Linux Academy

Live! Lab

Hands-On Practice Exam

Contents

Lab Connection Information

- Labs may take up to five minutes to build
- Access to the Horizon Dashboard is provided on the Live! Lab page, along with your login credentials
- SSH information is provided on the Live! Lab page
- Labs will automatically end once the alloted amount of time finishes

Related Courses

OpenStack Foundation Certified OpenStack Administrator

Need Help?

Linux Academy Community

... and you can always send in a support ticket on our website to talk to an instructor!

Introduction

This hands-on practice exam walks through the process of setting up an instance with the appropriate security groups, routing, and configurations.

Log in to the Horizon Dashboard using the *admin* tenant, and ensure you have the *admin* project selected. Navigate to the **Access & Security** page, then click the **API Access** tab. **Download OpenStack RC File**.

SSH into the server through your terminal using the SSH Details provided on the Live! Lab page.

Copy the content of the file to the terminal instance, either by copying and pasting it into a new file called *admin.sh*, or using `scp` to copy it up to your server. Ensure the file is named *admin.sh*.

Source the file, entering your *admin* user's password when prompted:

```
root@ubuntu-openstack:~# source admin.sh
```

Create a New Project

Return to the dashboard. From the **Identity** menu on the left, go to **Projects**, **+Create Project**. We named our project *project1*. Add any description you desire, then navigate to the **Quota** tab.

Set the **VCPUs** to *10*, **Instances** to *8*, and **Volume Snapshots** to *8*. Hit **Create Project**.

We now need a new user to administrate this new project. Select the **Users** link, also under **Identity**. Press **+Create User**. We gave ours a name of *user1* and used *openstack* as our password. The **Primary Project** is *project1* and set the **Role** to *admin*. **Create User**.

Log out as the *admin* user from the web GUI, then log in as the *user1* user just created.

Using similar steps to those above, download and source your OpenStack RC file on the terminal. We used *user1.sh* as our script name:

```
root@ubuntu-openstack:~# source user1.sh
```

Edit the Image Service Policy

We want only the users with the admin role to delete images from the image service. We need to edit the *policy.json* file to set this. The file is located at `/etc/glance/policy.json`; use your preferred text editor to open the file.

Look for the `delete_image` option, and add `role:admin` in between the quotes to the right:

```
"delete_image": "role:admin",
```

In actual environments it would be necessary to restart the service; due to the nature of this lab environment, we are not.

Tenant Network

Back in the dashboard, select the **Network** dropdown, and click on **Networks**. Press **+Create Network**. We named ours *proj1net* (for "project 1 network"). **Next**.

The subnet needs to be created. We named ours **practice**, and set the **Network Address** to *10.0.2.0/24*. Press **Next** then **Create**.

Routing

A router also needs to be set up so the server can access the Internet. Move to the **Routers** page, then press **+Create Router**. We named ours *router2*, left the **Admin State** as *UP*, and set the external network to *public*. **Create Router**.

Click on the name of the router to view its details. Move to the **Interfaces** tab. We need to associate this router to your private network. Click **+Add Interface**. Select the newly-created *proj1net*, then **Add Interface**.

If you now select **Network Topology** from the left-hand menu, you can view how the various parts of the network relate to one another.

Create a Custom Security Group

Return to the **Compute** dropdown and select **Access & Security**, then click the **Security Groups** tab. Press **+Create Security Group**.

We called ours *proj1_SG*. Press **Create Security Group**.

On the right of the newly-created group, click **Manage Rules**; we want to be able to ping our instances. Click **+Add Rule**.

For **Rule**, select *All ICMP*. The rest can be left with the default values. **Add**.

We also want to be able to SSH into the servers. Click **+Add Rule** again. Set the **Rule** to *SSH*, and again leave the rest with default values. Press **Add**.

Launch an Instance

Under **Compute**, select **Instances**, then click **Launch Instance**. For **Instance Name**, we choose *instance1*. We set the **Flavor** to *m1.nano*, and the **Instance Count** to *1*. The **Boot Source** is *Boot from Image*, and, finally, choose the *cirros* image under **Image Name**. Move to the **Access & Security** tab and check off both security groups.

We now need to move to the **Networking** tab and confirm that the **Selected networks** dropdown is set to our *proj1net* network. **Launch**.

Once running, use the dropdown on the far right of the new image row to select **Associate Floating IP**. Press the **+** sign next to the **IP Address** option, then press **Allocate IP**, followed by **Associate**. Now click on the instance's name, and select the **Console** tab. Click on the gray bar to trigger the terminal, in needed. For the instance login use *cirros*; for the password input *cubswin:)*. Press enter. You should be dropped into the terminal.

Return to the **Instances** main screen and copy the public IP. Go back to your original terminal, and `ping` your server at the copied IP to ensure the instance is working.

The lab is now complete.