



**Hewlett Packard
Enterprise**

Fundamentals of OpenStack® Technology

Lab guide

H6C68S E.01

Use of this material to deliver training without prior written permission from HPE is prohibited.

© Copyright 2017 Hewlett Packard Enterprise Development LP

The information contained herein is subject to change without notice. The only warranties for HPE products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Hewlett Packard Enterprise shall not be liable for technical or editorial errors or omissions contained herein.

This is an HPE copyrighted work that may not be reproduced without the written permission of Hewlett Packard Enterprise. You may not use these materials to deliver training to any person outside of your organization without the written permission of HPE.

Microsoft®, Windows®, Windows NT® are U.S. registered trademarks of Microsoft Corporation.

UNIX® is a registered trademark of The Open Group.

All other product names mentioned herein may be trademarks of their respective companies.

Export Compliance Agreement

Export Requirements. You may not export or re-export products subject to this agreement in violation of any applicable laws or regulations.

Without limiting the generality of the foregoing, products subject to this agreement may not be exported, re-exported, otherwise transferred to or within (or to a national or resident of) countries under U.S. economic embargo and/or sanction including the following countries: Cuba, Iran, North Korea, Sudan and Syria. This list is subject to change.

In addition, products subject to this agreement may not be exported, re-exported, or otherwise transferred to persons or entities listed on the U.S. Department of Commerce Denied Persons List; U.S. Department of Commerce Entity List (15 CFR 744, Supplement 4); U.S. Treasury Department Designated/Blocked Nationals exclusion list; or U.S. State Department Debarred Parties List; or to parties directly or indirectly involved in the development or production of nuclear, chemical, or biological weapons, missiles, rocket systems, or unmanned air vehicles as specified in the U.S. Export Administration Regulations (15 CFR 744); or to parties directly or indirectly involved in the financing, commission or support of terrorist activities.

By accepting this agreement you confirm that you are not located in (or a national or resident of) any country under U.S. embargo or sanction; not identified on any U.S. Department of Commerce Denied Persons List, Entity List, US State Department Debarred Parties List or Treasury Department Designated Nationals exclusion list; not directly or indirectly involved in the development or production of nuclear, chemical, biological weapons, missiles, rocket systems, or unmanned air vehicles as specified in the U.S. Export Administration Regulations (15 CFR 744), and not directly or indirectly involved in the financing, commission or support of terrorist activities.

Printed in US

Fundamentals of OpenStack® Technology

Lab guide

March 2017

Table of Contents

Module 1 Lab OpenStack® Docs and Building a DevStack Environment	1-1
Module 2 Lab OpenStack® Management Tools	2-1
Module 3 Lab OpenStack® Keystone (Identity).....	3-1
Module 4 Lab Glance (Image) Service	4-1
Module 5 Labs OpenStack® Neutron (Networking).....	5-1
Module 6 Lab OpenStack® Nova (Compute)	6-1
Module 7 Lab OpenStack® Cinder (Block Storage)	7-1
Module 8 Lab OpenStack® Object Storage (Swift).....	8-1
Module 9 Lab OpenStack® Heat (Orchestration)	9-1
Module 10 Lab OpenStack® Ceilometer (Telemetry)	10-1
Module 11 Lab Appendix—Installing and Configuring DevStack.....	11-1
Module 12 Lab Answers to Lab Questions	12-1

Fundamentals of OpenStack® Technology

Lab 01—OpenStack® Docs and Building a DevStack Environment

Objectives

After completing this lab, you should be able to:

- Locate useful OpenStack® development-related documents
- Access the course OpenStack® environment
- Build your DevStack environment

Requirements

- Access to HPE Virtual Lab equipment for completing the OpenStack® labs
- Information required to access and log in to your assigned VM (provided by your instructor)
- Approximately 30 minutes to complete

Exercise 1—Accessing the Lab VM and starting the OpenStack® environment

Prerequisites

- Access to the Internet
- PuTTY SSH client installed at the landing desktop
- The login credentials for the lab VM

Lab environment

Each student has a virtual machine assigned, containing the OpenStack® development environment (DevStack) ready to run.

DevStack is a documented shell script used to build the complete OpenStack® development environment. The OpenStack® services, such as Nova, Keystone, and Glance, can be scaled out and grown horizontally to accommodate large-scale cloud deployment environments. However, for the purposes of these labs, those services reside on your virtual system's VM.

Each lab provides you with the opportunity to examine the primary components of each of the OpenStack® services and learn how you can access these services, manage them, and verify their operations.

The following table provides a reference point for the setup of your lab settings and a place in which you can document the information you will need to reference while performing steps from the labs.

Lab system VM	Information
Hostname	hpedevstack
RAM	8GB
Disk space	60GB
Network	Bridged
OS version	Ubuntu LTS 16.04 with OpenStack® Newton installed
Host user name/password	stack/hpinvent
Host IP Address	192.168.5.4

Exercise 2—Accessing the HPE vLabs environment

To reach the landing server used to perform the labs in this lab guide, you need to log in to the vLabs environment.

- At your workstation, open the web browser and go to the <http://vlabs.hpe.com> URL.

- If this is the first time that you are accessing the vLabs environment, make sure that your browsers meet the requirements by following the **Test Access** link. Otherwise, click the **Access Virtual Labs Environment** link.

3. At the Virtual Labs Gateway page, enter the username and password assigned to you by the instructor, and click **Sign In**.

**Virtual Labs
Gateway**

Username:

Password:

Keyboard: English (United States) ▾

SIGN IN **HELP**



4. At the landing desktop, use the username of **Administrator** and the password of **hpinvent**.

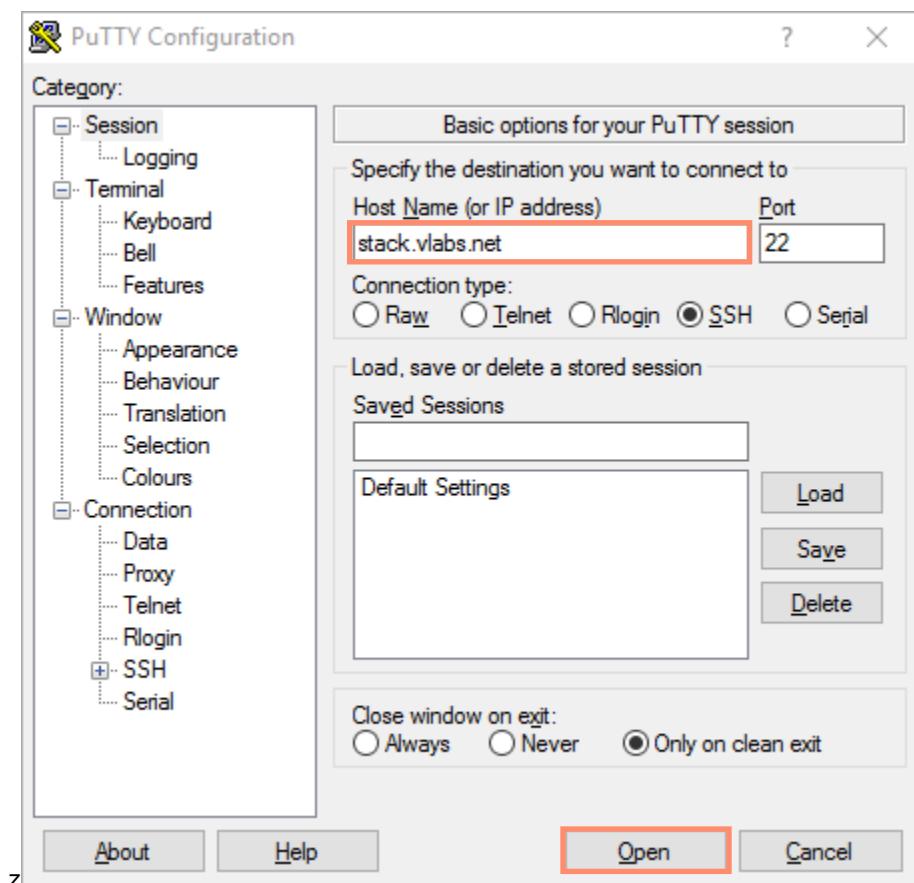
Exercise 3—Accessing the assigned VM

Before proceeding with the labs, you must launch the DevStack environment. DevStack is a script used to quickly create an OpenStack® development environment. DevStack is already installed and configured on your lab VM. To boot the environment, perform the steps below:

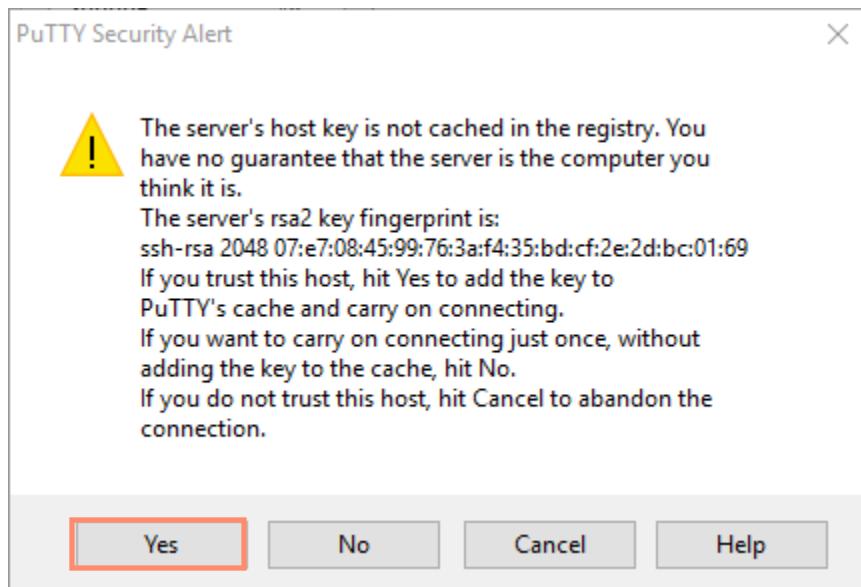
1. At the landing desktop, run the PuTTY application.



2. Enter the `stack.vlabs.net` Host Name and click the **Open** button.



3. If necessary, at the PuTTY Security Alert window, click **Yes**.



4. Log in with the username of **stack**, and the password of **hpinvent**.

The image shows a terminal window titled 'stack@stack: ~'. The window displays the following text:

```
stack@stack: ~
login as: stack
stack@192.168.5.4's password:
Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-53-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage
Last login: Wed Dec 21 09:12:05 2016
stack@stack:~$
```

5. To start building the OpenStack® development environment, change your directory to the **devstack** folder with the `cd ~/devstack` command.
6. To start the boot script, enter the following command: `./stack.sh`
7. Wait for the script to finish. Leave this window open for the next lab.

NOTE: The execution of the `./stack.sh` command might take about 21 minutes. After you have started the process, do not interrupt it.

NOTE: In case of any troubles with your DevStack installation, you can run `./unstack.sh` and, when complete, start the environment again.

NOTICE: After running the `unstack.sh` command, all your previous work will be destroyed, including images imported, networks built, and security associations and images.

NOTE: Proceed with the next exercise while environment is building.

Exercise 4—Obtaining the required OpenStack® documentation

In addition to this lab guide, several other OpenStack® reference resources are useful in completing the labs. These resources are available on the OpenStack® documentation home page.

To access the documentation, perform the steps below on your local workstation:

8. Access the OpenStack® documentation page at <http://docs.openstack.org>. At the home page, select the **More Releases & Languages** menu.

The screenshot shows the OpenStack Documentation homepage. At the top, there is a navigation bar with links for Software, Users, Community, Marketplace, Events, Learn, Docs, Join, and Log In. Below the navigation bar, there is a search bar and a cartoon character of a woman holding a tablet, labeled "Elissa Murphy OpenStack Operator". The main content area features a section titled "Welcome to OpenStack Documentation" and a paragraph about what OpenStack is. Below this, there is a section titled "What Are The Next Steps?" with three buttons: "View The Docs", "Get OpenStack", and "Get Training". A search bar is also present. On the right side, there is a "Documentation for Newton (October 2016)" section with a note that it is the latest release. A "More Releases & Languages" dropdown menu is open, showing options for "Newton (current release)", "Mitaka", and "Liberty".

9. From the list of releases, select **Newton**.

The screenshot shows the OpenStack Documentation homepage again. The "More Releases & Languages" menu is open, displaying a list of releases: "Newton (current release)", "Mitaka", and "Liberty". The "Newton (current release)" option is highlighted with a red border. At the bottom of the page, there are links for "Release Notes" and "User Guides".

10. Scroll down to the bottom of the web page. In the API Guides section, click the **API Guide** link.

The screenshot shows a section titled "API Guides" with a sub-section "API Guide" highlighted by a red box. Below it is "OpenStack API Documentation". To the right, there is a brief description: "Tools for OpenStack Training. Automatic deployment of lean virtualized multi-node openstack cluster."

NOTICE: In this release of the OpenStack® documentation, there is no PDF version of the documentation available for the download.

11. At the left-hand side, in the Contents area, click **API quick-start examples**.

The screenshot shows the "OpenStack API Documentation" page. On the left, a sidebar titled "Contents" has a list of links. The "API quick-start examples" link is highlighted with a red box. The main content area contains a brief introduction to the OpenStack APIs and links to "Bare Metal API v1 (microversions)" and "Block Storage API v3 (microversions)".

12. To access the API documentation, click the **OpenStack APIs** link at the bottom of the page.

- [OpenStack APIs](#)
 - [Authentication and API request workflow](#)
 - [Authenticate](#)
 - [Send API requests](#)
 - [OpenStack command-line clients](#)

13. The OpenStack APIs page provides the information on the API request authentication and workflow, as well as the information on sending the API requests. Bookmark this page in your browser. Most of the commands used in this lab guide are in the **OpenStack command-line clients** section. Click the **OpenStack command line clients** link in the Contents area at the left-hand side.

The screenshot shows the "OpenStack APIs" page. On the left, a sidebar titled "Contents" has a list of links. The "OpenStack command-line clients" link is highlighted with a red box. The main content area contains information about authentication, credentials, and API requests, along with a note about the X-Auth-Token header.

14. For information about the command-line clients, click the **OpenStack Command-Line Interface Reference** link in the OpenStack command-line clients section.

OpenStack command-line clients

For scripting work and simple requests, you can use a command-line client like the `openstack-client` client. This client enables you to use the Identity, Compute, Block Storage, and Object Storage APIs through a command-line interface. Also, each OpenStack project has a related client project that includes Python API bindings and a command-line interface (CLI).

For information about the command-line clients, see [OpenStack Command-Line Interface Reference](#).

15. As an example of how to locate the syntax for a particular CLI command, complete the following steps to locate the syntax for the `openstack user list` command:

- At the OpenStack Command-Line interface Reference page, click **Search in this guide** link.

The screenshot shows the OpenStack Documentation website. At the top, there's a navigation bar with links for Software, Users, Community, Marketplace, Events, Learn, and Docs. Below the navigation bar, there's a search bar and a main content area. The main content area has a title "OpenStack Command-Line Interface Reference". Underneath the title, there's an "Abstract" section, a "Contents" section, and a "Search in this guide" button, which is highlighted with a red box. The "Contents" section lists various services and their command-line clients, such as Application Catalog, Backup, Compute, Container Infrastructure Management, Database, DNS, Image, and Object Storage services. Below the "Contents" section, there's a yellow box containing an "Important" note: "OpenStackClient is a command-line client for OpenStack that brings the command set".

- In the search section, click the **Search Page** link.

The screenshot shows the "Search in this guide" page. It features a search bar at the top and a list of search results. The first result is "index service (gnocchi) command-line client", followed by "Workflow service (mistral) command-line client". Below the search results, there's a list of links under the heading "Important": "Object Storage service (swift) command-line client", "Orchestration service (heat) command-line client", "Rating service (cloudkitty) command-line client", "Shared File Systems service (manila) command-line client", "Software Development Lifecycle Automation service (solum) command-line client", "Telemetry service (ceilometer) command-line client", "A time series storage and resources index service (gnocchi) command-line client", and "Workflow service (mistral) command-line client".

Search in this guide

- [Search Page](#)

c. In the search field, enter the `openstack user list` command and click the **search** button.

• Backup, Restore, and Disaster Recovery service (freezer) command-line client
 • Bare Metal service (ironic) command-line client
 • Block Storage service (cinder) command-line client
 • Clustering service (senlin) command-line client
 • Compute service (nova) command-line clients
 • Container Infrastructure Management service (magnum) command-line client
 • Database service (trove) command-line clients
 • DNS service (designate) command-line client
 • Image service (glance) command-line clients

d. Click the first link on the results page to access the documentation.

All Wiki Docs Videos Q & A

About 86,300 results (0.47 seconds)

[OpenStack Docs: Manage projects, users, and roles](#)
 docs.openstack.org/administration/manage_projects_users_and_roles.html
 List projects; Create a project; Update a project; Delete a project. **Users.** **List users;** Create a **user**; Update a **user**; Delete a **user**. Roles and role assignments.
 Labeled **Docs**

[User Stories » OpenStack Open Source Cloud Computing Software](#)
 https://www.openstack.org/user-stories/
 openstack Thousands of **users** around the world run on **OpenStack**. Discover why organizations adopt **OpenStack**, what workloads they are running and the impact it has ...

[OpenStack Docs: OpenStack command-line interface cheat sheet](#)
 docs.openstack.org/user-guide/cli-cheat-sheet.html
 Here is a list of common commands for reference. ... **openstack user list** ... --user- data FILE INSTANCE \$ openstack server create --user-data userdata.txt --image ...
 Labeled **Docs**

[lists.openstack.org Mailing Lists](#)
 lists.openstack.org/
 Click on a **list** name to get more information about the **list**, or to subscribe, unsubscribe, and change ... **openstack-de**, **List** for German-speaking **OpenStack users**.

- e. At the **OpenStack Docs: Manage projects, users and roles** page, scroll down to find the information about listing users.

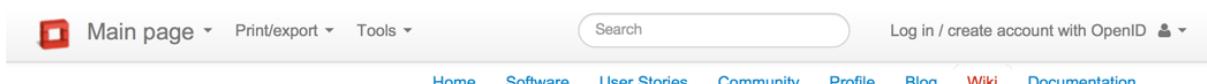
Users

List users

List all users:

```
$ openstack user list
+-----+-----+
| ID      | Name   |
+-----+-----+
| 352b37f5c89144d4ad0534139266d51f | admin   |
| 86c0de739bcb4802b8dc786921355813 | demo    |
| 32ec34aae8ea432e8af560a1cec0e881 | glance  |
| 7047fc7908e420cb36e13bbd72c972c | nova    |
+-----+-----+
```

16. Access the URL at <https://wiki.openstack.org> and view the OpenStack® wiki resources that are available.



The screenshot shows the OpenStack Wiki homepage. The navigation bar includes links for Main page, Print/export, Tools, Search, Log in / create account with OpenID, Home, Software, User Stories, Community, Profile, Blog, **Wiki**, and Documentation. The main content area features the OpenStack logo and the mission statement: "The OpenStack Mission: to produce the ubiquitous Open Source Cloud Computing platform that will meet the needs of public and private clouds regardless of size, by being simple to implement and massively scalable." It also states that OpenStack is open source, openly designed, and openly developed by an open community. Below this, there are three columns: "Get started" with a list of links; "Contribute to OpenStack" with a list of links; and "Join the Community" with a list of links. A sidebar on the left shows a 3D cube icon and a link to the official list of OpenStack project teams.

Please tell your instructor you have completed Lab 1.

Fundamentals of OpenStack® Technology

Lab 2—OpenStack® Management Tools

Objectives

After completing this lab, you should be able to:

- Use some of the more common OpenStack® CLI commands
- Use the Firefox REST client add-on to send the OpenStack® API requests to the OpenStack® services and review the responses
- Use the cURL command to send API requests to the OpenStack® services and review the responses
- Access and use the OpenStack® Horizon (dashboard) UI

Requirements

- Completion of Lab 1
- Access to the HPE Virtual Lab equipment for completing the OpenStack® labs
- PuTTY SSH client installed on your landing server's desktop
- Notepad ++ or equivalent editor installed on your system
- Firefox browser on your system with the RESTED rest client add-on installed
- Approximately 90 minutes to complete

NOTE: Notepad++ is a free programmer editor that can be obtained from <http://notepad-plus-plus.org>.

Exercise 1—Exploring OpenStack® CLI commands

Several different methods are available for implementing the OpenStack® commands, including:

- OpenStack® CLI commands
- REST client plug-in for browsers
- cURL commands
- OpenStack® Horizon (dashboard) UI
- Python commands

In this lab, you will have the opportunity to use the OpenStack® CLI commands, cURL commands, RESTed client in Firefox and the OpenStack® Horizon UI.

The DevStack environment comes with two preconfigured OpenStack® accounts:

- **admin/admin**—For administrative purposes
- **demo/demo**—For operations performed by unprivileged users

At the beginning, you will learn how to change the identity using the command line interface. Use the PUTTY session connected to the DevStack VM (with credentials of **stack/hpinvent**) to perform the steps below:

1. Make sure you are in the **devstack** directory by typing the following command and pressing **Enter**:

```
$ cd ~/devstack
```

2. To change the identity to admin user for your running terminal session, execute the following command in the terminal window and press **Enter**:

```
$ source ~/devstack/openrc admin admin
```

TIP: OpenStack® CLI is undergoing a transition between two different CLI command structures:

- OpenStack® service-specific commands are associated with an OpenStack® service (such as Keystone or Nova) and will begin with the name of that service.
- OpenStackClient (OSC) brings the command set for the OpenStack® services of Nova, Keystone, Glance, Swift, and Cinder APIs together in a single shell with a uniform command structure. All OSC commands begin with `openstack` followed by subcommands and command parameters.

Either of these methods will work, but the OSC will be the command structure used going forward, and the OpenStack® service-specific commands for Nova, Keystone, Glance, Swift, and Cinder will be deprecated in the future. In this lab, you will generally use the OSC commands when available.

NOTE: The `source` command might return the following message: `WARNING: setting legacy OS_TENANT_NAME to support cli tools.` You can ignore the message and proceed with your labs.

3. Access the *OpenStack Command-Line Interface Reference* document and answer the following questions.
 - a. **Q:** With which command do the OpenStack® service-specific commands for the OpenStack® Identity service begin?
 - b. **Q.** With which command do the OpenStack® service-specific commands for the OpenStack® Compute service begin?
 - c. **Q.** With which command do the OpenStack® service-specific commands for the OpenStack® Network service begin?
4. Each OpenStack® service has its own set of commands, depending on the respective service version of API. API versioning allows adding and deleting commands, as well as modifying their syntax to accommodate new features. For example, the OpenStack® Keystone service has two API versions:
 - API version 2, which is in the SUPPORTED state
 - API version 3, which is in the CURRENT state

A SUPPORTED API version is supported, but is not the latest version, and a CURRENT API version has all of the latest release features. An API version that is CURRENT is backward compatible with a SUPPORTED version, but a SUPPORTED version is not generally compatible with a CURRENT version.

You can verify the API versions of the various OpenStack® services at
<http://developer.openstack.org/api-ref.html>.

5. Access the *OpenStack Command-Line Interface Reference* document again and answer the following questions.
 - a. **Q:** Regions are a relatively new feature of the Keystone service. Is the command for creating a region available from the Identity service (Keystone)?
 - b. **Q:** What is the Keystone API version that provides the command for creating a region (hint: OpenStack® command-line client)?
 - c. **Q:** Does the command for creating a region begin with `keystone` or `openstack`?
 - d. **Q:** Are the API version 3-specific features of Keystone service-specific commands or OSC commands?

TIP: You will be using the *OpenStack Command-Line Interface Reference* document often in these labs, so you might want to spend a few minutes reviewing how the content is laid out.

6. **Q:** In looking at the *OpenStack Command-Line Interface Reference* document, what is the syntax for displaying information about a particular command?

7. At the command line, enter the following command: `openstack -h | more`. Notice that you do not receive the deprecation warning.

At this point, only the Keystone, Nova, Neutron, Glance, Cinder, and Swift services have the OSC functionality. Other OpenStack® services still use their service-specific command structure.

The following command, which will be discussed in the next lab, displays a list of users.

8. At the command line, enter the following command: `openstack user list`

A large amount of processing occurs between entering your command request and its execution.

By adding the `--debug` option, you will be able to see what happens during this processing.

9. At the command line, enter the following command: `openstack --debug user list`

A request is sent out to the Keystone service, which maintains the list of services. Keystone verifies that you have the proper access rights and sends out the information requested.

In the next few exercises, you will be provided with tools that will enable you to better understand what is happening during this process.

Before proceeding to the next exercise, you will need to capture some information that will be required in that exercise. The following terms will be discussed in the Keystone module and used in the associated lab. At this point, these terms will be defined at a very basic level so that you have some degree of understanding of what is being collected:

- **User**: Someone or something that can gain access to an OpenStack® service. The OpenStack® Keystone (Identity) service maintains a list of users and the rights they have to perform actions in the OpenStack® services. Any request for an OpenStack® service by a user must be verified and provided a security token that is used for any requests to that OpenStack® service.
- **Project** (also called **tenant**): A collection of OpenStack® resources (e.g. servers, storage) that is available to users associated with that project.
- **Endpoint URI**: The URL of a specific OpenStack® service. When requesting an OpenStack® service, you must specify the destination of that service by specifying its endpoint URI.
- **JSON** (JavaScript Object Notation): A data-interchange format standard that is supported by OpenStack®. It dictates the format of how data that is exchanged between two endpoints is formatted, so that a source endpoint knows how to write the data in a manner that can be read by the destination endpoint.

As you run the following CLI commands to collect information, you will need to copy that information to a notepad.

10. Open the editor on your landing system's desktop and type in the information as shown below.

```

1  Value obtained for lab 2
2
3  admin project ID:
4
5  admin user ID:
6
7  keystone publicURL:
8
9
10 keystoneSecTokenID_A:
11
12 curl statement to generate token:
13
14
15
16
17
18 KeystoneSecTokenID_B:
19
20 curl statement to display projects:
21
22

```

11. Complete the following steps to obtain the ID value of the admin project of your DevStack:

- a. Type the following command at the CLI of your terminal session to view the projects that exist on your DevStack:

```
openstack project list
```

- b. Note the highlighted ID value associated with **admin**.

ID	Name
1ae454c929b240a8b062e705e4bf4b6b	alt_demo
22ed21c7888940f38001b1b0cd8c5d92	admin
402eb90dd6de456abb672f4ec8f0b468	invisible_to_admin
486d83dcc07d4c108fb87628d2d99e4	swiftprojecttest1
6b514ed76ca1410faf2436794413e3e	swiftprojecttest4
7d259bc4d7504e359f54c54ad7d195d4	project_a
8ed05782e6084a55bc35eaeee3afb590	service
e3a47086ec6d479fb1715a3da1e2c4d9	demo
f7d080b392db4b5fb35f08217323d2b8	project_b
fed82a90b39c43b894a162ba70637051	swiftprojecttest2

NOTE: The OpenStack® client will display the following warning message:

WARNING: openstackclient.common.utils is deprecated and will be removed after Jun 2017. Please use osc_lib.utils

This message is made for developers and you can safely ignore it.

- c. Paste the highlighted value in the editor located to the right of admin project ID:

12. Complete the following steps to obtain the ID value of the **admin** user of your DevStack.

- Type the following command at the CLI of your terminal session to view the users that exist on your DevStack:

```
openstack user list
```

- Highlight the ID value associated with admin.

ID	Name
0a293210b07a43a88757c54faf9cad5b	project_a_creator
10f01b26921144498734d45db2163850	service-admin
158ba4a803fa4d3e93f2d277cce9fb8	demo
2636a455d0a244f3a679409aa67f80b4	swiftusertest1
4a61050957d044489cefdfacbebf44e9	swiftusertest2
51008ef007b34f5ba12d038518815316	neutron
528422fcef2f4245b1733763e0f788e9	project_b_auditor
5714c2cf75fc4fd39cb6c86687108ab3	heat
5a2b57724e9d43cba1a94a3f19ee029e	designate
6c55f476ba5045ecafa9d45e2e0daa60	heat_domain_admin
6e8fa76b62ed473c83949d5b85771115	swiftusertest3
808b50e9222e405d9ae2db6f60a24748	admin
9007f4c5edcb475f84eef4279a9f62ff	swift
9aacfa8dd6bf4d2fa0b4624ff2541d1c	swiftusertest4
a1860310c2e7403f92d57f64d48e140e	project_b_observer
adaeae02bbdb84bcd864d15d035fab830	glance-swift
ae3cf5975fef496aa6833e7d927feaff	nova
b83cd3b545a54d5e8ff04ff0fce7307	barbican
bc8431de17fd4c9c807b8a42d06251d7	project_a_admin
be0fce362044f53a3d5fe0116b49325	project_b_creator
c311d9e0ff5e487bb53f879e70769335	project_a_observer
ca7ea31a61d64cbb8f03bf03fa35b12e	glance
ceac80b994cd4a878a164b4498311250	project_b_admin
d1598eb6f603476786d4ffb7a30472c4	project_a_auditor
dea0f35997104ed0bcfc88ea3d9b742c	alt_demo
e09d59d8010141a19b8d0886c10d5ddb	project_a_creator_2
f07c65b9a0c74718a8d5af2a9bd425a4	cinder

- Paste the highlighted value in the editor, to the right of `admin` user ID:

13. Complete the following steps to obtain the public URL (URI) of the Keystone project of your DevStack:

- Type the following command at the CLI of your terminal session to view the URLs of the OpenStack® services installed on your DevStack:

```
openstack endpoint list
```

- b. To find the URL of the public Keystone service endpoint, run the following command:

```
stack@stack:~$ openstack endpoint list --long \  
-c "Service Name" -c PublicURL | grep keystone  
| keystone      | http://192.168.5.4/identity |
```

- c. Paste the highlighted value in the editor, to the right of the `keystone publicURL`:

- d. **Q:** Which version of the Keystone API is being used? (Hint: Check the documentation for current and supported Keystone API versions.)

You now have the information required to begin the exercise. Save your file, but keep it open. In the following labs, you will use the name instead of the collected ID information, as this is supported by the Keystone v3.

Exercise 2—Using a REST client browser plug-in to generate API requests to the OpenStack® services

A browser plug-in-based REST client is a very useful tool for understanding how the RESTful APIs used for communications between a client and an OpenStack® service operate. The RESTED add-on for Firefox and Google's Advanced Rest Client are a couple of examples of these tools. In this lab, you will use the Firefox RESTED plug-in.

Generate a Keystone security token

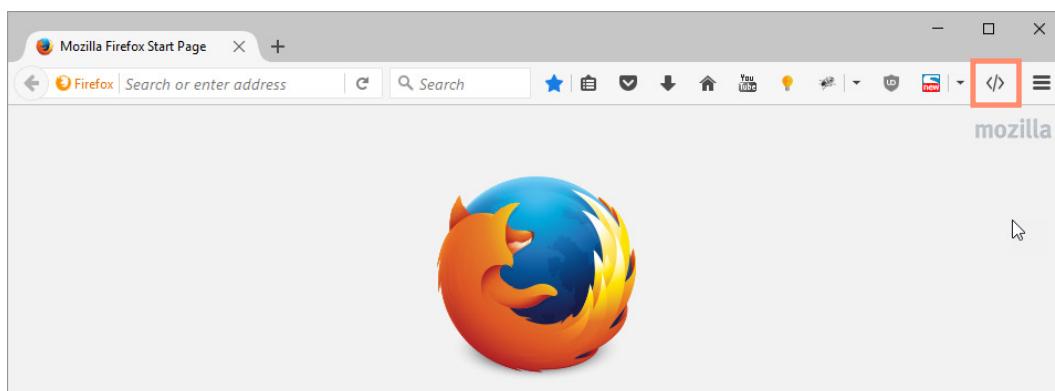
A security token is required when making any API request to an OpenStack® service. In fact, a large part of the output from your `openstack -debug user list` command from the previous exercise was the generation of a security token.

Any time you want to request an action from an OpenStack® service, you must provide a security token. Security tokens will be discussed in more detail in the Keystone module and lab.

Clients obtain this token, along with the URL to other service APIs, by first authenticating against the Identity Service (Keystone) with valid credentials.

After a security token is obtained, it can be used to authenticate all OpenStack® service requests.

1. Open the Firefox browser. From the top menu bar, click the **RESTED icon** of your Firefox browser toolbar.



The RESTED screen appears. All fields are empty, except the Method field. This screen provides a method for entering the information required to submit a RESTful API request to an OpenStack® service and view the information returned by the server.

You need the URL of the OpenStack® service to where you are sending the request. Keystone is responsible for all identity functions for OpenStack®, including generating and managing security tokens.

2. Access the OpenStack® API reference web page available at <http://developer.openstack.org/api-guide/quick-start/>.

3. At the left-hand side, select **Current API versions** and then click **Identity API v3**.

The screenshot shows the 'Current API versions' page. On the left, there's a sidebar with links to 'Deprecated API versions' and 'API quick-start examples'. Below that is a list of API links: 'Bare Metal API v1 (microversions)', 'Block Storage API v3 (microversions)', 'Clustering API v1', 'Compute API (microversions)', 'Container Infrastructure Management API (microversions)', 'Data Processing v1.1', 'Data Protection Orchestration v1', 'Database Service API v1.0', 'Domain Name Server (DNS) API v2', 'Identity API v3' (which is highlighted with a red box), 'Identity API v3 extensions', and 'Image service API v2'.

4. From the Contents list at the left-hand side, in the **Authentication and token management** section, click **Password authentication with scoped authorization**.

The screenshot shows the 'Identity API v3 (CURRENT)' page. The left sidebar has sections for 'Identity API v3 (CURRENT)' (with links to 'What's New in Version 3.8', '3.7', '3.6', '3.5', '3.4', '3.3', '3.2', '3.1', '3.0') and 'Authentication and token management' (with links to 'Password authentication with unscoped authorization', 'Password authentication with scoped authorization' (which is highlighted with a red arrow), 'Password authentication with explicit unscoped authorization', 'Token authentication with unscoped authorization', and 'Token authentication with scoped'). The main content area discusses authentication tokens and policy rules, and provides a link to the 'Identity API protection with role-based access control (RBAC)' guide.

5. At the top of the page, locate the **Authenticate for admin API** section. Note the **POST** method, and the URL of [/v3/auth/tokens](#). When ready, click the **detail** button.

The screenshot shows a portion of the OpenStack API documentation. On the left, there is a green button labeled "POST". To its right is the URL "/v3/auth/tokens". Below the URL, the text "Password authentication with scoped authorization" is displayed. In the top right corner of the screenshot area, there is a blue button with the word "detail" in white text.

6. Scroll down to the Request Example section and copy the example to the clipboard.

Request Example

```
{
  "auth": {
    "identity": {
      "methods": [
        "password"
      ],
      "password": {
        "user": {
          "id": "ee4dfb6e5540447cb3741905149d9b6e",
          "password": "devstacker"
        }
      }
    },
    "scope": {
      "project": {
        "id": "a6944d763bf64ee6a275f1263fae0352"
      }
    }
  }
}
```

7. Open the REST client in the Firefox browser and change the Method field to select the method you have identified in the previous step.

</> RESTED

The screenshot shows the RESTED extension interface in the Firefox browser. On the left, there is a "Collections" tab and a "History" tab. Below them is a note: "No collected requests. Add by pressing \"plus\" in the top right of the request panel." On the right, there is a "Request" panel. It contains fields for "URL" (set to "http://www.arngren.net/"), "Method" (set to "POST"), and "Headers" (with a small arrow indicating expandable content). The "Method" field is highlighted with a red border.

Notice that the URI (also called URL) listed in the table does not match the URL value in your editor. As mentioned in the note following the table, you must know its URL (endpoint). So, the value of the URL will be the value of the URL, which is what you pasted into editor, plus v3/auth/tokens.

8. Copy all the content in the grayed area of the example into the Body field of the RESTED screen.

9. Replace the devstacker value associated with the password of hpinvent and create the header "Content-Type application/json". Also, replace the ID values in the Project and User sections with values collected in the Exercise 1, steps 11 and 12.

Verify that your REST client screen now looks like the following screenshot. Click the **Send request** button.

The screenshot shows a REST client interface with two main panes. On the left, under 'Collections', there are two entries: 'Collection' and another 'Collection' with a POST method selected. The URL for the POST method is `http://192.168.5.4/identity/v3/auth/tokens`. On the right, the 'Request' pane displays the configuration for this POST request. The 'URL' field is set to `http://192.168.5.4/identity/v3/auth/tokens`, the 'Method' is set to 'POST', and the 'Content-Type' header is set to 'application/json'. Below these, the 'Request body' section contains a JSON payload:

```

{
    "password": "hpinvent",
    "username": "hpinvent",
    "tenant_id": "4a9825767ab91c7",
    "tenant_name": "admin",
    "auth": {
        "scope": {
            "domain": {
                "id": "default"
            }
        }
    }
}
  
```

10. In the Response section of the screen, expand the Headers subsection and verify you have a 201 created.
 11. Expand the Headers body subsection and locate the **x-Subject-Token** token information.

The screenshot shows the 'Response - http://192.168.5.4/identity/v3/auth/tokens' screen. At the top, it displays a '201 Created' status. Below this, the 'Headers' section is expanded, showing the following header information:

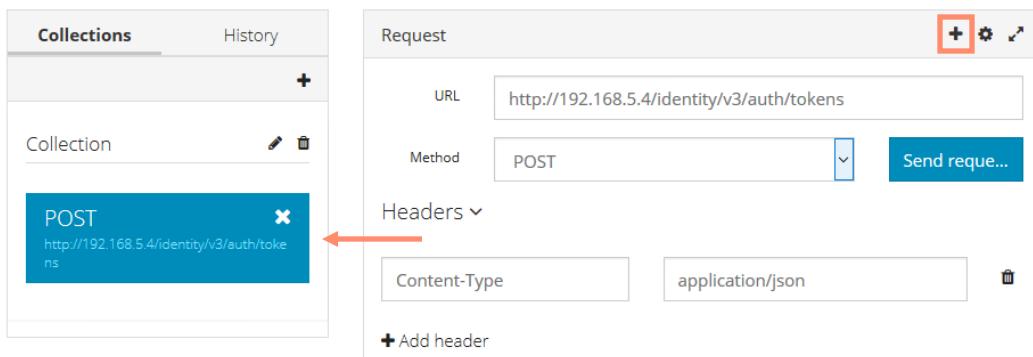
Date: Wed, 21 Dec 2016 19:55:52 GMT
Server: Apache/2.4.18 (Ubuntu)
X-Subject-Token: 01ee8a0a38d84194a989825767ab91c7
Vary: X-Auth-Token
x-openstack-request-id: req-3967e2cc-c75d-4700-9a87-e69287e7962f
Content-Length: 4921
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: application/json

12. Q: How long is the token valid?

13. Copy the token and paste it next to the `keystoneSecTokenID` A entry in the editor.

14. To save this request:

- Click + in the Collections section of the screen, click **the pencil icon**, and name the collection “Generate Security Token”
- Click the + sign in the top right of the Request header, which adds the request to the selected collection



15. Close the RESTED browser tab.

Using RESTED to request a list of projects

Now that you have a security token, you can use it to access other types of OpenStack® information, such as a listing of the existing projects.

1. Back at the API reference page, in the Users section, click **List users**.

Users

- [List users](#) **←**
- Create user
- Show user details
- Update user
- Delete user
- List groups to which a user belongs
- List projects for user
- Change password for user

Based on your user ID, [GET /users/{user_id}/projects](#)

- Tokens explicitly represent user+project or user+domain pairs
- Partial updates are performed using the HTTP [PATCH](#) method
- Token ID values no longer appear in URLs

This page lists the Identity API operations in the following order:

- [Authentication and token management](#)
- [Credentials](#)
- [Domains](#)
- [Domain configuration](#)
- [Groups](#)

2. In the Users section, make note of the HTTP method, and the path used for the **List users** API call.

3. Q: Which method is used to gather a list of tenants (projects)?

4. Q: What is the URI that will be used?

5. Click the **RESTED icon** of the Firefox browser to display a new request form.
6. Complete the form as shown below, using the value of the **X-Subject-Token** collected in the Exercise 2, step 11 of this lab.

NOTE: The Token ID value required by this form was obtained by the POST request in the previous lab and should have been pasted to your editor.

Request

URL: http://192.168.5.4/identity/v3/projects

Method: GET

Send request

Headers

X-Auth-Token: cbd02479fab0453d99e1b703a7ace8c!

Add header

Basic auth

Username: _____ Password: _____ Show password?

7. Click **Send request**. Verify that the Response Headers in the Response section of the Rested screen are showing a status of 200 OK. In the Response body field, examine the returned information.

Response - http://192.168.5.4/identity/v3/

200 OK

Headers >

Response body

```
{
  "version": {
    "status": "stable",
    "updated": "2016-10-06T00:00:00Z",
    "media-types": [
      {
        "base": "application/json",
        "type": "application/vnd.openstack.identity-v3+json"
      }
    ]
  }
}
```

8. **Q:** What are the names of the tenants (projects) returned by the request?

9. Save the GET request in RESTED using the procedure described in step 14 in the previous section of this lab guide.
10. Close all the browser windows on your landing VM desktop.

You should now be able to take information from the *OpenStack API Reference* document, apply it to the REST client, send a request, verify that the request completed successfully, and view the response information.

Exercise 3—Using cURL to manage OpenStack®

Developers and testers commonly use cURL, the command line tool from <http://curl.haxx.se/>. With cURL you can send HTTP requests and receive responses back from the command line. This is useful for seeing how requests are constructed and how information is returned from OpenStack®.

NOTE: You will need the *API Examples using cURL* guide located at http://docs.openstack.org/developer/keystone/devref/api_curl_examples.html for this portion of the lab.

Using cURL to generate a Keystone security token

In exercise 2 you generated a Keystone security token with the REST client browser plug-in. In this section you will use cURL to generate the security token. The benefit of understanding the REST client first is that you now understand where to get the required information from the *OpenStack API Complete Reference* document and will be able to recognize how the information can be applied to the cURL format. cURL enables you to do everything you did in the REST client from the command line.

You will need your editor to modify the cURL statements.

1. Open the *API Examples using cURL*, located at http://docs.openstack.org/developer/keystone/devref/api_curl_examples.html.
2. Scroll down the page to the *domain-scoped* example.

Domain-Scoped

Get a domain-scoped token (Note that you're going to need a role-assignment on the domain first!):

```
curl -i \
      -H "Content-Type: application/json" \
      -d '{
        "auth": {
          "identity": {
            "methods": ["password"],
            "password": {
              "user": {
                "name": "admin",
                "domain": { "id": "default" },
                "password": "adminpwd"
              }
            }
          },
          "scope": {
            "domain": {
              "id": "default"
            }
          }
        }
      }' \
      http://localhost:5000/v3/auth/tokens ; echo
```

3. Copy the domain-scoped cURL command example. To perform the cURL request using the Keystone v3, in the PuTTY terminal session connected to DevStack, enter the command below:

NOTE: Use the editor to paste the whole domain-scoped example, and then make the changes to the password and the URL to match the example below. Once you are ready, copy the command from the editor and paste into the terminal.

```
curl -i \
  -H "Content-Type: application/json" \
  -d '{
    "auth": {
      "identity": {
        "methods": ["password"],
        "password": {
          "user": {
            "name": "admin",
            "domain": { "id": "default" },
            "password": "hpinvent"
          }
        }
      },
      "scope": {
        "domain": {
          "id": "default"
        }
      }
    }
  }' http://192.168.5.4:5000/v3/auth/tokens ; echo
```

4. Examine the output of the command.
5. Use the `curl --help | less` command to find more information about parameters used with the `curl` command.
6. To get the token using the command line, run the command below. At the end of the line, you will filter out the output so that you extract only the X-Subnet-Token by using the `| grep X-S:`

```
curl -si -H "Content-Type: application/json" -d '{
    "auth": {
      "identity": {
        "methods": ["password"],
        "password": {
          "user": {
            "name": "admin",
            "domain": { "id": "default" },
            "password": "hpinvent"
          }
        }
      },
      "scope": {
        "domain": {
          "id": "default"
        }
      }
    }
  }' http://192.168.5.4:5000/v3/auth/tokens | grep X-S:
```

7. Copy the token and paste it next to the `keystoneSecTokenID` B entry in the editor.

Using cURL to display a list of projects

Now that you have obtained your security token, you can use it to submit requests to the service, which in your case is Keystone, and receive a response from the service. As with the RESTED, you will use the security token to obtain a list of projects.

1. On your local desktop, use the browser and navigate to <http://developer.openstack.org/api-ref/identity/v3/>.

At the left-hand side, in the Projects section, navigate to **List projects**. Note that this is a much simpler cURL statement because, as noted in the request documentation, you only need to specify a few parameters:

- Method : GET
- URI : /v3/projects
- Header: X-Auth-Token with the token ID

2. Copy the command below to your editor and modify the highlighted parameters to match your environment:

```
curl -s -X GET http://192.168.5.4/identity/v3/projects -H "X-Auth-  
Token:821b274712ad4f098c54c7641c236dc9" | python -m json.tool | grep name
```

3. Copy the corrected cURL from the editor to the command line of the terminal session and press **Enter**.

4. Verify the response of the cURL command.

Exercise 4—Managing OpenStack® from the Horizon GUI

Prerequisites

- Access to the HPEVL landing desktop
- Firefox web browser installed
- The login credentials for the lab VM

Horizon is the canonical implementation of the OpenStack® dashboard, which provides a web-based user interface to OpenStack® services including Nova, Swift, and Keystone.

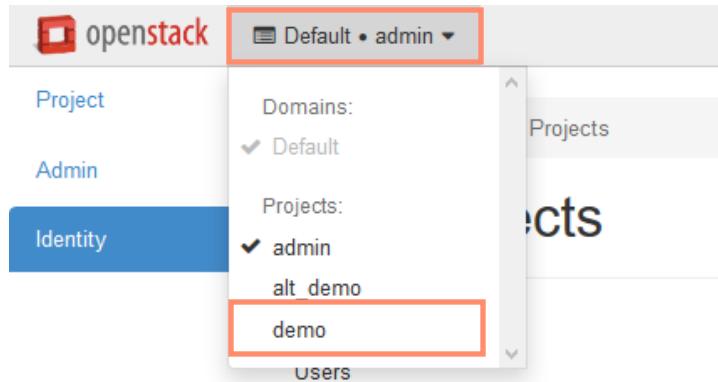
To access the lab environment using the Horizon GUI:

1. Start the Firefox web browser on your landing desktop.
2. Point the browser to the IP address of the lab VM (<http://stack.vlabs.net>).
3. Enter default as domain, admin and hpinvent as user name and password, and click **Connect**.

4. The Horizon window is divided into three main areas:
 - The **Settings** bar at the top—Providing project selection, user preferences, and sign-out controls
 - The **Navigation** bar on the left-hand side—Providing the navigation tree to access available options
 - **Content area**—Showing the content selected at the Navigation pane

Name	Description	Project ID	Domain Name	Enabled	Actions
swiftprojecttest1		22690382a2564837be7587f8d42a08b8	Default	Yes	<button>Manage Members</button>

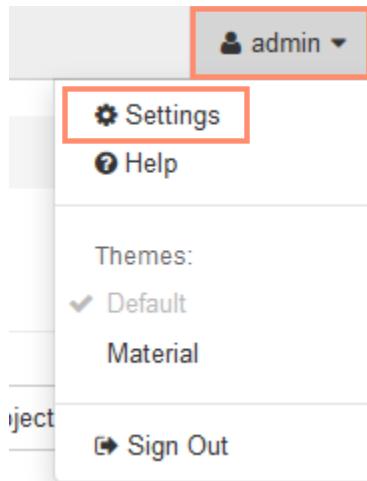
5. In the Settings pane, click the **project selector menu** on the left-hand side. From the list of projects, select **demo**.



6. The name of the selected demo project is displayed on the navigation bar.



7. In the top right-hand corner of the UI, click the **admin** drop-down list and click the **Settings** option.



8. Click various options, including the **question mark icons** (ⓘ) to see the information provided, but do not change the current settings.

User Settings

The screenshot shows the 'User Settings' page. It includes fields for 'Language' (British English), 'Timezone' (UTC), 'Items Per Page' (20), and 'Log Lines Per Instance' (35). To the right, there's a 'Description' section with a note about modifying dashboard settings. Question mark icons (ⓘ) are located next to the 'Timezone', 'Items Per Page', and 'Log Lines Per Instance' fields.

9. Click the **Help** option from the Admin drop-down list, and notice that it opens a new browser window.
10. **Q:** What is the URL of the page you were directed to? (this is an isolated environment, and you will get a server not found error)

11. At the left-hand side, click the **Project → Compute → Overview** option in the navigation panel of the UI and note the information provided on this screen.
12. Take some time to examine other options in the navigation panel.
13. Note that the dashboard supports changing the look of the user interface by switching the theme from the **admin** drop-down list in the top right-hand corner. You will be using the default theme for this lab guide.
14. When you are done browsing, in the top right corner of the Horizon screen, click **Admin → Sign Out**.

The screenshot shows the OpenStack Horizon dashboard for the 'Compute' section. The 'Overview' tab is active. In the top right, there's a sidebar for the user 'admin'. The 'Themes' section shows 'Default' is selected. A red box highlights the 'Sign Out' button. Below the sidebar, the 'Limit Summary' section displays resource usage with pie charts:

Resource	Used	Total
Instances	0 of 10	
VCPUs	0 of 20	
RAM	0Bytes of 50GB	
Security Groups	1 of 10	
Floating IPs	0 of 50	
Volumes	0 of 10	
Volume Storage	0Bytes of 1000GB	

15. Close the browser session.
16. Please tell your instructor you have completed Lab 2.

Fundamentals of OpenStack® Technology

Lab 3—OpenStack® Keystone (Identity)

Objectives

Keystone is an OpenStack® service that provides Identity, Token, Catalog, and Policy services to be used specifically by the projects in the OpenStack® family. It implements the OpenStack® Identity API.

After completing this lab, you should be able to use the CLI, the Horizon UI, or both to:

- List and add tenants
- List and add users
- Associate users to tenants
- View the service catalog

Requirements

- Completion of Lab 2
- Access to the HPE Virtual Lab equipment for completing the OpenStack® labs
- Access to the Identity Management section of the *OpenStack Administration Guide* at <http://docs.openstack.org/admin-guide/identity-management.html>
- Access to the *OpenStack Client* documentation at <http://docs.openstack.org/developer/python-openstackclient/>
- Approximately 45 minutes to complete

Exercise 1—Keystone identity functionality

The Identity service provides authentication credentials validation and data about users, tenants and roles, as well as any associated metadata.

In the basic case, all this data is managed by the service, allowing the service to manage all the Create, Read, Update, and Delete (CRUD) associated with the data.

In other cases, this data is pulled from an authoritative back-end service by varying degrees. An example of this would be using the LDAP backend.

1. If you are not already logged in to your assigned VM, use PuTTY to access your VM and log in with the student and hpinvent credentials.

2. To set the necessary environment variables for the command-line tools, enter the command:

```
source ~/devstack/openrc admin admin
```

3. At the command line, run the command:

```
env | grep OS_
```

4. **Q:** What API version is used by the Keystone in your environment?

5. You can display help for identity functions by entering the following command:

```
openstack help | more.
```

6. You can get help for individual commands by entering `openstack help <command>`, as shown in the following screenshot.

```
stack@stack:~/devstack$ openstack help project list
WARNING: openstackclient.common.utils is deprecated and will be removed after Jun 2017
. Please use osc_lib.utils
usage: openstack project list [-h] [-f {csv,json,table,value,yaml}]
                               [-c COLUMN] [--max-width <integer>] [--noindent]
                               [--quote {all,minimal,none,nonnumeric}]
                               [--domain <domain>] [--user <user>] [--long]

List projects

optional arguments:
  -h, --help            show this help message and exit
  --domain <domain>    Filter projects by <domain> (name or ID)
  --user <user>         Filter projects by <user> (name or ID)
  --long               List additional fields in output

output formatters:
  output formatter options

  -f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}
                                         the output format, defaults to table
  -c COLUMN, --column COLUMN
                                         specify the column(s) to include, can be repeated
```

A **project** is the group of cloud resources and users. A project is likely a company with the users being the entities that use OpenStack® within that company.

7. Enter the `openstack project list` command to display a list of the existing projects.

```
stack@stack:~/devstack$ openstack project list
WARNING: openstackclient.common.utils is deprecated and wi
. Please use osc_lib.utils
+-----+-----+
| ID | Name |
+-----+-----+
| 002bfccbcc2a49198d3e2dbb15d504c3 | service |
| 43646418f2434c28aa52d0d7b76204e0 | swiftprojecttest4 |
| 5ccff27a3f404aa4a112bcad23182301 | swiftprojecttest2 |
| 6195d9924f0a41ec997263b19c0a6467 | invisible_to_admin |
| 80f94d258ce743a48e6f85843ca8f078 | swiftprojecttest1 |
| 8a734d8a67f147caaac0d744dbd307b5 | alt_demo |
| 8a84a6dc091241d3a1d4d22bb68d5e32 | admin |
| dc8b28152df94ba0b10a057a8ecd496f | demo |
+-----+-----+
stack@stack:~/devstack$
```

NOTE: When referencing an OpenStack® entity, such as a project, the best practice is to use its ID and not its name because the ID value is unique, while the name is not unique. Consider, for example, whether clouds were merged and whether there were projects with the same name.

8. Enter the `openstack help` command to determine how to create a project with the following attributes. Then, create that project.

Field	Value
description	My first OpenStack Project
enabled	True
id	081ce09d83744e32ba73a4244b674b88
name	project4labs

9. **Q:** Which CLI command did you use to create the project with the attributes shown in the above screenshot?

A **user** is a digital representation of a person, a system, or a service that uses the OpenStack® cloud services. Identity authentication services validate that incoming requests are being made by the user that claims to be making the call. Users have a login and can be assigned tokens to access resources. Users can be directly assigned to a particular project and behave as if they are contained in that project.

10. Enter the `openstack user list` command to display a list of users.

```
stack@stack:~/devstack$ openstack user list
WARNING: openstackclient.common.utils is deprecated and
. Please use osc_lib.utils
+-----+-----+
| ID      | Name   |
+-----+-----+
| 08eed2de4219407ba77972929ae5e33d | swiftusertest3 |
| 2b89cc46894f4f688e6efa47e4e228e2 | glance |
| 3673eb16232b48dbabd2f28199532697 | heat |
| 3c55e8ce3f6040a99321dc81e25bd28c | swift |
| 6077014c94ba49a9bd03fc53a9687828 | neutron |
| 71a54564f953497d88c893271b3a98e6 | alt_demo |
| 7d75f2e9371e4df497cecafaf6fe332a | nova |
| 8175d86d461047ca8feaef98a0ed646 | swiftusertest4 |
| 83e92225b21b4987b516649fe5170db2 | swiftusertest1 |
| 8527bddf02a04db0bbdff6562e2a43ec | cinder |
| 85a71e6bf7e43dd965bd8a30a40e146 | admin |
| 8c254c96f51146fda801061b47c77dc8 | heat_domain_admin |
| 9df69a002d3745ca9655cec28fea6ee6 | glance-swift |
| a8da1d8212c74501a3a88c971d280259 | designate |
| daae67a03e8e441bbe950ba0e9fe1d95 | demo |
| f1318e7ec8a4435885c4dbd134bb245e | swiftusertest2 |
+-----+-----+
```

NOTE: When referencing an OpenStack® entity, such as a project or a user, the best practice is to use its ID and not its name because the ID value is unique, while the name is not unique.

NOTE: User names have been assigned for each of the services for communications between the services. The `swiftusertest` users were created by DevStack and would not be generated by a standard OpenStack® install.

11. By entering the `openstack help` command, create a user with the following attributes:

- User name: `user4labs`
- User's email: `user4labs@vlabs.net`
- User's primary project: `project4labs`

12. **Q:** Which command did you use to create the above user?

13. A **role** in OpenStack® is a personality that a user assumes that enables them to perform a specific set of operations. A role includes a set of rights and privileges. A user assuming that role inherits those rights and privileges.

In the Identity service, a token that is issued to a user includes the list of roles that the user can assume. Services that are being called by that user determine how they interpret the set of roles that a user has and which operations or resources each role grants access to.

It is up to individual services such as the Compute service and the Image service to assign meaning to these roles. As far as the Keystone service is concerned, a role is simply a name.

The scope of a role in Keystone API v2 is limited to a project, so the role in one project might be totally different from the role in a different project in terms of the operations that can be performed. The exception to this rule is the role of admin, which is equivalent among all projects.

The scope of a role in Keystone API v3 changed because of the addition of **domains**. In a domain, the admin role is equivalent among projects within a domain, but can vary among domains.

14. Enter the `openstack role create role4labs` command to create a role called `role4labs`.

```
stack@stack:~/devstack$ openstack role create role4labs
WARNING: openstackclient.common.utils is deprecated and
. Please use osc_lib.utils
+-----+-----+
| Field      | Value
+-----+-----+
| domain_id  | None
| id          | 00e0d7a3ebb74f16a3cc5de920b597da
| name        | role4labs
+-----+
```

15. Leave the terminal session running in the background and proceed with the next exercise.

Exercise 2—Horizon (dashboard) identity functionality

In this exercise, you will verify the identity functionality using the dashboard. On the landing server, open the Firefox web browser and perform the steps below:

1. Connect to the URL of the DevStack VM <http://stack.vlabs.net>.
2. Enter default as domain, admin and hpinvent as a user name and password, and click **Connect**.

3. In the navigation bar at the left-hand side, verify that **Identity → Projects** is selected. On the list of projects, find the project that is created using the command line in the previous exercise.

	Name	Description	Project ID	Domain Name	Enabled	Actions
<input type="checkbox"/>	service		002bfccbcc2a49198d3e2dbb15d504c3	Default	Yes	<button>Manage Members</button>
<input type="checkbox"/>	swiftprojectt est4		43646418f2434c28aa52d0d7b76204e0	-	Yes	<button>Manage Members</button>
<input type="checkbox"/>	project4labs	My first OpenStack Project	4741abb2b8294be2b1a994490c294fbe	Default	Yes	<button>Manage Members</button>
<input type="checkbox"/>	swiftprojectt est2		5ccff27a3f404aa4a112bcad23182301	Default	Yes	<button>Manage Members</button>

4. In the right-most column of the screen, there is an **Actions** column that lets you modify the users associated with a project, as well as manage the deletion of the parameters of the project.

5. Verify that the **user4labs** user is a member of the **project4labs** tenant by clicking **Manage Members** associated with **project4labs**.

A screenshot of the OpenStack Keystone 'Manage Members' interface. At the top, there are several project details: 'project4labs', 'My first OpenStack Project', '4741abb2b8294be2b1a994490c294fbe', 'Default', 'Yes', and a 'Manage Members' button which is highlighted with a red box. Below this, there are two tabs: 'Project Information *' and 'Project Members'. The 'Project Members' tab is selected and highlighted with a blue box. On the left, there is a list of 'All Users' with names like 'glance', 'glance-swift', 'cinder', 'neutron', 'swift', 'swiftusertest1', 'swiftusertest3', and 'swiftusertest2'. Each user name has a blue '+' button to its right. On the right, there is a 'Project Members' section showing 'user4labs' with a 'Member' dropdown and a '-' button. The entire interface is contained within a modal window titled 'Edit Project'.

6. Assign the **user4labs** and **swiftusertest2** users to the **project4labs** by clicking the **+** button next to the name of the user in the All Users column.

A screenshot of the 'Edit Project' modal. The 'Project Members' tab is active. On the left, under 'All Users', there is a list of users: glance, glance-swift, cinder, neutron, swift, swiftusertest1, swiftusertest3, and swiftusertest2. Each user has a blue '+' button to their right. On the right, under 'Project Members', there is a list with one item: 'user4labs' with a 'Member' dropdown and a '-' button. The entire modal has a header 'Edit Project' and a close button 'x' at the top right.

7. Q: How do you know that the **swiftusertest2** user is now a member of **project4labs**?

8. Remove the **swiftusertest2** user from **project4labs**, by clicking the – button next to the name of that user in the Project Members column and select the **edit** icon for **user3labs**.

9. Click the member drop-down list associated with **user4labs** and notice you can add a user to roles.
 10. Add the **role4labs**, **_member_** and **Member** roles to user4labs.

11. Click **Save** to save the changes and close the Edit Project screen.
 12. From the navigation panel of the Horizon UI, click **Identity → Users**.

<input type="checkbox"/>	user4labs	-	user4labs@vlabs.net	af8731f735b443feb6d8fa45662e9bd3	Yes	Default	Edit ▾
<input type="checkbox"/>	demo	-	demo@example.com	daae67a03e8e441bbe950ba0e9fe1d95	Yes	Default	Edit ▾

13. Q: What is the primary project associated with **user4labs**?

14. Note the information that can be modified on the Update User page and click **Cancel**.

Domain ID

Domain Name

User Name *

Description

Email

Primary Project

project4labs

Description:
Edit the user's details, including the Primary Project.

Cancel
Update User

15. Click the drop-down icon associated with **user4labs** and click the **Change Password** option.

<input type="checkbox"/>	demo	-	demo@example.com	daae67a03e8e441bbe950ba0e9fe1d95	Yes	<div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;">Change Password</div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;">Disable User</div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 2px;">Delete User</div> <div style="border: 1px solid #ccc; padding: 2px; width: fit-content;">Default</div> <div style="border: 1px solid #ccc; padding: 2px; width: fit-content;">Edit</div>
<input type="checkbox"/>	swiftusertest	-	test2@example.com	f1318e7ec8a4435885c4dbd134bb245e	Yes	

16. Change the password for the user to **hpinvent** and click **Save**.

Change Password

Password *
hpinvent 

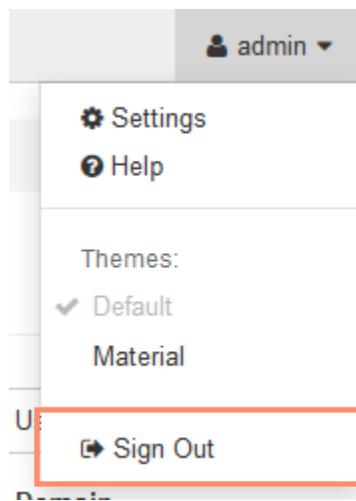
Confirm Password *
hpinvent 

User Name
user4labs

Description:
Change user's password. We highly recommend you create a strong one.

Cancel **Save** 

17. Click the **admin** link located at the top right of the browser screen and sign out.



18. Log in as **user4labs** with the password of **hpinvent**.

19. **Q:** Which project are you automatically logged in to, and why?

20. Click the **Sign Out** link located at the top right of the browser screen.

21. Log back in as **admin / hpinvent**.

22. Add the **admin** role to **user4labs** for the **project4labs** and then sign out.

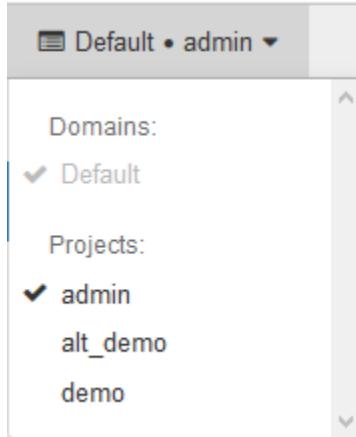
23. Log in to the Horizon screen as **user4labs** with the password of **hpinvent**.

24. **Q:** Which projects are accessible by **user4labs** (Hint: Click the project drop-down menu from the top of the screen).
25. Verify the Admin tab is available in the Horizon UI for **user4labs**.
26. Click the **Sign Out** link located at the top right of the browser screen.

Exercise 3—The OpenStack® service catalog and API access

In addition to providing identity services, the Keystone service also provides a catalog of the available OpenStack® services and the URIs for each of those services. The endpoint list provides the information necessary to access a specific endpoint, so when communicating a service through an API, you must specify both the endpoint ID to get to the endpoint and the ID of the service you want to communicate with.

1. Log in to the Horizon screen as **admin** with the password of `hpinvent`. Use the **default** domain.
2. In the Horizon settings bar, select the **admin** project from the list.



3. To display the service catalog, in the left navigation pane, expand **Admin** → **System** → **System Information**, and select the **Services** tab on the System Information screen.

	Name	Service	Region	Endpoints
Volumes	heat	orchestration	RegionOne	Admin http://192.168.5.4:8004/v1/7eb38352a129452da2cddcd9cd5cea12 Internal http://192.168.5.4:8004/v1/7eb38352a129452da2cddcd9cd5cea12 Public http://192.168.5.4:8004/v1/7eb38352a129452da2cddcd9cd5cea12
Flavours				
Images	heat-cfn	cloudformation	RegionOne	Admin http://192.168.5.4:8000/v1 Internal http://192.168.5.4:8000/v1 Public http://192.168.5.4:8000/v1
Networks				
Routers	designate	dns	RegionOne	Admin http://192.168.5.4:9001/ Internal http://192.168.5.4:9001/ Public http://192.168.5.4:9001/
Floating IPs				Admin http://192.168.5.4:8774/v2/7eb38352a129452da2cddcd9cd5cea12

The Services tab of the System Information screen shows all currently available OpenStack® services.

The Services tab on the System Information screen provides a list of available services and an endpoint URL for each of the services, similarly to the `openstack catalog list` command.

4. To display the API endpoints, in the navigation panel, expand **Project** → **Compute** → **Access & Security** and select the **API Access** tab.

Access & Security

Service	Service Endpoint
Orchestration	http://192.168.5.4:8004/v1/7eb38352a129452da2cddcd9cd5cea12
Cloudformation	http://192.168.5.4:8000/v1
Dns	http://192.168.5.4:9001/
Compute_Legacy	http://192.168.5.4:8774/v2/7eb38352a129452da2cddcd9cd5cea12
Compute	http://192.168.5.4:8774/v2.1/7eb38352a129452da2cddcd9cd5cea12
Identity	http://192.168.5.4/identity
Volumev3	http://192.168.5.4:8776/v3/7eb38352a129452da2cddcd9cd5cea12
Volumev2	http://192.168.5.4:8776/v2/7eb38352a129452da2cddcd9cd5cea12
Object Store	http://192.168.5.4:8080/v1/AUTH_7eb38352a129452da2cddcd9cd5cea12

5. Click the **admin** link, located at the top right of the browser screen and click **Sign Out**.
 6. Please tell your instructor you have completed Lab 3.

Fundamentals of OpenStack® Technology

Lab 04—Glance (Image) Service

Objectives

In this lab, you will have the opportunity to explore the various methods used to communicate with the Glance image management component of OpenStack®.

Requirements

- Completion of Lab 3
- Access to the HPE Virtual Lab equipment for completing the OpenStack® labs
- Approximately 30 minutes to complete

Introduction

The OpenStack® Glance service provides services for discovering, registering, and retrieving virtual machine images. VM images made available through Glance can be stored in a variety of locations from simple file systems to object-storage systems like the OpenStack® Swift project.

Exercise 1—Glance from the CLI

At present, the choice between project-specific commands (like `glance`) and the OSC is a matter of preference. The `openstack` command does not presently provide support for all project features, but it does offer greater consistency across projects. The examples in this module use the OSC `openstack` commands to demonstrate the Glance functionality.

1. Open an SSH terminal session to your assigned lab VM.
2. Run `cd ~/devstack` to change to the **devstack** directory.
3. Set up the required environment variables by running the command:

```
source openrc admin admin
```
4. Enter the `openstack image list` CLI command to view the images currently stored to Glance.

ID	Name	Status
a4dcfdfe-de37-40ec-b374-a73f1cb949a6	cirros-0.3.4-x86_64-uec	active
50154c1d-7f90-42a8-aab7-0f3f0a28f6fa	cirros-0.3.4-x86_64-uec-ramdisk	active
4b32e294-f161-4939-a34f-6adbd2f31af7	cirros-0.3.4-x86_64-uec-kernel	active
d4b00af9-fe6a-414c-9ecb-1ea8862d88c6	Fedora-Cloud-Base-24-1.2.x86_64	active

NOTE: The above images were installed as part of the DevStack install.

5. Q: Use the _____ `openstack` command to show information about the following Glance image.

Field	Value
checksum	8de08e3fe24ee788e50a6a508235aa64
container_format	bare
created_at	2016-12-22T15:39:57Z
disk_format	qcow2
file	/v2/images/d4b00af9-fe6a-414c-9ecb-1ea8862d88c6/file
id	d4b00af9-fe6a-414c-9ecb-1ea8862d88c6
min_disk	0
min_ram	0
name	Fedora-Cloud-Base-24-1.2.x86_64
owner	7eb38352a129452da2cddcd9cd5cea12
protected	False
schema	/v2/schemas/image
size	204590080
status	active
tags	
updated_at	2016-12-22T15:40:03Z
virtual_size	None
visibility	public

6. Q: Use the _____ openstack command to create a Glance image with the following attributes from the image at ~/devstack/file/ cirros-0.3.4-x86_64-disk.img.

Field	Value
checksum	ee1eca47dc88f4879d8a229cc70a07c6
container_format	bare
created_at	2016-12-23T08:08:51Z
disk_format	qcow2
file	/v2/images/3f73169c-4619-40d8-aaf1-4045bee960b1/file
id	3f73169c-4619-40d8-aaf1-4045bee960b1
min_disk	0
min_ram	0
name	cirros-0.3.4
owner	7eb38352a129452da2cddcd9cd5cea12
protected	False
schema	/v2/schemas/image
size	13287936
status	active
tags	
updated_at	2016-12-23T08:08:52Z
virtual_size	None
visibility	public

7. Enter the openstack image delete <image id or name> command to delete the Cirros image from Glance. You can enter openstack image list to find the exact name or ID of the image.

NOTE: In the next lab, you will upload the same image using the Horizon UI.

Exercise 2—Creating a Glance image from the Horizon UI

1. Log in to Horizon as admin / hpinvent using the domain of default.
2. Open the Project dashboard and select Compute → Images.
3. Click Create Image.

The screenshot shows the OpenStack Horizon interface. The top navigation bar includes the OpenStack logo, project name 'Default', user 'user4labs', and a search bar. Below the navigation is a dropdown menu with 'Project' and 'COMPUTE' selected. The main content area is titled 'Images' and contains tabs for 'Overview', 'Instances', 'Volumes', and 'Images'. Under the 'Images' tab, there is a table listing two images: 'cirros-0.3.4-x86_64-uec' and 'cirros-0.3.4-x86_64-uec-kernel'. A red box highlights the '+ Create Image' button at the top right of the table.

4. You are taken to the Create Image wizard screen.
5. At the Create Image screen, enter the values as shown on the following screenshot. The source image is located in the C:\Classfiles\Images\cirros-0.3.4_x86_64-disk.img folder on your HPEVL Windows landing desktop.

The screenshot shows the 'Create Image' wizard. The 'Image Details' step has 'Image Name' set to 'image4labs' and 'Image Description' set to 'Image for Lab exercises'. In the 'Image Source' step, 'Source Type' is 'File', the file path is 'cirros-0.3.4-x86_64-disk.img', and the format is 'QCOW2 - QEMU Emulator'. The 'Image Requirements' step shows 'Kernel' and 'Ramdisk' both set to 'Choose an image', 'Architecture' is empty, and 'Minimum Disk (GB)' and 'Minimum RAM (MB)' are both 0. The 'Image Sharing' step shows 'Visibility' as 'Public' and 'Protected' as 'No'. At the bottom, there are 'Cancel', 'Back', 'Next >', and a large blue 'Create Image' button, with the 'Create Image' button itself highlighted by a red box.

6. Click **Create Image** and you should see the image in the Images table.

Images

<input type="checkbox"/>	Name	Type	Status	Visibility	Protected	Disk Format	Size	<input type="button" value="Launch"/>
<input type="checkbox"/>	➤ cirros-0.3.4-x86_64-uec	Image	Active	Public	No	AMI	24.00 MB	<input type="button" value="Launch"/>
<input type="checkbox"/>	➤ cirros-0.3.4-x86_64-uec-kernel	Image	Active	Public	No	AKI	4.75 MB	
<input type="checkbox"/>	➤ cirros-0.3.4-x86_64-uec-ramdisk	Image	Active	Public	No	ARI	3.57 MB	
<input type="checkbox"/>	➤ Fedora-Cloud-Base-24-1.2.x86_64	Image	Active	Public	No	QCOW2	195.11 MB	<input type="button" value="Launch"/>
<input type="checkbox"/>	➤ image4labs	Image	Active	Public	No	QCOW2	12.67 MB	<input type="button" value="Launch"/>

Displaying 5 items

NOTE: Once the image is uploaded, refresh the browser to see the image status.

7. To see the details, click the arrow next to the username of the image owner.

<input type="checkbox"/> <input checked="" type="button" value="image4labs"/>	Image	Active	Public	No	QCOW2	12.67 MB	<input type="button" value="Launch"/>
Name image4labs ID e1ea255d-1e22-42ca-a3d3-973a1c482e2d	Visibility Public Protected No	Disk Format QCOW2 Size 12.67 MB	Min. Disk 0 Min. RAM 0				

8. Log out of the Horizon interface.
 9. Please tell the instructor you have completed Lab 4.

Fundamentals of OpenStack® Technology

Lab 05—OpenStack® Neutron (Networking)

Objectives

After completing this lab, you should be able to:

- Use the Horizon UI and CLI to examine network component configurations for the demo project
- Use the Horizon UI to create and configure a simple flat network
- Create a subnetwork
- Create a router
- Configure security group rules
- View network topology

Requirements

- Completion of Lab 4
- Access to the HPE Virtual Lab equipment for completing the OpenStack® labs
- Access to the Networking chapter of the *OpenStack Cloud Administrator Guide* at <http://docs.openstack.org/admin-guide/networking.html>
- Approximately 45 minutes to complete

Exercise 1—Using the CLI to view the current network parameters

Networking-related CLI commands used in this and subsequent labs use the OSC (`openstack`) client.

1. Start a terminal session to your assigned VM.

2. Change to the **devstack** directory with the following command:

```
cd ~/devstack
```

3. Set demo user rights for the demo project with the following command:

```
source openrc admin admin
```

4. Enter the `openstack network list` command. You should see the two networks listed.

```
stack@stack:~/devstack$ openstack network list
+----+-----+-----+
| ID           | Name     | Subnets
+----+-----+-----+
| 77faaa08-f1bb-42af-b727-26d6136c2c62 | private  | 57e784f0-4982-4ef6-8a13-159d16397f1b,
|                                         |          | 9c0e138c-5c79-4339-9f63-bcee3781040e
| 8e63faa0-641e-4619-ac32-030e6773fc01 | public   | 176d6495-16d5-43fd-b5e7-5fffa37250f9,
|                                         |          | e731b11c-87af-45ee-919c-afed517b4982
+----+-----+-----+
```

5. Run `openstack subnet list` to view the subnets.

```
stack@stack:~/devstack$ openstack subnet list
+----+-----+-----+-----+
| ID           | Name     | Network      | Subnet
+----+-----+-----+-----+
| 176d6495-16d5-43fd- | public-subnet | 8e63faa0-641e-4619-ac32-030e67 | 172.24.4.0/24
| b5e7-5fffa37250f9 |          | 73fc01
| 57e784f0-4982-4ef6-8a13-159d1 | private-subnet | 77faaa08-f1bb-
| 6397f1b          |          | 42af-b727-26d6136c2c62
| 9c0e138c-          | ipv6-private-subnet | 77faaa08-f1bb-
| 5c79-4339-9f63-bcee3781040e |          | 42af-b727-26d6136c2c62
| e731b11c-87af-45ee-919c- | ipv6-public-subnet | 8e63faa0-641e-4619-ac32-030e67 | 2001:db8::/64
| afed517b4982          |          | 73fc01
+----+-----+-----+-----+
```

NOTE: Screenshots in this lab guide display both IPv4 and IPv6 information. In your lab environment, only IPv4 is enabled, as IPv6 is not required for the lab. Use the information provided on the screenshot as a reference.

6. The previous command displayed the networks and their subnets. Neutron provides a DHCP service for private networks of a project. To display more details about the networks, enter the `openstack subnet show private-subnet` command.

```
stack@stack:~/devstack$ openstack subnet show private-subnet
+-----+-----+
| Field      | Value
+-----+-----+
| allocation_pools | 10.0.0.2-10.0.0.254
| cidr        | 10.0.0.0/24
| created_at   | 2016-12-23T14:56:51Z
| description   |
| dns_nameservers |
| enable_dhcp    | True
| gateway_ip     | 10.0.0.1
| host_routes    |
| id            | be9db964-f5ed-4399-a05e-c71f4faee9ab
| ip_version     | 4
| ipv6_address_mode | None
| ipv6_ra_mode    | None
| name          | private-subnet
| network_id     | 1389fe13-39ab-4ea9-923f-ee317127ceff
| project_id      | a573df355eb045e388a5d4d205672487
| project_id      | a573df355eb045e388a5d4d205672487
| revision_number | 2
| service_types   | []
| subnetpool_id   | None
| updated_at      | 2016-12-23T14:56:51Z
+-----+-----+
```

7. **Q:** What is the subnet range for the private network?
8. **Q:** What is the default gateway for the private subnet?
9. **Q:** What are the DNS servers configured for the private subnet?

10. To display the information about the network, enter the `openstack network show private` command.

```
stack@stack:~/devstack$ openstack network show private
+-----+-----+
| Field | Value |
+-----+-----+
| admin_state_up | UP
| availability_zone_hints | 
| availability_zones | nova
| created_at | 2016-12-26T14:22:58Z
| description | 
| id | 77faaaa08-f1bb-42af-b727-26d6136c2c62
| ipv4_address_scope | None
| ipv6_address_scope | None
| mtu | 1450
| name | private
| port_security_enabled | True
| project_id | df50aebaded045c689a19f844d691e2f
| project_id | df50aebaded045c689a19f844d691e2f
| provider:network_type | vxlan
| provider:physical_network | None
| provider:segmentation_id | 66
| revision_number | 7
| router:external | Internal
| shared | False
| status | ACTIVE
| subnets | 57e784f0-4982-4ef6-8a13-159d16397f1b, 9c0e138c-
| subnets | 5c79-4339-9f63-bcee3781040e
| tags | []
| updated_at | 2016-12-26T14:23:01Z
+-----+
```

11. Q: What is the MTU for the network **private**?

12. Q: What network virtualization technology is used for this private network (hint: check the type of the network)?

13. To list the available Neutron routes, enter the `openstack router list` command.

```
stack@stack:~/devstack$ openstack router list
+-----+-----+-----+-----+-----+-----+-----+
| ID | Name | Status | State | Distributed | HA | Project |
+-----+-----+-----+-----+-----+-----+-----+
| 4d05873b-cac9-4ae7-be28- | router1 | ACTIVE | UP | False | False | a573df355eb045e388a5d4d20 |
| 9e8dfe31f4a9 | | | | | | 5672487 |
+-----+-----+-----+-----+-----+-----+-----+
```

14. Enter the `openstack router show router1` command to display the details about the default router.

```
stack@stack:~/devstack$ openstack router show router1
+-----+-----+
| Field | Value |
+-----+-----+
| admin_state_up | UP
| availability_zone_hints |
| availability_zones | nova
| created_at | 2016-12-23T14:56:55Z
| description |
| distributed | False
| external_gateway_info | {"network_id": "23ebf159-9169-497c-a3f0-bb4423ca7bfc", "enable_snat": true,
| | "external_fixed_ips": [{"subnet_id": "c8fece55-fcaa-4eac-959c-
| | 73a3fc6be22b", "ip_address": "172.24.4.9"}]}
| flavor_id | None
| ha | False
| id | 4d05873b-cac9-4ae7-be28-9e8dfe31f4a9
| name | router1
| project_id | a573df355eb045e388a5d4d205672487
| project_id | a573df355eb045e388a5d4d205672487
| revision_number | 6
| routes |
| status | ACTIVE
| updated_at | 2016-12-23T14:57:05Z
+-----+
```

15. Q: What is the external IP address of the router?

16. Q: Is the source network address translation enabled for this router?

17. To display the information about network ports, enter the `openstack port list` command.

```
stack@stack:~/devstack$ openstack port list
+-----+-----+-----+
| ID | Name | MAC Address | Fixed IP Addresses |
+-----+-----+-----+
| 65f55a4e-556a-4c9f- | fa:16:3e:57:12:53 | ip_address='172.24.4.9',
| 89f4-7a105d08444b | | subnet_id='c8fece55-fcaa-4eac-959c-
| | 73a3fc6be22b' |
| d07105ae-82da-4c8c-a1de- | fa:16:3e:03:22:33 | ip_address='10.0.0.2',
| df34aa194391 | | subnet_id='be9db964-f5ed-4399-a05e-
| | c71f4faee9ab' |
| efcfc083d-b0e7-4234-978a- | fa:16:3e:e7:4e:32 | ip_address='10.0.0.1',
| 9951c95c1173 | | subnet_id='be9db964-f5ed-4399-a05e-
| | c71f4faee9ab' |
+-----+-----+-----+
```

18. To display the information about the port, enter the `openstack port show <port_id>` command.

19. Run `neutron help` to view all of the options available for Neutron.

A large number of Neutron options support a significant number of diverse network topologies. You have only used the Neutron CLI commands for displaying the network information; however, as shown in the Neutron help output, network components can also be created and managed from the Neutron CLI commands.

The OpenStack® security groups and floating IPs are considered Neutron features, but because they are applied to specific instances, they will be discussed in the Nova labs where the instances are generated.

20. Proceed with the next exercise.

Exercise 2—Using the Horizon UI to view the network parameters for the demo project

In this lab, you will examine the base DevStack Neutron configuration for the demo project.

1. Start the web browser and enter the IP address of your OpenStack® service to start the OpenStack® dashboard. Log in to the dashboard (to the default domain) with the credentials admin/hpinvent.
2. Having logged in to the dashboard, on the left-hand side navigation panel, expand **Admin** → **System** → **System Information** → **Services** and verify that the Neutron network service is present and enabled.

Name	Service	Region	Endpoints
cinderv3	volumev3	RegionOne	Admin http://192.168.5.4.8776/v3/1680bb85127e476aa613e5494c2d9ccc Internal http://192.168.5.4.8776/v3/1680bb85127e476aa613e5494c2d9ccc Public http://192.168.5.4.8776/v3/1680bb85127e476aa613e5494c2d9ccc
swift	object-store	RegionOne	Admin http://192.168.5.4.8080 Internal http://192.168.5.4.8080/v1/AUTH_1680bb85127e476aa613e5494c2d9ccc Public http://192.168.5.4.8080/v1/AUTH_1680bb85127e476aa613e5494c2d9ccc
heat	orchestration	RegionOne	Admin http://192.168.5.4.8004/v1/1680bb85127e476aa613e5494c2d9ccc Internal http://192.168.5.4.8004/v1/1680bb85127e476aa613e5494c2d9ccc Public http://192.168.5.4.8004/v1/1680bb85127e476aa613e5494c2d9ccc
nova	compute	RegionOne	Admin http://192.168.5.4.8774/v2.1 Internal http://192.168.5.4.8774/v2.1 Public http://192.168.5.4.8774/v2.1
cinderv2	volumev2	RegionOne	Admin http://192.168.5.4.8776/v2/1680bb85127e476aa613e5494c2d9ccc Internal http://192.168.5.4.8776/v2/1680bb85127e476aa613e5494c2d9ccc Public http://192.168.5.4.8776/v2/1680bb85127e476aa613e5494c2d9ccc
keystone	identity	RegionOne	Admin http://192.168.5.4.identity_v2_admin Internal http://192.168.5.4.identity Public http://192.168.5.4.identity
heat-cfn	cloudformation	RegionOne	Admin http://192.168.5.4.8000/v1 Internal http://192.168.5.4.8000/v1 Public http://192.168.5.4.8000/v1
nova_legacy	compute_legacy	RegionOne	Admin http://192.168.5.4.8774/v2/1680bb85127e476aa613e5494c2d9ccc Internal http://192.168.5.4.8774/v2/1680bb85127e476aa613e5494c2d9ccc Public http://192.168.5.4.8774/v2/1680bb85127e476aa613e5494c2d9ccc
neutron	network	RegionOne	Admin http://192.168.5.4.9696/ Internal http://192.168.5.4.9696/ Public http://192.168.5.4.9696/

3. Neutron supports pluggable architecture. Select the **Network Agents** tab to display currently installed network agents. In the bottom-right corner of the screen, check the Neutron version. Verify that all the agents have the status of Enabled.

The screenshot shows the OpenStack Horizon interface under the 'System' section. The 'Network Agents' tab is selected and highlighted with a red box. In the bottom right corner, the text 'Version: 10.0.2' is displayed. A red arrow points from the bottom right towards this text.

Instances	Type	Name	Host	Status	State	Last Updated	Actions
Volumes	Metadata agent	neutron-metadata-agent	stack	Enabled	Up	0 minutes	
Flavours	Open vSwitch agent	neutron-openvswitch-agent	stack	Enabled	Up	0 minutes	
Images	DHCP agent	neutron-dhcp-agent	stack	Enabled	Up	0 minutes	
Networks	L3 agent	neutron-l3-agent	stack	Enabled	Up	0 minutes	View Routers
Routers	Displaying 4 items						
Floating IPs							

4. At the top, make sure that the **alt_demo** project is selected. At the navigation panel, select **Admin → System → Networks**. You should see **private** and **public** networks enabled.

NOTICE: When you change project to **alt_demo**, Horizon will briefly display the following error: "Unable to get Orchestration service list.". Ignore the error message, and proceed with the lab.

The screenshot shows the OpenStack Horizon interface under the 'Networks' section. The 'private' and 'public' networks are highlighted with red boxes. The 'Admin State' column for both networks shows 'UP'. A red box also highlights the 'Edit Network' button for the public network.

Instances	Project	Network Name	Subnets Associated	DHCP Agents	Shared	External	Status	Admin State	Actions
Volumes	demo	private	<ul style="list-style-type: none"> private-subnet 10.0.0.0/24 ipv6-private-subnet fd6f:903:5dcc::/64 	1	No	No	Active	UP	Edit Network
Flavours	admin	public	<ul style="list-style-type: none"> ipv6-public-subnet 2001:db8::/64 public-subnet 172.24.4.0/24 	1	No	Yes	Active	UP	Edit Network

5. In the navigation panel, expand **Project → Network → Networks**.

You should see only public network.

6. Q: Why is this network view different from the view in step 4 of this exercise?

7. At the top of the screen, change the project from **alt_demo** to **demo**.

The screenshot shows the Neutron Networks interface. The top navigation bar has 'openstack' and 'Default • alt_demo' selected. The main menu on the left includes 'Project', 'Compute', and 'Network'. The 'Network' menu is expanded, showing 'Network Topology' and 'Networks'. The 'Networks' tab is selected. The table lists one network entry:

	Name	Subnets Associated	Shared	External	Status	Admin State	Actions
Admin	public	• ipv6-public-subnet 2001:db8::/64 • public-subnet 172.24.4.0/24	No	Yes	Active	UP	Edit Network

Below the table, it says 'Displaying 1 item'.

8. Q: How many networks do you see after changing the project?

9. Click the **private** link in the Networks table.

The screenshot shows the details for the 'private' network. The top navigation bar shows 'Project / Network / Networks / private'. The main content area is titled 'private' with an 'Edit Network' button. Below is a 'Network Overview' table:

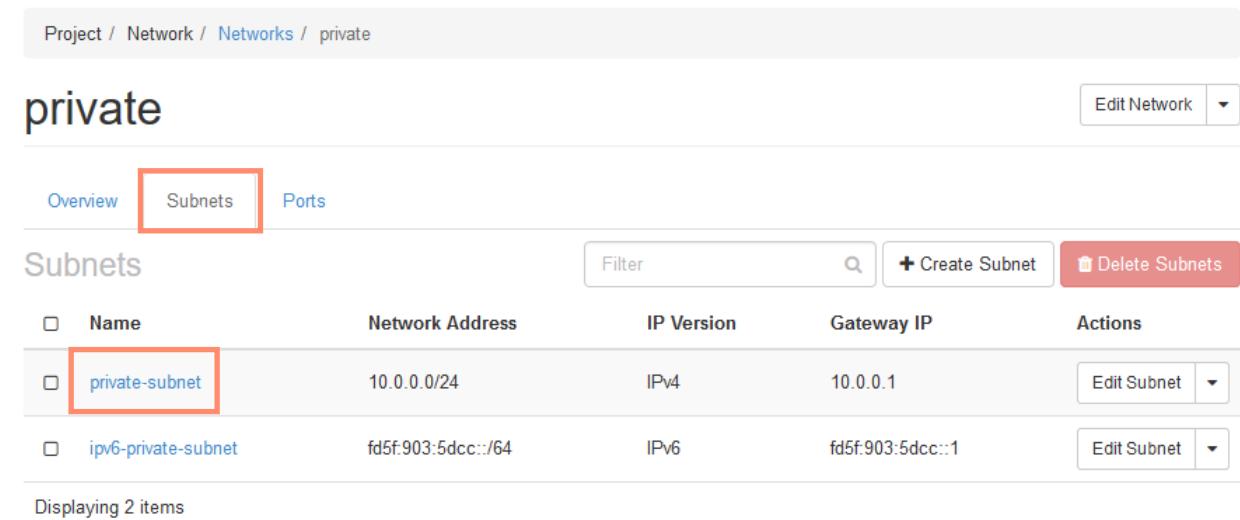
Name	private
ID	77faaa08-f1bb-42af-b727-26d6136c2c62
Project ID	df50aebaded045c689a19f844d691e2f
Status	Active
Admin State	UP
Shared	No
External Network	No
MTU	1450
Provider Network	Network Type: vxlan Physical Network: - Segmentation ID: 66

An orange arrow points to the 'Segmentation ID: 66' row.

Check the status of the network, the admin state, and the MTU parameters.

10. Q: What is the VxLAN ID of the provider network used for the **private** network? (Hint: check the segmentation ID parameter.)

11. Click the **Subnets** tab and then **private-subnet**.

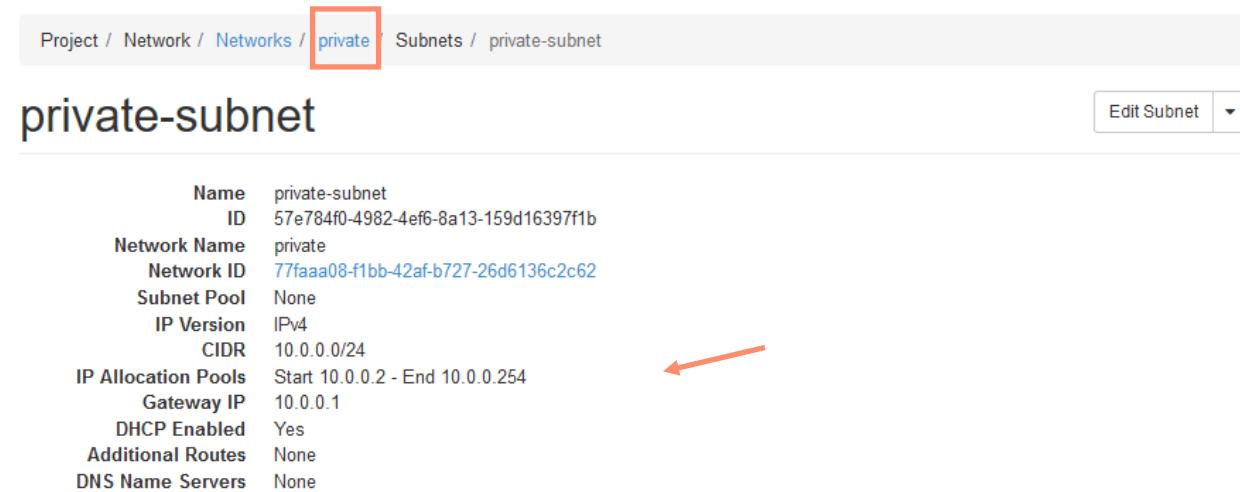


The screenshot shows the Neutron interface for a 'private' network. The 'Subnets' tab is selected. A red box highlights the first row, which contains the name 'private-subnet'. The table columns are: Name, Network Address, IP Version, Gateway IP, and Actions.

Name	Network Address	IP Version	Gateway IP	Actions
private-subnet	10.0.0.0/24	IPv4	10.0.0.1	Edit Subnet
ipv6-private-subnet	fd5f:903:5dcc::/64	IPv6	fd5f:903:5dcc::1	Edit Subnet

Displaying 2 items

12. At the private-subnet screen, check the information about the DHCP service and a default gateway. When ready, click **private** on top of the screen.



The screenshot shows the 'private-subnet' details page. The 'Gateway IP' field is highlighted with a red arrow. The table provides the following information:

Name	private-subnet
ID	57e784f0-4982-4ef6-8a13-159d16397f1b
Network Name	private
Network ID	77faaa08-f1bb-42af-b727-26d6136c2c62
Subnet Pool	None
IP Version	IPv4
CIDR	10.0.0.0/24
IP Allocation Pools	Start 10.0.0.2 - End 10.0.0.254
Gateway IP	10.0.0.1
DHCP Enabled	Yes
Additional Routes	None
DNS Name Servers	None

13. Click **Ports** and examine the information about existing network ports.

Project / Network / Networks / private

private

Edit Network ▾

Overview Subnets Ports

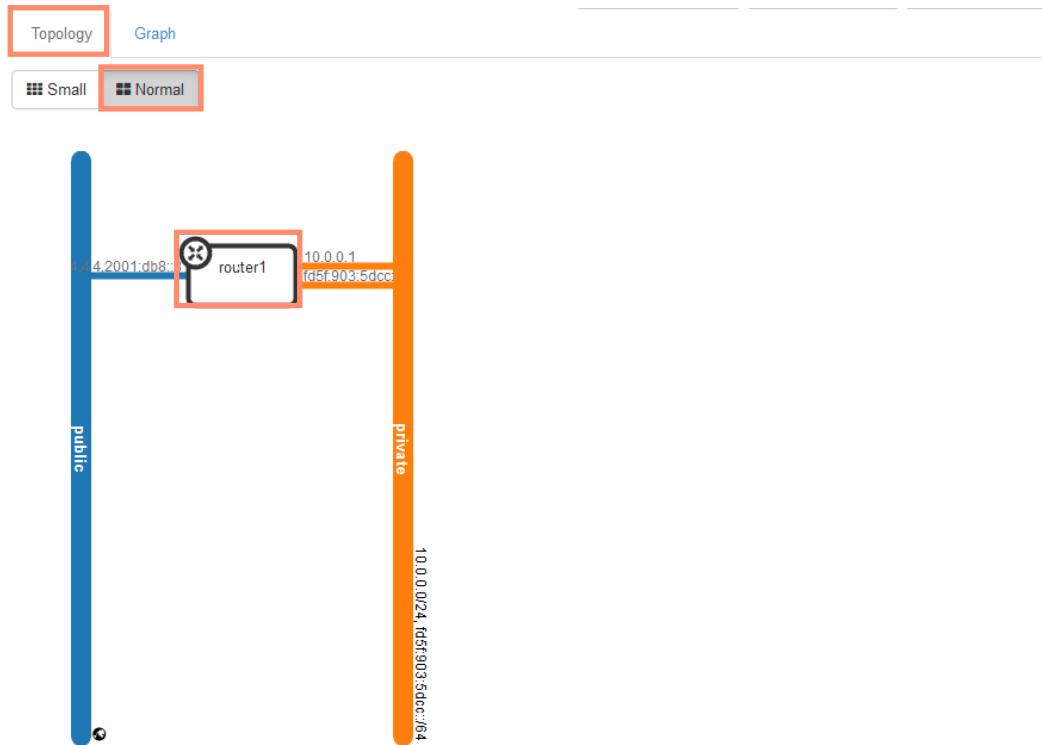
Ports

Name	Fixed IPs	Attached Device	Status	Admin State	Actions
(e6d3cace-b605)	• 10.0.0.2 • fd5f:903:5dcc:0:f816:3eff:fe69:ce77	network:dhcp	Active	UP	<button>Edit Port</button>
(eabf0c0d-590a)	• 10.0.0.1	network:router_interface	Active	UP	<button>Edit Port</button>
(fe097891-a43e)	• fd5f:903:5dcc::1	network:router_interface	Active	UP	<button>Edit Port</button>

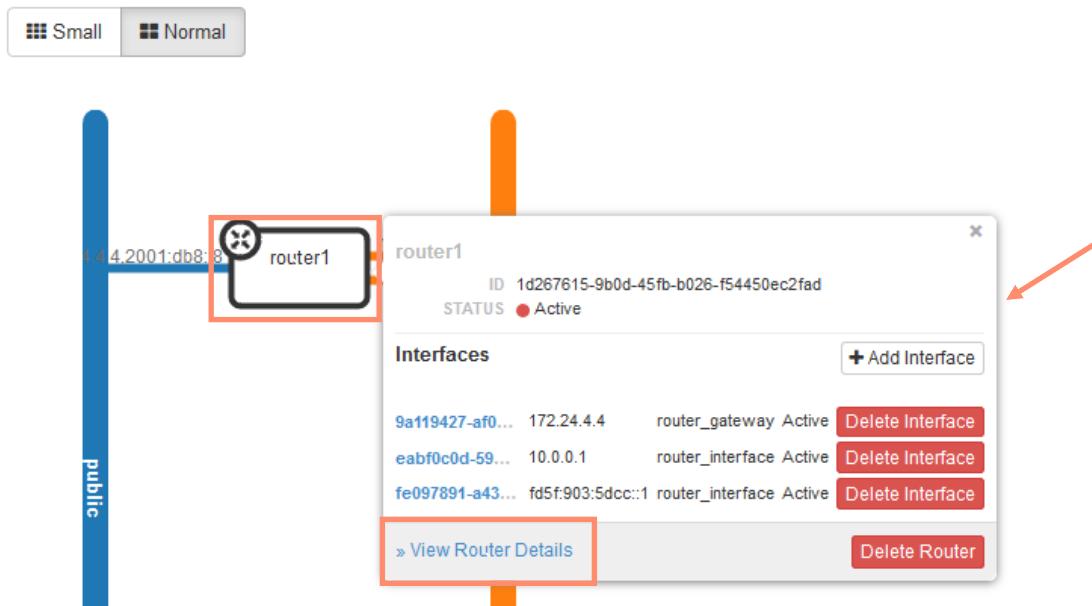
Displaying 3 items

14. At the navigation panel, expand **Project → Network → Network Topology**.

15. At the top of the screen, click **Normal** to make network components larger.



16. Hover the mouse over the **router1** icon to display the information about the router.



17. In the router information pop-up, click **View Router Details** and the **Overview** tab.

router1

	Overview	Interfaces	Static Routes
Name	router1		
ID	1d267615-9b0d-45fb-b026-f54450ec2fad		
Description			
Project ID	df50aebadded045c689a19f844d691e2f		
Status	Active		
Admin State	UP		
Availability Zones	• nova		
<hr/>			
External Gateway			
Network Name	public		
Network ID	8e63faa0-641e-4619-ac32-030e6773fc01		
External Fixed IPs	<ul style="list-style-type: none"> • Subnet ID 176d6495-16d5-43fd-b5e7-5fffa37250f9 • IP Address 172.24.4.4 • Subnet ID e731b11c-87af-45ee-919c-afed517b4982 • IP Address 2001:db8::8 		
SNAT	Enabled		

NOTE: The router information pop-up displays a lot of useful information, such as the state of the router, the router ID, and shortcuts to the management operations for this router. All of these actions and information can be gathered using the CLI, but this GUI is more convenient to visualize the network topology in complex environments.

18. Click the **Interfaces** tab of the router details screen to view two interfaces that are connected to the router.

Project / Network / Routers / router1

router1

Clear Gateway ▾

Overview Interfaces Static Routes

+ Add Interface Delete Interfaces

<input type="checkbox"/>	Name	Fixed IPs	Status	Type	Admin State	Actions
<input type="checkbox"/>	(9a119427-af00)	• 172.24.4.4 • 2001:db8::8	Active	External Gateway	UP	Delete Interface
<input type="checkbox"/>	(eabf0c0d-590a)	• 10.0.0.1	Active	Internal Interface	UP	Delete Interface

19. Sign out of the Horizon UI.

Exercise 3—Using the Horizon UI to create and configure a simple flat network

In this section of the lab, you will create a single flat network that will be used in subsequent labs.

Creating and configuring a network and a subnet

1. Start the web browser and enter the IP address of your OpenStack® service to start the OpenStack® dashboard.
2. Log in to the dashboard using the `default` domain with the credentials of `user4labs/hpinvent`.
3. Make sure that **project4labs** is selected from the Current Project drop-down list.
4. Click the **Project → Network → Networks** option to display the networks available for use by **project4labs**.

The screenshot shows the 'Networks' page in the OpenStack Horizon interface. At the top, there is a breadcrumb navigation: 'Project / Network / Networks'. Below the header, there are several buttons: 'Name = ▾', 'Filter', '+ Create Network' (which is highlighted with a red border), and 'Delete Networks'. A table lists the available networks:

<input type="checkbox"/>	Name	Subnets Associated	Shared	External	Status	Admin State	Actions
<input type="checkbox"/>	public		No	Yes	Active	UP	

Below the table, it says 'Displaying 1 item'.

5. **Q:** Which networks are currently available for **project4labs** to use?
6. Click the **+ Create Network** button, complete the fields as shown below, and then click **Next**.

The screenshot shows the 'Create Network' wizard. The title bar says 'Create Network' with a close button 'X'. The tabs at the top are 'Network' (which is selected and highlighted in blue), 'Subnet', and 'Subnet Details'. The 'Network Name' field contains 'net4labs' and has a red border around it, indicating it is the current focus. To the right of the form, there is a descriptive text: 'Create a new network. In addition, a subnet associated with the network can be created in the following steps of this wizard.' Below the network name, there is an 'Admin State' dropdown set to 'UP'. There are two checkboxes: 'Shared' (unchecked) and 'Create Subnet' (checked). At the bottom right of the form, there are three buttons: 'Cancel', '< Back', and 'Next >' (which is highlighted with a red border).

7. Fill in the fields on the subnet screen to create a subnet on the network and apply the **10.10.1.0/24** address to it. When ready, click **Next**.

Create Network

Network **Subnet** Subnet Details

Subnet Name
subnet4labs

Network Address Source
Enter Network Address manually

Network Address 10.10.1.0/24

IP Version IPv4

Gateway IP

Disable Gateway

Creates a subnet associated with the network. You need to enter a valid "Network Address" and "Gateway IP". If you did not enter the "Gateway IP", the first value of a network will be assigned by default. If you do not want gateway please check the "Disable Gateway" checkbox. Advanced configuration is available by clicking on the "Subnet Details" tab.

Cancel « Back **Next »**

In the following step, you are applying a DHCP service to the subnet and specifying a pool of IP addresses to be used by the DHCP service for use by any instances attached to the subnet.

8. Fill in the fields on the Subnet Details screen as shown below and click **Create**.

Create Network

Subnet Details

Enable DHCP Specify additional attributes for the subnet.

Allocation Pools

DNS Name Servers

Host Routes

Create

NOTE: When specifying the allocation pool, a comma with no spaces must be included between the IP address at the beginning of the IP pool range and the IP address at the end of the IP pool range.

9. Verify that the network you created and the associated subnet are listed in the Networks table and that the status is **Active**.

Networks

<input type="button" value="Name = ▾"/> <input type="button" value="Filter"/> <input type="button" value="+ Create Network"/> <input type="button" value="Delete Networks"/>							
<input type="checkbox"/>	Name	Subnets Associated	Shared	External	Status	Admin State	Actions
<input type="checkbox"/>	net4labs	• subnet4labs 10.10.1.0/24	No	No	Active	UP	<input type="button" value="Edit Network"/> ▾
<input type="checkbox"/>	public		No	Yes	Active	UP	

Displaying 2 items

Creating and configuring a router

1. Click the Project → Network → Routers option and click **Create Router**.

The screenshot shows the OpenStack Neutron interface. The top navigation bar includes the OpenStack logo, project name 'Default • project4labs', and user 'user4labs'. The left sidebar has sections for Project, Compute, Network, and Orchestration, with 'Network' currently selected. The main content area is titled 'Routers' and displays a table with columns: Name, Status, External Network, Admin State, and Actions. A search bar at the top right allows filtering by 'Router Name'. A prominent red box highlights the '+ Create Router' button.

2. Fill in the Create Router fields as shown below and then click **Create Router**. Notice that by selecting the External Network of **public**, you are attaching the existing external network to the router.

The screenshot shows the 'Create Router' dialog. It contains fields for 'Router Name' (set to 'router4labs'), 'Admin State' (set to 'UP'), and 'External Network' (set to 'public'). To the right, a 'Description:' section explains that it creates a router with specified parameters. At the bottom are 'Cancel' and 'Create Router' buttons, with 'Create Router' highlighted by a red box.

3. Verify that **router4labs** exists in the Router tab and then click the **router4labs** link.

The screenshot shows the 'Routers' table. The header includes 'Router Name = ▾', 'Filter', '+ Create Router', and 'Delete Routers'. The table has columns: Name, Status, External Network, Admin State, and Actions. One row is displayed, showing 'router4labs' in the Name column, 'Active' in Status, 'public' in External Network, 'UP' in Admin State, and a 'Clear Gateway' dropdown in the Actions column. The 'router4labs' link in the Name column is highlighted by a red box.

4. Select the **Interfaces** tab.

5. Click **Add Interface**.

The screenshot shows the 'router4labs' router configuration page. The 'Interfaces' tab is active. A red box highlights the '+ Add Interface' button in the top right corner of the interface table area.

Name	Fixed IPs	Status	Type	Admin State	Actions
No items to display.					

6. Select the **net4labs: 10.10.1.0/24 (subnet4labs)** option from the subnet drop-down list and click **Submit**.

Add Interface

Subnet *

net4labs: 10.10.1.0/24 (subnet4labs)

IP Address (optional) ?

Router Name *

Router ID *

Description:
 You can connect a specified subnet to the router.
 The default IP address of the interface created is a gateway of the selected subnet. You can specify another IP address of the interface here. You must select a subnet to which the specified IP address belongs to from the above list.

7. Verify that the internal interface now exists on the router.
 8. Make sure the interface has a status of **Active**. You might have to refresh the screen to see that the status of the internal interface changes from **Down** to **Active**.

The screenshot shows the 'router4labs' router configuration page. The 'Interfaces' tab is active. A red box highlights the 'Active' status of the interface listed in the table.

<input type="checkbox"/> Name	Fixed IPs	Status	Type	Admin State	Actions
<input type="checkbox"/> (3ea91bb4-74e6)	* 10.10.1.1	Active	Internal Interface	UP	<input type="button" value="Delete Interface"/>

Displaying 1 item

9. Select the **Overview** tab.

router4labs

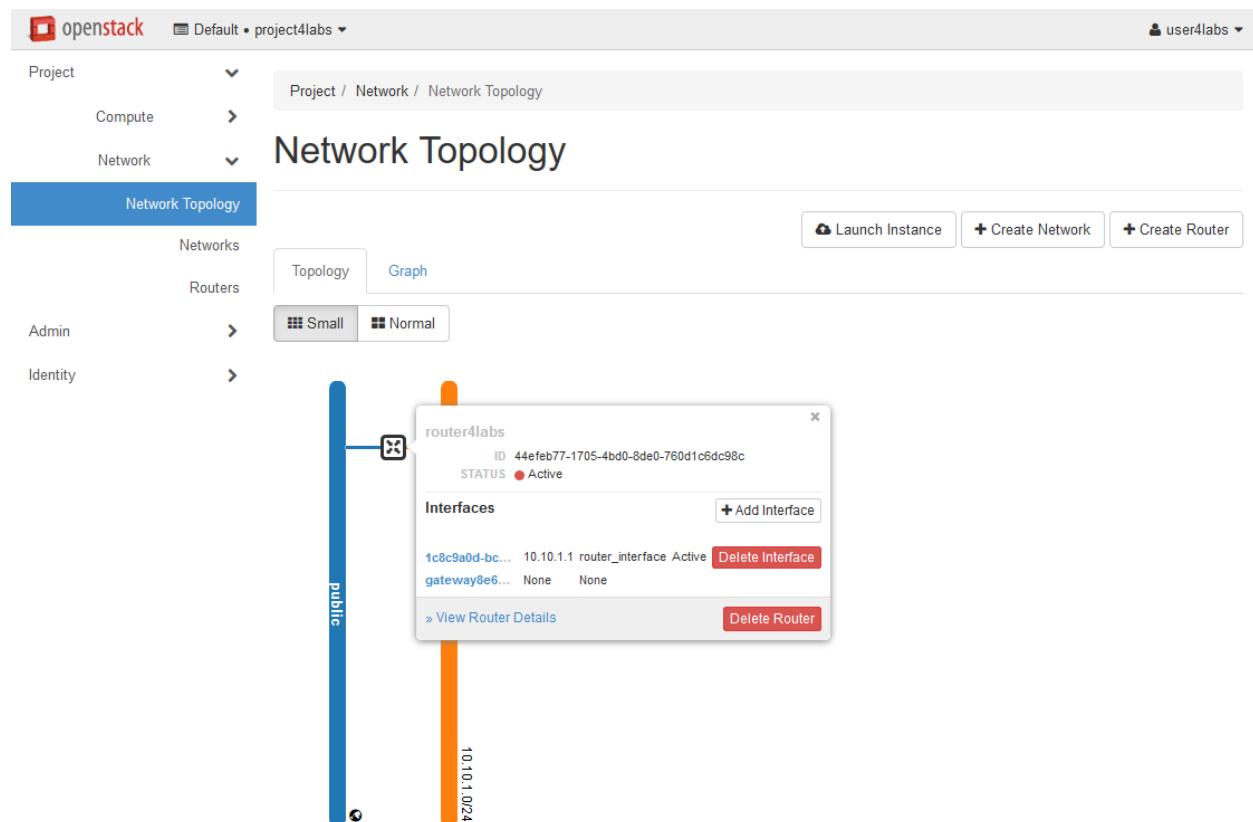
	Overview	Interfaces	Static Routes
Name	router4labs		
ID	44feb77-1705-4bd0-8de0-760d1c6dc98c		
Description			
Project ID	354adb7fbb9407298053b3d2cc24521		
Status	Active		
Admin State	UP		
Availability Zones	• nova		

External Gateway

Network Name	public
Network ID	8e63faa0-641e-4619-ac32-030e6773fc01
External Fixed IPs	<ul style="list-style-type: none"> Subnet ID 1766495-16d5-43fd-b5e7-5fffa37250f9 IP Address 172.24.4.6 Subnet ID e731b11c-87af-45ee-919c-afed517b4982 IP Address 2001:db8::4
SNAT	Enabled

10. Q: The internal interface has an assigned IP address. Where did it obtain the IP address?

11. Click the **Project → Network → Network Topology** option.



12. Make sure you understand where the IP addresses shown in the topology diagram were obtained.

13. Several points of interest you should notice about the topology:

- The **net4labs** network is now attached to the public network through the router.
- IP addresses were assigned by OpenStack® to the router interfaces of both the public and **net4labs** networks.

You now have a network to which you will add instances in the next lab.

14. Please let your instructor know that you have completed Lab 5.

Fundamentals of OpenStack® Technology

Lab 06—OpenStack® Nova (Compute)

Objectives

After completing this lab, you should be able to:

- Verify that the required Nova services are enabled and running
- Use the Horizon UI to create an OpenStack® instance, which includes:
 - Creating a security key pair
 - Modifying security groups
 - Viewing available flavors
- Configure the instance parameters, which includes:
 - Associating a Floating IP to an instance
 - Accessing the created instance
 - Pausing and suspending an instance
 - Creating and deleting a snapshot of an image

Requirements

- Completion of Lab 5
- Access to HPEVL
- Approximately 45 minutes to complete

Introduction

Let us take a look at what you have covered in the labs up to this point:

- Lab 2, Management Tools, introduced you to the OpenStack® CLI and the Horizon UI. You were able to use some of the more common OpenStack® CLI commands, a REST client browser plug-in, and CURL to make API requests of the OpenStack® services.
- In Lab 3, Keystone, you created the **project4labs** project and **user4labs** user, which will be used in this lab.
- In Lab 4, Glance, you created the **image4labs** image, which will be used in this lab.

- In Lab 5, Neutron, you created the **net4labs** network, **subnet4labs** subnet, and **router4labs** router, which will be used in this lab.

Now you have most of the primary OpenStack® components required to generate an OpenStack® instance, which you will do in this lab.

Nova is the project name for OpenStack® Compute, a cloud computing fabric controller, the main part of an Infrastructure as a Service (IaaS) system. Several of its major features, such as networking and block storage, are now provided by newer standalone service, such as Neutron and Cinder. Nova can still perform those features to accommodate existing installations, but it is generally recommended that the newer services be used for new installations.

Exercise 1—Verifying the required Nova services are enabled and running

The DevStack implementation of OpenStack® you are running is a single-node setup with all the services running on the same machine.

An instance cannot be created properly if all the required Nova services are not enabled and operational.

1. In your terminal session window, run the `openstack compute service list` command.

openstack compute service list						
ID	Binary	Host	Zone	Status	State	Updated At
5	nova-conductor	stack	internal	enabled	up	2017-01-30T17:02:39.000000
7	nova-scheduler	stack	internal	enabled	up	2017-01-30T17:02:35.000000
8	nova-consoleauth	stack	internal	enabled	up	2017-01-30T17:02:36.000000
9	nova-compute	stack	nova	enabled	up	2017-01-30T17:02:35.000000

Note the status and the state of Nova services on your system. Make sure that all the services are **enabled** and in the **up** state.

Exercise 2—Create an instance

Generating a security key pair

Key pairs provide secure authentication to your instances. As part of the first boot of a virtual image, the private key of your key pair is added to the authorized key file of the login account. For security purposes, you typically need to use an SSH key pair to log in to a running instance.

Some images have built-in accounts created with associated passwords. However, because images are often shared by many users, you should not put passwords into the images. Nova supports injecting SSH keys into instances before they are booted. This enables you to securely log in to the instances that you create.

Nova generates a public and a private key pair, and it sends the private key to the user. The public key is stored so that it can be injected into instances.

Key pairs can be generated from the Horizon UI. You will generate the key pair from the command line, which will simplify the process:

1. Start a terminal session to your assigned VM.
2. Type `cd ~/devstack` to change to the **devstack** directory.
3. Type `source openrc user4labs project4labs` to set the environment variables for the user and project.
4. Enter the following command to create a key pair called **keypair4labs** and save the public key to `keypair4labs.pem`:
`openstack keypair create keypair4labs > keypair4labs.pem`
5. Enter the `openstack keypair list` command to verify the key pair was created successfully.

```
stack@stack:~/devstack$ openstack keypair list
+-----+-----+
| Name | Fingerprint |
+-----+-----+
| keypair4labs | 9c:1a:78:7a:37:84:0e:41:f0:b0:fc:3a:1c:c4:aa:d8 |
+-----+-----+
```

6. SSH requires specific file rights for a public key, so type in the following Linux command to change the file rights:

```
chmod 600 keypair4labs.pem
```

Modifying a security group

In the remaining steps of this exercise, you will use the Horizon UI, because it is easier to follow the process flow for creating an instance.

1. Start a browser window and enter the IP address of your lab VM to display the OpenStack® Dashboard login screen. Log in to the default domain with the user4labs/hpinvent credentials.
2. Expand Project → Compute → Access & Security and click the Key Pairs tab.
3. Verify the keypair4labs key pair exists in the Key Pair table.

The screenshot shows the 'Access & Security' page in the Horizon interface. The 'Key Pairs' tab is selected. A table lists one item: 'keypair4labs'. The 'Name' column for this item is highlighted with a red box. The table has columns for 'Key Pair Name', 'Fingerprint', and 'Actions'. The 'Actions' column contains a 'Delete Key Pair' button, which is also highlighted with a red box.

Key Pair Name	Fingerprint	Actions
keypair4labs	9c:1a:78:7a:37:84:0e:41:f0:b0:fc:3a:1c:c4:aa:d8	<button>Delete Key Pair</button>

Security groups are used to group a list of rules that can be applied to allow or prevent types of communications with instances.

4. Expand Project → Compute → Access & Security in the navigation panel and select the **Security Groups** tab.

The screenshot shows the 'Access & Security' page in the Horizon interface. The 'Security Groups' tab is selected. A table lists one item: 'default'. The 'Name' column for this item is highlighted with a red box. The table has columns for 'Name', 'Description', and 'Actions'. The 'Actions' column contains a 'Manage Rules' button, which is also highlighted with a red box.

Name	Description	Actions
default	Default security group	<button>Manage Rules</button>

The default security group is automatically created in OpenStack®. It can be modified, but not deleted. Additional security groups can be created, and they can be deleted only when not associated to one or more instances.

In this lab you will modify the existing default security group.

5. Click the **Manage Rules** button associated with the default security group.

The following rule enables an SSH session to any instance that uses the default security group.

6. Click **+ Add Rule**, fill in the fields as shown below, and then click **Add**.

Add Rule

Rule *	<input type="text" value="Custom TCP Rule"/>	Description:
Direction	<input type="text" value="Ingress"/>	Rules define which traffic is allowed to instances assigned to the security group. A security group rule consists of three main parts:
Open Port *	<input type="text" value="Port"/>	Rule: You can specify the desired rule template or use custom rules, the options are Custom TCP Rule, Custom UDP Rule, or Custom ICMP Rule.
Port	<input type="text" value="22"/>	Open Port/Port Range: For TCP and UDP rules you may choose to open either a single port or a range of ports. Selecting the "Port Range" option will provide you with space to provide both the starting and ending ports for the range. For ICMP rules you instead specify an ICMP type and code in the spaces provided.
Remote *	<input type="text" value="CIDR"/>	Remote: You must specify the source of the traffic to be allowed via this rule. You may do so either in the form of an IP address block (CIDR) or via a source group (Security Group). Selecting a security group as the source will allow any other instance in that security group access to any other instance via this rule.
CIDR	<input type="text" value="0.0.0.0/0"/>	

Cancel **Add**

The following rule enables ICMP pings to any instance that uses the default security group.

7. Click **+ Add Rule**, fill in the fields as shown in the following screenshot, and then click **Add**.

Add Rule

Rule *	<input type="text" value="All ICMP"/>	Description:
Direction	<input type="text" value="Ingress"/>	Rules define which traffic is allowed to instances assigned to the security group. A security group rule consists of three main parts:
Remote *	<input type="text" value="CIDR"/>	Rule: You can specify the desired rule template or use custom rules, the options are Custom TCP Rule, Custom UDP Rule, or Custom ICMP Rule.
CIDR	<input type="text" value="0.0.0.0/0"/>	Open Port/Port Range: For TCP and UDP rules you may choose to open either a single port or a range of ports. Selecting the "Port Range" option will provide you with space to provide both the starting and ending ports for the range. For ICMP rules you instead specify an ICMP type and code in the spaces provided.
		Remote: You must specify the source of the traffic to be allowed via this rule. You may do so either in the form of an IP address block (CIDR) or via a source group (Security Group). Selecting a security group as the source will allow any other instance in that security group access to any other instance via this rule.

Cancel **Add**

- Verify that the two new rules have been added to the list of rules in the default security group.

<input type="checkbox"/>	Direction	Ether Type	IP Protocol	Port Range	Remote IP Prefix	Remote Security Group	Actions
<input type="checkbox"/>	Egress	IPv6	Any	Any	::/0	-	<button>Delete Rule</button>
<input type="checkbox"/>	Ingress	IPv6	Any	Any	-	default	<button>Delete Rule</button>
<input type="checkbox"/>	Egress	IPv4	Any	Any	0.0.0.0/0	-	<button>Delete Rule</button>
<input type="checkbox"/>	Ingress	IPv4	Any	Any	-	default	<button>Delete Rule</button>
<input type="checkbox"/>	Ingress	IPv4	ICMP	Any	0.0.0.0/0	-	<button>Delete Rule</button>
<input type="checkbox"/>	Ingress	IPv4	TCP	22 (SSH)	0.0.0.0/0	-	<button>Delete Rule</button>

Displaying 6 items

Flavors

Virtual hardware templates are called “flavors” in OpenStack®. Flavors define sizes for RAM, disk, number of cores, and so on, and are configurable by admin users. The rights can also be delegated to other users by redefining the access controls, and they can be made available to all projects or to a specific project.

- In the navigation panel, expand **Admin** → **System** → **Flavors**.
- Use the “m1.tiny” flavor for your instance. Click the **Edit Flavor** button associated with the **m1.tiny** flavor.

IMPORTANT: Do not change any of the values on the Flavor info tab.

- Click the **Flavor Access** tab.
- Q:** Can the **project4labs** project use the **m1.tiny** flavor?
- Click **Cancel**.

Creating an instance

- Make sure **project4labs** is selected from the Current Project drop-down list.
- In the navigation panel, expand **Project** → **Compute** → **Instances** and click the **+ Launch Instance** button.

Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
No items to display.										

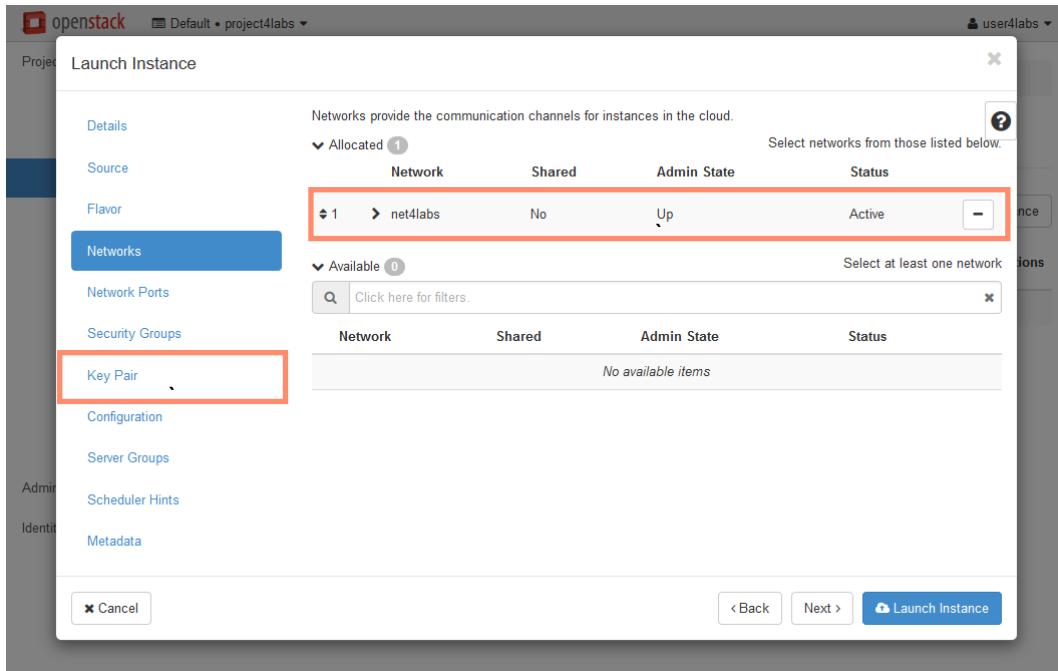
8. Fill in the fields at the Details screen as shown on the screenshot below and click **Next**.

9. Fill in the details at the Source screen as shown below and click **Next**. To see the list of images, you must first select the **Image** as the boot source. Then, from the list of available images, you can select **image4labs**.

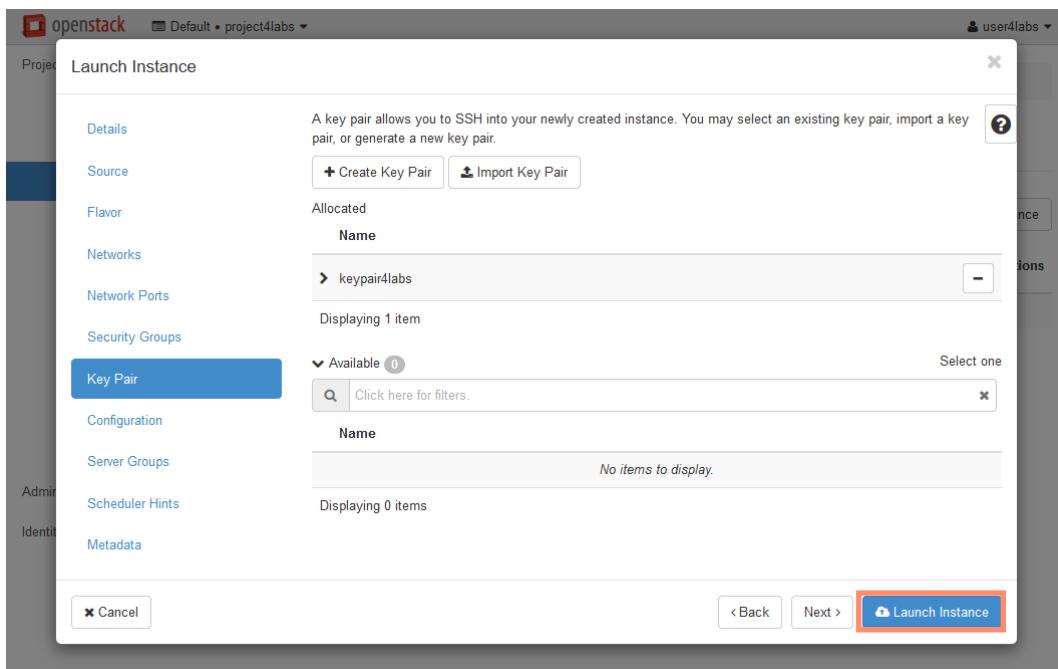
Name	Updated	Size	Type	Visibility
image4labs	12/23/16 4:20 PM	12.67 MB	qcow2	Public
cirros-0.3.4-x86_64-uec	12/23/16 3:56 PM	24.00 MB	ami	Public
Fedora-Cloud-Base-24-1.2.x86_64	12/23/16 3:56 PM	195.11 MB	qcow2	Public

10. At the Flavor screen, select **m1.tiny** flavor from the list and click **Next**.

11. Note that the private network **net4labs** is automatically selected, as it is the only available network. At the left-hand side, click **Key Pair**.



12. Note that the **keypair4labs** is automatically selected. Examine the remaining options at the left-hand side, but do not change anything. When ready, click **Launch Instance**.



NOTE: The instance typically takes 1 or 2 minutes to spawn.

13. Verify that the created instance is displayed in the Instances table and has a status of **Active**.

Instances

The screenshot shows a table with the following columns: Instance Name, Image Name, IP Address, Size, Key Pair, Status, Availability Zone, Task, Power State, Time since created, and Actions. The first row contains the header names. Below them, there is one data row highlighted with a red border. This row contains the value 'instance4labs' in the 'Instance Name' column, and 'm1.tiny' in the 'Image Name' column. The 'Status' column shows 'Active'. The 'Actions' column contains a button labeled 'Create Snapshot'.

Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
instance4labs	m1.tiny	10.10.1.104		keypair4labs	Active	nova	None	Running	0 minutes	Create Snapshot

Displaying 1 item

14. After the instance status changes to **Active**, click the **instance4lab** link in the Name column of the Instances table. Then, click the **Overview** tab to view the details about the instance.

15. Note the following information:

- The ID value can be used for tracking the process of the instance.
- The instance specs listed are determined by the selected flavor.
- The security rules of the security group are associated with the instance.

instance4labs

The screenshot shows the 'Overview' tab of the instance details screen. It displays the following information:

Name	instance4labs
ID	6c939d7f-51e0-4afe-a171-22307c50f093
Status	Active
Availability Zone	nova
Created	26 Dec 2016, 3:13 p.m.
Time Since Created	2 minutes

Specs

Flavour Name	m1.tiny
Flavour ID	1
RAM	512MB
VCPUs	1 VCPU
Disk	1GB

IP Addresses

Net4labs	10.10.1.104
----------	-------------

16. Click the **Log** tab to display the log associated with the instance.

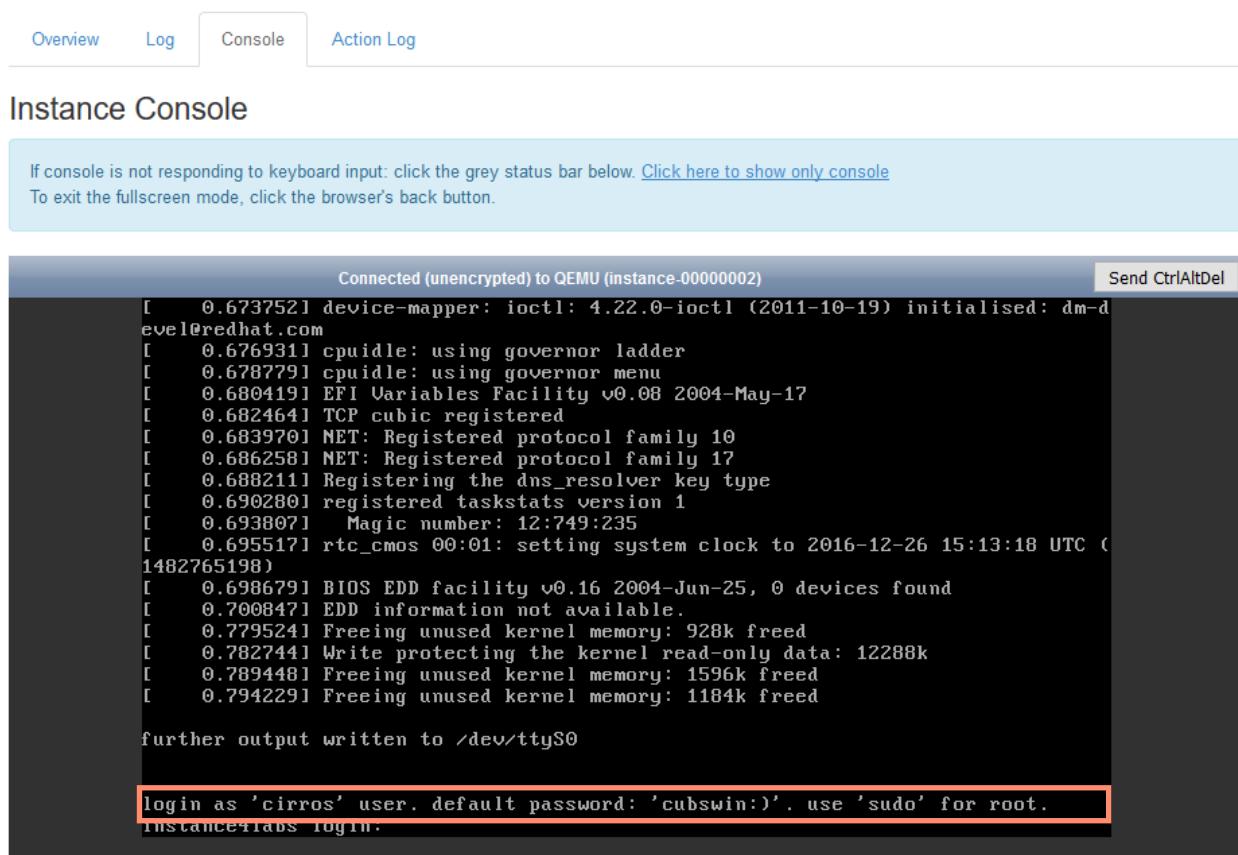
17. In the Log Length field, enter the value of **50** to display more log content.

You need to know the user name and the password for the CirrOS instance later in the lab.

18. On the Instance Details screen, click the **Console** tab.

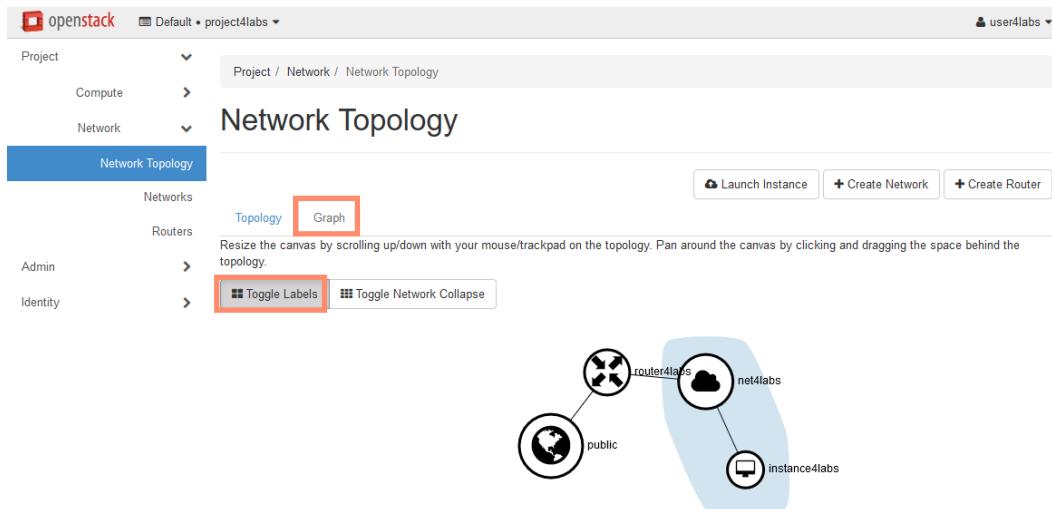
19. Click the **Send Ctrl + Alt + Del** button. The instance restarts and prompts you for the credentials.

20. Note the username of **cirros** and password of **cubswin:)** listed at the login prompt of the screen to log in to the instance. Do not log in using the console. You will use these credentials to SSH to the instance.



You have now created an operational OpenStack® instance. Let us see what it looks like in the network.

21. In the navigation panel, expand **Project → Network → Network Topology** and click the **instance** icon. Select the **Graph** tab and click **Toggle Labels**.



22. Click the **instance4labs** icon.



23. **Q:** What is the IP address of the instance?

24. **Q:** Where was the instance IP address obtained?

25. Try to ping your instance from a terminal session.

NOTE: Press **Control + C** to stop the ping.

26. **Q:** Why are you not able to ping the instance?

Exercise 3—Accessing an instance from an external network

Associating a Floating IP to an instance

A **Floating IP** is an IP address on an external network which is associated with a specific port, and optionally a specific IP address, on a private OpenStack® Networking network. A Floating IP allows access to an instance on a private network from an external network.

Associating a Floating IP to your instance allows you to access the instance from the external network.

1. In the navigation panel, expand **Project** → **Compute** → **Instances**.
2. Click the **drop-down icon** next to the **Create Snapshot** button that is associated with your instance and click the **Associate Floating IP** option.

The screenshot shows the OpenStack Compute Instances page. At the top, there is a navigation bar with 'Project / Compute / Instances'. Below it is a search bar with 'Instance Name = ▾' and a dropdown menu. To the right of the search bar are buttons for 'Filter', 'Launch Instance', 'Delete Instances', and 'More Actions ▾'. The main table has columns: Instance Name, Image Name, IP Address, Size, Key Pair, Status, Availability Zone, Task, Power State, Time since created, and Actions. A row for 'instance4labs' is selected, showing details: IP Address 10.10.1.104, Image m1.tiny, Key Pair keypair4labs, Status Active, Availability Zone nova, Task None, Power State Running, and Time since created 12 minutes. To the right of the table are three buttons: 'Create Snapshot' (with a dropdown arrow), 'Associate Floating IP' (highlighted with a red box), and 'Attach Interface'.

3. Click the **+** button next to the IP address field.

The screenshot shows the 'Manage Floating IP Associations' dialog box. It has fields for 'IP Address*' (dropdown menu 'No floating IP addresses allocated' with a '+' button highlighted) and 'Port to be associated*' (dropdown menu 'instance4labs: 10.10.1.104'). A note says 'Select the IP address you wish to associate with the selected instance or port.' At the bottom are 'Cancel' and 'Associate' buttons.

A pool of Floating IP addresses called “public” was created by DevStack. In the next step, you will select an IP address from that Floating IP address pool.

4. Q: How many Floating IPs are currently available for allocation?

5. Make sure the public pool is selected and then click **Allocate IP**.

The screenshot shows the 'Allocate Floating IP' dialog box. The 'Pool' dropdown is set to 'public'. The 'Allocate IP' button is highlighted with a red border.

6. Make sure that **Instance4Labs** is selected from the Ports to be associated drop-down list and then click **Associate**.

The screenshot shows the 'Manage Floating IP Associations' dialog box. The 'IP Address' field contains '172.24.4.11'. The 'Port to be associated' dropdown contains 'instance4labs: 10.10.1.104'. The 'Associate' button is highlighted with a red border.

7. Verify that the Floating IP address exists in the IP address column pertaining to the **instance4Labs** instance of the Instances table.

NOTE: You might need to refresh your browser window to see the Floating IP address.

The screenshot shows the Instances table. The table has columns: Instance Name, Image Name, IP Address, Size, Key Pair, Status, Availability Zone, Task, Power State, Time since created, and Actions. One row is visible, corresponding to the 'instance4labs' instance. The 'IP Address' column shows two entries: '10.10.1.104' and '172.24.4.11'. The entire row for 'instance4labs' is highlighted with a red border.

8. **Q:** What is the current status of the instance?

9. **Q:** What is the IP address of the assigned Floating IP?

Accessing the instance

1. From your terminal session, type the following command to initiate an SSH connection to your instance. Make sure to replace <assigned floating IP address> with the Floating IP assigned to your system:
ssh -i ~/devstack/keypair4labs.pem cirros@<assigned floating IP Address>
2. Make sure you understand all of the parameters in the above command.

NOTE: The `~/devstack` portion of the above command is a Linux shortcut to the location where the `keypair4labs.pem` is located.

3. Type `yes` at the `Are you sure you want to continue connecting (yes/no) ?` prompt.
The `$` prompt indicates you are at the command prompt of the instance. On the command prompt, run the `ec2metadata` command.

```
stack@stack: ~/devstack
$ ec2metadata
ami-id: None
ami-launch-index: 0
ami-manifest-path: FIXME
block-device-mapping/ami: vda
block-device-mapping/ebs0: /dev/vda
block-device-mapping/root: /dev/vda
hostname: instance4labs.novalocal
instance-action: none
instance-id: i-00000002
instance-type: m1.tiny
local-hostname: instance4labs.novalocal
local-ipv4: 10.10.1.104
placement/availability-zone: nova
public-hostname: instance4labs.novalocal
public-ipv4: 172.24.4.11
public-keys/0: openSSH-key
reservation-id: r-kjyhx2v0
security-groups: default
$
```

4. **Exit** the SSH session to the instance.

Exercise 4—Pausing and suspending the VM

In this section, you will have the opportunity to see the difference between pausing and suspending an instance.

Pausing an instance

NOTE: You will be pausing your instance from the command line. You can also pause an instance from the Horizon GUI, but the command line enables you to use the commands to pause and then quickly verify the effect of those commands.

- From your terminal session, enter the following command to find the ID of the instance you created and copy it:

```
openstack server list
```

```
stack@stack:~/devstack$ openstack server list
+-----+-----+-----+-----+
| ID      | Name     | Status | Networks
+-----+-----+-----+-----+
| 6c939d7f-51e0-4afe-a171-22307c50f093 | instance4labs | ACTIVE | net4labs=10.10.1.104, 172.24.4.11 |
+-----+-----+-----+-----+
```

- Run the following command to pause the instance:

```
openstack server pause <instance id>
```

```
stack@stack:~/devstack$ openstack server list
+-----+-----+-----+-----+
| ID      | Name     | Status | Networks
+-----+-----+-----+-----+
| 6c939d7f-51e0-4afe-a171-22307c50f093 | instance4labs | PAUSED | net4labs=10.10.1.104, 172.24.4.11 |
+-----+-----+-----+-----+
```

- Q: Can you ping the Floating IP address of the instance?

4. View the instance VM image process by entering the following command:

```
ps -e l | grep -v grep | grep <paste in your copied instance ID here>
```

The output should be similar to the following.

```
stack@stack:~/devstack$ ps -e l | grep -v grep | grep 6c939d7f-51e0-4afe-a171-22307c50f093
6 116 26945 1 20 0 1486788 232812 - S1 ? 0:04 qemu-system-x86_64 -enable-kvm -name instance-00000002 -S -machine pc-i440fx-xenial,accel=kvm,usb=off -m 512 -realtime mlock=off -smp 1,sockets=1,cores=1,threads=1 -uuid 6c939d7f-51e0-4afe-a171-22307c50f093 -mbios type=1,manufacturer=OpenStack Foundation,product=OpenStack Nova,version=14.0.3,serial=bbafab6d-9bde-4979-9f48-cab4a69c9912,uuid=6c939d7f-51e0-4afe-a171-22307c50f093,family=Virtual Machine -no-user-config -nodefaults -chardev socket,id=charmonitor,path=/var/lib/libvirt/qemu/domain-instance-00000002/monitors.sock,server,nowait -mon chardev=charmonitor,id=monitor,mode=control -rtc base=utc,driftfix=slew -global kvm-pit.lost_tick_policy=discard -no-hpet -no-shutdown -boot strict=on -device pix3-usb-uhci,id=usb,bus=pci.0,addr=0x1.0x2 -drive file=/dev/disk/by-path/ip-192.168.5.4:3260-iscsi-ign.2010-0.org.openstack:volume-fb0d55a1-f0e9-4c70-bf1c-c5652ee910d7-lun-1,format=raw,if=none,id=drive-virtio-disk0,serial=fb0d55a1-f0e9-4c70-bf1c-c5652ee910d7,cache=none,aio=native -device virtio-blk-pci,scsi=off,bus=pci.0,addr=0x4,drive=drive-virtio-disk0,bootindex=1 -netdev tap,fd=26,id=hostnet0,vhost=on,vhostid=28 -device virtio-net-pci,netdev=hostnet0,id=net0,mac=fa:16:3e:e7:26:b1,bus=pci.0,addr=0x3 -chardev file,id=charserial10,path=/opt/stack/data/nova/instances/6c939d7f-51e0-4afe-a171-22307c50f093/console.log -device isa-serial,chardev=charserial0,id=serial10 -chardev pty,id=charserial11 -device isa-serial,chardev=charserial11,id=serial11 -vnc 127.0.0.1:0 -k en-us -device cirrus-vga,id=video0,bus=pci.0,addr=0x2 -device virtio-balloon-pci,id=balloon0,bus=pci.0,addr=0x5 -msg timestamp=on
```

NOTE: You should not be concerned about all of the detailed information displayed; you just want to verify that the instance process is operational. The `ps` command returned the instance ID several times, indicating the instance process is still operational.

5. Run the following command to unpause the instance:

```
openstack server unpause <paste in your copied instance ID here>
```

6. Run the following command to check the status of your instance and then verify you can ping it:

```
openstack server list
```

```
stack@stack:~/devstack$ openstack server unpause 6c939d7f-51e0-4afe-a171-22307c50f093
stack@stack:~/devstack$ openstack server list
+-----+-----+-----+-----+
| ID      | Name       | Status    | Networks          | Image Name |
+-----+-----+-----+-----+
| 6c939d7f-51e0-4afe-a171-22307c50f093 | instance4labs | ACTIVE    | net4labs=10.10.1.104, 172.24.4.11 | |
+-----+-----+-----+-----+
stack@stack:~/devstack$ ping -c 3 172.24.4.11
PING 172.24.4.11 (172.24.4.11) 56(84) bytes of data.
64 bytes from 172.24.4.11: icmp_seq=1 ttl=63 time=1.06 ms
64 bytes from 172.24.4.11: icmp_seq=2 ttl=63 time=0.471 ms
64 bytes from 172.24.4.11: icmp_seq=3 ttl=63 time=0.424 ms

--- 172.24.4.11 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1999ms
rtt min/avg/max/mdev = 0.424/0.653/1.064/0.291 ms
```

Suspending an instance

In this section, you will have the opportunity to compare the differences between pausing (from the previous section of this lab) and suspending an instance.

NOTE: You will be suspending your instance from the command line. You can also suspend an instance from the Horizon GUI, but the command line enables you to use the commands to suspend and then quickly verify the effect of those commands.

- From your terminal session, enter the following command to find the ID of the instance you created and copy it:

```
openstack server list
```

ID	Name	Status	Networks	Image Name
6c939d7f-51e0-4afe-a171-22307c50f093	instance4labs	ACTIVE	net4labs=10.10.1.104, 172.24.4.11	

- Run the following command to suspend the instance:

```
openstack server suspend <instance id>
```

- Run `openstack server list` again and view the current status.

ID	Name	Status	Networks	Image Name
6c939d7f-51e0-4afe-a171-22307c50f093	instance4labs	SUSPENDED	net4labs=10.10.1.104, 172.24.4.11	

- Q: Can you ping the Floating IP address of the instance?

- Enter the following command to view the instance VM image process:

```
ps -e l | grep -v grep | grep <paste in your copied instance ID here>
```

NOTE: There should be no output, indicating that no instance-related process is currently running.

- Run the command below to resume the instance. It might take a minute or two for the instance to become active:

```
openstack server resume <paste in your copied instance ID here>
```

- Run the following command to check the status of your instance and then verify you can ping it:

```
openstack server list
```

- Log in to the Horizon UI as user4labs / hpinvent.

- In the navigation panel, expand **Project** → **Compute** → **Instances**.

10. In the Actions column of the table, click the **drop-down icon** and locate the actions you have just performed at the command line.

<input type="checkbox"/>	Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
<input type="checkbox"/>	instance4labs	-	Floating IPs: • 10.10.1.104 • 172.24.4.11	m1.tiny	keypair4labs	Active	nova	None	Running	23 minutes	Create Snapshot ▾ Snapshot an Instance : captures an image of a running instance without the need to pause that instance. Note: The state of file system is captured but not that of the memory, so it may be necessary to quiesce the instance Pause Instance : stores the state of the VM in RAM. A paused instance continues to run in a frozen state. Suspend Instance : initiates a hypervisor-level suspend operation Shelve Instance : shuts down the instance, and stores it together with associated data and resources (a snapshot is taken if not volume backed). Anything in memory is lost. Lock Instance : prevents a normal (non-admin) user from being able to execute actions on that instance

Displaying 1 item

- [Disassociate Floating IP](#)
- [Attach Interface](#)
- [Detach Interface](#)
- [Edit Instance](#)
- [Attach Volume](#)
- [Detach Volume](#)
- [Update Metadata](#)
- [Edit Security Groups](#)
- [Console](#)
- [View Log](#)
- [Pause Instance](#)
- [Suspend Instance](#)
- [Shelve Instance](#)
- [Resize Instance](#)
- [Lock Instance](#)
- [Unlock Instance](#)
- [Soft Reboot Instance](#)
- [Hard Reboot Instance](#)
- [Shut Off Instance](#)
- [Rebuild Instance](#)
- [Delete Instance](#)

Exercise 5—Creating a snapshot

An **instance snapshot** is an image. The only difference between an image that you upload directly to Glance and the one you create by a snapshot is that an image created by a snapshot has additional properties in the Glance database (for example, the image type and the UUID of the instance that was snapshotted).

Keep in mind that a snapshot captures the state of the file system, but not the state of the memory. To ensure that a snapshot contains the data that you want, before creating your snapshot, you must ensure that running programs have written their contents to disk, and that the file system does not have any "dirty" buffers (where programs have issued the command to write to disk, but the operating system has not yet completed writing data).

To ensure that important services have written their contents to disk (such as, databases), HPE recommends that you read the documentation for those applications to determine which commands to use to have them synchronize their contents to disk. If you are unsure of how to do this, the safest approach is to stop these running services.

1. On the **Project → Compute → Instances** screen in Horizon, click the **arrow button** to display the list of options and select **Create Snapshot**.

Instances

<input type="checkbox"/>	Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
<input type="checkbox"/>	instance4labs	-	• 10.10.1.104 Floating IPs: • 172.24.4.11	m1.tiny	keypair4labs	Active	nova	None	Running	54 minutes	Create Snapshot

Displaying 1 item

2. At the prompt, enter the snapshot name of `snapImageInstance4labs` and click **Create Snapshot**.

Create Snapshot

Snapshot Name *

Description:

A snapshot is an image which preserves the disk state of a running instance.

Create Snapshot

3. Notice that the screen changes to the Images & Snapshots screen, and the snapshot appears in the Images list.

Images							
	Name	Type	Status	Visibility	Protected	Disk Format	Size
<input type="checkbox"/>	ciros-0.3.4-x86_64-uec	Image	Active	Public	No	AMI	24.00 MB
<input type="checkbox"/>	ciros-0.3.4-x86_64-uec-kernel	Image	Active	Public	No	AKI	4.75 MB
<input type="checkbox"/>	ciros-0.3.4-x86_64-uec-ramdisk	Image	Active	Public	No	ARI	3.57 MB
<input type="checkbox"/>	image4labs	Image	Active	Public	No	QCOW2	12.67 MB
<input type="checkbox"/>	snapImage4labs	Image	Active	Private	No	QCOW2	0 bytes

Displaying 5 items

4. Q: Does snapshotting an instance start a new instance?

Instances

<input type="button"/> Instance Name = <input type="text"/> Filter <input type="button"/> Launch Instance <input type="button"/> Delete Instances <input type="button"/> More Actions ▾											
<input type="checkbox"/>	Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
<input type="checkbox"/>	snappedImage4labs	snapImage4labs • 10.10.1.106	m1.tiny	keypair4labs	Active	nova	None	Running	0 minutes	<input type="button"/> Create Snapshot ▾	
		• 10.10.1.104									
<input type="checkbox"/>	instance4labs	-	Floating IPs:	m1.tiny	keypair4labs	Active	nova	None	Running	1 hour, 2 minutes	<input type="button"/> Create Snapshot ▾
			• 172.24.4.11								

Displaying 2 items

5. From the image screen, create an instance from the snapImage4labs image with the following parameters:

- Name: **snappedImage4labs**
- Flavor: **m1.tiny**
- Keypair: **keypair4labs**
- Network: **net4labs**
- The Floating IP address from a public pool

6. Please let your instructor know you have completed Lab 6.

Fundamentals of OpenStack® Technology

Lab 07—OpenStack® Cinder (Block Storage)

Objectives

After completing this lab, you should be able to:

- Create a volume and attach it to an instance
- Create an instance that boots from a volume

Requirements

- Completion of Lab 6
- Access to the HPEVL equipment for completing the OpenStack® labs
- Approximately 45 minutes to complete

Introduction

Cinder provides an infrastructure for managing volumes in OpenStack®. It was originally a Nova component called “nova-volume,” but it has become an independent project since the Folsom release of OpenStack®.

Let us take a look at what you have covered in the labs up to this point:

- Lab 2, Management Tools, introduced you to the OpenStack® CLI and the Horizon UI. You were able to use some of the more common OpenStack® CLI commands, a REST client browser plug-in, and CURL to make API requests of OpenStack® services.
- In Lab 3, Keystone, you created the **project4labs** project and **user4labs** user, which will be used in this lab.
- In Lab 4, Glance, you created the **image4labs** image, which will be used in this lab.
- In Lab 5, Neutron, you created the **net4labs** network, **subnet4labs** subnet, and **router4labs** router, which will be used in this lab.
- In Lab 6: Nova, you created an instance called **instance4labs**.

In this lab, you will use the components created in each of the previous labs to create volumes, attach a volume to an existing instance, and create an instance that boots from a volume.

Exercise 1: Creating a Cinder volume from the CLI

1. In a terminal window on your assigned system, enter the following command:

```
cd ~/devstack
```

2. Set up the necessary environment variables with the following command:

```
source ~/devstack/openrc user4labs project4labs
```

3. Create a Cinder volume with the following command:

```
openstack volume create --size 1 vol4labs
```

```
stack@stack:~/devstack$ openstack volume create --size 1 vol4labs
+-----+-----+
| Field | Value |
+-----+-----+
| attachments | [] |
| availability_zone | nova |
| bootable | false |
| consistencygroup_id | None |
| created_at | 2016-12-27T15:48:51.447522 |
| description | None |
| encrypted | False |
| id | 16bc38ce-cabc-4d0e-9e81-06a1828dfffc0 |
| multiattach | False |
| name | vol4labs |
| properties | |
| replication_status | disabled |
| size | 1 |
| snapshot_id | None |
| source_valid | None |
| status | creating |
| type | lvmdriver-1 |
| updated_at | None |
| user_id | 49a0998341994269905dbe98adc6e6c2 |
+-----+
```

4. Q: Is this newly created volume bootable? (Hint: Look at the above screenshot.)

5. At the command prompt, type `openstack volume list` to verify that the volume was successfully created.

```
stack@stack:~/devstack$ openstack volume list
+-----+-----+-----+-----+
| ID | Display Name | Status | Size | Attached to |
+-----+-----+-----+-----+
| 0766b97d-0cd5-4414-9564-f3382fd1ff9f | | in-use | 1 | Attached to snappedImageInstance4Labs on /dev/vda |
| 16bc38ce-cabc-4d0e-9e81-06a1828dfffc0 | vol4labs | available | 1 |
```

NOTE: Note the volume ID, because the volume ID is required when associating the instance to a volume.

6. To view the available instances, enter the `openstack server list` command.

```
stack@stack:~/devstack$ openstack server list
+-----+-----+-----+-----+
| ID | Name | Status | Networks | Image Name |
+-----+-----+-----+-----+
| ac106163-d6a7-4349-a9ac-d864f8211855 | snappedImageInstance4Labs | ACTIVE | net4labs=10.10.1.103 | |
| ec4d5049-d68f-4644-90fc-a3e2746e8654 | instance4labs | ACTIVE | net4labs=10.10.1.107, 172.24.4.4 | image4labs |
```

7. To add the volume to the **instance4labs** instance (server), use the following command line syntax:

```
openstack server add volume <server id> <volume id>
```

8. To verify that the volume is attached to the instance, enter the `openstack volume list` command.

```
stack@stack:~/devstack$ openstack volume list
+-----+-----+-----+-----+
| ID      | Display Name | Status | Size | Attached to
+-----+-----+-----+-----+
| 0766b97d-0cd5-4414-9564-f3382fd1ff9f |          | in-use |   1 | Attached to snappedImageInstance4Labs on /dev/vda |
| 16bc38ce-cabc-4d0e-9e81-06a1828dffc0 | vol4labs | in-use |   1 | Attached to instance4labs on /dev/vdb |
+-----+-----+-----+-----+
```

9. To detach the volume from the instance, use the following syntax:

```
openstack server remove volume <server id> <volume id>
```

```
stack@stack:~/devstack$ openstack volume list
+-----+-----+-----+-----+
| ID      | Display Name | Status | Size | Attached to
+-----+-----+-----+-----+
| 0766b97d-0cd5-4414-9564-f3382fd1ff9f |          | in-use |   1 | Attached to snappedImageInstance4Labs on /dev/vda |
| 16bc38ce-cabc-4d0e-9e81-06a1828dffc0 | vol4labs | available |  1 |          |
+-----+-----+-----+-----+
```

Exercise 2—Creating a bootable Cinder volume

Cinder can create a bootable volume from an image stored in Glance.

- Enter the `openstack image list` command to view the ID of the available Glance images.

ID	Name	Status
ebb0ed60-d69d-4005-80bf-0d701d27128f	snapshotInstance4labs	active
96296ac1-97bb-44a4-828c-728e4b8eee32	image4labs	active
b300fe18-3e37-41f3-92b3-0457d89e83db	cirros-0.3.4-x86_64-uec	active
0aa2869e-7a3d-41e7-ab38-ace90b13e73f	cirros-0.3.4-x86_64-uec-ramdisk	active
20aef97f-b716-44b3-a32a-e90aac2557b5	cirros-0.3.4-x86_64-uec-kernel	active
1555dad9-abea-4d4e-bd72-4cc952ff13bb	Fedora-Cloud-Base-25-1.3.x86_64	active

- Use the `openstack volume create --size <in GB> --description <description of vol> --image <image_id> <name of vol>` syntax to create the following volume using the `image4labs` image. (Hint: Use quotes around variables that contain spaces.)

Field	Value
attachments	[]
availability_zone	nova
bootable	false
consistencygroup_id	None
created_at	2016-12-27T17:00:25.181453
description	bootable vol 4 labs
encrypted	False
id	93b112eb-e144-4180-a624-86fc0eb537f3
multiattach	False
name	bootbol4labs
properties	
replication_status	disabled
size	1
snapshot_id	None
source_volid	None
status	creating
type	lvmdriver-1
updated_at	None
user_id	49a0998341994269905dbe98adc6e6c2

- To display the status of the previous command, enter the `openstack volume list --long` command.

ID	Display Name	Status	Size	Type	Bootable
93b112eb-e144-4180-a624-86fc0eb537f3	bootbol4labs	available	1	lvmdriver-1	true
0766b97d-0cd5-4414-9564-f3382fd1ff9f		in-use	1	lvmdriver-1	true
16bc38ce-cabc-4d0e-9e81-06a1828dff0	vol4labs	available	1	lvmdriver-1	false

- To boot from the volume, enter the command `openstack server create --volume <volume id of bootable volume> --flavor m1.tiny --key-name keypair4labs instfromvol`.

5. Run `openstack server list` and verify that the new instance is in the ACTIVE state.

ID	Name	Status	Networks	Image Name
3a9e39f3-86e8-43a2-8ef0-e	instfromvol	ACTIVE	net4labs=10.10.1.112	
6d5f5387387				
ac106163-d6a7-4349-a9ac-	snappedImageInstance4Labs	ACTIVE	net4labs=10.10.1.103	
d864f8211855				
ec4d5049-d68f-4644-90fc-	instance4labs	ACTIVE	net4labs=10.10.1.107, 172.24.4.4	image4labs
a3e2746e8654				

6. Proceed with the next exercise.

Exercise 3—Using Horizon to manage Cinder volumes

Cinder options in the Horizon UI

Now that you have a general idea of the command necessary to create and manage volumes, let us take a look at what volumes look like in Horizon.

1. Access the Horizon UI and log in with the credentials of admin/hpinvent, using the domain of default.
2. In the navigation pane, expand **Admin** → **System** → **Volumes**. The Volumes panel offers three tabs:
 - Volumes
 - Volume Types
 - Volume Snapshots
3. The Volumes panel provides detailed information about all volumes and their attachments to the running instances. You can use the links to get more details, delete volumes that are not in use, or change the status of a volume by clicking **Update Volume Status**.

Project	Host	Name	Size	Status	Type	Attached To	Bootable	Encrypted	Actions
project4labs	stack@lvmdriver-1#lvmdriver-1	bootbol4labs	1GiB	In-use	lvmdriver-1	Attached to instfro mvol on /dev/vda	Yes	No	Update Volume Status
project4labs	stack@lvmdriver-1#lvmdriver-1	0766b97d-0cd5-4414-9564-f3382fd1ff9f	1GiB	In-use	lvmdriver-1	Attached to snappe dimage instance4La bs on /dev/vda	Yes	No	Update Volume Status
project4labs	stack@lvmdriver-1#lvmdriver-1	vol4labs	1GiB	Available	lvmdriver-1		No	No	Delete Volume

4. At the Update Volume State dialog, expand the **Status** drop-down list to examine available options. When you have finished, press **Cancel**.

Status *

in-use

Select a new status

Attaching

Available

Creating

Description:

The status of a volume is normally managed automatically. In some circumstances an administrator may need to explicitly update the status value. This is equivalent to the `cinder reset-state` command.

Cancel **Update Status**

5. Select the **Volume Types** tab. The Volume Types tab mirrors the functionality of the `openstack volume` commands. Multiple types require multiple Cinder backends. This environment is configured with a single LVM backend, which is already configured and available.

Name	Description	Associated QoS Spec	Encryption	Public	Actions
lvmdriver-1	-	-	Yes		Create Encryption

Displaying 1 item

QoS Specs			
Name	Consumer	Specs	Actions
No items to display.			

[+ Create QoS Spec](#)

6. QOS Specs can be associated with volume types. They are used to map to a set of quality of service capabilities requested by the volume owner. This is equivalent to the Cinder `qos-create` command. After the QOS Spec is created, click the **Manage Specs** button to manage the key-value specs for the QOS Spec.

Each QOS Spec entity has a "consumer" value that indicates where the administrator would like the QOS policy to be enforced. This value can be "front-end" (Nova Compute), "back-end" (Cinder backend), or both.

When you select **Create Encryption**, the system creates an encrypted volume. Encryption is not supported on your DevStack system.

7. The Volume Snapshots tab lists the available snapshots made from Cinder-based volumes.

Project	Host	Name	Description	Size	Status	Volume Name	Actions
No items to display.							

8. Sign out from Horizon as admin and log in again using the credentials of `user4labs / hpinvent` with the domain of `default`.

9. In the navigation panel, expand **Project** → **Compute** → **Volumes**. Note that the Volumes panel for Project looks similarly to the Volumes Admin panel. It lacks the capability to manage Volume Types, but provides you the option to work with available volumes. To view all of the available actions, click the arrow next to the **Edit Volume** action.

Volumes

The screenshot shows the 'Volumes' page in the OpenStack Cinder interface. At the top, there are tabs for 'Volumes' (which is selected) and 'Volume Snapshots'. Below the tabs are buttons for 'Filter', '+ Create Volume', 'Accept Transfer', and 'Delete Volumes'. A table lists three volumes:

	Name	Description	Size	Status	Type	Attached To	Availability Zone	Bootable	Encrypted	Actions
<input type="checkbox"/>	bootbol4labs	bootable vol 4 labs	1GiB	In-use	lvmdriver-1	Attached to instfromvol on /dev/vda	nova	Yes	No	<button>Edit Volume</button> ▾
<input type="checkbox"/>	0766b97d-0cd5-44f 14-9564-f3382fd1ff9	-	1GiB	In-use	lvmdriver-1	Attached to snappedImage Instance4Labs on /dev/vda	nova	Yes	No	<button>Edit Volume</button> ▾
<input type="checkbox"/>	vol4labs	-	1GiB	Available	lvmdriver-1		nova	No	No	<button>Edit Volume</button> ▾

Below the table, it says 'Displaying 3 items'. The 'Actions' column for the third volume has a dropdown menu open, highlighted with a red box, containing the following options:

- Extend Volume
- Manage Attachments
- Create Snapshot
- Change Volume Type
- Upload to Image
- Create Transfer
- Delete Volume

Creating an instance from a bootable Cinder volume

At the top of the Volumes panel, click the **Create Volume** button at the right side.

NOTE: Options available in this dialog change dynamically. The fields with the asterisk sign appended are mandatory (such as Volume Name *). Combining these values lets you achieve the same functionality as using the command line.

1. To create a new instance from an existing image, at the **Project → Compute → Volumes** page, click the **Create Volume** button.

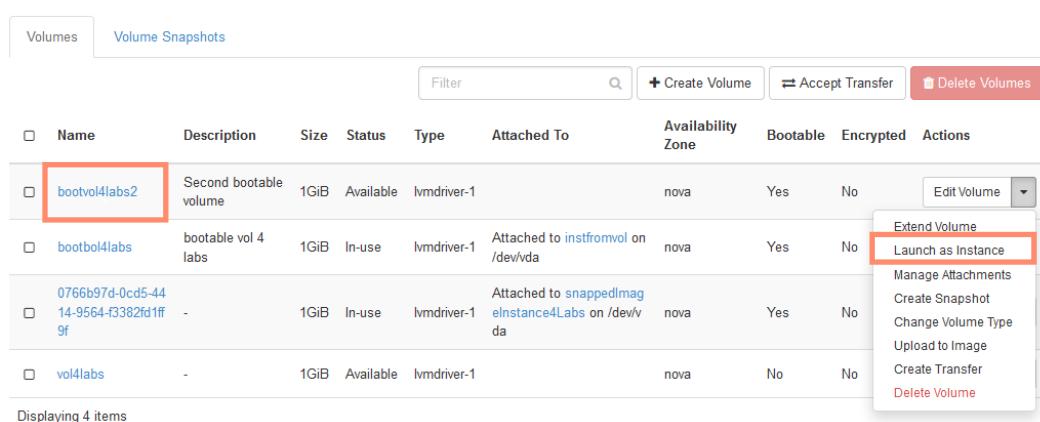
The screenshot shows the OpenStack Compute Volumes interface. The left sidebar has sections for Project, Compute, Overview, Instances, and Volumes (which is selected and highlighted in blue). The main area shows a table of volumes with columns: Name, Description, Size, Status, Type, Attached To, Availability Zone, Bootable, Encrypted, and Actions. Three volumes are listed: 'bootvol4labs', '0766b97d-0cd5-44f', and 'vol4labs'. A red box highlights the '+ Create Volume' button in the top right of the toolbar. The URL in the browser bar is 'Project / Compute / Volumes'.

2. Set the options in the Create Volume dialog as shown on the image below. Note that the panel shows a graphical representation of Volume Limits. Click **Create Volume** when ready.

The screenshot shows the 'Create Volume' dialog. It includes fields for Volume Name (set to 'bootvol4labs2'), Description (set to 'Second bootable volume'), Volume Source (set to 'Image'), Use image as a source (set to 'image4labs (12.7 MB)'), Type (set to 'lvmdriver-1'), Size (GiB) (set to '1'), Availability Zone (set to 'nova'), and Volume Limits (showing Total Gibibytes (3 GiB) and Number of Volumes (3)). A red box highlights the 'Volume Limits' section. The 'Create Volume' button is at the bottom right.

3. At the Volumes panel, wait for the new volume to appear. Then, click the arrow next to **Edit Volume** and select **Launch as Instance** to create a new instance.

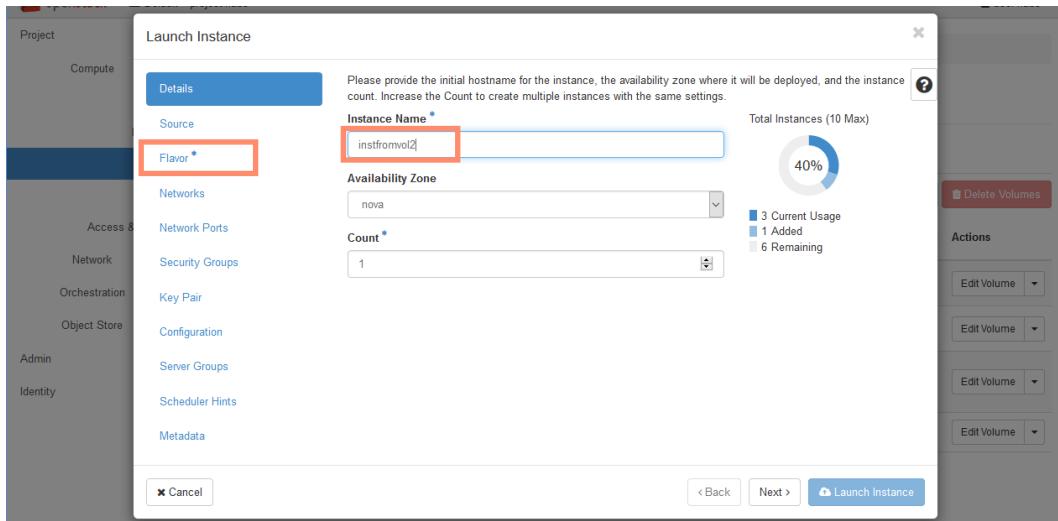
Volumes



<input type="checkbox"/>	Name	Description	Size	Status	Type	Attached To	Availability Zone	Bootable	Encrypted	Actions
<input type="checkbox"/>	bootvol4labs2	Second bootable volume	1GiB	Available	lvmdriver-1		nova	Yes	No	<button>Edit Volume</button>
<input type="checkbox"/>	bootvol4labs	bootable vol 4 labs	1GiB	In-use	lvmdriver-1	Attached to instfromvol on /dev/vda	nova	Yes	No	<button>Edit Volume</button>
<input type="checkbox"/>	0766b97d-0cd5-449f-9f56-f3382fd1ff9f	-	1GiB	In-use	lvmdriver-1	Attached to snappedImage elinstance4Labs on /dev/vda	nova	Yes	No	<button>Edit Volume</button>
<input type="checkbox"/>	vol4labs	-	1GiB	Available	lvmdriver-1		nova	No	No	<button>Edit Volume</button>

Displaying 4 items

4. Provide a name for the new instance and click **Flavor** when ready.



Project

Compute

Access & Network

Orchestration

Object Store

Admin

Identity

Launch Instance

Details

Source

Flavor *

Instance Name *

instfromvol2

Availability Zone

nova

Count *

1

Total Instances (10 Max)

40%

3 Current Usage
1 Added
6 Remaining

Actions

Edit Volume

Edit Volume

Edit Volume

Cancel Back Next Launch Instance

5. Select the **m1.tiny** flavor and click **Launch Instance**.

openstack Default • project4labs

Project

Compute

Access & Security

Network

Orchestration

Object Store

Admin

Identity

Launch Instance

Details

Flavors manage the sizing for the compute, memory and storage capacity of the instance.

Allocated

Name	VCPUS	RAM	Total Disk	Root Disk	Ephemeral Disk	Public
m1.tiny	1	512 MB	1 GB	1 GB	0 GB	Yes

Source

Flavor

Networks

Network Ports

Security Groups

Key Pair

Configuration

Server Groups

Scheduler Hints

Metadata

Available 9

Select one

Click here for filters.

Name	VCPUS	RAM	Total Disk	Root Disk	Ephemeral Disk	Public
m1.small	1	2 GB	20 GB	20 GB	0 GB	Yes
m1.medium	2	4 GB	40 GB	40 GB	0 GB	Yes
m1.large	4	8 GB	80 GB	80 GB	0 GB	Yes
m1.xlarge	8	16 GB	160 GB	160 GB	0 GB	Yes
cirros256	1	256 MB	0 GB	0 GB	0 GB	Yes
ds512M	1	512 MB	5 GB	5 GB	0 GB	Yes
ds1G	1	1 GB	10 GB	10 GB	0 GB	Yes
ds2G	2	2 GB	10 GB	10 GB	0 GB	Yes
ds4G	4	4 GB	20 GB	20 GB	0 GB	Yes

Delete Volumes

Actions

Edit Volume

Edit Volume

Edit Volume

Edit Volume

Cancel

Back

Next >

Launch Instance

- Select **Project** → **Compute** → **Instances**. At the Instances panel, click the instance name for the newly created instance.

Instances

		Instance Name = <input type="text"/>			Filter		Launch Instance		Delete Instances		More Actions	
	Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions	
<input type="checkbox"/>	instfromvol2	-	• 10.10.1.106	m1.tiny	keypair4labs	Active	nova	None	Running	0 minutes	Create Snapshot	<input type="button" value="▼"/>
<input type="checkbox"/>	instfromvol	-	• 10.10.1.112	m1.tiny	keypair4labs	Active	nova	None	Running	1 hour, 9 minutes	Create Snapshot	<input type="button" value="▼"/>
<input type="checkbox"/>	snappedImageIn stance4Labs	-	• 10.10.1.103	m1.tiny	keypair4labs	Active	nova	None	Running	2 hours, 7 minutes	Create Snapshot	<input type="button" value="▼"/>

7. At the Volume Details screen, scroll down and check the attached volumes list.

The screenshot shows the 'Overview' tab selected for a volume named 'instfromvol2'. The volume details are as follows:

Name	instfromvol2
ID	e4a066c7-c789-47dd-8759-12b99437b4f4
Status	Active
Availability Zone	nova
Created	27 Dec 2016, 6:18 p.m.
Time Since Created	2 minutes

Specs

Flavour Name	m1.tiny
Flavour ID	1
RAM	512MB
VCPUs	1 VCPU
Disk	1GB

IP Addresses

Net4labs	10.10.1.106
----------	-------------

Security Groups

default	<ul style="list-style-type: none">ALLOW IPv4 22/tcp from 0.0.0.0/0ALLOW IPv4 icmp from 0.0.0.0/0ALLOW IPv6 from defaultALLOW IPv4 from defaultALLOW IPv4 to 0.0.0.0/0ALLOW IPv6 to ::/0
---------	--

Metadata

Key Name	keypair4labs
Image	None

Volumes Attached

Attached To	bootvol4labs2 on /dev/vda
-------------	---------------------------

8. Please let the instructor know you have completed Lab 7.

Fundamentals of OpenStack® Technology

Lab 08—OpenStack® Object Storage (Swift)

Objectives

After completing this lab, you should be able to:

- Describe the major features of the OpenStack® Object Storage service
- Use Swift to manage storage objects

Requirements

- Completion of Lab 7
- Access to the HPE Virtual Lab equipment for completing the OpenStack® labs
- Approximately 30 minutes to complete

Exercise 1: Using the OpenStack® Object Storage service

In this lab, you will log in to the OpenStack® Dashboard and perform operations using the Object Storage service.

1. Start the Firefox web browser and browse to the OpenStack® dashboard.
2. Log in to the default domain as `user4labs/hpinvent` user, and select **project4labs**.
3. In the navigation panel, browse to **Project → Object Store → Containers**.

A **container** is a storage compartment for your data and provides a way for you to organize your data. You can think of a container as a folder in Windows or a directory in UNIX. The primary difference between a container and these other file system concepts is that containers cannot be nested. You can, however, create an unlimited number of containers within your account. Data must be stored in a container, so you must have at least one container defined in your account before uploading data.

4. Click **+ Container**.

The screenshot shows the OpenStack Dashboard interface. The top navigation bar includes the OpenStack logo, project selection (Default • project4labs), and user information (user4labs). The left sidebar has a dropdown menu under 'Project' and links for Compute, Network, Orchestration, and Object Store. Under 'Object Store', the 'Containers' tab is active, indicated by a blue background. To its right is a red-bordered button labeled '+ Container'. A callout box to the right of the button says 'Select a container to browse.'

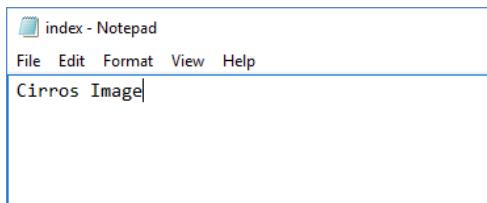
5. At the Create Container dialog, use the parameters shown in the following screenshot. Click **+Create**.

The screenshot shows a modal dialog titled 'Create Container'. Inside, there's a field labeled 'Container Name *' with 'MyImages' entered, which is also highlighted with a red box. Below it is a section for 'Container Access' with a checkbox labeled 'Public' checked. To the right of these fields is a descriptive note: 'A container is a storage compartment for your data and provides a way for you to organize your data. You can think of a container as a folder in Windows® or a directory in UNIX®. The primary difference between a container and these other file system concepts is that containers cannot be nested. You can, however, create an unlimited number of containers within your account. Data must be stored in a container so you must have at least one container defined in your account prior to uploading data.' At the bottom of the dialog are two buttons: 'Cancel' and '+ Create'.

6. The Containers panel displays a newly created container. Click the **MyImages** container to display the details.

Containers

7. At the landing desktop, use the Notepad editor to create a file with the below content. Save the file to the desktop using the name of **index.txt**.



8. Click the **Upload Object icon** next to the filter field.

Containers

9. At the Upload Objects dialog, click **Browse**. Using the file browser on your landing desktop, select the file **index.txt** you have created previously. Provide the name of **The Index of Images** and click **Upload File**.

10. Verify the object has been loaded to the **MyImages** container.

The screenshot shows the OpenStack Swift interface. On the left, there's a navigation sidebar with 'Project', 'Compute', 'Network', 'Orchestration', and 'Object Store' sections. Under 'Object Store', 'Containers' is selected. In the main area, the path 'Project / Object Store / Containers' is shown, followed by 'Containers'. A 'MyImages' container is selected, indicated by a blue border. The container details show: Object Count: 1, Size: 12 bytes, and Date Created: Dec 27, 2016. Below this, a table lists the objects in the container, with 'The Index of Images' highlighted by a red box. The table includes columns for Name, Size, and actions like Download and Delete. A note at the bottom says 'Displaying 1 item'.

11. To create a pseudo-folder under the **MyImages** container, click the **MyImages** link in the Containers table, then click **+ Folder**, name it **Cirros** and click **+Create Folder**.

This screenshot shows a modal dialog titled 'Create Folder In: MyImages'. It has a 'Folder Name' input field containing 'Cirros', which is also highlighted with a red box. Below the input field is a note: 'Note: Delimiters ('/') are allowed in the folder name to create deep folders.' At the bottom right of the dialog are 'Cancel' and '+ Create Folder' buttons, with the latter also highlighted with a red box.

12. Verify that the Cirros pseudo-folder was created as shown below.

The screenshot shows the 'Containers' page again. The 'MyImages' container is selected. The table now shows two items: 'Cirros' (highlighted with a red box) and 'The Index of Images'. The 'Cirros' entry is marked as a 'Folder' and has a 'Delete' button next to it. The table includes columns for Name, Size, and actions like Download and Delete. A note at the bottom says 'Displaying 2 items'.

13. Click the **Cirros** link.

The screenshot shows the OpenStack Swift interface. On the left, there's a navigation sidebar with categories like Project, Compute, Network, Orchestration, Object Store, Admin, and Identity. Under Object Store, 'Containers' is selected. In the main area, there's a breadcrumb trail: Project / Object Store / Containers. Below it, a container named 'MyImages' is listed under the 'Cirros' ring. The container details show: Object Count: 2, Size: 12 bytes, Date Created: Dec 27, 2016, and Public Access: link. To the right of the container list is a search bar with a placeholder 'Click here for filters...', and buttons for upload (+), folder creation (+), and delete (red). A red arrow points to the 'Cirros' link next to the container name.

14. Click the **upload icon** and upload the CirrOS image from the landing desktop. The image can be found on C:\Classfiles\Images\cirros-0.3.4-x86_64-disk.img.

This screenshot shows the same OpenStack Swift interface as the previous one, but now with an uploaded file. The container 'MyImages' now has an object count of 3. The newly added object is 'cirros-0.3.4-x86_64-disk.img', which is 12.67 MB in size. A red box highlights this new object entry. To the right of the object list are download and other file management buttons.

15. Click the **Download** button associated with the CirrOS image and download the image to the landing server desktop to save and then drag and drop from default download location to the desktop.

Files can easily be stored in Swift, then recovered, and they are in their original form (not encrypted).

16. Now that you have stored files to Swift, use the OpenStack® CLI to locate the files and better understand how Swift rings works.

17. Sign out of the Horizon GUI and close the browser window.

Exercise 2: Exploring the Swift configuration

In this lab, you will explore the Swift configuration.

1. Start a terminal session window to your assigned VM and log in with `stack /hpinvent`.

2. To set the necessary environment and other settings, enter the following CLI command:

```
source ~/devstack/openrc user4labs project4labs
```

3. To list the available containers, enter the `openstack container list` command.

```
stack@stack:~$ openstack container list
+-----+
| Name   |
+-----+
| MyImages |
+-----+
```

4. To view the information about the container, enter the `openstack container show <container name>` command. Take a note to review the information about the account name and number of objects.

```
stack@stack:~$ openstack container show MyImages
+-----+
| Field      | Value
+-----+
| account    | AUTH_7b853dd380da4c5794e05f87031479af
| bytes_used | 13287948
| container  | MyImages
| object_count| 3
| read_acl   | .r:*,.rlistings
+-----+
```

5. You will need the account name later, so remember how to obtain that piece of information.

6. **Q:** How many containers exist?

7. **Q:** How many objects exist?

8. To display the details about all objects, enter the `openstack object list <container> --long` command.

```
stack@stack:~$ openstack object list MyImages --long
+-----+-----+-----+-----+
| Name        | Bytes | Hash                | Content Type      | Last Modified   |
+-----+-----+-----+-----+
| Cirros/     | 0     | d41d8cd98f00b204e980 | application/octet- | 2016-12-27T20:49:59.233 |
|             |       | 0998ecf8427e          | stream              | 250
| Cirros/cirros-0.3.4- | 13287936 | ee1eca47dc88f4879d8a | application/octet- | 2016-12-27T20:56:17.145 |
| x86_64-disk.img |       | 229cc70a07c6          | stream              | 860
| The Index of Images | 12    | 2c65a1150cb00cf2af09 | application/octet- | 2016-12-27T20:44:20.972 |
|             |       | 762585b0a95a          | stream              | 120
+-----+-----+-----+-----+
```

9. To display a list of the rings, enter the `ls /etc/swift/*.gz` command.

```
stack@stack:~$ ls /etc/swift/*.gz
/etc/swift/account.ring.gz  /etc/swift/container.ring.gz  /etc/swift/object.ring.gz
```

NOTE: The output should display a container ring file, an account ring, and an object ring. Because you will be seeking information about an account in the next step, you will use the account ring file name as a parameter.

In the following syntax, the `swift-get-nodes` command is used to get detailed information about an object:

```
swift-get-nodes <ring> <auth> <container> <object>
```

10. Enter the `swift-get-nodes` command to get information on the **project4labs** account using the account ring and the AUTH account string from `openstack container show` (you must replace the AUTH_* value with the one from your DevStack host in step 4).

```
stack@stack:~$ openstack container show MyImages
+-----+-----+
| Field | Value |
+-----+-----+
| account | AUTH_7b853dd380da4c5794e05f87031479af |
| bytes_used | 13287948 |
| container | MyImages |
| object_count | 3 |
| read_acl | .r:*,.rlistings |
+-----+
stack@stack:~$ swift-get-nodes /etc/swift/account.ring.gz AUTH_7b853dd380da4c5794e05f87031479af
```



```
Account      AUTH_7b853dd380da4c5794e05f87031479af
Container    None
Object       None
```


Partition	299
Hash	95bfdd08dc3413ecd9830f4ac83a39fd
Server:Port	127.0.0.1:6612
Device	sdb1

```
curl -g -I -XHEAD "http://127.0.0.1:6612/sdb1/299/AUTH_7b853dd380da4c5794e05f87031479af"
```

```
Use your own device location of servers:
such as "export DEVICE=/srv/node"
ssh 127.0.0.1 "ls -lah ${DEVICE:-/srv/node*/}/sdb1/accounts/299/9fd/95bfdd08dc3413ecd9830f4ac83a39fd"
note: '/srv/node*' is used as default value of `devices`, the real value is set in the config file on each storage node.
```

Because you are using a DevStack one-node system, your Swift installation creates only one copy of each object, and all of our storage is local. The only location of your account data is on the server 127.0.0.1 (local host), port 6612, (virtual) device sdb1.

11. Enter the `swift-get-nodes` command to get information on the **MyImages** container using the container ring and the AUTH account string from `openstack container show` (you must replace the `AUTH_*` value with the one from your DevStack host in step 4).

```
stack@stack:~$ swift-get-nodes /etc/swift/container.ring.gz AUTH_7b853dd380da4c5794e05f87031479af MyImages

Account      AUTH_7b853dd380da4c5794e05f87031479af
Container    MyImages
Object       None

Partition    330
Hash        a53ae031a21d535ba0eb98691d97e762

Server:Port Device      127.0.0.1:6611 sdb1

curl -g -I -XHEAD "http://127.0.0.1:6611/sdb1/330/AUTH_7b853dd380da4c5794e05f87031479af/MyImages"

Use your own device location of servers:
such as "export DEVICE=/srv/node"
ssh 127.0.0.1 "ls -lah ${DEVICE:-/srv/node*/}/sdb1/containers/330/a53ae031a21d535ba0eb98691d97e762"

note: `/srv/node*` is used as default value of `devices`, the real value is set in the config file on each storage node.
```

12. Enter the following command to determine the Swift mount point on the lab VM (which is the storage node for your environment):

```
df | grep loop
```

```
stack@stack:~$ df | grep loop
/dev/loop0           2086912  306092   1780820  15% /opt/stack/data/swift/drives/sdb1
```

NOTE: The actual mount point includes a `/1` at the end, so in your example that is `/opt/stack/data/swift/drives/sdb1/1`.

13. Enter the `swift-get-nodes` command to get information on the Cirros container using the container ring and the AUTH account string from `openstack container show` (you must replace the `AUTH_*` value with the one from your DevStack host in step 4).

```
stack@stack:~$ swift-get-nodes /etc/swift/object.ring.gz AUTH_7b853dd380da4c5794e05f87031479af MyImages "The Index of Images"

Account      AUTH_7b853dd380da4c5794e05f87031479af
Container    MyImages
Object       The Index of Images

Partition    486
Hash        f361e6ad3f6466c1e869540844572759

Server:Port Device      127.0.0.1:6613 sdb1

curl -g -I -XHEAD "http://127.0.0.1:6613/sdb1/486/AUTH_7b853dd380da4c5794e05f87031479af/MyImages/The%20Index%20of%20Images"

Use your own device location of servers:
such as "export DEVICE=/srv/node"
ssh 127.0.0.1 "ls -lah ${DEVICE:-/srv/node*/}/sdb1/objects/486/759/f361e6ad3f6466c1e869540844572759"

note: `/srv/node*` is used as default value of `devices`, the real value is set in the config file on each storage node.
```

14. Now you can combine information from steps 13 and 12 to locate the object on the storage node.

Type the following `ls -l` command to view the contents of the directory and display the object name.

```
stack@stack:~$ ls -l /opt/stack/data/swift/drives/sdb1/1/sdb1/objects/486/759/f361e6ad3f6466c1e869540844572759/
total 8
-rw-r-xr-x 1 stack stack 12 Dec 27 21:44 1482871460.97212.data
```

15. Now that you have the full path to the object on the Swift node, you can obtain the object information.

Use the `swift-object-info` command to display the information about the object.

```
stack@stack:~$ swift-object-info /opt/stack/data/swift/drives/sdb1/1/sdb1/objects/486/759/f361e6ad3f6466c1e869540844572759/1482871460.97212.data
Path: /AUTH_7b853dd380da4c5794e05f87031479af/MyImages/The Index of Images
Account: AUTH_7b853dd380da4c5794e05f87031479af
Container: MyImages
Object: The Index of Images
Object hash: f361e6ad3f6466c1e869540844572759
Content-Type: application/octet-stream
Timestamp: 2016-12-27T20:44:20.972120 (1482871460.97212)
System Metadata:
  No metadata found
User Metadata:
  X-Object-Meta-Orig-Filename: index.txt
Other Metadata:
  No metadata found
ETag: 2c65a1150cb00cf2af09762585b0a95a (valid)
Content-Length: 12 (valid)
Partition      486
Hash          f361e6ad3f6466c1e869540844572759

Server:Port Device      127.0.0.1:6613 sdb1

curl -g -I -XHEAD "http://127.0.0.1:6613/sdb1/486/AUTH_7b853dd380da4c5794e05f87031479af/MyImages/The%20Index%20of%20Images" -H "X-Backend-Storage-Policy-Index: 0"
```

Use your own device location of servers:
such as "export DEVICE=/srv/node"

```
ssh 127.0.0.1 "ls -lah ${DEVICE:-/srv/node*/}/sdb1/objects/486/759/f361e6ad3f6466c1e869540844572759"
```

note: `/srv/node*` is used as default value of `devices`, the real value is set in the config file on each storage node.

16. The `swift-account-audit` program provides a consistency check for accounts, containers, and objects. Run this with the Account ID that you copied and saved in step 4 of this exercise.

```
stack@stack:~$ swift-account-audit AUTH_7b853dd380da4c5794e05f87031479af
Auditing account "AUTH_7b853dd380da4c5794e05f87031479af"
Auditing container "MyImages"

Accounts checked: 1
Containers checked: 1
Objects checked: 3
```

17. Perform the audit check on one of your objects.

```
stack@stack:~$ swift-account-audit AUTH_7b853dd380da4c5794e05f87031479af/MyImages/The\ Index\ of\ Images
Auditing container "MyImages"
Auditing account "AUTH_7b853dd380da4c5794e05f87031479af"

Accounts checked: 1
Containers checked: 1
Objects checked: 1
```

Exercise 3: Examine the Glance images stored in Swift

Glance images in DevStack are by default stored in Swift. In this exercise, you will use the Swift CLI to examine a Glance image stored in Swift.

The **service project** is a special project created in the Keystone system that is designed for the use of the core OpenStack® services.

1. Configure the environment variables:

```
source ~/devstack/openrc admin admin
```

2. Run `openstack project list` to view the projects that exist on your DevStack.

ID	Name
9dc7de8f5e4d4ed09b3598955c087edc	admin
fb7685f07c7a4a4790b7743308493db2	alt_demo
26c5e2a9268b40c199ec332ee71482a3	demo
35da87f4c79c4a2588bc67f1337a27d2	invisible_to_admin
7b853dd380da4c5794e05f87031479af	project4labs
ff7401ba42aa4c588305cbbb6cc7f792	service
8cc9a70c56d6479e8025e32c6fd02007	swiftprojecttest1
20db0f3c9f934538ae08ded8af38deb1	swiftprojecttest2

3. Run `openstack user list` to view a list of users.

ID	Name
b3dac5291b5c4b9b8cfdbaa2b1ab39a8	admin
2503be441dcc4ea2b69574912097eab3	demo
8ddaa31d7ced4b09bcd2576b8335a630	alt_demo
f315f9f7c561495fa31ad3823bbd1ba1	nova
0d687390b0424c6fb9826614244c1c82	glance
782466468c5542ed9c2e09cd9c43d2d9	glance-swift
1b332b7e80244a0b858cd18a726a5d56	cinder
a7abdabda104adf9b83ad66d052a472	neutron
0d5fe63cdf3b4c6d93256e56acf6792	swift
96ff9097168447b38b7ff5ccee6723f1	swiftusertest1
6bb2267536dc4051b986b8d9e80875ec	swiftusertest3
c2245914aaa942b09b37170b2288df31	swiftusertest2
6589e7650c4c49bda587fbbabd20d70a	heat
ab55370cabbb48e2b383492b3baf1cdf	aodh
49a0998341994269905dbe98adc6e6c2	user4labs

To locate the Glance image in Swift, you need to set the environment with a user of `swift` and a project of `service`.

4. Run the `source ~/devstack/openrc swift service` command to change the user and project environment variables.

Now you can use various Swift tools; you can look up the identity of the various images that you have been booting from. In this scenario, you will use the **image4labs** image.

5. Locate the ID of the **image4labs** image, by running the `openstack image list` command.

ID	Name	Status
ebb0ed60-d69d-4005-80bf-0d701d27128f	snapshotImageIsntance4labs	active
96296ac1-97bb-44a4-828c-728e4b0eee32	image4labs	active
b300fe18-3e37-41f3-92b3-0457d89e83db	cirros-0.3.4-x86_64-uec	active
0aa2869e-7a3d-41e7-ab38-ace90b13e73f	cirros-0.3.4-x86_64-uec-ramdisk	active
20aeef97f-b716-44b3-a32a-e90aac2557b5	cirros-0.3.4-x86_64-uec-kernel	active
1555dad9-abea-4d4e-bd72-4cc952ff13bb	Fedora-Cloud-Base-25-1.3.x86_64	active

6. Locate the Swift account ID for the `service` project with the `openstack container show glance` CLI command.

Field	Value
account	AUTH_ff7401ba42aa4c588305cbbb6cc7f792
bytes_used	265921971
container	glance
object_count	12

7. You now have the information necessary to look up the physical object locations for the image using the `swift-get-nodes` command using the Swift account ID, the `glance` container, and the image ID.

```
stack@stack:~$ swift-get-nodes /etc/swift/object.ring.gz AUTH_ff7401ba42aa4c588305cbbb6cc7f792 glance 96296
ac1-97bb-44a4-828c-728e4b8eee32

Account      AUTH_ff7401ba42aa4c588305cbbb6cc7f792
Container    glance
Object       96296ac1-97bb-44a4-828c-728e4b8eee32

Partition    442
Hash         dd4faf7922bcb1eda66eb64fbaab8613

Server:Port Device      127.0.0.1:6613 sdb1

curl -g -I -XHEAD "http://127.0.0.1:6613/sdb1/442/AUTH_ff7401ba42aa4c588305cbbb6cc7f792/glance/96296ac1-97b
b-44a4-828c-728e4b8eee32"

Use your own device location of servers:
such as "export DEVICE=/srv/node"
ssh 127.0.0.1 "ls -lah ${DEVICE:-/srv/node*}/sdb1/objects/442/613/dd4faf7922bcb1eda66eb64fbaab8613"
note: `/srv/node*` is used as default value of `devices`, the real value is set in the config file on each
storage node.
```

8. Add the device path for the object to the Swift root to display the object name of the image file.

```
stack@stack:~$ ls -l /opt/stack/data/swift/drives/sdb1/1/sdb1/objects/442/613/dd4faf7922bcb1eda66eb64fbaab8
613/
total 4
-rw-r-xr-x 1 stack stack 0 Dec 27 16:12 1482851565.10955.data
stack@stack:~$
```

9. Now that you have the full path to the object on the Swift node, you can obtain object information. Use the `swift-object-info` command to display the information about the object. Run the command using the information from step 8.

10. Please let the instructor know you have completed Lab 8.

Fundamentals of OpenStack® Technology

Lab 09—OpenStack® Heat (Orchestration)

Objectives

In this lab you will have the opportunity to verify that Heat is operational.

Requirements

- Completion of Lab 8
- Access to the HPE Virtual Lab equipment for completing the OpenStack® labs
- Heat-related services installed and enabled on your virtual lab system
- Approximately 60 minutes to complete

Exercise 1—Configuring OpenStack® for Heat

1. Log in to the Lab VM using the `stack/hpinvent` credentials.
2. To verify whether the Heat service is running, source the **devstack environment** file with admin credentials:


```
source ~/devstack/openrc admin admin
```
3. Run the `openstack orchestration service list` command. Verify that all services have the status of **up**.


```
stack@stack:~$ openstack orchestration service list
+-----+-----+-----+-----+-----+-----+
| hostname | binary | engine_id | host | topic | updated_at | status |
+-----+-----+-----+-----+-----+-----+
| stack | heat-engine | 2eb5b094-fee7-4e17-a5e6- | stack | engine | 2016-12-28T11:02:51.00000 | up |
| | | 882693f91808 | | | 0 |
| stack | heat-engine | 50ac2486-5100-41fe-8a5c- | stack | engine | 2016-12-28T11:02:51.00000 | up |
| | | f190bd6ad649 | | | 0 |
| stack | heat-engine | c8ec6e9c-93ca-4003-ad40- | stack | engine | 2016-12-28T11:02:51.00000 | up |
| | | b2e081359b46 | | | 0 |
| stack | heat-engine | 8a2b39aa-17ff- | stack | engine | 2016-12-28T11:02:51.00000 | up |
| | | 48ad-9035-12b072246915 | | | 0 |
+-----+-----+-----+-----+-----+-----+
```

4. You are going to use the unprivileged account to work with the Orchestration service. Source the **devstack environment** file with unprivileged user credentials:


```
source ~/devstack/openrc user4labs project4labs
```
5. As an unprivileged user, you cannot check the status of the services. Enter the following command to confirm that Heat is running and responding:


```
openstack stack list
```

If Heat is running, no error message will be returned.
6. To view the current flavors, enter the following command:

```
openstack flavor list
```

stack@stack:~\$ openstack flavor list						
ID	Name	RAM	Disk	Ephemeral	VCPUs	Is Public
1	m1.tiny	512	1	0	1	True
2	m1.small	2048	20	0	1	True
3	m1.medium	4096	40	0	2	True
4	m1.large	8192	80	0	4	True
5	m1.xlarge	16384	160	0	8	True
c1	cirros256	256	0	0	1	True
d1	ds512M	512	5	0	1	True
d2	ds1G	1024	10	0	1	True
d3	ds2G	2048	10	0	2	True
d4	ds4G	4096	20	0	4	True

The resource consumption can grow fast while using Heat. You will be using the **m1.tiny** flavor for the Heat labs.

Heat requires a key pair to launch an instance. You could use an existing key, but you will create a new key using the Horizon user interface.

7. Log in to the Horizon dashboard using the `default` domain with the `user4labs/hpinvent` credentials.
8. Change the project to **project4labs**, and navigate to **Project → Compute → Access & Security**.

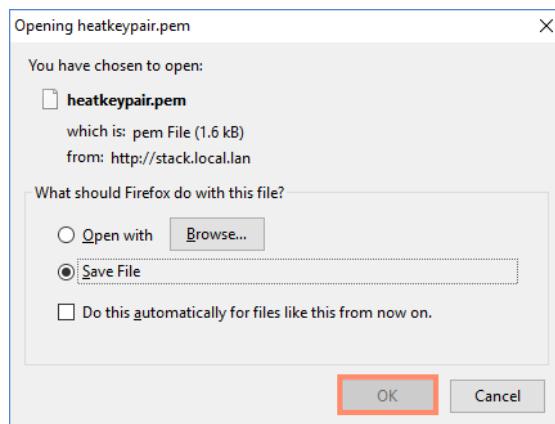
9. Select the **Key Pairs** tab and click **+ Create Key Pair**.

The screenshot shows the OpenStack Compute Access & Security interface. The 'Key Pairs' tab is selected, indicated by a red box. Below it, there is a search bar with a filter icon and a '+ Create Key Pair' button, also highlighted with a red box. The main table lists one key pair named 'keypair4labs' with a fingerprint of '62:d5:18:5e:f1:f2:33:44:1b:9d:42:c3:6b:22:ea:dc'. There are 'Actions' and 'Delete Key Pair' buttons for each row.

10. At the Create Key Pair dialog, enter `heatkeypair` as the key pair name and click **Create Key Pair**.

The screenshot shows the 'Create Key Pair' dialog box. The 'Key Pair Name' field contains 'heatkeypair', which is highlighted with a red box. The 'Description' section provides information about key pairs. The 'Create Key Pair' button at the bottom right is highlighted with a red box.

11. Select **Save File**, click **OK**, and save `heatkeypair.pem` to your desktop.



12. Make sure that the new key is listed in the Key Pairs table.

Access & Security

Security Groups			Key Pairs	Floating IPs	API Access	
			<input type="text" value="Filter"/>	Create Key Pair	Import Key Pair	Delete Key Pairs
<input type="checkbox"/>	Key Pair Name	Fingerprint	Actions			
<input type="checkbox"/>	heatkeypair	e1:04:9e:7c:07:44:f2:9a:78:05:f9:2d:6d:2c:3c:b8	Delete Key Pair			
<input type="checkbox"/>	keypair4labs	62:d5:18:5e:f1:f2:33:44:1b:9d:42:c3:6b:22:ea:dc		Delete Key Pair		

Displaying 2 items

13. Proceed with the next exercise.

Exercise 2—Basic Heat template example

Before using the OpenStack® orchestration template, you should familiarize with the Heat Orchestration Template (HOT) format. The HOT template format is meant to replace the Heat CloudFormation-compatible format (CFN) as the native format supported by Heat.

To ensure this lab runs smoothly, the template example referred to in this lab is located in two places:

- In the C:\Classfiles\hotTemplates folder on your HPEVL landing server
 - In the ~/devstack/hotTemplates folder on your Lab VM
1. On your terminal, change the location to the ~/devstack/hotTemplates directory:
\$ cd ~/devstack/hotTemplates
 2. The most basic template contains only a single resource definition using only predefined properties, along with the mandatory Heat template version tag. To display the basic template example, execute this command:

```
nano hotTemplate_basic.yaml
```

```

1  heat_template_version: 2015-04-30
2
3  description: Simple template to deploy a single compute instance
4
5  resources:
6    my_instance:
7      type: OS::Nova::Server
8      properties:
9        key_name: my_key
10       image: image4labs
11       flavor: m1.tiny

```

Each HOT template must include the `heat_template_version` key with a value of 2015-04-30 (or newer)—line 1 in the screenshot above.

While a description is optional, it is good practice to include some useful text that describes what users can do with the template—line 3 in the screenshot above.

The resource section is required and must contain at least one resource definition. In the example above, a compute instance is defined with fixed values for the `key_name`, `image`, and `flavor` parameters—lines 5 to 11 in the screenshot above.

NOTE: All these elements—a key pair with the given name, the image, and the flavor—must exist in the OpenStack® environment where the template is used. Typically, a template is made reusable more easily, though, by defining a set of input parameters instead of hard-coding such values.

3. To exit the nano editor, press the **Ctrl+X** keys concurrently.

The `hotTemplate_basic.yaml` template is in the `~/devstack/hotTemplates` directory.

4. Execute the command to create a new stack from the `hotTemplate_basic.yaml` template.

```
openstack stack create -t ./hotTemplate_basic.yaml myFirstHeatStack
```

Field	Value
<code>id</code>	<code>67008596-1d2a-4c6b-9cbe-fc42e4a2ed91</code>
<code>stack_name</code>	<code>myFirstHeatStack</code>
<code>description</code>	<code>Simple template to deploy a single compute instance</code>
<code>creation_time</code>	<code>2016-12-28T11:49:07Z</code>
<code>updated_time</code>	<code>None</code>
<code>stack_status</code>	<code>CREATE_IN_PROGRESS</code>
<code>stack_status_reason</code>	<code>Stack CREATE started</code>

5. The creation of the new stack is almost instant in this environment. To verify whether the creation completed successfully, repeat the `openstack stack list` command until you get the output below.

ID	Stack Name	Stack Status	Creation Time	Updated Time
<code>67008596-1d2a-4c6b-9cbe-fc42e4a2ed91</code>	<code>myFirstHeatStack</code>	<code>CREATE_COMPLETE</code>	<code>2016-12-28T11:49:07Z</code>	<code>None</code>

6. To check the available instances after running your template, execute the `openstack server list` command.

ID	Name	Status	Networks	Image Name
<code>ae3a75b4-8ea0-4fc7-bca5-2337f06a0385</code>	<code>myFirstHeatStack-my_instance-vc7ynqhcmxyz</code>	<code>ACTIVE</code>	<code>net4labs=10.10.1.105</code>	<code>image4labs</code>
<code>e4a066c7-c789-47dd-8759-12b99437b4f4</code>	<code>instfromvol2</code>	<code>ACTIVE</code>	<code>net4labs=10.10.1.106</code>	
<code>3a9e39f3-86e8-43a2-8ef0-e6d5f5387387</code>	<code>instfromvol</code>	<code>ACTIVE</code>	<code>net4labs=10.10.1.112</code>	
<code>ac106163-d6a7-4349-a9ac-d864f8211855</code>	<code>snappedImageInstance4Labs</code>	<code>ACTIVE</code>	<code>net4labs=10.10.1.103</code>	
<code>ec4d5049-d68f-4644-90fc-a3e2746e8654</code>	<code>instance4labs</code>	<code>ACTIVE</code>	<code>net4labs=10.10.1.107, 172.24.4.4</code>	<code>image4labs</code>

Exercise 3—Viewing the status of a stack from Horizon

1. Open the Firefox browser and access the Lab VM using the user4labs/hpinvent credentials.
2. Change your project to **project4labs**, and in the navigation pane, expand **Project → Orchestration → Stacks**. The Stacks panel appears and shows the list of currently running HOT templates.

	Stack Name	Created	Updated	Status	Actions
<input type="checkbox"/>	myFirstHeatStack	6 minutes	Never	Create Complete	<button>Check Stack</button>

3. In the Stack Name column, click the **myFirstHeatStack** link to display the Stack Details window.
- The Stack Details window offers five views in the form of tabs at the top of the window.

The topology view displays the visual representation of your stack. Your basic example has only one server component, but there might be much more objects including routers, firewalls, and so on.

4. Select the **Overview** tab to display the stack overview.

Overview provides the stack name and ID, the status providing the running time, the time of the last modification, as well as stack outputs and stack parameters (which is discussed in the exercises that follow).

The screenshot shows the 'myFirstHeatStack' stack details. The 'Overview' tab is selected, highlighted with a red box. The 'Topology' tab is also visible. The 'Check Stack' button is located in the top right corner.

Name	myFirstHeatStack
ID	67008596-1d2a-4c6b-9cbe-fc42e4a2ed91
Description	Simple template to deploy a single compute instance
Status	
Created	8 minutes
Last Updated	Never
Status	Create_Complete: Stack CREATE completed successfully
Outputs	
Stack Parameters	
OS::project_id	7b853dd380da4c5794e05f87031479af
OS::stack_name	myFirstHeatStack
OS::stack_id	67008596-1d2a-4c6b-9cbe-fc42e4a2ed91
Launch Parameters	
Timeout	None Minutes
Rollback	Disabled

5. Select the **Resources** tab. The list of consumed resources for this template is displayed. In this case, this is a single VM.

The screenshot shows the 'myFirstHeatStack' stack resources. The 'Resources' tab is selected, highlighted with a red box. The 'Topology', 'Overview', 'Events', and 'Template' tabs are also visible. The 'Check Stack' button is located in the top right corner.

Stack Resource	Resource	Stack Resource Type	Date Updated	Status	Status Reason
my_instance	ae3a75b4-8ea0-4fc7-bca5-2337f06a0385	OS::Nova::Server	10 minutes	Create Complete	state changed

Displaying 1 item

6. Select the **Events** tab, then select the **template** tab. The table displays all events connected to the running template.

The screenshot shows the 'Template' tab selected in the navigation bar. The template content is as follows:

```
description: Simple template to deploy a single compute instance
heat_template_version: '2015-04-30'
resources:
  my_instance:
    properties: {flavor: m1.tiny, image: image4labs, key_name: heatkeypair}
    type: OS::Nova::Server
```

7. In the navigation panel, click **Stacks** to return to the Stacks screen.
 8. Check the box next to your **myFirstHeatInstance** stack and click **Delete Stacks**.

	Stack Name	Created	Updated	Status	Actions
<input checked="" type="checkbox"/>	myFirstHeatStack	13 minutes	Never	Create Complete	Check Stack

9. To confirm the action, click **Delete Stacks**.

The screenshot shows a confirmation dialog box with the title "Confirm Delete Stacks". The message inside the box reads: "You have selected: 'myFirstHeatStack'. Please confirm your selection. This action cannot be undone." At the bottom right of the dialog box is a blue "Delete Stacks" button, which is highlighted with a red box.

Exercise 4—Template input parameters

In this exercise, the Template Input Parameters are introduced. You will learn how to initiate the stack using Horizon, and you will have to provide the parameters to the template using GUI.

HOT files are YAML-formatted. YAML is a simple data format that uses plain text to describe information in key value pairs (<http://www.yaml.org/>). If you had to create a YAML file from scratch, never use tabs and do not copy and paste from this document; it will almost certainly embed invisible characters, which will cause your template to fail and be hard to debug. Keep in mind that tab characters are illegal in YAML files. Always use only spaces to indent components of your HOT file.

1. On your HPEVL landing desktop, open Windows Explorer and navigate to C:\Classfiles\hotTemplates.
2. Using Notepad++, open the template called hotTemplate_inputA.yml.
3. From the main menu of Notepad++, click **Language → YAML**.
4. The content in the template file should look like the content from the screenshot below.

```

1 heat_template_version: 2015-04-30
2
3 description: Simple template to deploy a single compute instance
4
5 parameters:
6   key_name:
7     type: string
8     label: Key Name
9     description: Name of key-pair to be used for compute instance
10    image_id:
11      type: string
12      label: Image ID
13      description: Image to be used for compute instance
14    instance_type:
15      type: string
16      label: Instance Type
17      description: Type of instance (flavor) to be used
18
19 resources:
20   my_instance:
21     type: OS::Nova::Server
22     properties:
23       key_name: { get_param: key_name }
24       image: { get_param: image_id }
25       flavor: { get_param: instance_type }
26

```

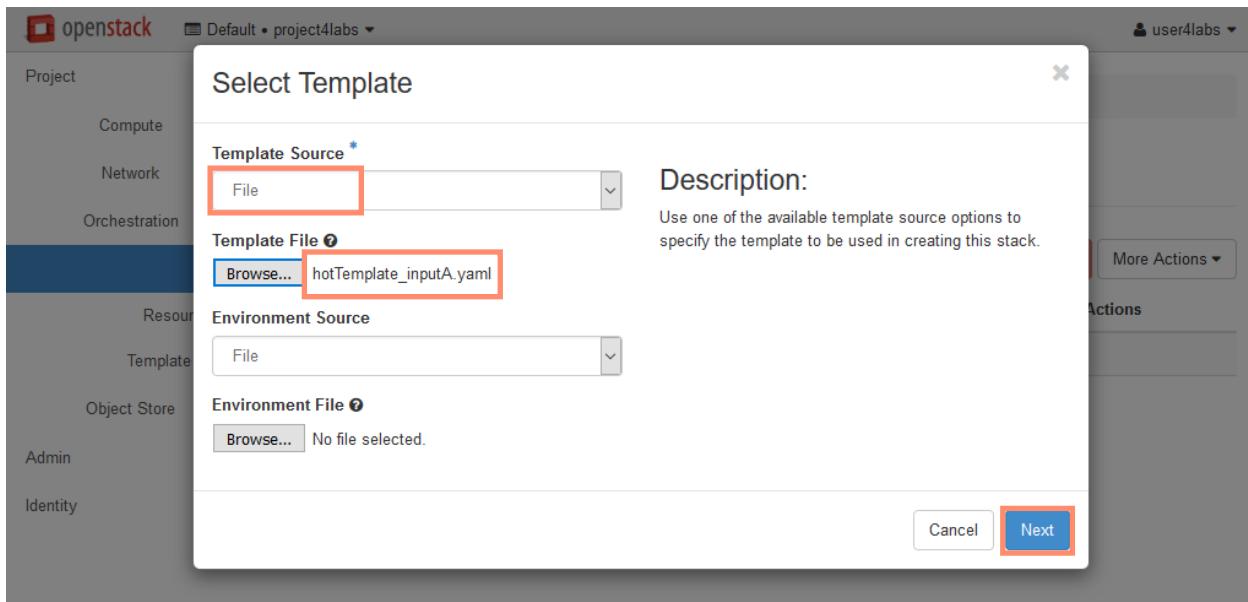
In this example template, three input parameters (6), (10), and (14) have been defined to be provided by the user upon deployment. The fixed values for the respective resource properties have been replaced by references to the corresponding parameters by means of `get_param` functions (23), (24), and (25).

5. Back at the Horizon dashboard, at the navigation pane, expand **Project → Orchestration → Stacks**. At the top right corner of the Stacks panel, click the **+ Launch Stack** button.

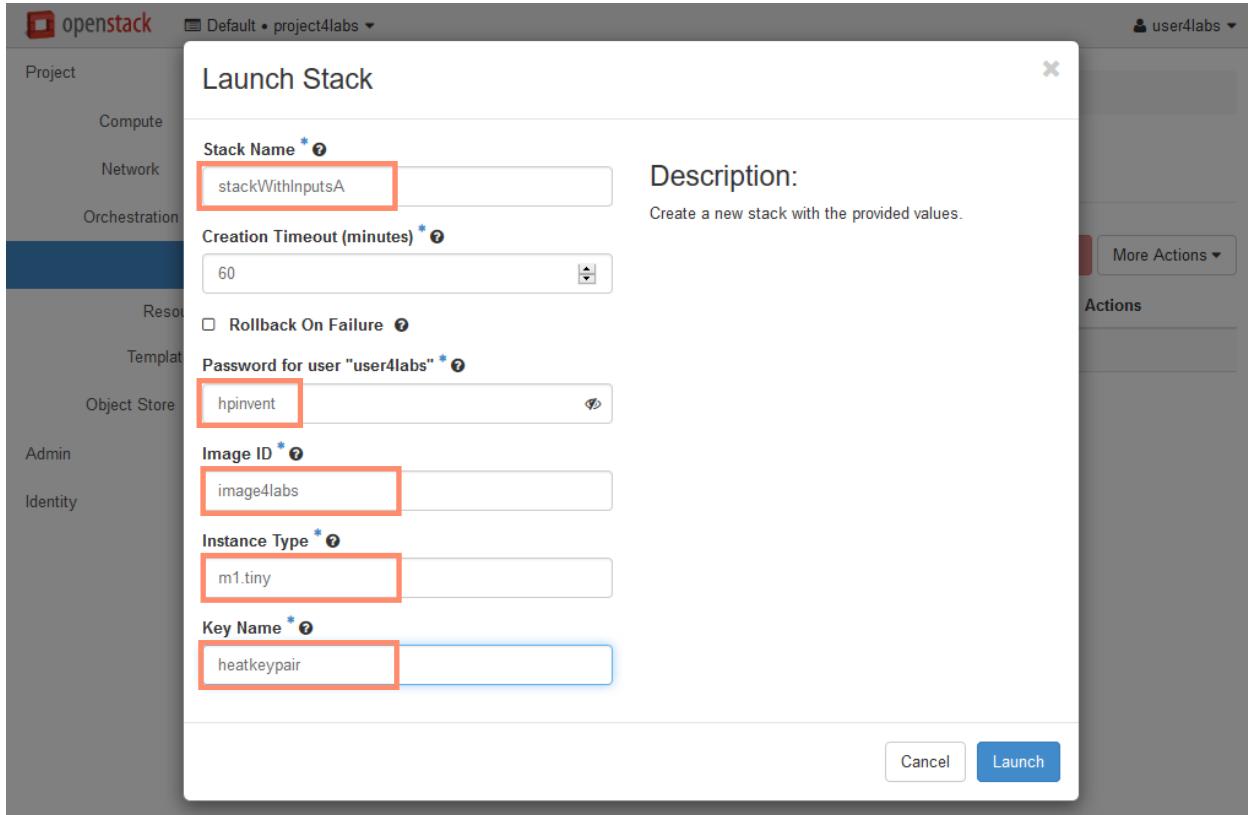
The screenshot shows the OpenStack Horizon dashboard with the following details:

- Header:** openstack Default • project4labs ▾ user4labs ▾
- Navigation:** Project / Orchestration / Stacks
- Section:** Stacks
- Actions Bar:** Filter, + Launch Stack (highlighted with a red box), Preview Stack, Delete Stacks, More Actions ▾
- Table Headers:** Stack Name, Created, Updated, Status, Actions
- Table Content:** No items to display.

6. At the Select Template dialog, select **File** as the template source. Select the `hotTemplate_inputA.yaml` file from the `c:\classFiles\HOT_templates` directory of your landing page, and then click **Next**.



7. At the Launch Stack dialog, provide the values from the screenshot below. Use `hpinvent` for the admin password.
8. Click **Launch** when ready.



9. Verify that the `stackwithInputsA` stack shows a status of `Create Complete`.

Stacks

<input type="checkbox"/>	Stack Name	Created	Updated	Status	Actions
<input type="checkbox"/>	stackWithInputsA	0 minutes	Never	Create Complete	<button>Check Stack</button> ▾

Displaying 1 item

10. Check the box next to your `stackWithInputsA` stack and click **Delete Stacks**.
11. In the confirmation dialog, click **Delete Stacks**.
12. Proceed with the next exercise.

Exercise 5—Improving templates

In the previous exercise, you used a basic form of user interaction. However, it might be useful to limit the choices for the user's input, or validate the user's input. This exercise will improve your template used in exercise 4. The final form of this template is saved for your convenience as C:\Classfiles\hotTemplates\hotTemplate_inputB.yml.

1. Use Notepad++ to open the hotTemplate_inputB.yml file.

To provide the initial value to the HOT parameters, provide the default values as shown in the lines 10 and 15 of the file below.

In some cases, you can restrict the values of input parameters that users can supply. For example, you might know that the software running in a compute instance needs a certain number of resources, so you might want to restrict the `instance_type` parameter introduced below.

Parameters in HOT templates can be restricted by adding a constraints section (lines 20 to 22).

```

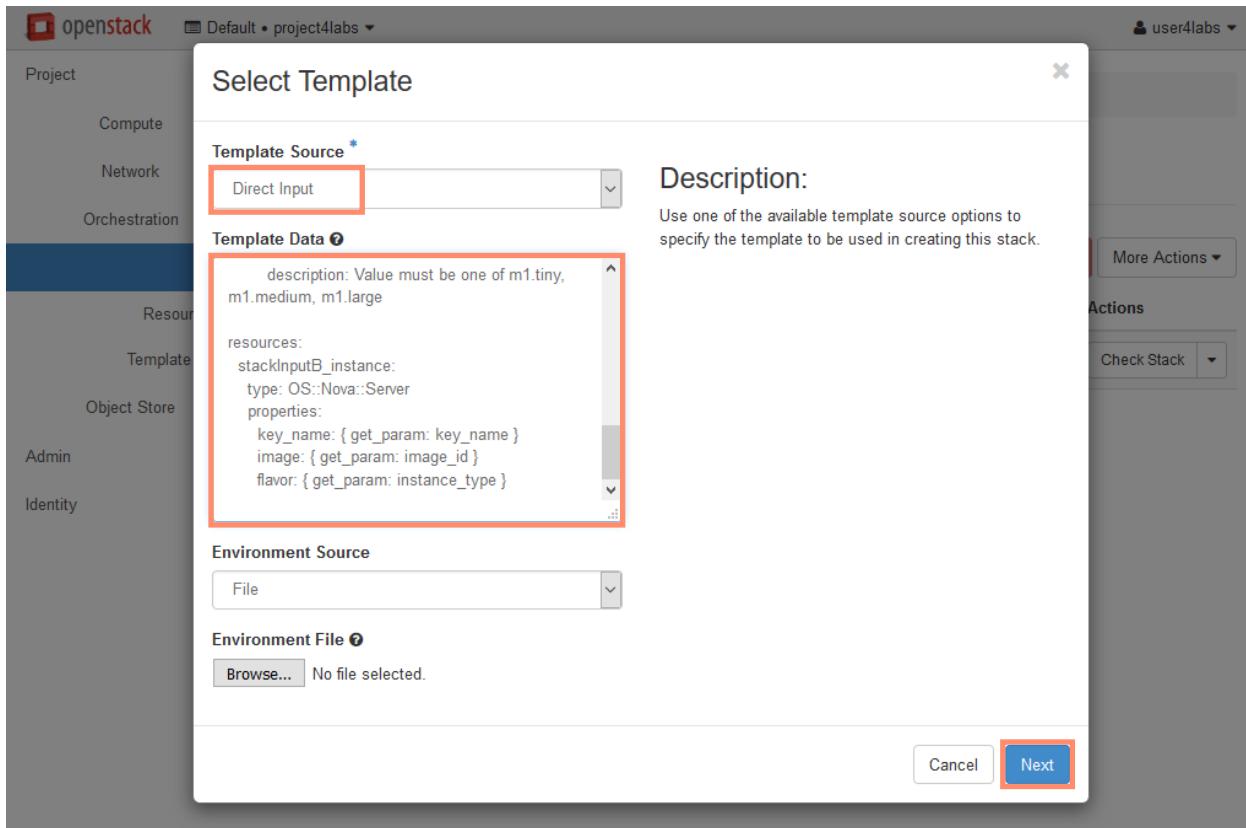
1 heat_template_version: 2015-04-30
2
3 description: Simple template to deploy a single compute instance
4
5 parameters:
6   key_name:
7     type: string
8     label: Key Name
9     description: Name of key-pair to be used for compute instance
10    default: heatkeypair
11   image_id:
12     type: string
13     label: Image ID
14     description: Image to be used for compute instance
15     default: image4labs
16   instance_type:
17     type: string
18     label: Instance Type
19     description: Type of instance (flavor) to be used
20     constraints:
21       - allowed_values: [ m1.tiny, m1.medium, m1.large ]
22         description: Value must be one of m1.tiny, m1.medium, m1.large
23
24 resources:
25   stackInputB_instance:
26     type: OS::Nova::Server
27     properties:
28       key_name: { get_param: key_name }
29       image: { get_param: image_id }
30       flavor: { get_param: instance_type }
```

2. Press **Ctrl + A** to highlight the content of the template file, and then **Ctrl C** to copy it to clipboard.
3. Back at the Horizon dashboard, at the navigation pane, expand **Project → Orchestration → Stacks** and click **+ Launch Stack**.

Stack Name	Created	Updated	Status	Actions
No items to display.				

In the previous exercise, you used the File template source, but in this exercise you will use the Direct Input option.

4. At the Select Template dialog, from the Template Source drop-down menu, select **Direct Input**.
5. Place the cursor in the Template Data field and click **Ctrl + V** to copy the template to the Template Data field. Then, click **Next**.



6. At the Launch Stack dialog, note the values that are provided as a result of the template changes. Also, note the selection box for the Instance Type field, providing only the fields defined in the template.

The screenshot shows the 'Launch Stack' dialog. The 'Stack Name' field contains 'stackWithInputsB'. The 'Creation Timeout (minutes)' field is set to 60. The 'Instance Type' dropdown menu is open, showing 'm1.tiny' as the selected option. The 'Launch' button at the bottom right is highlighted with a red box.

7. Click **Launch**.
8. Back at the Stacks panel, verify that the new stack has deployed successfully.

Stacks

<input type="checkbox"/>	Stack Name	Created	Updated	Status	Actions
<input checked="" type="checkbox"/>	stackWithInputsB	1 minute	Never	Create Complete	<input type="button" value="Check Stack"/>

Displaying 1 item

9. Check the box next to your **stackWithInputsB** stack and click **Delete Stacks**.
10. In the confirmation dialog, click **Delete Stacks**.
11. Proceed with the next exercise.

Exercise 6—Providing parameters to the Heat command line

In this exercise, the same template that was used in exercise 5 will be used again to demonstrate how you can provide parameters through the command line.

- At the Lab VM terminal session, make sure that you have sourced the `openrc` as `user4labs` / `project4labs`:


```
$ source ~/devstack/openrc user4labs project4labs
```
- Change your directory to the `~/devstack/hotTemplates` folder, and check for the file named `hotTemplate_inputB.yaml`.
- At the command line, execute the following command to create a new stack, and provide the required parameter, which in this case is the instance flavor:

```
openstack stack create -t hotTemplate_inputB.yaml --parameter "instance_type=m1.large" --dry-run stackWithInputsB
```

Note that the command does not create the actual stack due to the `--dry-run` parameter being specified. In the output of the command, search for the value of the `instance_type` parameter.

```
+-----+-----+
| capabilities | []
| creation_time | 2016-12-28T13:20:03Z
| deletion_time | None
| description | Simple template to deploy a single compute instance
| disable_rollback | True
| id | None
| links | http://192.168.5.4:8004/v1/7b853dd380da4c5794e05f87031479af/stacks/stackWithInp
| utsB/None (self)
| notification_topics | []
| outputs | []
| parameters | {
|   "OS::project_id": "7b853dd380da4c5794e05f87031479af",
|   "OS::stack_id": "None",
|   "OS::stack_name": "stackWithInputsB",
|   "instance_type": "m1.large",
|   "key_name": "heatkeypair",
|   "image_id": "image4labs"
| }
| parent | None
| resources | [
|   {
|     "resource_name": "stackInputB_instance",
|     "resource_identity": {
|       "stack_name": "stackWithInputsB",
|     }
|   }
| ]
```

- To actually create the new stack, change the `instance_type` parameter to `m1.tiny`, and remove the `--dry-run` option as shown below.

```
openstack stack create -t hotTemplate_inputB.yaml --parameter "instance_type=m1.tiny" stackWithInputsB
```

Field	Value
<code>id</code>	<code>1fd0229e-88f3-4aa0-a653-38d3f14d167b</code>
<code>stack_name</code>	<code>stackWithInputsB</code>
<code>description</code>	<code>Simple template to deploy a single compute instance</code>
<code>creation_time</code>	<code>2016-12-28T13:23:33Z</code>
<code>updated_time</code>	<code>None</code>
<code>stack_status</code>	<code>CREATE_IN_PROGRESS</code>
<code>stack_status_reason</code>	<code>Stack CREATE started</code>

- Run the `openstack stack list` command to verify the completion of the stack creation process.

ID	Stack Name	Stack Status	Creation Time	Updated Time
1fd0229e-88f3-4aa0-a653-38d3f14d167b	stackWithInputsB	CREATE_COMPLETE	2016-12-28T13:23:33Z	None

- If you do not provide the required parameters, the `openstack stack create` command reports an error.

```
stack@stack:~/devstack/hotTemplates$ openstack stack create -t hotTemplate inputB.yaml test
ERROR: The Parameter (instance_type) was not provided.
```

- Delete the stack you just created with the `heat stack-delete` `stackWithInputsB` CLI command.

At the prompt, answer **y** and press **Enter**.

```
stack@stack:~/devstack/hotTemplates$ openstack stack delete stackWithInputsB
Are you sure you want to delete this stack(s) [y/N]? Y
```

- Proceed with the next exercise.

Exercise 7—Providing template outputs

In addition to template customization through input parameters, you can provide outputs to users, which can be done in the outputs section of a template.

For example, provide an output for the IP address through which the instance defined in the previous exercise can be accessed and then provided to users. Otherwise, users would have to look it up themselves. The definition for providing the IP address of the compute instance as an output is shown in the following snippet.

1. Using the steps described in previous exercises and using the template located at C:\classFiles\hotTemplates\hotTemplate_IO.yml use the GUI or CLI to create a stack called **stackWithIO**.
2. Wait for deployment to finish. From the Horizon UI, click the **stackWithIO** name.

Stacks

<input type="checkbox"/>	Stack Name	Created	Updated	Status	Actions
<input type="checkbox"/>	stackWithIO	0 minutes	Never	Create Complete	<button>Check Stack</button> ▾

Displaying 1 item

3. Click the **Overview** tab and check the **Outputs** panel of the **stackWithIO** stack.

stackWithIO

[Check Stack](#) ▾

Topology	Overview	Resources	Events	Template
Name	stackWithIO			
ID	9321e6be-1b0c-4936-9755-2a3f86b3ec14			
Description	Simple template to deploy a single compute instance			

Status

Created	1 minute
Last Updated	Never
Status	Create_Complete: Stack CREATE completed successfully

Outputs

instance_ip	The IP address of the deployed instance 10.10.1.108
-------------	--

Stack Parameters

OS::project_id	7b853dd380da4c5794e05f87031479af
OS::stack_id	9321e6be-1b0c-4936-9755-2a3f86b3ec14
OS::stack_name	stackWithIO
instance_type	m1.tiny
key_name	heatkeypair
image_id	image4labs

Launch Parameters

Timeout	60 Minutes
Rollback	Disabled

NOTE: It takes some time for the output to become available. Try refreshing the browser window a few times.

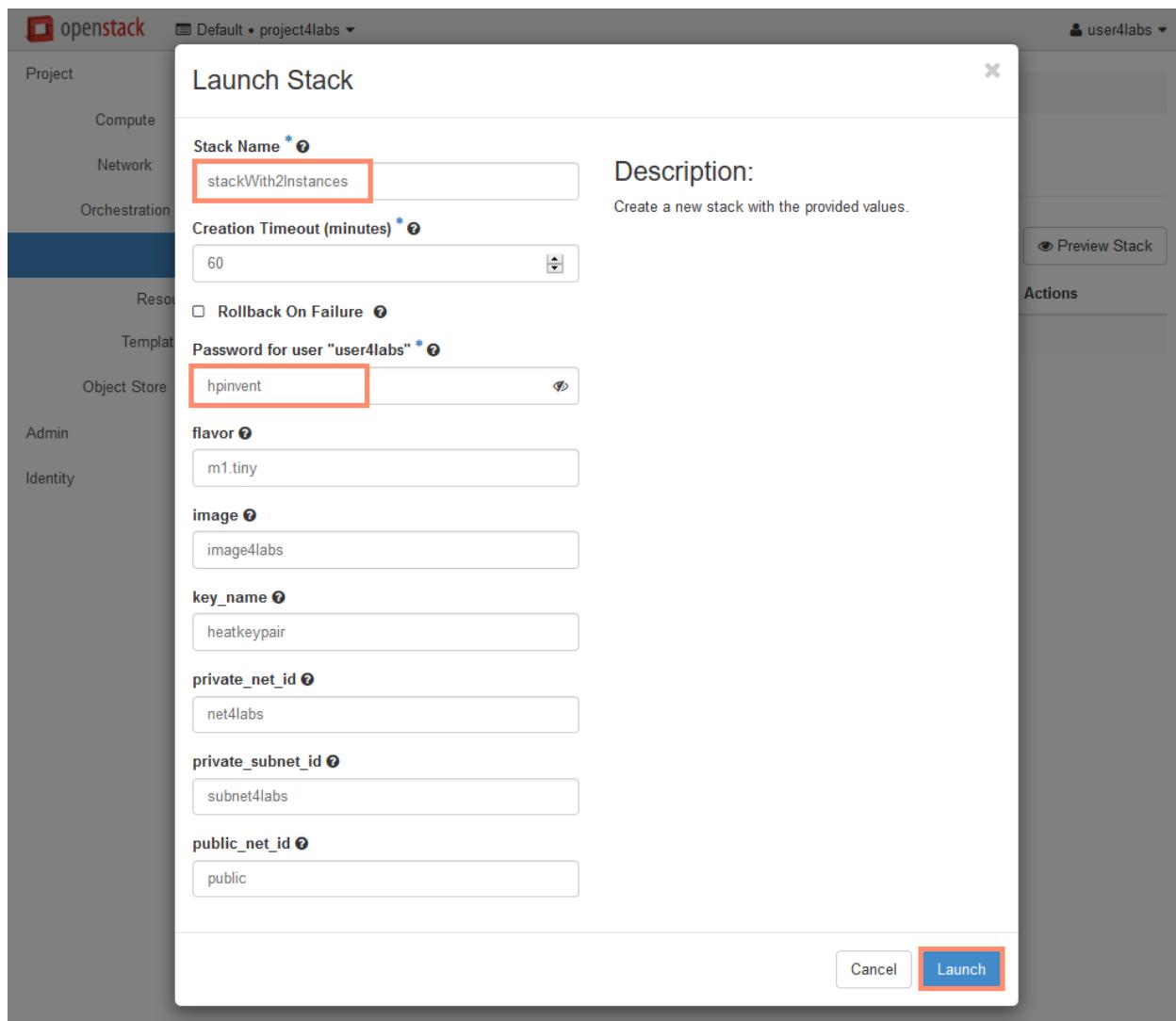
4. Check the box next to your **stackWithIO** stack and click **Delete Stacks**.

Exercise 8—Complex template deployment

In this exercise, you will deploy a more complex template to demonstrate the ease of using Heat for more sophisticated deployment scenarios.

1. Using Notepad++, on your HPEVL landing desktop, open the `C:\classFiles\hotTemplates\hotTemplate_2Instances.yml` template.
2. Examine the content of the file. Check the parameters, resources, and output sections of the template file.
3. Using the method demonstrated in previous examples, deploy the template as `stackWith2Instances`.

NOTE: Per template, you must specify the password of `hpinvent`, and the name of the template. The remaining values will be populated automatically.



The screenshot shows the 'Launch Stack' dialog box from the OpenStack Heat interface. The dialog has the following fields filled in:

- Stack Name ***: stackWith2Instances
- Creation Timeout (minutes) ***: 60
- Password for user "user4labs" ***: hpinvent
- flavor**: m1.tiny
- image**: image4labs
- key_name**: heatkeypair
- private_net_id**: net4labs
- private_subnet_id**: subnet4labs
- public_net_id**: public

At the bottom right of the dialog are two buttons: 'Cancel' and 'Launch', with 'Launch' being highlighted by a red border.

4. Wait for the stack to appear on the list of available stacks.

Stacks

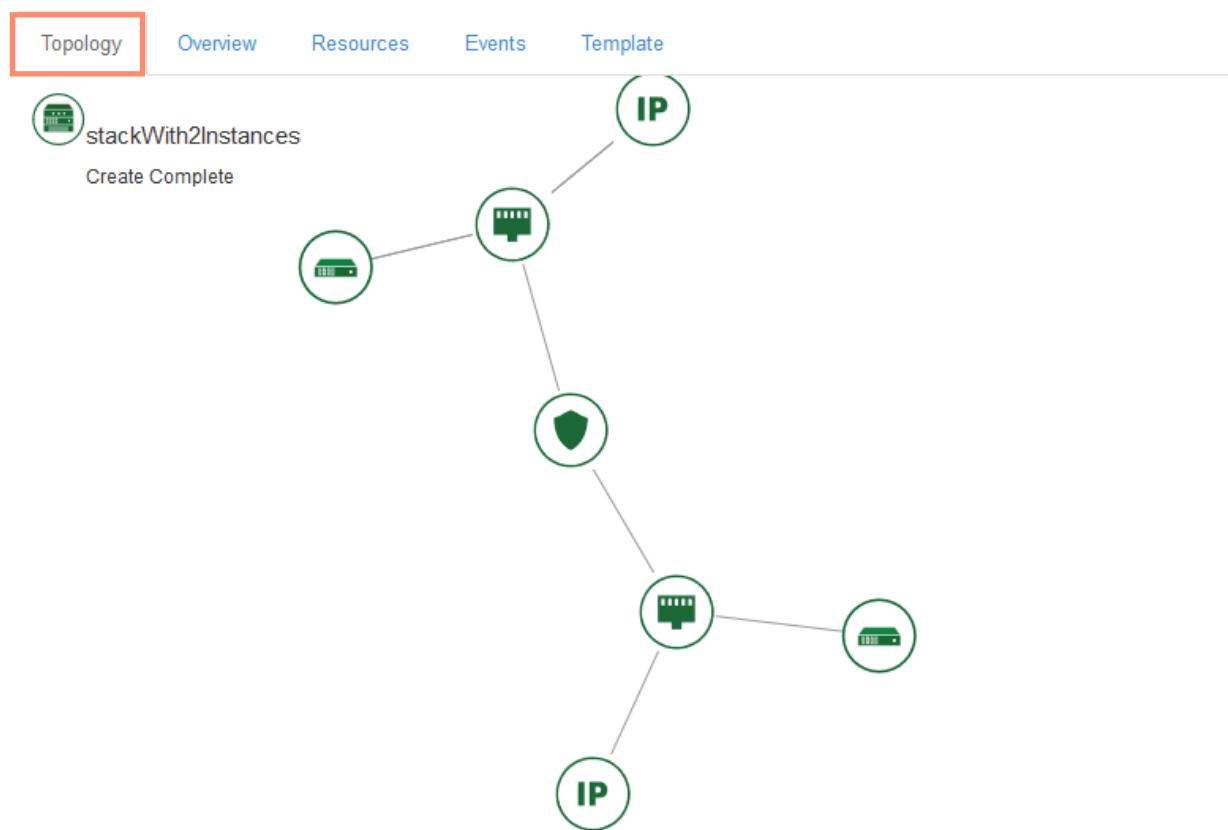
<input type="checkbox"/> Stack Name	Created	Updated	Status	Actions
<input type="checkbox"/> stackWith2Instances	6 minutes	Never	Create Complete	Check Stack ▾

Displaying 1 item

5. In the Stack Name column, click the **stackWith2Instances** template link to display the Stack Overview.
 6. Click the **Topology** tab and hover your mouse over different objects on the screen to examine their meaning.

stackWith2Instances

[Check Stack](#) ▾



7. Click the **Overview** tab and note the IP addresses assigned to the instances in the Outputs section of the screen.

stackWith2Instances

Check Stack ▾

Topology	Overview	Resources	Events	Template
Name	stackWith2Instances			
ID	2e7eac30-f129-4b97-b1f6-ce912a521b94			
Description	HOT template to deploy two servers into an existing neutron tenant network and assign floating IP addresses to each server so they are routable from the public network.			
Status				
Created	0 minutes			
Last Updated	Never			
Status	Create_Complete: Stack CREATE completed successfully			
Outputs				
server2_private_ip	IP address of server2 in private network 10.10.1.101			
server1_private_ip	IP address of server1 in private network 10.10.1.114			
server2_public_ip	Floating IP address of server2 in public network 172.24.4.9			
server1_public_ip	Floating IP address of server1 in public network 172.24.4.6			

Stack Parameters

OS::project_id	7b853dd380da4c5794e05f87031479af
OS::stack_id	2e7eac30-f129-4b97-b1f6-ce912a521b94
OS::stack_name	stackWith2Instances
key_name	heatkeypair
image	image4labs
public_net_id	public
private_net_id	net4labs
flavor	m1.tiny
private_subnet_id	subnet4labs

8. Select the **Resources** tab and note the resources used by this stack.

stackWith2Instances

[Check Stack](#) ▾

Topology	Overview	Resources	Events	Template	
Stack Resource	Resource	Stack Resource Type	Date Updated	Status	Status Reason
server_security_group	e5d7f3e4-49ff-437a-8ce7-92d1dde5af5	OS::Neutron::SecurityGroup	9 minutes	Create Complete	state changed
server1_port	1f77b191-98f1-491b-9937-17f0feb0ad85	OS::Neutron::Port	9 minutes	Create Complete	state changed
server2_floating_ip	1d30c103-5a98-40c9-83d8-d47f3948826f	OS::Neutron::FloatingIP	9 minutes	Create Complete	state changed
server1	5dafd1c6-dfd0-45b1-bd42-150d6465cbde	OS::Nova::Server	9 minutes	Create Complete	state changed
server2	274244ec-3585-4ac7-aeb2-e8a5eb430acd	OS::Nova::Server	9 minutes	Create Complete	state changed
server2_port	cda6a329-a4a5-4ae3-88eb-0f86e8e8abec	OS::Neutron::Port	9 minutes	Create Complete	state changed
server1_floating_ip	ff0de4c1-4061-47bc-8625-d40e61a92fcb	OS::Neutron::FloatingIP	9 minutes	Create Complete	state changed

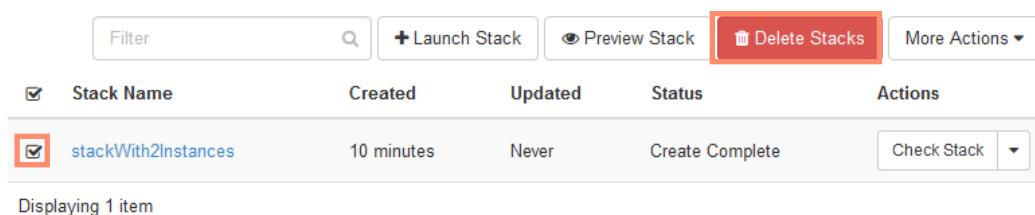
Displaying 7 items

Exercise 9—Cleanup

To clean up the environment for the next exercise, perform the following steps:

1. Navigate to **Project** → **Orchestration** → **Stacks**. In the stack list, select the check box in the top left corner to mark all available stacks, and click the **Delete Stacks** button.

Stacks



		Stack Name	Created	Updated	Status	Actions
<input checked="" type="checkbox"/>	stackWith2Instances	10 minutes	Never	Create Complete	<button>Check Stack</button>	<button>More Actions ▾</button>

Displaying 1 item

2. To confirm the action, click the **Delete Stack** button in the confirmation dialog.
3. Leave the browser open for the next lab.
4. Please let your instructor know that you have completed Lab 9.

Fundamentals of OpenStack® Technology

Lab 10—OpenStack® Ceilometer (Telemetry)

Objectives

After completing this lab, you should be able to:

- Describe the major features of the OpenStack® metering service
- Gather metering data from the environment

Requirements

- Completion of Lab 9
- Access to the HPE Virtual Lab equipment for completing the OpenStack® labs
- An instance from the previous lab exercise should exist
- Approximately 45 minutes to complete

Introduction

In this lab, you will examine the OpenStack® Telemetry service, called Ceilometer, which is a service designed to collect measurements of the utilization of the physical and virtual resources in a cloud. It can also be used to set alarms when specific thresholds are met.

Ceilometer can be used to provide data on various OpenStack® project services and can be used for optimizing performance, capacity planning, troubleshooting, and for billing use of those OpenStack® services.

In this lab, you will examine the OpenStack® Ceilometer metering service and get a better understanding of its architecture and function. In particular, you will examine the metering data associated with compute instances.

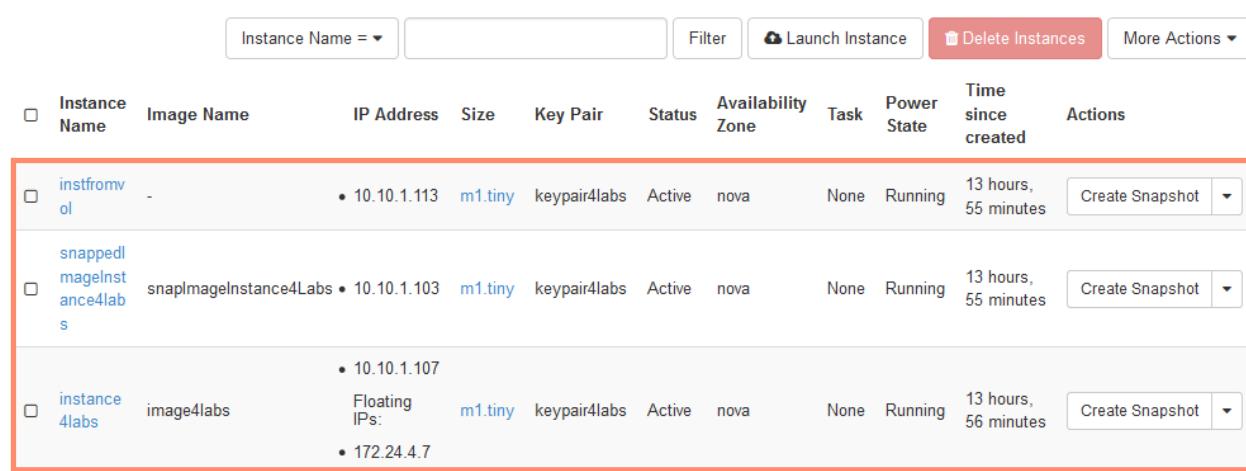
You will use the instance from a previous lab and then extract the utilization information about the instance.

Exercise 1—Ceilometer overview

Generating traffic for the meters

- In this exercise, you will use the `instance4labs`, `instfromvol` and `snappedImageInstance4labs` instances you created previously in the Lab 6, Exercise 2. These instances have been running for a while and should provide some interesting data that you can collect using the OpenStack® Telemetry service.

Instances



<input type="checkbox"/>	Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
<input type="checkbox"/>	instfromvol	-	• 10.10.1.113	m1.tiny	keypair4labs	Active	nova	None	Running	13 hours, 55 minutes	<button>Create Snapshot</button> ▾
<input type="checkbox"/>	snappedImageInstance4labs	snappedImageInstance4labs	• 10.10.1.103	m1.tiny	keypair4labs	Active	nova	None	Running	13 hours, 55 minutes	<button>Create Snapshot</button> ▾
<input type="checkbox"/>	instance4labs	image4labs	• 10.10.1.107 Floating IPs: • 172.24.4.7	m1.tiny	keypair4labs	Active	nova	None	Running	13 hours, 56 minutes	<button>Create Snapshot</button> ▾

Displaying 3 items

You will need the Floating IP address for the `instance4labs` instance.

- From a terminal session, enter the following command to access your instance using SSH:
- ```
ssh -i ~/devstack/keypair4labs.pem cirros@<assigned floating IP Address of instance4labs>
```
- The `nohup yes > yes /dev/null &` command is used to generate compute data points for Ceilometer. The command keeps the `yes` command running in a loop, and `nohup` (no hang up) keeps the `yes` program running, even after you log out of the instance.
- At the \$ prompt of the terminal session, enter `nohup yes > /dev/null &` press the **Enter** key, and then enter `exit` to exit the session.

```
$ nohup yes > yes /dev/null &
$
[1]+ Done(1) nohup yes /dev/null 1>yes
$ exit
Connection to 172.24.4.7 closed.
stack@stack:~/devstack$
```

Notice that you will not be generating traffic on the `snappedImageInstance4labs` instance so that you can tell the difference between the two.

## Outputting Ceilometer data from the CLI

- To make sure you are at the `devstack` directory, enter the `cd ~/devstack` command. To set the environment variables, enter the `source openrc user4labs project4labs` command.
- Ceilometer uses the concept of a meter to indicate the type of measurement being taken. For example, a CPU usage data point might be in the `cpu_util` meter, which represents the percentage of utilization. Other CPU usage data points can be in the `cpumeter`, which represents cumulative CPU time in nanoseconds.
- In a terminal window on your DevStack host, run the `ceilometer meter-list` command to view the large list of available meters.

| Name                          | Type       | Unit      | Resource ID                              | User ID                         | Project ID                      |
|-------------------------------|------------|-----------|------------------------------------------|---------------------------------|---------------------------------|
| compute.instance.booting.time | gauge      | sec       | 24601d87-456e-4b6c-98b4-5e112fd08bf7     | ec02ef546f9d45609e8670ebbsfe459 | afcd8a4bb0a454aa4a71f79ec386685 |
| compute.instance.booting.time | gauge      | sec       | 432ccae0-f554-4da5-9c92-6269d32bc472     | ec02ef546f9d45609e8670ebbsfe459 | afcd8a4bb0a454aa4a71f79ec386685 |
| compute.instance.booting.time | gauge      | sec       | d226b22e-a723-4e17-8043-fb1ea2e32488     | ec02ef546f9d45609e8670ebbsfe459 | afcd8a4bb0a454aa4a71f79ec386685 |
| cpu                           | cumulative | ns        | 24601d87-456e-4b6c-98b4-5e112fd08bf7     | ec02ef546f9d45609e8670ebbsfe459 | afcd8a4bb0a454aa4a71f79ec386685 |
| cpu                           | cumulative | ns        | d226b22e-a723-4e17-8043-fb1ea2e32488     | ec02ef546f9d45609e8670ebbsfe459 | afcd8a4bb0a454aa4a71f79ec386685 |
| cpu.delta                     | delta      | ns        | 24601d87-456e-4b6c-98b4-5e112fd08bf7     | ec02ef546f9d45609e8670ebbsfe459 | afcd8a4bb0a454aa4a71f79ec386685 |
| cpu.delta                     | delta      | ns        | d226b22e-a723-4e17-8043-fb1ea2e32488     | ec02ef546f9d45609e8670ebbsfe459 | afcd8a4bb0a454aa4a71f79ec386685 |
| cpu.util                      | gauge      | %         | 24601d87-456e-4b6c-98b4-5e112fd08bf7     | ec02ef546f9d45609e8670ebbsfe459 | afcd8a4bb0a454aa4a71f79ec386685 |
| cpu.util                      | gauge      | %         | d226b22e-a723-4e17-8043-fb1ea2e32488     | ec02ef546f9d45609e8670ebbsfe459 | afcd8a4bb0a454aa4a71f79ec386685 |
| disk.allocation               | gauge      | B         | 24601d87-456e-4b6c-98b4-5e112fd08bf7     | ec02ef546f9d45609e8670ebbsfe459 | afcd8a4bb0a454aa4a71f79ec386685 |
| disk.allocation               | gauge      | B         | 432ccae0-f554-4da5-9c92-6269d32bc472     | ec02ef546f9d45609e8670ebbsfe459 | afcd8a4bb0a454aa4a71f79ec386685 |
| disk.allocation               | gauge      | B         | d226b22e-a723-4e17-8043-fb1ea2e32488     | ec02ef546f9d45609e8670ebbsfe459 | afcd8a4bb0a454aa4a71f79ec386685 |
| disk.capacity                 | gauge      | B         | 24601d87-456e-4b6c-98b4-5e112fd08bf7     | ec02ef546f9d45609e8670ebbsfe459 | afcd8a4bb0a454aa4a71f79ec386685 |
| disk.capacity                 | gauge      | B         | d226b22e-a723-4e17-8043-fb1ea2e32488     | ec02ef546f9d45609e8670ebbsfe459 | afcd8a4bb0a454aa4a71f79ec386685 |
| disk.device.allocation        | gauge      | B         | 24601d87-456e-4b6c-98b4-5e112fd08bf7-vda | ec02ef546f9d45609e8670ebbsfe459 | afcd8a4bb0a454aa4a71f79ec386685 |
| disk.device.allocation        | gauge      | B         | 432ccae0-f554-4da5-9c92-6269d32bc472-vda | ec02ef546f9d45609e8670ebbsfe459 | afcd8a4bb0a454aa4a71f79ec386685 |
| disk.device.allocation        | gauge      | B         | d226b22e-a723-4e17-8043-fb1ea2e32488-vda | ec02ef546f9d45609e8670ebbsfe459 | afcd8a4bb0a454aa4a71f79ec386685 |
| disk.device.capacity          | gauge      | B         | 24601d87-456e-4b6c-98b4-5e112fd08bf7-vda | ec02ef546f9d45609e8670ebbsfe459 | afcd8a4bb0a454aa4a71f79ec386685 |
| disk.device.capacity          | gauge      | B         | 432ccae0-f554-4da5-9c92-6269d32bc472-vda | ec02ef546f9d45609e8670ebbsfe459 | afcd8a4bb0a454aa4a71f79ec386685 |
| disk.device.capacity          | gauge      | B         | d226b22e-a723-4e17-8043-fb1ea2e32488-vda | ec02ef546f9d45609e8670ebbsfe459 | afcd8a4bb0a454aa4a71f79ec386685 |
| disk.device.read.bytes        | cumulative | B         | 24601d87-456e-4b6c-98b4-5e112fd08bf7-vda | ec02ef546f9d45609e8670ebbsfe459 | afcd8a4bb0a454aa4a71f79ec386685 |
| disk.device.read.bytes        | cumulative | B         | 432ccae0-f554-4da5-9c92-6269d32bc472-vda | ec02ef546f9d45609e8670ebbsfe459 | afcd8a4bb0a454aa4a71f79ec386685 |
| disk.device.read.bytes        | cumulative | B         | d226b22e-a723-4e17-8043-fb1ea2e32488-vda | ec02ef546f9d45609e8670ebbsfe459 | afcd8a4bb0a454aa4a71f79ec386685 |
| disk.device.read.bytes        | cumulative | B         | 24601d87-456e-4b6c-98b4-5e112fd08bf7-vda | ec02ef546f9d45609e8670ebbsfe459 | afcd8a4bb0a454aa4a71f79ec386685 |
| disk.device.read.bytes        | cumulative | B         | 432ccae0-f554-4da5-9c92-6269d32bc472-vda | ec02ef546f9d45609e8670ebbsfe459 | afcd8a4bb0a454aa4a71f79ec386685 |
| disk.device.read.bytes        | cumulative | B         | d226b22e-a723-4e17-8043-fb1ea2e32488-vda | ec02ef546f9d45609e8670ebbsfe459 | afcd8a4bb0a454aa4a71f79ec386685 |
| disk.device.read.requests     | cumulative | request   | 24601d87-456e-4b6c-98b4-5e112fd08bf7-vda | ec02ef546f9d45609e8670ebbsfe459 | afcd8a4bb0a454aa4a71f79ec386685 |
| disk.device.read.requests     | cumulative | request   | 432ccae0-f554-4da5-9c92-6269d32bc472-vda | ec02ef546f9d45609e8670ebbsfe459 | afcd8a4bb0a454aa4a71f79ec386685 |
| disk.device.read.requests     | cumulative | request   | d226b22e-a723-4e17-8043-fb1ea2e32488-vda | ec02ef546f9d45609e8670ebbsfe459 | afcd8a4bb0a454aa4a71f79ec386685 |
| disk.device.read.requests     | cumulative | request   | 24601d87-456e-4b6c-98b4-5e112fd08bf7-vda | ec02ef546f9d45609e8670ebbsfe459 | afcd8a4bb0a454aa4a71f79ec386685 |
| disk.device.read.requests     | cumulative | request   | 432ccae0-f554-4da5-9c92-6269d32bc472-vda | ec02ef546f9d45609e8670ebbsfe459 | afcd8a4bb0a454aa4a71f79ec386685 |
| disk.device.read.requests     | cumulative | request   | d226b22e-a723-4e17-8043-fb1ea2e32488-vda | ec02ef546f9d45609e8670ebbsfe459 | afcd8a4bb0a454aa4a71f79ec386685 |
| disk.device.read.requests     | cumulative | request   | 24601d87-456e-4b6c-98b4-5e112fd08bf7-vda | ec02ef546f9d45609e8670ebbsfe459 | afcd8a4bb0a454aa4a71f79ec386685 |
| disk.device.read.requests     | cumulative | request   | 432ccae0-f554-4da5-9c92-6269d32bc472-vda | ec02ef546f9d45609e8670ebbsfe459 | afcd8a4bb0a454aa4a71f79ec386685 |
| disk.device.read.requests     | cumulative | request   | d226b22e-a723-4e17-8043-fb1ea2e32488-vda | ec02ef546f9d45609e8670ebbsfe459 | afcd8a4bb0a454aa4a71f79ec386685 |
| disk.device.write.requests    | cumulative | request   | 24601d87-456e-4b6c-98b4-5e112fd08bf7-vda | ec02ef546f9d45609e8670ebbsfe459 | afcd8a4bb0a454aa4a71f79ec386685 |
| disk.device.write.requests    | cumulative | request   | 432ccae0-f554-4da5-9c92-6269d32bc472-vda | ec02ef546f9d45609e8670ebbsfe459 | afcd8a4bb0a454aa4a71f79ec386685 |
| disk.device.write.requests    | cumulative | request   | d226b22e-a723-4e17-8043-fb1ea2e32488-vda | ec02ef546f9d45609e8670ebbsfe459 | afcd8a4bb0a454aa4a71f79ec386685 |
| disk.device.write.requests    | cumulative | request   | 24601d87-456e-4b6c-98b4-5e112fd08bf7-vda | ec02ef546f9d45609e8670ebbsfe459 | afcd8a4bb0a454aa4a71f79ec386685 |
| disk.device.write.requests    | cumulative | request   | 432ccae0-f554-4da5-9c92-6269d32bc472-vda | ec02ef546f9d45609e8670ebbsfe459 | afcd8a4bb0a454aa4a71f79ec386685 |
| disk.device.write.requests    | cumulative | request   | d226b22e-a723-4e17-8043-fb1ea2e32488-vda | ec02ef546f9d45609e8670ebbsfe459 | afcd8a4bb0a454aa4a71f79ec386685 |
| disk.read.bytes               | gauge      | request/s | d226b22e-a723-4e17-8043-fb1ea2e32488-vda | ec02ef546f9d45609e8670ebbsfe459 | afcd8a4bb0a454aa4a71f79ec386685 |
| disk.read.bytes               | cumulative | B         | 24601d87-456e-4b6c-98b4-5e112fd08bf7     | ec02ef546f9d45609e8670ebbsfe459 | afcd8a4bb0a454aa4a71f79ec386685 |

Meters have a name (what is being measured) and a type (gauge for an instantaneous reading, delta for the change since last meter, cumulative, and so on). Meters are associated with resources being measured and can also have an associated user.

While Ceilometer supports a wide range of built-in meters, it cannot anticipate all metering needs and is, therefore, very extensible, making it easy to add new meter types. Meters define types of data points. The data points themselves are called samples. You can list a set of samples for a particular meter using the `sample-list` command.

- In the following step, you will generate a sample list of values from the CPU meter.

| Resource ID                          | Name | Type       | Volume        | Unit | Timestamp                  |
|--------------------------------------|------|------------|---------------|------|----------------------------|
| 24601d87-456e-4b6c-98b4-5e112fd08bf7 | cpu  | cumulative | 32350000000.0 | ns   | 2016-12-29T11:53:57.162166 |
| d226b22e-a723-4e17-8043-fb1ea2e32488 | cpu  | cumulative | 36180000000.0 | ns   | 2016-12-29T11:53:57.162166 |
| 432ccae0-f554-4da5-9c92-6269d32bc472 | cpu  | cumulative | 35420000000.0 | ns   | 2016-12-29T11:53:57.162166 |
| 24601d87-456e-4b6c-98b4-5e112fd08bf7 | cpu  | cumulative | 32340000000.0 | ns   | 2016-12-29T11:53:43.879665 |
| d226b22e-a723-4e17-8043-fb1ea2e32488 | cpu  | cumulative | 36170000000.0 | ns   | 2016-12-29T11:53:43.879665 |
| 432ccae0-f554-4da5-9c92-6269d32bc472 | cpu  | cumulative | 35410000000.0 | ns   | 2016-12-29T11:53:43.879665 |
| 24601d87-456e-4b6c-98b4-5e112fd08bf7 | cpu  | cumulative | 32320000000.0 | ns   | 2016-12-29T11:52:57.340629 |
| d226b22e-a723-4e17-8043-fb1ea2e32488 | cpu  | cumulative | 36140000000.0 | ns   | 2016-12-29T11:52:57.340629 |
| 432ccae0-f554-4da5-9c92-6269d32bc472 | cpu  | cumulative | 35380000000.0 | ns   | 2016-12-29T11:52:57.340629 |

Ceilometer meters have default collection intervals, and the data can be displayed as a cumulative, actual value (gauge) or delta (difference from previous value). Most of those collection parameters can be modified.

4. Q: What is the time interval for samples in the ceilometer sample-list --meter cpu command?

5. Enter the following command to specify the number of samples (5) to be displayed for the disk.write.bytes meter:

```
ceilometer sample-list --meter disk.write.bytes -l 5
```

| Resource ID                          | Name             | Type       | Volume    | Unit | Timestamp                  |
|--------------------------------------|------------------|------------|-----------|------|----------------------------|
| 24601d87-456e-4b6c-98b4-5e112fd08bf7 | disk.write.bytes | cumulative | 4542464.0 | B    | 2016-12-29T11:57:46.059513 |
| d226b22e-a723-4e17-8043-fb1ea2e32488 | disk.write.bytes | cumulative | 207872.0  | B    | 2016-12-29T11:57:46.059513 |
| 432ccae0-f554-4da5-9c92-6269d32bc472 | disk.write.bytes | cumulative | 211968.0  | B    | 2016-12-29T11:57:46.059513 |
| 24601d87-456e-4b6c-98b4-5e112fd08bf7 | disk.write.bytes | cumulative | 4542464.0 | B    | 2016-12-29T11:56:57.110238 |
| d226b22e-a723-4e17-8043-fb1ea2e32488 | disk.write.bytes | cumulative | 207872.0  | B    | 2016-12-29T11:56:57.110238 |

A -q (or --query) parameter, which is added to many of the Ceilometer commands, lets you filter the output by a parameter specified in the display output (such as resource ID, name, or status).

You will need the ID of an instance (server) to query for meters being used to monitor that instance.

6. Run the openstack server list command to display the instances.

| ID                                   | Name                      | Status   Networks                         | Image Name             |
|--------------------------------------|---------------------------|-------------------------------------------|------------------------|
| 432ccae0-f554-4da5-9c92-6269d32bc472 | instfromvol               | ACTIVE   net4labs=10.10.1.113             |                        |
| d226b22e-a723-4e17-8043-fb1ea2e32488 | snappedImageInstance4labs | ACTIVE   net4labs=10.10.1.103             | snapImageInstance4Labs |
| 24601d87-456e-4b6c-98b4-5e112fd08bf7 | instance4labs             | ACTIVE   net4labs=10.10.1.107, 172.24.4.7 | image4labs             |

7. Enter the following command to query the meters associated with the instance4labs instance:

```
ceilometer meter-list -q "resource_id=<instance id for instance4labs>"
```

| Name                          | Type       | Unit      | Resource ID                          |
|-------------------------------|------------|-----------|--------------------------------------|
| compute.instance.booting.time | gauge      | sec       | 24601d87-456e-4b6c-98b4-5e112fd08bf7 |
| cpu                           | cumulative | ns        | 24601d87-456e-4b6c-98b4-5e112fd08bf7 |
| cpu.delta                     | delta      | ns        | 24601d87-456e-4b6c-98b4-5e112fd08bf7 |
| cpu.util                      | gauge      | %         | 24601d87-456e-4b6c-98b4-5e112fd08bf7 |
| disk.allocation               | gauge      | B         | 24601d87-456e-4b6c-98b4-5e112fd08bf7 |
| disk.capacity                 | gauge      | B         | 24601d87-456e-4b6c-98b4-5e112fd08bf7 |
| disk.ephemeral.size           | gauge      | GB        | 24601d87-456e-4b6c-98b4-5e112fd08bf7 |
| disk.read.bytes               | cumulative | B         | 24601d87-456e-4b6c-98b4-5e112fd08bf7 |
| disk.read.bytes.rate          | gauge      | B/s       | 24601d87-456e-4b6c-98b4-5e112fd08bf7 |
| disk.read.requests            | cumulative | request   | 24601d87-456e-4b6c-98b4-5e112fd08bf7 |
| disk.read.requests.rate       | gauge      | request/s | 24601d87-456e-4b6c-98b4-5e112fd08bf7 |
| disk.root.size                | gauge      | GB        | 24601d87-456e-4b6c-98b4-5e112fd08bf7 |
| disk.usage                    | gauge      | B         | 24601d87-456e-4b6c-98b4-5e112fd08bf7 |
| disk.write.bytes              | cumulative | B         | 24601d87-456e-4b6c-98b4-5e112fd08bf7 |
| disk.write.bytes.rate         | gauge      | B/s       | 24601d87-456e-4b6c-98b4-5e112fd08bf7 |
| disk.write.requests           | cumulative | request   | 24601d87-456e-4b6c-98b4-5e112fd08bf7 |
| disk.write.requests.rate      | gauge      | request/s | 24601d87-456e-4b6c-98b4-5e112fd08bf7 |
| instance                      | gauge      | instance  | 24601d87-456e-4b6c-98b4-5e112fd08bf7 |
| memory                        | gauge      | MB        | 24601d87-456e-4b6c-98b4-5e112fd08bf7 |
| memory.resident               | gauge      | MB        | 24601d87-456e-4b6c-98b4-5e112fd08bf7 |
| vcpus                         | gauge      | vcpus     | 24601d87-456e-4b6c-98b4-5e112fd08bf7 |

Each column provides a specific type of information:

- The **Name** column provides the information about the meter name.
- The **Type** column describes the type of measurement.
- The **Unit** column provides the information about the unit of measurement.
- The **Resource ID** and **User ID** columns (the latter is not shown on the screenshot) provide the identity information.

---

**NOTE:** If you create the instance other than the one specified in the `source openrc` command, you must specify the `--os-tenant-name` option.

---

As you recall, you only applied the traffic generator to the **instance4labs** instance, so there should be a noticeable CPU utilization difference between the two instances.

8. Generate a `sample-list` of the **cpu\_util** meter for the last five samples of both instances, and note the difference between the two instances (Hint: step 3).

| Resource ID                          | Name     | Type  | Volume          | Unit | Timestamp                  |
|--------------------------------------|----------|-------|-----------------|------|----------------------------|
| 24601d87-456e-4b6c-98b4-5e112fd08bf7 | cpu_util | gauge | 0.0684660175084 | %    | 2016-12-29T12:16:40.992197 |
| d226b22e-a723-4e17-8043-fb1ea2e32488 | cpu_util | gauge | 0.0456440116723 | %    | 2016-12-29T12:16:40.992197 |
| 432ccae0-f554-4da5-9c92-6269d32bc472 | cpu_util | gauge | 0.0684660175084 | %    | 2016-12-29T12:16:40.992197 |
| 24601d87-456e-4b6c-98b4-5e112fd08bf7 | cpu_util | gauge | 0.150838731223  | %    | 2016-12-29T12:15:57.174839 |
| d226b22e-a723-4e17-8043-fb1ea2e32488 | cpu_util | gauge | 0.301677462446  | %    | 2016-12-29T12:15:57.174839 |

9. Use a similar command to list samples for various other meters such as `cpu.delta`, `disk.ephemeral.size`, and `disk.root.size`.

## Ceilometer in the Horizon UI

1. Access the Horizon UI and log in to the domain of `default` as `admin / hpinvent`
2. Click **Admin** → **System** → **Resource Usage** → **Usage Report** tab from the navigation panel.
3. Note the following points of interest:
  - Not all of the information displayed is from Ceilometer.
  - Each resource is associated with its project and the OpenStack® service that provided the data.
  - The meter name and its description are provided.
  - The date of collection is provided.
  - The average value of the meter is provided with the associated units (dimensions).
  - Click the **Modify Usage Report Parameters>** button to modify the usage report date ranges.
  - You can download the report to a `.csv` file.

The screenshot shows the 'Resources Usage Overview' page under the 'System' section of the navigation bar. The 'Usage Report' tab is highlighted with a red box. The table below lists resource usage data:

|                 | Project      | Service | Meter      | Description         | Day        | Value (Avg)     | Unit |
|-----------------|--------------|---------|------------|---------------------|------------|-----------------|------|
| Host Aggregates | admin        | Glance  | image.size | Uploaded image size | 2016-12-29 | 48,922,199.3799 | B    |
| Instances       | project4labs | Glance  | image.size | Uploaded image size | 2016-12-29 | 22,020,096.0    | B    |
| Volumes         | admin        | Glance  | image.size | Image is served out | 2016-12-29 | 13,287,936.0    | B    |
| Flavours        |              |         |            |                     |            |                 |      |

4. Click the **Stats** tab.

You will likely have very little or no information displayed on the graph because the smallest display value is for one day. You might be able to see some data by moving your mouse to the left-most axis of the table.

## Resources Usage Overview

The screenshot shows the 'Stats' tab of the 'Resources Usage Overview' page. The form fields are:

- Metric:** storage.objects
- Group by:** Project
- Value:** Avg.
- Period:** Last week
- From:** 2016-12-28
- To:** 2016-12-31

Statistics of all resources

No data available.

---

**NOTE:** The Ceilometer resource usage functionality does not function properly in your environment. You might actually not be able to get statistical data in Horizon.

---

5. Click various drop-down boxes to display various meters, grouping, values, and periods available for filtering the displayed information.
6. Please let your instructor know you have completed Lab 10.

---

# Fundamentals of OpenStack® Technology

## Appendix—Installing and Configuring DevStack

### Objective

After completing this lab, you should be able to:

- Explain the prerequisites for installing DevStack
- Discuss various modes of DevStack installation
- Explain the steps to install the all-in-one, single-VM DevStack environment

### Requirements

- Internet connection
- A VMware workstation or an ESXi virtual host

### Introduction

As explained on the DevStack official page (<http://docs.openstack.org/developer/devstack/>), DevStack is a script used to quickly create an OpenStack® development environment. It was created to build a running OpenStack® environment with minimal effort and is used by most project developers. Do not try to use DevStack in the production environment.

Currently, DevStack can be deployed in three scenarios:

- All-In-One Single VM
- All-In-One Single Machine
- Multi-Node Lab

This appendix covers the All-In-One Single VM scenario.

## Exercise 1—Meeting the DevStack requirements

As previously explained, you will run DevStack within the virtual machine. A host running the hypervisor must be running on the CPU supporting the Intel-VT technology, because your guest (DevStack) VM must run a hypervisor. In VMware terminology, this setup is called “a nested hypervisor.”

Your virtual machine requirements are:

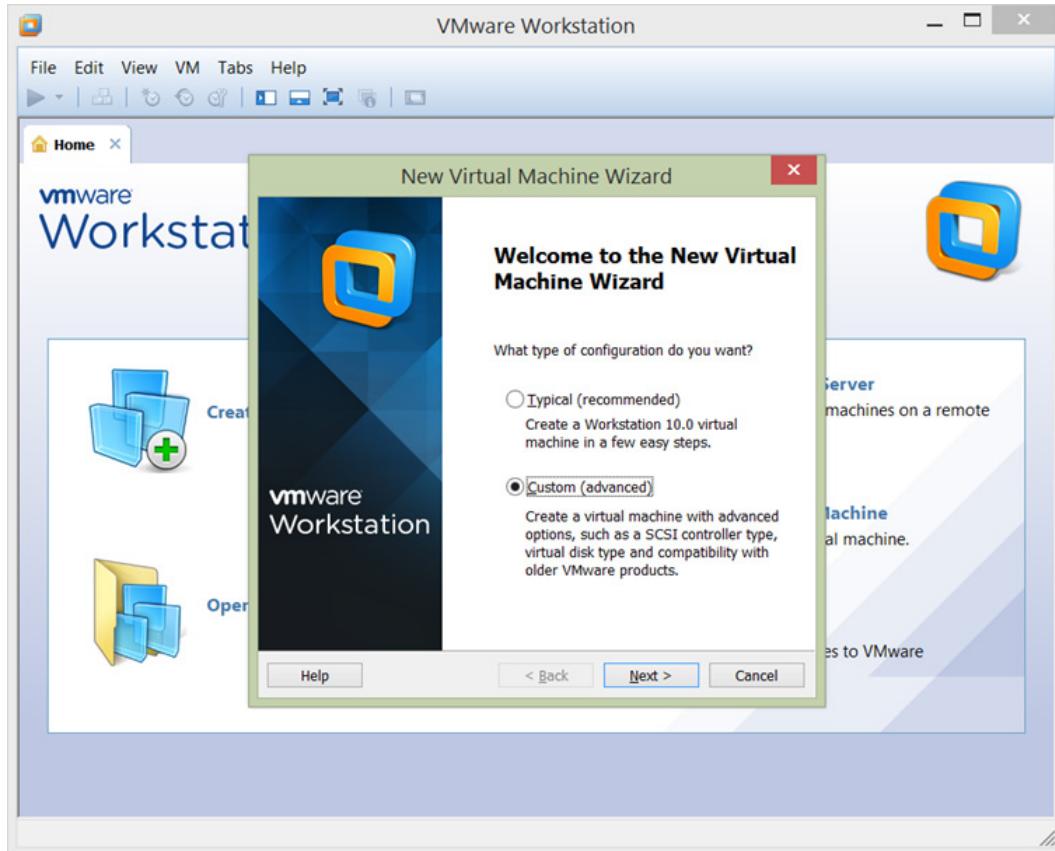
- 8GB of RAM memory
- 60GB large Virtual HD
- Two VCPUs
- One Virtual Network adapter with Internet connectivity
- Hypervisor Nesting enabled

If you decide to run DevStack using the VMware workstation (for example), you might need twice as much memory. You can run the DevStack environment with less than 30GB of HDD space, but your environment will lack the disk space for anything except the “CirrOS” demo OS installation.

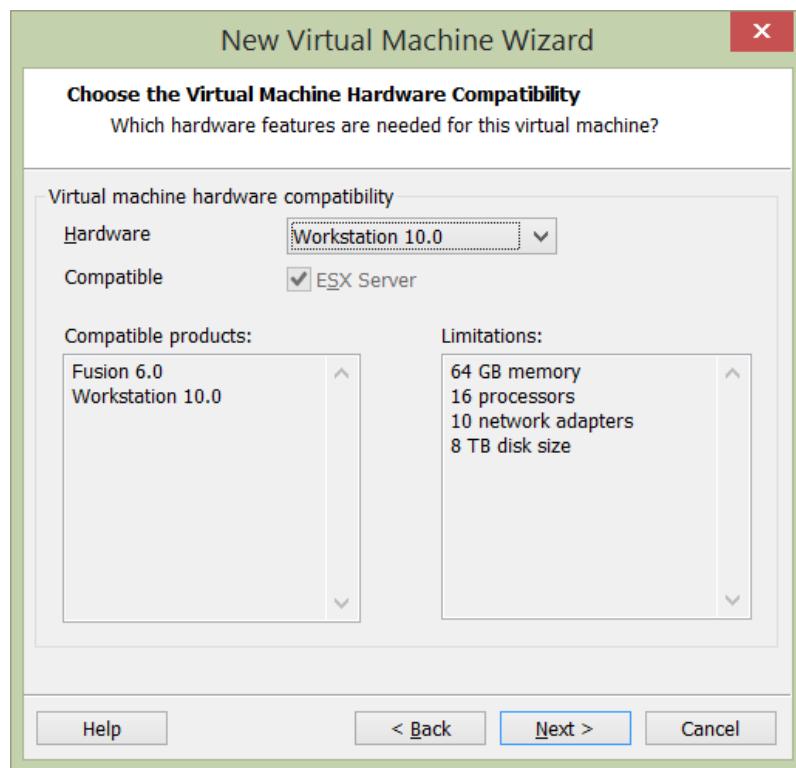
## Exercise 2—Preparing the VM using a VMware workstation

In this example, you will use a VMware workstation running on the Windows platform, but configuration will be similar for other platforms.

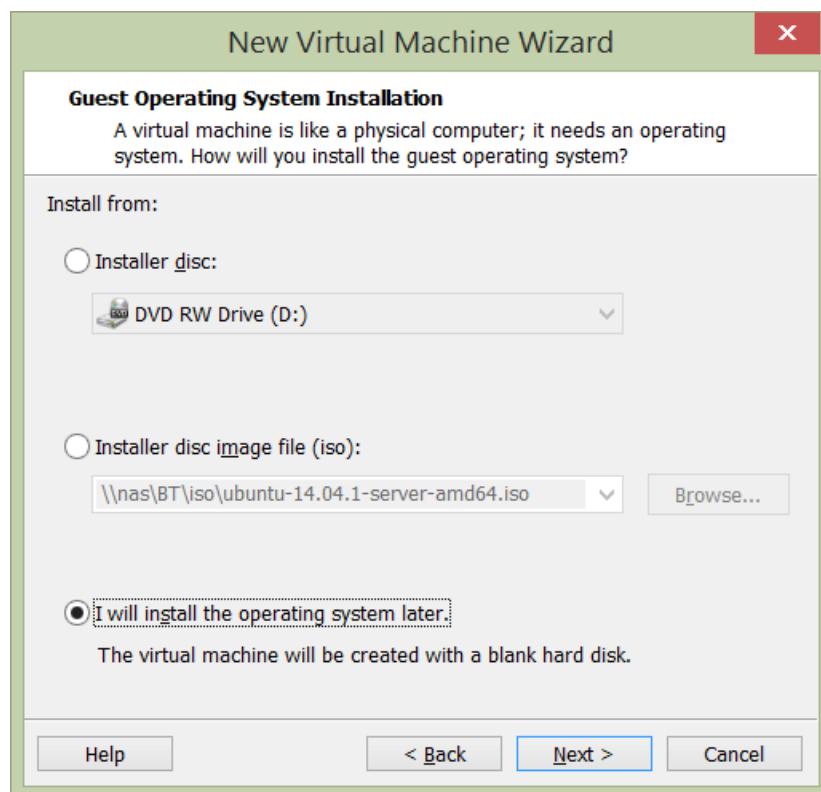
1. On your Windows host, start the VMware workstation, and press the **Ctrl+N** keys to start the creation of a new virtual machine.
2. On the New Virtual Machine Wizard dialog, select **Custom (advanced)** and press **Next** to continue.



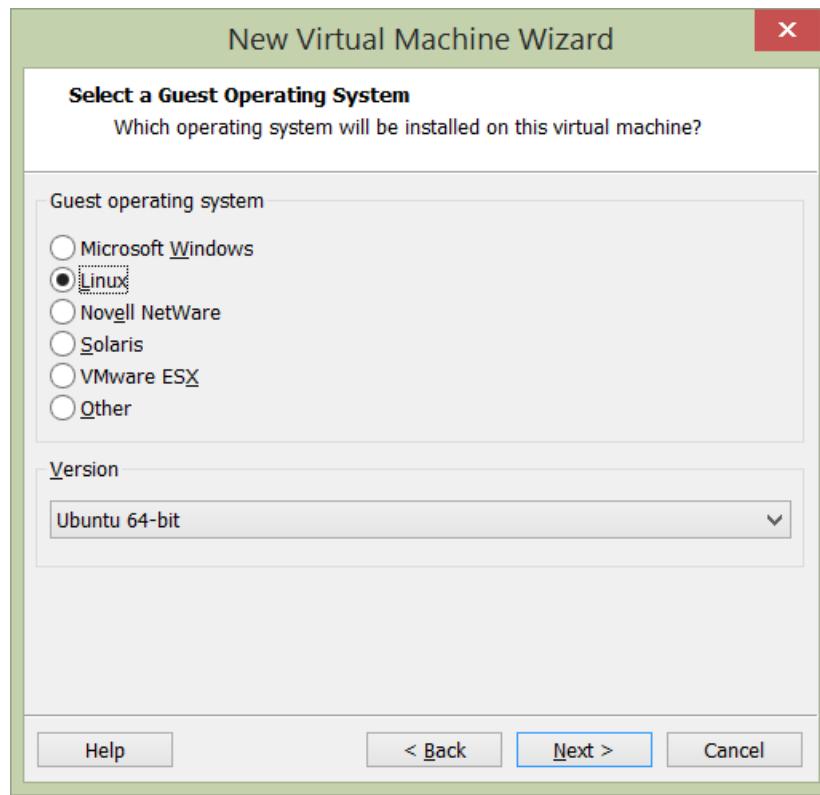
3. Keep the default values and press **Next** to continue.



4. Select the option **I will install the operating system later**, and click **Next** to continue.



5. Select **Linux** from the list of supported operating systems. From the Version drop-down list, select **Ubuntu 64-bit**.

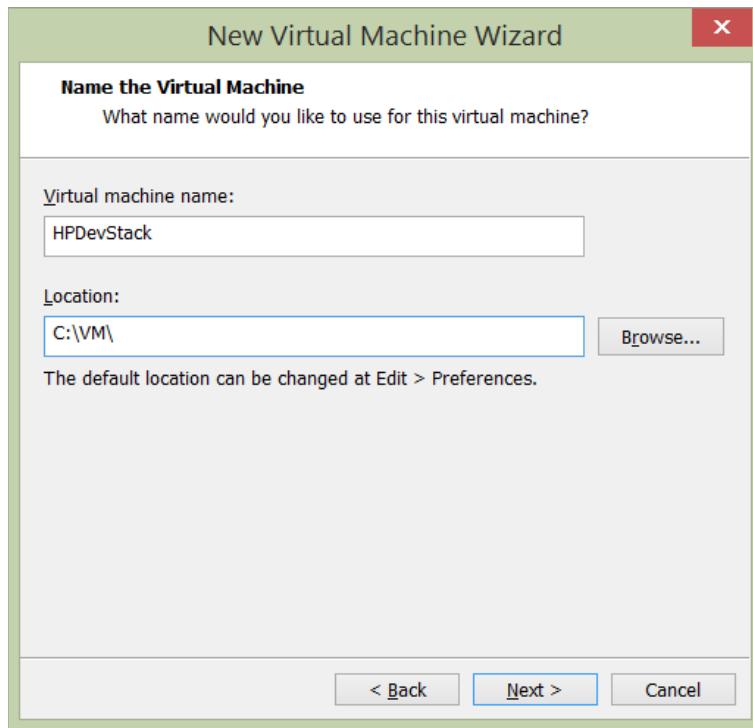


---

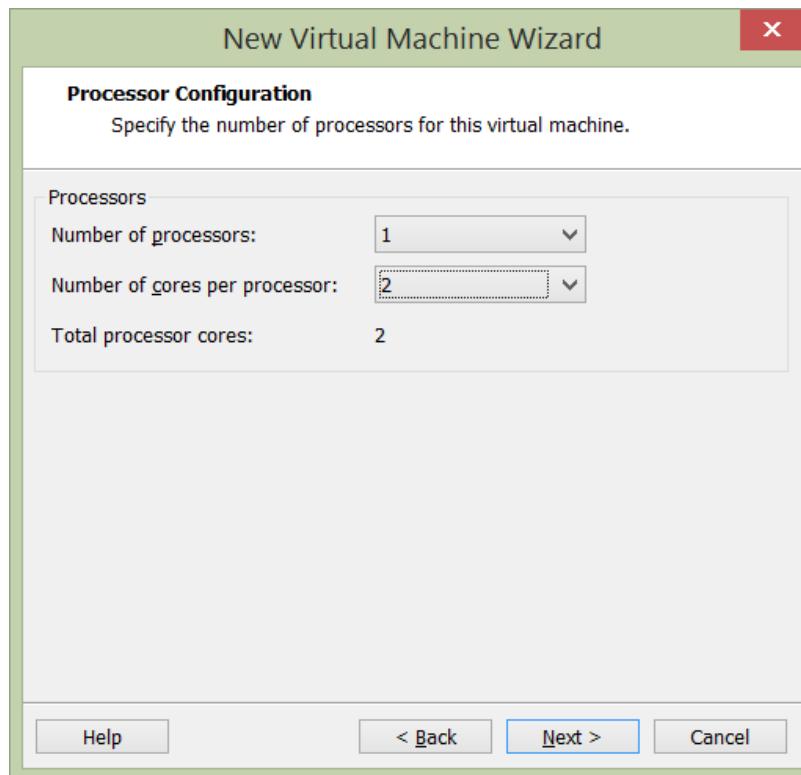
**NOTE:** Currently, DevStack environment supports Ubuntu 16.04 64-bit LTS and CentOS. For this environment, Ubuntu is used.

---

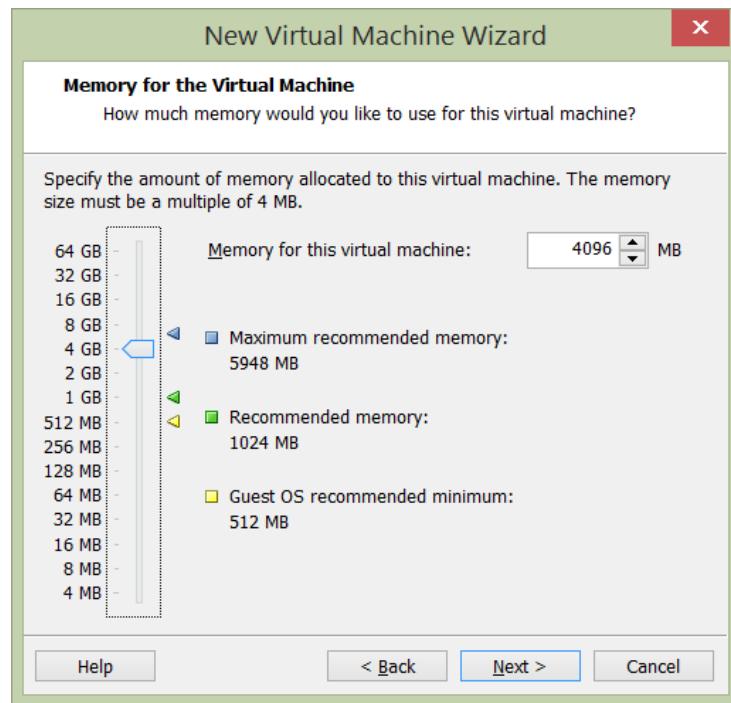
6. Provide the virtual machine name (for example `HPDevStack`), and a location for your VM files. Click **Next** to continue.



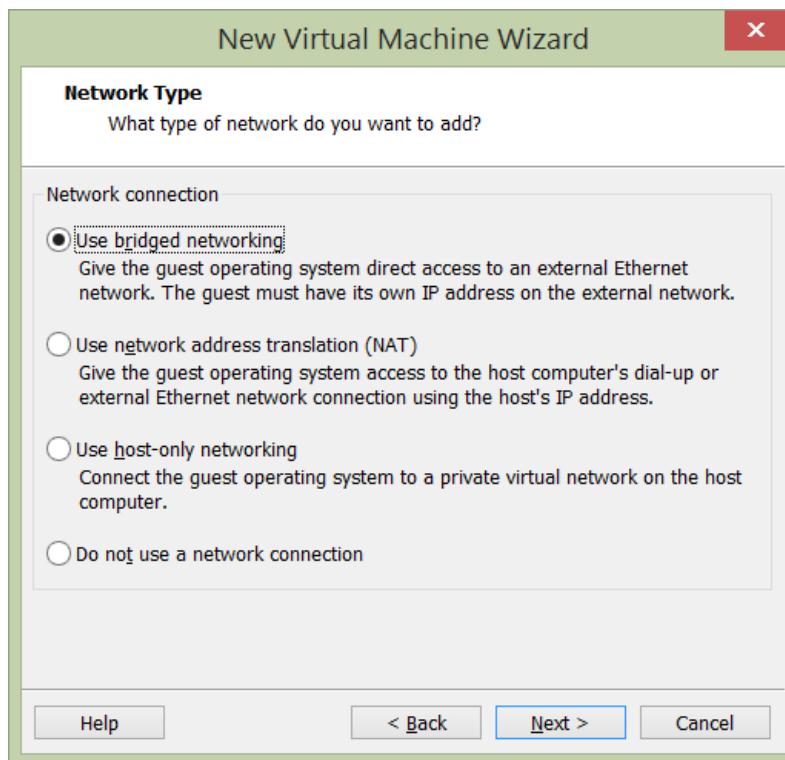
7. Set the total number of processor cores to **minimum 2**. Click **Next** to continue.



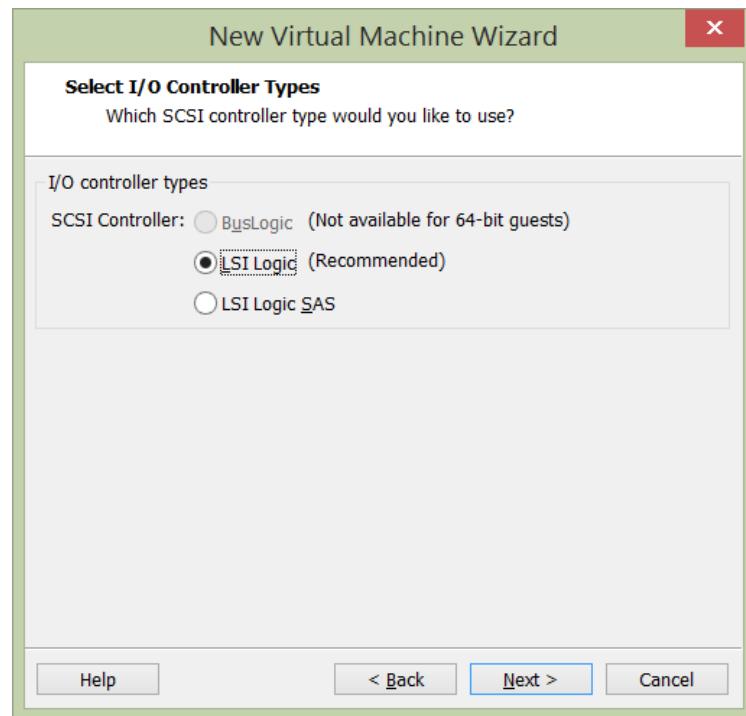
8. Set the amount of memory to **4096** and click **Next** to continue.



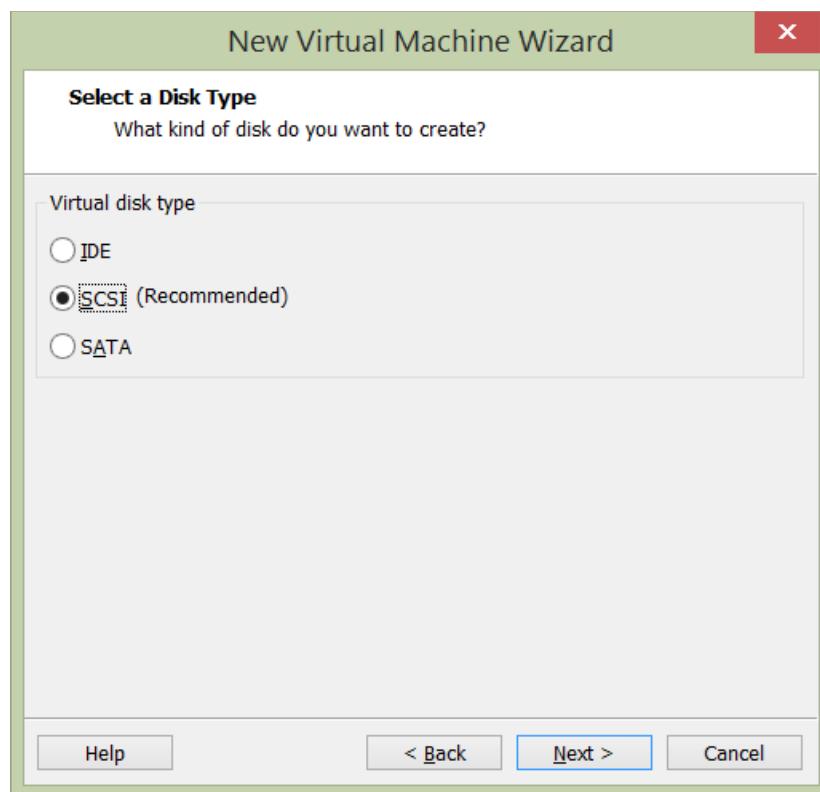
9. You can select either the **bridge** or **NAT** setting for the network. Your selection does not influence the manner in which the VM runs. The only thing that differs depending on the setup you choose is the accessibility of your DevStack from external networks. In this case, choose the bridge setting and click **Next** to proceed.



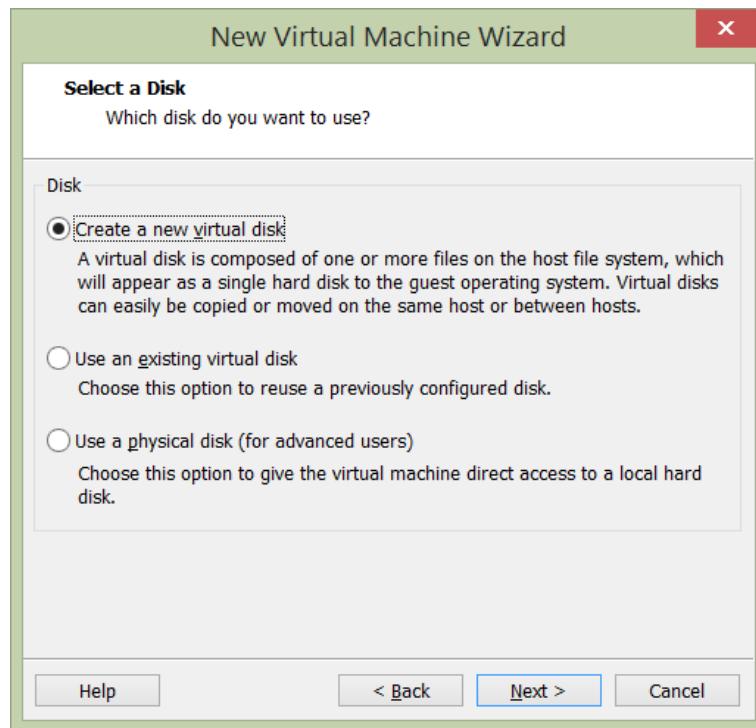
10. Use the recommended settings and click **Next**.



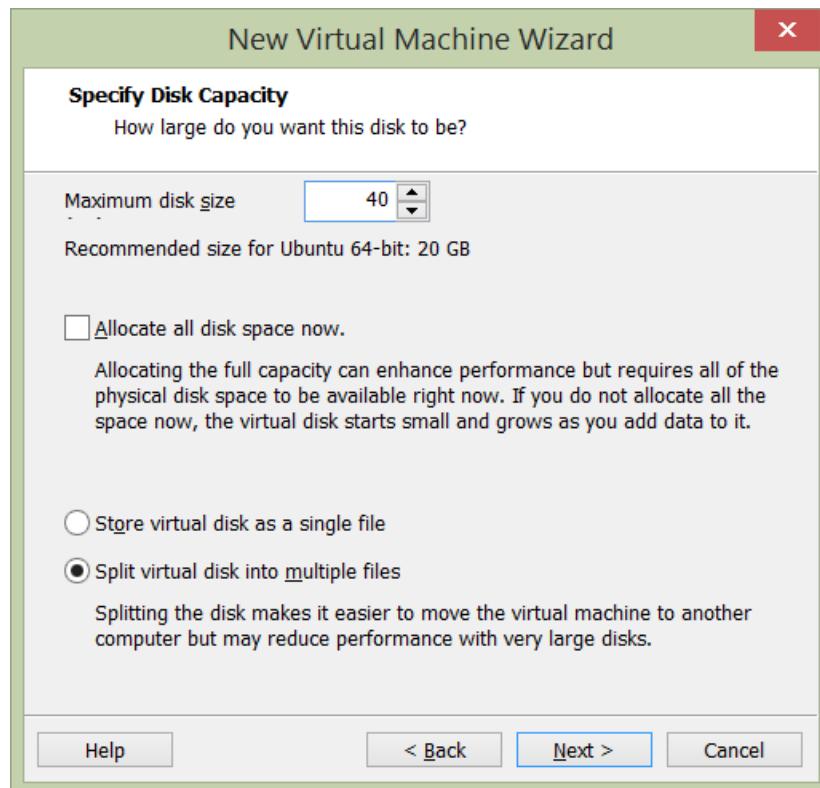
11. Leave the default Virtual Disk Type and click **Next**.



12. Click **Next** to create a new Virtual Machine Disk.

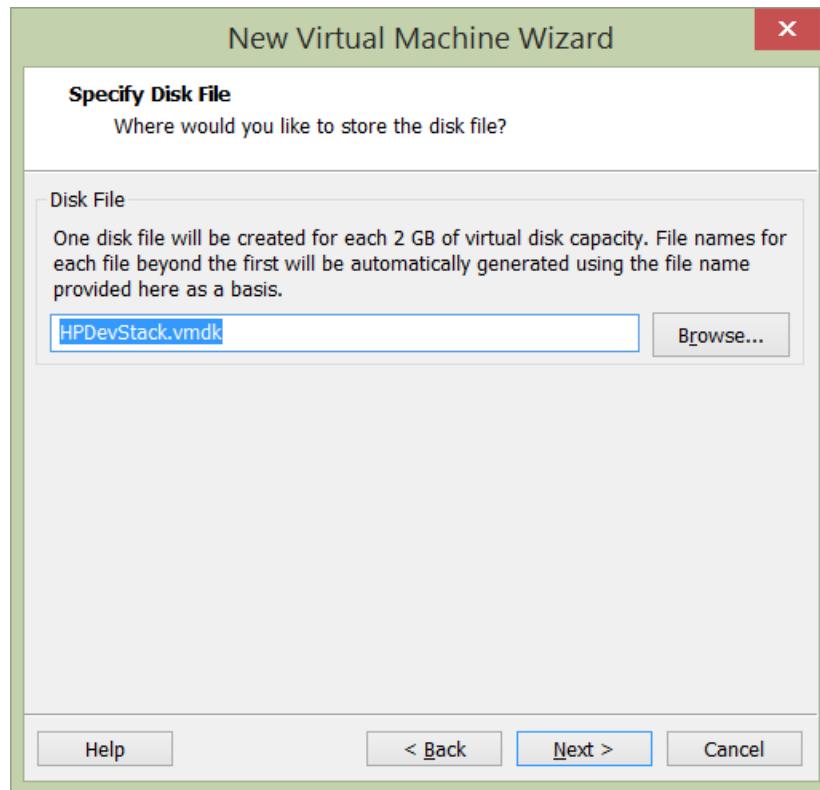


13. Enter the maximum disk size and click **Next**.

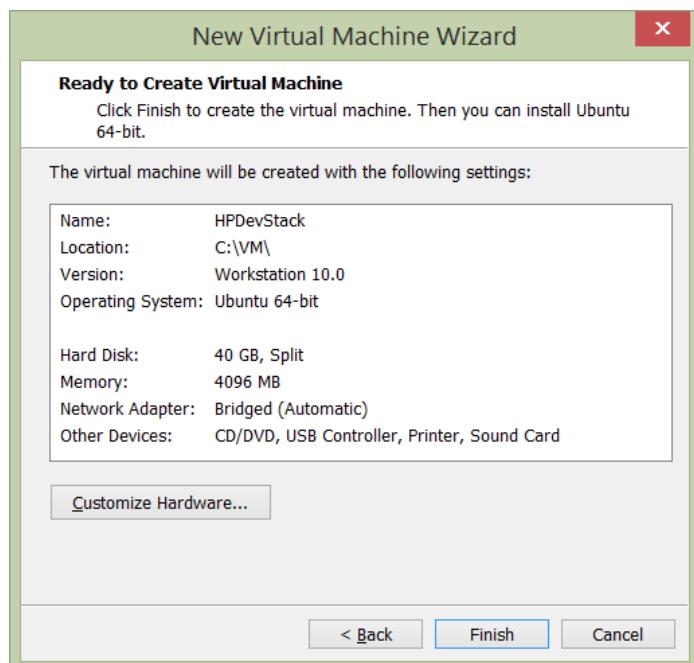


**IMPORTANT:** In this case, a 40GB large disk is used, but the required space depends on what you plan to do with your environment. For example, working with Cinder might require more disk space. Any database (Trove) instance requires a minimum of 5GB of disk space. Running two VMs at the same time might exhaust the available disk space, because this HDD is shared between the DevStack and the guest virtual machines.

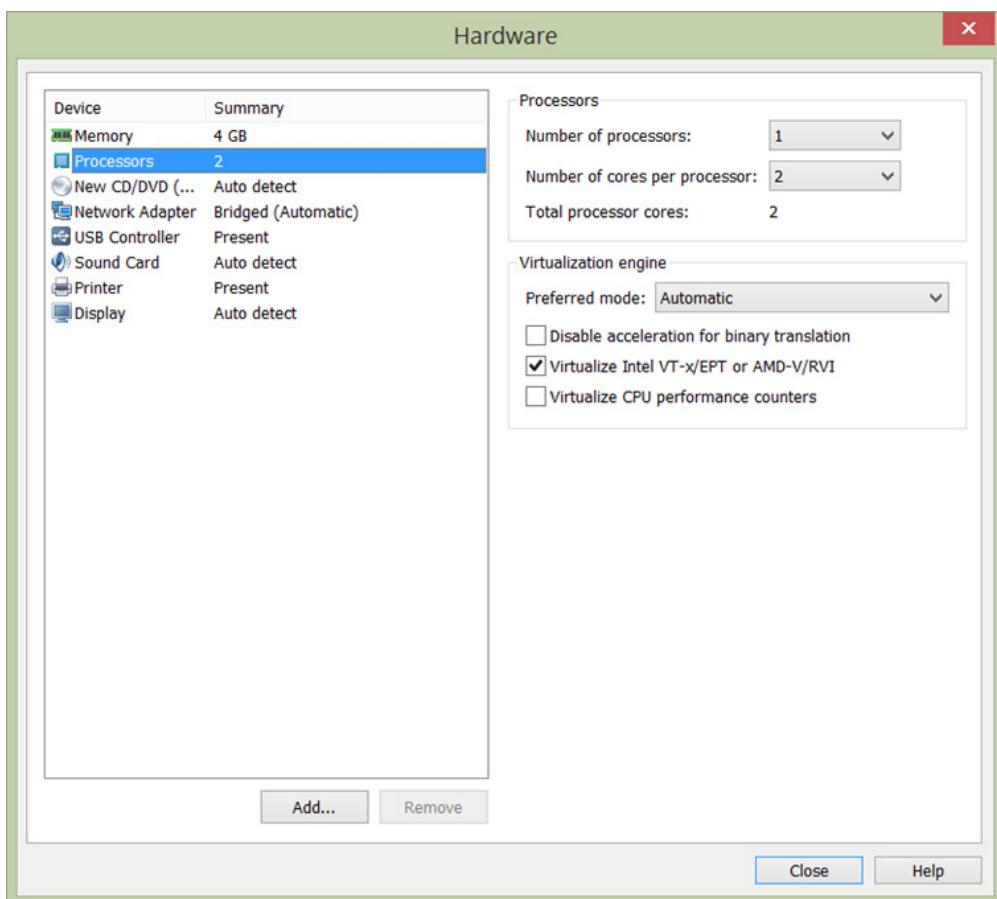
14. Click **Next** to confirm the VMDK disk file name.



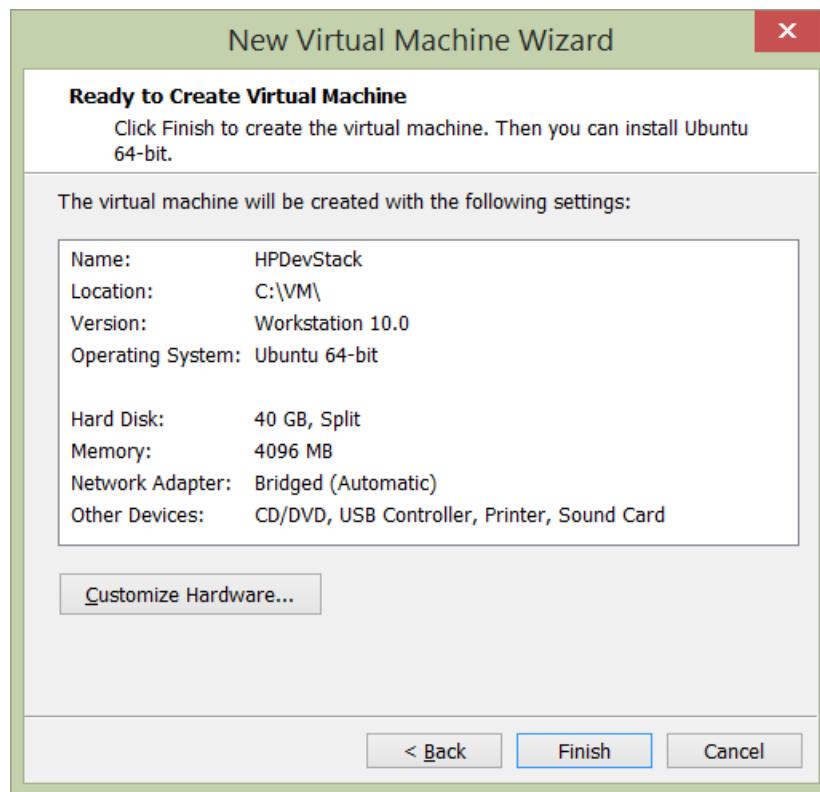
15. On the Ready to Create Virtual Machine screen, click **Customize Hardware**.



16. In the Hardware window, on the left-hand side, select **Processors**. Within the Virtualization Settings, select **Virtualize Intel VT-x/EPT or AMD-V/RVI**, and click **Close**.



17. Back at the Ready to Create Virtual Machine screen, click **Finish**.



## Exercise 3—Operating system installation

18. Before starting the operating system installation, provide the **ubuntu-14.04.1-server-amd64.iso** image to the virtual CD-ROM device of your virtual machine.
19. Using the cursor keys on your keyboard, move to select the language, and press **Enter**.



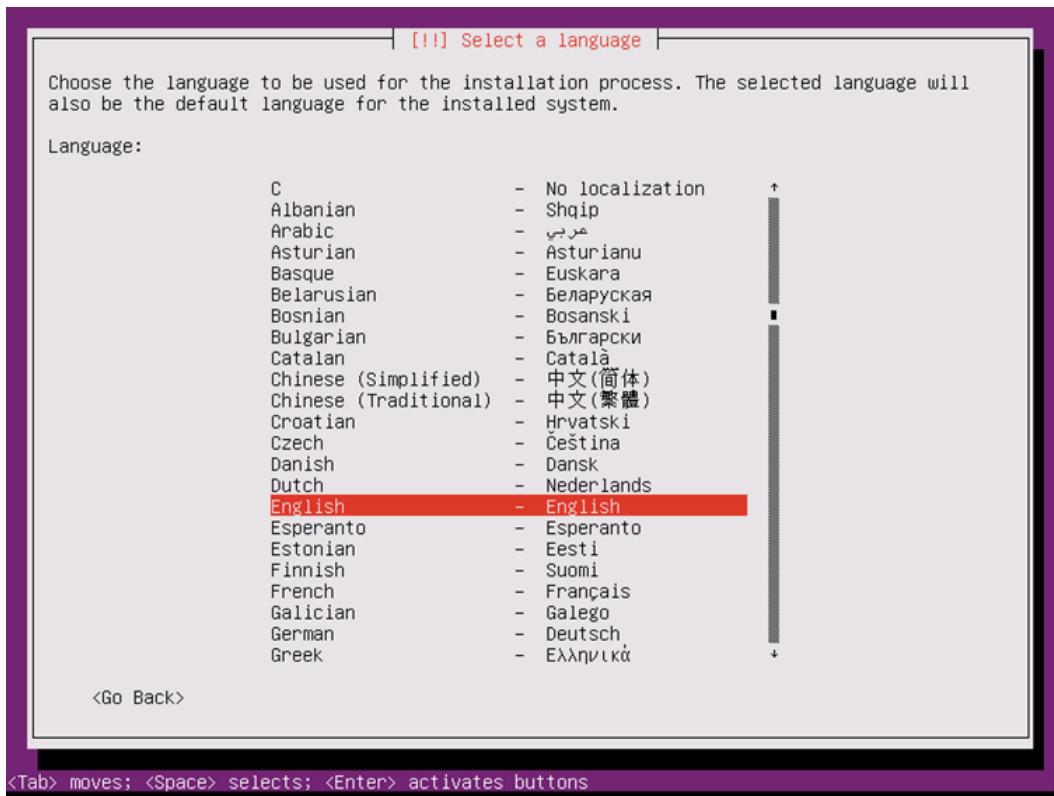
20. The main installer screen offers many installation options. While keeping the Install Ubuntu Option highlighted, press **F4** at your keyboard, and select the **Install a minimal virtual machine** option by pressing **Enter**.



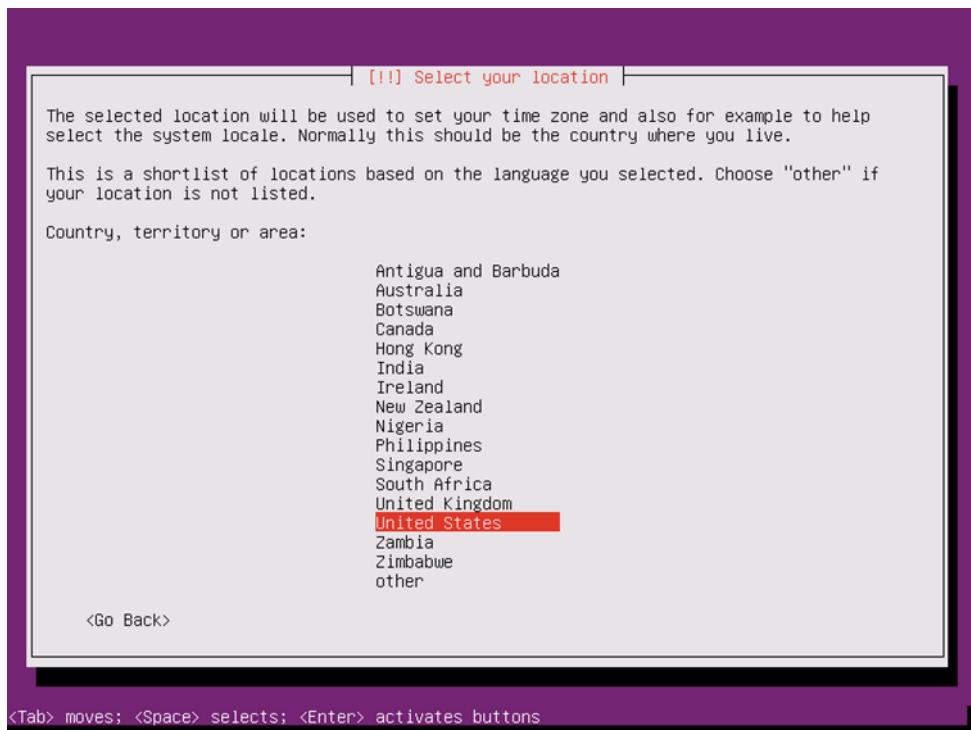
21. Back at the main screen, press **Enter** to begin the installation.



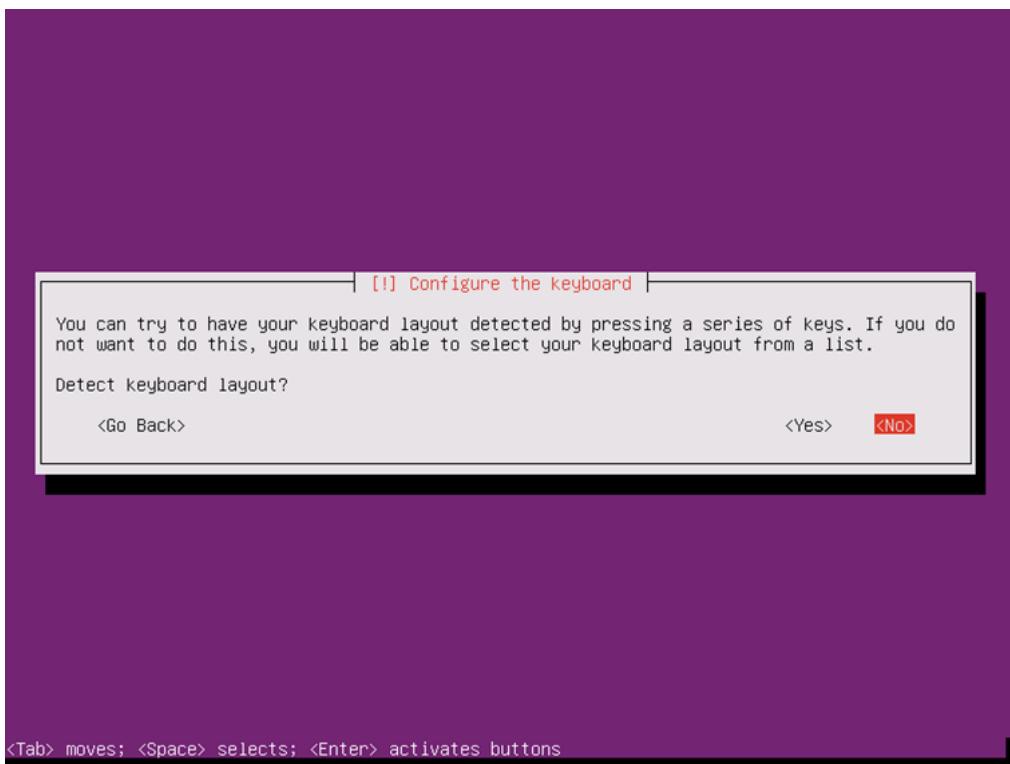
22. At the Ubuntu Installer screen, press **Enter**.



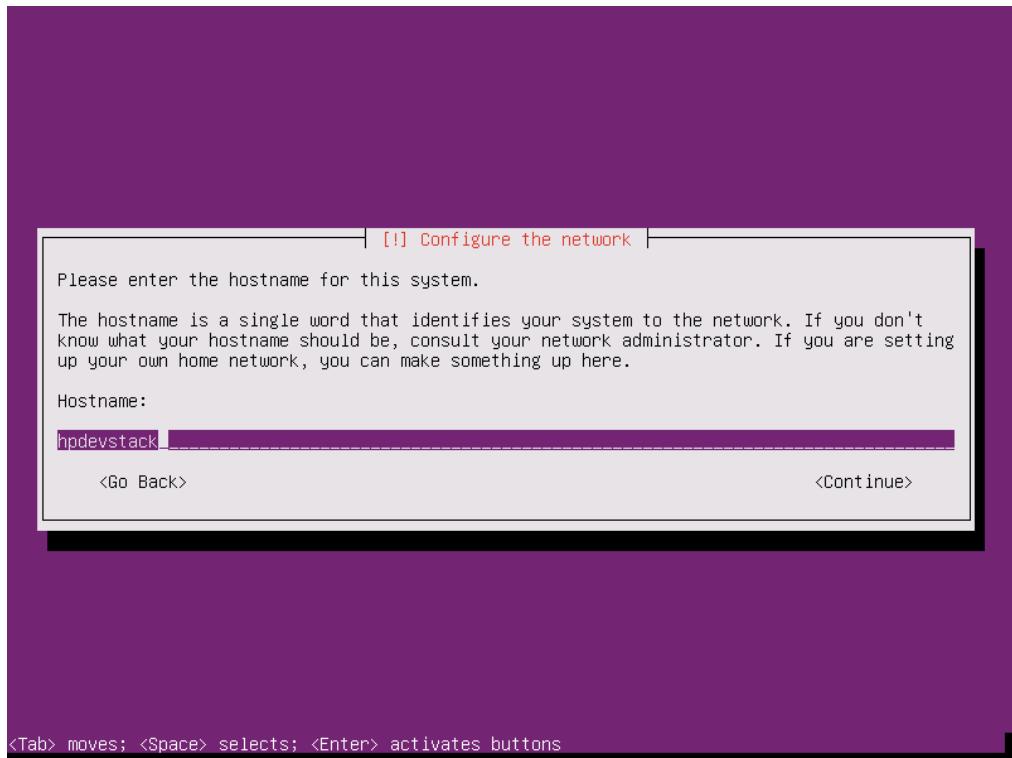
23. Select a country from the list and press **Enter**.



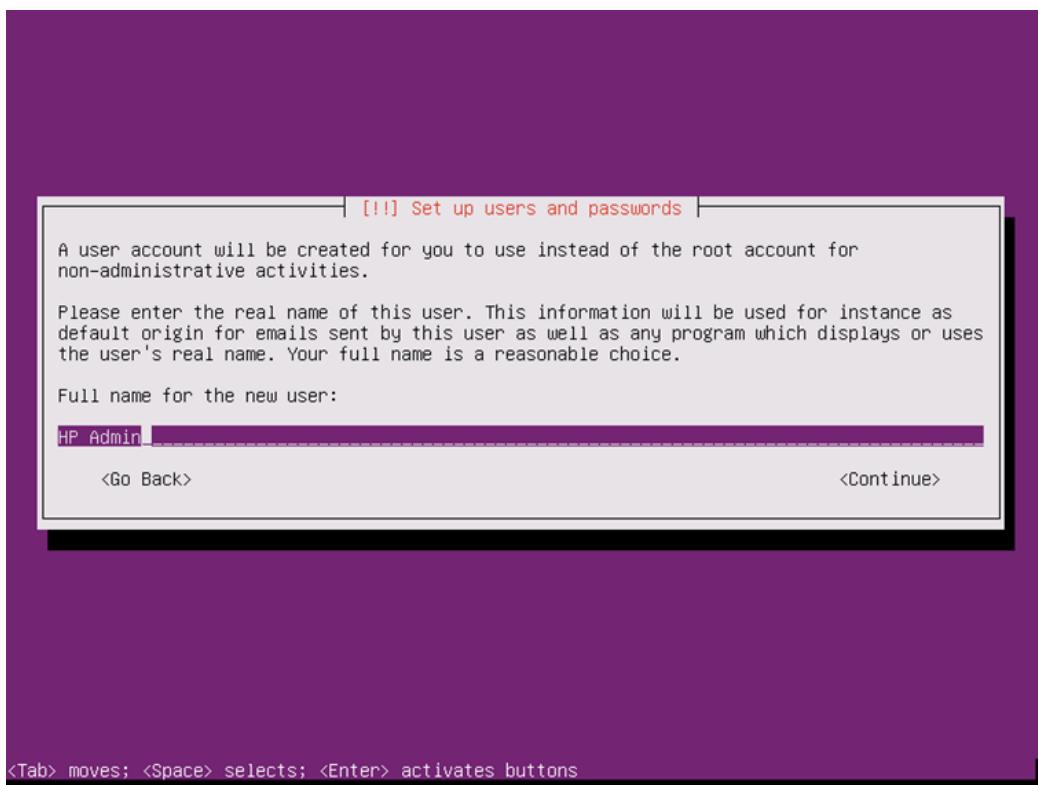
24. If necessary, configure your keyboard, or press **Enter** to continue.



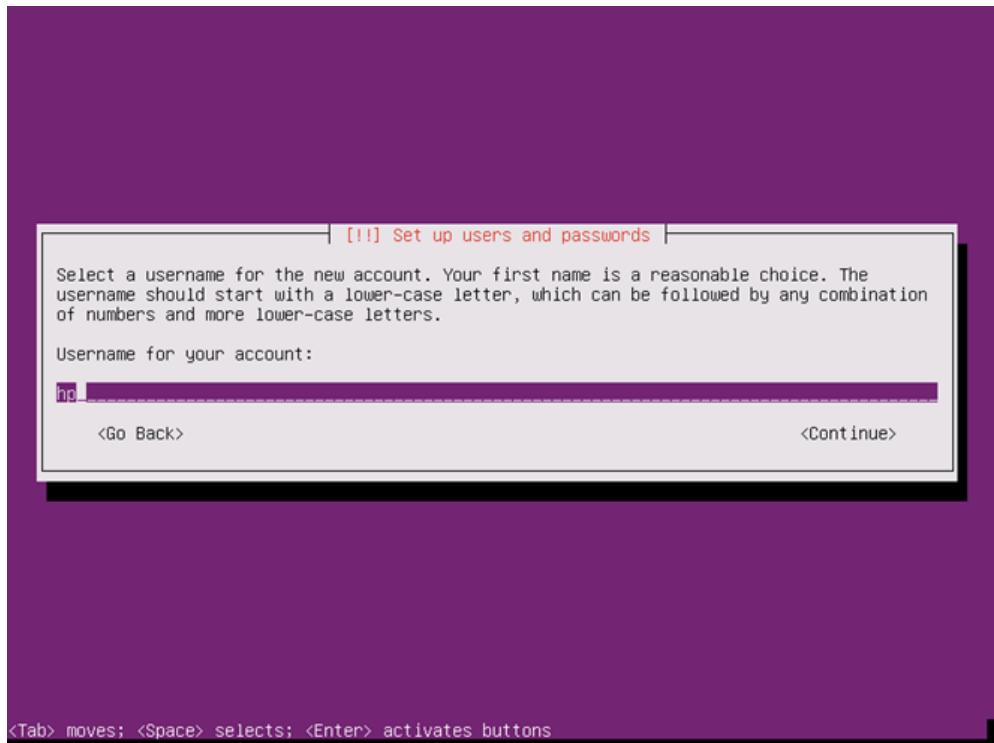
25. Provide the host name for the virtual machine.



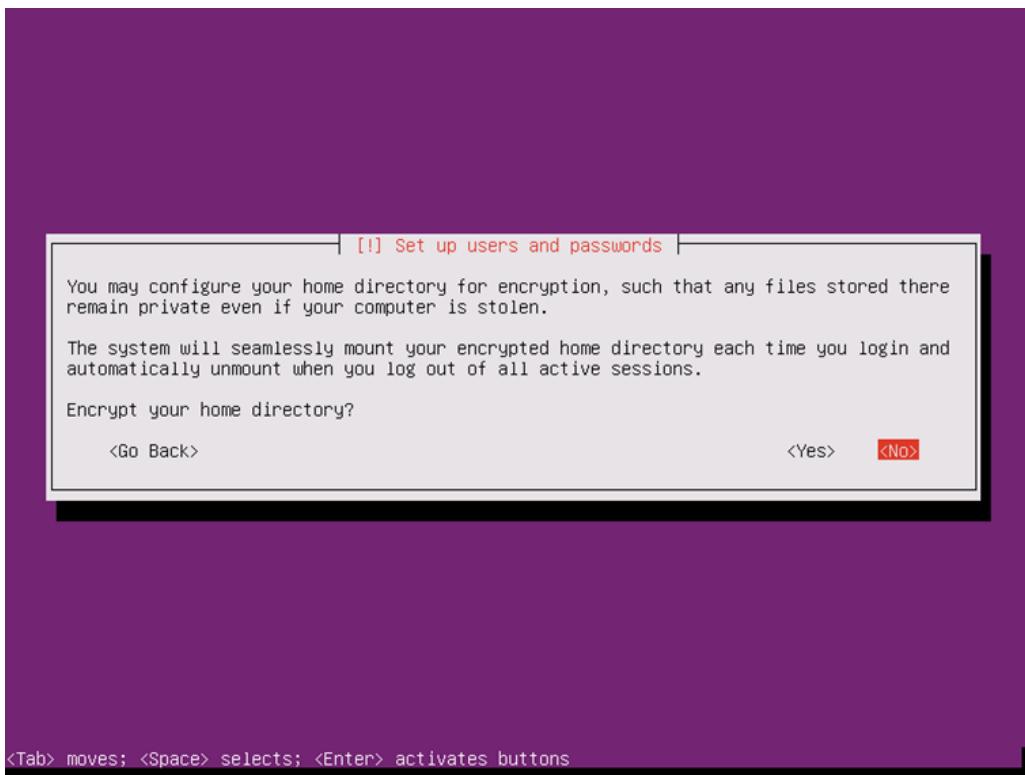
26. Provide your name to create a default user account.



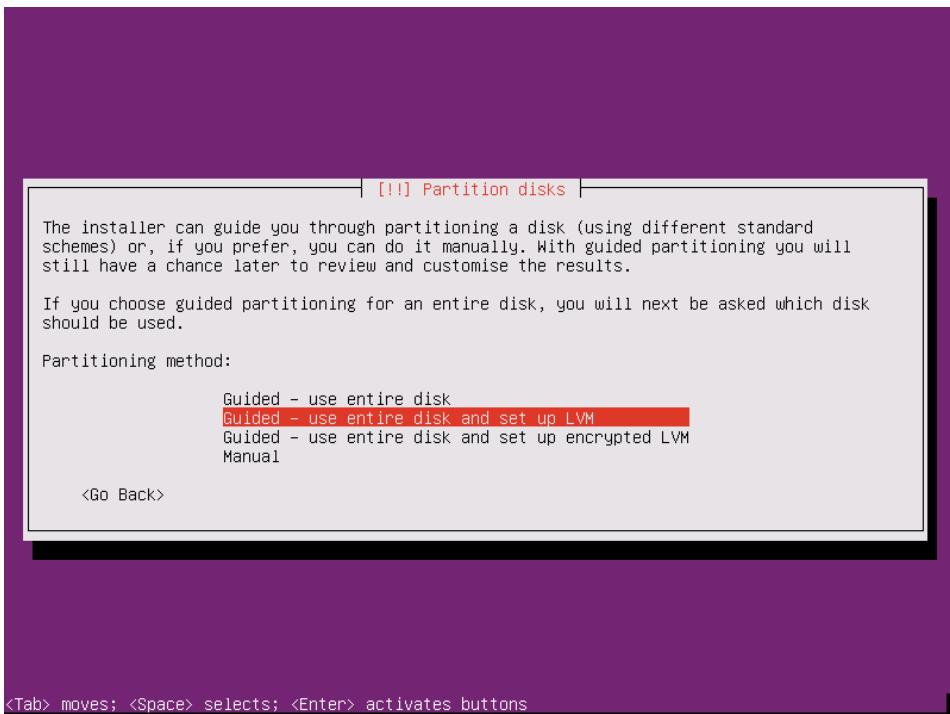
27. Provide the username and password for the default user account.



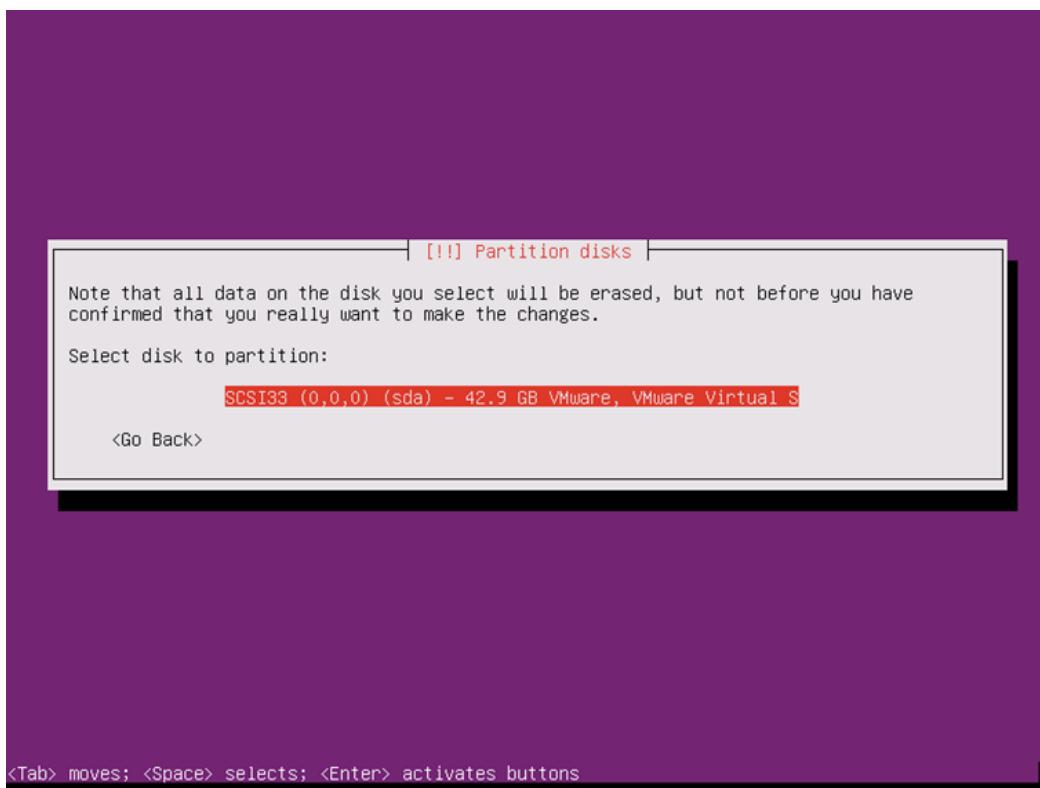
28. Do not encrypt your home directory to gain some speed. Press **Enter** to continue.



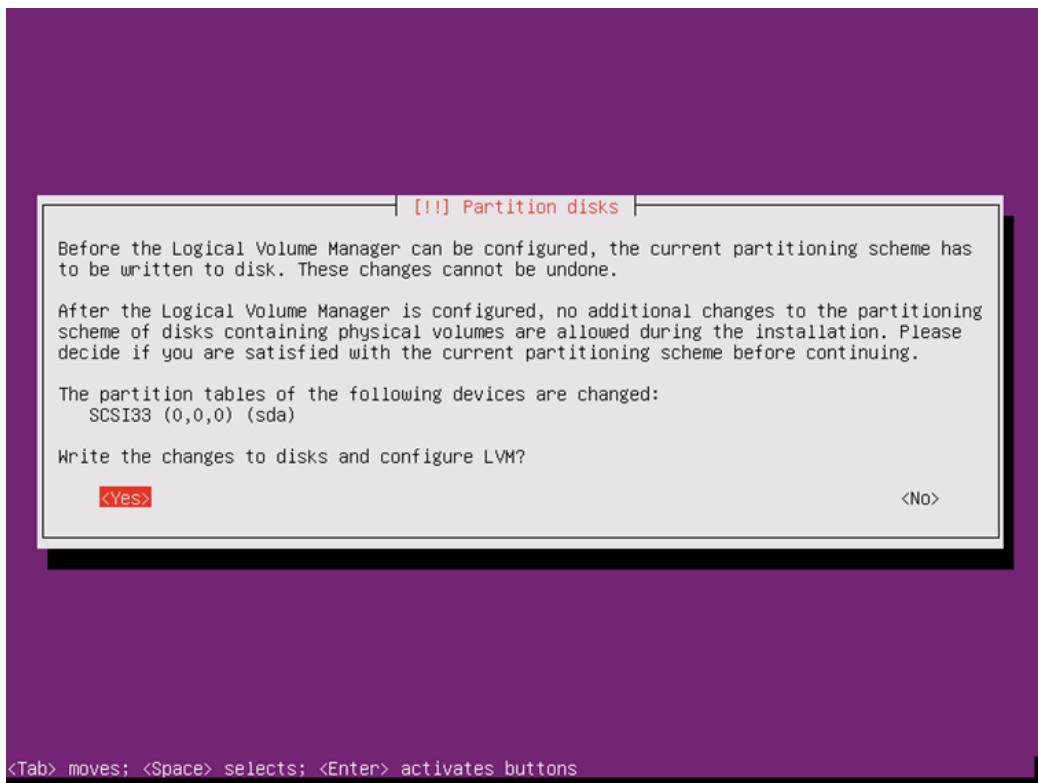
29. Set your time zone.
30. For partitioning method, select **Guided – use entire disk and setup LVM**.



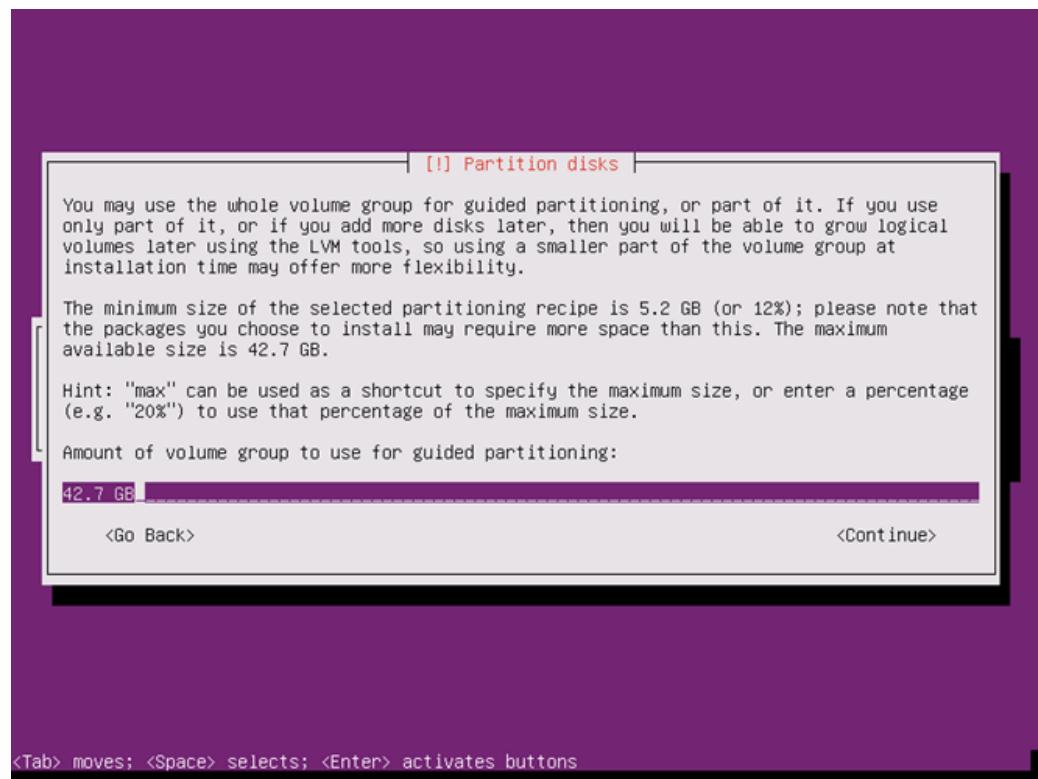
31. Use the defaults for disk partition. Press **Enter**.



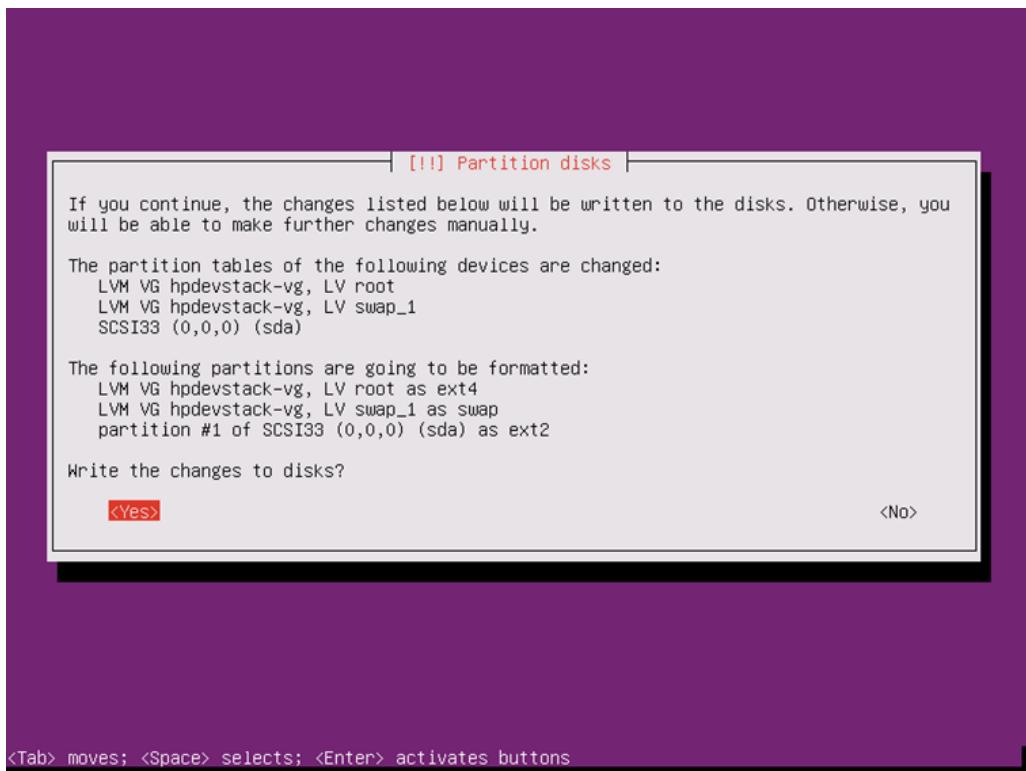
32. Select **Yes** and press **Enter**.



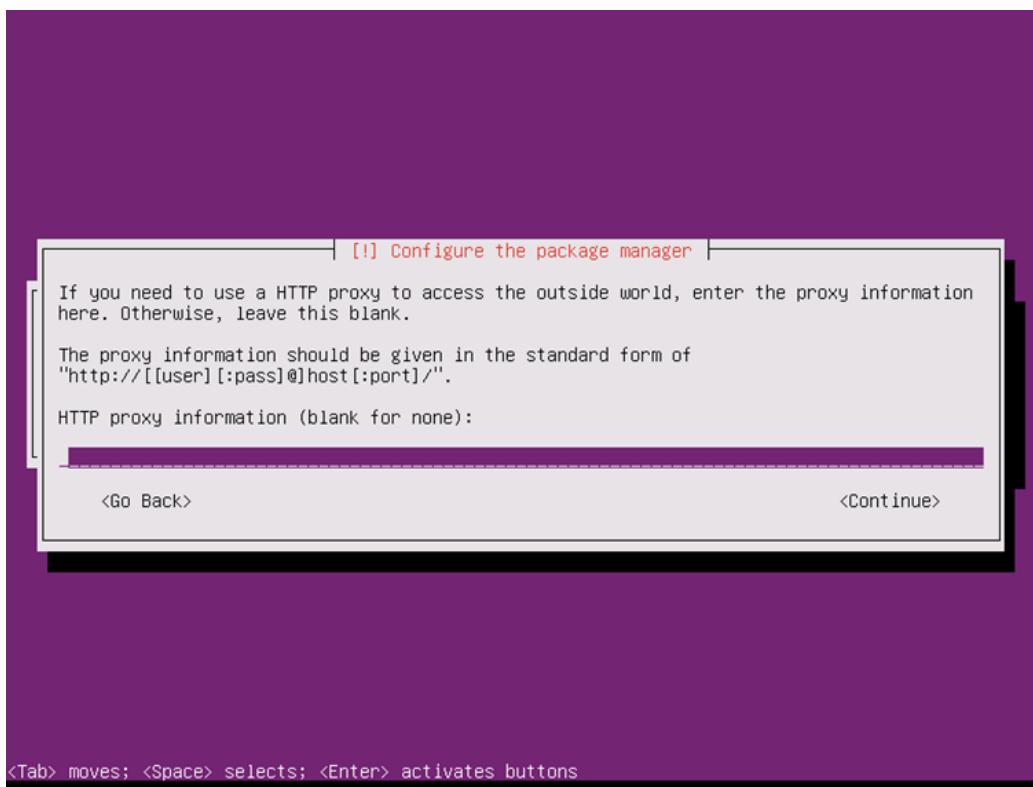
33. Press **Enter** to accept the defaults.



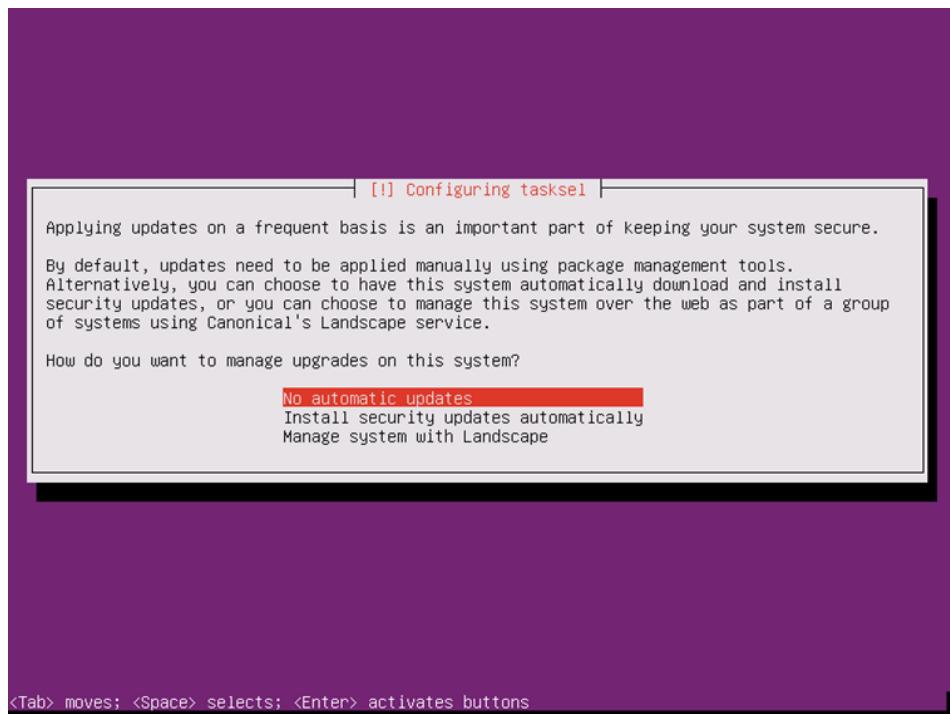
34. Select **Yes** and press **Enter** to confirm the partition configuration.



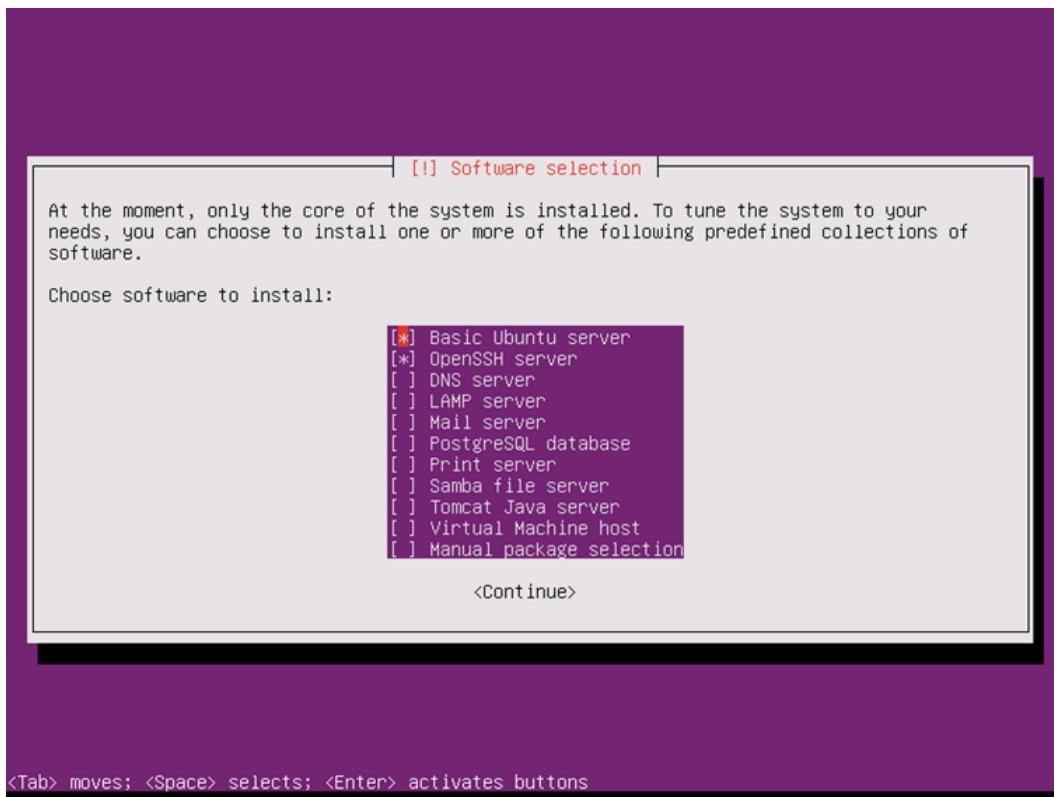
35. Provide the proxy configuration settings for your environment.



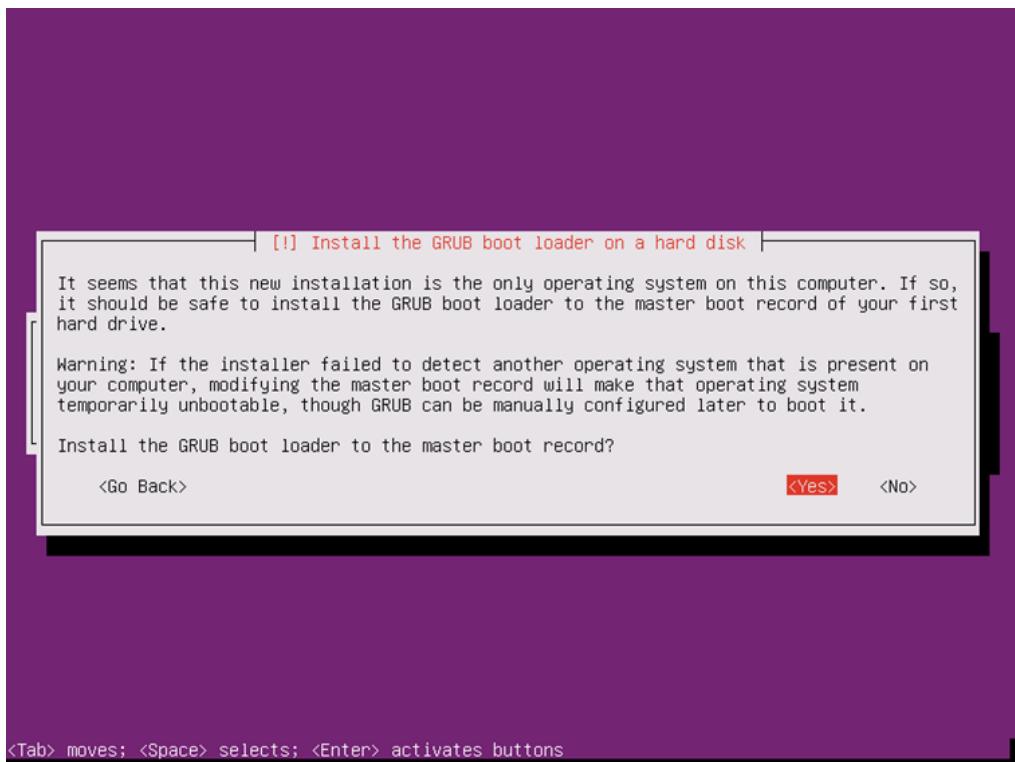
36. Select the **No automatic updates** option and press **Enter**.



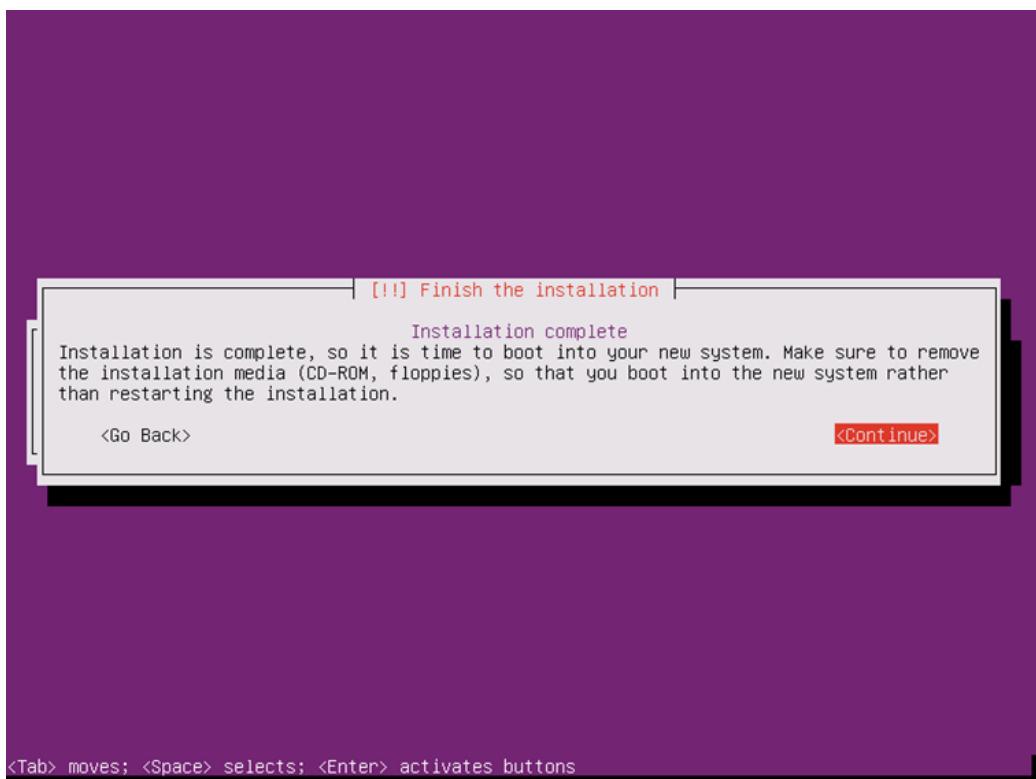
37. Use the up and down keys to move your cursor, and the spacebar to select options, as shown in the picture below. Press **Enter** to proceed.



38. At the boot loader screen, press **Enter** to continue.



39. To finish the installation and reboot your virtual machine, select **Continue** and press **Enter**.



40. Wait for the system to finish the installation and reboot.
41. After the system has booted, log in to the terminal using the user's credentials you have provided during the installation, and execute the following command to find the IP address of your VM:

```
hp@hpdevstack:~$ ifconfig eth0
eth0 Link encap:Ethernet HWaddr 00:0c:29:0a:16:4a
 inet addr:192.168.5.143 Bcast:192.168.5.255 Mask:255.255.255.0
 inet6 addr: fe80::20c:29ff:fe0a:164a/64 Scope:Link
 UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
 RX packets:281 errors:0 dropped:0 overruns:0 frame:0
 TX packets:84 errors:0 dropped:0 overruns:0 carrier:0
 collisions:0 txqueuelen:1000
 RX bytes:25693 (25.6 KB) TX bytes:15036 (15.0 KB)
```

42. Minimize your VMware window and use an SSH client such as PuTTY to log in to your virtual machine.
43. Leave the PuTTY session open for the next exercise.

## Exercise 4—DevStack installation

Before proceeding with the DevStack installation, you must install some packages required for proper installation.

1. At the PuTTY terminal window, use the following command sequence to update the system. At the password prompt, provide the password for the user you created during the installation. If prompted during the installation, press **Enter** to confirm:

```
hp@hpdevstack:~$ sudo apt-get -y update; sudo apt-get -u upgrade
[sudo] password for hp: hpinvent
```

2. Install additional packages using the following command:

```
hp@hpdevstack:~$ sudo apt-get install -y git curl python-dev libffi-dev
libssl-dev libxml2-dev build-essential libssl1.0-dev gettext screen python-
mysqldb memcached tgt genisoimage python-prettytable openvswitch-switch
```

3. When the installation has finished, enter the following command to pick up the current DevStack source from development hub:

```
git clone https://git.openstack.org/openstack-dev/devstack -b
stable/newton
```

4. The previous command generates a folder called “devstack.” Execute the following command to generate the “stack” account required by the DevStack environment:

```
hp@hpdevstack:~$ sudo ./devstack/tools/create-stack-user.sh
Creating a group called stack
Creating a user called stack
Giving stack user passwordless sudo privileges
```

5. Change your identity to the newly created “stack” account.

```
hp@hpdevstack:~$ sudo su - stack
stack@hpdevstack:~$
```

6. Again, make a copy of the DevStack development code using the `git` command.

```
git clone https://git.openstack.org/openstack-dev/devstack -b
stable/newton
```

7. Create the `local.conf` configuration file within the `devstack` folder, enter the folder, and execute the command to start the DevStack build process.

```
stack@hpdevstack:~/devstack$./stack.sh
```

---

**NOTE:** In the next exercise, the content of the `local.conf` file will be discussed.

---

8. Depending on the speed of your Internet connection and the CPU, the compilation might take more than one hour. At the end of the compilation process, you get a message similar to the following:

```
Horizon is now available at http://192.168.5.143/
Keystone is serving at http://192.168.5.143:5000/v2.0/
Examples on using novaclient command line is in exercise.sh
The default users are: admin and demo
The password: hpinvent
This is your host ip: 192.168.5.143
```

9. Open the web browser and navigate to the URL reported at the end of the build process, for example <http://192.168.5.143>. You should be able to log in to Horizon.
10. Close the web browser, but leave the terminal window open for the next exercise.

## Exercise 5—DevStack configuration file

The configuration file describing the features and functionality of the `stack.sh` script is called `local.conf` and is located in the `~/devstack/` folder. The file is a UNIX-formatted list of commands. Commands begin with the `#` character. What follows is the configuration file used for this training. The format of the file should be self-explanatory.

```
[[local|localrc]]
HOST_IP=192.168.5.4
SERVICE_HOST=$HOST_IP

IP_VERSION=4
SERVICE_IP_VERSION=4

ADMIN_PASSWORD=hpinvent
DATABASE_PASSWORD=$ADMIN_PASSWORD
SERVICE_TOKEN=$ADMIN_PASSWORD
SERVICE_PASSWORD=$ADMIN_PASSWORD
RABBIT_PASSWORD=$ADMIN_PASSWORD
MYSQL_PASSWORD=$ADMIN_PASSWORD

ceilometer Settings
enable_service ceilometer-acompute ceilometer-acentral ceilometer-anotification ceilometer-collector
enable_plugin ceilometer https://git.openstack.org/openstack/ceilometer.git
enable_plugin aodh https://git.openstack.org/openstack/aodh master
enable_service aodh-evaluator,aodh-notifier,aodh-api
disable_service ceilometer-alarm-notifier,ceilometer-alarm-evaluator
CEILOMETER_BACKEND=mysql

Cinder Settings
ENABLED_SERVICES+=,cinder,c-api,c-vol,c-sch,c-bak

Glance Settings

Images
Use this image when creating test instances
IMAGE_URLS+="http://download.cirros-cloud.net/0.3.2/cirros-0.3.2-x86_64-disk.img"
Use this image when working with Orchestration (Heat)
```

```

IMAGE_URLS+="http://download.fedoraproject.org//pub/fedora/linux/releases/25
/CloudImages/x86_64/images/Fedora-Cloud-Base-25-1.3.x86_64

Heat Settings
ENABLED_SERVICES+=,heat,h-api,h-api-cfn,h-api-cw,h-eng

Neutron settings
ENABLED_SERVICES+=,q-svc,q-agt,q-dhcp,q-l3,q-meta,neutron,q-metering,q-lbaas
#Q_ML2_TENANT_NETWORK_TYPE=vxlan
#Q_DVR_MODE=dvr_snat

Nova Settings
ENABLED_SERVICES+=,n-cauth,n-novnc
disable_service n-net

Swift Settings
ENABLED_SERVICES+=,s-proxy,s-object,s-container,s-account
SWIFT_REPLICAS=1
SWIFT_HASH=011688b4413657e209e

Tempest Settings
ENABLED_SERVICES+=,tempest
API_RATE_LIMIT=False
disable_service tempest

Trove Settings
ENABLED_SERVICES+=,trove,tr-api,tr-tmgr,tr-cond

Logging Info
LOGFILE=stack.sh.log
SCREEN_LOGDIR=$DEST/logs/screen
VERBOSE=True
LOG_COLOR=True

Miscellaneous
RECLONE=no
OFFLINE=true
GIT_BASE=${GIT_BASE:-http://git.openstack.org}

```

```
[[post-config|$NOVA_CONF]]
[DEFAULT]

#notification_driver=nova.openstack.common.notifier.rabbit_notifier,ceilometer.compute.nova_notifier
Ceilometer notification driver
instance_usage_audit=True
instance_usage_audit_period=hour
notify_on_state_change=vm_and_task_state
notification_driver=nova.openstack.common.notifier.rpc_notifier
notification_driver=ceilometer.compute.nova_notifier
```

---

---

# Fundamentals of OpenStack® Technology

## Answers to Lab Questions

### Lab 02—OpenStack® Management Tools

#### Exercise 1—Exploring OpenStack® CLI commands

7. Access the *OpenStack Command-Line Interface Reference* document and answer the following questions.
  - a. Q: With which command do the OpenStack® service-specific commands for the OpenStack® Identity service begin? keystone
  - b. Q: With which command do the OpenStack® service-specific commands for the OpenStack® Compute service begin? nova
  - c. Q: With which command do the OpenStack® service-specific commands for the OpenStack® Network service begin? neutron
8. Access the *OpenStack Command-Line Interface Reference* document again and answer the following questions.
  - a. Q: Regions are a relatively new feature of the Keystone Service. Is the command for creating a region available from the Identity Service (Keystone)? **No**
  - b. Q: What is the Keystone API version that provides the command for creating a region (hint: OpenStack® command-line client)? API v3
  - c. Q: Does the command for creating a region begin with keystone or openstack? openstack
  - d. Q: The API version 3-specific features of Keystone is a service-specific command or an OSC command? OSC
9. Q: In looking at the *OpenStack Command-Line Interface Reference* document, what is the syntax for displaying information about a particular command? -h or -help
15. d. Q: What version of Keystone API is being used? (hint: Check the documentation for current and supported Keystone API versions.) 3.0

## Exercise 2—Using a REST client browser plugin to generate API requests to OpenStack® services

### Generate a Keystone Security Token

12. Q: For how long is the token valid? 1 hour

### Using a REST client browser plugin to request a list of projects (tenants)

Now that you have a security token, you can use it to access other types of OpenStack® information, like a listing of the existing projects.

2. Q: What method is used to gather a list of tenants (projects)? GET
3. Q: What is the URI that is used? /v3/projects
4. Q: What is the name of the header parameter required to gather a list of tenants (projects)? X-AUTH-TOKEN
10. Q: What are the names of the tenants (projects) returned by the request? demo, alt\_demo, swiftprojecttest1, swiftprojecttest2, swiftprojecttest4, service, invisible\_to\_admin, admin

## Exercise 4—Managing OpenStack® from the Horizon GUI

11. Q: What is the URL of the page you were directed to? <http://docs.openstack.org/>

## Lab 03—OpenStack® Keystone (Identity)

### Exercise 1: Keystone Identity functionality

12. Q: API version is used by the Keystone in your environment? OS\_IDENTITY\_API\_VERSION=2.0
9. Q: Use the openstack project create --description "My first OpenStack Project" --enable project4labs command to create the project with the following attributes?
12. Q: What command did you use to create the above user? openstack user create --email user4labs@vlabs.net --project project4labs user4labs

### Exercise 2: Horizon (Dashboard) Identity functionality

7. Q: How do you know that the swiftusertest2 user is now a member of project4labs? User Swiftusertest2 now exists in the Project Members column.
13. Q: What is the primary project associated with user4labs? project4labs, which occurred when you created user4labs.
19. Q: What project are you automatically logged into, and why? project4labs because it is the primary project for the user4labs user.
24. Q: What projects are accessible by user4labs (hint: click the project drop-down list from the top the screen). Project4labs

## Lab 04—Glance (Image Repository)

### Exercise 1—Glance from the CLI

5. Q: Use the openstack image show <image id or image name> openstack command to show information about the following Glance image?
6. Q: Use the openstack image create image4labs --disk-format qcow2 --container-format bare --file ~/devstack/files/cirros-0.3.2-x86\_64-disk.img openstack command to create a Glance image with the following attributes from the image at ~/devstack/file/ cirros-0.3.2-x86\_64-disk.img

## Lab 05—OpenStack® Neutron (Networking)

### Exercise 1—Using the CLI to view the current network parameters

7. Q: What is the subnet range for the private network?  
10.0.0.0/24
8. Q: What is the default gateway for the private subnet?  
10.0.0.1
9. Q: What are the DNS servers configured for the private subnet?  
No DNS server is configured
11. Q: What is the MTU for the network **private**?  
1450
12. Q: What network virtualization technology is used for this private network (hint: check the type of the network)?  
vxlan
15. Q: What is the external IP address of the router?  
172.24.4.x
16. Q: Is the source network address translation enabled for this router?  
It is enabled since enable\_snat parameter has the value of true.

### Exercise 2—Using the Horizon UI to view the network parameters for the demo project

6. Q: Why is this network view different from the view in step 4 of this exercise?  
It is the only network, of those available, that is owned by the alt\_demo project.
8. Q: How many networks do you see after changing the project?  
Both public and private networks.
9. Q: What is the VxLAN ID of the provider network used for the **private** network? (Hint: check the segmentation ID parameter).  
The value will vary.

### Exercise 3—Using the Horizon UI to create and configure a simple flat network

#### Creating a network and a subnet

5. Q: What networks are currently available for **project4labs** to use? one:public

## Creating a router

6. Q: The internal interface has an IP address assigned. Where did it obtain the IP address from? The DHCP service assigned to the subnet the interface is connected to.

## Lab 06—OpenStack® Nova (Compute)

### Exercise 2—Create an instance from the Horizon GUI

#### Flavors

4. Q: Can the **project4labs** project use the **m1.tiny** flavor? Yes, because there are no projects selected for exclusive use of the flavor, it will be able to use the flavor.

#### Create an instance

23. Q: What is the IP address of the instance? 10.10.1.101
24. Q: Where was the instance IP address obtained from? Subnet4lab's DHCP server
26. Q: Why are you not able to ping the instance? You are pinging from the 192.168.5.0/24 external network and the instance is on 10.10.1.0/24 net4labs network.

### Exercise 3: Accessing an instance from an external network

#### Associating a Floating IP to an instance

4. Q: How many Floating IPs are currently available for allocation? 50
8. Q: What is the current status of the instance? Active
9. Q: What is the IP address of the assigned Floating IP? 172.24.4.x

### Exercise 4: Pausing and suspending the VM

#### Pause an instance

3. Q: Can you ping the Floating IP address of the instance? No

#### Suspend an instance

4. Q: Can you ping the Floating IP address of the instance? No

### Exercise 5: Creating a snapshot

4. Q: Does snapshotting an instance start a new instance?

No, a snapshot creates an image of the instance's image at the time the snapshot was taken. If you want to create an instance of the snapshot, you must create an instance that uses the snapshotted image.

## Lab 07—OpenStack® Cinder (Block Storage)

### Exercise 1—Creating a Cinder volume from the CLI

4. Q: Is this newly created volume bootable? (hint: look at the above screenshot) No

## **Lab 08—OpenStack® Object Storage (Swift)**

### **Exercise 2—Exploring the Swift configuration**

6. Q: How many containers exist? 1
7. Q: How many objects exist? 3

## **Lab 10—OpenStack® Ceilometer (Telemetry)**

### **Exercise 1—Ceilometer overview**

#### **Outputting Ceilometer data from the CLI**

4. Q: What is the time interval for samples in the ceilometer sample-list --meter cpu command? 10 minutes.