ORACLE
University

Oracle Linux: System Administration on Oracle Cloud Infrastructure

Activity Guide

S1111828GC10

Learn more from Oracle University at ***education.oracle.com***

Disclaimer

This document contains proprietary information and is protected by copyright and other intellectual property laws. The document may not be modified or altered in any way. Except where your use constitutes "fair use" under copyright law, you may not use, share, download, upload, copy, print, display, perform, reproduce, publish, license, post, transmit, or distribute this document in whole or in part without the express authorization of Oracle.

The information contained in this document is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

Restricted Rights Notice

If this documentation is delivered to the United States Government or anyone using the documentation on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software" or "commercial computer software documentation" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

Trademark Notice

Oracle®, Java, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

Third-Party Content, Products, and Services Disclaimer

This documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Table of Contents

Practices for Lesson 1: Course Introduction

Practices for Lesson 1

Overview

There are no practices for Lesson 1.

Practices for Lesson 2: Foundations of OCI

Practices for Lesson 2

Overview

In these practices, you will connect to your Oracle Cloud Infrastructure (OCI) web portal and generate a new public and private key pair to be used with your lab compute instances.

Practice 2-1: Connecting to the Oracle Cloud Infrastructure Web Portal

Overview

In this practice, you will use your browser and connect to the OCI web portal.

Assumptions

- Instructions are written using the Microsoft Windows 11 operating system.
- No specific Microsoft Windows 11 (or 10) dependencies are relied upon in this course's instructions.
- Translate instructions to the respective equivalent when using another local operating system.
- You will need to use a local text editor, a local web browser, and a command-line prompt window.
- Instructions are written against CMD command-line for the Microsoft Windows environment.

Tasks

1. Obtain and record your OCI user account details.

   a. You will be provided your own user login and the initial password to use when connecting to the OCI web portal. You may be required to reset the password upon first login. For the practices, we will make use of the example credentials as listed:

Example Credentials	Example Credential Value
Username	**lab.user16**
Password	Initial password to be changed on first login

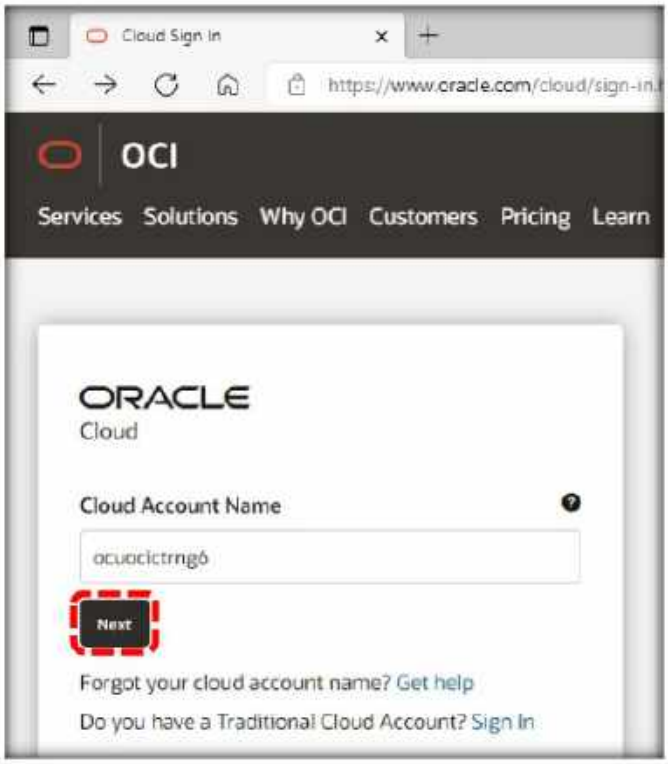
   b. Create a directory with a short name without spaces and a text file in a convenient location to record various kinds of text from the clipboard to simplify copying and pasting information during the course activities. The local directory used in this course's activities will be `D:\OL9labs`.

2. Create a text file to keep clipboard values handy. Use any text editor you are familiar with.
   Example: `D:\OL9labs>notepad lab-work.txt`

   Note: While working on the practices, keep the text file editor open. Save the content periodically and every time you add or change content.

3. Keep the editor open and add OCI username and password entries to copy/paste from when signing in to the OCI portal. Example:

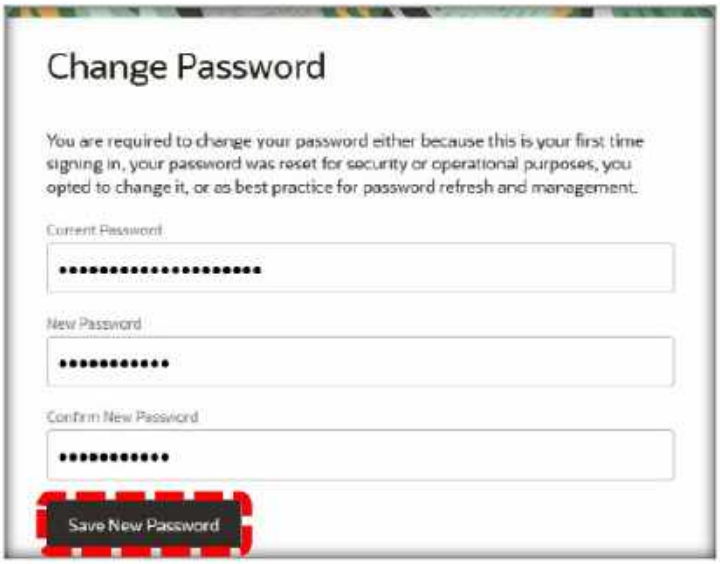
4. Sign in to the OCI portal. Enter the Cloud Account Name and click **Next**.



5. On the next screen, click the down arrow to expand the Direct Sign-In dialog box.

6. Enter your username and password. Then click the **Sign In** button.
   a. Enter the assigned username and the initial password. Click **Sign In**.

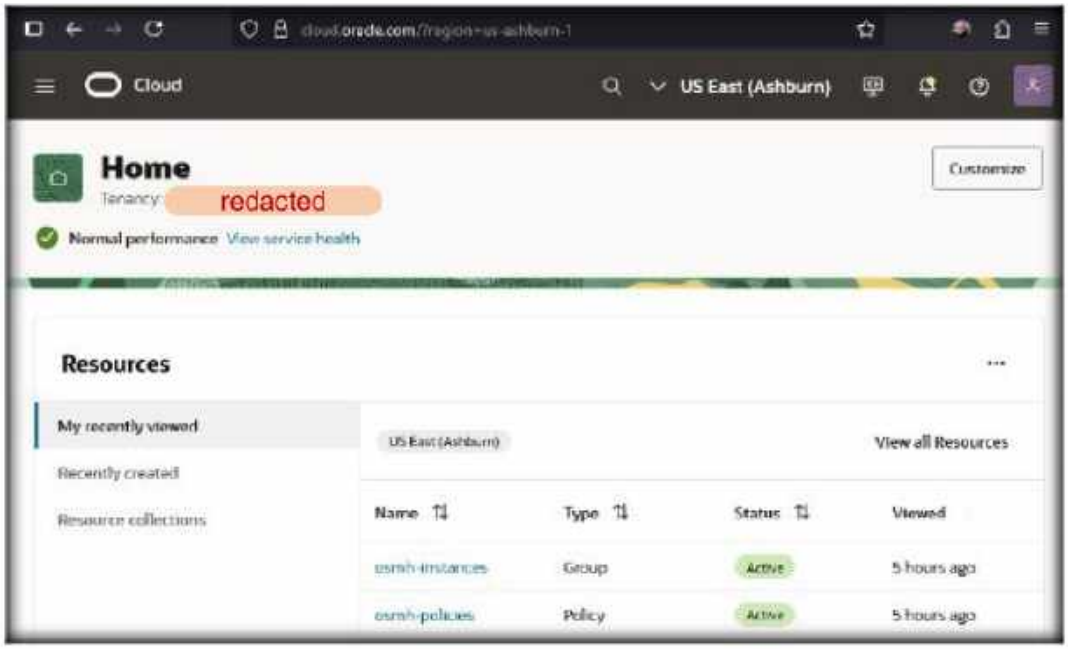
b. Reset password panel appears. Provide the initial and a new password with confirmation as instructed. Click **Save New Password** to continue.



c. Record the new password in your work document for future sign-ins while working on this course's practices.

7. The OCI portal webpage loads, and the **Resources** section is displayed by default.

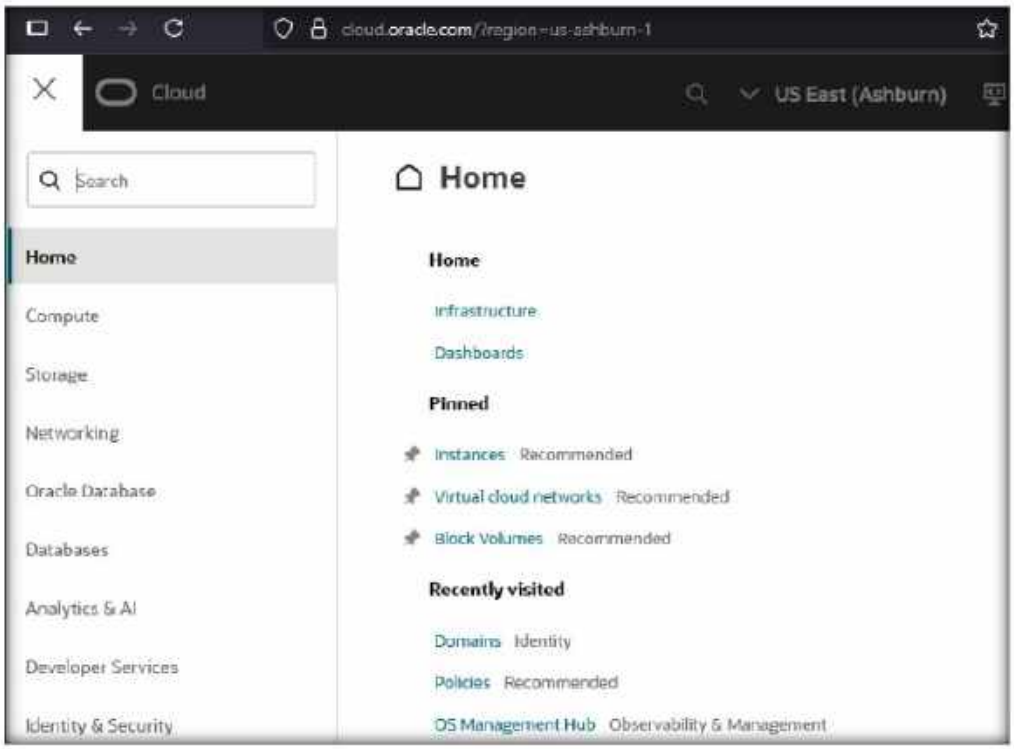
Note: Webpage may look different in the OCI Portal version that you will be using. Links and sections would still be the same.



8. Make a note of the top toolbar of the OCI portal.
   - The top-left corner has the main portal menu (often referred to as the "hamburger" menu button): 
   - On the center-right side, you see the OCI region detail, which shows the **home** region by default:

   - Next to the region is the **Developer tools** (Cloud Shell) launcher button: 
   - Further to the right you see the **Notifications** button: 
   - The **Help** button follows: 
   - The right-most button allows access to the **User profile**: 

9. Click the main portal (hamburger) menu  at the top-left corner and examine the menu choices.



10. Experiment with various menu items.

Note: Not all possible choices are shown in the screen capture.

Solution 2-1: Connecting to the Oracle Cloud Infrastructure Web Portal

Overview

There is no solution for this practice. The practice has to be completed.

Practice 2-2: Generating a Public and Private Key Pair for OCI Instance Access

Overview

In this practice, you will generate a public and private key pair for SSH communications with the OCI compute instances that you will be creating and managing throughout the course.

Assumptions

You have completed Practice 2-1 and created a working directory on your local PC for the lab activities.

Tasks

1. Open or reuse the command-prompt window and change the directory to your lab practices location. For example (in the Windows CMD environment), carry out the following commands:

   a. Type `D:` and press the **Enter** key.

   ```
   D:
   ```

   b. Type `cd \OL9labs` and press the **Enter** key.

   ```
   cd \OL9Labs
   ```

   c. Type `dir` and press the **Enter** key.

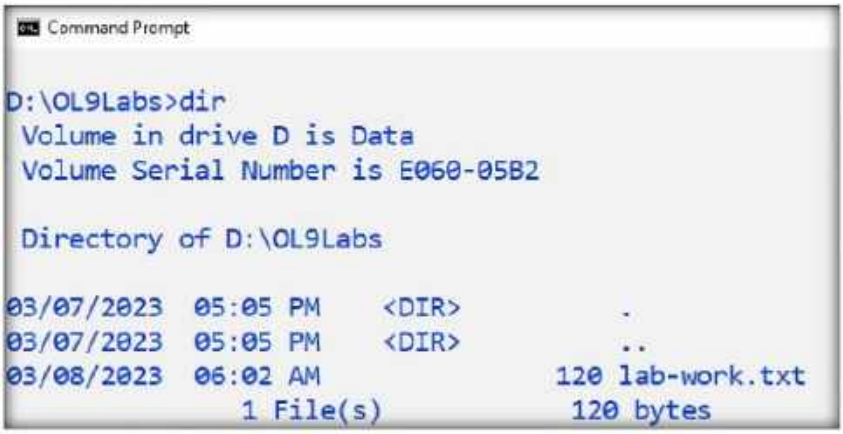
   ```
   dir
   ```

   ```
   C:\temp>D:

   D:\>cd \OL9Labs

   D:\OL9Labs>
   ```

d. The output should resemble the following:



2. Generate a public and private key pair using the **ssh-keygen** command. (Syntax would be similar for other operating systems.)

```
D:\OL9Labs> ssh-keygen -t rsa -b 2048 -f OL9lab_key
```

3. A passphrase is not required for our purposes. Press **Enter** to keep the passphrase empty when prompted.
4. The entire activity should resemble the following:

5. Type **dir** and press the **Enter** key.

```
D:\OL9Labs> dir
```

6. The output should look like this:

```
D:\OL9Labs>dir
 Volume in drive D is Data
 Volume Serial Number is E060-05B2

 Directory of D:\OL9Labs

03/08/2023  09:29 AM    <DIR>          .
03/08/2023  09:29 AM    <DIR>          ..
03/08/2023  06:02 AM               120 lab-work.txt
03/08/2023  09:29 AM             1,831 OL9lab_key
03/08/2023  09:29 AM               403 OL9lab_key.pub
               3 File(s)          2,354 bytes
               2 Dir(s)  137,563,848,704 bytes free
```

7. File **OL9lab_key** is the private key and **OL9lab_key.pub** is the public key generated by the **ssh-keygen** utility.

Solution 2-2: Generating a Public and Private Key Pair for OCI Instance Access

Overview

There is no solution for this practice. Please complete all the steps.

Practice 2-3: Creating a Virtual Cloud Network

Overview

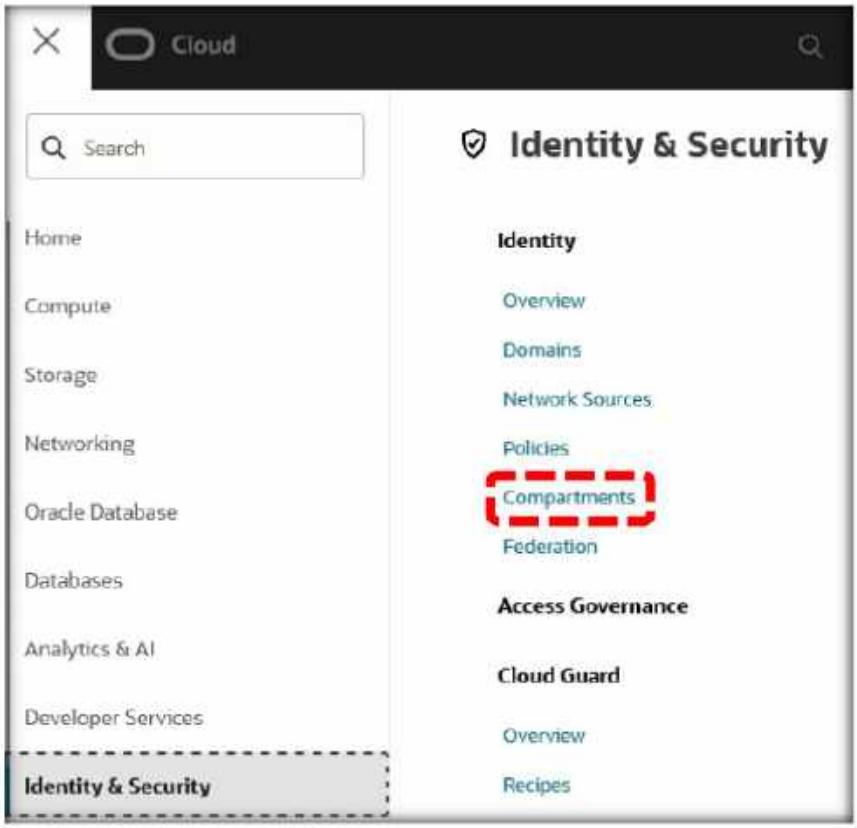
In this practice, you will create a new Virtual Cloud Network (VCN) to use with your practices.

Assumptions

You have completed Practice 2-1 successfully.

Tasks

1. Log in to the OCI portal.
2. Navigate to the main portal menu → **Identity & Security** → **Compartments**.
   Note: All practice work will be completed in one single OCI compartment. A specific compartment would be assigned to you.



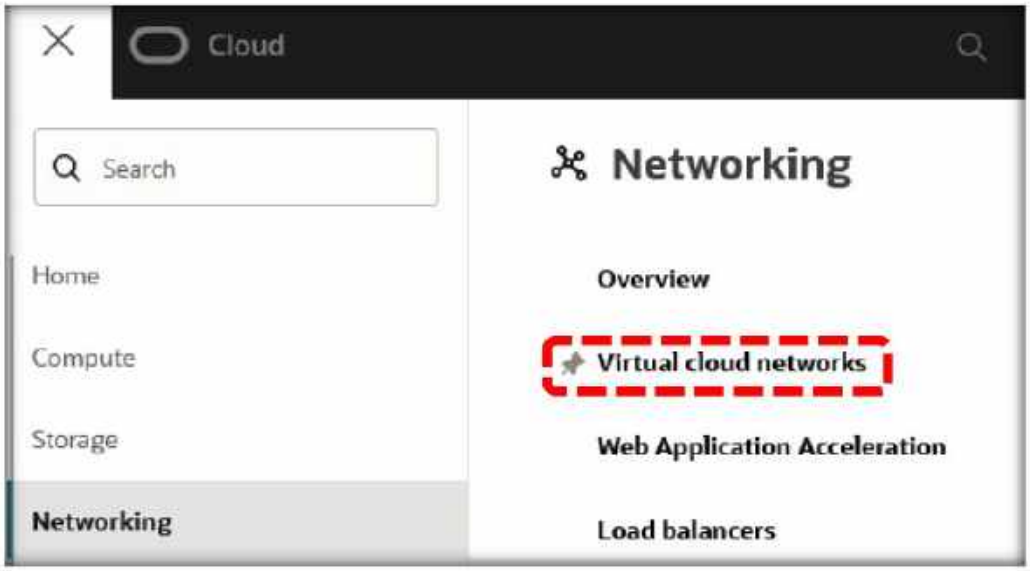
3. Locate your compartment in the list, then copy and save your compartment's OCID.

   a. Hover your cursor above the hyperlink of the compartment's OCID column.

   b. Click **Copy** when the pop-up window displays the full OCID.

   c. Paste the copied value into your text editor (it will be needed later). It is always possible to look up the OCID. However, having it instantly visible in a text file will be handy.

4. Navigate to main portal menu → **Networking** → **Virtual cloud networks**.



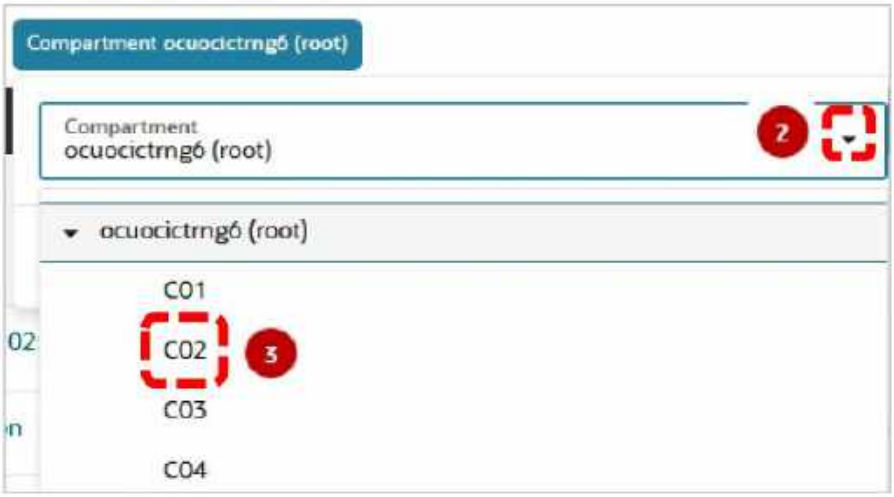
5. Create a new VCN.

   a. Under **List Scope**, select **<your compartment>** in the **Compartment** input.

      You will need to expand the root compartment to locate the specific compartment allocated to you.

      Newer OCI Console layout shows the currently selected compartmenrt view:

      Click on the **Compartment** field (1), expand the list (2), then click your assigned compartment in the list (3). Example shows C02 compartment.



Click the **Apply filter** button to select the chosen compartment.

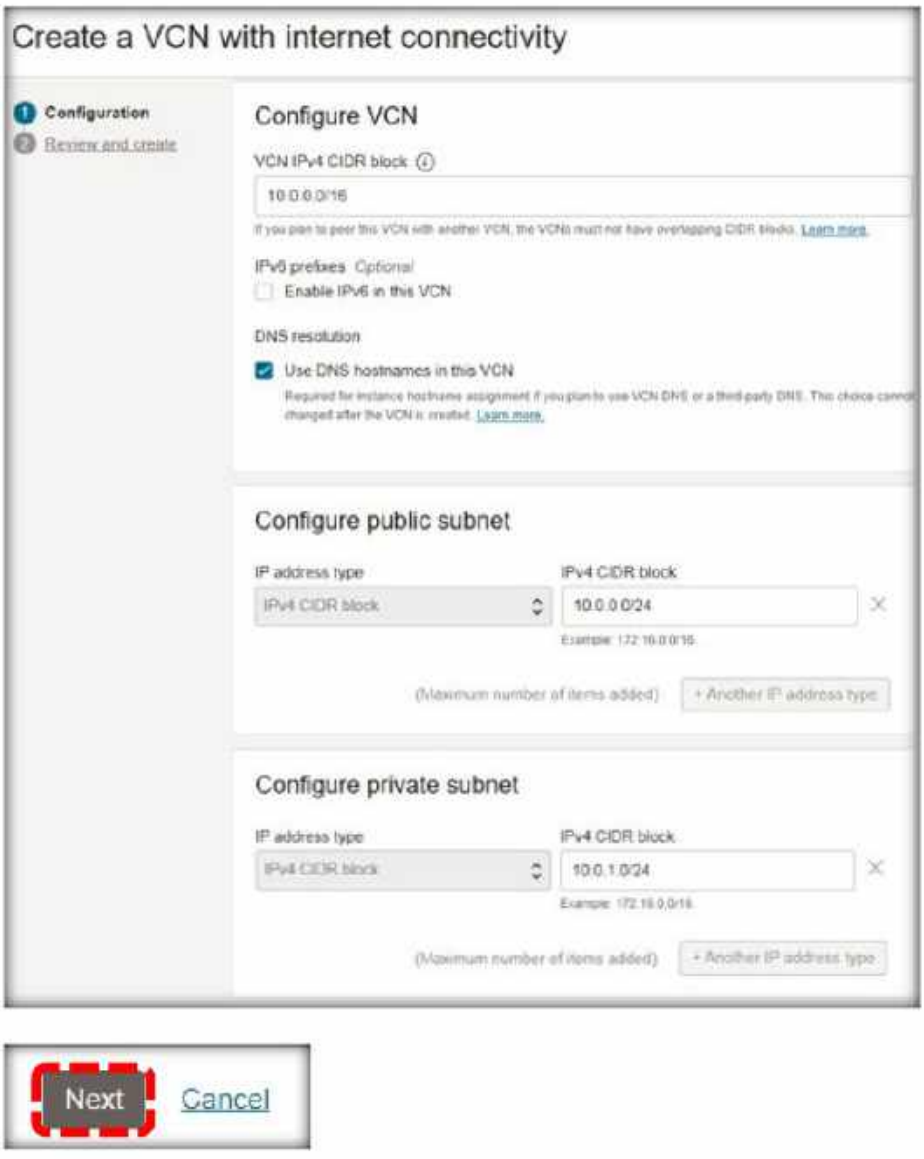
b. Click **Start VCN Wizard**. 

Newer OCI Console – equivalent screen segment is shown below.

c. Using the default selection, proceed by clicking the **Start VCN Wizard** button.

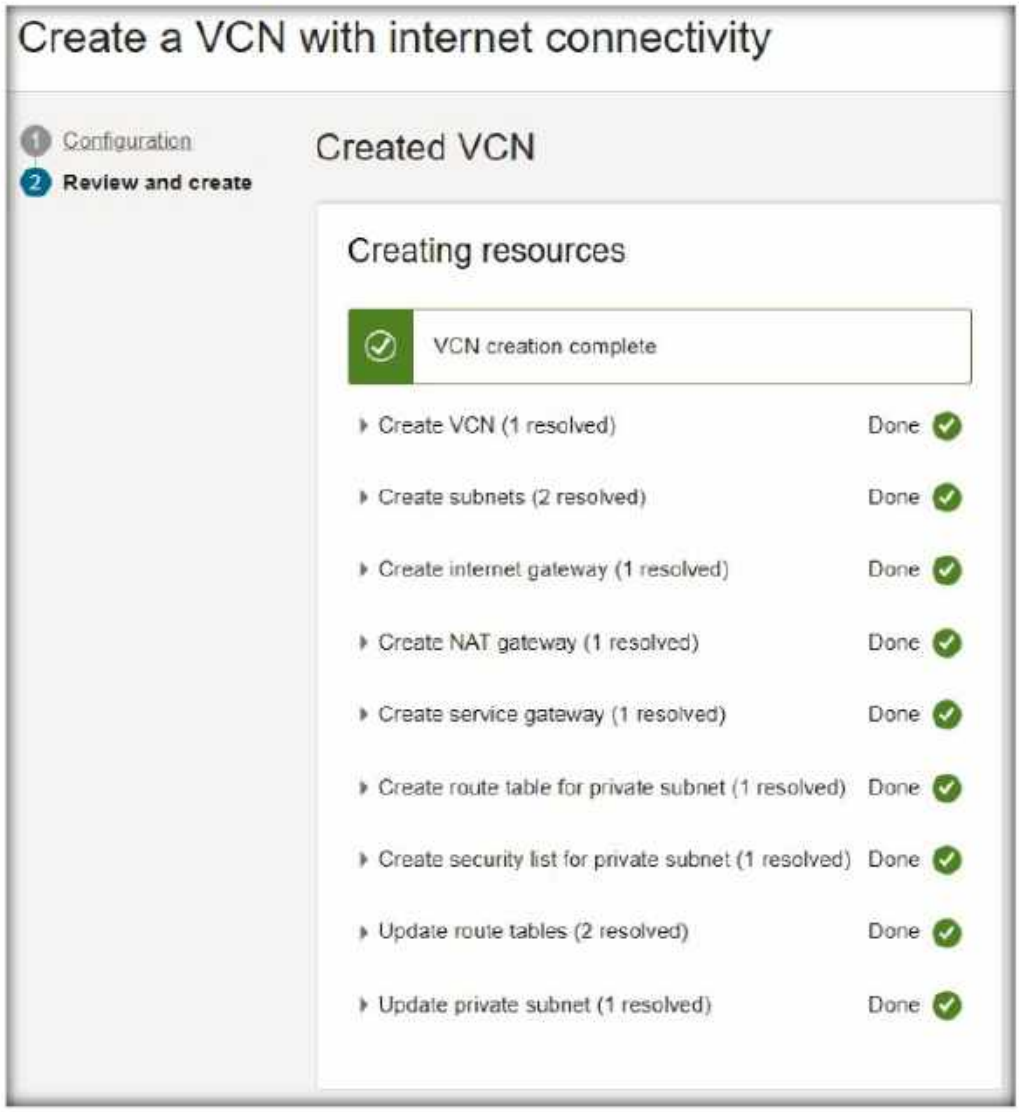
Newer OCI Console with a different view is shown below.

d. Specify **labVCN** as the VCN Name, ensure it is defined in the **<labCompartment>**, accept other choices, and click the **Next** button (bottom-left corner).



e. Review the properties and click **Create**.

f. A public VCN and its Internet Gateway will be created for your compartment.



Solution 2-3: Creating a Virtual Cloud Network

Overview

There is no solution for this practice. The practice has to be completed.

Practice 2-4: Creating a Compute Instance

Overview

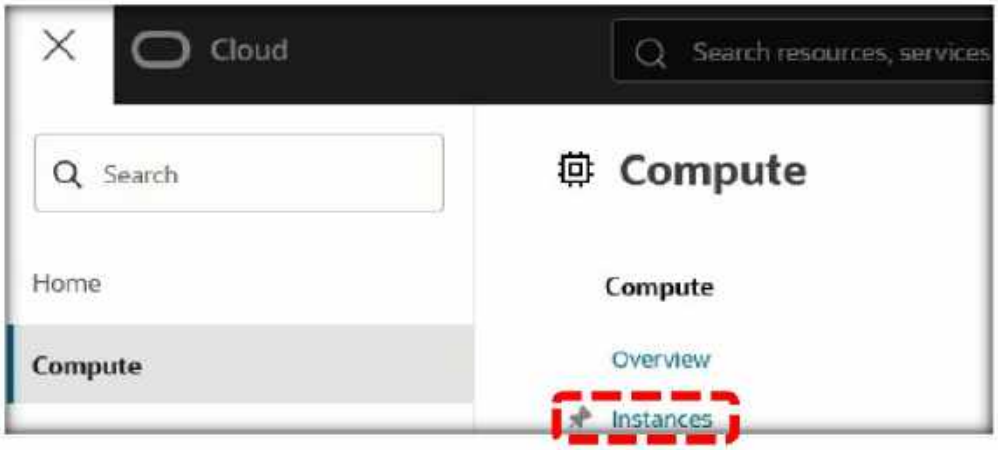
In this practice, you will create a new Oracle Linux 9 compute instance.

Assumptions

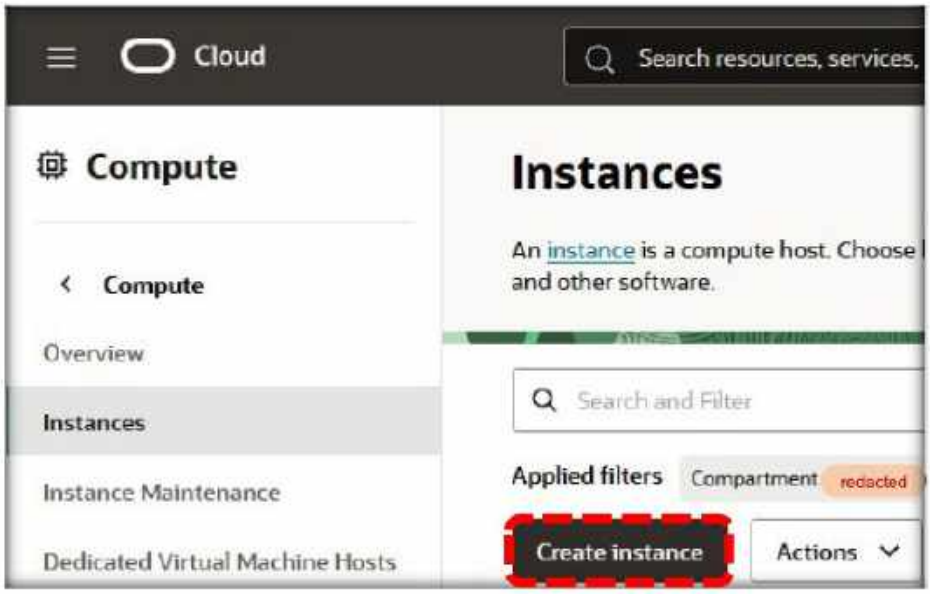
You have completed practices 2-1, 2-2, and 2-3.

Tasks

1. Log in to the OCI portal.
2. Navigate to main portal menu → **Compute** → **Instances**.



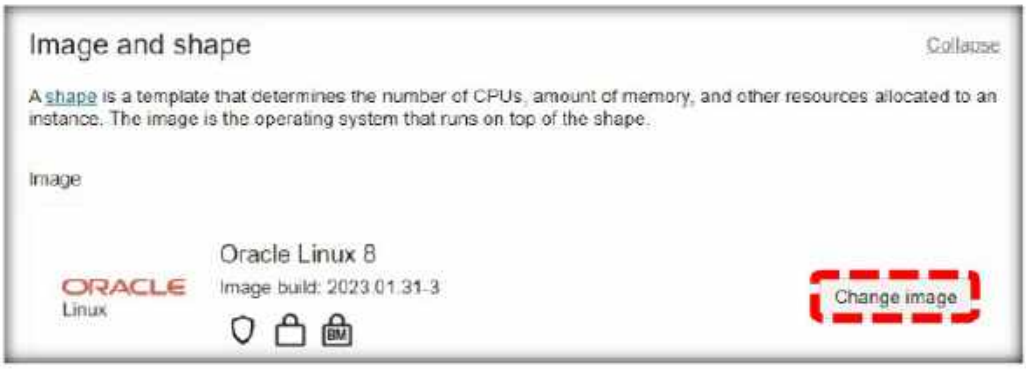
3. Click **Create instance**.



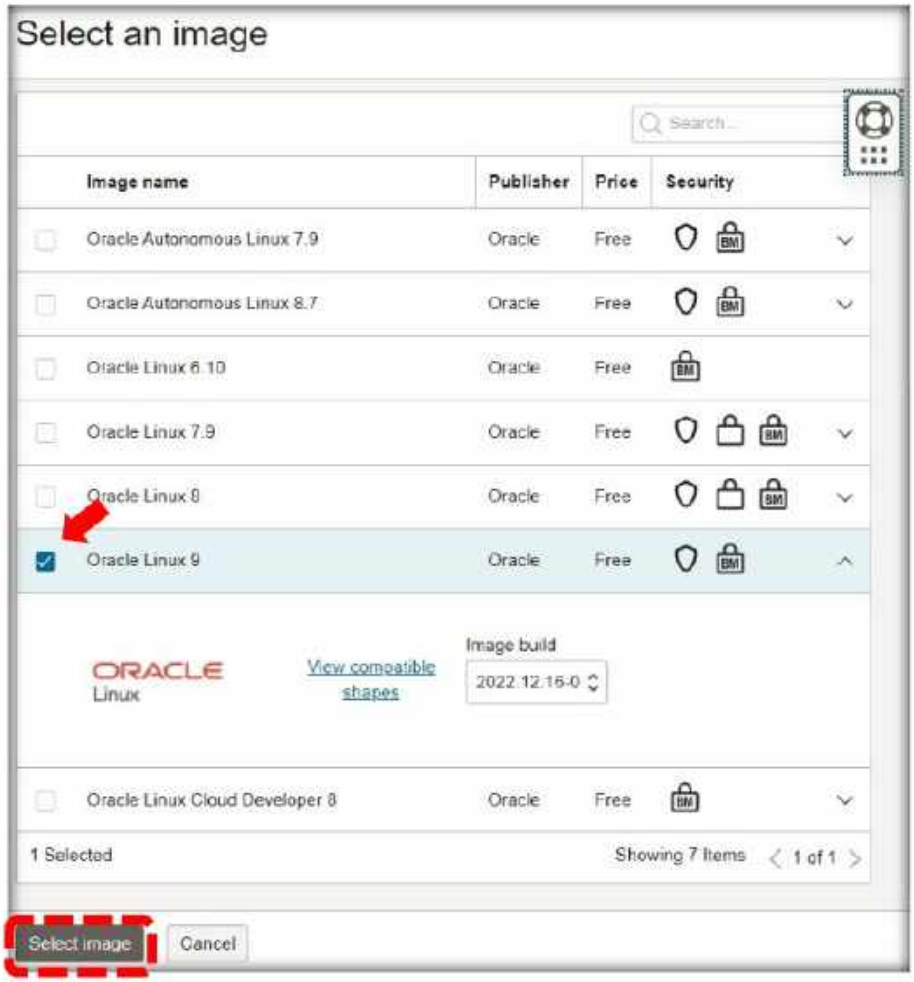
Note: Verify, that your assigned compartment is being used.

a. Name the new instance **lab-node1** and ensure **Create in compartment** is set to **<your Compartment>**. The availability domain is not important at this time, but make note of it.

b. Select the compute image. If the image you need does not appear, click **Change Image** button and select the desired image. Skip the next step if the desired image is already shown.

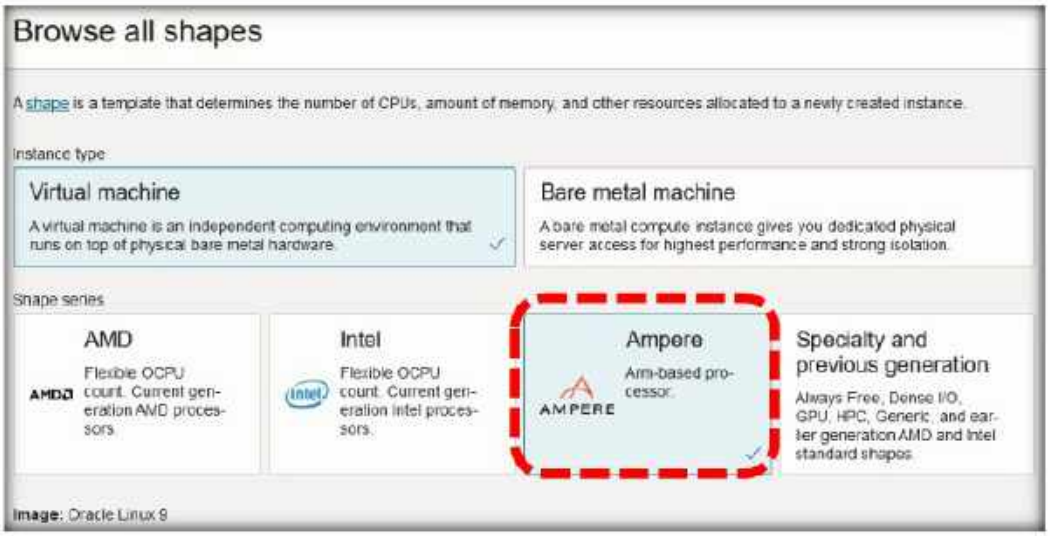


c. Select the check box next to **Oracle Linux 9**, verify the latest Image Build date, and click the **Select image** button below.



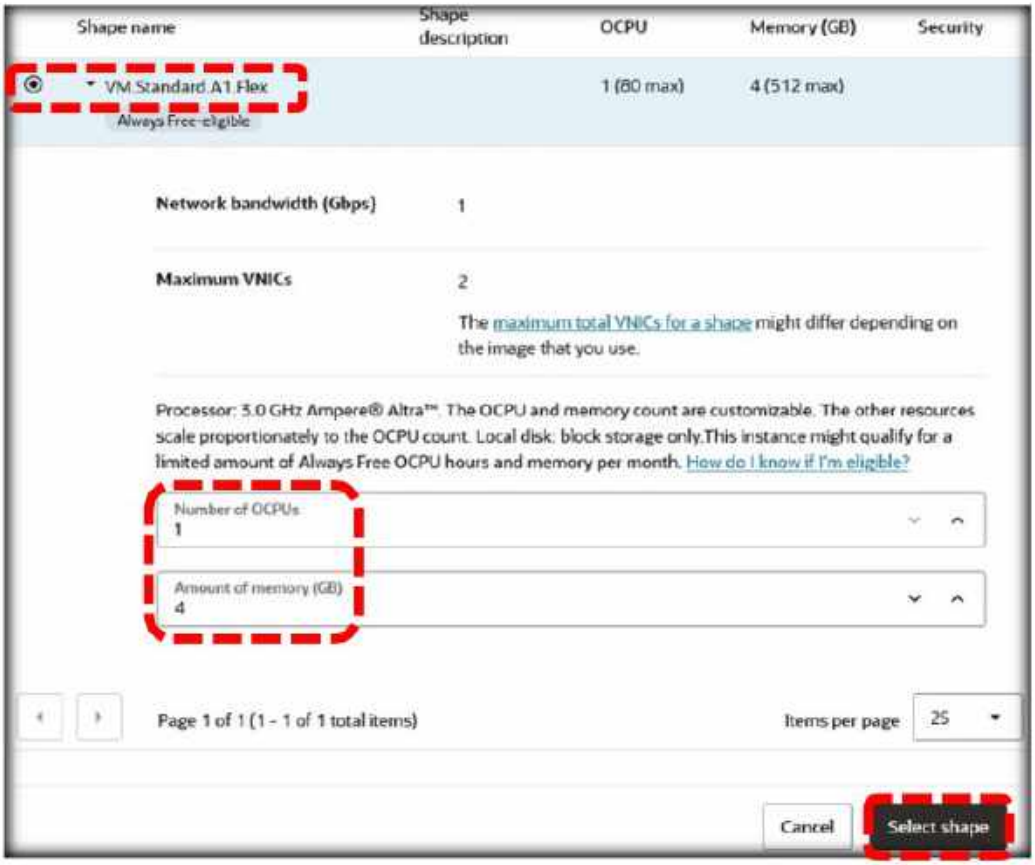
d. Pick the compute shape to assign to the new instance. Please be mindful that not all shapes may be permitted by the tenancy or compartment resource quotas. To select a different shape, click the **Change shape** button.

e. Select the hardware type to locate the desired shape. In this course, we use the **(Ampere) VM.Standard.A1.Flex** shape. Click the **Ampere** tile to see compatible choices.



f. Click the small arrow next to the **VM.Standard.A1.Flex** shape to adjust VM resources.

Keep **1 OCPU**, reduce memory from default to **4 GB**, then click the **Select shape** button.



In the newer OCI Console, please click the **Next** button (twice) to continue.

g. Review the **Primary VNIC information** section.
   1) Under **Primary network**, select **Select existing virtual cloud network**.
   2) Ensure **VCN in <*your compartment*>** has the VCN selected.
   3) Under **Subnet**, select **Select existing subnet**.
   4) Ensure **Subnet in <*your compartment*>** has the public regional subnet selected.

h. Review the **Primary VNIC IP address** section.
   1) Under **Private IPv4 address**, select **Automatically assign private IPv4 address**.
   2) Under **Public IPv4 address**, select **Automatically assign public IPv4 address**.

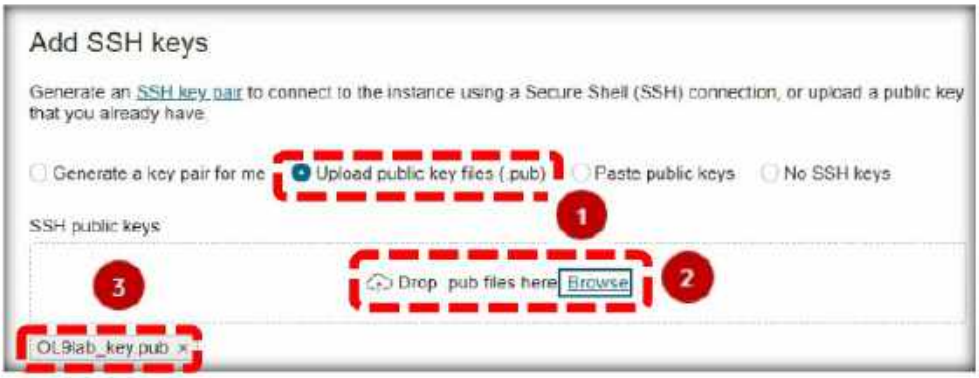
You will be using a public IP address for the instances.

i. Review the **Add SSH keys** section.

1) Select **Upload public key files (.pub)**.

2) Click the **Browse** hyperlink, navigate to your key file location, and use the key you generated earlier.

After the key is uploaded, your screen should resemble the following image:



Newer OCI Console: click the **Next** button, skip to the next task.

j. Keep the default settings in the **Boot volume** and **Block volumes** sections. Click **Next**.

4. Click the **Create** button and wait for the creation process to complete.

The Running instance is displayed with the green tile.

5. In the **Work requests** section, notice the operation completion state.

6. Review the **Resources** section. Click **Boot volume** (created for you automatically). The default size would typically be **47 GB**. Note the actions menu on the right: ⋮

7. There are no block volumes currently attached to the instance. You will work with block volumes later.

Solution 2-4: Creating a Compute Instance

Overview

There is no solution for this practice. Please complete all the steps.

Practices for Lesson 3: Introduction to Oracle Linux

Practices for Lesson 3

Overview

In these practices, you will learn to:

- Make use of the Oracle Cloud Shell
- Edit text files and scripts with the VIM editor
- Create SSH connections and work with an Oracle Linux instance in OCI
- Manage user and group information with Oracle Linux command-line tools

Practice 3-1: Experimenting with the OCI Cloud Shell

Overview

In this practice, you will start an Oracle Cloud Shell session and use it to run some Linux utilities.

Tasks

1. Log in to the OCI portal.
2. Start the Oracle Cloud Shell session.

    a. Click the **Developer tools** button on the top toolbar of the OCI portal  and then select **Cloud Shell** from the drop down.

    b. It may take a few moments for the new session to launch.

    The Cloud Shell connection opens at the bottom of the screen.

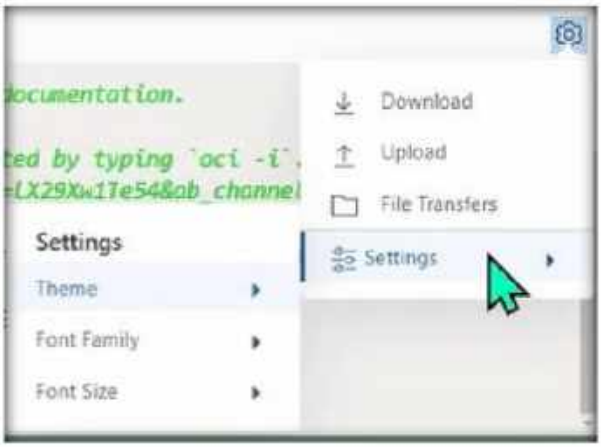
    Note: You may be offered to run a Cloud Shell tutorial – this is not required to complete the practices.

c. Note the **Minimize**, **Maximize**, and **Close** buttons on the right of the Cloud Shell toolbar.

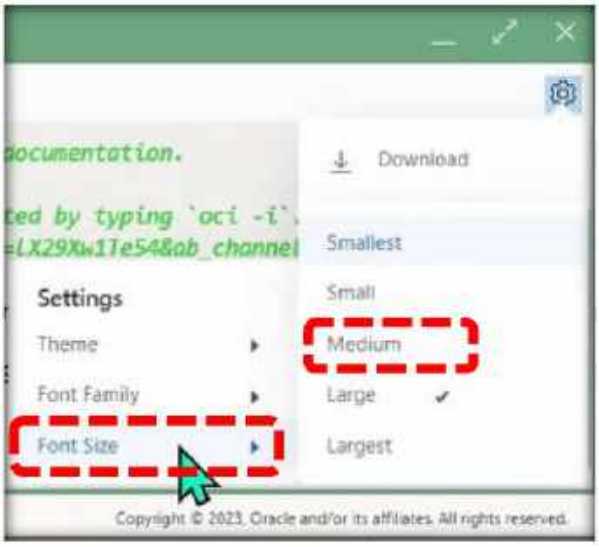
3. Customize Cloud Shell settings (optional task).

a. Click the **Gear** icon  on the right of the Cloud Shell toolbar.

b. Hover your cursor over **Settings.**



c. Then hover over **Font Size** and click **Medium.**



d. The font size will be enlarged.

e. Switch the Cloud Shell window background between **Dark** and **Light** using the **Theme** setting.

f. Use Maximize  and Minimize  controls when convenient.

Note: Oracle Cloud Shell is a small Linux VM with 5 GB of disk space. You can use this disk space for file storage and other limited-scope operations.

4. Run a command to find out the OS version.

   a. At the Cloud Shell prompt ending with "**$** ", type **`uname -opr`** and press the **Enter** key.

   ```
   uname -opr
   ```

   The output will contain the OS version and platform properties.

   ```
   x_99285267@cloudshell:~ (us-ashburn-1)$ uname -opr
   4.14.35-2047.521.4.el7uek.x86_64 x86_64 GNU/Linux
   ```

5. Find out the current location in the Cloud Shell VM file system.

   a. At the command prompt, type **`pwd`** and press the **Enter** key.

   ```
   pwd
   ```

   ```
   x_99285267@cloudshell:~ (us-ashburn-1)$ pwd
   /home/x_99285267
   ```

   Note: **`pwd`** stands for print working directory.

6. Create directories in the Cloud Shell VM.

   a. At the command prompt, type **`mkdir scripts`** and press the **Enter** key.

   ```
   mkdir scripts
   ```

   b. Create another directory, **`mkdir keys`**, and press the **Enter** key.

   ```
   mkdir keys
   ```

   Note: **`mkdir`** stands for "make directory."

7. List the contents of the current location:

   a. At the command prompt, type **`ls`** and press the **Enter** key.

   ```
   ls
   ```

   Note: `ls` stands for list.

b. The output should now contain the directories you have created.

```
x_99285267@cloudshell:~ (us-ashburn-1)$ ls
keys  scripts
```

Note: There might be other entries and directories, too.

c. To view more details about entries, type `ls -l` (dash and lower L) and press the **Enter** key.

```
ls -l
```

```
x_99285267@cloudshell:~ (us-ashburn-1)$ ls -l
total 0
drwxr-xr-x. 2 x_99285267 oci 6 Mar  8 17:22 keys
drwxr-xr-x. 2 x_99285267 oci 6 Mar  8 17:22 scripts
```

Note: You see the entry type (left-most character), followed by the entry access permissions, username and group of the owner, and the entry modification date preceding the entry name. Hidden entries (names beginning with the period character) will not appear.

d. To display all entries, including the hidden ones, type `ls -a` and press the **Enter** key.

```
ls -a
```

```
x_99285267@cloudshell:~ (us-ashburn-1)$ ls -a
.   .bash_history  .bash_profile  .emacs  .npm
..  .bash_logout   .bashrc        keys    scripts
```

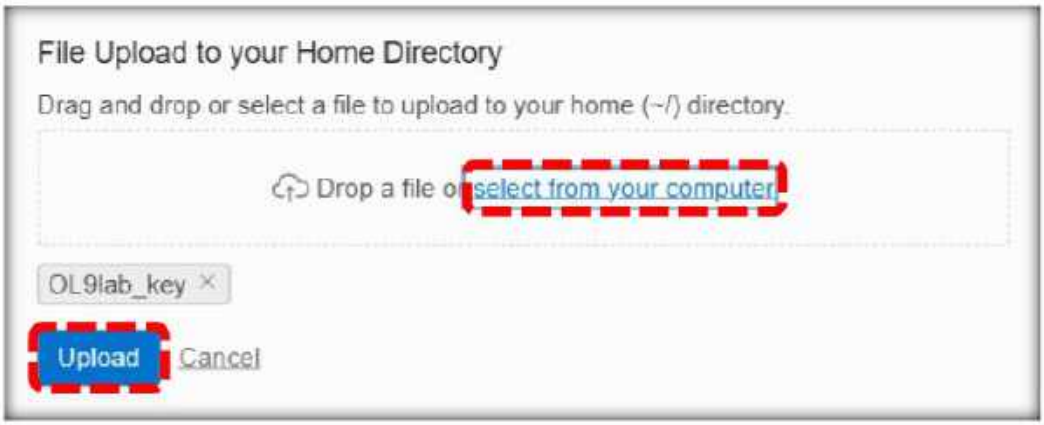
Note: There might be other entries too. Hidden entries are often made use of by various utilities and tools in Linux or UNIX. The single period character entry represents the current directory. Double period entry is the parent directory of the current one. Entry names beginning with **.bash** are used by the Linux command-line interpreter. It is permissible to combine multiple flags together like `ls -al` to see all entries and more details. Multiple other flags can be used with the `ls` command.

Background details: The first ever UNIX command-line interpreter was just **sh** – short for shell or **Bourne shell**, referring to its original author. Linux uses the newer and more feature-rich command-line interpreter **bash**, which stands for **b**orn **a**gain **sh**ell. The bash shell incorporates all the features and capabilities of the Bourne shell, as well as another shell – **C-shell** or **csh**; the latter gained popularity in some UNIX flavors. Besides **bash**, **csh**, and **sh**, some prefer a different command-line interpreter – **Korn shell** or **ksh**, which also supports the syntax of the original **sh**, but adds a different set of command-line syntax. If desired, the Korn shell can be installed and used in your Linux OS.

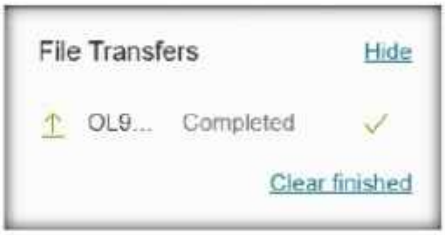
8. Upload your SSH keys to the Cloud Shell. Once uploaded, you can connect to your running instances by using SSH authentication from within the Cloud Shell VM environment. This is not a requirement; however, it may be very handy later on. You can always do it when you need it later, too.

   a. Click the **Gear** icon  and select the **Upload** option.

   b. Click **select from your computer** and then select the private key you generated in the previous practices. For example, select the **`OL9lab_key`** file and click the **Upload** button.



c. The Confirmation dialog box may look like this:



d. Click the **Hide** link.

9. List the contents of the directory after upload.

a. At the command prompt, type **`ls -l`** (dash and lower L) and press the **Enter** key.

```
ls -l
```

```
x_99285267@cloudshell:~ (us-ashburn-1)$ ls -l
total 4
drwxr-xr-x. 2 x_99285267 oci    6 Mar  8 17:22 keys
-rw-r--r--. 1 x_99285267 oci 1831 Mar  8 17:34 OL9lab_key
drwxr-xr-x. 2 x_99285267 oci    6 Mar  8 17:22 scripts
```

b. The uploaded file appears in the current directory. Move the private key into the **keys** subdirectory. At the command prompt, type **`mv OL9lab_key keys`** and press the **Enter** key.

```
mv OL9lab_key keys
```

Note: Command **`mv`** moves file system objects. This command takes at least two parameters. If you want to move multiple objects, the last parameter has to be a directory or it will result in an error message. The move command can be used to rename objects (only two parameters may be used in that case). If the last parameter is a directory, all objects before the last one would be moved into that directory.

c. List the contents of the **keys** directory. At the command prompt, type **ls keys** and press the **Enter** key. Optionally, use the dash-l option to see more details.

```
ls keys    ### ls -l keys
```

```
x_99285267@cloudshell:~ (us-ashburn-1)$ mv OL9lab_key keys
x_99285267@cloudshell:~ (us-ashburn-1)$ ls -l keys
total 4
-rw-r--r--. 1 x_99285267 oci 1831 Mar  8 17:34 OL9lab_key
```

Note: If you are familiar with Linux or UNIX shell, you may skip the remainder of this practice.

10. Find out the current user identity. At the command prompt, type `whoami` and press the **Enter** key. The username will be displayed.

```
whoami
```

```
x_99285267@cloudshell:~ (us-ashburn-1)$ whoami
x_99285267
```

11. There are two typical paginating tools in Linux: **more**, an older one, and **less**, the newer option. When the screen vertical size is not sufficient, one can use the **pipe** character—vertical bar—to forward the output to a preferred paginating utility. Which one to use is a matter of personal preference.

    a. Experiment with the **more** tool. At the command prompt, type **ls --help | more** and press the **Enter** key.

    ```
    ls --help | more
    ```

    The **Enter** key advances the output by a single line. The **space bar** moves by the screenful. Pressing **Q** (either upper or lower) will terminate the output.

    ```
    Usage: ls [OPTION]... [FILE]...
    List information about the FILEs (the current directory by default).
    Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.

    Mandatory arguments to long options are mandatory for short options too.
      -a, --all                  do not ignore entries starting with .
      -A, --almost-all           do not list implied . and ..
          --author               with -l, print the author of each file
      -b, --escape               print C-style escapes for nongraphic characters
          --block-size=SIZE      scale sizes by SIZE before printing them; e.g.,
                                   '--block-size=M' prints sizes in units of
                                   1,048,576 bytes; see SIZE format below
    --More--
    ```

    b. Experiment with the **less** tool. It is quite similar to the **more** tool but does have some differences. At the command prompt, type **ls --help | less** and press the **Enter** key.

    ```
    ls --help | less
    ```

The **Enter** key advances the output by a single line. The **space bar** moves by the screenful. Pressing **Q** (either upper or lower) will terminate the output.

```
Usage: ls [OPTION]... [FILE]...
List information about the FILEs (the current directory by default).
Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.

Mandatory arguments to long options are mandatory for short options too.
  -a, --all                  do not ignore entries starting with .
  -A, --almost-all           do not list implied . and ..
      --author               with -l, print the author of each file
  -b, --escape               print C-style escapes for nongraphic characters
      --block-size=SIZE      scale sizes by SIZE before printing them; e.g.,
                               '--block-size=M' prints sizes in units of
                               1,048,576 bytes; see SIZE format below
:
```

12. To print out the contents of a text document, a simple command **cat** (or catenate) can be used.

    a. At the command prompt, type **cat keys/OL9lab_key** to print out the private key characters.

    ```
    cat keys/OL9lab_key
    ```

    Note: Be careful with the unknown file contents. Special characters may cause the terminal to misbehave.

    b. View the file type using the **file** command.

    1) At the command prompt, type **file keys/OL9lab_key** and press the **Enter** key.

    ```
    file keys/OL9lab_key
    ```

    You will see the type of the private key file contents.

    ```
    x_99285267@cloudshell:~ (us-asburn-1)$ file keys/OL9lab_key
    keys/OL9lab_key: OpenSSH private key
    ```

    2) At the command prompt, type **file /usr/bin/bash** and press the **Enter** key.

    ```
    file /usr/bin/bash
    ```

    You will see a very different output.

13. Learn to navigate directories in Linux.

    a. Explore a typical Linux file system. At the command prompt, type **ls -l /** and press the **Enter** key.

    ```
    ls -l  /
    ```

```
x_99285267@cloudshell:~ (us-ashburn-1)$ ls -l /
total 80
lrwxrwxrwx.   1 root root    7 Jan 24 22:47 bin -> usr/bin
dr-xr-xr-x.   2 root root 4096 Apr 11  2018 boot
drwxr-xr-x.  14 root root 3160 Mar  8 16:59 dev
drwxr-xr-x.   1 root root 4096 Mar  8 16:59 etc
drwxr-xr-x.   3 root root 4096 Mar  1 12:20 ggs_client
drwxr-xr-x.   1 root root 4096 Mar  8 16:59 home
lrwxrwxrwx.   1 root root    7 Jan 24 22:47 lib -> usr/lib
lrwxrwxrwx.   1 root root    9 Jan 24 22:47 lib64 -> usr/lib64
drwxr-xr-x.   2 root root 4096 Apr 11  2018 media
drwxr-xr-x.   2 root root 4096 Apr 11  2018 mnt
```

Note: All the objects in the file system root directory belong to the privileged user **root**. (Not all entries are shown.) Entries followed by an arrow pointing to another object are **links** (or shortcuts).

b. Navigate from the current directory to **/home**. At the command prompt, type **cd /home** and press the **Enter** key.

```
cd /home
```

Note: Using the **cd** command without parameters has the very same effect.

In the Cloud Shell, the prompt incorporates the current directory. This is not always the case. Command **pwd** is your reliable assistant.

```
pwd
```

```
x_99285267@cloudshell:~ (us-ashburn-1)$ pwd
/home/x_99285267
x_99285267@cloudshell:~ (us-ashburn-1)$ cd /home
x_99285267@cloudshell:home (us-ashburn-1)$ pwd
/home
```

c. The previous format was using an absolute path expression starting with the forward slash "**/**" character. The path may be also relative to the current location. Navigate to the **/bin** directory from the current **/home** directory using a relative path notation. At the command prompt, type **cd ../bin** and press the **Enter** key.

```
cd ../bin
```

That way, you used the parent directory of the current one by typing two period characters in front of the target directory.

```
x_99285267@cloudshell:home (us-ashburn-1)$ pwd
/home
x_99285267@cloudshell:home (us-ashburn-1)$ cd ../bin
x_99285267@cloudshell:bin (us-ashburn-1)$ pwd
/bin
```

You have now mastered some Linux commands that can be used in the Oracle Cloud Shell or any other Linux environment.

Solution 3-1: Experimenting with the OCI Cloud Shell

Overview

There is no solution for this practice. Please follow the practice steps.

Practice 3-2: Using VIM to Edit Files and Create a Simple Script

Overview

In this practice, you will use the VIM text editor to edit files and create a simple shell script.

Assumptions

You have completed all the practices in Lesson 2, and Practice 3-1, tasks 1 through 9 (inclusive).

Tasks

1. Open the Oracle Cloud Shell session.
   a. Log in to the OCI console, if not already logged in.
      1) Click the **Developer tools** button on the top toolbar of the OCI portal  and select the **Cloud Shell** option.
      2) It may take a few moments for the new session to launch. The Cloud Shell connection opens at the bottom of the screen.
2. Create a new directory in the Cloud Shell VM.
   a. In the previous practice, the **scripts** directory was created. Switch to this directory:

```
cd scripts
```

```
x_99285267@cloudshell:~ (us-ashburn-1)$ cd scripts/
x_99285267@cloudshell:scripts (us-ashburn-1)$
```

   b. Start the VIM (vi) editor. Type `vi  new-file.txt` at the prompt and press the **Enter** key.

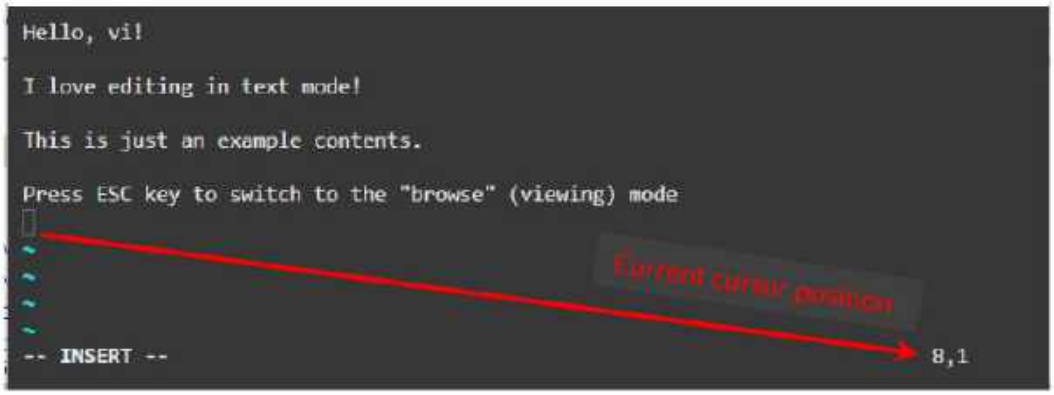
```
vi  new-file.txt
```

```
x_99285267@cloudshell:scripts (us-ashburn-1)$ vi new-file.txt
```

c. The new file **`new-file.txt`** opens in the editor window.

d. Press the **a** key. The editor enters **INSERT** mode.

e. Start typing some text. Look at the following screenshot for an example (it really does not matter what you type for this task). Feel free to use the arrow navigation keys to move around in the typed content.



f. Press the **ESC** key. The **INSERT** mode indicator at the bottom disappears. The editor is now in the "browse" mode and specific key presses represent commands.

g. Press the colon key **:** Command input opens at the bottom.

h. Type **f** and then press the **Enter** key:  File information will be shown.

```
"new-file.txt" [Modified][New file] 8 lines --100%--
```

i. Save the file. Type **:** and then type **w** (write command). Now type **q** (quit editor command):  When you press the **Enter** key, the contents of the file will be committed, and the editor will close.

3. Edit the existing file.

a. Use the up arrow key on your keyboard to recall the last command (**vi new-file.txt**) and press the **Enter** key. The editor opens in browse mode (usually, the last cursor position will be restored).

```
Hello, vi!

I love editing in text mode!

This is just an example contents.

Press ESC key to switch to the "browse" (viewing) mode

~
~
~
~
"new-file.txt" 8L, 133C                                       8,0-1
```

b. Search for some text in the file. Press the forward slash key **/** and follow with some text (for example, the word **just**), and then press the **Enter** key:  The editor locates matching text. Keep in mind that, by default, vi is case-sensitive.

```
Hello, vi!

I love editing in text mode!

This is just an example contents.

Press ESC key to switch to the "browse" (viewing) mode

~
~
~
~
search hit BOTTOM, continuing at TOP                                    5,9
```

Note: The search goes down from the current cursor position. To search in the reverse direction, press the question mark key instead of the forward slash.

c. Type uppercase **O**. A new line opens above your cursor position and the editor switches into **INSERT** mode. Lowercase **o** opens a new line below the cursor.

d. Type some text (for example, **something**).

```
Hello, vi!

I love editing in text mode!

something
This is just an example contents.

Press ESC key to switch to the "browse" (viewing) mode

~
~
~
-- INSERT --                                                    5,10
```

e. Press the **ESC** key. The **INSERT** mode indicator at the bottom disappears. The editor is now back in "browse" mode and specific key presses represent commands.

f. Type **f** and then press the **Enter** key: `:f` File information will be shown.

```
"new-file.txt" [Modified] 9 lines --55%--
```

g. Exit without saving the changes. Type **:** and then type **q!** The exclamation mark means **force** command execution.

4. Create a simple script. (Continue to work in the **scripts** directory.)

   a. Type **vi my-shell-script.sh** and press the **Enter** key to start editing the new script.

   ```
   vi  my-shell-script.sh
   ```

   b. The best scripting practice is to explicitly add the path of the interpreter for the script in the first line. We want this to be a **bash** script. The **bash** program usually resides in the **/bin** directory. Our first line of the script would look like: `#!/bin/bash`

   Note: Although the hash character is a comment symbol for the Linux command-line, the special sequence of the hash and the exclamation mark symbols - **#!** - is reserved for the identification of the script interpreter.

   c. Enter some of the familiar Linux commands in the following lines (for example, use the **whoami**, **ls**, and **pwd** commands).

   d. Save the new file.

      1) Press the **ESC** key.
      2) Type **:wq** and then press the **Enter** key.
      3) The editor closes.

5. List the contents of the directory.

   a. Type `ls -la` at the prompt and press the **Enter** key to obtain the long listing format.

```
ls -la
```

```
x_99285267@cloudshell:scripts (us-ashburn-1)$ ls -la
total 8
drwxr-xr-x. 2 x_99285267 oci  52 Mar  8 19:29 .
drwxr-xr-x. 5 x_99285267 oci 152 Mar  8 19:29 ..
-rw-r--r--. 1 x_99285267 oci  31 Mar  8 19:29 my-shell-script.sh
-rw-r--r--. 1 x_99285267 oci 133 Mar  8 19:23 new-file.txt
x_99285267@cloudshell:scripts (us-ashburn-1)$
```

   Make a note of the file access mask on the left of the newly created shell script. The file is read/write for the owner (you), and read-only for group members and others.

   b. To make the script file executable, use the **chmod** command and add the execute permission. Type `chmod u+x my-shell-script.sh` (sequence **u+x** for **chmod** tool means "add execute permission for the user-owner" only).

```
chmod u+x my-shell-script.sh
```

```
x_99285267@cloudshell:scripts (us-ashburn-1)$ chmod u+x my-shell-script.sh
x_99285267@cloudshell:scripts (us-ashburn-1)$ ls -la
total 8
drwxr-xr-x. 2 x_99285267 oci  52 Mar  8 19:29 .
drwxr-xr-x. 5 x_99285267 oci 152 Mar  8 19:29 ..
-rwxr--r--. 1 x_99285267 oci  31 Mar  8 19:29 my-shell-script.sh
-rw-r--r--. 1 x_99285267 oci 133 Mar  8 19:23 new-file.txt
x_99285267@cloudshell:scripts (us-ashburn-1)$
```

   Note: To add access permissions for the group, use **g+x**. For non-group-members (others), use the **o+x** sequence. A similar notation, with minus instead of plus, is used to take away the respective permission.

6. Run the new script.

   a. The Linux command-line interpreter looks for applications and scripts to run in the environment variable **PATH**. Unless you added the location to the **PATH** expression, you have to qualify the location of your script (or program) to run from. Use the current directory notation – period – followed by the path separator – forward slash symbol – before the script name in the current directory.

```
./my-shell-script.sh
```

```
x_99285267@cloudshell:scripts (us-ashburn-1)$ ./my-shell-script.sh
/home/x_99285267/scripts
x_99285267
total 8
drwxr-xr-x. 2 x_99285267 oci  52 Mar  8 19:29 .
drwxr-xr-x. 5 x_99285267 oci 152 Mar  8 19:29 ..
-rwxr--r--. 1 x_99285267 oci  31 Mar  8 19:29 my-shell-script.sh
-rw-r--r--. 1 x_99285267 oci 133 Mar  8 19:23 new-file.txt
```

Script output

Solution 3-2: Using VIM to Edit Files and Create a Simple Script

Overview

There is no automated solution for this practice.

Practice 3-3: Managing Users and Groups

Overview

In this practice, you will manage user accounts and group membership.

Assumptions

- You have completed the previous practices in this lesson.
- The authentication private key file is located in `D:\OL91abs\ol91ab-key`.

Tasks

1. Look up the instance's public IP.

   Note: All **public IP addresses** in these practices **are ephemeral** (temporarily assigned) and constitute no security risk. You will have your own public IP addresses to work with.

   a. In the OCI console, access **Instances**.

Name	State	Public IP	Private IP
lab-node1	Running	150.136.71.103	10.0.0.200

   b. Copy the Public IP value and record it in your scratch text file (for example, `lab-work.txt`). This practice assumes **150.136.71.103** as the public IP value.

   Note: By default, all OCI instances have ephemeral public IPs assigned. Upon re-creation (always), and (sometimes) after an update of the instance, this public IP address can be reassigned. There is an easy way to use a fixed, never-changing public IP. However, this is outside the scope of our course and practice.

   c. Open a command-prompt window on your local PC.

   d. Change the directory to the location where your private authentication key is stored.
   Note: In this document, the instructions follow MS Windows command-line syntax (CMD and not PowerShell).

      1) Type at the prompt `D:` and press the **Enter** key.
      2) Type at the prompt `cd \OL91abs` and press the **Enter** key.

e. Use the SSH command to connect to the instance.

Type `ssh -i OL9lab_key opc@150.136.71.103` and press the **Enter** key.

```
ssh -i OL9lab_key opc@150.136.71.103
```

f. Confirm that you accept the SSL public key of the instance upon first connection. The connection is now established. You land on the `lab-node1` home directory of the `opc` user.

```
D:\OL9Labs>ssh -i OL9lab_key opc@150.136.71.103
The authenticity of host '150.136.71.103 (150.136.71.103)' can't be established.
ECDSA key fingerprint is SHA256:VlUoAsdtaa3MIsFHlGrL4xYsczx02xngHYbXhhPLQJU.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '150.136.71.103' (ECDSA) to the list of known hosts.
[opc@lab-node1 ~]$
```

```
[opc@lab-node1 ~]$ pwd
/home/opc
```

User `opc` is not a privileged user; however, permission exists for this user account to run some privileged commands using the standard **sudo** tool to elevate privileges when required.

2. View the user information currently present in the OS.

a. User data is traditionally stored in the file `/etc/passwd`. At the Linux terminal prompt (within your open SSH connection), type `cat /etc/passwd` and press the **Enter** key.

```
cat /etc/passwd
```

The output may look like the following (partial output example):

```
[opc@lab-node1 ~]$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
```

b. Toward the end of the file, you may notice an entry for the **opc** user as well.

```
opc:x:1000:1000:Oracle Public Cloud User:/home/opc:/bin/bash
```

c. Tokens in this file are separated with the colon (:) character. The first token is the username and the second token is a placeholder for a password value (actual passwords are stored elsewhere). The next token is the user ID, followed by the user's group ID, description, home directory, and the login shell command interpreter application.

d. View the group information. Group data is stored in the **/etc/group** file. At the Linux terminal prompt, type **cat /etc/group** and press the **Enter** key.

```
cat /etc/group
```

The output may look like the following (partial output example):

```
[opc@lab-node1 ~]$ cat /etc/group
root:x:0:
bin:x:1:
daemon:x:2:
sys:x:3:
adm:x:4:oracle-cloud-agent,oracle-cloud-agent-updater,ocarun,opc
tty:x:5:
disk:x:6:
lp:x:7:
```

e. Search for the **opc** username occurrence in this file. Use the **grep** utility to filter the output.

```
cat /etc/group | grep opc
```

It may look like this example:

```
[opc@lab-node1 ~]$ cat /etc/group | grep opc
adm:x:4:oracle-cloud-agent,oracle-cloud-agent-updater,ocarun,opc
systemd-journal:x:190:opc
opc:x:1000:
```

Multiple occurrences of the **opc** user ID mean that this user is a member of multiple groups.

3. Add a new group.

a. Adding user and group information requires the privileged account. Make use of the **sudo** tool to execute the **groupadd** utility. You will be adding the **devops** group and then viewing the **devops** record in the **/etc/group** file. At the Linux terminal

prompt, type `sudo groupadd devops ; cat /etc/group | grep devops` and press the **Enter** key.

```
sudo groupadd devops ; cat /etc/group | grep devops
```

```
[opc@lab-node1 ~]$ sudo groupadd devops ; cat /etc/group | grep devops
devops:x:1001:
```

The group **devops** has been added without members.

Note: The semicolon character is used to separate distinct Linux commands, which may be entered on separate lines independently.

4. Add a new user.

   a. Add the new user with the username "nancy_k." Make use of the **sudo** tool to execute the **useradd** utility. At the Linux terminal prompt, type `sudo useradd nancy_k ; grep nancy_k /etc/passwd` and press the **Enter** key.

```
sudo useradd nancy_k ; grep nancy_k /etc/passwd
```

```
[opc@lab-node1 ~]$ sudo useradd nancy_k ; grep nancy_k /etc/passwd
nancy_k:x:1001:1002::/home/nancy_k:/bin/bash
```

User `nancy_k` has been created with the user ID **1001** and the new group ID **1002** (group ID **1001** was assigned to the **devops** group). Note that there is no description yet for the new user.

Note: The new user gets the home directory `/home/nancy_k` and a new Linux mailbox `/var/spool/mail/nancy_k` created as a result.

   b. Add the user description for the new user. At the Linux terminal prompt, type `sudo usermod -c "Nancy K. - new DevOps" nancy_k ; grep nancy_k /etc/passwd` and press the **Enter** key.

```
sudo usermod -c "Nancy K. - new DevOps" nancy_k ; grep nancy_k
/etc/passwd
```

Optionally, enter commands on separate lines as shown below:

```
[opc@lab-node1 ~]$ sudo usermod -c "Nancy K. - new DevOps" nancy_k
[opc@lab-node1 ~]$ grep nancy_k /etc/passwd
nancy_k:x:1001:1002:Nancy K. - new DevOps:/home/nancy_k:/bin/bash
```

The Comment field has the new description.

Note: The quoted comment value ensures that whitespace within the quoted text is not breaking the command-line syntax.

5. Add the user `nancy_k` to the **devops** group.

   a. At the Linux terminal prompt, type **`sudo usermod -a -G devops nancy_k ; grep nancy_k /etc/group`** and press the **Enter** key.

   ```
   sudo usermod -a -G devops nancy_k ; grep nancy_k /etc/group
   ```

   Optionally, enter commands on separate lines as shown below:

   ```
   [opc@lab-node1 ~]$ sudo usermod -a -G devops nancy_k
   [opc@lab-node1 ~]$ grep nancy_k /etc/group
   devops:x:1001:nancy_k
   nancy_k:x:1002:
   ```

6. Set the password value for the user. Privileged operation allows a password to be set without knowledge of the existing password (for any other user). When the user needs to change their own password, an old password value is typically required.

   a. At the Linux terminal prompt, type **`sudo passwd nancy_k`** and press the **Enter** key.

   ```
   sudo passwd nancy_k
   ```

   For the password, you may use any suitable value (for example, **Or@cle1234**).

   ```
   [opc@lab-node1 ~]$ sudo passwd nancy_k
   Changing password for user nancy_k.
   New password:
   Retype new password:
   passwd: all authentication tokens updated successfully.
   ```

   The password value is not reflected, so type carefully. In case of a mismatch, you will have to repeat the entries.

   Note: Password values are subject to password structure enforcement, which may vary. (Typically: at least eight-characters length with at least one of each – upper and lower case letters, and numbers.)

7. Test the login for the user **nancy_k**. Linux includes the **su** tool to switch identities. If the command is followed by the dash character, the target user's home directory initialization files will be processed.

   a. At the Linux terminal prompt, type **id** and press the **Enter** key.

```
id
```

```
[opc@lab-node1 ~]$ id
uid=1000(opc) gid=1000(opc) groups=1000(opc),4(adm),190(systemd-journal)
context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

   User information for the **opc** user is displayed along with the group membership.

   b. At the Linux terminal prompt, type **su - nancy_k** and press the **Enter** key.

```
su - nancy_k
```

   Enter the newly defined password to complete the login. This example shows the output of the **id** and **pwd** commands:

```
[opc@lab-node1 ~]$ su - nancy_k
Password:
[nancy_k@lab-node1 ~]$ id
uid=1001(nancy_k) gid=1002(nancy_k) groups=1002(nancy_k),1001(devops) con
text=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[nancy_k@lab-node1 ~]$ pwd
/home/nancy_k
[nancy_k@lab-node1 ~]$
```

   c. Optionally, use the **whoami** command.

```
whoami
```

   d. Type **CTRL-D** to exit the **nancy_k** account.

   e. Optionally, test **su nancy_k** without the dash and then run the **whoami** and **pwd** commands.

```
su nancy_k ; whoami ; pwd
```

```
[opc@lab-node1 ~]$ su nancy_k
Password:
[nancy_k@lab-node1 opc]$ whoami
nancy_k
[nancy_k@lab-node1 opc]$ pwd
/home/opc
```

Notice that the identity changed to the one of **nancy_k**, but the current directory stayed the same.

Note: Command **su** followed by dash character means executing user profile when switching user identity. Without the dash, only identity is changed, and the user profile is not processed, therefore the current working directory did not change.

f. Exit the **nancy_k** user account by either typing **CTRL-D** or typing **exit** and pressing the **Enter** key.

Solution 3-3: Managing Users and Groups

Overview

There is no automated solution for this practice.

Practices for Lesson 4: Operating System Management

Practices for Lesson 4

Overview

In these practices, you will use Oracle Linux operating system management tools.

Practices 4-2 and 4-3 depend on the following prerequisites: IAM artifacts – these had been configured for you.

Required IAM groups and a dynamic group for your Tenancy

In the Identity `Cloud Service` → `Identity domains` → **`default`** Identity domain, the following groups had been set up for you:

- User group `osmh-admins` to set up, configure and otherwise manage OSMH resources.
- User group `osmh-operators` to make use of the existing OSMH resources and functions.
- Dynamic group `osmh-instances` with the specific membership rules:

```
instance.compartment.id = '<tenancy_OCID>'
all {resource.type='managementagent',
resource.compartment.id='<tenancy_OCID>'}

instance.compartment.id = '<your_compartment_OCID>'
all {resource.type='managementagent',
resource.compartment.id='<your_compartment_OCID>'}
```

Note: The matching rules for the entire tenancy and similar rules for your allocated compartment had been preconfigured for your practice environment.

Required IAM Policies for a Compartment

To apply the policies for OS Management Hub only to a specific compartment inside the tenancy, the following policies had been defined (in most cases, similar policies would be present for the entire tenancy, as well):

```
ALLOW GROUP default/osmh-admins TO MANAGE osmh-family IN
COMPARTMENT <compartment_name>
ALLOW GROUP default/osmh-admins TO MANAGE management-agents IN
COMPARTMENT <compartment_name>
ALLOW GROUP default/osmh-admins TO MANAGE management-agent-
install-keys IN COMPARTMENT <compartment_name>
ALLOW GROUP default/osmh-admins TO USE appmgmt-family IN
COMPARTMENT <compartment_name>
ALLOW GROUP default/osmh-operators TO READ osmh-family IN
COMPARTMENT <compartment_name>
ALLOW GROUP default/osmh-operators TO USE appmgmt-family IN
COMPARTMENT <compartment_name>
```

```
ALLOW DYNAMIC-GROUP default/osmh-instances TO
{OSMH_MANAGED_INSTANCE_ACCESS} IN COMPARTMENT <compartment_name>
WHERE request.principal.id = target.managed-instance.id
ALLOW DYNAMIC-GROUP default/osmh-instances TO USE metrics IN
COMPARTMENT <compartment_name> WHERE target.metrics.namespace =
'oracle_appmgmt'
ALLOW DYNAMIC-GROUP default/osmh-instances TO
{MGMT_AGENT_DEPLOY_PLUGIN_CREATE,MGMT_AGENT_INSPECT,MGMT_AGENT_R
EAD} IN COMPARTMENT <compartment_name>
ALLOW DYNAMIC-GROUP default/osmh-instances TO
{APPMGMT_MONITORED_INSTANCE_READ,APPMGMT_MONITORED_INSTANCE_ACTI
VATE} IN COMPARTMENT <compartment_name> WHERE
request.instance.id = target.monitored-instance.id
ALLOW DYNAMIC-GROUP default/osmh-instances TO
{INSTANCE_READ,INSTANCE_UPDATE} IN COMPARTMENT
<compartment_name> WHERE request.principal.id =
target.instance.id
ALLOW DYNAMIC-GROUP default/osmh-instances TO
{APPMGMT_WORK_REQUEST_READ,INSTANCE_AGENT_PLUGIN_INSPECT} IN
COMPARTMENT <compartment_name>
```

```
ALLOW GROUP <your-user_group> TO MANAGE instance-agent-plugins
family IN COMPARTMENT <compartment_name>
ALLOW GROUP <your-user_group> TO MANAGE virtual-network-family IN
COMPARTMENT <compartment_name>
ALLOW GROUP <your-user_group> TO MANAGE instances IN COMPARTMENT
<compartment_name>
ALLOW GROUP <your-user_group> TO READ instance-images IN
COMPARTMENT <compartment_name>
```

```
ALLOW GROUP <your-user_group> TO MANAGE app-catalog-listing IN
COMPARTMENT <compartment_name>
ALLOW GROUP <your-user_group> TO MANAGE volumes IN COMPARTMENT
<compartment_name>
ALLOW GROUP <your-user_group> TO MANAGE volume-attachments IN
COMPARTMENT <compartment_name>
```

Required IAM Policies for Metrics

To allow the OS Management Hub service to emit metrics, the following policies had been configured:

```
ALLOW GROUP default/osmh-admins TO READ metrics IN COMPARTMENT
<compartment_name>
ALLOW GROUP default/osmh-operators TO READ metrics IN COMPARTMENT
<compartment_name>
```

Important: These policies must be specified at the tenancy level.

Required other OSMH resources

These resources had been preconfigured for your practice environment:

1. Software Sources for your OS version – set up on the root compartment/tenancy level.
2. OSMH Profile `OL9-sw-profile` – set up on the root compartment/tenancy level.

Practice 4-1: Enabling and Using OCI Utilities

Overview

In this practice, you will ensure that Oracle Cloud Infrastructure (OCI) utilities are available in your instance.

This activity may not always be necessary for the OS Management Hub functions to activate, depending on the Linux distribution release for OCI. However, you should consider these tasks to be useful, and, often, simply necessary to benefit from the OCI features applicable to daily management and monitoring.

Assumptions

- Practices for all previous lessons have been completed.
- Compute instance **`lab-node1`** that you created is running and has a public IP address.
- The private key to authenticate over SSH is stored in the file named **`OL9lab_key`**.

Tasks

1. Verify that the **`lab-node1`** instance is running.

   a. Log in to the OCI console.

   b. Navigate to **Instances** (verify compartment).

   c. Click the hyperlink of the **`lab-node1`** instance.

2. Connect to the instance using SSH.

   a. Record the **Public IP** value of the instance in a scratch text document if not previously recorded.

   b. Open a command-prompt window in your local host and navigate to the directory where the private key **`OL9lab_key`** was stored.

c. Execute the following command:

```
ssh  -i  OL9lab_key opc@<instance public IP>
```

d. The connection opens.

```
[opc@lab-node1 ~]$
```

Note: Confirm the public key suitability, if prompted.

3. Install the required packages.

a. Install Python SDK for OCI – **python39-oci-sdk**, **python39-oci-cli** and **oci-utils** packages. At your Linux terminal prompt, enter the following command (**-y** allows installation of any dependencies or upgrades without prompting):

```
sudo dnf install -y python39-oci-sdk python39-oci-cli oci-utils
```

```
[opc@lab-node1 ~]$ sudo dnf install -y python39-oci-sdk python39-oci-cli oci-utils
Last metadata expiration check: 0:29:30 ago on Fri 16 May 2025 06:46:43 PM GMT.
Package python39-oci-sdk-2.151.0-1.el9.aarch64 is already installed.
Package python39-oci-cli-3.55.0-1.el9.noarch is already installed.
Package oci-utils-0.14.0-14.el9.noarch is already installed.
Dependencies resolved.
Nothing to do.
Complete!
```

b. Agree to the upgrade or install dependencies, if not using **-y**, and prompted.

4. Make sure **ocid.service** is running.

a. At your Linux terminal prompt, enter the following command (skip the next step if active), press the "Q" key to return to the command prompt: **systemctl status ocid**

```
systemctl status ocid
```

```
[opc@lab-node1 ~]$ systemctl status ocid
○ ocid.service - Oracle Cloud Infrastructure utilities daemon
     Loaded: loaded (/etc/systemd/system/ocid.service; disabled; preset: disabled)
     Active: inactive (dead)
```

b. If not active, enable and start the service. At your Linux terminal prompt, enter the following command:

```
sudo systemctl enable --now ocid
```

```
[opc@lab-node1 ~]$ sudo systemctl enable --now ocid
Created symlink /etc/systemd/system/multi-user.target.wants/ocid.service → /etc/sys
temd/system/ocid.service.
[opc@lab-node1 ~]$ systemctl status ocid
● ocid.service - Oracle Cloud Infrastructure utilities daemon
     Loaded: loaded (/etc/systemd/system/ocid.service; enabled; preset: disabled)
     Active: active (running) since Fri 2025-05-16 19:20:14 GMT; 8s ago
   Main PID: 12449 (python3)
      Tasks: 4 (limit: 10388)
     Memory: 94.8M
        CPU: 4.847s
     CGroup: /system.slice/ocid.service
             └─12449 /usr/bin/python3 /usr/lib/python3.9/site-packages/oci_utils/i
```

5. Experiment with the **oci-iscsi-config** utility. Observe the generated output.

   a. At your Linux terminal prompt, enter the following command:

   ```
   sudo oci-iscsi-config
   ```

   b. At your Linux terminal prompt, enter the following command:

   ```
   sudo oci-iscsi-config -h
   ```

   The help message is displayed.

6. Experiment with the **oci-network-config** utility:

   a. At your Linux terminal prompt, enter the following command:

   ```
   sudo oci-network-config
   ```

   The current configuration is displayed.

   b. At your Linux terminal prompt, enter the following command:

   ```
   sudo oci-network-config -h
   ```

   The help message is displayed.

7. Obtain the public IP of the instance.

   a. At your Linux terminal prompt, enter the following command:

   ```
   oci-public-ip
   ```

8. Get the instance's comprehensive metadata.

   a. At your Linux terminal prompt, enter the following command:

   ```
   oci-metadata
   ```

   The entire set of metadata properties is displayed.

b. At your Linux terminal prompt, enter the following command and filter the output with the **grep** standard Linux utility:

```
oci-metadata | grep compartment
```

Only the compartment OCID is displayed.

c. Copy the compartment OCID into your scratch text file.

You will need it later in the practices.

9. It is possible to expand the root file system of the instance to its full configured size. This is a privileged command. You do not need to grow the file system for these practices. However, use the help option to see the syntax.

a. At your Linux terminal prompt, enter the following command:

```
sudo /usr/libexec/oci-growfs -h
```

Solution 4-1: Enabling and Using OCI Utilities

Overview

There is no automated solution for this practice.

Practice 4-2: OS Management Hub

Overview

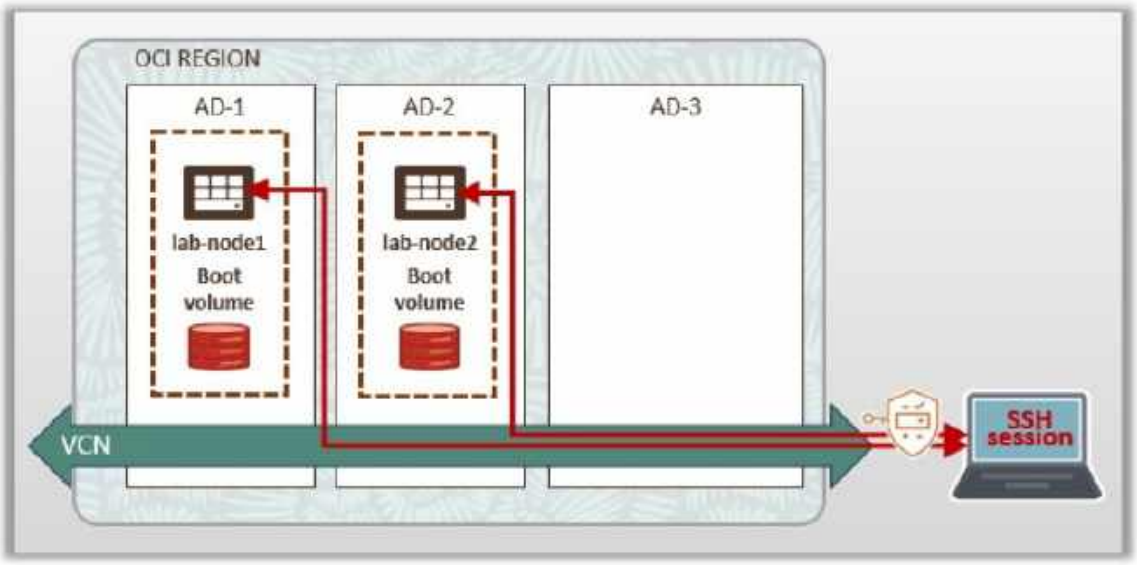
In this practice, you will work with OS Management.

- Set up OS Management Hub Agent.
- Create another Oracle Linux instance.

Certain activities will not result in the immediate visibility of the changes.

Some OS Management Hub operations take time to complete, please be patient.

Practice environment topology



Assumptions

- Practices for all previous lessons have been completed.
- Compute instance **`lab-node1`** that you created is running and has a public IP address.
- The private key to authenticate over SSH is stored in the file named **`OL9lab_key`**.

Tasks

1. Verify that the **`lab-node1`** instance is running.

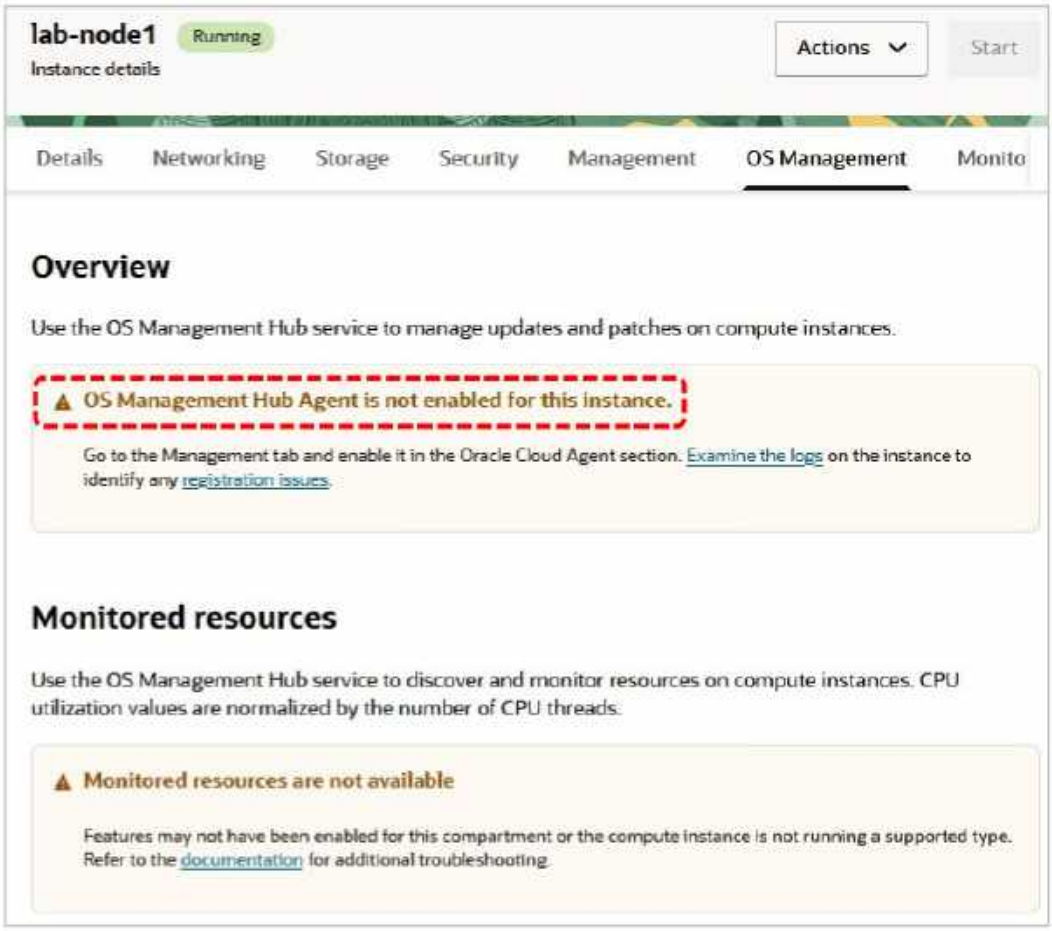
   a. Log in to the OCI console.

b. Navigate to **Instances** (verify compartment).

c. Click the hyperlink of the **`lab-node1`** instance.

2. View **OS Management** tab of the instance:

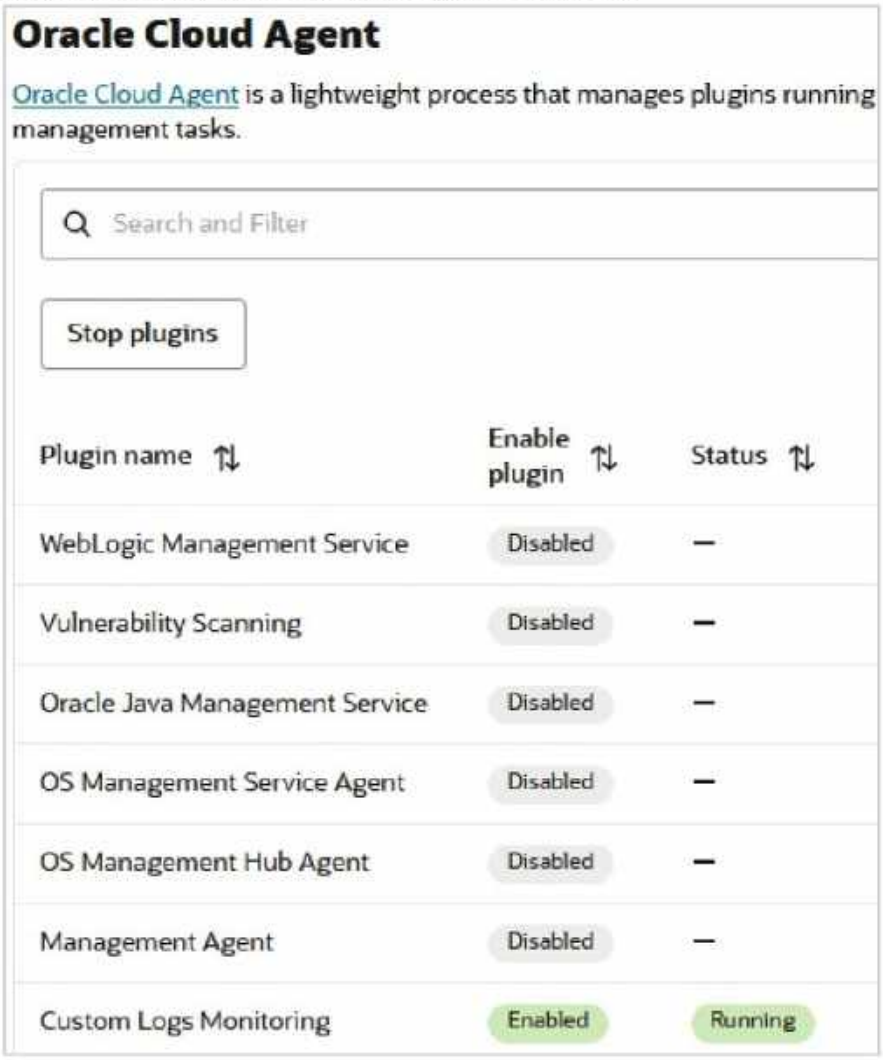
   a. On the **Instance details** page menu, click the **OS Management** tab. If the VM instance had been created without OS Management agent option, you will see the following view – notably, the part about OS Management Hub Agent not being enabled for this instance:



3. View OS Management Hub functionality.

   a. Click the **Management** tab on the **Instance details** menu.

   b. Scroll down to view the **Oracle Cloud Agent** details.

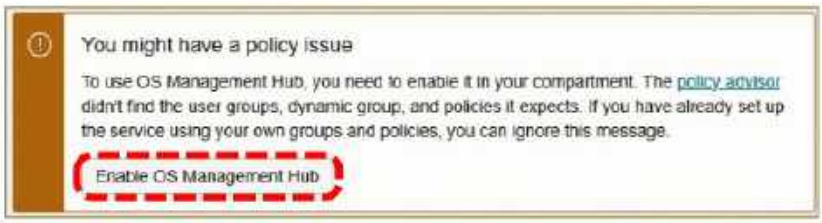


4. Verify the necessary policies for the OS Management Hub.

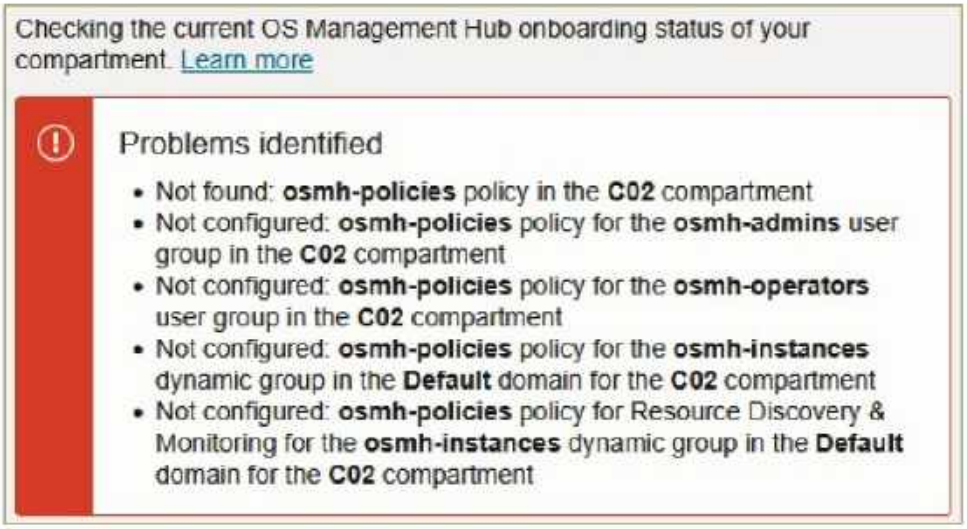
   Note: Your login user is not given permissions to manage policies – so necessary policies had been preconfigured for your environment. Policy advisor tasks are provided to illustrate the automatic set up capability of all required OSMH artifacts.

a. From the main OCI menu, click **Observability & Management**.

b. Click the **Overview** link. Examine the center panel. If you see the following warning, click the **Enable OS Management Hub** button to continue with the **Policy advisor** operation.



c. Policy advisor looks for the OSMH policy with the specific name **`osmh-policies`**, and can configure such policy for you automatically, if you have administrator permissions. Since the required policy was created manually for you and has a different name, policy advisor identifies issues (example follows). Click the **Run Policy Advisor** button even though you will not use the **Policy Advisor**'s capabilities.



d. At the foot of the panel, click the **Next** button.

e. Policy advisor shows the details of the OSMH artifacts needing to be defined. Partial examples follow. Your output may look different, since the policies had been configured for you.

The following groups and policies will be **created**. Existing items are listed for completeness.

User groups and dynamic group

The user groups that administer and operate the OS Management Hub service and the dynamic group of instances to manage.

User groups and dynamic group	Action
osmh-admins	No action
osmh-operators	No action

OS Management Hub Administrators group policies

Allows users in this group to manage OS Management Hub, Management Agents, and Management Agent Keys in this compartment and its subcompartments.

Policies	Action
Allow group Default/osmh-admins to manage osmh-family in compartment C02	Create
Allow group Default/osmh-admins to manage management-agents in compartment C02	Create

OS Management Hub Operators group policies

Allows users in this group to view OS Management Hub resources in this compartment and its subcompartments.

Policies	Action
Allow group Default/osmh-operators to read osmh-family in compartment C02	Create
Allow group Default/osmh-operators to use appmgmt-family in compartment C02	Create

f. Click the **Set up** button. Confirmation dialog is displayed.

g. Click the **Close** link. Your login user is not given permissions to manage policies (all necessary policies had been preconfigured for your environment). And you will receive an error message if you click **Set up** button on this dialog. This concludes the introduction to the OS Management Hub Policy advisor.

Note: Policy advisor does not create policy statement needed to turn on and off the agent plugins – absence of such policy will prevent you from enabling the OSMH and other OCI instance agents. If working with the agent plugins in your own environment, ensure that policy statement like below is present in one of your compartment policies:

```
ALLOW GROUP <your-user_group> TO MANAGE instance-agent-plugins
IN COMPARTMENT <your-compartment_name>
```

5. Verify Software sources for your OS version and architecture.

a. From the main OCI menu, click **Observability & Management**.

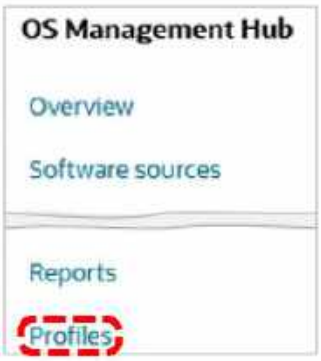
b. On the lower-right, under the **OS Management Hub**, select option: **Software sources**. (You may have to scroll down to see this section.)

c. Select the root compartment – that is where the software sources were defined.

d. Examine the list of software sources matching your OS and architecture – partial list is shown below:

6. Verify OSMH Profiles for your OS and architecture.

   a. From the **OS Management Hub**, click **Profiles**.

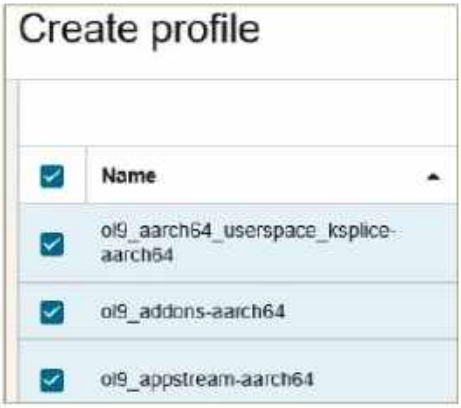


7. Create a new profile on the **Profiles** panel for your compartment.

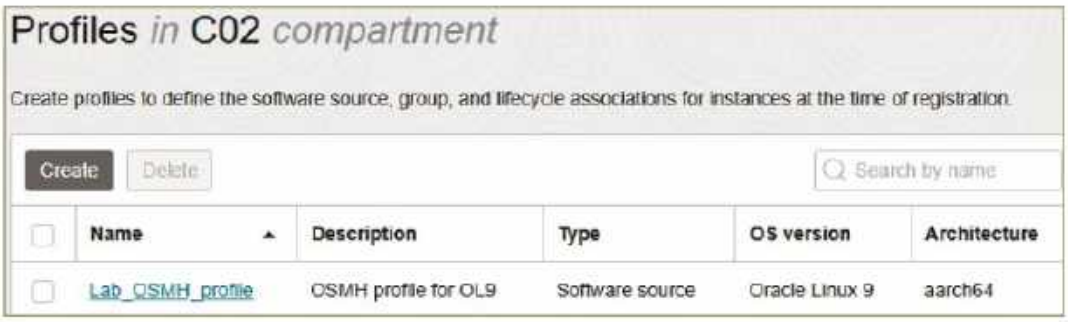
   a. Ensure your compartment appears in Compartment. Click the **Create** button.

b. Complete the profile details. Please only select the indicated sources, additional sources significantly prolong the required time to set up.

Property	Value
Name	`Lab_OSMH_profile`
Description	`OSMH profile for OL9`
Profile instance type	`Oracle Cloud Infrastructure`
OS vendor	`Oracle`
OS version	`Oracle Linux 9`
Architecture	`aarch64`
Type	`Software source`
Software source compartment	(root compartment only)
Software sources all names ending with '**`-aarch64`**'	`ol9_appstream, ol9_aarch64_userspace_ksplice, ol9_baseos_latest, ol9_addons`



c. Click the **Create** button.



8. Enable the OSMH agent for the instance.

   a. From the main OCI menu, click **Compute → Instances**.

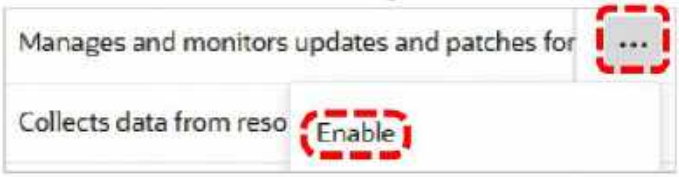
   Tip: Instances may be pinned to the **Home** screen and/or show under **Recently visited.**

   b. Click your instance hyperlink to see the Instance details panel.

   c. Click the **Management** tab.

   d. Scroll down to the **Oracle Cloud Agent** section and locate the row for the **OS Management Hub Agent**.

   e. Click the **Action** menu on the right and select **Enable**.



f. **Set profile** page appears.

g. Ensure your compartment and the profile name match, then click **Set profile** button on the bottom.

h. Panel with the **Oracle Cloud Agent** view is restored. **OS Management Hub Agent** displays the **Stopped** state. This is normal. Activation of the agent takes time. While this process runs, your instance SSH connection, if left open, will close. You will not be able to reconnect to the instance or interact with it while the agent is being activated.

9. While the OSMH Agent setup on **lab-node1** is progressing, you will create another instance. This new instance will be used in subsequent practices.

a. From the main OCI menu, select **Compute → Instances**, verify your assigned compartment and create a new VM as follows:

New instance attribute	Value	Notes
Name	`lab-node2`	
Compartment	<your-compartment>	Your assigned OCI compartment
Availability domain	Example: `AD 2`	If multiple Availability domains are accessible,

		select an AD **different from lab-node1 AD**
Image: **Change image**	`Oracle Linux 9`	
Shape: **Change shape**	**`Ampere`**	
Shape name	`VM.Standard.A1.Flex`	
Number of OCPUs	1	
Amount of memory (GB)	**4**	
Security panel	All defaults	
Primary network	Default: **`labVCN`**	Existing VCN
Subnet	Default: **`subnet-labVCN`**	Existing public subnet
Add SSH keys	Upload public key file	
Select or drag from local file system	**`OL9lab_key.pub`**	
Storage panel	All defaults	
Review panel	Review settings and **Create**	

b. When the instance **lab-node2** provisioning completes (typically a few minutes), repeat OSMH Agent enabling for the **lab-node2**. Refer to the previous task – ***Enable the OSMH agent for the instance*** if more details are desired.

See tasks 4 through 6 and task 8 for details. Reuse the existing Software profile that was created in task 7. There is no need for an additional profile.

10. View OS Management Hub Jobs.

a. From the main OCI menu, select **Observability & Management → OS Management Hub → Jobs.**

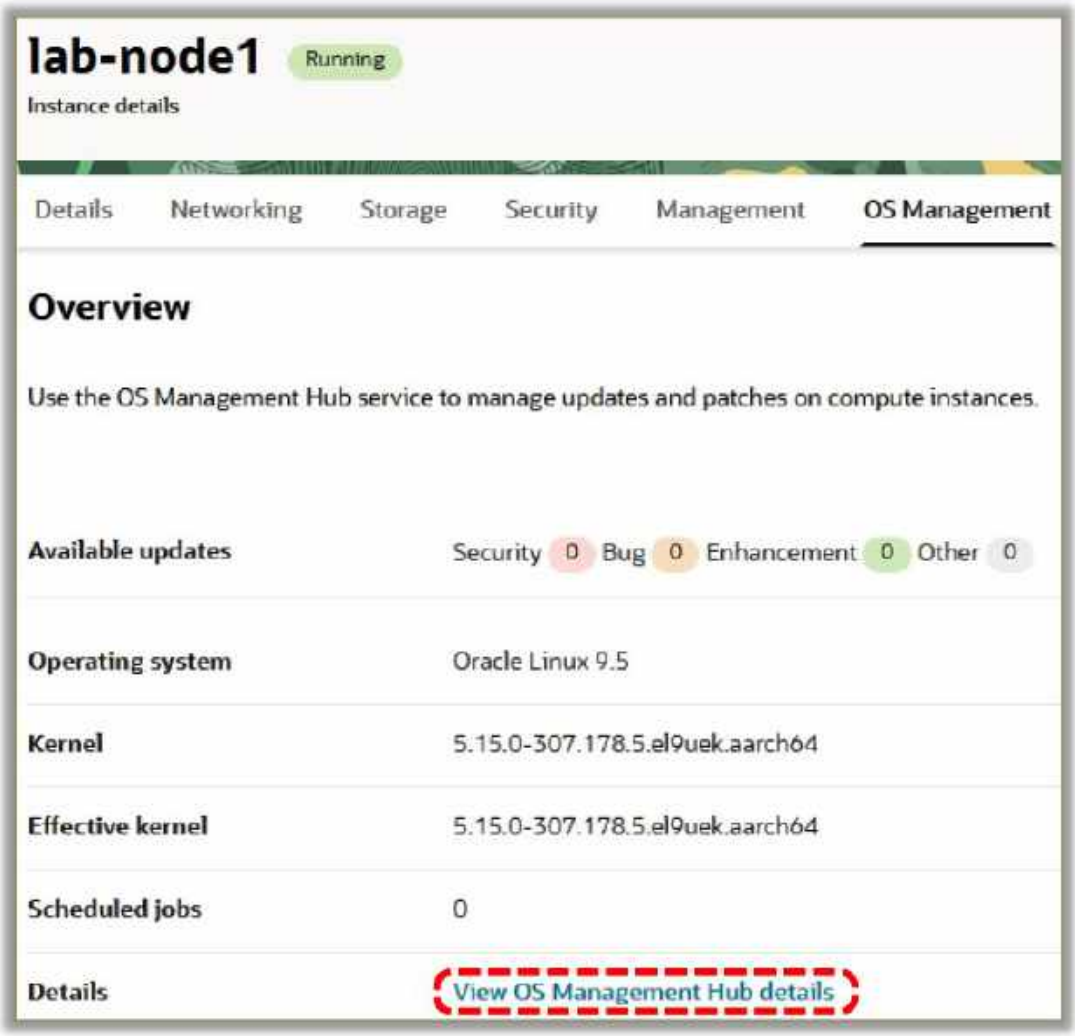
b. Click **In progress jobs** tab. If not present on this tab, try the **Completed jobs** tab. You should see the following entry (either by itself, or among other jobs):

Note: count of the software sources may be different for you.

c. Once the job has progressed, the OSMH Agent should indicate the **Running** state.

Tip: It is a good point in your practice session to take a break while waiting for this job to run its course. Sometimes, this job takes significant time.

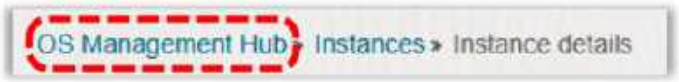
d. When the agent activates, you can view the **OS Management Hub** tab of the instance. Refer to task ***View OS Management tab of the instance*** in the beginning of this practice. While the activation job is still in progress, you will likely see the following output:



e. Absence of Available updates is normal, while the activation job is still in progress. This will likely change after the job completes. If the OS Image you have used is very recent, there may be no updates (rare). You still need to wait for the job to complete.

f. Click **View OS Management Hub details** hyperlink OS Management Hub Dashboard view may show "all clear" state at this time. Examine the list of any available updates.

Then click the **OS Management Hub** hyperlink on the top left to access the **Overview**.



Note: when there are multiple instances, circle coloring may have multiple sectors of different color, depending on the OSMH agents' reporting statuses. Depending on your navigation, **Dashboard** can display overall statuses or the status of an individual instance. You may also explore the **Reports** sections.

11. Once the job completes, you will likely see the following **Dashboard** state (depending on available Security updates and Bug updates):

12. Apply Bug updates.

    a. Hover your mouse cursor over the **Bug updates report** tile.

    b. When the count of instances with outstanding Bug updates shows, click the tile.

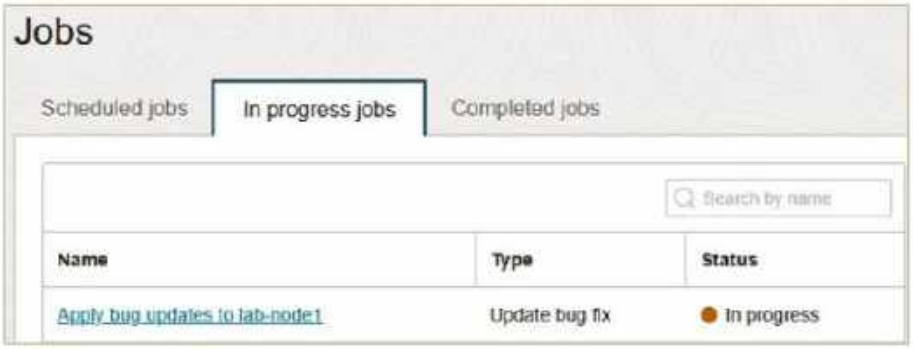
c. Panel **Bug updates report** displays, showing count of available updates:

d. Select the affected instance(s), then click **Apply updates** button.

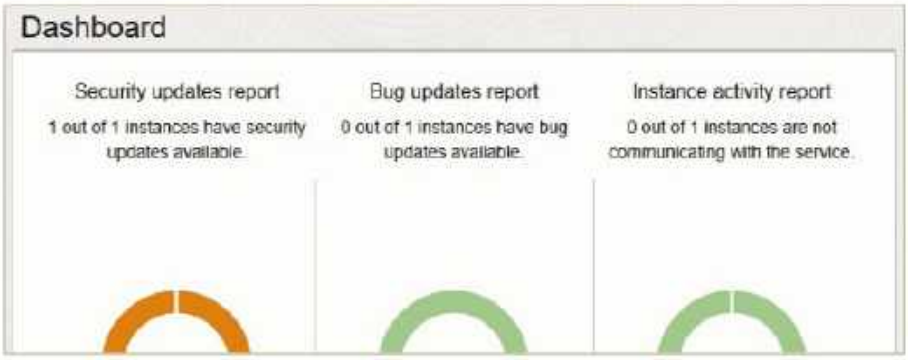
Note: you could click the instance's hyperlink to view which updates are available.

e. On the new Job scheduling panel, keep (or modify) the **Job name**, add an optional description, select Run immediately, and click the **Apply** button.

f. A new update job will appear in the jobs section in a few minutes:



g. Wait for the job to complete (several minutes, typically), then return to the **Dashboard** panel. Your view should indicate no outstanding **Bug updates**.



Note: when there are multiple instances, circle coloring may have multiple sectors of different color, depending on the OSMH agents' reporting statuses.

Optional step: Locate the completed update job to view its details. Navigate to: **OS Management Hub** → **Completed jobs** tab → **Apply bug updates to lab-node1** hyperlink. On the Job details panel, click the **Apply bug updates to lab-node1** hyperlink and examine Log and Error messages.

Note: If looking at the **OS Management Hub Overview** panel, the **Completed jobs** tab may show **Apply bug updates to 2 instances** – click on that hyperlink to view individual instance's jobs. For each Log or Error message, on the far right, you can expand the message details.

Solution 4-2: OS Management Hub

Overview

There is no automated solution for this practice.

Practice 4-3: Creating a Managed Instance Group

Overview

In this practice, you will:

- Create an OSMH Group.
- Add two existing instances to the new OSMH Group.
- View and apply updates to ensure Compliance of the instances.

Assumptions

- The **`lab-node1`** and **`lab-node2`** instances are available and running.
- The OSMH agents of **`lab-node1`** and **`lab-node2`** are enabled and communicating.
- The private key to authenticate over SSH is stored in the file named **`OL9lab_key`**.
- All policy rules permitting your user identity to manage OSMH features are in effect.

Tasks

1. Ensure that the **OS Management Hub agent** has been enabled for the **lab-node2** instance. If it is **Disabled**, refer to the previous practice's task 8 steps to enable it. Use the existing **Lab_OSMH_profile** profile for the **lab-node2** agent setup. Wait for completion.
2. Create a new Managed Instance Group.

   a. Click the OCI console "hamburger" menu: 

   b. Navigate to **Observability & Management → OS Management Hub → Groups** (under **OS Management Hub**).

   c. Verify the compartment and click the **Create** button. 

   d. In the Create Instance Group page, specify the following:
      1) OS Family: **`Linux`**
      2) Name: **`lab_Linux_group`**

3) User Description (optional): **`Managed Linux 9 instances`**

e. Click the **Next** button at the bottom of the panel. 

f. Select OS properties:

1) OS vendor – **`Oracle`**
2) OS version – **`Oracle Linux 9`**
3) Architecture – **`aarch64`**.

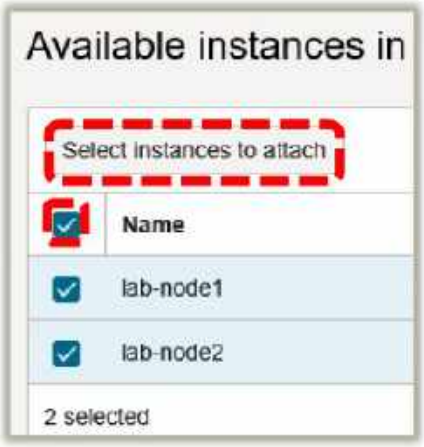
g. To view the **Available software sources**, select the **root** (tenancy) compartment.

h. Only select the following packages:
1) `ol9_appstream-aarch64`,
2) `ol9_aarch64_userspace_ksplice-aarch64`,
3) `ol9_baseos_latest-aarch64`,
4) `ol9_addons-aarch64`

i. Click **Next**

j. Make sure that your compartment is selected under the **Compartment**, Available instances appear in the list.

k. Select both instances: **`lab-node1`** and **`lab-node2`** and click **Select instances to attach** button.



l. Click **Next.**

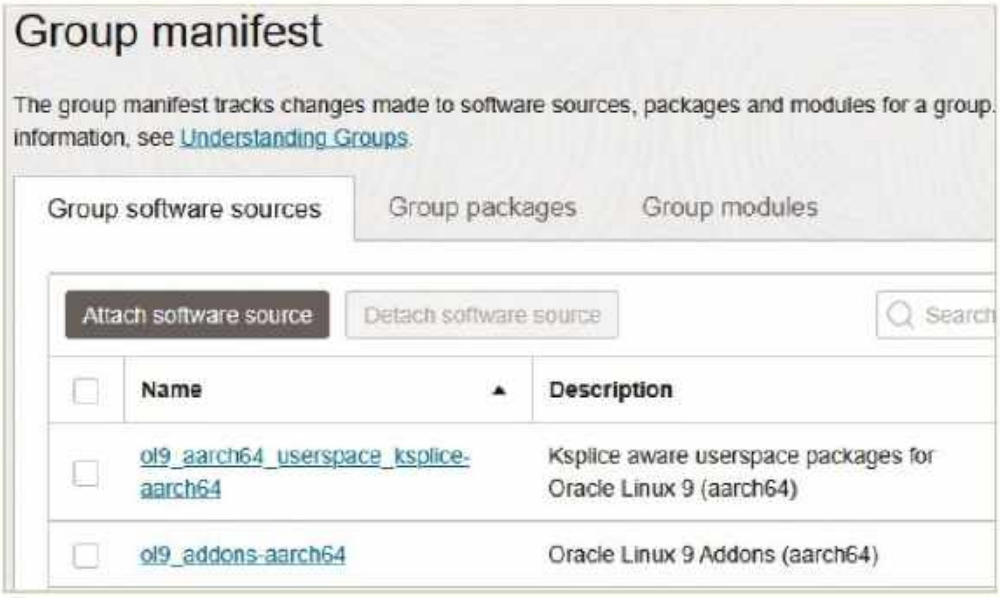
m. Review and click **Submit**.

3. View group status

a. Return to **OS Management Hub → Groups**.

b. Click the group's `lab_Linux_group` hyperlink to view the group's page.

c. To view and manage group's associated software sources, on the left, under the **Resources**, click the **Group manifest**:



d. (Optional) If you wish to see the packages from the software source, click on the source's hyperlink.

e. (Optional) Jobs, related to the group itself, could be seen in the **Jobs** panel. Only one job should be showing in the **Completed Jobs** panel:

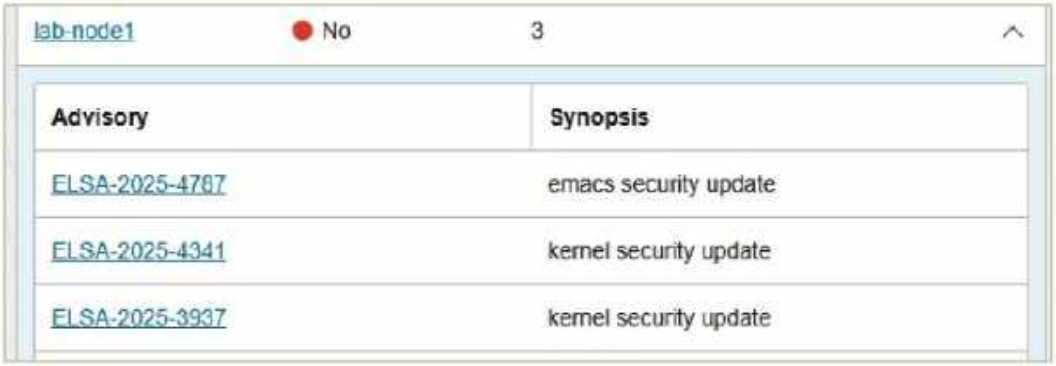
```
Assign 4 software sources to lab_Linux_group .
```

f. Click **Reports**. Group's `Security updates report` and `Bug updates report` indicate the Compliance status of the group.

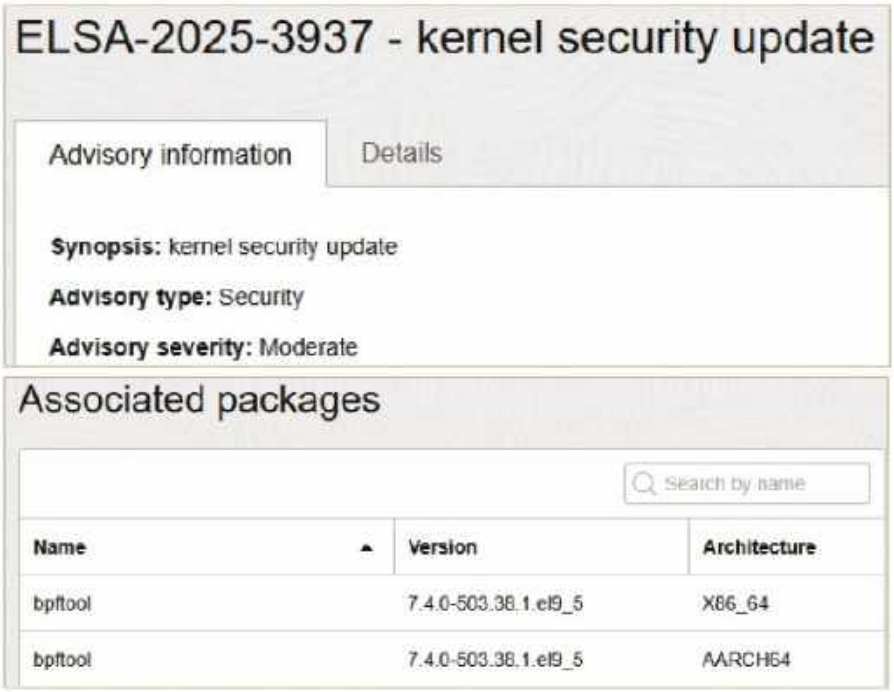
4. Click the **Security updates report** tile to visualize instances with available security updates.

   a. Take note of the update counts – these may be different across the member instances.

   b. Expand the list of available updates for one of the instances.



c. If you click on any of the listed **Advisories** hyperlink, associated packages would be shown. For example (a kernel security update is shown):



d. Section **Affected instances** provides the list of all group members, that need this update to mitigate an outstanding security issue.

5. Apply security updates to restore compliance.

a. Return to **OS Management Hub → Groups.**

b. Click the group's `lab_Linux_group` hyperlink to view the group's page.

c. Note the group operations toolbar contains the 

button – there you can reboot all group members or delete this group.

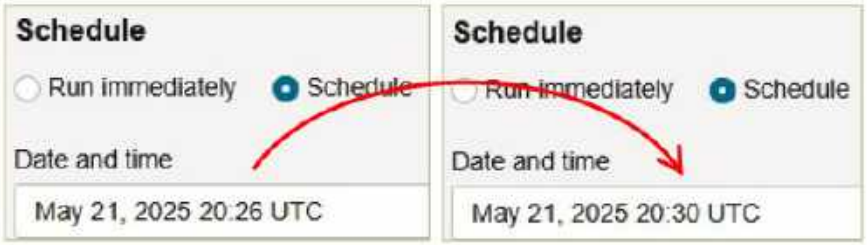
d. Click **Jobs.**

e. Create a job to apply security updates:

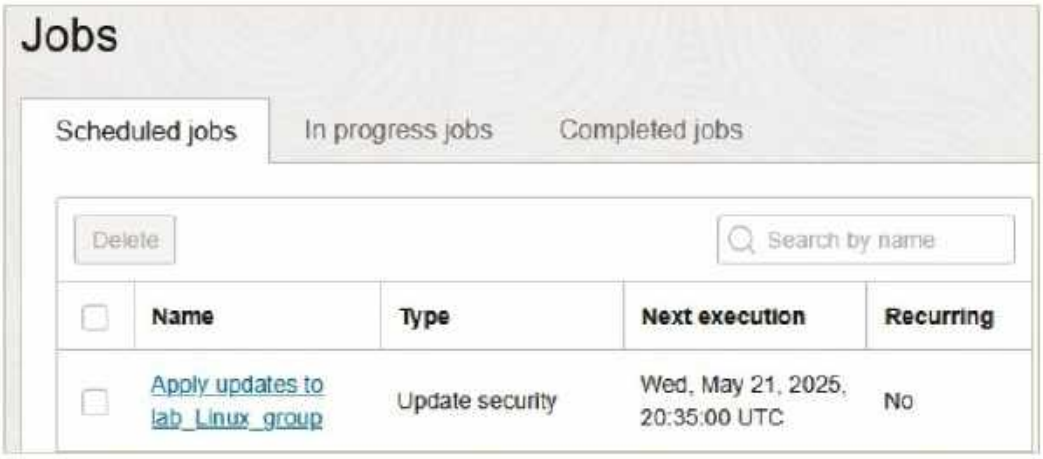
1) Click the Create update job button.

2) Keep or modify the job name, add a description, mark **Ksplice userspace** only check box.

3) You can run the job immediately or schedule it for later – if you want to schedule – add 3-4 minutes to the current execute time.

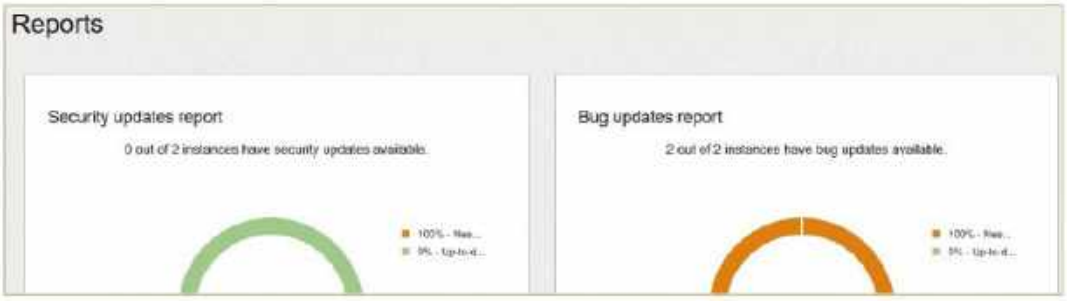


4) Click **Submit**. Look at the Jobs panel to see your new scheduled job.



f. When the job has completed (in about 2-5 minutes), click **Reports** again.

g. The **Security updates report** shows all instances up-to-date:



Security compliance had been restored!

6. (Optional) Apply bug fixes.

   a. Click **Bug updates report** tile.

   b. Create another update job – give it a name: `Apply bug fixes to lab_Linux_group`, optional description, and check only the **Bug fix** check box.

   c. Run immediately or schedule once again.

   d. Upon the above job completion, check the Reports panel for status.

   e. Final status may look like the example below:

You have successfully completed practices for this lesson! Great Job!

Solution 4-3: Creating a Managed Instance Group

Overview

There is no automated solution for this practice.

Practices for Lesson 5: Patching and GUI Configuration

Practices for Lesson 5

Overview

In these practices, you will:

- Use Ksplice to patch the OS packages
- Enable graphical user interface capability for an instance
- Install and use the Tiger VNC server

Practice 5-1: Linux Kernel Patching with Ksplice

Overview

In this practice, you use Ksplice to patch an instance OS without rebooting.

Assumptions

- Practices for Lesson 3 have been completed.
- Compute instance **lab-node1** that you created is running and has a public IP address.
- The private key to authenticate over SSH is stored in the file named **`ol9lab_key`**.

Tasks

1. Verify that the **lab-node1** instance is running.

   a. Log in to the OCI console.

   b. Navigate to **Instances** (verify compartment).

   c. Click the hyperlink of the **lab-node1** instance.

2. Connect to the instance using SSH.

   a. Record the **Public IP** value of the instance in a scratch text document if not previously recorded.

   b. Open a command prompt window in your local host and navigate to the directory where the private key **OL9lab_key** was stored.

   c. Execute the following command:

```
ssh -i  OL9lab_key opc@<instance public IP>
```

   d. The connection opens. `[opc@lab-node1 ~]$`

   Note: Confirm the public key suitability, if prompted.

3. OS patching requires root user privileges. Start a privileged **bash** session to complete the activities in the remainder of this practice.

   a. Type **sudo bash** and press the **Enter** key.

   ```
   sudo bash
   ```

   ```
   [opc@lab-node1 ~]$ sudo bash
   ```

   b. The privileged session begins. Use **whoami** to confirm the **root** user identity.

   ```
   whoami
   ```

   ```
   [root@lab-node1 opc]# whoami
   root
   ```

4. The Uptrack suite comprises tools for managing Ksplice updates of the OS. The Uptrack configuration file resides in **/etc/uptrack/uptrack.conf**.

   a. Examine the contents of the Uptrack configuration file. At the Linux terminal prompt, enter the following command: **more /etc/uptrack/uptrack.conf**

   ```
   more /etc/uptrack/uptrack.conf
   ```

   ```
   [root@lab-node1 opc]# more /etc/uptrack/uptrack.conf
   ```

   Note: Alternatively, use the following command: **less /etc/uptrack/uptrack.conf**

   ```
   less /etc/uptrack/uptrack
   ```

   You may also use the command **view /etc/uptrack/uptrack.conf** to view the file in **vi** editor read-only mode.

   ```
   view /etc/uptrack/uptrack.conf
   ```

   b. Find the **[Auth]** section on top of the screen.

   ```
   [Auth]
   accesskey = dfc21b3ced9a
   ```

   Note: The access key value above is truncated for security reasons.

   c. Using `more` tool, to advance the file display by one line, press the **Enter** key. To move by a screenful, use the **Space bar**. To quit viewing, type lowercase **q**.

d. Take note of the update repository information. It is not used by default because Oracle Cloud Infrastructure has an internal version hosted to reduce network costs of access.

```
# The location of the Uptrack updates repository.
#update_repo_url=
```

Uncomment and set the specific URL value if a different repository is required.

e. Note the **[Settings]** section and read the comments. Notice the `install_on_reboot` and `autoinstall` parameter meanings. Those are enabled by default.

f. Exit the viewing session. If using **more** or **less**, type lowercase `q`.

Note: If your terminal window misbehaves, use the command `reset`. (It does not happen often.)

5. View the Linux kernel version and installed updates on the system.

a. At the Linux terminal prompt, execute the command `uptrack-show`. Note the Effective kernel version value. If you are prompted to use `uptrack-upgrade -n` to download the latest package list, use that command.

```
uptrack-show
```

```
[root@lab-node1 opc]# uptrack-show
Installed updates:
None

Effective kernel version is 6.12.0-1.23.3.2.el9uek
```

b. Execute the standard Linux command `uname -r`.

```
uname -r
```

```
[root@lab-node1 opc]# uname -r
6.12.0-1.23.3.2.el9uek.aarch64
```

c. Now, execute the command `uptrack-uname -svr`.

```
uptrack-uname -svr
```

```
[root@lab-node1 opc]# uptrack-uname -r
6.12.0-1.23.3.2.el9uek.aarch64
```

Note: Your kernel details may be different.

6. View and apply upgrades with uptrack.

   a. At the Linux terminal prompt, execute the command **uptrack-upgrade.**

```
uptrack-upgrade
```

```
[root@lab-node1 opc]# uptrack-upgrade
The following steps will be taken:
Install [3pxnycek] Enablement update for live patching.
Install [46bgavfm] Known exploit detection.
Install [6vk1cquz] Known exploit detection for CVE-2016-8655.
Install [o7odxw03] Known exploit detection for CVE-2017-1000253.
Install [6i5x3x6j] Known exploit detection for CVE-2017-11176.
Install [otdbua54] Known exploit detection for CVE-2019-9213.
Install [2xp5jqum] Known exploit detection for CVE-2021-22600.
Install [l1gvpcio] Known exploit detection for CVE-2021-26708.
Install [l0paz4t3] Known exploit detection for CVE-2021-27363.
Install [jankrkvi] Known exploit detection for CVE-2021-27364.
Install [3qfwxd9q] Known exploit detection for CVE-2021-27365.
Install [o435ci65] Known exploit detection for CVE-2021-4034.
Install [8w45fk7r] Known exploit detection for CVE-2021-4154.
Install [8htq9ncf] Known exploit detection for CVE-2022-27666.
Install [hfyh87l9] Known exploit detection for CVE-2022-4378.
Install [edq6gxnl] Known exploit detection for CVE-2023-0386.
Install [1lmppsnn] Known exploit detection for CVE-2024-26925.
Install [m6fvgvx8] Known exploit detection for CVE-2023-4244.
Install [q4iq493u] Known exploit detection for CVE-2024-0193.

Go ahead [y/N]?
```

Note: Your list of available updates will be different.

   b. Agree to apply by typing **y**.

```
Installing [3pxnycek] Enablement update for live patching.
Installing [46bgavfm] Known exploit detection.
Installing [6vk1cquz] Known exploit detection for CVE-2016-8655.
Installing [o7odxw03] Known exploit detection for CVE-2017-1000253.
Installing [6i5x3x6j] Known exploit detection for CVE-2017-11176.
Installing [otdbua54] Known exploit detection for CVE-2019-9213.
Installing [2xp5jqum] Known exploit detection for CVE-2021-22600.
Installing [l1gvpcio] Known exploit detection for CVE-2021-26708.
Installing [l0paz4t3] Known exploit detection for CVE-2021-27363.
Installing [jankrkvi] Known exploit detection for CVE-2021-27364.
Installing [3qfwxd9q] Known exploit detection for CVE-2021-27365.
Installing [o435ci65] Known exploit detection for CVE-2021-4034.
Installing [8w45fk7r] Known exploit detection for CVE-2021-4154.
Installing [8htq9ncf] Known exploit detection for CVE-2022-27666.
Installing [hfyh87l9] Known exploit detection for CVE-2022-4378.
Installing [edq6gxnl] Known exploit detection for CVE-2023-0386.
Installing [1lmppsnn] Known exploit detection for CVE-2024-26925.
Installing [m6fvgvx8] Known exploit detection for CVE-2023-4244.
Installing [q4iq493u] Known exploit detection for CVE-2024-0193.
Your kernel is fully up to date.
Effective kernel version is 6.12.0-1.23.3.2.el9uek
```

Note: Your kernel versions may be different.

7. Compare the effective version from the Uptrack with the version reported by the **uname** utility. Example outputs below are shown for a different kernel version before and after.

   a. Execute the standard Linux command **uname -r**.

```
uname -r
```

```
Your kernel is fully up to date.
Effective kernel version is 5.15.0-103.114.4.el9uek
[root@lab-node1 opc]#
[root@lab-node1 opc]# uname -r
5.15.0-102.110.5.1.el9uek.aarch64
```

   b. Now, execute the command **uptrack-uname -svr**.

```
uptrack-uname -svr
```

   (The sample output below has been trimmed.)

```
t@lab-node1 opc]# uptrack-uname -svr
x 5.15.0-103.114.4.el9uek.aarch64 #2 SMP Mon Jun 26 10:21:19 PDT
```

   c. Compare the kernel versions reported by the previously mentioned utilities.

   Note: Kernel version may be the very same in both cases if your image is very recent and had utilized the latest available Kernel.

   d. Exit the privileged **bash** session by either typing **CTRL-D** or entering **exit** followed by pressing the **Enter** key.

8. View the Linux kernel information in the OCI console.

   Note: This information is only available after the OS Management Hub agent had fully activated on the instance. It may take several hours in extreme cases to complete. If you do not see this level of detail right away, try to check back in 2..4 hours.

   a. Return to the OCI console.

   b. Navigate to **Instances** (verify compartment).

   c. Click the hyperlink of the **lab-node1** instance.

d. Click **OS Management**. The **Overview** tab shows OS information.

Example immediately below is shown for a different kernel version before and after.

Operating system: oraclelinux-release

Kernel: 5.15.0-5.76.5.1.el9uek.x86_64

Effective kernel: 5.15.0-7.86.6.1.el9uek.x86_64

Note: Kernel version may be the very same in both cases if your image is very recent and had utilized the latest Kernel – shown in the example below.

Operating system	Oracle Linux 9.6
Kernel	6.12.0-1.23.3.2.el9uek.aarch64
Effective kernel	6.12.0-1.23.3.2.el9uek.aarch64

e. Return to **Instances** and repeat steps c and d for the **lab-node2** instance. Can you spot any differences?

Solution 5-1: Linux Kernel Patching with Ksplice

Overview

There is no automated solution for this practice.

Practice 5-2: Installing and Setting Up VNC Server for GUI

Overview

In this practice, you configure the GUI capability for your OCI Linux instance and set up the Tiger VNC server.

Assumptions

- Practices for Lesson 3 have been completed.
- Compute instance **lab-node1** that you created is running and has a public IP address.
- The private key to authenticate over SSH is stored in the file named **`OL9lab_key`**.

Tasks

1. If you are reusing the SSH connection from the previous practice, skip to task 4.
2. Verify that the **lab-node1** instance is running.

   a. Log in to the OCI console.

   b. Navigate to **Instances** (verify compartment).

   c. Click the hyperlink of the **lab-node1** instance.

3. Connect to the instance using SSH.

   a. Record the **Public IP** value of the instance in a scratch text document if not previously recorded.

   b. Open a command prompt window in your local host and navigate to the directory where the private key **ol9lab_key** was stored.

   c. Execute the following command:

```
ssh  -i  OL9lab_key opc@<instance public IP>
```

d. The connection opens.

```
[opc@lab-node1 ~]$
```

Note: Confirm the public key suitability, if prompted.

4. GUI configuration requires root user privileges. Start a privileged **bash** session to complete the activities in the remainder of this practice.

a. Type **sudo bash** and press the **Enter** key.

```
sudo bash
```

```
[opc@lab-node1 ~]$ sudo bash
```

b. The privileged session begins. Use **whoami** to confirm the **root** user identity.

```
whoami
```

```
[root@lab-node1 opc]# whoami
root
```

5. List the available environmental Oracle Linux groups.

a. At the privileged Linux command prompt, execute the command **dnf grouplist**.

```
dnf grouplist
```

```
[root@lab-node1 opc]# dnf grouplist
This system is receiving updates from OSMS server.
Last metadata expiration check: 2:50:04 ago on Thu
Available Environment Groups:
   Server with GUI
   Server
   Minimal Install
   Workstation
```

Note: Partial output is shown.

b. Install the **"Server with GUI"** group. Use quotation marks to ensure the integrity of the group name. Execute the following command:

```
dnf groupinstall -y "Server with GUI"
```

```
[root@lab-node1 opc]# dnf groupinstall -y "Server with GUI"
```

Note: Installation takes several minutes. If you get any errors about conflicting or broken dependencies, add `--skip-broken` and `--allowerasing` flags as follows:

```
dnf groupinstall -y "Server with GUI"  --skip-broken \
    --allowerasing
```

c. Optionally, repeat the command **dnf grouplist**. Notice that the installed group is now listed under **Installed Environmental Groups**.

d. Do not exit the privileged **bash** session yet.

6. Install the VNC server package.

a. At the privileged Linux command prompt, execute the following command (dependencies are pulled in automatically): **dnf install -y tigervnc-server**

```
dnf install -y tigervnc-server
```

```
[root@lab-node1 opc]# dnf install -y tigervnc-server
```

b. Exit the privileged **bash** session by either typing **CTRL-D** or entering **exit** followed by pressing the **Enter** key.

```
[root@lab-node1 opc]# exit
exit
[opc@lab-node1 ~]$ whoami
opc
```

```
exit
whoami
```

7. Configure the VNC server properties.

a. Ensure your current working directory is the home folder of the **opc** user. Execute the **cd** command without parameters. It will bring your current working directory back to the **/home/opc** location.

```
cd
```

b. Set the password for the VNC user. Execute the command **vncpasswd**. Enter and confirm the desired password. Because the view-only password is not a requirement, answer **n** when prompted.

```
vncpasswd
```

```
[opc@lab-node1 ~]$ vncpasswd
Password:
Verify:
Would you like to enter a view-only password (y/n)? n
A view-only password is not used
```

Hint: For example, use the following password: **Or@cle1234**

8. Add mapping for the VNC user **opc**.

   a. Execute the command to edit the **/etc/tigervnc/vncserver.users** file. This requires the **sudo** tool to make modifications.

      The command to run is as follows: **sudo vi /etc/tigervnc/vncserver.users**

```
sudo vi /etc/tigervnc/vncserver.users
```

```
[opc@lab-node1 ~]$ sudo vi /etc/tigervnc/vncserver.users
```

   b. The file opens for editing. Sample user mappings are present.

```
# TigerVNC User assignment
#
# This file assigns users to specific VNC display numbers.
# The syntax is <display>=<username>. E.g.:
#
# :2=andrew
# :3=lisa

~
```

   c. To move directly to the last line, press **SHIFT-G** and the cursor will move to the end of the text.

   d. Enter "edit" mode by typing the lowercase **a** key. The editor enters "insert" mode (**-- INSERT --** appears at the bottom of the screen).

   e. Add the mapping for the **:1** display and associate it with the **opc** user. Type **:1=opc** and press the **Enter** key.

```
# TigerVNC User assignment
#
# This file assigns users to specific VNC display numbers.
# The syntax is <display>=<username>. E.g.:
#
# :2=andrew
# :3=lisa
:1=opc
```

      Note: The VNC display number following the colon corresponds to the listen port of the VNC server. Display **:1** is associated with port **5901**, **:2** – with port **5902**, and so on.

   f. Press the **ESC** key and then save the changes by typing **:wq**.

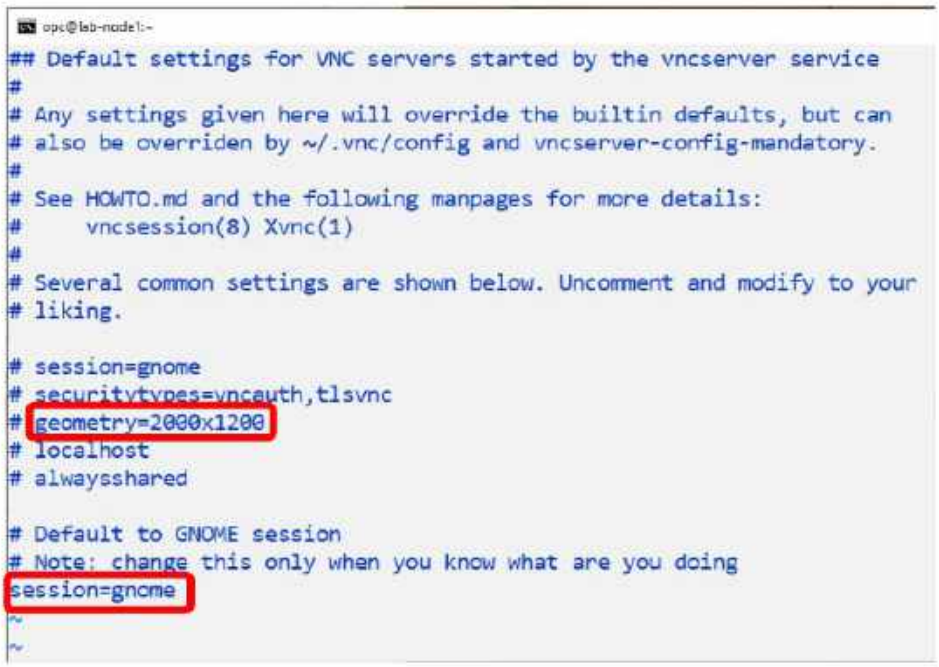
9. Verify the Linux desktop type. **Gnome** is the typical desktop.

   a. Run the following command: **`sudo vi /etc/tigervnc/vncserver-config-defaults`**

   ```
   sudo vi /etc/tigervnc/vncserver-config-defaults
   ```

   ```
   [opc@lab-node1 ~]$ sudo vi /etc/tigervnc/vncserver-config-defaults
   ```

   The file opens in the editor. Note the value for the **session**. It is **gnome** by default and is enabled. Also, note the default screen **geometry**. This is commented out; however, you can uncomment and set it to the desired initial screen resolution for the remote clients.

   b. Optionally, uncomment the geometry part and set resolution to **1440x900,** for instance. In that case, don't forget to save the changes. To position it to the **geometry** line, use keystrokes **/geometry** or use the navigation arrow keys.

   ```
   # geometry=2000x1200
   ####> example setting
   geometry=1440x900
   ```

   Note: To add a new blank line press lowercase "o" key, which opens a new line below the current position of the cursor.

   c. Exit the **vi** editor. (Press the **ESC** key followed by **:wq** key sequence, to save.)

10. Update boot time service configuration and enable the VNC server.

    a. Execute the following command: **sudo systemctl daemon-reload**

```
sudo systemctl daemon-reload
```

    b. Enable the VNC server by executing the following command:

```
sudo systemctl enable --now vncserver@:1.service
```

    c. Verify the status of the VNC server process by executing the following command:

```
systemctl status vncserver@:1
```

```
[opc@lab-node1 ~]$ sudo systemctl daemon-reload
[opc@lab-node1 ~]$
[opc@lab-node1 ~]$ sudo systemctl enable --now vncserver@:1.service
Created symlink /etc/systemd/system/multi-user.target.wants/vncserver@:1.service →
 /usr/lib/systemd/system/vncserver@.service.
[opc@lab-node1 ~]$ systemctl status vncserver@:1
● vncserver@:1.service - Remote desktop service (VNC)
     Loaded: loaded (/usr/lib/systemd/system/vncserver@.service; enabled; vendor >
     Active: active (running) since Fri 2023-03-10 14:01:08 GMT; 30s ago
```

11. Establish SSH port forwarding to access port 5901 on the OCI Linux instance.

    a. Open a new command-prompt window on your local PC.

    b. Change the directory to the location of the SSH authentication key. In our case, to the **D:\OL9labs** folder.

    c. Start SSH port forwarding in that window by running the following command (all on one line – text below was wrapped by formatting):

```
ssh -L 5901:localhost:5901 -i OL9lab_key
opc@<your_instance_public_IP>
```

```
D:\OL9Labs>ssh -L 5901:localhost:5901 -i OL9lab_key opc@150.136.150.14
Last login: Wed Jul 19 20:27:07 2023 from 99.228.228.74
[opc@lab-node1 ~]$
```

    d. Keep this connection open while using a VNC viewer to access the desktop.

    Note: Port forwarding moves the traffic between the local PC port 5901 to the instance's port 5901.

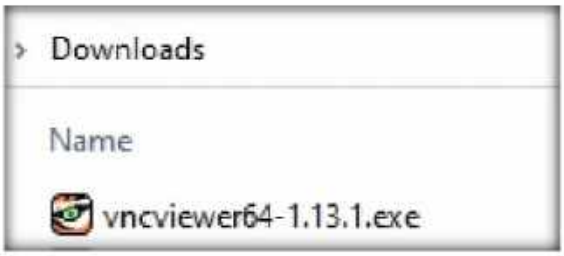
12. Start the GUI session with the OCI Linux host.

    a. Launch the VNC viewer on your local PC. The example uses the Tiger VNC viewer, which is freely available from the https://tigervnc.org downloads. Only the viewer is required, so do not download the entire suite for the purpose of this practice. Look for

`vncviewer64-1.13.1.exe` for the MS Windows download. The specific page for that release's binaries is https://sourceforge.net/projects/tigervnc/files/stable/1.13.1 TigerVNC - Browse /stable/1.13.1. at SourceForge.net. The latest release may have a newer version too. Download **vncviewer64-1.13.1.exe** vncviewer64-1.13.1.exe by clicking the link.

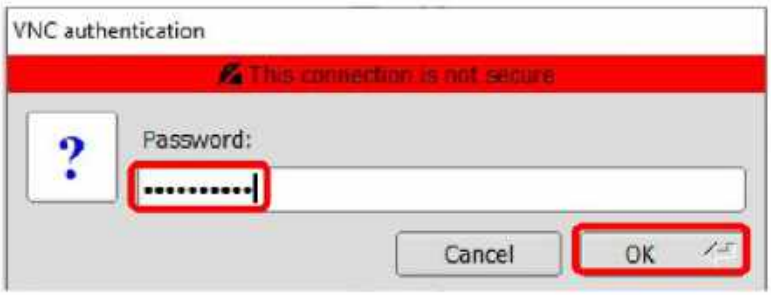
Note: A newer version may be available. Download the latest version.

b. Run the VNC viewer executable after installing.



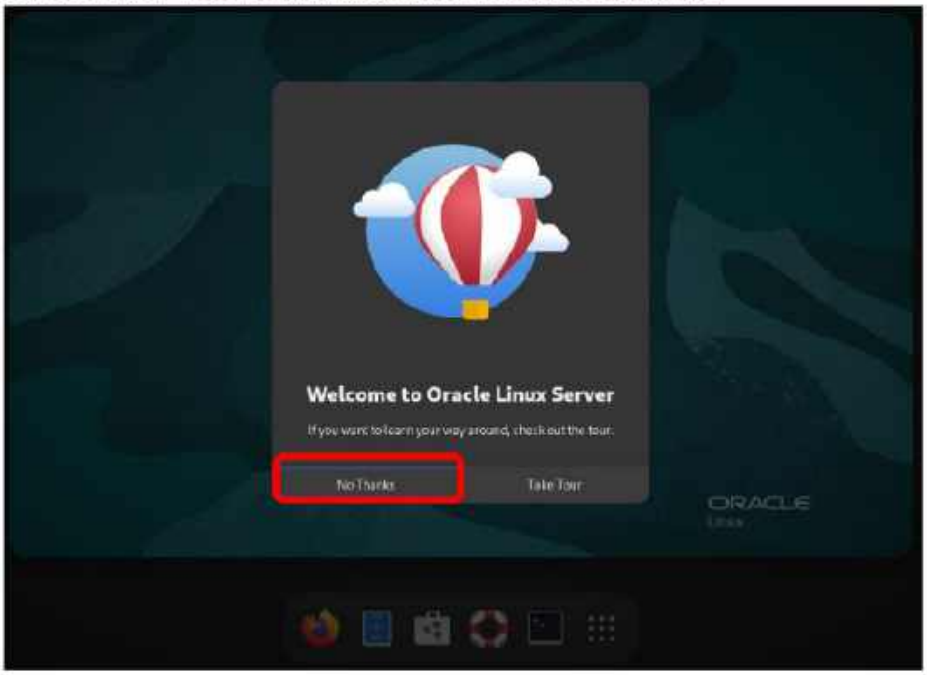
c. For the VNC server input, specify **localhost:5901**.

d. Click the **Connect** button. Enter the VNC user password that you defined for the `opc` user and click **OK**.



13. You will see some option selection dialog boxes when the newly displayed Linux desktop appears for the first time. Make your choices as needed.

    a. Use the **Take Tour**, or simply click **No Thanks** to continue.

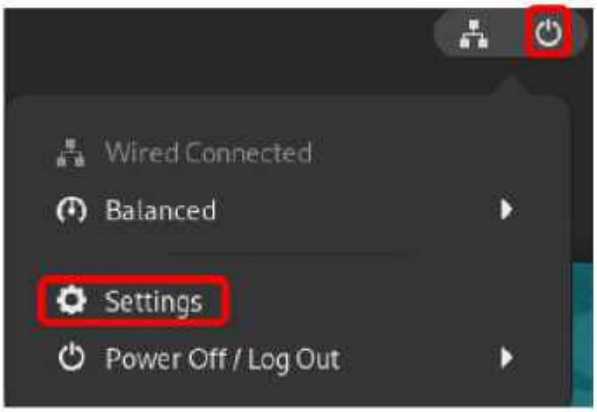


    b. You may be prompted to select the language and the keyboard layout. Complete these selections.

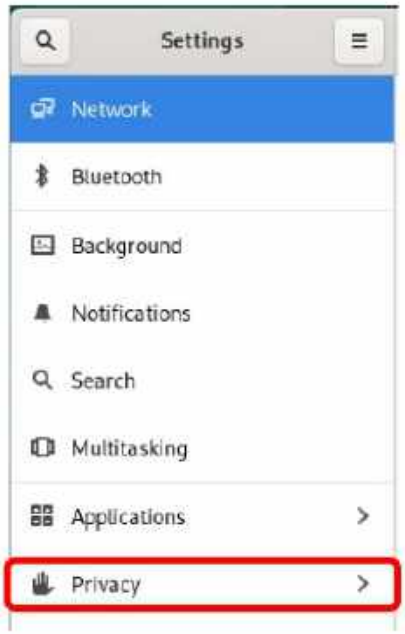
c. Gnome desktop would be displayed.

d. Ensure the location services are OFF – these are not useful for an OCI Linux host.

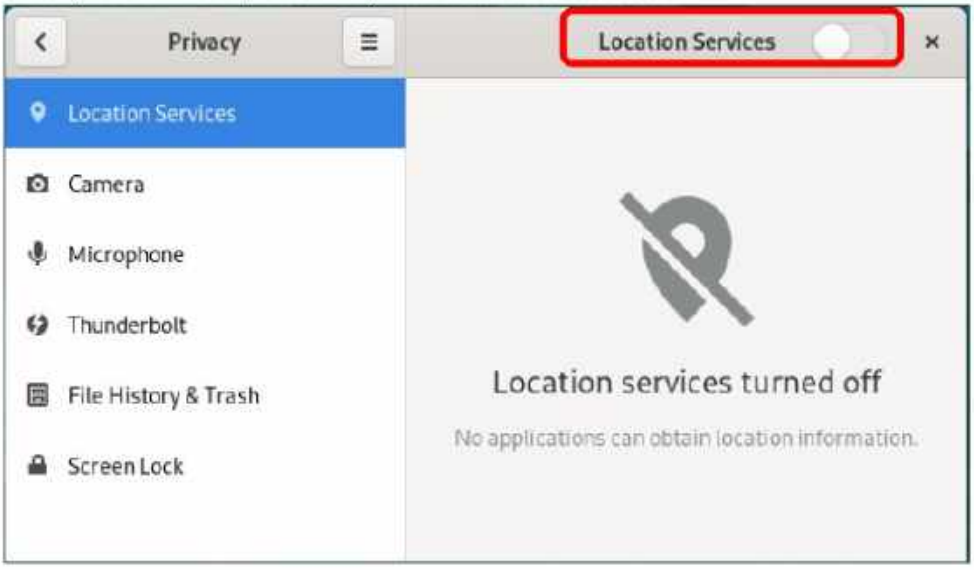
1) Click the power button at the top-right corner.
2) Click the **Settings** option.



3) Click **Privacy** on the panel.



4) View (and turn OFF, if needed) the **Location Services**.



e. Close the **Settings** panel by clicking top-right "**x**" icon.

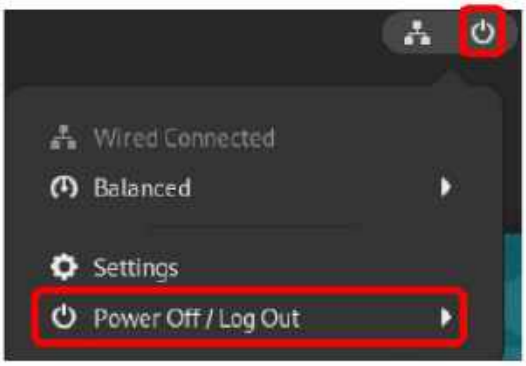
f. To begin using your GUI environment, click **Activities** at the top-left corner.

g. Floating island (toolbar) with popular choices appears on the bottom. Click any of the icons to access the functionality. To see more application choices, click the **Show Applications** button.

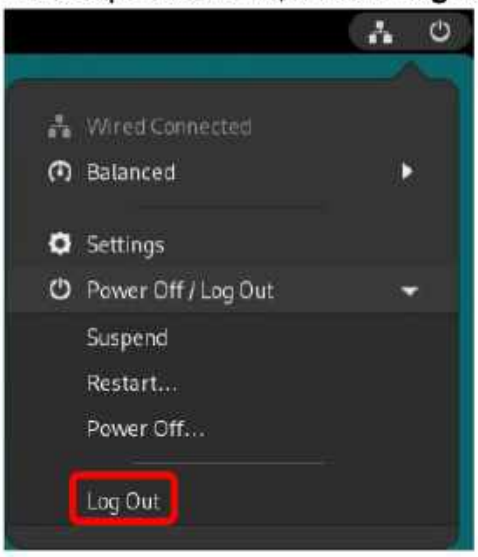
14. Log out of the Linux desktop session.

a. Click the power button at the top-right of the desktop.

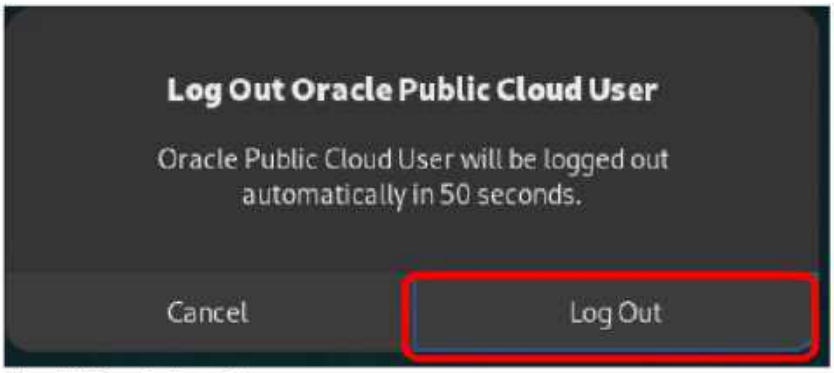
b. Then, click the **Power Off/Log Out** row.



c. In the expanded menu, click the **Log Out** menu option.



d. Confirm by clicking **Log Out** button.



The VNC window closes.

e. Stop the SSH tunnel. In the Linux command prompt window, where you had started the SSH tunnel functionality, close the connection by either typing **CTRL-D** or entering **exit** followed by pressing the **Enter** key.

Solution 5-2: Installing and Setting Up VNC Server for GUI

Overview

There is no automated solution for this practice.

Practices for Lesson 6: Managing iSCSI and OCFS Storage

Practices for Lesson 6

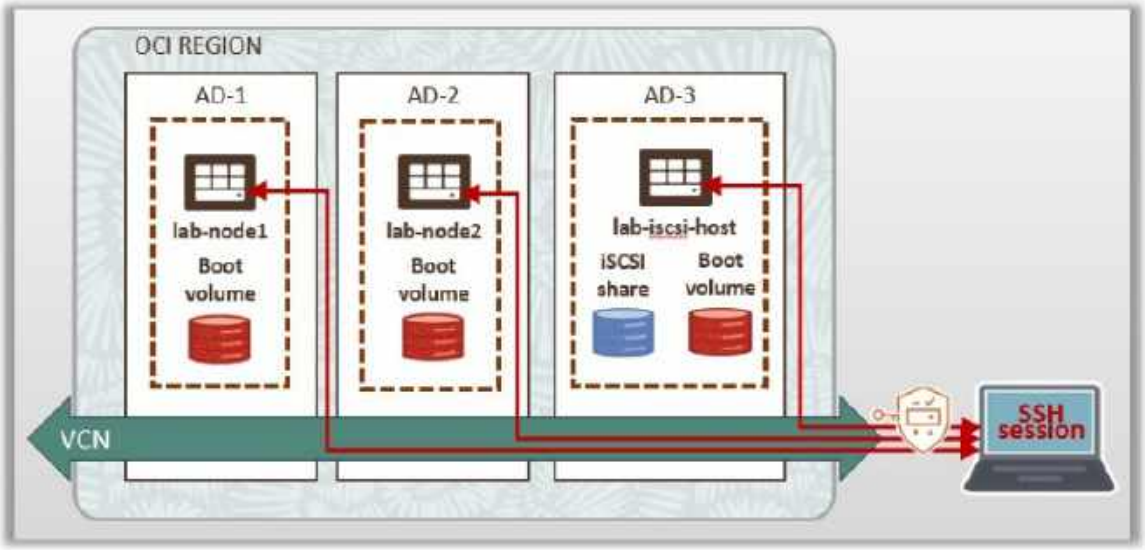
Overview

In these practices, you will manage the storage for the Linux instances in OCI.

You will be:

- Using OCI utilities to manage iSCSI storage
- Configuring a Linux iSCSI target and an initiator
- Configuring and testing an OCFS2 cluster

Practice environment topology



Practice 6-1: Using OCI Utilities to Manage iSCSI Storage

Overview

In this practice, you will confirm installation and make use of the OCI iSCSI command-line utilities.

Assumptions

- Practices for Lesson 3 have been completed.
- Compute instance **lab-node1** that you created earlier is running and has a public IP address.
- The private key to authenticate over SSH is stored in the file named **`OL9lab_key`**.

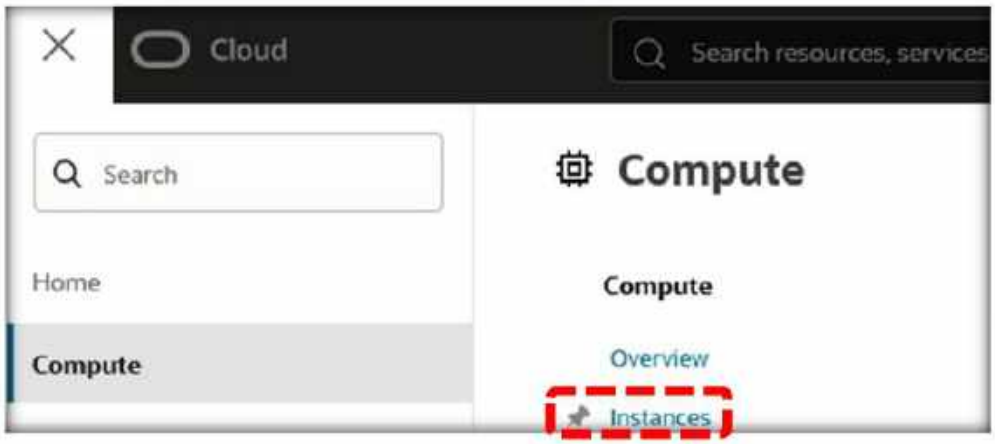
Tasks

1. Create an additional Oracle Linux instance named **lab-iscsi-host**. The steps are very similar to those for the first instance creation in the practices for Lesson 2.

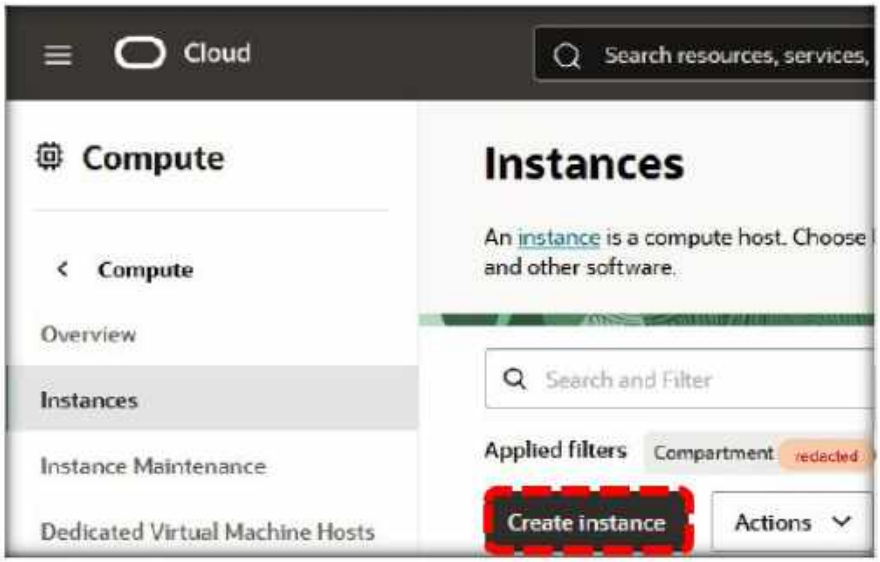
   a. Log in to the OCI console.

   b. Click the main (top-left corner) menu: 

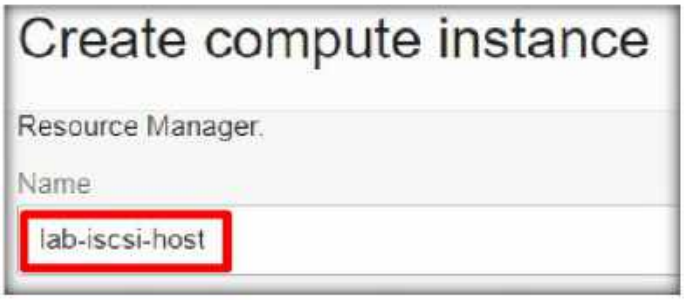
2. Navigate to **Compute → Instances**.



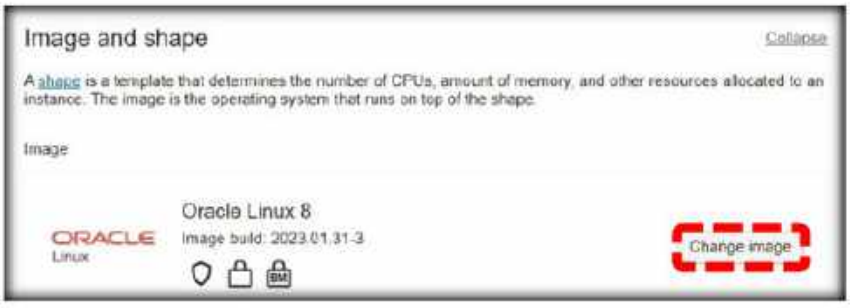
3. Click the **Create instance** button.



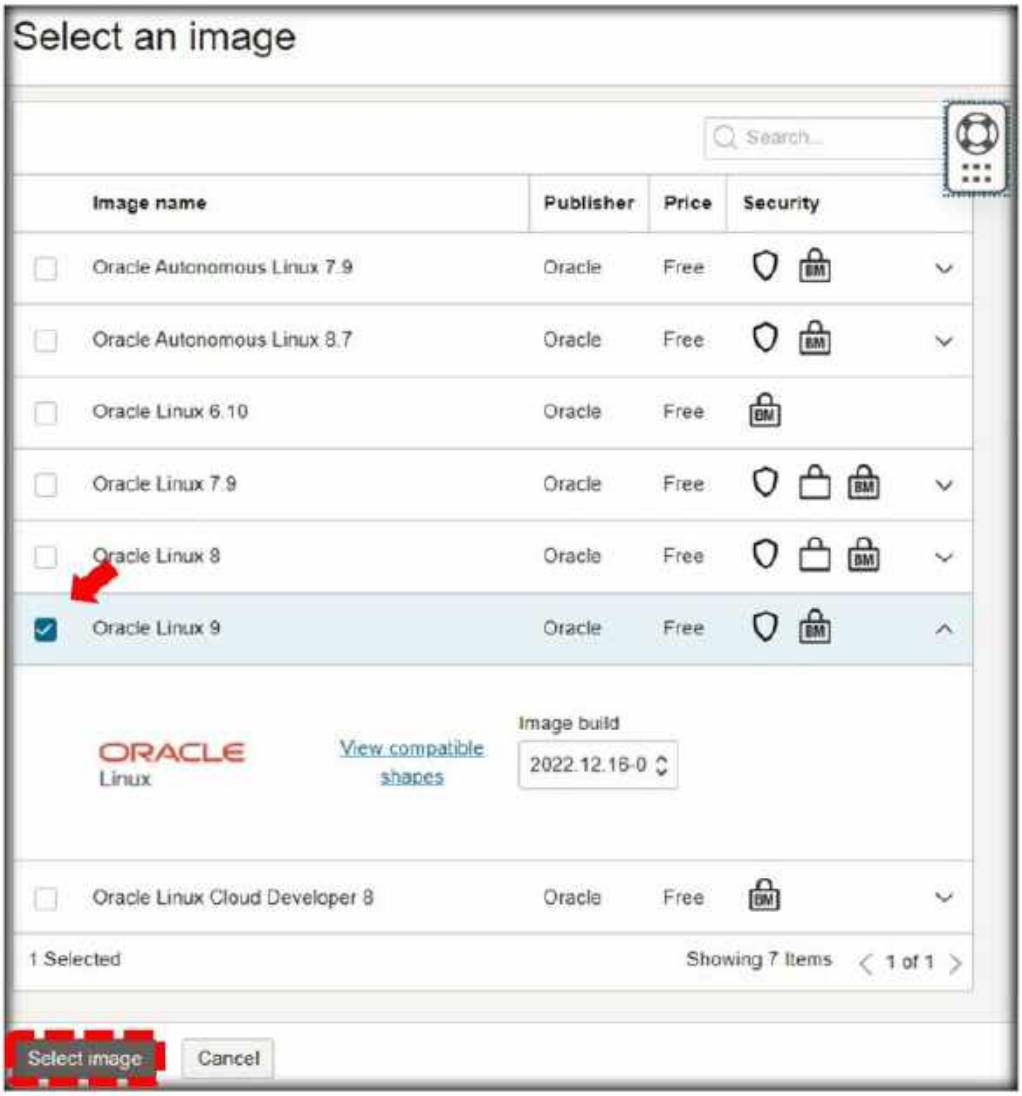
a. Name the new instance **lab-iscsi-host** and ensure that the **Create in compartment** input is set to the same compartment where the **lab-node1** instance resides. Select a **different Availability Domain**. For example, if **lab-node1** was created in **AD-1**, the new instance should be assigned to **AD-3**. (Instance **lab-node2** uses **AD-2**.)



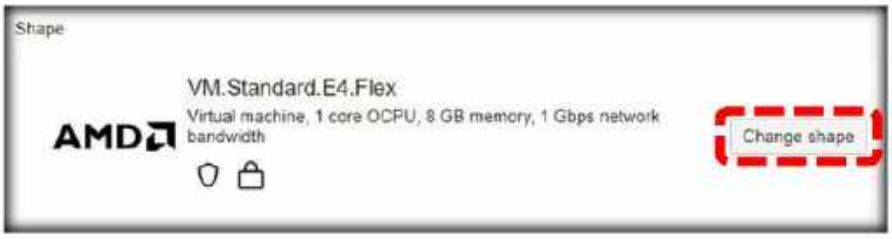
b. Select the compute image. If the image you need does not appear, click **Change Image** button and select the desired image. Skip the next step if the desired image is already shown.



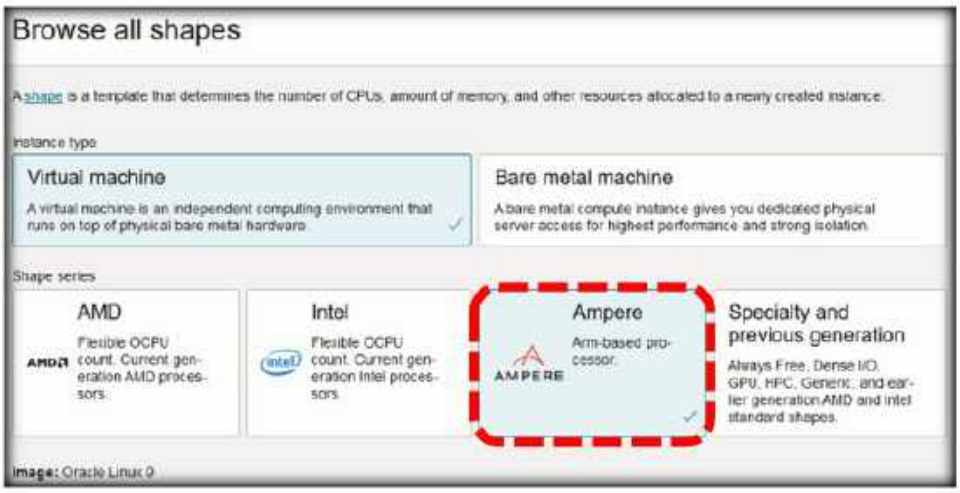
c. Select the check box next to **Oracle Linux 9**, verify the latest Image Build date, and click the **Select image** button below.



d. Pick the compute shape to assign to the new instance. Please be mindful that not all shapes may be permitted by the tenancy or compartment resource quotas. To select a different shape, click the **Change shape** button.

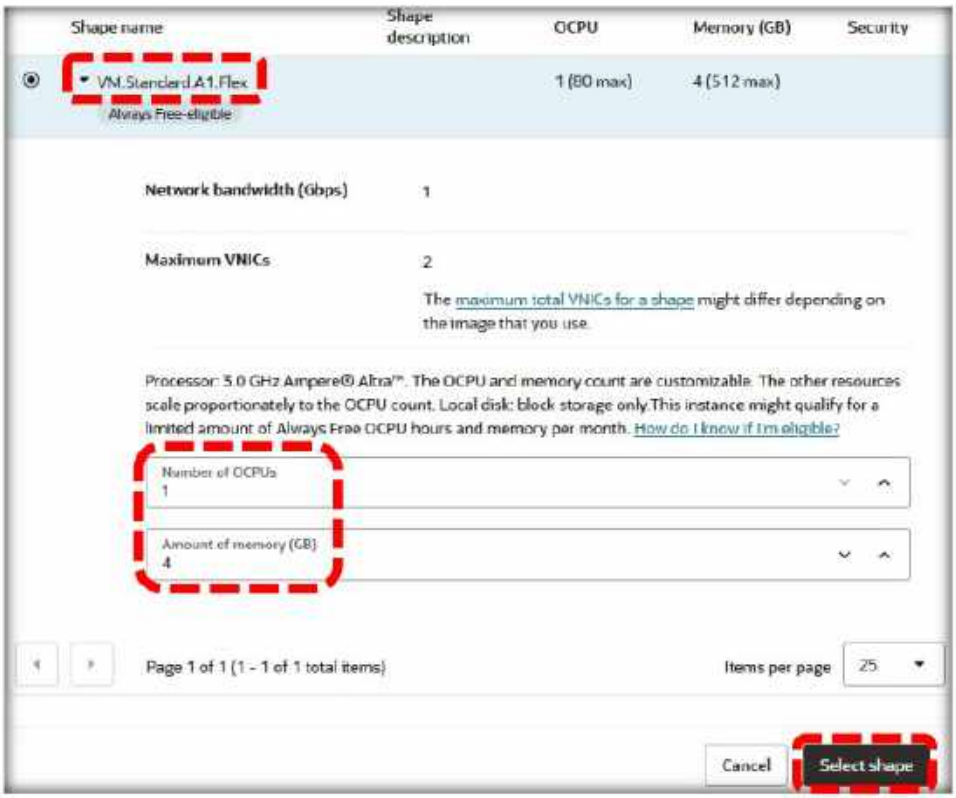


e. Select the hardware type to locate the desired shape. In this course, we use the **(Ampere) VM.Standard.A1.Flex** shape. Click the **Ampere** tile to see compatible choices.



f. Click the small arrow next to the **VM.Standard.A1.Flex** shape to adjust VM resources.

Keep **1 OCPU**, reduce memory from default to **4 GB**, then click the **Select shape** button.



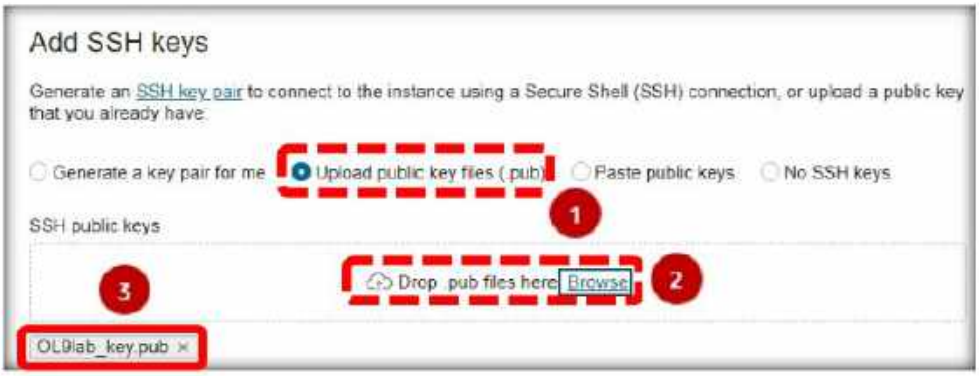
In the newer OCI Console, please click the **Next** button (twice) to continue.

g. Review the Primary VNIC information section.
   1) Under Primary network, select Select existing virtual cloud network.
   2) Ensure **VCN in <*your compartment*>** has the VCN selected.
   3) Under Subnet, select Select existing subnet.
   4) Ensure **Subnet in <*your compartment*>** has the public regional subnet selected.

h. Review the Primary VNIC IP address section.
   1) Under **Private IPv4 address**, select **Automatically assign private IPv4 address**.
   2) Under **Public IPv4 address**, select **Automatically assign public IPv4 address**.
   3) You will be using a public IP address for this instance.

i. Add SSH keys to create the instance.
   1) Enable **Upload public key files (.pub)**.
   2) Click the **Browse** hyperlink, navigate to your key file location, and use the key you generated before.

3) After the key has been uploaded, your screen should resemble the following image:



Newer OCI Console: click the **Next** button, skip to the next task.

j. Keep the default settings in the **Boot volume** and **Block volumes** sections. Click **Next**.

4. Click the **Create** button and wait for the creation process to complete.

The Running instance displays with the green tile. Note the **Availability Domain** value

Example below shows an arbitrary value of AD-2, in your case, it should be **AD-3.**

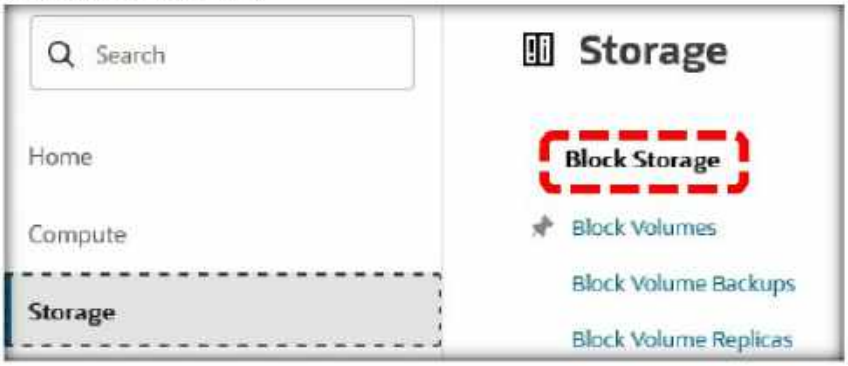
5. Copy the instance's public IP address to your scratch text document.

6. In the **Work requests** section, notice the operation completion state.

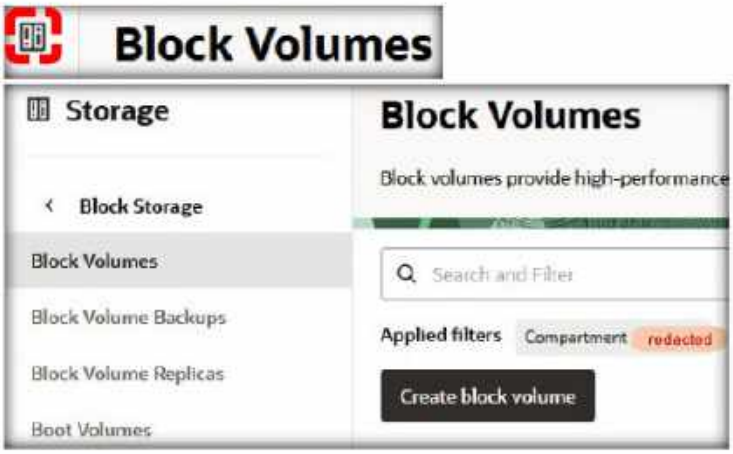
7. Create a new block volume.

   a. Click the main (top-left corner) menu: 

   b. Click **Block Volumes**.



   c. Click **Block Volumes** on the left (if not already selected). Click the Storage icon to expand left-hand panel.



d. Click the **Create Block Volume** button.

e. Name the new volume **lab-iscsi-vol1**.

Important Note: Ensure the compartment and Availability Domain are the very same ones where your instance was located.

f. Set the Availability Domain value to the AD of the **lab-iscsi-host** instance. Example below shows an arbitrary value of AD-2, in your case, it should be **AD-3.**

g. Set the **Volume Size and Performance** using the **Custom** option. The default size of 1 TB is not needed for our practices and we set a much smaller size.

1) For **Volume Size**, enter **50**.

2) Leave the **Target Volume Performance** at the lowest default **10** VPU setting.

3) Click the Create Block Volume button. 

4) Wait for the volume creation to complete.

5) The label next to the Block Volume name turns green (Available) when the creation process has completed.

8. Return to the **lab-iscsi-host** instance.

   a. Click the main (top-left corner) menu: 

   b. Navigate to **Compute → Instances.**

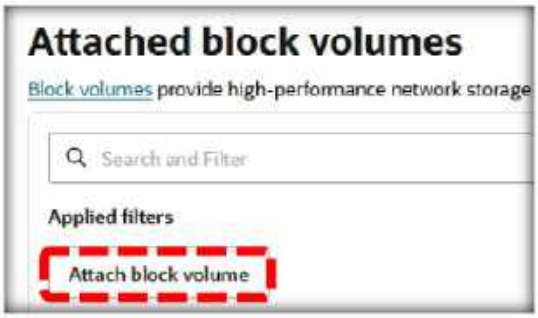
   c. Verify (select, if not shown) your compartment next to the **Applied filters** on the left.

   d. Click the hyperlink for the **lab-iscsi-host** instance. Then click the **Storage** tab.

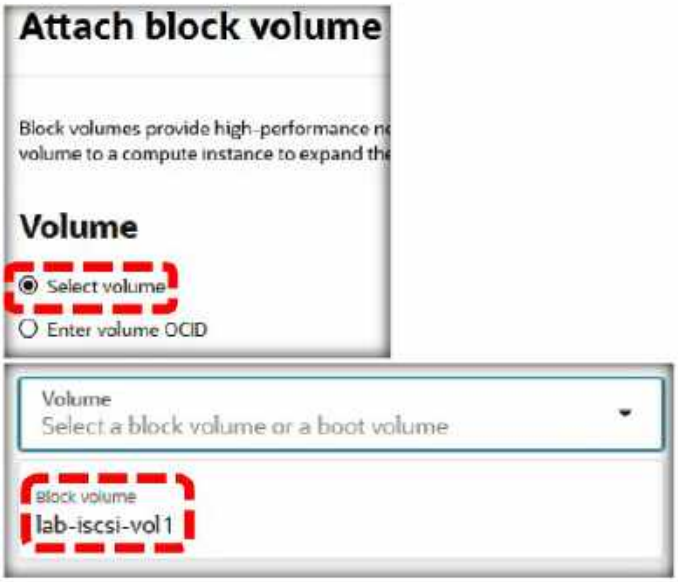
9. Attach the new volume to the instance.

   a. Scroll down to the **Attached block volumes** section.

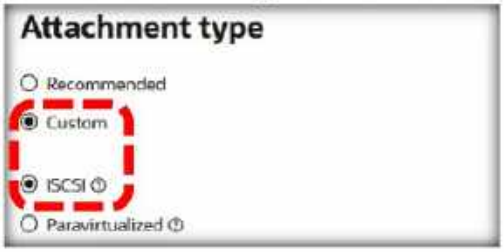
   b. Click the **Attach block volume** button.



c. Set the radio button **Select volume**, then select the volume from the drop-down list on the right.

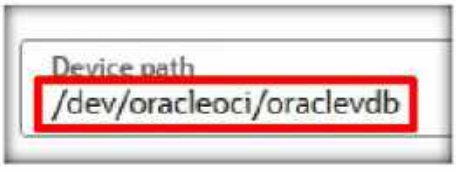


d. In the **Attachment type** section, set **Custom** and **ISCSI** radio buttons.



e. Leave deselected the two check boxes below (Require CHAP credentials and Use Oracle Cloud Agent …).

f. Select the device path. Example: `/dev/oracleoci/oraclevdb`



g. Use the default **Read/write** access option.

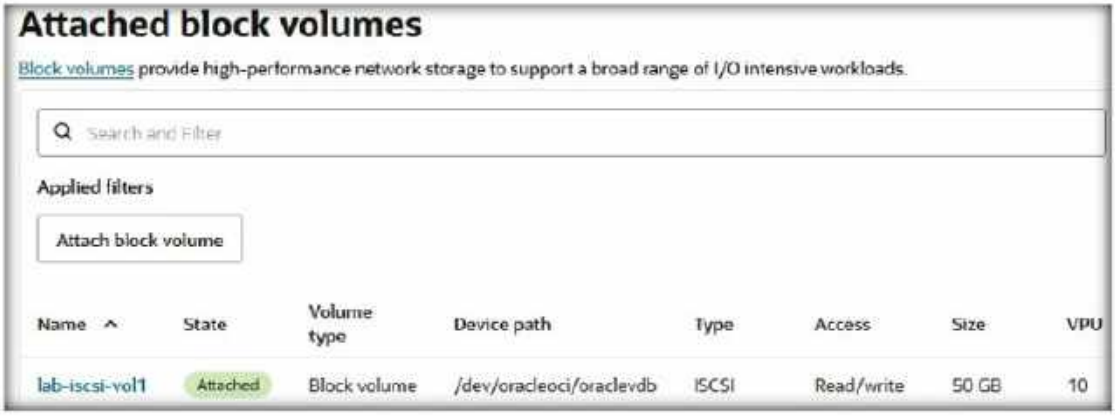
h. Click the **Attach** button,

i. The informational note is displayed.

j. Optional: Click the hyperlink on the dialog box to view related documentation.

k. Click **Close** to continue.

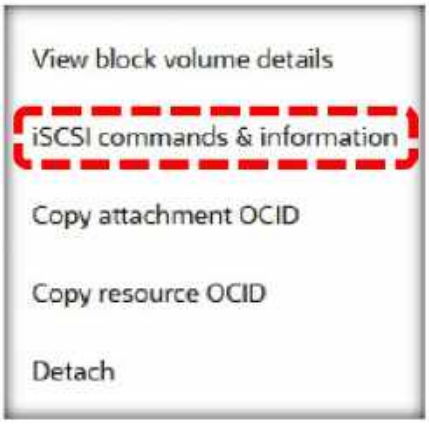
l. The volume is now attached to the instance.



10. Examine iSCSI commands for the volume (un)mounting.

a. Click the Actions ... menu on the right.

b. Select the **iSCSI commands & information** option.



c. View the presented commands. Click **Copy** below the **Connect** box.

d. Paste iSCSI connect and disconnect commands into your scratch text document for future use. **Note:** There are three Connect commands, but onlly two Disconnect commands. Keep these groups of commands separate in the scratch text document.

Note: When copying from the scratch text document beware of the automatic line breaks (wrapping) that may have been added by your text editor.

e. Click the **Cancel** button to continue.

11. Run iSCSI connect commands on the instance.

a. Connect to the instance using SSH. Confirm that you intend to connect, if prompted.

1) Open a command-prompt window on your local PC.

2) Change the directory to the location of the SSH authentication key (for example, `D:\OL9labs`).

3) Start an SSH connection to the public IP address of the **lab-iscsi-host** instance.

```
ssh -i OL9lab_key opc@<lab-iscsi-host_public-IP>
```

```
D:\OL9Labs>ssh -i OL9lab_key opc@150.136.155.230
The authenticity of host '150.136.155.230 (150.136.155.230)' can't be established.

ECDSA key fingerprint is SHA256:i8JYSuOc7YJ0vPe40MjiMO2Nkiq+25bGiD8VXsIQHfI.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '150.136.155.230' (ECDSA) to the list of known hosts.
```

Note: Your public IP will be different.

b. Copy the iSCSI commands from the scratch text document and paste these three commands into the Linux command-prompt window to connect the new volume to the instance. (You can paste all three as a block or one by one individually.)

```
[opc@lab-iscsi-host ~]$ sudo iscsiadm -m node -o new -T iqn.2015-12.com.oracleiaas
:4ddf73b5-3457-45cd-b441-89e9c22d4ccc -p 169.254.2.2:3260
New iSCSI node [tcp:[hw=,ip=,net_if=,iscsi_if=default] 169.254.2.2,3260,-1 iqn.201
5-12.com.oracleiaas:4ddf73b5-3457-45cd-b441-89e9c22d4ccc] added
[opc@lab-iscsi-host ~]$ sudo iscsiadm -m node -o update -T iqn.2015-12.com.oraclei
aas:4ddf73b5-3457-45cd-b441-89e9c22d4ccc -n node.startup -v automatic
[opc@lab-iscsi-host ~]$ sudo iscsiadm -m node -T iqn.2015-12.com.oracleiaas:4ddf73
b5-3457-45cd-b441-89e9c22d4ccc -p 169.254.2.2:3260 -l
Logging in to [iface: default, target: iqn.2015-12.com.oracleiaas:4ddf73b5-3457-45
cd-b441-89e9c22d4ccc, portal: 169.254.2.2,3260]
Login to [iface: default, target: iqn.2015-12.com.oracleiaas:4ddf73b5-3457-45cd-b4
41-89e9c22d4ccc, portal: 169.254.2.2,3260] successful.
```

Note: The device **IQN** and the **IP addresses** shown are ephemeral and would be different in your environment.

12. Enable the **ocid** service. Continue in the same Linux terminal environment.

    a. Use the command **sudo systemctl** to enable and start the **ocid** service.

```
sudo systemctl enable --now ocid
```

```
[opc@lab-iscsi-host ~]$ sudo systemctl enable --now ocid
Created symlink /etc/systemd/system/multi-user.target.wants/ocid.service → /etc/sy
stemd/system/ocid.service.
```

    b. Check the **ocid** service status:

```
systemctl status ocid
```

```
[opc@lab-iscsi-host ~]$ systemctl status ocid
● ocid.service - Oracle Cloud Infrastructure utilities daemon
     Loaded: loaded (/etc/systemd/system/ocid.service; enabled; vendor preset: en
     Active: active (running) since Fri 2023-03-10 18:15:10 GMT; 1min 14s ago
```

13. Use the **oci-iscsi-config** utility. (This utility is already installed in Oracle Linux 9 instances.)

    a. Run the following commands:

```
sudo oci-iscsi-config  show
sudo oci-iscsi-config  show --details --no-truncate
sudo oci-iscsi-config  show --output-mode text
```

```
[opc@lab-iscsi-host ~]$ sudo oci-iscsi-config show
Currently attached iSCSI devices:
  Volume Name   | Attached Device |  Size  |
----------------------------------------
 lab-iscsi-... |       sdb       |  50G   |
```

Note: Volume name may or may not be shown.

```
[opc@lab-iscsi-host ~]$ sudo oci-iscsi-config show --details --no-truncate
Currently attached iSCSI devices:
                          Target                                  | Volume Name   |
                          Volume OCID                                             | Persistent
 Portal |  Current Portal  | Session State | Attached Device | Size | Mountpoint | Filesy
stem |
--------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------
-----
 iqn.2015-12.com.oracleiaas:4ddf73b5-3457-45cd-b441-89e9c22d4ccc | lab-iscsi-vol1 | ocid1
.volume.oc1.iad.abuwcljt47aic4wmwihmal2j2mufi7vqsgzujqn6gnt22opwlp3h3vs4nbdq |  169.254.2
.2:3260 | 169.254.2.2:3260 |   LOGGED_IN   |       sdb       | 50G |     -      |     -
```

```
[opc@lab-iscsi-host ~]$ sudo oci-iscsi-config show --output-mode text
Currently attached iSCSI devices:
Volume Name: lab-iscsi-vol1
Attached Device: sdb
Size: 50G
Mountpoint: -
Filesystem: -

Partitions:
```

14. Create a new partition on the attached volume.

a. Run the following command to list the known disks.

```
sudo fdisk -l
```

```
[opc@lab-iscsi-host ~]$ sudo fdisk -l
Disk /dev/sda: 46.58 GiB, 50010783744 bytes, 97677312 sectors
Disk model: BlockVolume
   ...
Disk /dev/sdb: 50 GiB, 53687091200 bytes, 104857600 sectors
Disk model: BlockVolume
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 1048576 bytes
```

b. Identify the new volume. It will most likely appear as **/dev/sdb**.

c. Run the **fdisk** command again with the new disk device as its parameter. The utility will automatically create a new partition table for you.

Enter the following command:

```
sudo fdisk /dev/sdb
```

```
[opc@lab-iscsi-host ~]$ sudo fdisk /dev/sdb

Welcome to fdisk (util-linux 2.37.4).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table.
Created a new DOS disklabel with disk identifier 0xb377b3ac.

Command (m for help):
```

d. At the **fdisk Command:** prompt, enter lowercase **p** to print the partition table. There should be no partitions yet.

e. At the **fdisk Command:** prompt, enter lowercase **n** to create a new partition.

Note: Please view the following screen capture (may appear on the next page).

1) Use a default **(p)** for the primary partition.
2) This is going to be partition number one; keep the default.
3) Keep the default **2048** for the first sector.
4) This is only used to illustrate, and the 10 GB size is sufficient. For the size, enter "**+10G**"

5) Write the changes. At the **fdisk Command:** prompt, enter lowercase **w**.

```
[opc@lab-iscsi-host ~]$ sudo fdisk /dev/sdb

Welcome to fdisk (util-linux 2.37.4).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table.
Created a new DOS disklabel with disk identifier 0xb377b3ac.

Command (m for help): p
Disk /dev/sdb: 50 GiB, 53687091200 bytes, 104857600 sectors
Disk model: BlockVolume
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 1048576 bytes
Disklabel type: dos
Disk identifier: 0xb377b3ac

Command (m for help): n
Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)
Select (default p): 1

Using default response p.
Partition number (1-4, default 1): 2
First sector (2048-104857599, default 2048): 3                                  4
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-104857599, default 104857599): +10G

Created a new partition 1 of type 'Linux' and of size 10 GiB.

Command (m for help): w 5
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.
```

15. Format the new volume.

a. Run the command **sudo fdisk -l** (dash L) to list the known disks inclusive of the new 10 GB partition you created in the previous step.

```
sudo fdisk -l
```

```
[opc@lab-iscsi-host ~]$ sudo fdisk -l
Disk /dev/sda: 46.58 GiB, 50010783744 bytes, 97677312 sectors
Disk model: BlockVolume
   ...
Disk /dev/sdb: 50 GiB, 53687091200 bytes, 104857600 sectors
Disk model: BlockVolume
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 1048576 bytes
Disklabel type: dos
Disk identifier: 0xb377b3ac

Device     Boot Start      End  Sectors Size Id Type
/dev/sdb1        2048 20973567 20971520  10G 83 Linux
```

b. Create the EXT4 file system on the added disk.

```
sudo mkfs -t ext4 /dev/sdb1
```

```
[opc@lab-iscsi-host ~]$ sudo mkfs -t ext4 /dev/sdb1
mke2fs 1.46.5 (30-Dec-2021)
Discarding device blocks: done
Creating filesystem with 2621440 4k blocks and 655360 inodes
Filesystem UUID: 5a2854ea-4506-4f43-9369-2b8e2d0f6cbc
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done
```

c. Create a mount point to mount the disk where **/mnt/mydisk** directory is arbitrarily chosen for this example.

```
sudo mkdir /mnt/mydisk
```

```
[opc@lab-iscsi-host ~]$ sudo mkdir /mnt/mydisk
```

d. Mount the new disk. The disk has to be mounted with the **_netdev** option.

```
sudo mount /dev/sdb1 -o _netdev /mnt/mydisk
```

```
[opc@lab-iscsi-host ~]$ sudo mount /dev/sdb1 -o _netdev /mnt/mydisk
```

e. Change permissions on the new directory to be able to write to it.

```
sudo chmod 777 /mnt/mydisk
ls -la /mnt/mydisk
```

Now, any user can write to this directory.

```
[opc@lab-iscsi-host ~]$ sudo chmod 777 /mnt/mydisk
[opc@lab-iscsi-host ~]$ ls -la /mnt/mydisk
total 20
drwxrwxrwx. 3 root root  4096 Mar 10 18:39 .
drwxr-xr-x. 3 root root    20 Mar 10 18:40 ..
drwx------. 2 root root 16384 Mar 10 18:39 lost+found
```

f. Add a new file to the mounted disk. You may use a text editor to create a new file of arbitrary content in the **/mnt/mydisk** directory. Alternatively, use the **fallocate** utility.

```
fallocate -l 4G /mnt/mydisk/four-GB-file.dat
ls -l /mnt/mydisk/
```

```
[opc@lab-iscsi-host ~]$ fallocate -l 4G /mnt/mydisk/four-GB-file.dat
[opc@lab-iscsi-host ~]$ ls -l /mnt/mydisk/
total 4194324
-rw-r--r--. 1 opc  opc  4294967296 Mar 10 18:42 four-GB-file.dat
drwx------. 2 root root      16384 Mar 10 18:39 lost+found
```

16. View the iSCSI block volume details from the OCI console.

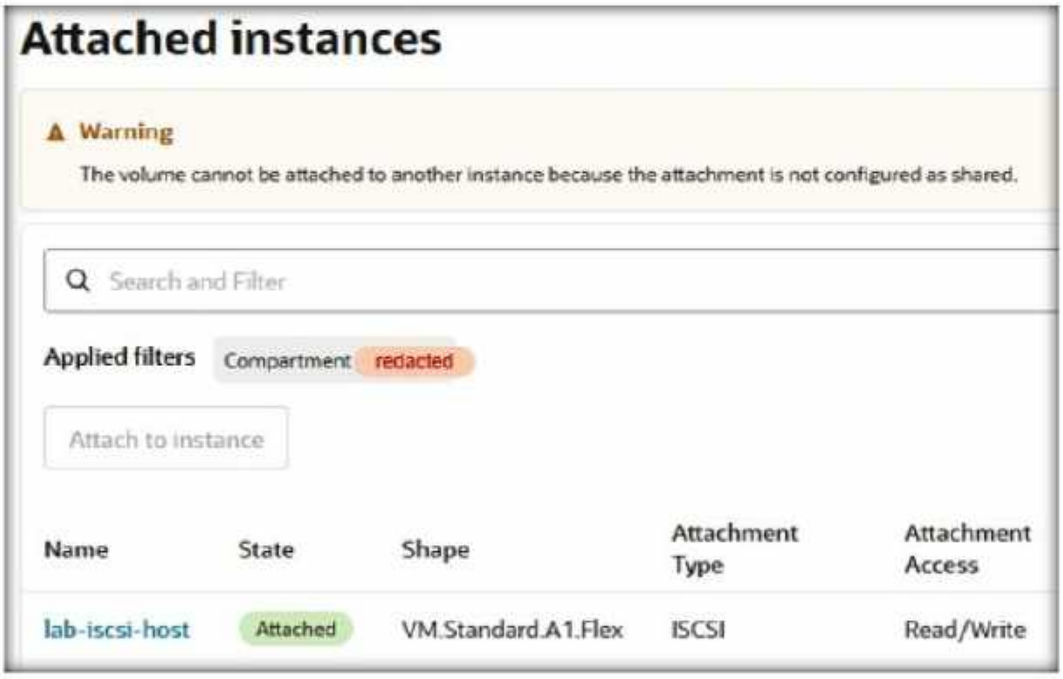
a. Return to the **OCI console → Instances**.

b. Click the hyperlink for the **lab-iscsi-host** instance, then click the **Storage** tab.

c. Scroll down to the **Attached block volumes**.

d. Click the **lab-iscsi-vol1** hyperlink and view (click) the volume **Metrics** tab.
Some activity should be observed for both read and write Throughput and Operations graphs.

e. Click **Attached Instances** tab.



17. Prepare OCI iSCSI utilities to be able to modify iSCSI storage.

Note: Previously used **oci-iscsi-config show** did not perform any modifications to the iSCSI storage. It did not require additional setup and did not need the API signing key. To be able to run modifying OCI iSCSI utilities, additional setup is needed. Information required includes the OCID of the user, the OCID of the tenancy, and the hosting region name. If you have previously recorded the above values (in your scratch text document), you are ready to run the setup.

a. Install OCI utilities by running the following command (this package may already be installed):

```
sudo dnf install -y oci-utils
```

```
[opc@lab-iscsi-host ~]$ sudo dnf install -y oci-utils
```

b. Install the python OCI packages by running the following command:

```
sudo dnf install -y python39-oci-sdk python39-oci-cli
```

```
[opc@lab-iscsi-host ~]$ sudo dnf install -y python39-oci-sdk python39-oci-cli
```

c. Execute the configuration setup.

Note: The API signing key is being generated for the `root` user in this case.

```
sudo oci setup config
```

```
[opc@lab-iscsi-host ~]$ sudo oci setup config
```

1) Accept the default location for the configuration file.
2) Paste the user's OCID.
3) Paste the OCID of the tenancy.
4) Enter the "home" region number.
5) Agree: Enter `Y` to generate the API signing key.
6) Accept the default location to store the API signing key (`/root/.oci`).
7) Enter a new name for the API key: **`oci_api_iscsi_key`**.
8) A passphrase is not required. Specify **N/A** and press **Enter** for the empty passphrase value. Repeat **N/A** for confirmation.

```
Enter a location for your config [/root/.oci/config]: 1
Enter a user OCID: ocid1.user.oc1..a 2 OCID redacted to protect privacy
7xtxgq
Enter a tenancy OCID: ocid1.tenancy.oc1..aaaaaa 3 OCID redacted to protect privacy
7w4hw6eio5uq
Enter a region by index or name(e.g.
1: af-johannesburg-1, 2: ap-chiyoda-1, 3: ap-chuncheon-1, 4: ap-dcc-canberra-1, 5: ap-hyd
erabad-1,
6: ap-ibaraki-1, 7: ap-melbourne-1, 8: ap-mumbai-1, 9: ap-osaka-1, 10: ap-seoul-1,
11: ap-singapore-1, 12: ap-sydney-1, 13: ap-tokyo-1, 14: ca-montreal-1, 15: ca-toronto-1,
16: eu-amsterdam-1, 17: eu-dcc-dublin-1, 18: eu-dcc-dublin-2, 19: eu-dcc-milan-1, 20: eu-
dcc-milan-2,
21: eu-dcc-rating-1, 22: eu-dcc-rating-2, 23: eu-frankfurt-1, 24: eu-madrid-1, 25: eu-mar
seille-1,
26: eu-milan-1, 27: eu-paris-1, 28: eu-stockholm-1, 29: eu-zurich-1, 30: il-jerusalem-1,
31: me-abudhabi-1, 32: me-dcc-muscat-1, 33: me-dubai-1, 34: me-jeddah-1, 35: mx-queretaro
-1,
36: sa-santiago-1, 37: sa-saopaulo-1, 38: sa-vinhedo-1, 39: uk-cardiff-1, 40: uk-gov-card
iff-1,
41: uk-gov-london-1, 42: uk-london-1, 43: us-ashburn-1, 44: us-chicago-1, 45: us-gov-ashb
urn-1,
46: us-gov-chicago-1, 47: us-gov-phoenix-1, 48: us-langley-1, 49: us-luke-1, 50: us-phoen
ix-1,
51: us-sanjose-1): 43 4
Do you want to generate a new API Signing RSA key pair? (If you decline you will be asked
 to supply the path to an existing key.) [Y/n]: Y 5
Enter a directory for your keys to be created [/root/.oci]: 6
Enter a name for your key [oci_api_key]: oci_api_iscsi_key 7
Public key written to: /root/.oci/oci_api_iscsi_key_public.pem
Enter a passphrase for your private key (empty for no passphrase): 8
Private key written to: /root/.oci/oci_api_iscsi_key.pem
Fingerprint: 7a:7e:f3:25:52:cb:e7:88:ce:83:cb:dd:c1:9d:4f:b6
Config written to /root/.oci/config
```

d. Copy the API public key to the `/tmp` directory to make it available to the `opc` user.

```
sudo cp /root/.oci/oci_api_iscsi_key_public.pem /tmp
```

e. Make the key readable to the **opc** user. Execute

```
sudo chmod 666 /tmp/oci_api_iscsi_key_public.pem
```

f. Transfer the public key file to your local PC.

1) Open a new command-prompt window on your local PC. Change the directory to where your authentication private key - **OL9lab_key** - is stored (for example, **D:\OL9labs**)

2) Run a secure copy command. Note that there is a white space and a period following the "pem" file extension (note the space and the period characters):

Note: Your instance's public IP address will be different.

```
scp -i OL9lab_key
opc@@<lab-iscsi-host-IP>:/tmp/oci_api_iscsi_key_public.pem .
```

```
D:\OL9Labs>scp -i OL9lab_key opc@150.136.155.230:/tmp/oci_api_iscsi_key_public.pem .
oci_api_iscsi_key_public.pem                  100%  451    16.3KB/s   00:00
```

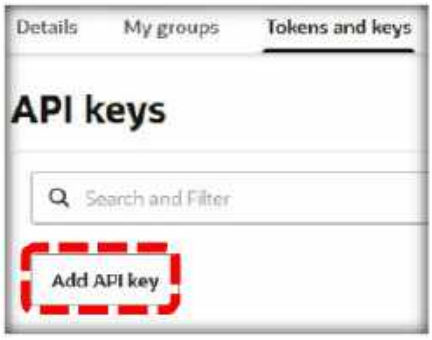
g. Upload the public key to the OCI by using the OCI console.

1) In the OCI console, click the user profile icon  in the top-right corner.

2) Click the **User settings** menu option.

3) Select **Tokens and keys** tab.

4) Click the **Add API key** button under **API keys**.



Note: OCI user is allowed up to three API signing keys. There might be already an API key present. If you see more than one, locate and delete the oldest API signing key, which may have existed from before you have begun using this OCI account.

5) Select the **Choose Public Key File** option button. Then, browse and select the `oci_api_iscsi_key_public.pem` file from your local PC location.

6) Click the **Add** button.

7) The pop-up shows the **Configuration File Preview**. You may use the **Copy** link to copy and paste the contents to the local PC clipboard. Click **Close** when copied.

8) Note the **TODO** section in the displayed configuration.

   Paste the contents into your local scratch text file to have it handy. Usually, you will not need to copy this content as the OCI configurtion file had been created by the **oci setup config** for you.

9) If not logged in with SSH to the **lab-iscsi-host**, please log back in. Using the command prompt of **lab-iscsi-host** instance, run the command **`sudo cat /root/.oci/config`** to view the generated configuration file.

```
sudo cat /root/.oci/config
```

10) Compare its contents with the text recorded in the scratch text file.

11) If there are any differences, fix those using **`sudo vi /root/.oci/config`** and save. In most cases, the content will be the same. Path in the **`key_file`** property should be correct on the **lab-iscsi-host**, do not modify that value.

```
sudo vi /root/.oci/config
```

12) Exit the **vi** editor (Press the **ESC** key, colon ":" key, followed by lowercase **q**). Use **`wq`**, if you have made any changes.

18. You can now make use of the OCI iSCSI utilities to manage iSCSI storage.

    a. To create a new iSCSI volume of 50 GB size and have it attached to the instance run the command:

```
sudo oci-iscsi-config create -S 50 \
   --volume name=new-block-vol50g --attach-volume
```

The output should resemble the following:

```
[opc@lab-iscsi-host ~]$ sudo oci-iscsi-config create -S 50 \
>    --volume name=new-block-vol50g --attach-volume
Creating a new 50 GB volume name=new-block-vol50g
Volume [name=new-block-vol50g] created
Attaching the volume to this instance
Attaching iSCSI device.
Volume [name=new-block-vol50g] is attached.
```

b. Use **sudo fdisk -l** to see the attached volumes.

```
sudo fdisk -l
```

The output will resemble:

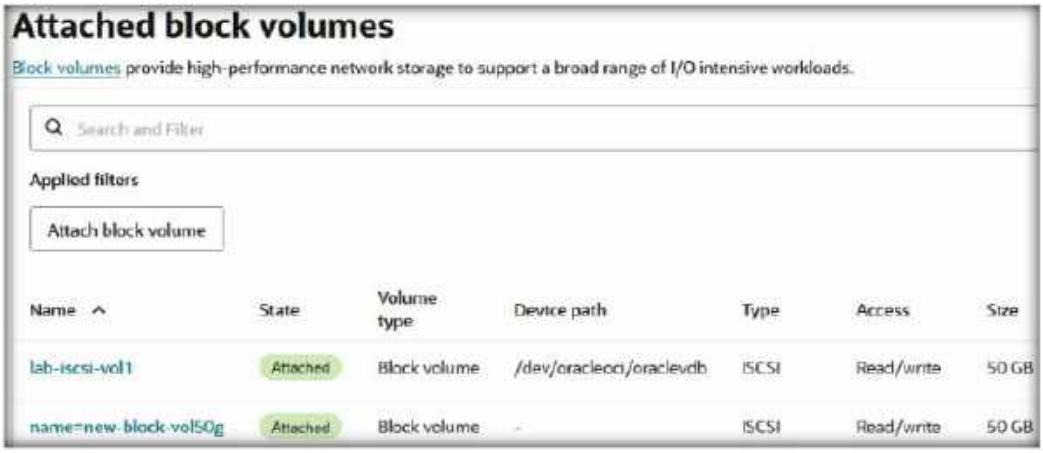
```
[opc@lab-iscsi-host ~]$ sudo fdisk -l
Disk /dev/sda: 46.58 GiB, 50010783744 bytes, 97677312 sectors
Disk model: BlockVolume
   ...

Device     Boot Start      End  Sectors Size Id Type
/dev/sdb1        2048 20973567 20971520  10G 83 Linux


Disk /dev/sdc: 50 GiB, 53687091200 bytes, 104857600 sectors
Disk model: BlockVolume
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 1048576 bytes
```

Note: Disks **/dev/sdb** and **/dev/sdb1** are from the previous steps in this practice. The new disk added by the OCI iSCSI tool appears as **/dev/sdc**.

c. Optionally, return to the OCI console and view the new block volume attached to the **lab-iscsi-host** instance. Refer to **Task 16** of this practice for details.



19. Detach the iSCSI volume using **oci-iscsi-config.**

Tip: (Optional) If you are logged in to the OCI portal and observe the **Attached block volumes** section, you may see the effects of the below commands reflected in real-time.

a. To view the volumes' IQN values, run the following command:

```
sudo oci-iscsi-config show --details --no-truncate --output-mode
text
```

These IQN values are needed when detaching a volume. (Partial output is shown.)

```
Target: iqn.2015-12.com.oracleiaas:a5491878-ca0e-49f4-b11a-92ae8d8ad6a2
Volume Name: name=new-block-vol50g
Volume OCID: ocid1.volume.oc1.iad.abuwcljt535iumndsn7bp4hnmxojmk4voxagp26krk
a
Persistent Portal: 169.254.2.3:3260
Current Portal: 169.254.2.3:3260
Session State: LOGGED_IN
Attached Device: sdc
Size: 50G
Mountpoint: -
Filesystem: -
```

b. Locate the IQN of the volume we recently created **(name=new-block-vol50g)**.

It is best to copy and save the IQN and the OCID of the volume for later use in this practice.

Note: The volume is not currently mounted.

c. Detach the volume.

The keyword "detach" is followed by the dash and uppercase "**I**" character.

```
sudo oci-iscsi-config detach -I <your volume IQN value>
```

Partial output is shown.

```
[opc@lab-iscsi-host ~]$ sudo oci-iscsi-config detach -I iqn.2015-12.com.oracleiaas:a5-
a0e-49f4-b11a-92ae8d8ad6a2
Detaching volume name=new-block-vol50g [iqn.2015-12.com.oracleiaas:a5491878-ca0e-49f4-
ae8d8ad6a2]
Volume [iqn.2015-12.com.oracleiaas:a5491878-ca0e-49f4-b11a-92ae8d8ad6a2] is detached.
```

Note: If the volume is mounted, you must unmount it first and/or use the force option. Data loss may result in the latter case.

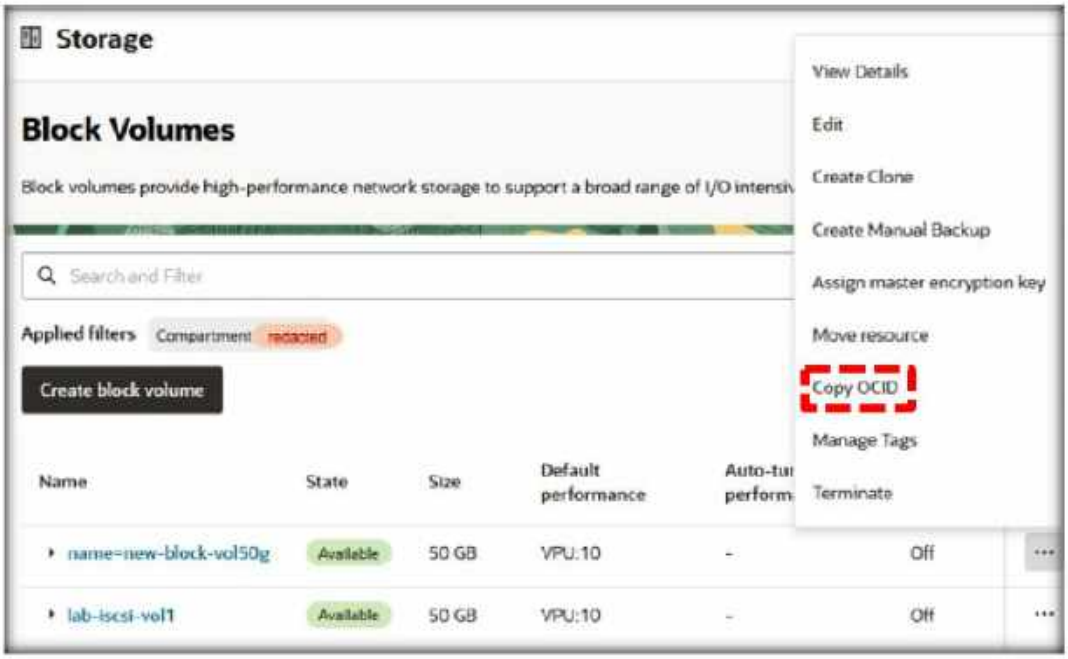
d. View attached iSCSI volumes.

```
sudo oci-iscsi-config show
```

e. Optionally, look at the instance in the OCI console for the attached volumes. See **Task 16** of this practice for details.

20. The detached volume can be destroyed. This is an irrevocable operation and any data on the destroyed volume will also disappear for good.

a. To destroy an iSCSI volume, the command uses the volume's OCID. (Get the volume's OCID from the **OCI console →Storage → Block Volumes** if you have not recorded it before. Alternatively, look in the terminal buffer by scrolling up. You may find the output depending on the terminal buffer settings.)



b. Destroy the volume with the following command:

```
sudo oci-iscsi-config destroy -O <volume's OCID>
```

```
[opc@lab-iscsi-host ~]$ sudo oci-iscsi-config destroy -O ocid1.volume.oc1.iad.abuwcljt535iumn
dsn7bp4hnmxojmk4voxagp26krkdvjxc6adu4s7xt7r7a
Volume : [ocid1.volume.oc1.iad.abuwcljt535iumndsn7bp4hnmxojmk4voxagp26krkdvjxc6adu4s7xt7r7a]
WARNING: the volume(s) will be destroyed.  This is irreversible.  Continue?  (y/N) y

Volume [ocid1.volume.oc1.iad.abuwcljt535iumndsn7bp4hnmxojmk4voxagp26krkdvjxc6adu4s7xt7r7a] is
 destroyed
```

Confirm the operation when prompted.

c. Optional: If you viewed the block volumes in the OCI console, go back to the Block Volumes page and see the new volume showing in Terminated state.

Name	State	Size	Default performance
name=new-block-vol50g	Terminated	50 GB	VPU:10
lab-iscsi-vol1	Available	50 GB	VPU:10

d. Note that the volume will show as **Terminated** for a while. Eventually, the record of the volume will disappear from the **Block Volumes** in your compartment.

Solution 6-1: Using OCI Utilities to Manage iSCSI Storage

Overview

There is no automated solution for this practice.

Practice 6-2: Configuring an iSCSI Target and an Initiator

Overview

In this practice, you configure:

- An iSCSI Target
- An iSCSI Initiator

Assumptions

- Practices for Lesson 3 have been completed.
- Practice 6-1 has been completed.
- Compute instance **lab-node1** that you created earlier is running and has a public IP address.
- Compute instance **lab-iscsi-host** that you created in the previous practice is running and has a public IP address.
- The private key to authenticate over SSH is stored in the file named **`OL9lab_key`**.
- Block volume **lab-iscsi-vol1** is mounted.

Tasks

1. Connect to the **lab-iscsi-host** instance using SSH.

   a. Open a command-prompt window on your local PC.

   b. Change the directory to the location of the SSH authentication key (for example, **`D:\OL9labs`**).

   c. Start an SSH connection to the public IP address of the **`lab-iscsi-host`** instance.

   ```
   ssh -i OL9lab_key opc@<lab-iscsi-host_public-IP>
   ```

   ```
   D:\OL9Labs>ssh -i OL9lab_key opc@150.136.155.230
   ```

   Note: Your public IP will be different.

2. List the existing disk devices.

   a. Run the following command:

```
ls -l /dev/disk/by-path
```

```
[opc@lab-iscsi-host ~]$ ls -l /dev/disk/by-path
total 0
lrwxrwxrwx. 1 root root  9 Mar 10 18:30 ip-169.254.2.2:3260-iscsi-iqn.2015-12.com.oracleiaas:
4ddf73b5-3457-45cd-b441-89e9c22d4ccc-lun-2 -> ../../sdb
lrwxrwxrwx. 1 root root 10 Mar 10 18:39 ip-169.254.2.2:3260-iscsi-iqn.2015-12.com.oracleiaas:
4ddf73b5-3457-45cd-b441-89e9c22d4ccc-lun-2-part1 -> ../../sdb1
lrwxrwxrwx. 1 root root  9 Mar 10 16:20 pci-0000:00:04.0-scsi-0:0:0:1 -> ../../sda
lrwxrwxrwx. 1 root root 10 Mar 10 16:20 pci-0000:00:04.0-scsi-0:0:0:1-part1 -> ../../sda1
lrwxrwxrwx. 1 root root 10 Mar 10 16:20 pci-0000:00:04.0-scsi-0:0:0:1-part2 -> ../../sda2
lrwxrwxrwx. 1 root root 10 Mar 10 16:20 pci-0000:00:04.0-scsi-0:0:0:1-part3 -> ../../sda3
```

   When the iSCSI volume was attached, it used to be the **Read/Write** attachment. For the iSCSI target, the attachment must be a shareable one. The disk needs to be unmounted, detached, and then reattached as a shareable iSCSI disk.

   b. Unmount **/dev/sdb1**.

```
sudo umount /dev/sdb1
```

```
[opc@lab-iscsi-host ~]$ sudo umount /dev/sdb1
```

3. Return to the **lab-iscsi-host** instance in the OCI console.

   a. Click the main (top-left corner) menu: 

   b. Navigate to **Compute → Instances.**

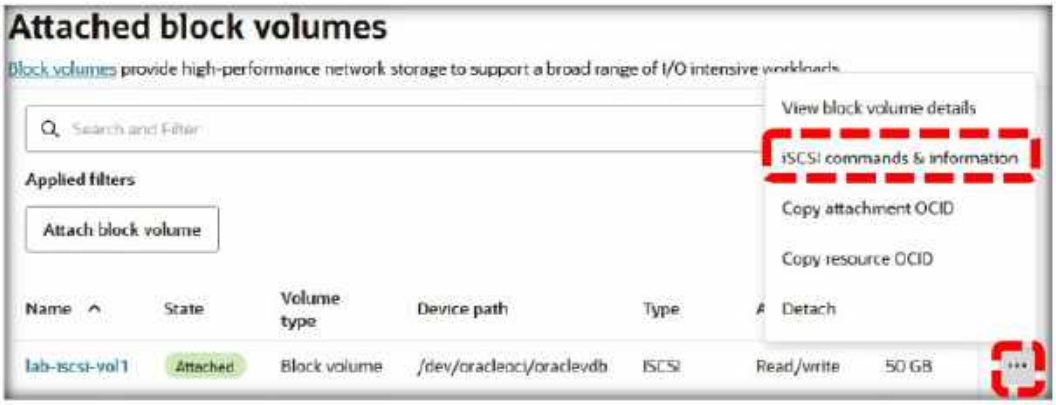
   c. Verify (select, if not shown) your compartment next to the **Applied filters** on the left.

   d. Click the hyperlink for the **lab-iscsi-host** instance. Then click the **Storage** tab.

4. Look up and copy the iSCSI admin commands from the OPC console to disconnect the volume.

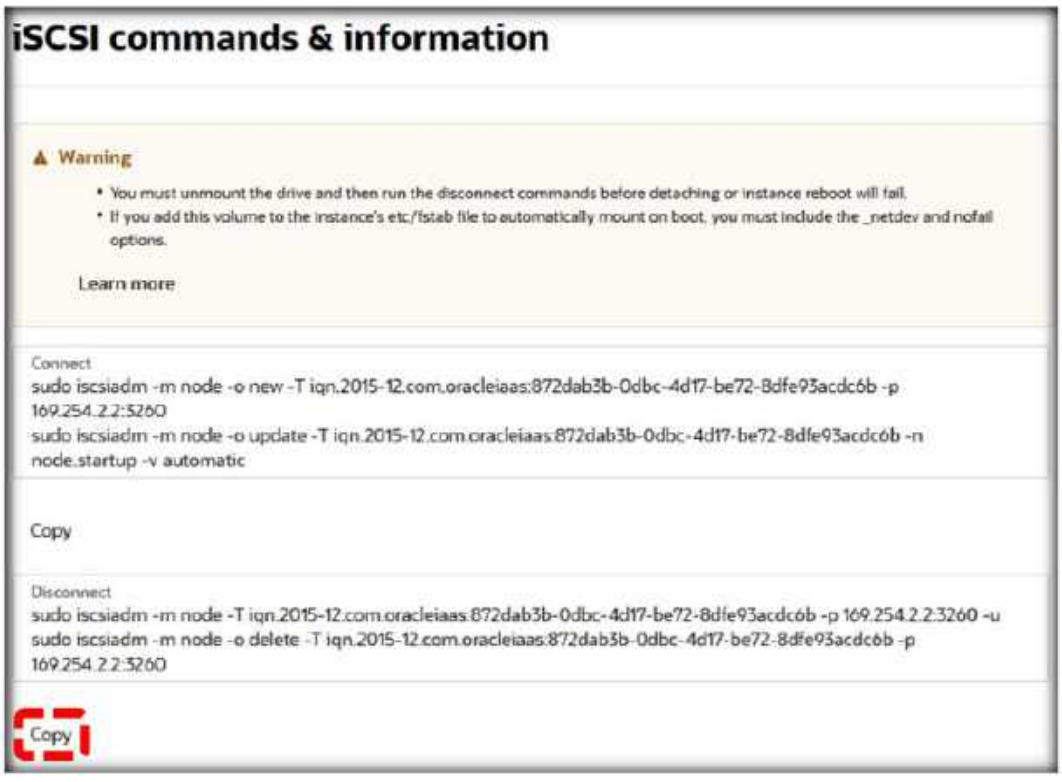
   a. Scroll down to the **Attached block volumes**.

b. From the **Actions** menu, select the **iSCSI commands & information** option.



c. View the presented commands. Click **Copy** to the right of the **Disconnect** box.

You may reuse the iSCSI commands from your scratch text document, if you had copied both Connect and Disconnect commands before and saved them.



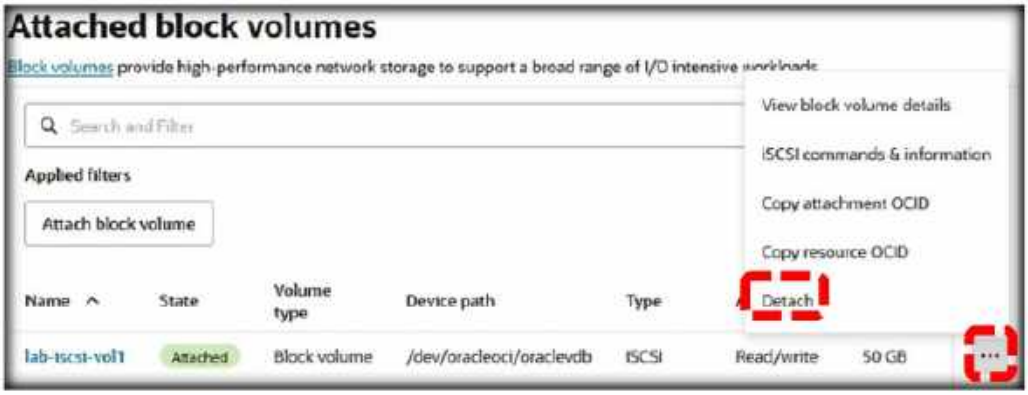
d. Paste the iSCSI disconnect commands into your scratch text document for future use.

e. Run the iSCSI disconnect commands on the instance. Copy the iSCSI commands from the scratch text document and paste these two commands into the Linux command-prompt window to connect the new volume to the instance. (You can paste both as a block, or one by one individually.)

```
[opc@lab-iscsi-host ~]$ sudo iscsiadm -m node -T iqn.2015-12.com.oracleiaas:4ddf73b5-
d-b441-89e9c22d4ccc -p 169.254.2.2:3260 -u
Logging out of session [sid: 1, target: iqn.2015-12.com.oracleiaas:4ddf73b5-3457-45cd
e9c22d4ccc, portal: 169.254.2.2,3260]
Logout of [sid: 1, target: iqn.2015-12.com.oracleiaas:4ddf73b5-3457-45cd-b441-89e9c22
ortal: 169.254.2.2,3260] successful.
[opc@lab-iscsi-host ~]$ sudo iscsiadm -m node -o delete -T iqn.2015-12.com.oracleiaas
5-3457-45cd-b441-89e9c22d4ccc -p 169.254.2.2:3260
```

Note: The device **IQN** and the **IP addresses** shown are ephemeral and would be different in your environment.

f. Detach the volume from the instance. The OCI console panel for this block volume should currently be open. Return to it and click **Detach** from the **Actions** menu.

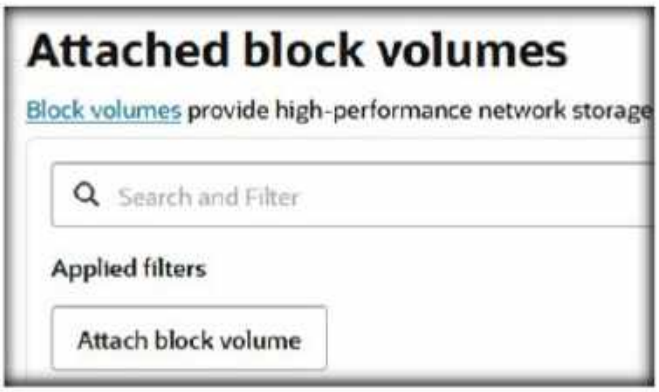


g. The confirmation dialog box with instructions is displayed. The commands to log off and disconnect were already run. Click the **Detach volume** button.

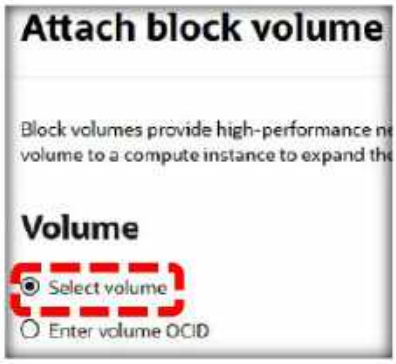
Note: The volume will be reattached, so there is no need to remove the `/etc/fstab` entry in this case, whether it had or had not been added.

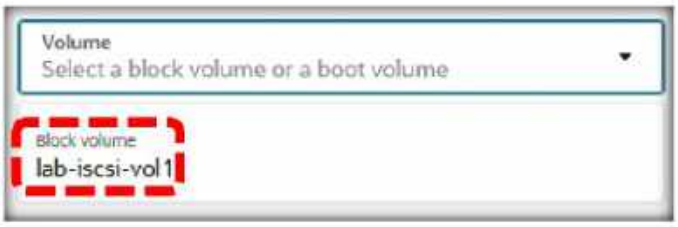
Wait for the detach operation to complete. The volume will disappear from the list.

h. Click the **Attach block volume** button.

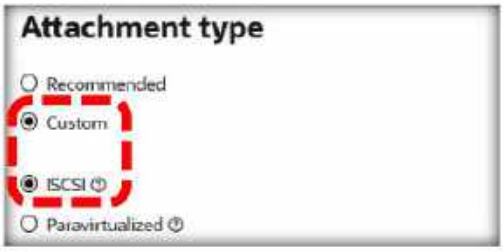


i. Select the volume from the drop-down list on the right.



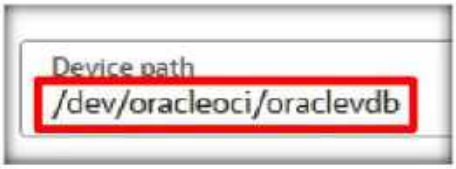


j. In the **Attachment type** section, set **Custom** and **ISCSI** radio buttons.



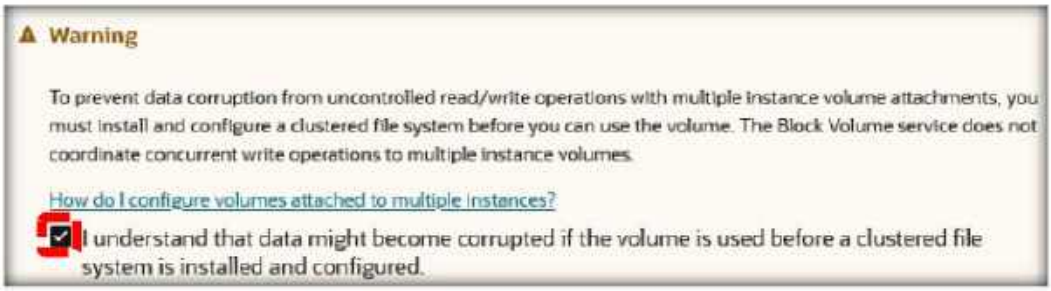
k. Leave deselected the two check boxes below (Require CHAP credentials and Use Oracle Cloud Agent ...).

l. Select the device path. Example: `/dev/oracleoci/oraclevdb`



m. Select the **Read/write - shareable** access option.

n. Agree your understanding of the data potential corruption before the sharing complete configuration.



o. Click the **Attach** button,

p. The informational note is displayed. Read the notice text. Note the mounting option **_netdev** and close the dialog box. Wait for the process to complete.

q. Click **Close** to continue.

5. Continue in the Linux terminal.

a. Ensure a firewall rule exists to allow access on port **3260** for **TCP** traffic.

1) Execute the following command:

```
sudo firewall-cmd --zone=public --permanent --add-port=3260/tcp
```

```
[opc@lab-iscsi-host ~]$ sudo firewall-cmd --zone=public --permanent \
> --add-port=3260/tcp
Warning: ALREADY_ENABLED: 3260:tcp
success
```

Note: Ignore the warning if the port had been previously added.

2) Refresh the firewall settings. Execute the following command:

```
sudo  firewall-cmd  --reload
```

b. Connect the iSCSI volume.

Repeat the connect commands. If you stored **iscsiadm** commands in the scratch text document, copy the text and paste it into the Linux terminal. The commands are identical because we are using the very same block volume. If you have not stored the command text, refer to step 4-b to obtain the connect command syntax. It takes a few moments to connect the device.

```
[opc@lab-iscsi-host ~]$ sudo iscsiadm -m node -o new -T iqn.2015-12.com.oracleiaas
:4ddf73b5-3457-45cd-b441-89e9c22d4ccc -p 169.254.2.2:3260
New iSCSI node [tcp:[hw=,ip=,net_if=,iscsi_if=default] 169.254.2.2,3260,-1 iqn.201
5-12.com.oracleiaas:4ddf73b5-3457-45cd-b441-89e9c22d4ccc] added
[opc@lab-iscsi-host ~]$ sudo iscsiadm -m node -o update -T iqn.2015-12.com.oraclei
aas:4ddf73b5-3457-45cd-b441-89e9c22d4ccc -n node.startup -v automatic
[opc@lab-iscsi-host ~]$ sudo iscsiadm -m node -T iqn.2015-12.com.oracleiaas:4ddf73
b5-3457-45cd-b441-89e9c22d4ccc -p 169.254.2.2:3260 -l
Logging in to [iface: default, target: iqn.2015-12.com.oracleiaas:4ddf73b5-3457-45
cd-b441-89e9c22d4ccc, portal: 169.254.2.2,3260]
Login to [iface: default, target: iqn.2015-12.com.oracleiaas:4ddf73b5-3457-45cd-b4
41-89e9c22d4ccc, portal: 169.254.2.2,3260] successful.
```

c. Mount the disk to the **/mnt/mydisk** mount point. The disk has to be mounted with the **_netdev** option. (Recall: There was the **/dev/sdb1** partition with a data file. If unsure, you may always repeat the **ls -l /dev/disk/by-path** command to verify.)

Enter the following command:

```
sudo mount /dev/sdb1 -o _netdev /mnt/mydisk
```

```
[opc@lab-iscsi-host ~]$ sudo mount /dev/sdb1 -o _netdev /mnt/mydisk
```

6. Ensure the **ocid** service is running. If it is already active, there is no harm in repeating the command.

```
sudo systemctl enable --now ocid
```

```
[opc@lab-iscsi-host ~]$ sudo systemctl enable --now ocid
```

7. View iSCSI device details.

```
sudo oci-iscsi-config -s
```

Partial output is shown.

```
[opc@lab-iscsi-host ~]$ sudo oci-iscsi-config -s
Currently attached iSCSI devices:
Target: iqn.2015-12.com.oracleiaas:4ddf73b5-3457-45cd-b441-89e9c22d4ccc
Volume Name: lab-iscsi-vol1
Volume OCID: ocid1.volume.oc1.iad.abuwcljt47aic4wmwihmal2j2mufi7vqsgzujqn6g
```

Note: Copy and record the volume's IQN into your scratch text document for future use.

8. Install the **targetcli** utilities.

```
sudo dnf install -y targetcli
```

```
[opc@lab-iscsi-host ~]$ sudo dnf install -y targetcli
```

9. Ensure **target** service is active.

   a. Start the **target** service.

   ```
   sudo systemctl enable target && sudo systemctl start target
   ```

   b. Verify the **target** service is active:

   ```
   sudo systemctl status target
   ```

   c. Unmount the disk now:

   ```
   sudo umount /dev/sdb1
   ```

10. This set of steps will be conducted in the **targetcli** tool interface. Run the **targetcli** tool to configure the iSCSI target properties:

    ```
    sudo targetcli
    ```

    ```
    [opc@lab-iscsi-host ~]$ sudo targetcli
    Warning: Could not load preferences file /root/.targetcli/prefs.bin.
    targetcli shell version 2.1.53
    Copyright 2011-2013 by Datera, Inc and others.
    For help on commands, type 'help'.

    />
    ```

    a. Change the location to **/backstores/block**.

    ```
    cd /backstores/block
    ```

    ```
    /> cd /backstores/block
    ```

    b. Use the volume's device name - **/dev/sdb1** - to create a new storage object.

    ```
    create name=target-disk dev=/dev/sdb1
    ```

    (The name **target-disk** is arbitrary.)

    ```
    /backstores/block> create name=target-disk dev=/dev/sdb1
    Created block storage object target-disk using /dev/sdb1.
    ```

    c. Change the location to **/iscsi**. Enter the following command:

    ```
    cd /iscsi
    ```

Create a new IQN object—iSCSI qualified name.

```
create
```

```
/backstores/block> cd /iscsi
/iscsi> create
Created target iqn.2003-01.org.linux-iscsi.lab-iscsi-host.x8664:sn.9cc4c31fca17.
Created TPG 1.
Global pref auto_add_default_portal=true
Created default portal listening on all IPs (0.0.0.0), port 3260.
```

Copy the target IQN value and store it in your scratch text document for future use.

Note: TPG stands for the Target Portal Group.

d. Change the location to the above IQN object. Enter the following command:

```
cd <iSCSI-IQN-value>   ### cd iqn<TAB-Key>
```

```
/iscsi> cd iqn.2003-01.org.linux-iscsi.lab-iscsi-host.x8664:sn.9cc4c31fca17/
/iscsi/iqn.20....9cc4c31fca17>
```

Tip: You don't usually have to type the whole value; simply enter "cd iqn" and press the TAB key to autocomplete. This may not work if you have multiple IQN values.

e. Change the location to **tpg1/acls**. Do not use the forward slash in front of the **tpg**; this is a relative path. Enter the following command:

```
cd  tpg1/acls
```

```
/iscsi/iqn.20....9cc4c31fca17> cd tpg1/acls
/iscsi/iqn.20...a17/tpg1/acls>
```

f. Copy the iSCSI IQN value from above. Record the IQN value (if not done before) in the scratch text file for future use. Create the node ACL (Access Control List). Paste the IQN value from the clipboard to complete the **create** command parameter. Enter the following command:

```
create <iSCSI-IQN-value>
```

A confirmation appears about the node ACL creation.

```
/iscsi/iqn.20...a17/tpg1/acls> create iqn.2003-01.org.linux-iscsi.lab-iscsi-host.x8664:sn
4c31fca17
Created Node ACL for iqn.2003-01.org.linux-iscsi.lab-iscsi-host.x8664:sn.9cc4c31fca17
```

g. Navigate to the parent location using double-period notation.

1) Enter the command `cd  ..` (there are two period characters after the white space).

```
cd  ..
```

The location is set to **tpg1**.

2) Set the **authentication** attribute to zero. Enter the following command:

```
set attribute authentication=0
```

3) Set the **generate_node_acls** attribute to one. Enter the following command:

```
set attribute generate_node_acls=1
```

```
/iscsi/iqn.20...a17/tpg1/acls> cd ..
/iscsi/iqn.20...c31fca17/tpg1> set attribute authentication=0
Parameter authentication is now '0'.
/iscsi/iqn.20...c31fca17/tpg1> set attribute generate_node_acls=1
Parameter generate_node_acls is now '1'.
```

h. Navigate to the **luns** location.

1) Enter the command **cd luns** (relative path).

```
cd luns
```

2) Create the **luns** entry.

The name should be the same as in **Step 10b**.

```
create /backstores/block/target-disk
```

LUN creation is confirmed.

3) Change the location to the root.

```
cd /
```

```
/iscsi/iqn.20...c31fca17/tpg1> cd luns
/iscsi/iqn.20...a17/tpg1/luns> create /backstores/block/target-disk
Created LUN 0.
Created LUN 0->0 mapping in node ACL iqn.2003-01.org.linux-iscsi.lab-iscsi-h
c31fca17
/iscsi/iqn.20...a17/tpg1/luns> cd /
/>
```

i. Use the **ls** command to view the details. Verify against the directions above. Your device IQN will be different from the example below. Take note of the hierarchy of details.

```
ls
```

```
/> ls
o- / ..... [...]
  o- backstores ..... [...]
  | o- block ..... [Storage Objects: 1]
  | | o- target-disk ..... [/dev/sdb1 (10.0GiB) write-thru activated]
  | |   o- alua ..... [ALUA Groups: 1]
  | |     o- default_tg_pt_gp ..... [ALUA state: Active/optimized]
  | o- fileio ..... [Storage Objects: 0]
  | o- pscsi ..... [Storage Objects: 0]
  | o- ramdisk ..... [Storage Objects: 0]
  o- iscsi ..... [Targets: 1]
  | o- iqn.2003-01.org.linux-iscsi.lab-iscsi-host.aarch64:sn.a6e67f1dd121 ..... [TPGs: 1]
  |   o- tpg1 ..... [gen-acls, no-auth]
  |     o- acls ..... [ACLs: 1]
  |     | o- iqn.2003-01.org.linux-iscsi.lab-iscsi-host.aarch64:sn.a6e67f1dd121 .. [Mapped LUNs: 1]
  |     |   o- mapped_lun0 ..... [lun0 block/target-disk (rw)]
  |     o- luns ..... [LUNs: 1]
  |     | o- lun0 ..... [block/target-disk (/dev/sdb1) (default_tg_pt_gp)]
  |     o- portals ..... [Portals: 1]
  |       o- 0.0.0.0:3260 ..... [OK]
  o- loopback ..... [Targets: 0]
  o- vhost ..... [Targets: 0]
```

j. Save the configuration.

```
saveconfig
```

```
/> saveconfig
Configuration saved to /etc/target/saveconfig.json
```

k. Exit the `targetcli` session.

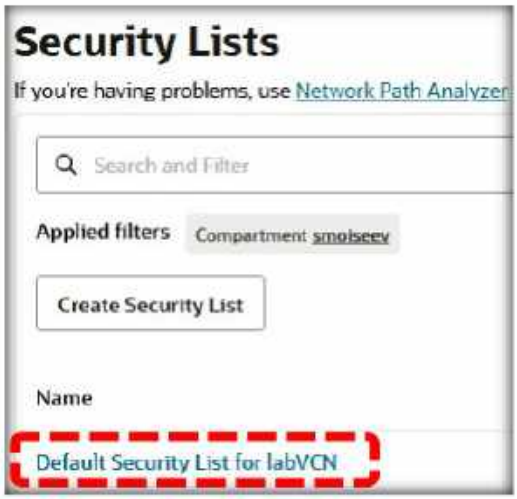
```
exit
```

```
/> exit
Global pref auto_save_on_exit=true
Last 10 configs saved in /etc/target/backup/.
Configuration saved to /etc/target/saveconfig.json
```

11. The iSCSI target has been configured. Exit the SSH session with **lab-iscsi-host** by either pressing **CTRL-D** or entering the **exit** command at the Linux prompt.
12. OCI traffic filtering needs to allow port 3260 access on your compartment's VCN. Return to the OCI console. Click **Networking → Virtual Cloud Networks**.

13. Verify your compartment (select it, if another is shown) and then click the **labVCN** hyperlink.

14. On the **labVCN** page, click the **Security** tab. Under **Security Lists**, click the **Default Security List for labVCN** hyperlink.



15. Click the **Security rules** tab.
16. Check for any existing rules for TCP access to target port 3620. The rule may already exist. If not, add a new Ingress rule:

    a. Click the **Add Ingress Rules** button.

    b. Complete the presented form:

        1) Leave the **Stateless** slider disabled (default).
        2) Add the **Source CIDR**. The value **10.0.0.0/16** is the least restrictive in your compartment.
        3) Set the protocol to **TCP**.
        4) Leave the **Source Port Range** with the default empty (All) value.
        5) Set the **Destination Port Range** to **3260**.
        6) Enter a meaningful description.

7) Click the **Add Ingress Rules** button when finished.

17. Connect to the **lab-node1** instance using SSH. Reuse the local command-prompt window from before. Execute the following local command:

```
ssh -i OL9lab_key opc@<public IP of lab-node1>
```

```
D:\OL9Labs>ssh -i OL9lab_key opc@150.136.71.103
Last login: Fri Mar 10 21:48:21 2023 from 99.228.228.74
[opc@lab-node1 ~]$
```

Note: The public IP address of your **lab-node1** would be different from the example above.

18. Install the iSCSI initiator utilities. There is no harm in performing this task if the utilities have been already installed. At the Linux command prompt, execute the following command:

```
sudo dnf install -y iscsi-initiator-utils
```

```
[opc@lab-node1 ~]$ sudo dnf install -y iscsi-initiator-utils
```

19. Enable and start the iSCSI service - **iscsid**.

    a. Execute the following command:

```
sudo systemctl enable iscsid && sudo systemctl start iscsid
```

```
[opc@lab-node1 ~]$ sudo systemctl enable iscsid && sudo systemctl start iscsid
Created symlink /etc/systemd/system/multi-user.target.wants/iscsid.service → /usr/lib/sys
temd/system/iscsid.service.
```

b. Check the status of the iSCSI service.

```
systemctl status iscsid
```

```
[opc@lab-node1 ~]$ sudo systemctl status iscsid
● iscsid.service - Open-iSCSI
     Loaded: loaded (/usr/lib/systemd/system/iscsid.service; enabled; vendor preset: disa
bled)     Active: active (running) since Fri 2023-03-10 21:50:48 GMT; 2min 50s ago
```

20. Edit the iSCSI initiator's configuration file **/etc/iscsi/initiatorname.iscsi**

a. Use the vi editor.

```
sudo vi /etc/iscsi/initiatorname.iscsi
```

b. `[opc@lab-node1 ~]$ sudo vi /etc/iscsi/initiatorname.iscsi`

c. Replace the volume's IQN in the configuration file with the value that you had saved from the iSCSI target IQN.

Tip: You may have the desired value saved in the step 10.c.

```
InitiatorName=iqn.1988-12.com.oracle:83dcf2704ccb
~
~
~
~
~
~
~
~
~
~
~
"/etc/iscsi/initiatorname.iscsi" 1L, 50B
```

1) In the **vi** editor, while in "browse" mode, press the forward slash character and type **iqn.**

```
/iqn
```

Then press **Enter** to locate the IQN entry.

```
InitiatorName=iqn.1988-12.com.oracle:83dcf2704ccb
~
~
~
~
~
~
~
~
~
~
/iqn
```

2) Hold the **Shift** key and press the letter **C** key (**Shift-c**). The editor wipes off the text from the cursor position to the end of the line and enters "INSERT" mode.

```
InitiatorName=
~
~
~
~
~
~
~
~
~
~
-- INSERT --
```

3) While in "insert" mode, paste the IQN value from the iSCSI target.

```
InitiatorName=iqn.2003-01.org.linux-iscsi.lab-iscsi-host.x8664:sn.9cc4c
~
~
~
~
~
~
~
~
~
~
:wq
```

4) Switch to "browse" mode by pressing the **ESC** key. Then press the colon (**:**), **w**, and **q** keys in sequence to write the changes (**:wq**).

Hint: If you have not copied the IQN of the iSCSI target device, open another local command-prompt window and connect over SSH to the **lab-iscsi-host** instance. When connected, run **`sudo targetcli`** and use the **`ls`** command, and the device's IQN will appear in two places. Copy it. Exit the **`targetcli`** and close that SSH connection to the **lab-iscsi-host**.

21. Perform the iSCSI Target discovery. This task is performed on the **lab-node1** VM.

For the iSCSI target host IP address, use the private IP in the OCI (look up from the **OCI console → Instances**). This is both faster and more secure as compared to the public IP of the **lab-iscsi-host**.

Name	State	Public IP	Private IP
lab-iscsi-host	Running	132.226.66.137	10.0.0.09

Note: Additional configuration will be necessary if you need to access the iSCSI target from outside OCI. This will require the Ingress rules definitions for the public network. In your environment, all IP addresses will be different.

a. Execute the following command:

```
sudo iscsiadm -m discovery -t st -p <iSCSI_target_private_IP>
```

```
[opc@lab-node1 ~]$ sudo iscsiadm -m discovery -t st -p 10.0.0.9
10.0.0.9:3260,1 iqn.2003-01.org.linux-iscsi.lab-iscsi-host.x8664:sn.9cc4
```

b. Update iSCSI DB. Execute the following command:

```
sudo iscsiadm -m discoverydb -t st -p <iSCSI_target_private_IP>
```

```
[opc@lab-node1 ~]$ sudo iscsiadm -m discoverydb -t st -p 10.0.0.9
# BEGIN RECORD 6.2.1.4
discovery.startup = manual
discovery.type = sendtargets
discovery.sendtargets.address = 10.0.0.9
discovery.sendtargets.port = 3260
discovery.sendtargets.auth.authmethod = None
discovery.sendtargets.auth.username = <empty>
discovery.sendtargets.auth.password = <empty>
discovery.sendtargets.auth.username_in = <empty>
discovery.sendtargets.auth.password_in = <empty>
node.session.auth.chap_algs = MD5
discovery.sendtargets.timeo.login_timeout = 15
discovery.sendtargets.use_discoveryd = No
discovery.sendtargets.discoveryd_poll_inval = 30
discovery.sendtargets.reopen_max = 5
discovery.sendtargets.timeo.auth_timeout = 45
discovery.sendtargets.timeo.active_timeout = 30
discovery.sendtargets.iscsi.MaxRecvDataSegmentLength = 32768
# END RECORD
```

c. Restart the **iscsid** service. Run the following command:

```
sudo systemctl restart iscsid
```

```
[opc@lab-node1 ~]$ sudo systemctl restart iscsid
```

22. Log in to the iSCSI session. Use background login by adding **&** at the end of the command. Execute the following command (on **lab-node1**):

```
sudo iscsiadm -m node -l &
```

You will get the prompt back. Press the **Enter** key if you are not immediately seeing the prompt. (Example output below was truncated. Your output may not contain Logging in ... messages.)

```
sudo iscsiadm -m node -l &
[1] 258208
[opc@lab-node1 ~]$ Logging in to [iface: default, target: iqn.2015-02.oracle.boot:ue
fi, portal: 169.254.0.2,3260]
Logging in to [iface: default, target: iqn.2003-01.org.linux-iscsi.lab-iscsi-host.x8
664:sn.9cc4c31fca17, portal: 10.0.0.9,3260]
```

23. Verify the device availability. Run the following command:

```
ls -l /dev/disk/by-path
```

Note the device name - similar to **/dev/sdb**, or **/dev/sdc**, and so on. (Partial output is shown.)

```
[opc@lab-node1 ~]$ ls -l /dev/disk/by-path
total 0
lrwxrwxrwx. 1 root root  9 Mar 10 22:11 ip-10.0.0.9:3260-iscsi-iqn.2003-01.org.linux
-iscsi.lab-iscsi-host.x8664:sn.9cc4c31fca17-lun-0 -> ../../sdb
lrwxrwxrwx. 1 root root  9 Mar  9 23:56 pci-0000:00:04.0-scsi-0:0:0:1 -> ../../sda
lrwxrwxrwx. 1 root root 10 Mar  9 23:56 pci-0000:00:04.0-scsi-0:0:0:1-part1 -> ../..
/sda1
lrwxrwxrwx. 1 root root 10 Mar  9 23:56 pci-0000:00:04.0-scsi-0:0:0:1-part2 -> ../..
/sda2
lrwxrwxrwx. 1 root root 10 Mar  9 23:56 pci-0000:00:04.0-scsi-0:0:0:1-part3 -> ../..
/sda3
```

Note: Disk **/dev/sdb** appears as the 10 GB partition that had been created in an earlier practice (part of the 50 GB iSCSI attached volume). On **lab-iscsi-host** that very same volume appeared as **/dev/sdb1**. This is only an observation.

24. The device is now available. It can be partitioned and mounted.

    a. Create a mount point. Run the following command:

    ```
    sudo mkdir /mnt/remote
    ```

    ```
    [opc@lab-node1 ~]$ sudo mkdir /mnt/remote
    ```

    b. Mount the disk with the file system. Run the following command:

    ```
    sudo mount /dev/sdb  /mnt/remote
    ```

    ```
    [opc@lab-node1 ~]$ sudo mount /dev/sdb /mnt/remote
    ```

    c. List the contents of the directory in **/mnt/remote**.

    ```
    ls -l /mnt/remote
    ```

    ```
    [opc@lab-node1 ~]$ ls -l /mnt/remote
    total 4194324
    -rw-r--r--. 1 opc  opc  4294967296 Mar 10 18:42 four-GB-file.dat
    drwx------. 2 root root      16384 Mar 10 18:39 lost+found
    ```

    Note: The file shown had been created in the previous practices.

Solution 6-2: Configuring an iSCSI Target and an Initiator

Overview

There is no automated solution for this practice.

Practice 6-3: Configuring and Testing an OCFS2 Cluster

Overview

In this practice, you will:

- Configure an OCFS2 Cluster
- Test OCFS2 Cluster behavior

Assumptions

- Practices for Lesson 3 have been completed.
- Practice 4-1 has been completed.
- Compute instances **lab-node1**, **lab-node2**, and **lab-iscsi-host** that you created earlier are running and have public IP addresses.
- The private key to authenticate over SSH is stored in the file named **`OL9lab_key`**.

Tasks

1. Working in the OCI console, navigate to **Instances** (select your compartment if not shown).
2. Ensure that all three instances are running.

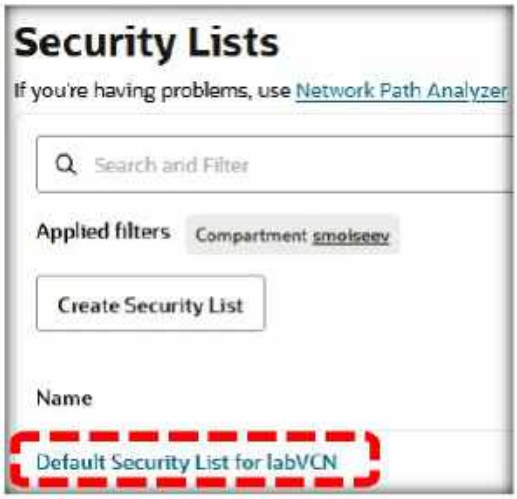
   For the OCFS2 cluster, you will use **lab-iscsi-host** as the OCFS2 server, and the **lab-node** instances as the OCFS2 cluster members sharing the file system.
3. In the OCI console, add a VCN security rule to the Default Security List to permit OCFS cluster traffic.

   The OCI firewall needs to allow port 7777 access on your compartment's VCN for the OCFS2 cluster messages.

In the OCI console, click **Networking → Virtual Cloud Networks**.

4. Verify your compartment (select it, if another is shown) and then click the **labVCN** hyperlink.

5. On the **labVCN** page, click the **Security** tab. Under **Security Lists**, click the **Default Security List for labVCN** hyperlink.



6. Click the **Security rules** tab.
7. Check for any existing rules for TCP access to target port 7777. A rule may already exist. If not, add a new Ingress rule:

   a. Click the **Add Ingress Rules** button.

   b. Complete the presented form:

      1) Leave the **Stateless** slider disabled (default).

2) Add the **Source CIDR.** The value **10.0.0.0/16** is the least restrictive in your compartment.
3) Set the protocol to **TCP**.
4) Leave the **Source Port Range** with the default empty (All) value.
5) Set the Destination Port Range to **7777**.
6) Enter a meaningful description. Example: Port for OCFS cluster communications.
7) Click the **Add Ingress Rules** button when finished.

The rule should appear in the list as the following entry:

☐	No	10.0.0.0/16	TCP	All	7777

8. Configure the Linux firewall on each cluster node, and on the iSCSI host.

Hint: It is best to leave **lab-iscsi-host** for last.

a. Open a command-prompt window on your local PC.

b. Navigate to the directory where the SSH authentication key, **OL9lab_key**, is stored.

c. Connect over SSH to **lab-node1**. Run the following command:

```
ssh -i OL9lab_key opc@<instance-public-IP>
```

d. Run the following command to allow port 7777 traffic over TCP:

```
sudo firewall-cmd --permanent --zone=public --add-port=7777/tcp
```

```
[opc@lab-node1 ~]$ sudo firewall-cmd --permanent --zone=public \
> --add-port=7777/tcp
success
```

Note: Ignore the warning if the port had been previously added.

e. Run the following command to make the new rule active:

```
sudo firewall-cmd --reload
```

```
[opc@lab-node1 ~]$ sudo firewall-cmd --reload
success
```

f. Create a mount point for the future OCFS2 disk. Run the following command:

```
sudo mkdir /u01
```

Do not perform this step on the **lab-node1** and **lab-iscsi-host** instance.

g. Log out of the SSH session. (Keep the connection to **lab-iscsi-host** open for the next steps.)

h. Repeat steps **a** to **g** for **lab-node2** and **lab-iscsi-host**.

9. While connected to the **lab-iscsi-host** instance, run the following command to view the available volumes.

```
sudo fdisk -l
```

```
[opc@lab-iscsi-host ~]$ sudo fdisk -l
...
Disk /dev/sdb: 50 GiB, 53687091200 bytes, 104857600 sectors
Disk model: BlockVolume
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 1048576 bytes
Disklabel type: dos
Disk identifier: 0xb377b3ac

Device     Boot Start      End  Sectors Size Id Type
/dev/sdb1        2048 20973567 20971520  10G 83 Linux
```

10. Assuming the attached iSCSI volume `/dev/sdb` exists, create a new primary partition of 30 GB size. There is already a partition `/dev/sdb1` on disk, use `/dev/sdb2` for the new one.

a. Execute the following command:

```
sudo fdisk /dev/sdb
```

```
[opc@lab-iscsi-host ~]$ sudo fdisk /dev/sdb
```

Ignore the warning displayed.

1) Type `n` at the **fdisk Command** prompt to create a new partition.
2) Type `p` at the **Select** input prompt.
3) For the **Partition number**, type `2` (this is going to the second partition).
4) Keep the default for the **First sector**; just press the **Enter** key.
5) For the last sector, use the size metric. Type **+30G** without spaces.

6) The confirmation is displayed.

```
Command (m for help): n 1
Partition type
   p   primary (1 primary, 0 extended, 3 free)
   e   extended (container for logical partitions)
Select (default p): p 2
Partition number (2-4, default 2): 2 3
First sector (20973568-104857599, default 20973568): 4
Last sector, +/-sectors or +/-size{K,M,G,T,P} (20973568-104857599, default
7599): +30G 5

Created a new partition 2 of type 'Linux' and of size 30 GiB. 6

Command (m for help): w 7
The partition table has been altered.
Syncing disks.
```

7) Write and exit **fdisk** changes by typing **w**.

b. Do not exit the SSH session to the **lab-iscsi-host** instance.

11. Configure iSCSI target for the new disk partition. (Steps are similar to the iSCSI target configuration from before.) Most steps in this task are performed within **targetcli** tool.

a. Run the **targetcli** utility using elevated privileges:

```
sudo targetcli
```

b. Change the location to **/backstores/block**.

```
cd /backstores/block
```

```
/> cd /backstores/block
```

c. Use the volume's device name - **/dev/sdb2** - to create a new storage object. Enter the command:

```
create name=ocfs2-disk dev=/dev/sdb2
```

The name **ocfs2-disk** is arbitrary.

```
/backstores/block> create name=ocsf2-disk dev=/dev/sdb2
Created block storage object ocsf2-disk using /dev/sdb2.
```

d. Change the location to **/iscsi**.

```
cd /iscsi
```

Create a new IQN object - iSCSI qualified name.

```
create
```

```
/backstores/block> cd /iscsi/
/iscsi> create
Created target iqn.2003-01.org.linux-iscsi.lab-iscsi-host.x8664:sn.9fbd54c516c5.
Created TPG 1.
Global pref auto_add_default_portal=true
Created default portal listening on all IPs (0.0.0.0), port 3260.
```

Copy the new IQN value to the clipboard and save it in your scratch document for later use.

Note: TPG stands for the Target Portal Group.

e. Change the location to the above IQN object. Enter the following command:

```
cd <iSCSI-IQN-value>    ### cd iqn<TAB-Key> may not work here
```

```
/iscsi> cd iqn.2003-01.org.linux-iscsi.lab-iscsi-host.x8664:sn.9fbd54c516c5
/iscsi/iqn.20....9fbd54c516c5>
```

Note: You don't usually have to type the whole value; simply enter "cd iqn" and press the TAB key to autocomplete. This may not work if you have multiple IQN values.

f. Change the location to **tpg1/acls**. Do not use the forward slash in front of the **tpg**; this is a relative path. Enter the following command:

```
cd  tpg1/acls
```

```
/iscsi/iqn.20....9fbd54c516c5> cd tpg1/acls
```

g. Copy the iSCSI IQN value from above. Record the IQN value in the scratch text file for future use. Create the node ACL (Access Control List). Paste the IQN value from the clipboard to complete the **create** command parameter. Enter the following command:

```
create <iSCSI-IQN-value>
```

A confirmation appears about the node ACL creation.

```
/iscsi/iqn.20...6c5/tpg1/acls> create iqn.2003-01.org.linux-iscsi.lab-iscsi-host
.x8664:sn.9fbd54c516c5
Created Node ACL for iqn.2003-01.org.linux-iscsi.lab-iscsi-host.x8664:sn.9fbd54c
516c5
```

h. Navigate to the parent location using double-period notation. Set parameters.

1) Enter the command **cd** .. (there are two period characters after the white space). The location is set to **tpg1**.

```
cd  ..
```

2) Set the **authentication** attribute to zero. Enter the following command:

```
set attribute authentication=0
```

3) Set the **generate_node_acls** attribute to one. Enter the following command:

```
set attribute generate_node_acls=1
```

```
/iscsi/iqn.20...6c5/tpg1/acls> cd ..
/iscsi/iqn.20...54c516c5/tpg1> set attribute authentication=0
Parameter authentication is now '0'.
/iscsi/iqn.20...54c516c5/tpg1> set attribute generate_node_acls=1
Parameter generate_node_acls is now '1'.
/iscsi/iqn.20...54c516c5/tpg1>
```

4) Set the attribute **demo_mode_write_protect=0**. Enter the following command:

```
set attribute demo_mode_write_protect=0
```

i. Navigate to the **luns** location (relative path).

```
cd luns
```

j. Create the **luns** entry. The name should be the same as in **Step 11c**.

```
create /backstores/block/ocfs2-disk
```

LUN creation is confirmed.

```
/iscsi/iqn.20...54c516c5/tpg1> cd luns
/iscsi/iqn.20...6c5/tpg1/luns> create /backstores/block/ocsf2-disk
Created LUN 0.
Created LUN 0->0 mapping in node ACL iqn.2003-01.org.linux-iscsi.lab-
x8664:sn.9fbd54c516c5
```

k. Change the location to the root.

```
cd /
```

```
/iscsi/iqn.20...6c5/tpg1/luns> cd /
/>
```

l. Use the `ls` command to view the details. Verify against the directions above.

```
ls
```

```
/> ls
o- / ..... [...]
  o- backstores ..... [...]
  | o- block ..... [Storage Objects: 2]
  | | o- ocfs2-disk ..... [/dev/sdb2 (30.0GiB) write-thru activated]
  | | | o- alua ..... [ALUA Groups: 1]
  | | |   o- default_tg_pt_gp ..... [ALUA state: Active/optimized]
  | | o- target-disk ..... [/dev/sdb1 (10.0GiB) write-thru activated]
  | |   o- alua ..... [ALUA Groups: 1]
  | |     o- default_tg_pt_gp ..... [ALUA state: Active/optimized]
  | o- fileio ..... [Storage Objects: 0]
  | o- pscsi ..... [Storage Objects: 0]
  | o- ramdisk ..... [Storage Objects: 0]
  o- iscsi ..... [Targets: 2]
  | o- iqn.2003-01.org.linux-iscsi.lab-iscsi-host.aarch64:sn.a6e67f1dd121 ..... [TPGs: 1]
  | | o- tpg1 ..... [gen-acls, no-auth]
  | |   o- acls ..... [ACLs: 1]
  | |   | o- iqn.2003-01.org.linux-iscsi.lab-iscsi-host.aarch64:sn.a6e67f1dd121 .. [Mapped LUNs: 1]
  | |   |   o- mapped_lun0 ..... [lun0 block/target-disk (rw)]
  | |   o- luns ..... [LUNs: 1]
  | |   | o- lun0 ..... [block/target-disk (/dev/sdb1) (default_tg_pt_gp)]
  | |   o- portals ..... [Portals: 1]
  | |     o- 0.0.0.0:3260 ..... [OK]
  | o- iqn.2003-01.org.linux-iscsi.lab-iscsi-host.aarch64:sn.df82b6374fb9 ..... [TPGs: 1]
  |   o- tpg1 ..... [gen-acls, no-auth]
  |     o- acls ..... [ACLs: 1]
  |     | o- iqn.2003-01.org.linux-iscsi.lab-iscsi-host.aarch64:sn.df82b6374fb9 .. [Mapped LUNs: 1]
  |     |   o- mapped_lun0 ..... [lun0 block/ocfs2-disk (rw)]
  |     o- luns ..... [LUNs: 1]
  |     | o- lun0 ..... [block/ocfs2-disk (/dev/sdb2) (default_tg_pt_gp)]
  |     o- portals ..... [Portals: 1]
  |       o- 0.0.0.0:3260 ..... [OK]
  o- loopback ..... [Targets: 0]
  o- vhost ..... [Targets: 0]
/>
```

m. Save the configuration.

```
saveconfig
```

```
/> saveconfig
Configuration saved to /etc/target/saveconfig.json
```

n. Exit the **targetcli** session.

```
exit
```

```
/> exit
Global pref auto_save_on_exit=true
Last 10 configs saved in /etc/target/backup/.
Configuration saved to /etc/target/saveconfig.json
```

o. Exit the SSH session to the **lab-iscsi-host** (either click **CTRL-D**, or type `exit` and press the **Enter** key).

12. Install OCFS2 utilities.

    a. Start the SSH connection to the first instance: **lab-node1**. Run the local PC command from the directory where the SSH authentication key **OL9lab_key** is located:

```
ssh -i OL9lab_key opc@<lab-node1-public-IP>
```

    b. While connected to the instance, execute the installer using the following command:

```
sudo dnf install -y ocfs2-tools
```

```
[opc@lab-node1 ~]$ sudo dnf install -y ocfs2-tools
```

    Note: You will install the package on the remaining cluster members (in our case, only on the **lab-node2** instance) at a later step.

13. Add a new OCFS2 cluster.

```
sudo o2cb add-cluster myOLcluster
```

```
[opc@lab-node1 ~]$ sudo o2cb add-cluster myOLcluster
```

    Note: The cluster name is arbitrarily chosen for this practice.

14. Add cluster members to this OCFS2 cluster. Use private IP addresses only.

    a. Run the following command to add **lab-node1** to the OCFS2 cluster:

```
sudo o2cb add-node myOLcluster lab-node1 \
  --ip <lab-node1 private IP>
```

```
[opc@lab-node1 ~]$ sudo o2cb add-node myOLcluster lab-node1  --ip 10.0.0.200
```

    b. Run the following command to add **lab-node2** to the OCFS2 cluster.

```
sudo o2cb add-node myOLcluster lab-node2 \
  --ip <lab-node2 private IP>
```

```
[opc@lab-node1 ~]$ sudo o2cb add-node myOLcluster lab-node2  --ip 10.0.0.243
```

    The cluster configuration file is created in the `/etc/ocfs2/cluster.conf` file. This file needs to be copied to the second cluster member (and all subsequent cluster members).

    c. Copy the configuration file to the `/tmp` directory to prepare for copying it to another node.

```
cp  /etc/ocfs2/cluster.conf /tmp
```

d. Optionally, view the contents.

```
cat /tmp/cluster.conf
```

```
[opc@lab-node1 ~]$ cat /tmp/cluster.conf
cluster:
        name = myOLcluster
        heartbeat_mode = local
        node_count = 2

node:
        cluster = myOLcluster
        number = 0
        ip_port = 7777
        ip_address = 10.0.0.200
        name = lab-node1

node:
        cluster = myOLcluster
        number = 1
        ip_port = 7777
        ip_address = 10.0.0.243
        name = lab-node2
```

e. Exit the SSH session with the **lab-node1** instance.

f. Copy the cluster configuration file to the local PC by running at the local command prompt: **scp -i OL9lab_key opc@**<lab-node1 public IP>**:/tmp/cluster.conf .** ← The period character is the last part of the command (following a white space), which is the current local working directory. The file is copied to the local PC.

```
scp -i OL9lab_key opc@<lab-node1 public IP>:/tmp/cluster.conf .
```

```
D:\OL9Labs>scp -i OL9lab_key opc@150.136.71.103:/tmp/cluster.conf .
cluster.conf                                  100%  272     8.6KB/s   00:00
```

15. Complete the configuration of the second (or any subsequent) cluster member instance. These steps are performed on the **lab-node2** instance. (Some screen capture examples were generated from the **lab-node1** instance SSH connection.)

a. Transfer the OCFS2 cluster configuration file to the second instance. Copy the cluster configuration file from the local PC by running it at the local command prompt:

```
scp -i OL9lab_key cluster.conf opc@<lab-node2-public-IP>:/tmp
```

The file is transferred to the temporary directory on the second instance.

```
D:\OL9Labs>scp -i OL9lab_key cluster.conf opc@150.136.108.7:/tmp
cluster.conf                                  100%  272     8.6KB/s   00:00
```

b. Start the SSH connection to the second instance: **lab-node2**. Run the local PC command from the directory where the SSH authentication key **OL9lab_key** is located:

```
ssh -i OL9lab_key opc@<lab-node2-public-IP>
```

c. While connected to the instance, execute the installer using the following command (**lab-node2** needs iSCSI initiator capabilities, too):

```
sudo dnf install -y ocfs2-tools iscsi-initiator-utils
```

d. Enable and start **iscsid** on **lab-node2** instance:

```
sudo systemctl enable iscsid ; sudo systemctl start iscsid
```

e. Verify that **iscsid** is active:

```
sudo systemctl status iscsid
```

```
[opc@lab-node2 ~]$ sudo systemctl status iscsid
● iscsid.service - Open-iSCSI
   Loaded: loaded (/usr/lib/systemd/system/iscsid.service; enabled; vendor preset:
   Active: active (running) since Wed 2023-03-08 23:00:12 GMT; 2 days ago
```

f. Use the vi editor to edit the `/etc/iscsi/initiatorname.iscsi` file available in lab-node2 instance.

```
sudo vi /etc/iscsi/initiatorname.iscsi
```

g. Replace the volume's IQN in the configuration file with the value that you had saved from the iSCSI target of the **ocfs2-disk** partition IQN.

1) In the **vi** editor, while in "browse" mode, press the forward slash character and type **iqn.** Then press **Enter** to locate the IQN entry.
2) Hold the **Shift** key and press the letter **C** key (**Shift-c**). The editor wipes off the text from the cursor position to the end of the line and enters "INSERT" mode.
3) While in "insert" mode, paste the IQN value from the iSCSI target. (This IQN value is different from previously configured iSCSI target. Be mindful.)

```
InitiatorName=iqn.2003-01.org.linux-iscsi.lab-iscsi-host.x8664:sn.9fbd54c516c5
~
```

4) Switch to "browse" mode by pressing the **ESC** key. Then press the colon (**:**), **w**, and **q** keys in sequence to write the changes (**:wq**).

Hint: If you have not copied the IQN of the iSCSI target device, open another local command-prompt window and connect over SSH to the **lab-iscsi-host** instance. When connected, run **sudo targetcli** and use the **ls** command, and the device's IQN will appear in two places. Exit the **targetcli** and close the SSH connection to the **lab-iscsi-host**.

16. Perform the iSCSI Target discovery. For the iSCSI target host IP address, use the private IP in the OCI (look up from the **OCI console → Instances**). This is both faster and more secure as compared to the public IP of the **lab-iscsi-host**.

Name	State	Public IP	Private IP
lab-iscsi-host	Running	132.226.66.137	10.0.0.09

Note: Additional configuration will be necessary if you need to access the iSCSI target from outside OCI. This will require the Ingress rules definitions for the public network. In your environment, all IP addresses will be different.

a. Execute the following command:

```
sudo iscsiadm -m discovery -t st -p <iSCSI_target_private_IP>
```

```
[opc@lab-node2 ~]$ sudo iscsiadm -m discovery -t st -p 10.0.0.9
10.0.0.9:3260,1 iqn.2003-01.org.linux-iscsi.lab-iscsi-host.x8664:sn.9cc4c31fca17
10.0.0.9:3260,1 iqn.2003-01.org.linux-iscsi.lab-iscsi-host.x8664:sn.9fbd54c516c5
```

Note: First IQN belongs to the **target-disk**, while the second one belongs to the **ocfs2-disk.**

b. Update iSCSI DB. Execute the following command:

```
sudo iscsiadm -m discoverydb -t st -p <iSCSI_target_private_IP>
```

c. Restart the **iscsid** service.

```
sudo systemctl restart iscsid
```

17. Log in to the iSCSI session. Use background login by adding **&** at the end of the command.

a. Execute the following command (**lab-node2**):

```
sudo iscsiadm -m node -l &
```

You will get the prompt back. Press the **Enter** key if you are not immediately seeing the prompt.

b. Verify the device availability.

```
ls -l /dev/disk/by-path
```

Note the device name - similar to **/dev/sdb**, or **/dev/sdc**, and so on. In the example below, OCFS2 volume shows as **/dev/sdd**. Double-check the OCFS2 volume's IQN before continuing (refer to the recorded IQN from the iSCSI Target configuration of the **ocfs2-disk – Step 11d**).

```
[opc@lab-node1 ~]$ ls -l /dev/disk/by-path
total 0
lrwxrwxrwx. 1 root root  9 Mar 11 22:42 ip-10.0.0.9:3260-iscsi-iqn.2003-01.org.linux
-iscsi.lab-iscsi-host.x8664:sn.9cc4c31fce17-lun-0 -> ../../sdc
lrwxrwxrwx. 1 root root  9 Mar 11 22:45 ip-10.0.0.9:3260-iscsi-iqn.2003-01.org.linux
-iscsi.lab-iscsi-host.x8664:sn.9fbd54c516c5-lun-0 -> ../../sdd
lrwxrwxrwx. 1 root root  9 Mar  9 23:56 pci-0000:00:04.0-scsi-0:0:0:1 -> ../../sda
lrwxrwxrwx. 1 root root 10 Mar  9 23:56 pci-0000:00:04.0-scsi-0:0:0:1-part1 -> ../..
/sda1
lrwxrwxrwx. 1 root root 10 Mar  9 23:56 pci-0000:00:04.0-scsi-0:0:0:1-part2 -> ../..
/sda2
lrwxrwxrwx. 1 root root 10 Mar  9 23:56 pci-0000:00:04.0-scsi-0:0:0:1-part3 -> ../..
/sda3
```

18. Set up the cluster configuration on **lab-node2**.

a. Create the directory for the cluster configuration file and place the file in it.

1) Run the following command:

```
sudo mkdir /etc/ocfs2
```

The directory is created.

2) Copy the file from the temporary directory to the **/etc/ocfs2** location.

```
sudo cp  /tmp/cluster.conf   /etc/ocfs2
```

The file is copied.

b. Cluster setup continues on one of the cluster members only. While connected to the **lab-node2** instance, complete the following steps:

1) Run the command and answer **y** when prompted:

```
sudo /sbin/o2cb.init configure
```

2) At the prompt **Cluster stack backing O2CB [o2cb]:** (keep/Enter) press the **Enter** key.

3) At the prompt **Cluster to start on boot,** type **myOLcluster** and press the **Enter** key. Value is case-sensitive.

4) Accept all the following defaults:

```
[opc@lab-node2 ~]$ sudo /sbin/o2cb.init configure
Configuring the O2CB driver.

This will configure the on-boot properties of the O2CB driver.
The following questions will determine whether the driver is loaded on
boot.  The current values will be shown in brackets ('[]').  Hitting
<ENTER> without typing an answer will keep that current value.  Ctrl-C
will abort.

Load O2CB driver on boot (y/n) [n]: y                                    1
Cluster stack backing O2CB [o2cb]:                                       2
Cluster to start on boot (Enter "none" to clear) [ocfs2]: myOLcluster    3
Specify heartbeat dead threshold (>=7) [31]:
Specify network idle timeout in ms (>=5000) [30000]:                     4
Specify network keepalive delay in ms (>=1000) [2000]:
Specify network reconnect delay in ms (>=2000) [2000]:
Writing O2CB configuration: OK
checking debugfs...
Loading stack plugin "o2cb": OK
Loading filesystem "ocfs2_dlmfs": OK
Creating directory '/dlm': OK
Mounting ocfs2_dlmfs filesystem at /dlm: OK
Setting cluster stack "o2cb": OK
Registering O2CB cluster "myOLcluster": OK
Setting O2CB cluster timeouts : OK
```

c. Check the cluster status.

```
sudo /sbin/o2cb.init status
```

```
Checking O2CB cluster "myOLcluster": Online
  Heartbeat dead threshold: 31
  Network idle timeout: 30000
  Network keepalive delay: 2000
  Network reconnect delay: 2000
  Heartbeat mode: Local
Checking O2CB heartbeat: Not active
Debug file system at /sys/kernel/debug: mounted
```

Note: The cluster heartbeat should be inactive at this time because the shared volume has not been formatted to the ocfs2 file system yet.

19. Enable cluster services (do this on all member instances).

```
sudo systemctl enable o2cb
sudo systemctl start o2cb
```

```
[opc@lab-node2 ~]$ sudo systemctl enable o2cb
Created symlink /etc/systemd/system/multi-user.target.wants/o2cb.service → /usr/lib/
systemd/system/o2cb.service.
[opc@lab-node2 ~]$ sudo systemctl start o2cb
```

20. Set Linux kernel parameters (all member instances).

```
sudo sysctl kernel.panic=30
sudo sysctl kernel.panic_on_oops=1
```

```
[opc@lab-node2 ~]$ sudo sysctl kernel.panic=30
kernel.panic = 30
[opc@lab-node2 ~]$ sudo sysctl kernel.panic_on_oops=1
kernel.panic_on_oops = 1
```

Note: To persist the above, edit **`/etc/sysctl.conf`** and add these values. Otherwise, these values will be lost after the instance reboot.

21. Close the SSH session to the **lab-node2** instance.
22. Open an SSH session to the other cluster node instance.
23. Repeat **Steps 16, 17, 18 b., 18 c., 19,** and **20** of this practice on the other cluster members (**lab-node1**).
24. Continue with the SSH connection to **lab-node1**.

    There was an available partition that you had created before; it should be available as **`/dev/sdb2`** (or **`/dev/sdc`**).
25. Create the ocfs2 file system on this partition. Do this on one of the cluster members only. (Optionally, verify availability using **`sudo fdisk -l`**.)

```
sudo fdisk -l
```

```
[opc@lab-node1 ~]$ sudo fdisk -l
```

Remember to complete **Steps 19** and **20** on **lab-node1** host.

```
Disk /dev/sdb: 10 GiB, 10737418240 bytes, 20971520 sectors
Disk model: target-disk
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 1048576 bytes


Disk /dev/sdc: 30 GiB, 32212254720 bytes, 62914560 sectors
Disk model: ocsf2-disk
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 1048576 bytes
```

a. If the new partition from the **lab-iscsi-host** (30 GB) is not visible:

1) Perform the following commands to update visibility of the new ISCSI target:

```
sudo iscsiadm -m discovery -t st \
   -p <private IP of the lab-iscsi-host>
sudo iscsiadm -m discoverydb -t st \
   -p <private IP of the lab-iscsi-host>
```

2) Restart **iscsid**.

```
sudo systemctl restart iscsid
```

3) Repeat sudo fdisk -l.

```
sudo fdisk -l
```

b. Execute the following command (option **-L** causes the file system label **ocfs2** - value is arbitrary - to be assigned):
sudo mkfs.ocfs2 -L "ocfs2" /dev/sdc (or **/dev/sdd**, depending on what **fdisk** had reported for the **Disk model: ocfs2-disk** entry).
The device will be associated with the label "**ocfs2**", which had been chosen arbitrarily for this course. This takes a few moments.

```
sudo mkfs.ocfs2 -L "ocfs2" /dev/sdc
```

26. You had created a mount point **/u01** in the previous tasks. If the directory **/u01** does not exist, create it using **sudo mkdir /u01** command. Use this directory to mount the shared disk:

```
sudo mount -L "ocfs2" /u01
```

Note: If the command fails, simply run **sudo /sbin/o2cb.init configure** again and try mounting after that. See task **18**, step **b** for details (to update the cluster name to **myOLcluster**). Then repeat the **mount** command.

```
sudo /sbin/o2cb.init configure
sudo mount -L "ocfs2" /u01
```

27. Check the cluster status again.

```
sudo /sbin/o2cb.init status
```

Now the heartbeat should be active.

```
Checking O2CB heartbeat: Active
```

28. Change permissions on the **/u01** directory..

```
sudo chmod 777 /u01
```

```
[opc@lab-node1 ~]$ sudo chmod 777 /u01
```

Note: The **/u01** directory is owned by the **root** user and the **opc** user has no write access. That is the reason to change permissions.

29. Create a file on the shared volume; the contents do not matter.

```
ls -la  /  > /u01/sample-ls.txt
```

The directory listing of the file system root would be written to the file **/u01/sample-ls.txt** on the shared disk.

```
[opc@lab-node1 ~]$ ls -la / > /u01/sample-ls.txt
```

30. View the sample file while still on **lab-node1** instance.

```
cat /u01/sample-ls.txt
```

Partial output is shown.

```
[opc@lab-node1 ~]$ cat /u01/sample-ls.txt
total 4194332
dr-xr-xr-x.  21 root   root         285 Mar 11 21:39 .
dr-xr-xr-x.  21 root   root         285 Mar 11 21:39 ..
dr-xr-xr-x.   2 root   root           6 Jan 10  2022 afs
lrwxrwxrwx.   1 root   root           7 Jan 10  2022 bin -> usr/bin
dr-xr-xr-x.   5 root   root        4096 Mar 10 00:07 boot
```

31. Disconnect the SSH session from **lab-node1**.
32. Start the SSH connection to **lab-node2**.

33. Mount the shared disk on the second instance. Device information needs to be refreshed on this node or mounting will fail.

    a. View the shared volume device info.

    ```
    sudo fdisk -l
    ```

    Note: In the example below the shared volume shows as **/dev/sdc.** It may be different for you and may be different compared with the **lab-node1** discovery results.

    Mounting by label is safe and not dependent on the discovered device name.

    ```
    [opc@lab-node2 ~]$ sudo fdisk -l
    Disk /dev/sda: 46.6 GiB, 50010783744 bytes, 97677312 sectors

    Disk /dev/sdc: 30 GiB, 32212254720 bytes, 62914560 sectors
    Units: sectors of 1 * 512 = 512 bytes
    Sector size (logical/physical): 512 bytes / 4096 bytes
    I/O size (minimum/optimal): 4096 bytes / 1048576 bytes
    ```

    b. You had created a mount point **/u01** in the previous tasks. Use this to mount the shared disk.

    ```
    sudo mount -L "ocfs2" /u01
    ```

    c. List the directory contents (again, use **sudo** or change the access to permit the **opc** user to work in the **/u01** location).

    ```
    ls -l /u01
    ```

    You should see the text file **sample-ls.txt**.

    d. You can view or edit this file now. To view, repeat **Step 30** on this instance.

    ```
    [opc@lab-node2 ~]$ cat /u01/sample-ls.txt
    total 4194332
    dr-xr-xr-x.  21 root   root         285 Mar 11 21:39 .
    dr-xr-xr-x.  21 root   root         285 Mar 11 21:39 ..
    dr-xr-xr-x.   2 root   root           6 Jan 10  2022 afs
    lrwxrwxrwx.   1 root   root           7 Jan 10  2022 bin -> usr/bin
    dr-xr-xr-x.   5 root   root        4096 Mar 10 00:07 boot
    ```

Solution 6-3: Configuring and Testing an OCFS2 Cluster

Overview

There is no automated solution for this practice.

Reference: o2cb command syntax

```
usage: o2cb [--config-file=path] [-h|--help] [-v|--verbose] [-V|--
version] COMMAND [ARGS]
```

COMMAND	Capability
add-cluster	Add cluster to the config file.
remove-cluster	Removes cluster from the config file.
add-node	Adds a node to the cluster in the config file.
remove-node	Removes a node from the cluster in the config file.
add-heartbeat	Adds a heartbeat region to the cluster in the config file.
remove-heartbeat	Removes a heartbeat region from the cluster in the config file.
heartbeat-mode	Toggles the heartbeat mode between global and local.
list-clusters	Lists all the cluster names in the config file.
list-cluster	Lists all the nodes and heartbeat regions associated with the cluster in the config file.
list-nodes	Lists all the nodes associated with the cluster in the config file.
list-heartbeats	Lists all the heartbeat regions associated with the cluster in the config file.

register-cluster	Registers the cluster with configfs.
unregister-cluster	Unregisters the cluster from configfs.
start-heartbeat	Starts global heartbeat.
stop-heartbeat	Stops global heartbeat.
cluster-status	Returns 0 if cluster online, 1 otherwise.