

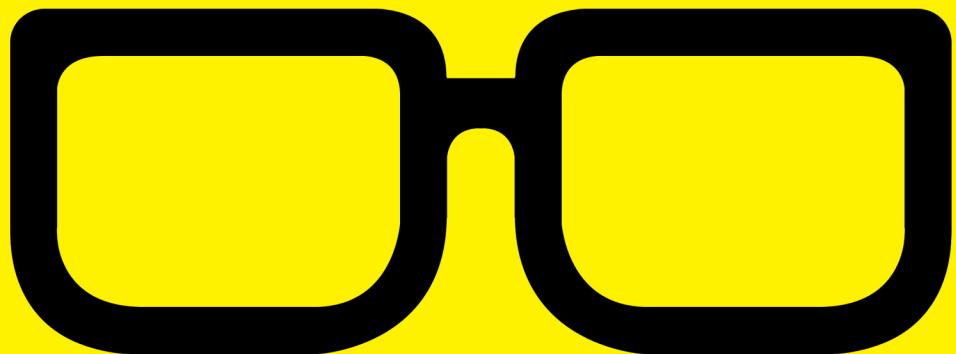
SPONSORED BY



SERVERSDIRECT.



GEEK GUIDE



Build a
Private Cloud
for Less Than
\$10,000!

Table of Contents

Goals	5
Introduction	6
OpenStack Architecture	11
Identity Service (Keystone)	12
Messaging Service.....	13
Image Service (Glance)	13
Dashboard (Horizon)	13
Compute (Nova)	13
Hardware Considerations.....	14
The Installation Process	15
Operation	16
Resources	23

MIKE DIEHL has been using Linux since the days when Slackware came on 14 5.25" floppy disks and installed kernel version 0.83. He has built and managed several servers configured with either hardware or software RAID storage under Linux, and he has hands-on experience with both the VMware and KVM virtual machine architectures. Mike has written numerous articles for *Linux Journal* on a broad range of subjects, and he has a Bachelor's degree in Mathematics with a minor in Computer Science. He lives in Blythewood, South Carolina, with his wife and four sons.

GEEK GUIDES:

Mission-critical information for the most technical people on the planet.

Copyright Statement

© 2014 *Linux Journal*. All rights reserved.

This site/publication contains materials that have been created, developed or commissioned by, and published with the permission of, *Linux Journal* (the “Materials”), and this site and any such Materials are protected by international copyright and trademark laws.

THE MATERIALS ARE PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. The Materials are subject to change without notice and do not represent a commitment on the part of *Linux Journal* or its Web site sponsors. In no event shall *Linux Journal* or its sponsors be held liable for technical or editorial errors or omissions contained in the Materials, including without limitation, for any direct, indirect, incidental, special, exemplary or consequential damages whatsoever resulting from the use of any information contained in the Materials.

No part of the Materials (including but not limited to the text, images, audio and/or video) may be copied, reproduced, republished, uploaded, posted, transmitted or distributed in any way, in whole or in part, except as permitted under Sections 107 & 108 of the 1976 United States Copyright Act, without the express written consent of the publisher. One copy may be downloaded for your personal, noncommercial use on a single computer. In connection with such use, you may not modify or obscure any copyright or other proprietary notice.

The Materials may contain trademarks, services marks and logos that are the property of third parties. You are not permitted to use these trademarks, services marks or logos without prior written consent of such third parties.

Linux Journal and the *Linux Journal* logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners. If you have any questions about these terms, or if you would like information about licensing materials from *Linux Journal*, please contact us via e-mail at info@linuxjournal.com.

About the Sponsors



Servers Direct is the premier provider of advanced, custom-configured server and storage solutions. We're a privately owned company with more than 100 technology experts on staff who are committed to delivering systems designed around your unique needs. Those solutions are produced in our state-of-the-art server build facility with ISO and UL certifications and more than 100,000 square feet of manufacturing space. With nearly 25 years of design, manufacturing and logistics experience, Servers Direct can be counted on as your trusted technology partner both today and in the future.

1-800-576-7931

<http://www.serversdirect.com>



Seagate is a world leader in hard disk drives and storage solutions. Learn more at <http://www.seagate.com>.

Seagate and Seagate Technology are registered trademarks of Seagate Technology LLC.

Build a Private Cloud for Less Than \$10,000!

MIKE DIEHL

Goals

This eBook presents a compelling argument as to why you should consider re-architecting your enterprise toward a private cloud. It outlines some of the design considerations that you need to be aware of before implementing your own private cloud, and it describes using the DevCloud installer in order to install OpenStack on an Ubuntu 14 server. Finally, this eBook will familiarize you with the features and day-to-day operations of an OpenStack-based private cloud architecture.

Introduction

Cloud computing is a new technology that's designed to make a system administrator's job easier. However, cloud computing comes in two varieties, public and private, and each is aimed at solving different problems. Public cloud computing, such as the offerings from Amazon and Rackspace, is designed to allow system administrators to scale their computing resources rapidly by bringing resources into existence upon demand. Underutilized resources can be retired with the knowledge that they can be brought back into production if needed at a later date. This allows administrators to manage (and pay for) only those resources that are being used.

Private cloud computing uses the same technology as public cloud computing, but tackles different issues. By centralizing an enterprise's file storage system, system administrators are able to use the storage more efficiently, and perhaps even put off having to upgrade. For example, it's entirely plausible to have one server using 75% of its disk space, while another server is using only 25% of its space. Virtualizing the servers and storage allows system administrators to allocate exactly the amount of storage that each server needs, regardless of how much capacity each spindle provides.

Historically, when system administrators needed more server or storage capacity, they had to build and install a new server. Then, they had to configure and test it before putting it into production. By using virtual machines, perhaps running on a blade server, this process is reduced to (perhaps?) adding a CPU, cloning

An enterprise's servers may need to reside within the enterprise whether they are real servers or virtual servers in a cloud environment.

an existing server image and starting the new server—no screwdriver required.

Many factors may make public cloud computing inappropriate in a given organization. Some organizations simply can't have their private data exposed to the public Internet. Public cloud servers can be protected by firewalls and VPNs, but a local LAN that is entirely under an organization's control is easier to protect. Financial and health information, as well as intellectual property, probably shouldn't be processed on a public cloud server. Keeping these functions hosted within the organization's private cloud infrastructure mitigates these difficulties.

This idea of maintaining control really isn't a new concept, since most enterprises maintain their own server infrastructure on-site already. The private cloud architecture is simply a variation on a theme. An enterprise's servers may need to reside within the enterprise whether they are real servers or virtual servers in a cloud environment.

Many IT departments maintain a separate development, quality and production server for every compute service they provide. Ostensibly, this is meant to ensure that changes and upgrades can be performed with as little risk

of failure as possible. However, this also necessitates strict change management protocols in order to ensure that the development and quality environments mimic the actual production environment. In a virtualized environment, the production environment can be cloned, fed a copy of production data and tested in parallel. There is no need to maintain separate resources, as the entire test environment can be implemented using spare processor capacity on production hardware.

Because all of the processing power is virtualized, IT services can migrate from one physical hardware device to another with little or no downtime. Obviously, in the event of hardware failure, this is a good thing. But consider that this also opens up the possibility of being able to perform rolling upgrades during prime time instead of having to cut over to a new server after hours.

So how does it work? Conceptually, every server and compute service that an IT organization provides is executed within an isolated container or virtual machine. These containers, or virtual machines, are completely self-contained and isolated from each other. Each virtual machine is allocated as much CPU and storage as it needs—and no more. These virtual machines are isolated so that they function as individual servers, even though they reside on one or more much more powerful servers. The end result is CPU and storage resources that aren't used by one virtual machine are available for use by other virtual machines. This resource sharing is what makes cloud computing so economically attractive.

The OpenStack system was developed by Rackspace as a

OpenStack takes advantage of various Linux features, such as CPU virtualization, process containers, software-defined networking and very flexible filesystem support.

means of managing a very large number of virtual machines running on several real servers. OpenStack takes advantage of various Linux features, such as CPU virtualization, process containers, software-defined networking and very flexible filesystem support.

CPU virtualization enables a host server to allow a guest server to execute as no more than a simple process. From the guest server's point of view, the server has complete access to the CPU, dedicated file storage and direct network access. From the host server's point of view, there is a process running that is isolated from other processes. Any time that the guest server attempts direct access to hardware, that access is emulated by the host server. When a guest server is idle, the resources that it normally would consume are made available, by the host operating system, to other guest servers. Because of this emulation, the guest never knows that it's merely a guest. Because of hardware-imposed isolation, the host server can host many guest servers without any of them being able to interfere with each other. Hardware emulation is managed by a hardware feature known as a hypervisor. OpenStack supports both the KVM and QEMU hypervisors, among others, under Linux.

The important thing to remember is that a Linux container is a low-overhead alternative to the virtual machine capability.

Linux process containers simply are a means of isolating server processes without incurring the overhead of a full-up virtual machine. The actual implementation entails putting process IDs, network interfaces and file handles into separate namespaces, much like namespaces work in XML or structured programming. The important thing to remember is that a Linux container is a low-overhead alternative to the virtual machine capability.

Software-defined networking allows the various virtual machines to be connected to each other on an entirely virtualized network. A software-defined network is very similar to any other network. It's still composed of routers, bridges and nodes. The only difference is that a software-defined network doesn't have actual routers, bridges and nodes; it's implemented in terms of host-based firewall rules, bridging and VLANs. The host server actually provides the network isolation that a real router normally would be expected to provide.

The filesystem capabilities that Linux supports are extremely flexible. Filesystems can be built that span multiple terabytes. They can be distributed across multiple machines for redundancy, and RAID arrays can be built using any combination of striping and mirroring.

Furthermore, system administrators can mix and match IDE, SATA, SCSI, iSCSI and ATAoE drives, depending on what makes sense in their environment.

However, some design considerations need to be dealt with before you can begin implementation. The obvious among these is memory and processor capacity. Because of the resource efficiencies offered by cloud computing, the temptation is to buy just as much as you think you need. Avoid this temptation; enterprises almost always grow. They never shrink, so the system that is adequate today may not be sufficient next year. When making a commitment like this, you need to be thinking about the long-term payoff.

Another thing to consider before you deploy a private cloud is whether you have sufficient bandwidth between your compute and storage nodes. If all of these nodes share a chassis, this obviously isn't a problem. On the other hand, you may find that the cloud computing architecture concentrates more traffic onto a smaller segment of your network. You might find it necessary to create dedicated links between your cloud chassis. Placing your cloud infrastructure closer to your network core is also worth considering.

OpenStack Architecture

OpenStack is composed of a number of other software packages configured to cooperate with each other. Some of these packages can be installed and configured onto other servers to enhance the overall system's scalability and reliability.

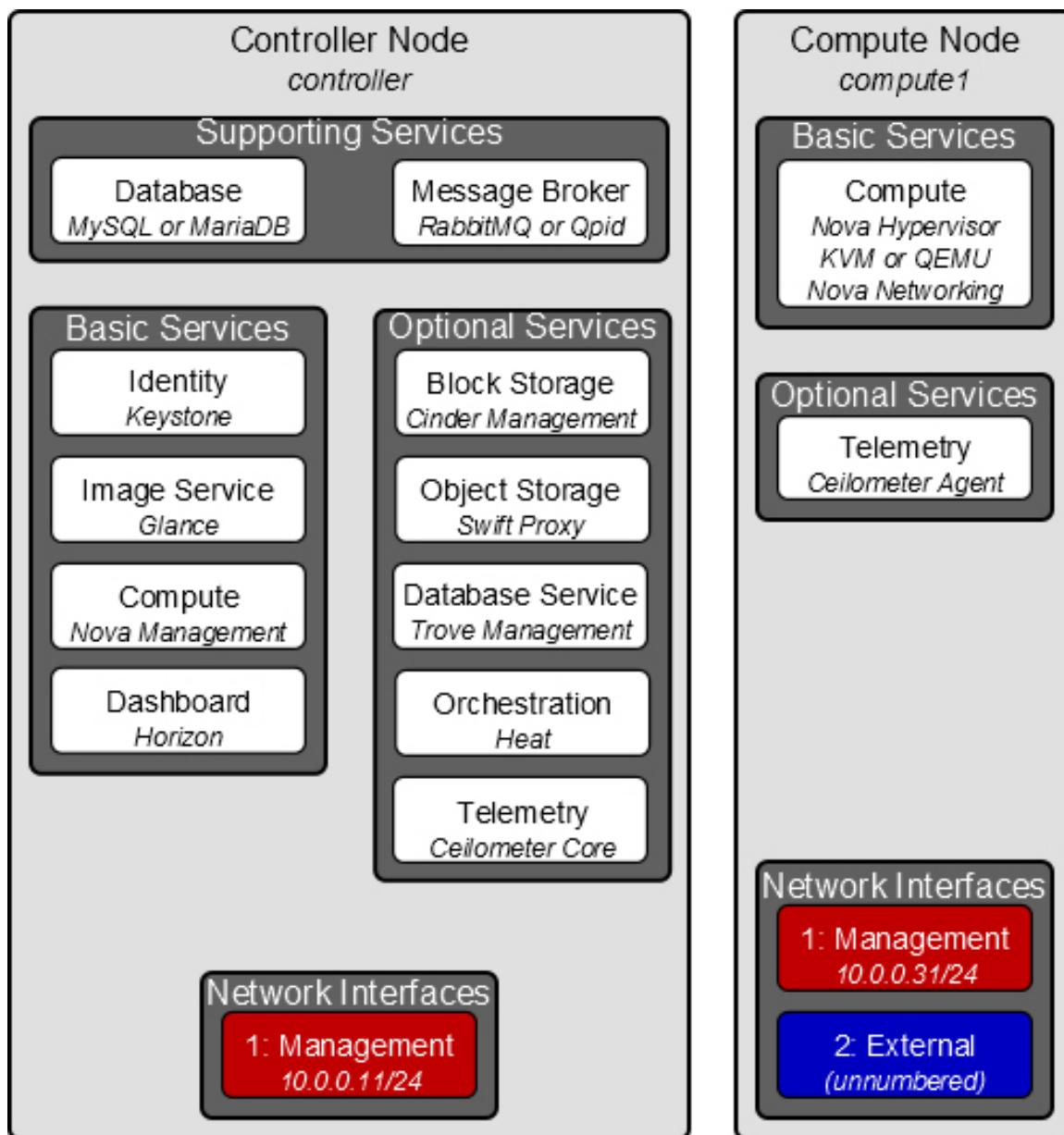


FIGURE 1. OpenStack Architecture Diagram

(image from http://docs.openstack.org/icehouse/install-guide/install/apt/content/ch_overview.html#architecture_example-architectures)

Identity Service (Keystone): The Keystone identity service provides authentication and authorization for users, servers and services. Most of the other software packages

will be configured with credentials to allow them to authenticate with the Keystone server. This server in turn, relies upon a MySQL database for its storage needs.

Messaging Service: All of the various services communicate with each other via the messaging service. OpenStack supports the RabbitMQ, Qpid and ZeroMQ messaging services. The installation process outlined in this eBook installs and configures the RabbitMQ messaging service. In general, a messaging service allows software modules to “publish” messages, while other software modules “subscribe” to messages. In this way, information is transmitted efficiently, where it needs to be.

Image Service (Glance): The Glance image service manages the access and distribution of virtual machine images. Glance prevents users from accessing images they aren’t authorized to access and ensures that images are available to every compute node, so that virtual machines can be launched on any compute node at any time.

Dashboard (Horizon): Most system administrators’ day-to-day interactions with OpenStack will be through the Web-based dashboard provided by Horizon.

Compute (Nova): The Nova service is responsible for managing the entire lifecycle of a virtual server. This includes everything from starting up a new virtual server, creating a VNC session for it, and shutting it down when it is no longer needed. Note that Nova also manages the legacy networking capability, but this function is being phased out in favor of the newer Neutron networking service. Nova is the service that system administrators will be interacting with most frequently for functions that can’t be performed from the dashboard.

OpenStack allows system administrators to mix and match everything from a Servers Direct SD-6017R-TDF 1U Rackmount Server with two E5-2600 processors to a Servers Direct SD-5037MC-H8TRF 3U monster with eight blades and one E3-1200 processor on each blade.

Hardware Considerations

One of the goals of server virtualization is chassis-count reduction—that is, consolidating as many real servers into a smaller number of chassis without sacrificing function, manageability or performance.

Initially, an enterprise could start out by deploying on one or more 1U servers. As the process of assimilating legacy servers into the private cloud architecture progresses, larger servers may be justified. OpenStack allows system administrators to mix and match everything from a Servers Direct SD-6017R-TDF 1U Rackmount Server with two E5-2600 processors to a Servers Direct SD-5037MC-H8TRF 3U monster with eight blades and one E3-1200 processor on each blade. These servers are available for sale today at <http://www.serversdirect.com>. As your needs change, your entire architecture can change with them.

No matter how you choose to build your enterprise, it will take reliable disk storage to make it work. You will be putting much of your entire enterprise on just a few

spindles, so this isn't the time to go bargain shopping for disk space. Keeping that in mind, you seriously should consider buying more than one drive at a time with the idea that you will be pairing them up in a RAID configuration in one form or another. Given the price of disk space these days and the nature of what you're trying to accomplish, it really doesn't make sense to buy anything smaller than 500GB, and 1TB as a minimum purchase size might actually make sense by the time you read this. Larger drives mean fewer moving parts and less heat in your server chassis. (You also might consider buying one for the shelf, just in case.) On the 1TB side, you should consider the Seagate Constellation ES.3 1TB SATA drive. At the other end of the scale is the Seagate ST6000NM0024 SATA drive with a whopping 6TB of storage on one spindle. Two of these in a RAID mirror would provide 6T of bullet-proof storage, more than most enterprises ever will need.

The Installation Process

Let's build an example cloud computing environment using OpenStack running on Ubuntu Server 14.04 LTS. Because configuring OpenStack from scratch can be quite painful, let's use DevStack to install and configure OpenStack automatically. This will result in a working single-node stack server. You will be able to use this configuration to become familiar with OpenStack operations.

The first step is to install Ubuntu Server onto the hardware. This installation should be fairly straightforward, so I don't discuss the details here.

DevStack is installed via git, so you need to install git

before you can proceed:

```
sudo apt-get install git -y
```

Once git is installed, you can start the DevStack installation:

```
git clone https://github.com/openstack-dev/devstack.git  
cd devstack
```

You can run DevStack without any pre-configuration. In this case, you will be prompted for the various passwords that the system requires, and the installation script will attempt to create a rational default configuration. Alternatively, the installation script can be provided with a configuration file that will allow a system administrator to customize the installation.

There is a sample configuration file in `./samples/local.conf`. This file needs to be copied to the installation directory and then modified as required. The file is well commented, so it should be fairly intuitive.

Finally, start the installation:

```
./stack.sh
```

At this point, the script will ask you for any additional information that it requires. Then, it will download and configure all of the software packages it needs. This will take some time!

Operation

Most of the command-line tools, such as Nova or Keystone, require your to provide credentials before they will function.

You can pass these credentials as command-line arguments, but doing so is tedious. You also can pass the credentials from the shell's environment. The best way to do that is to create a short script that exports the appropriate variables. Here is an example:

```
#!/bin/bash
export OS_USERNAME=admin
export OS_PASSWORD=supersecret
export OS_TENANT_NAME=demo
export OS_AUTH_URL=http://192.168.1.231:35357/v2.0
```

Then, you simply can source this file before using the command-line tools. Before going much further, it would behoove you to generate and install an SSH key to simplify your access to the potentially many virtual servers that you will be creating. After setting your environment variables, as discussed earlier:

```
ssh-keygen
nova keypair-add --pub-key ~/.ssh/id_rsa.pub demo-key
```

You can see that your key was imported into OpenStack:

```
nova keypair-list
```

Which results in something like this:

Name	Fingerprint
demo-key	66:13:9e:2b:f4:39:4c:e0:67:7d:dc:b2:0c:9e:3b:14

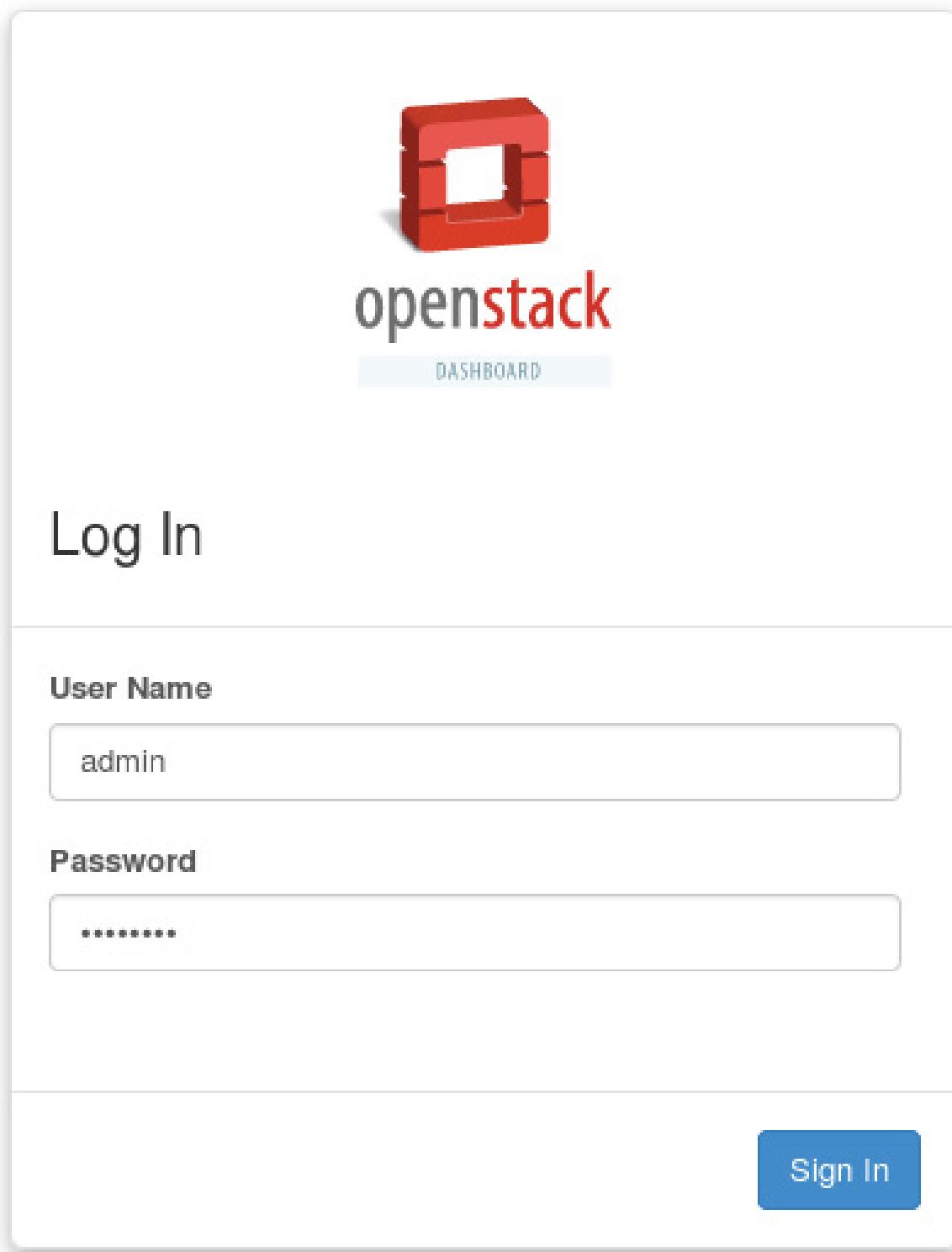


FIGURE 2. OpenStack Login

Every administrative and operational task can be performed from the command line. However, some tasks are done more easily using the dashboard, which is accessible from a Web browser. To access the OpenStack dashboard, simply point a Web browser at its IP address. You will log in with the same credentials discussed above. Once you are logged in, you can go to the Images tab and get a list of all the available images. DevStack installs a couple images by default. You can start a virtual machine by launching an image. You will be asked for a few parameters for the new virtual machine, including what flavor of server to launch it in. OpenStack allows you to define various flavors of virtual server by specifying how many cores, how much RAM and how much disk space the virtual machine has. Tracking how many of each flavor you have deployed enables you to do capacity planning. Once you have specified what type of server you want, you will get an instance of that server. You can see all of the running instances by clicking on the Instances tab.

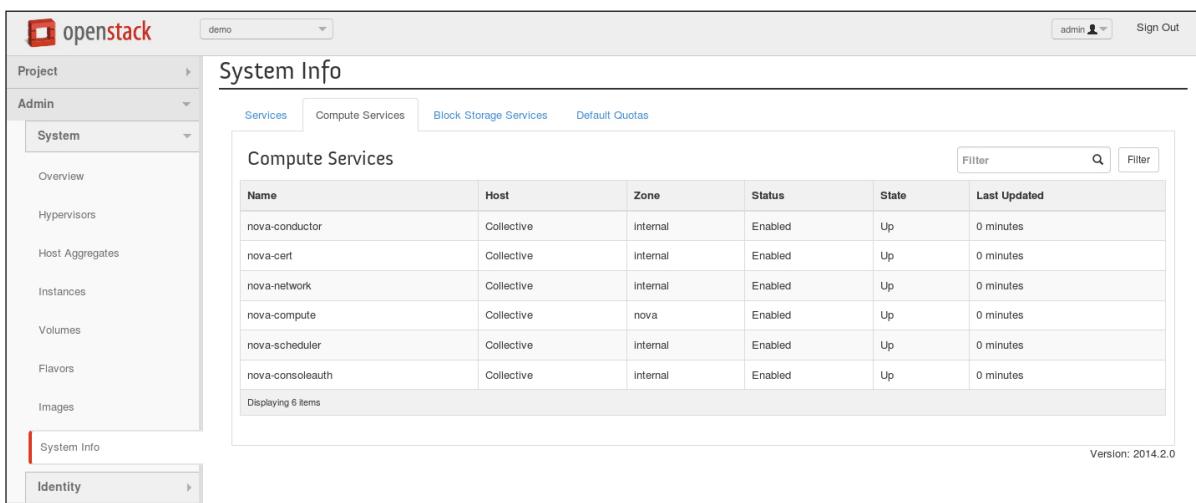


FIGURE 3. OpenStack User Interface

User management in OpenStack shouldn't be a large burden, but managing users probably is easier from the dashboard. If you find yourself managing servers in an environment where various departments need to be isolated from each other for whatever reason, you can use the dashboard to configure each department as its own separate project. Each project has its own access control lists that determine what traffic can reach its servers.

The easiest way to add new images to OpenStack is through the dashboard. From the dashboard, you either can upload an image to the server, or you can paste in a URL, and OpenStack will download and import the image for you. If you import an image under the Admin tab, that image will be made available for use by everyone. On the other hand, if you import an image under a particular project, only people who have access to that project will be able to launch the new image.

Once you have a virtual machine running, you're going to want to connect to it. There are a few ways to do this. However, if you are attempting to access the virtual server from a machine other than the host server, you may have to create an access control rule to allow traffic from the LAN to the virtual machine. You can do that from the Access and Security tab for the particular project that owns the server instance. Then, you should be able to ssh to the device as though it were a "real" server.

However, if you need access to the server's console, you will need to get a token which will allow you to use

the vnc protocol to access the console. In the command, below, you simply supply the appropriate instance name.

```
nova get-vnc-console instance_name novnc
```

```
+-----+-----+
| Type | Url |
+-----+-----+
| novnc | http://192.168.1.231.6080/vnc_auto.html?token=
|       | ↗c2ea5348-86b3-41c5-8896-836bd426dc7e |
+-----+-----+
```

Then, simply copy the URL from the output and paste it into your browser's location field. OpenStack will provide you with a window that contains the contents of your virtual server's console.

So, at this point, you should have a fully functional OpenStack server with a few virtual machines configured and ready to run. You also should have a basic understanding of how to manage the system and connect to the virtual servers.

But, this system isn't ready to be put into actual production. It's only configured well enough to allow system administrators to become familiar with the day-to-day operation. A production-ready server probably would have more than one compute node configured. Also, you should consider configuring redundant controllers. Performance-tuning and high-availability are beyond the scope of this eBook, but are well documented at the OpenStack Web site.■

GEEK GUIDE ▶ BUILD A PRIVATE CLOUD FOR LESS THAN \$10,000!

TABLE 1. Private Cloud Cost Breakdown

EQUIPMENT	MAKE	MODEL	DESCRIPTION	COUNT	UNIT COST
Server	Supermicro	SD-5037MC-H8TRF	3U Rackmount Server (8 Node MicroCloud), 1x E3-1200v2, 32GB ECC Per Node, 2x 3.5" HS Bays Per Node, 1620WRPS	1	\$6,800.00
Disk Drive	Seagate	ST6000NM0024	6T SATA 6Gb/s v.4 with 128Mb cache	2	\$500
Operating System	Ubuntu Server			1	\$0.00
Software	OpenStack			1	\$0.00
Total:					\$7,800.00

TABLE 2. Private Cloud Build Time

TASK	TIME SPENT
Server setup and configuration	2 hours
Operating system installation	1 hour
OpenStack/DevStack installation	1 hour
Server installation	2 hours
Total Time:	6 hours

Resources

OpenStack: <http://www.openstack.org>

OpenStack Installation Guide for Ubuntu 12.04/14.04 (LTS):
<http://docs.openstack.org/icehouse/install-guide/install/apt/content/index.html>

DevStack: <http://devstack.org>

Servers Direct Supermicro SD-5037MC-H8TRF:
http://www.serversdirect.com/Solutions/Datacenter/id-SD-5037MC-H8TRF/Supermicro_SD-5037MC-H8TRF.
Starting price \$7,357 with eight 1TB hard drives or \$11,245 with eight 6TB hard drives.

Servers Direct Supermicro SD-6017R-TDF:
http://www.serversdirect.com/Servers/Rackmount/id-SD-6017R-TDF/Supermicro_SD-6017R-TDF.
Configured price: \$1,195.

Seagate ST6000NM0034 6TB SAS 3.5":
http://www.serversdirect.com/Components/Drives/id-HD1537/Seagate_ST6000NM0034_6TB_SAS_3_5.
List price \$599.

TurnKey Hub: <http://www.turnkeylinux.org>

Rackspace Public Cloud: <http://www.rackspacecloud.com>