF2bf4llCbFi4PSrbg9x80MWuCNy0MFxXvP3HSKigUp  
GQIDAQAB
```

```
-----END PUBLIC KEY-----
```

4. Copy the contents of `oci_api_key_public.pem`.
5. Go to the Oracle Cloud Infrastructure Web console and upload the public key (`~/.oci/oci_api_key_public.pem`).
 - a. Next, click the user icon and your **username** as shown in the following:



- b. In the **Resources** section, click **API Keys**, and then click **Add API Key**:



- c. Select the **Paste Public Key** option, paste the key value copied in the previous step, and click **Add**:

Add API Key Help

Note: An API key is an RSA key pair in PEM format used for signing API requests. You can generate the key pair here and download the private key. If you already have a key pair, you can choose to upload or paste your public key file instead. [Learn more](#)

☐ Generate API Key Pair ☐ Choose Public Key File ☒ Paste Public Key

Public Key

```
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAwU7DmT4z7lpF5+x1/9Da
N0mc7DDAdMtY4dQpfEawhOzwkoWckhnuVnnZi11dQx+R66MqPNgiM8e89KM9uOCU
U02cGXKf+f8u/RHsyik79jLNJVlBcThKv4HFdeJtR/z7Phdr/UTWXKz89lcweOIW
IvwAu/22BrkDOPCI+wo3lMOBld0CqF3AEolwrYSxJArKnveYhKj1YOIRUKfzqhtD
QV3ZXwR7UfryomIM3QbOcVdp/nJHhnWRJh+AcIt9VQt6D3peXQl2wAOv4Yxsw3V1
nuuOUAWR25adRKkh/9HCTm/tXVWZQalJ9ZaFEQH354LoqjzxpG5U3tcwlcwq2Zq
JwIDAQAB
-----END PUBLIC KEY-----
```

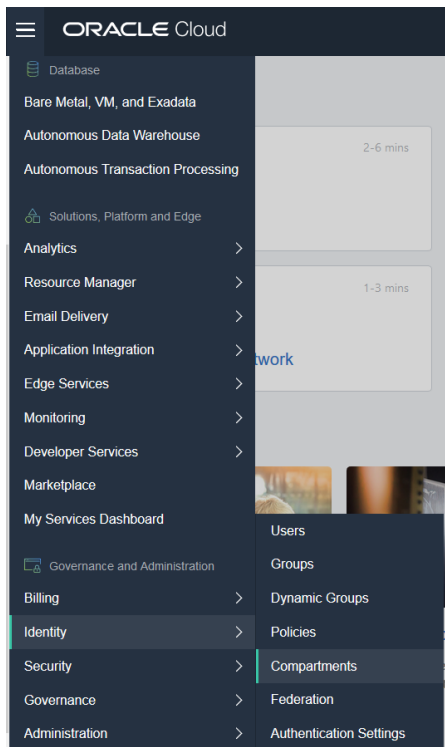
Add [Cancel](#)

- d. Once done, the fingerprint value will appear as shown in the following:

Resources	API Keys				
<div>Groups</div> <div>API Keys</div> <div>Auth Tokens</div> <div>Customer Secret Keys</div> <div>SMTP Credentials</div>	<div>Add API Key</div> <table><tr><th>Fingerprint</th><th>C</th></tr><tr><td>f0:60:f9:4c:da:b1:30:f8:86:da:df:61:5c:d4:54:7b</td><td>TI</td></tr></table>	Fingerprint	C	f0:60:f9:4c:da:b1:30:f8:86:da:df:61:5c:d4:54:7b	TI
Fingerprint	C				
f0:60:f9:4c:da:b1:30:f8:86:da:df:61:5c:d4:54:7b	TI				

6. Collect the required OCIDs for Terraform.


- a. In the web console, click Menu > Identity > Compartments.



- b. Against your compartment, click the partly displayed OCID and it will display the whole OCID with a copy option.
- c. Copy and paste OCID into Notepad to maintain all OCIDs and key information.

C16	Active	ocid1.compartment.oc1..aaaaaaaanrd36vsh44etv2	Yes	0	Fri, 26 Jan 2018 19:39:25 GMT
C17	Active	ocid1.compartment.oc1..aaaaaaaanrd36vsh44etv2	Yes	0	Fri, 26 Jan 2018 19:39:35 GMT
C18	Active	ocid1.compartment.oc1..aaaaaaaanrd36vsh44etv2	Yes	0	Fri, 26 Jan 2018 19:39:51 GMT

- d. Similarly, navigate to **Identity > Users** and copy the OCID for your user login (user_ocid).



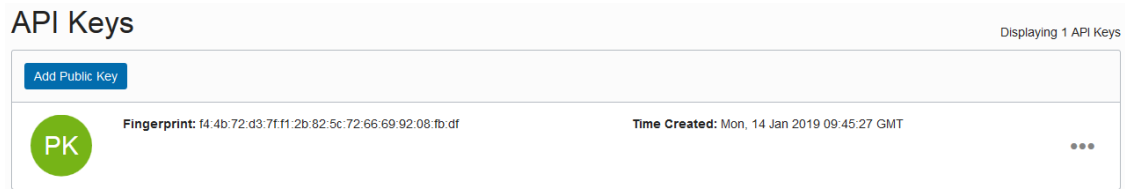
lab.user18
OCID: ..4hy6ca [Show](#) [Copy](#)
Email: -

ACTIVE

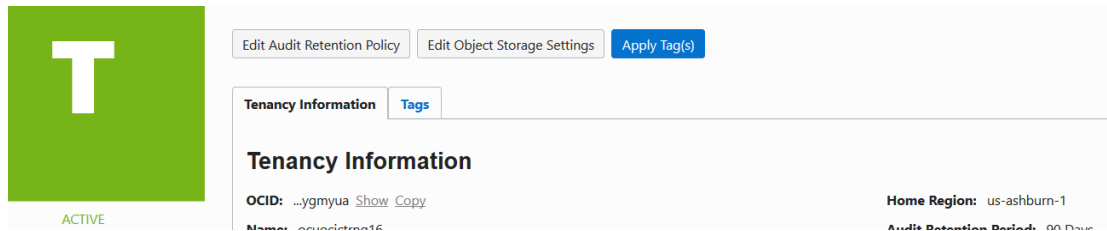
Description: New user lab.user18
Federated: No

Created: Fri, 26 Jan 2018 21:11:05 GMT

Fetch fingerprint details in the API Keys section. You had generated this in the previous practice.



- e. Click **Menu > Governance & Administration > Tenancy Details**. Here copy and record the tenancy OCID and Name (Namespace).



- f. Go to <https://docs.us-phoenix-1.oraclecloud.com/images/>

Click on the Oracle-Linux-7.9-2021.07.27-0 and scroll to your region to find the image OCID and record it.

uk-cardiff-1	ocid1.image.oc1.uk-cardiff-1.aaaaaaaaxvowgnqyr36dxxd25qhfhwpc4lhx6hk57imekifmusjxqtlpp6a
uk-london-1	ocid1.image.oc1.uk-london-1.aaaaaaa646hmq7yvlxk6wqhhdzrljfxdy7iyy6wk7xtmdf3x73ko45nwqfsa
us-ashburn-1	ocid1.image.oc1.iad.aaaaaaaab3w3vzjenuyy3idksenczspj77wz74o7unpxid6xr7zmsyi7u47q
us-gov-ashburn-1	ocid1.image.oc3.us-gov-ashburn-1.aaaaaaa3gbjraaxyq6m3ymkisab2d45mw67u6wpbobbkwsfzujxqfvveiq
us-gov-chicago-1	ocid1.image.oc3.us-gov-chicago-1.aaaaaaa3bqipd2hlwabwilyhq5v2phgftmn2gxoul44e6ttago4nsot2q
us-gov-phoenix-1	ocid1.image.oc3.us-gov-phoenix-1.aaaaaaa2ppx67w7w4gvcz5gipz7774cn4mxx2xozoss17xczwujw63pola

7. Create a Terraform directory and download the `compute.tf` file from the object storage using its URL

```
$ mkdir terraform
$ cd terraform
$ wget https://objectstorage.us-ashburn-1.oraclecloud.com/n/ocuocictrng6/b/Terraform_Binary/o/compute.tf
--2021-05-13 03:56:29-- https://objectstorage.us-ashburn-1.oraclecloud.com/n/ocuocictrng6/b/Terraform_Binary/o/compute.tf
Resolving objectstorage.us-ashburn-1.oraclecloud.com
(objectstorage.us-ashburn-1.oraclecloud.com) ... 134.70.32.1,
134.70.24.1, 134.70.28.1
```

```
Connecting to objectstorage.us-ashburn-1.oraclecloud.com
(objectstorage.us-ashburn-1.oraclecloud.com) |134.70.32.1|:443...
connected.
```

```
HTTP request sent, awaiting response... 200 OK
```

```
Length: 8572 (8.4K) [application/octet-stream]
```

```
Saving to: 'compute.tf'
```

```
100%[=====
=====
=====
=====
=====>] 8,572          --.-K/s   in 0s
```

```
2021-05-13 03:56:29 (63.4 MB/s) - 'compute.tf' saved [8572/8572]
```

8. There are multiple ways to export the variables. Either we can use the direct export command noted above or use the export commands in `.bash_profile` as shown in the following. Depending on the requirement, either all the required environment variables or only a few of them can be exported. The OCIDs and values collected from the previous steps are to be substituted in the file.

```
[opc@opsterraform ~]$ cd
```

```
[opc@opsterraform ~]$ ssh-keygen
```

```
Generating public/private rsa key pair.
```

```
Enter file in which to save the key (/home/opc/.ssh/id_rsa):
```

```
Enter passphrase (empty for no passphrase):
```

```
Enter same passphrase again:
```

```
Your identification has been saved in /home/opc/.ssh/id_rsa.
```

```
Your public key has been saved in /home/opc/.ssh/id_rsa.pub.
```

```
The key fingerprint is:
```

```
SHA256:bOagsCPkoBluRcI+B6VKsxmPHW7Pa9MBpe5fU026shw
```

```
opc@opsterraform
```

```
The key's randomart image is:
```

```
+---[RSA 3072]-----+
```

```
|  .                |
| . o   .           |
| O o o   .         |
|+ & .o .   +       |
|+B.B. o S  o .     |
|*o=ooo *   . .     |
|++o oo. oE .       |
|.. . +..o =        |
|   ..o. o          |
```

```
+-----[SHA256]-----+
```

```
[opc@opsterraform ~]$ vi .bash_profile
```

[in Vi editor, update bash profile as shown below]

```
[opc@opsterraform ~]$ cat .bash_profile
```

```
# .bash_profile
```

```
# Get the aliases and functions
```

```
if [ -f ~/.bashrc ]; then
```

```
    source ~/.bashrc
```

```
fi
```

```
# User specific environment and startup programs
```

```
PATH=$PATH:$HOME/bin
```

```
export PATH
```

```
####Required information #####
```

```
####Tenant Information
```

```
export
```

```
TF_VAR_tenancy_ocid="ocid1.tenancy.oc1..aaaaaaaamd6f5rb4cmrnq4pjjr3o3wbnoouq36hvwnog6qxvuadtyo2kanea"
```

```
export
```

```
TF_VAR_user_ocid="ocid1.user.oc1..aaaaaaaapy2yglbpke63ol6igpny27kosnear6paqufqxnm4nipnc6csh7gq"
```

```
export
```

```
TF_VAR_fingerprint="2b:2d:6d:a2:39:7c:7c:2f:9c:b7:5e:bc:c6:e4:83:3e"
```

```
export TF_VAR_private_key_path="/home/opc/.oci/oci_api_key.pem"
```

```
export
```

```
TF_VAR_compartment_ocid="ocid1.tenancy.oc1..aaaaaaaamd6f5rb4cmrnq4pjjr3o3wbnoouq36hvwnog6qxvuadtyo2kanea"
```

```
export TF_VAR_region="us-ashburn-1"
```

```
export TF_VAR_namespace="myocitenancyname"
```

```
export TF_VAR_ssh_public_key=$(cat /home/opc/.ssh/id_rsa.pub)
```

```
### availability domain (1, 2 or 3)
```

```
export TF_VAR_ADs="1"
```

```
export TF_VAR_avail_dom=AMVH:US-ASHBURN-AD-3
```

```
### Instance credentials
```

```
export
```

```
TF_VAR_instance_image_ocid=ocid1.image.oc1.iad.aaaaaaaab3w3vzjenuyy3idksenczspj77wz74o7unpxid6xr7zmsyi7u47q
```

```
[opc@opsterraform ~]$ source ~/.bash_profile
```

9. Run the `compute.tf` script. This script creates a VCN and a compute instance `webserv1`.

Note: Substitute your region in the script wherever applicable.

```
$ cd terraform
$ ls
compute.tf
$ terraform init
Initializing the backend...

Initializing provider plugins...
- Reusing previous version of hashicorp/oci from the dependency
lock file
- Using previously-installed hashicorp/oci v4.26.0

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.
$ terraform plan -out file
All resources to be created will be shown

$ terraform apply
Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value:yes
Apply complete! Resources: 10 added, 0 changed, 0 destroyed.
```




Select **Compute** from the Main menu and verify that the instance **webserv1** is created.

This completes the task of installing Terraform and creating a compute instance.

10. Click the Webserv1 instance and click **Terminate** to drop the instance.