

Ubuntu Openstack Installer, Single Machine Mode

A Dell and Canonical Technical White Paper

Kent Baxley

Canonical Field Engineer



THIS WHITE PAPER IS FOR INFORMATIONAL PURPOSES ONLY, AND MAY CONTAIN TYPOGRAPHICAL ERRORS AND TECHNICAL INACCURACIES. THE CONTENT IS PROVIDED AS IS, WITHOUT EXPRESS OR IMPLIED WARRANTIES OF ANY KIND.

© 2014 Dell Inc. All rights reserved. Reproduction of this material in any manner whatsoever without the express written permission of Dell Inc. is strictly forbidden. For more information, contact Dell.

August 2014

Contents

Introduction.....	4
Introducing the Ubuntu OpenStack Installer	4
What’s different about this installer?.....	4
Getting Started	4
What’s happening under the hood?	6
Adding Resources.....	12
Uninstalling the Cloud.....	12
Contributing, Getting Help, and Learning More.....	13

Introduction

You've read about using [Juju and MAAS on Dell PowerEdge Servers](#), you've also learned about using [LXC](#), the [Juju Local Provider](#) and how to leverage LXC and KVMs using Juju on a single machine.

Now, let's have a look at how these technologies can help you install an OpenStack cloud on one machine in just a few simple steps.

Introducing the Ubuntu OpenStack Installer

The Ubuntu OpenStack Installer is designed to help take the headaches out of bringing up a cloud. The installer works with a single machine or multiple machines. This whitepaper will focus on deploying a cloud on a single system.

What's different about this installer?

There are many other single-node solutions for deploying OpenStack, with devstack being one of the more popular tools.

The Ubuntu OpenStack installer is different from devstack in that harnesses the power of Juju to deploy the various services onto a single machine using a combination of KVM and LXC containers. For those that don't have a lot of spare systems lying around, yet wish to see or demonstrate how easy it is to deploy a working cloud using Juju, the Ubuntu OpenStack installer is an ideal option. The finished product is a running OpenStack environment with a pre-configured Neutron network, pre-synced images, and ready to deploy instances via Nova-Compute.

Getting Started

For a single-machine installation, the following system specifications are recommended:

- CPU that supports VT-x/AMD-V. Quad-core or better is recommended.
- At least 12GB of memory.
- 100GB+ Hard Drive (a Solid State Drive is recommended for best results).
- Juju version 1.18.3 or higher
- Ubuntu 14.04 LTS, 64-bit version
- Internet access (for downloading packages and images)

Since the single-machine scenario is generally used in development or demo environments, a Dell Precision T7610 with Ubuntu 14.04 LTS desktop, 12GB of memory, 250GB of disk space, and a ten-core Intel Xeon CPU was used for this write up. Any Dell system, however, that meets at least the recommended requirements above can be used. This includes laptops, workstations, or servers.

Once the operating system is installed, some extra software repositories are required to install the cloud-installer as well as the latest stable juju release:

```
$ sudo apt-add-repository ppa:cloud-installer/ppa
```

```
$ sudo apt-add-repository ppa:juju/stable
$ sudo apt-get update
```

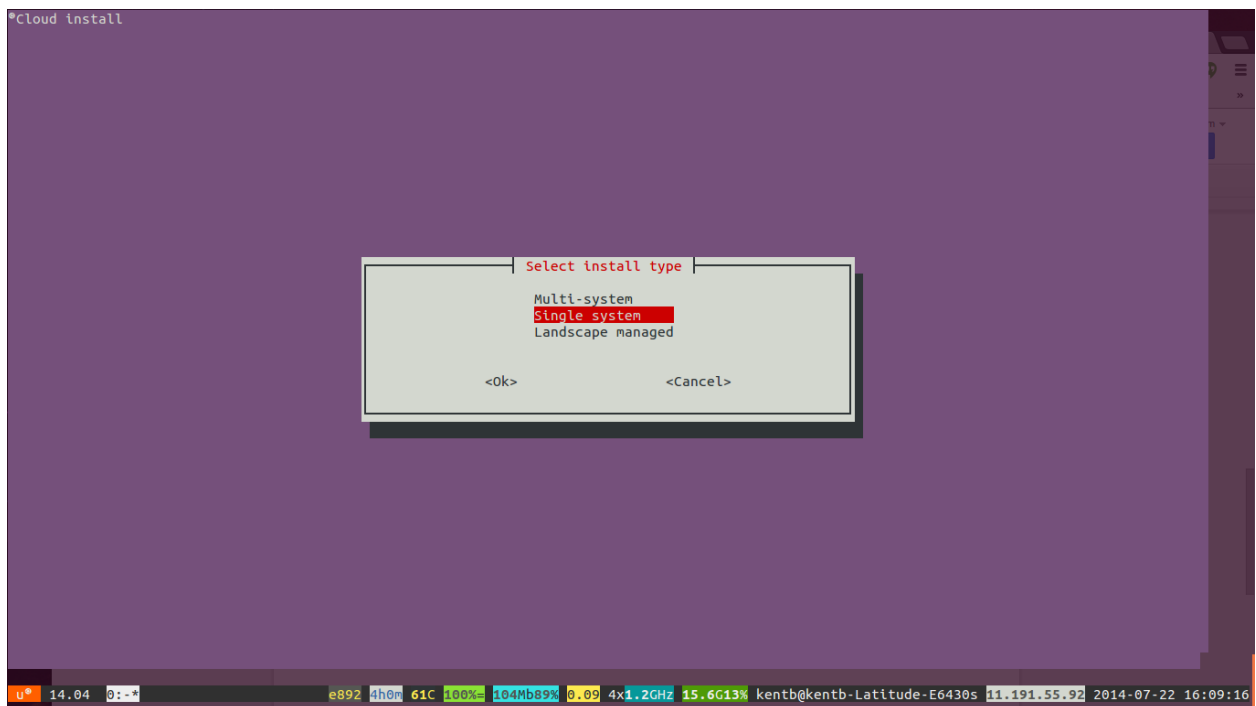
From there, install the packages:

```
$ sudo apt-get install cloud-installer
```

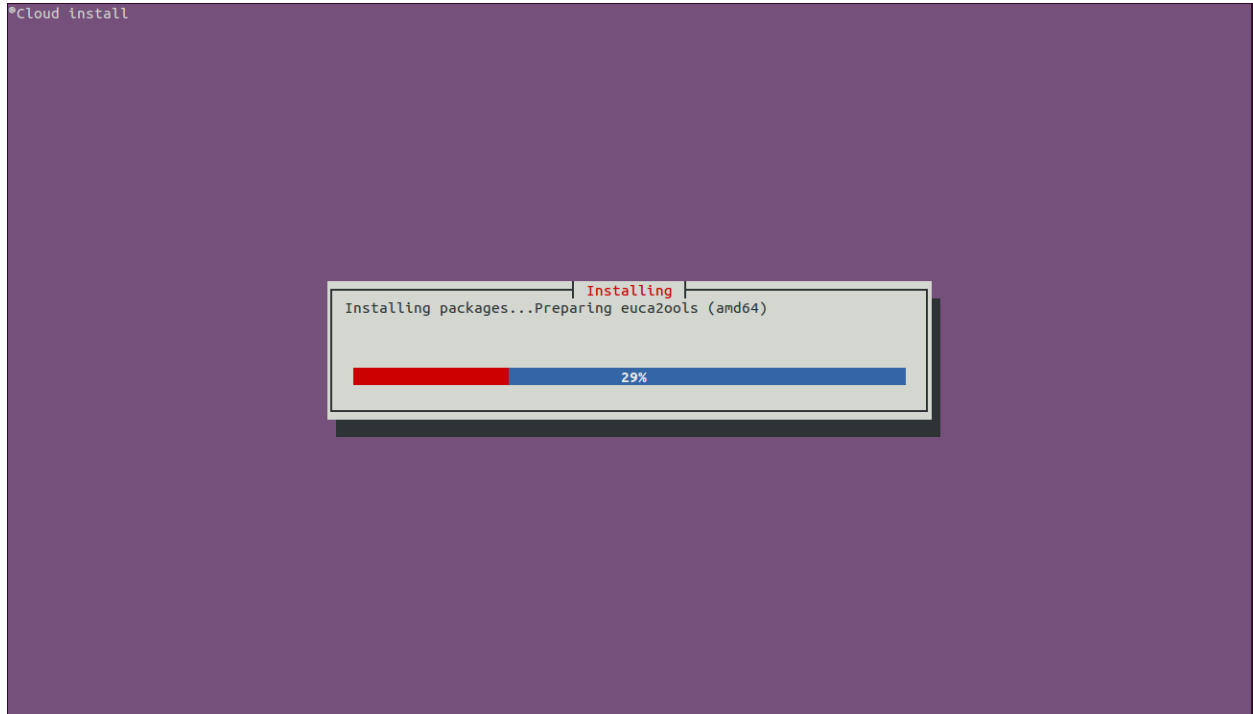
Then, kick off the installer:

```
$ sudo cloud-install
```

The first screen will prompt for an install type, choose "Single System":



Next, when prompted, enter an OpenStack administrative password for your cloud. The system will then proceed to install any extra packages, bootstrap the Juju environment, and begin the process of bringing up machines and services:

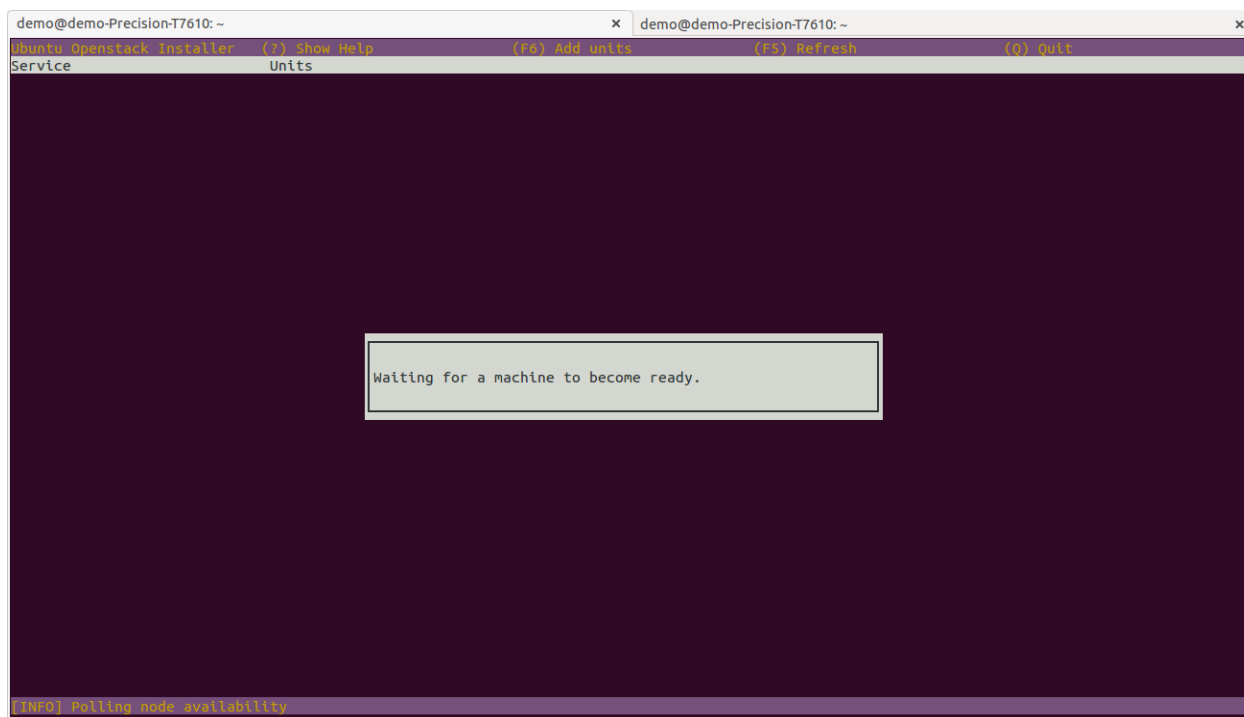


What's happening under the hood?

The cloud-installer installs juju-core and creates a Juju local environment on the standalone system, as seen below from `~/juju/environments.yaml`:

```
default: local
environments:
  local:
    type: local
    container: kvm
    lxc-clone: true
    admin-secret: admin
EOF
```

Juju bootstraps the local environment and a 14.04 LTS image, needed for the containers, is downloaded. The image download may take a while to complete, based on the internet connection speed. The following is displayed in the console while the installer waits for a system to come available:



Once the image is downloaded, the first system to be brought up is a KVM container, which acts as the OpenStack controller node. Once this node is up it will proceed to deploy the following services inside of it in individual LXC containers:

- glance
- juju-gui
- keystone
- mysql
- nova-cloud-controller
- glance-simplestreams-sync
- openstack-dashboard
- rabbitmq-server

Two other KVM containers are also spun up by Juju. One will house a nova-compute node and the other the quantum-gateway node. The console window will also change to show the status of the individual services being deployed:

```

demo@demo-Precision-T7610: ~
Ubuntu Openstack Installer (?) Show Help (F6) Add units (F5) Refresh (Q) Quit
Service Units
glance glance/0 (pending)

keystone keystone/0 (pending)

mysql mysql/0 (pending)

nova-cloud-controller nova-cloud-controller/0 (pending)

openstack-dashboard openstack-dashboard/0 (pending)

quantum-gateway quantum-gateway/0 (pending)
address: 10.0.3.105

rabbitmq-server rabbitmq-server/0 (pending)

nova-compute nova-compute/0 (pending)
address: 10.0.3.244

glance-simplestreams-sync glance-simplestreams-sync/0 (pending)

juju-gui juju-gui/0 (pending)

[INFO] Nodes are still deploying Horizon: Pending Juju-GUI: Pending

```

More details on the status of the deploying services can be viewed in a separate console window by running the 'juju status' command.

Again, depending on the internet connection speed, it may take a while for the charms to download and for the services to deploy and start up in their respective containers.

As each service comes online, the status in the console UI window will change from 'pending' to 'installed' to 'started'. Each service will also get its own IP address assigned to it:


```
demo@demo-Precision-T7610: ~
Ubuntu Openstack Installer (?) Show Help (F6) Add units (F5) Refresh (Q) Quit
Service Units
keystone keystone/0 (started)
address: 10.0.3.124

mysql mysql/0 (started)
address: 10.0.3.79

nova-cloud-controller nova-cloud-controller/0 (started)
address: 10.0.3.237

openstack-dashboard openstack-dashboard/0 (started)
address: 10.0.3.233

quantum-gateway quantum-gateway/0 (started)
address: 10.0.3.105

rabbitmq-server rabbitmq-server/0 (started)
address: 10.0.3.75

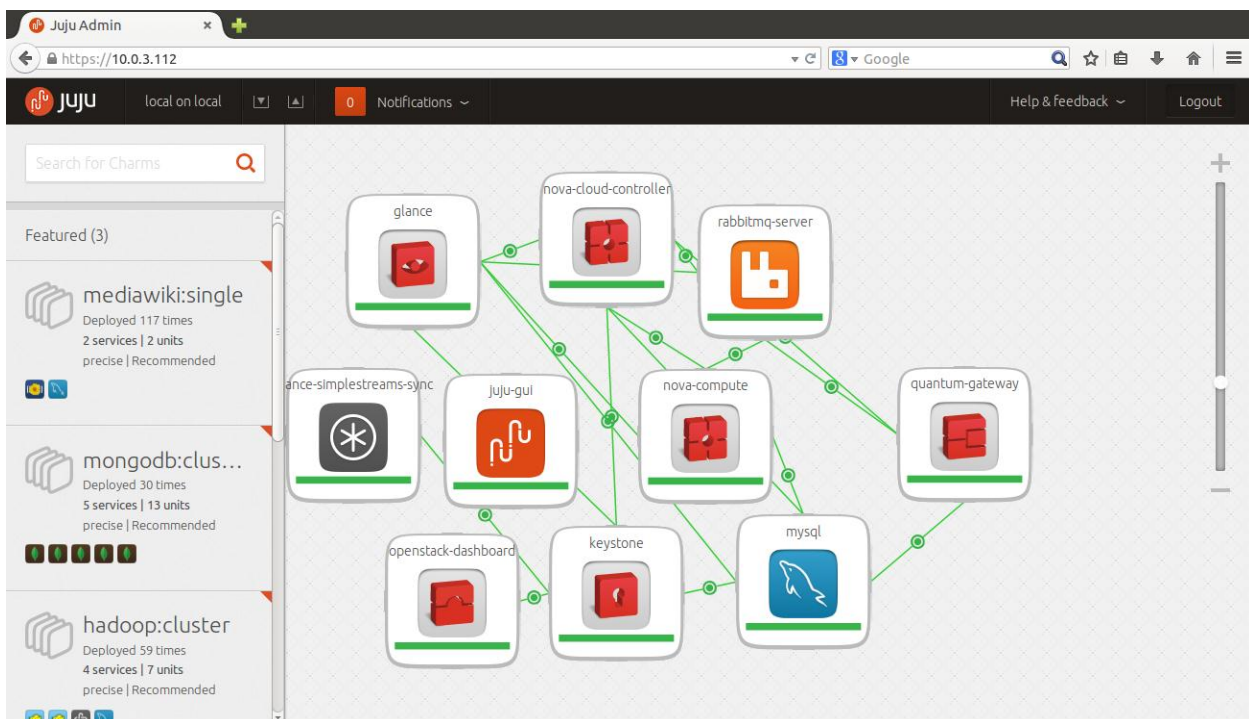
nova-compute nova-compute/0 (started)
address: 10.0.3.244

glance-simplereams-sync glance-simplereams-sync/0 (started)
address: 10.0.3.48

juju-gui juju-gui/0 (started)
address: 10.0.3.112

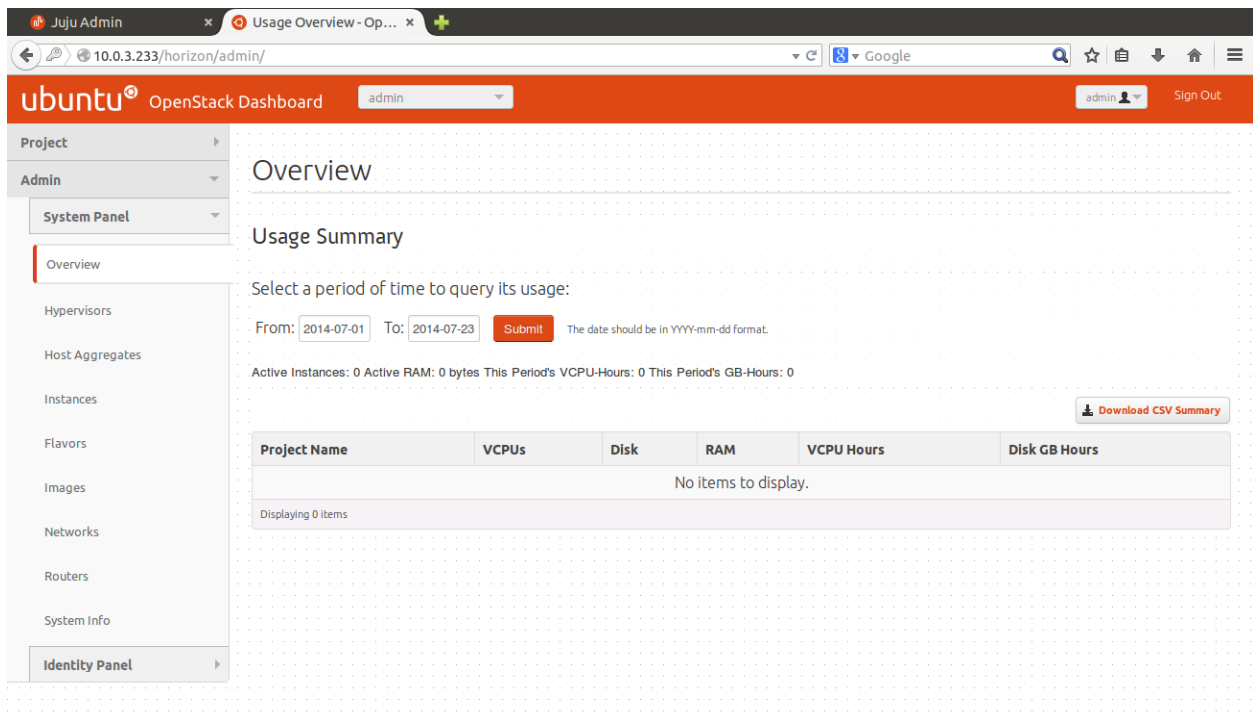
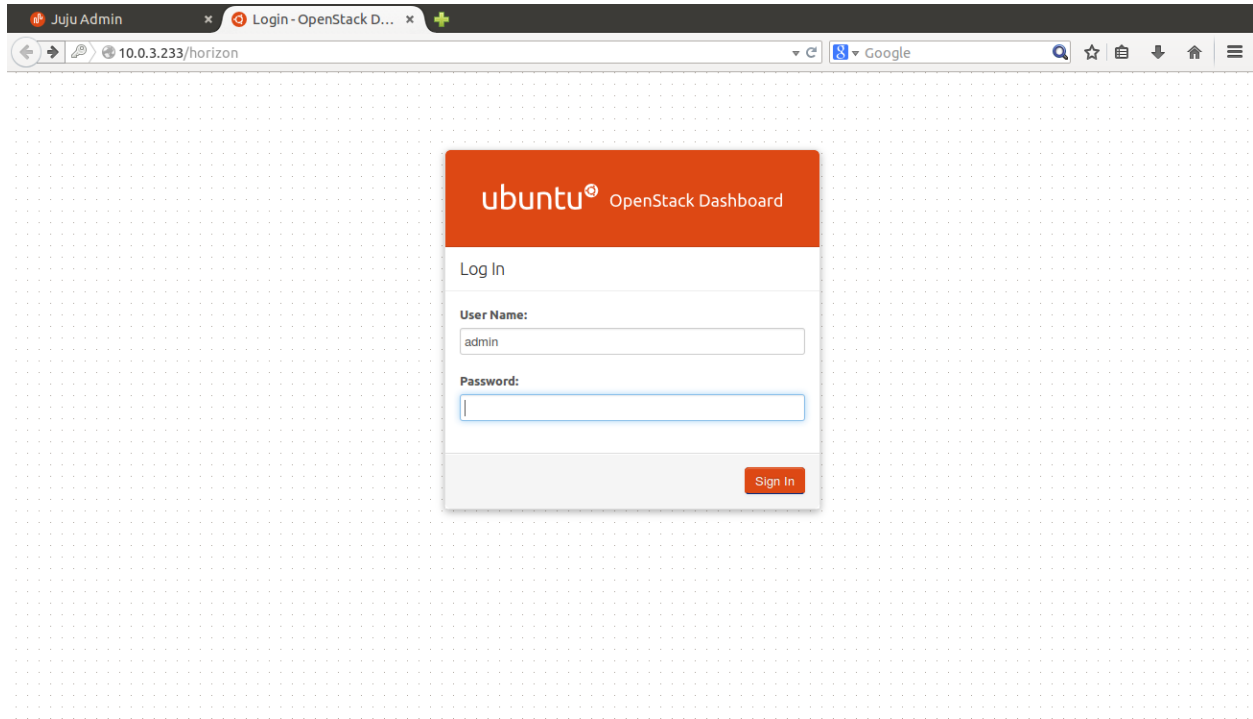
[INFO] Nodes are accessible Horizon: http://10.0.3.233/horizon Juju-GUI: http://10.0.3.112/
```

Once the Juju-GUI service is in a 'started' state, the graphical status of the OpenStack deployment can be viewed by navigating to Juju-GUI URL in a browser. The cloud-installer's console UI window above provides a URL to the Juju-GUI along the bottom of the window for easy access:



When all of the services are started, the OpenStack cloud is ready to go!

It is now possible to navigate to the Horizon dashboard (the URL is also provided in the console status UI window along the bottom) and log in with the administrative credentials entered at the beginning of the installation:



Navigating to the Networks tab shows that an external (ext-net) and internal (ubuntu-net) has been created already:

The screenshot shows the OpenStack Horizon interface for the 'admin' project. The 'Networks' tab is selected in the left sidebar. The main content area displays a table of existing networks. Two networks are listed: 'ubuntu-net' (internal) and 'ext-net' (external). Both are in an 'ACTIVE' state with an 'UP' admin state. The 'ext-net' is shared. Buttons for '+ Create Network' and 'Delete Networks' are visible at the top right of the table. A footer message from Firefox is also present.

<input type="checkbox"/>	Project	Network Name	Subnets Associated	Shared	Status	Admin State	Actions
<input type="checkbox"/>	ubuntu	ubuntu-net	ubuntu-subnet 10.0.4.0/24	No	ACTIVE	UP	Edit Network More
<input type="checkbox"/>	admin	ext-net	ext-subnet 10.0.3.0/24	Yes	ACTIVE	UP	Edit Network More

Displaying 2 items

The images tab reveals that the some images have been pre-populated into this cloud by the installer tool:

The screenshot shows the OpenStack Horizon interface for the 'admin' project. The 'Images' tab is selected in the left sidebar. The main content area displays a table of existing images. Two images are listed: 'ubuntu-trusty-14.04-i386-server-20140607.1-disk1.img' and 'ubuntu-raring-13.04-amd64-server-20140111-disk1.img'. Both are in an 'Active' state. The format for both is 'QCOW2'. Buttons for '+ Create Image' and 'Delete Images' are visible at the top right of the table. A footer message from Firefox is also present.

<input type="checkbox"/>	Image Name	Type	Status	Public	Protected	Format	Actions
<input type="checkbox"/>	ubuntu-trusty-14.04-i386-server-20140607.1-disk1.img	Image	Active	Yes	No	QCOW2	Edit More
<input type="checkbox"/>	ubuntu-raring-13.04-amd64-server-20140111-disk1.img	Image	Active	Yes	No	QCOW2	Edit More

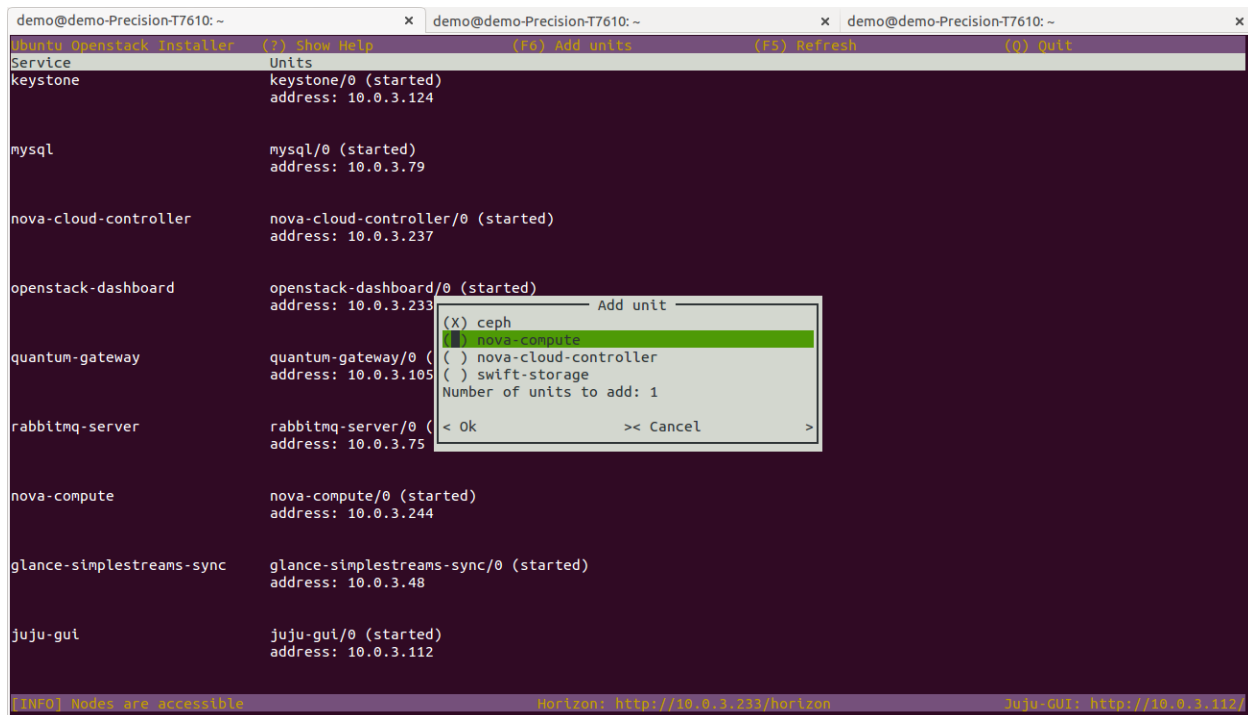
Displaying 2 items

Pretty much all that needs to be done next is fire up some instances and have some fun with your new OpenStack cloud!

Adding Resources

While the default setup is generally good enough for experimenting with OpenStack, the Ubuntu OpenStack installer allows users the opportunity to grow their clouds.

Adding new resources to the cloud is as easy as pressing the F6 key from the console ui window. Options include ceph, swift-storage, or additional nova-cloud-controller and nova-compute nodes. The number of units to add is also configurable:



Warning: Be mindful of system resources when adding units! Since all of this is being done on a single machine, make sure the system has the capacity (i.e. memory and disk space) to take on additional Virtual Machines. Swift Storage alone, for example, will utilize 3 VMs. cloud-installer does not provide any policing of system resources when using this feature.

Uninstalling the Cloud

To tear down and uninstall the cloud, simply run:

```
$ sudo cloud-install -u
```

This will destroy the juju environment and uninstall all related packages (including cloud-installer and juju-core). The cloud-installer package will need to be reinstalled via apt-get for any new installations.

Contributing, Getting Help, and Learning More

Documentation on the cloud-installer can be located [here](#).

For those interested in contributing to the cloud-installer, all of the code is available on [github](#).

Testing, filing bugs, and contributing fixes or enhancements are always welcome.

For additional help that can't be found in the documentation, or for any other questions, please feel free to join the #ubuntu-solutions IRC channel on freenode.net.