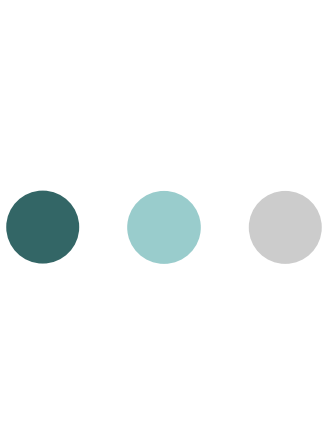
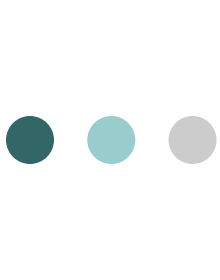


Lecture 4

Intro to Cloud & OpenStack



Chapter 13

Introduction to the Cloud
and
Introduction to the OpenStack Cloud

Just What is a Cloud, Anyway?

IBM (2016) defines cloud computing as follows:

> “Cloud computing, often referred to as simply “the cloud,” is the delivery of on-demand computing resources—everything from applications to data centers—over the Internet on a pay-for-use basis.”

Just What is a Cloud, Anyway? (cont'd)

Public cloud:

Instead of doing your own computing and storing your data on the computer on your desktop yourself:

- you hire a company to do the computing and store the data on their big computer servers that you access via the web.

Just What is a Cloud, Anyway? (cont'd)

Private cloud:

Instead of having computing and data storage on employees' desks:

- the company can buy its own big servers
- then the employees do their computing on those servers and store their data on those servers.

Just What is a Cloud, Anyway? (cont'd)

Hybrid cloud:

- where part of a company's computing is provided by a separate cloud company
- and part is done in house

Just What is a Cloud, Anyway? (cont'd)

There are three cloud service models:

- Infrastructure as a Service (IaaS)
- Platform as a Service (PaaS)
- Software as a Service (SaaS)

Cloud Service Models: Infrastructure as a Service

- Cloud provider “rents out” infrastructure, servers, storage, networking:
  - offers pools of virtual machines (or can be physical computers)
  - Storage
    - Block storage accessed directly or by network drive
    - Object storage accessed through web interface
  - firewalls, etc.
- Users provide OS images and their own application software
  - User maintains their own OS and Application SW
- Example: Amazon Elastic Compute Cloud (EC2)

Cloud Service Models: Platform as a Service

Cloud provider manages the computing platform and underlying hardware and offers a complete computing platform, including: OS, Web server, Database, storage (block or object), firewalls, etc.

- User provides the application software
- User manages the application software
- Examples:
  - Amazon Web Services Elastic Beanstalk
  - Google App Engine

Cloud Service Models: Software as a Service

Cloud provider offers application software and databases

Cloud provider manages and operates the application software

- Software is “rented” rather than purchased
- Software as a Service is sometimes known as “Software-on-demand”
- Examples:
  - Microsoft Office 365
  - DropBox
  - Google Docs
  - Citrix GoToMeeting
  - Various Salesforce.com offerings: Sales Cloud, Service, Cloud, etc.

Cloud Service Models: Software as a Service

- Sometimes a SaaS offering can be run on top of some other cloud provider's IaaS or PaaS offering
  - For example, DropBox originally ran its offerings on top of the Amazon cloud –see Metz (2016)

Cloud Service Models: intermediate Cloud Offerings: some examples

- Microsoft Azure allows both IaaS and PaaS operation
- Salesforce.com offers Force.com, which allows external developers to create add on applications that integrate with the main salesforce.com offerings

Why the Cloud? Why not the Cloud?

- Under what circumstances should a company move its application(s) or general IT services to a public cloud?
- When it is better for applications and IT services to stay un-clouded ☺?
- Should a company run its own private cloud?
- Should a company use a hybrid cloud?

Why the Cloud? Why not the Cloud? (cont'd)

- Many startup companies begin by hosting their applications on a cloud—see Metz (2016) for DropBox on Amazon:
  - They don't have to buy servers
  - They don't have to hire IT staff
  - It saves capital to spend on development or marketing that is more directly related to making money

Why the Cloud? Why not the Cloud? (cont'd)

- Eventually it can be worthwhile financially to move away from a cloud. An example is DropBox moving away from Amazon, see Metz (2016) :
  - the huge scale of DropBox in terms of quantities of data (500 petabytes) and numbers of users (500 million) as of March 2016:
    - financially justified buying their own servers and hiring their own system (cloud) administrators
  - running their own cloud allowed them to have end to end control so they could improve a user's experience (along with potentially improving performance and reliability)

Why the Cloud? Why not the Cloud? (cont'd)

- However, a mobile app company named Zynga started on Amazon Web Services. See Metz(2016) and Butler (2016):
  - As their business grew they created their own Z-cloud
  - Their business later dropped off greatly and they returned to Amazon Web Services cloud

Why the Cloud? Why not the Cloud? (cont'd): Benefits of Using a Cloud (cont'd)

If you use a (public) cloud:

- if you don't have a huge quantity of users, data, or processing, then your costs are probably lower:
  - you won't have to pay for as many system administrators
  - you won't have to buy big servers
  - you don't have to manage big networks
- you can have scalability, that is, if you have a sudden increase in users then the cloud infrastructure can handle it (of course, at a price).
  - to be able to do this yourself you would have to pay for and maintain extra infrastructure in terms of servers, networks, and system administrators
- your data may be backed up at a different location, such that it can survive a large natural disaster. Or possibly even a war.

Why the Cloud? Why not the Cloud? (cont'd): Benefits of Using a Cloud (cont'd)

There are some security advantages:

- the cloud provider has a big incentive to keep all its software up to date with security patches
  - whereas you would have to have a good staff of system administrators, as well as regular update procedures, to make sure this is done in your own company
- the cloud most likely provides some sort of physical security for its servers
  - if you do the same in your company, this requires locked rooms with access control, and additional procedures that must be followed. For example, who gets a key? If someone is fired, do you change all the locks? Etc.
- the cloud provider has a big incentive to hire good system administrators who have been background-checked

Why the Cloud? Why not the Cloud? (cont'd): Benefits of Using a Cloud (cont'd)

There are some advantages in regard to data privacy:

- there are some expensive certifications that can be difficult for a small company to acquire:
  - Wood and Tracy (2011) say that certification can be expensive for the healthcare industry to comply with the Health Insurance Portability and Accountability Action (HIPAA) and for the federal government organizations to comply with the Federal Information Security Management Act (FISMA)

Why the Cloud? Why not the Cloud? (cont'd): Negatives of Using a Cloud

What are some negatives for having your company's software on a cloud?

- Your particular application may not benefit from added scalability, particularly if it is a traditional monolithic application architecture for which a stable demand is expected
- It may be difficult to organize or re-write your application such that it would run well when located on a cloud.
  - There is some literature that says that an application must have a Service Oriented Architecture in order to be able to be hosted on a cloud
- Even if it is possible to move your application to the cloud such that its performance is good, there will probably be some cost involved in making this happen. Perhaps significant cost.

Why the Cloud? Why not the Cloud? (cont'd): Negatives of Using a Cloud (cont'd)--Security

There may be security disadvantages associated with being on a cloud:

- The cloud provider has a big incentive to have good security, however, if you onboard your software to a cloud, you give up control of its security
  - you can perhaps specify in your Service Level Agreement that certain security measures be taken
  - But how do you know this is actually being done?

Why the Cloud? Why not the Cloud? (cont'd): Negatives of Using a Cloud (cont'd)--Privacy

There can be data privacy disadvantages associated with being on a cloud:

- your data would be managed by the cloud provider, not by you.
  - The system administrators would be vetted (given background checks) by the cloud provider, not by you
- in large cloud companies with data centers located in other countries (maybe not that friendly to your own country), then physical location of the data can be a big issue
- how do you know the cloud provider is not mining your private data for its own purposes—see Winkler (2012)—and perhaps selling this information to a third party?
- how do you ensure that your cloud provider is, indeed, meeting legal security requirements for your data, in cases where data privacy is legally required?
  - What is your company's liability if the cloud provider does this wrong?

Why the Cloud? Why not the Cloud? (cont'd): Negatives of Using a Cloud (cont'd)—Privacy (cont'd)

- there are cases where by law or by privacy agreement, an individual wishes to delete his/her private information. How can you ensure the cloud provider actually does this?
- What is your cloud provider's policy on notifying you if there has been a data breach? How long will this take?—see Winkler (2012)
- What is your responsibility for notifying the cloud provider if your data has been breached? Do you have any liability?

Why the Cloud? Why not the Cloud? (cont'd): Negatives of Using a Cloud (cont'd)—Performance

There may be performance disadvantages associated with being on a cloud:

- It could be that the cloud's internal technologies are not well-tuned to how your application runs
  - Perhaps the cloud's infrastructure has been tuned to work well for a bigger customer
  - Thinking in a sneaky or maybe paranoid way, how do you know that "bigger customer" isn't your major competitor? After all, both companies can buy services from the same cloud provider

Who are the Clouds?

- As of May, 2015, far and away the largest cloud is Amazon Web Services—see Darrow (2015) of Fortune Magazine
- According to the Gartner Magic Quadrant report AWS supplied more than ten times the computing power to customers than the combined other fourteen top cloud providers (of the top fifteen).
  - The Gartner, Inc. Magic Quadrant report is market research into a specific technology industry.

Who are the Clouds? (cont'd)

According to Tsidulko (2015), the remaining cloud providers in the top 15 are:

- Microsoft Azure (the second biggest cloud provider)
- Google Compute Engine
- CenturyLink
- VMWare vCloud Air
- IBM SoftLayer
- Rackspace
- Verizon Terremark
- Virtustream
- CSC
- Fujitsu
- Joyent
- NTT Communications, Interoute
- Dimension Data

Who are the Clouds? (cont'd)

Of the top 15 cloud providers, several provide proprietary cloud implementations, these include:

- Amazon Web Services
- Microsoft Azure
- Google Compute Engine
- VMWare vCloud Air
- Verizon Terremark
- Joyent

Who are the Clouds? (cont'd)

Of the top 15 cloud providers, several provide an open source cloud:

- The following employ the OpenStack cloud computing platform:
  - IBM SoftLayer
  - Rackspace
  - CSC BizCloud
  - NTT Communications
  - Fujitsu
- The following employs the CloudStack cloud computing platform:
  - Interoute

Who are the Clouds? (cont'd)

- According to Butler (2015), Cantrill, the CTO of Joyent considers OpenStack to be an example of what *not* to do.

- Joyent employs Triton containers instead of virtual machines. According to Cantrill:

  - "Triton lets you run secure Linux containers directly on bare metal via an elastic Docker host that offers tightly integrated software-defined networking."

Who are the Clouds? (cont'd)

- A Docker container, Docker (2016) is different from a virtual machine in that an operating system is not included in the container
- Instead, a Docker container uses the same Linux kernel as the system it's running on. See Opensource.com (2016):
  - This gives a Docker container higher performance than a virtual machine
  - Also, Docker containers only need to contain software that the host computer doesn't already have

Open Source Software supporting PaaS

- Some PaaS services allow developers to quickly create and deploy code on a cloud infrastructure. According to Butler (2016), the following PaaS technologies are being integrated with OpenStack:
  - Kubernetes—used/experimented with by 42% of OpenStack users
  - Cloud Foundry—used/experimented with by 24% of OpenStack users
  - Apache Mesos
  - RedHat OpenShift

Open Source Software supporting PaaS

- Kubernetes (made by Google) is used to automate the deployment of application containers across hosts

- According to StackOverflow (2015):

  “Kubernetes is a cluster orchestration system inspired by the container orchestration that runs at Google. Built by many of the same engineers who built that system. It was designed from the ground up to be an environment for building distributed applications from containers. It includes primitives for replication and service discovery as core primitives, whereas such things are added via frameworks in Mesos. The primary goal of Kubernetes is a system for building, running and managing distributed systems.”

Open Source Software supporting PaaS

- Kubernetes can be compared to Docker Swarm.
- Docker Swarm makes a pool of Docker hosts look like a single, virtual Docker host
  - That is, it makes a cluster of Docker hosts look like a single Docker API
- Comparing Kubernetes to Docker, Farcic (2015) says:

  “Kubernetes and Docker Swarm are probably [the] two most commonly used tools to deploy containers inside a cluster. Both are created as helper tools that can be used to manage a cluster of containers and treat all servers as a single unit.”

Open Source Software supporting PaaS

- Cloud Foundry is an open source PaaS technology that is used to build and deploy applications on public and private clouds. See Babcock (2015):
  - providing scalability is one of its abilities
  - to provide lifecycle management CloudFoundry uses an open source tool called BOSH
  - The Cloud Foundry Foundation has 50+ members, including as of 2016: Pivotal, VMWare, IBM, Accenture, Docker, Verizon, Cisco, Fujitsu, SUSE, among many others. Several companies offer PaaS services implemented using Cloud Foundry, these include IBM BlueMix and HP Helion among several others.
- Cloud Foundry was originally created by VMWare
  - VMWare and its parent company EMC spun off CloudFoundry when it spun off Pivotal in 2013
  - There is a separate commercial product called Pivotal Cloud Foundry

Open Source Software supporting PaaS

- According to Babcock (2015), CloudFoundry:
  - supplies data services, including MySQL-as-a-Service,
  - supplies RabbitMP messaging (we will talk briefly about RabbitMQ later on).
  - provides a lightweight application server called Spring Boot that can be included with an application.
    - The Spring framework is an application framework for Java, and Spring Boot is a method for creating a stand-alone, easy to run Spring framework-based application.
  - CloudFoundry can also be integrated with Apache Tomcat.
  - CloudFoundry supports Java, PHP, Python, Ruby, Node.js, Perl, and Google's Golang (Go)

Open Source Software supporting PaaS

Apache Mesos (previously called Nexus), is an open source cluster manager, see Vivien (2015):

- It provides abstractions of compute resources (CPU, storage, etc.) away from both physical computers and virtual machines
- Distributed systems running on top of Mesos are called frameworks
- Mesos decides what resources to give to a framework, then the framework decides how resources are allocated within its distributed system.
- originally developed at the University of California in Berkeley
- Mesos supports C/C++, Java, Scala, Python, Perl, Haskell, and Go,

Open Source Software supporting PaaS

OpenShift is the name of a PaaS offering by RedHat that uses the open source OpenShift Origin (2016) application container platform

- OpenShift Origin is used to build, test, deploy, and manage cloud applications
- It also provides lifecycle management
- It is built on Docker containers and Kubernetes cluster management
- OpenShift Origin has “cartridges,” which provide the functionality required to run an application
- These cartridges can include programming language support, database support, etc.
  - are extensible so any programming language or tool can be added

Open Source Platforms Supporting IaaS

According to Kleman (2015) arguably the major open source cloud platforms in the market that provide infrastructure as a service (IaaS) support are:

- OpenStack
- Apache CloudStack
- Eucalyptus
- OpenNebula

Open Source Platforms Supporting IaaS

When comparing open source cloud platforms, OpenStack is by far the most widely used at the present time.

- Kleman (2015) mentioned that as of 2015 more than 150 companies were contributing to OpenStack development. Kleman says:
  - “Let’s face facts: OpenStack is a more mature and more widely adopted platform. But that doesn’t mean it’s not facing the heat of other players in the market. There is a lot of money being pumped into platforms like CloudStack and even Eucalyptus”

Open Source Platforms Supporting IaaS

According to Butler (2016), OpenStack can be used to create a public cloud or a private cloud:

- 65% of the OpenStack users have used it for private clouds
- 16% have used it to create public clouds.
- Another 12% hire a service provider to host deployment of a dedicated OpenStack

Open Source Platforms Supporting IaaS (cont'd)

Cloud	History	License	Governance Model	Language Written in	Installaiton Difficulty
OpenStack	Founded by Rackspace and NASA in 2010	Apache v2.0	Foundation	Python	difficult
CloudStack	Released by Citrix into Apache in 2011	Apache v2.0	Technical Meritocracy	Java,C	Medium
Eucalyptus	Began at Rice University and UCSB, now sponsored by HP. Has formal compatibility agreement with AWS	GPL v 3.0	Benevolent Dictator	Java,C	Medium
OpenNebula	Sponsored by OpenNebula Systems	Apache v2.0	Benevolent Dictation	C++, C, Ruby, Java, shell script, lex, yacc	

OpenStack Cloud--OpenStack Component Services

Component	Service	Description
Nova	Compute	Manages and automates computing resources, controls virtual machines
Glance	Image	Stores disk images, and snapshots of virtual machines
Keystone	Identity	User authentication for all OpenStack services, role-based access
Horizon	Dashboard/GUI	allows provisioning and automating services
Neutron	Networking	Manages the networking of the cloud. IP addresses, VLANS, etc.
Cinder	Block Storage	Allows blocks of storage to be assigned to compute instances.

OpenStack Cloud (cont'd)-- OpenStack Component Services (cont'd)

Component	Service	Description
Swift	Object Storage	Object storage, accessed through web API
Heat	Orchestration	allows a virtual machine's compute, networking and storage to be automatically configured
Trove	Database	Handles databases
Ceilometer	Telemetry	Monitoring and Billing system
Sahara	Elastic Map Reduce	Handles Hadoop Clusters
Zaqar	Multiple Tenant Cloud Messaging	Can be used in SaaS to send messages between multiple components of the SaaS

OpenStack Cloud (cont'd)—OpenStack Component Services (cont'd)

Component	Service	Description
Manila	Shared file system service	API includes primitives to create, delete, and give/deny access to a share of a file system
Searchlight	Search service	Allows search across different OpenStack services
Barbican	Security API	Secure storage and management of passwords, encryption keys, and certificates

Some OpenStack Supported Public Clouds as of Jan. 2015

Internap's Agile Cloud (Netherlands, Canada)	IO Cloud (USA)
Rackspace (USA, China, UK, Australia)	City Cloud (Sweden)
DreamCompute (USA)	HP Helion Public Cloud
teutoStack Public cloud (Germany)	DataCentred Public Cloud (UK)
OVH Group RunAbove (Canada, France)	Numergy Cloud (France)
UOS Cloud (China)	KLOUD (Mexico)
Dualtec Public Cloud (Brazil)	Elastx (Sweden)
Anchor OpenCloud (Australia)	Cloudwatts Cloud Services (France)
Auro Enterprise Public Cloud (Canada)	

Some OpenStack Supported Private Clouds as of Jan. 2015

IBM Cloud Openstack Services (Netherlands, USA, China, UK, Australia, Singapore, Canada)	Rackspace Private Cloud (China, USA, UK, Australia)
Metacloud OpenStack (China, Netherlands, UK, USA,Singapore)	Blue Box Cloud Private Cloud (USA, Switzerland)
BlueBox Cloud (USA)	UOS Managed Private Cloud (China)
Morphlabs Enterprise Cloud (Philippines)	DataCentred Private Cloud (UK)

OpenStack History

The OpenStack cloud was originally created by combining the Nebula cloud computing project from NASA with the Swift Object storage project from Rackspace. Llewellyln (2012) quotes the former Nebula project manager, RayO'Brien from NASA, as follows:

- "…on the first open source release of our Nova controller, we found that Rackspace had taken a strikingly similar technical approach to their storage systems and were set to begin the construction of a compute controller just as we were preparing to focus on storage. Given our technical alignment and with the open source release of Rackspace's Swift storage software, we joined forces to create OpenStack. Our hope was that a community would form around these two pieces of software toward the construction of an open source cloud operating system. To say that our greatest hopes in this regard were met would be an understatement. OpenStack today has the support of hundreds of individuals and organizations around the world, all set on realizing the original vision for the project."

OpenStack Suggested Minimal Installs circa 2016

There are two suggested minimal installation architectures for OpenStack.

First, if you use Neutron, there is a three node minimal installation:

- Controller node—runs Keystone Identity Service, Glance image service, Horizon dashboard, message queue and supporting SQL database, as well as management portions for the Compute node, Networking node. Also runs Networking plugins.
- Compute node (can be more than one)—runs portion of the hypervisor that operates virtual machines (the KVM hypervisor is the default). It also runs the Networking plug-in and an agent to connect project (tenant) networks to VMs and to provide firewall and security group services.
- Networking node—runs agents to handle internet connectivity for VMs. Also provisions project (tenant) networks and provide switching, routing, Dynamic Host Configuration Protocol (DHCP), and Network Address Translation (NAT) services.

OpenStack Suggested Minimal Installs circa 2016 (cont'd)

Second, if you don't use Neutron but instead use nova-network (which has been deprecated):

- Controller node—runs Keystone Identity Service, Glance image service, Horizon dashboard, message queue and supporting SQL database, as well as management portion of the Compute node.
- Compute node (can be more than one)—runs portion of the hypervisor that operates virtual machines (the KVM hypervisor is the default). . It provisions project (tenant) networks and provides firewall and security group services

OpenStack Install Tips circa 2016

- Installing OpenStack can be difficult. One who has recently gone through the OpenStack installation process might be tempted to add a few expletives in front of the word “difficult.”
- One of the major reasons an OpenStack install is so difficult is that OpenStack consists of numerous sub-applications, each with its own issues.
- You also have to choose which hypervisor you’re going to work with, and install that. According to Butler (2016), 93% of OpenStack clouds use the Kernel Virtual Machine (KVM) hypervisor.
  - OpenStack sits alongside the hypervisor, and gives you control over it, it basically gives an administrator interface to the hypervisor to start up virtual machines and make changes to virtual machines.
- First of all, make sure you have enough RAM !!!!! The absolute smallest quantity of RAM that we’ve been able to make OpenStack really work was 16 Gigabytes, and you really need at the very least 32 Gigabytes.

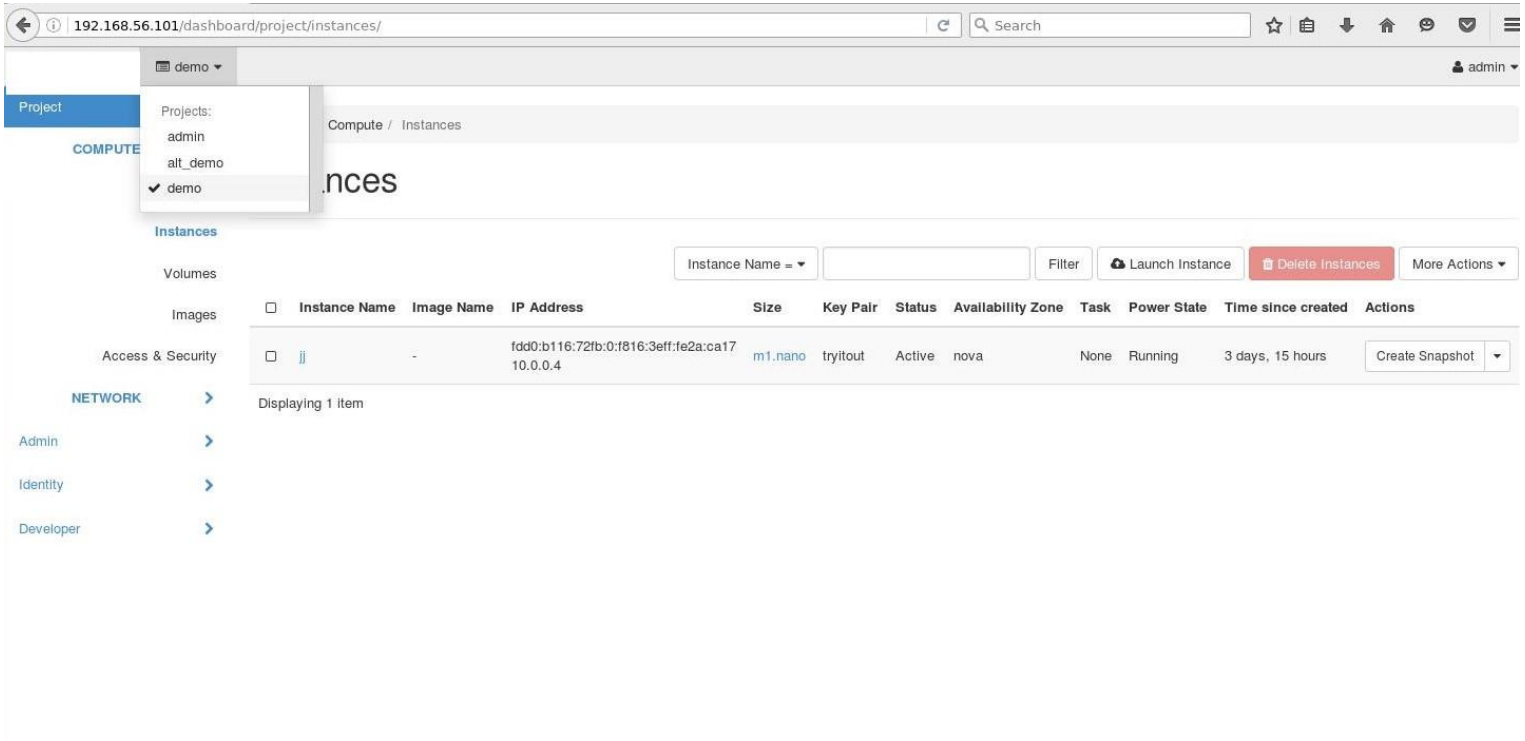
OpenStack Install Tips circa 2016 (cont'd)

- Our easiest installs have been achieved using DevStack:
  - http://docs.openstack.org/developer/devstack/
- A DevStack install, however, is not intended to be a full production install of OpenStack.
- DevStack is not a general OpenStack installer. However, for using OpenStack for academic purposes it works *fairly* well. Note that you may end up with the occasional unexpected thing:
  - In the middle of Spring, 2016, the latest version of the DevStack install didn't have the OpenStack Amazon Web Services EC2 interface enabled, we found we had to add a plugin for that.

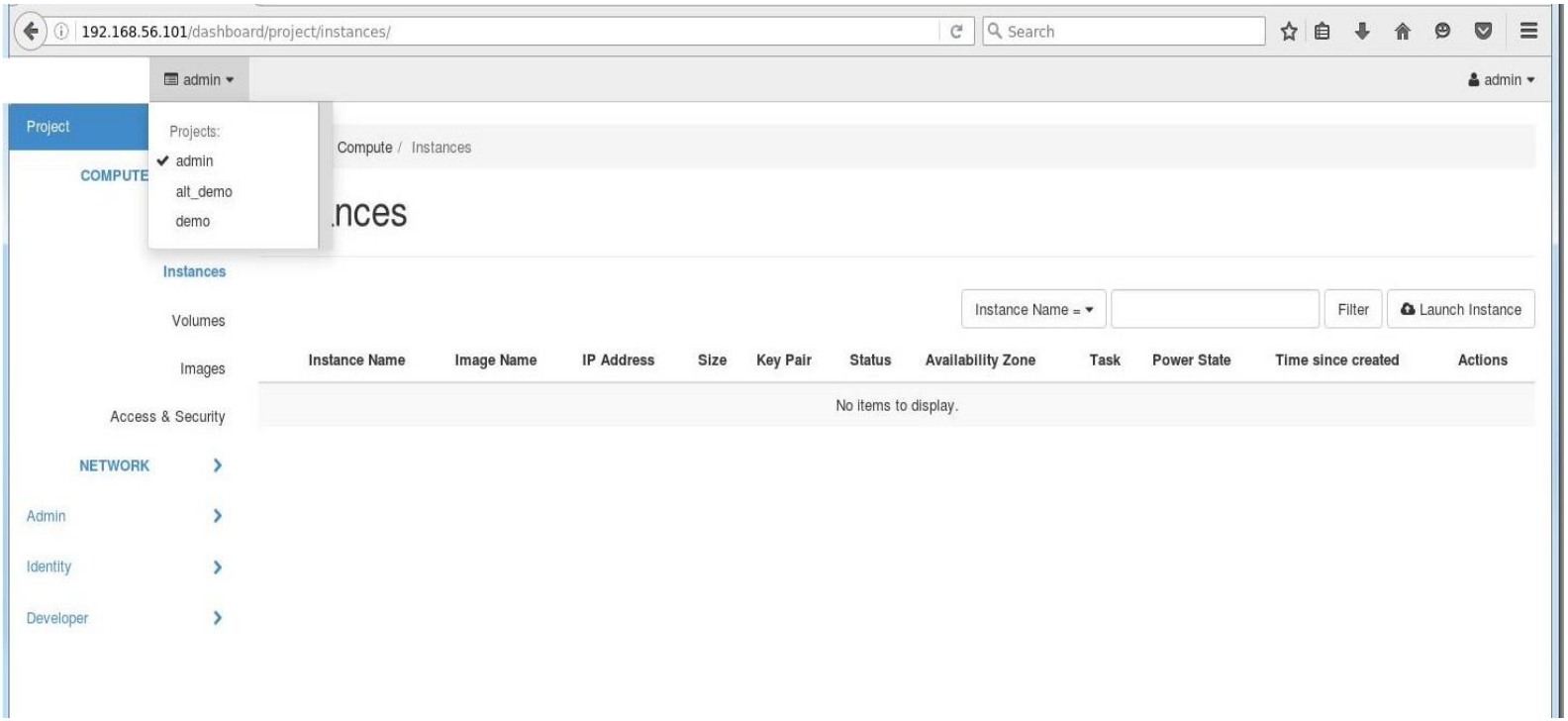
How to use OpenStack Dashboard (cont'd)

- Basic DevStack install gives you a project and login named demo, and a project and login named admin.
- You can, of course, add additional projects.

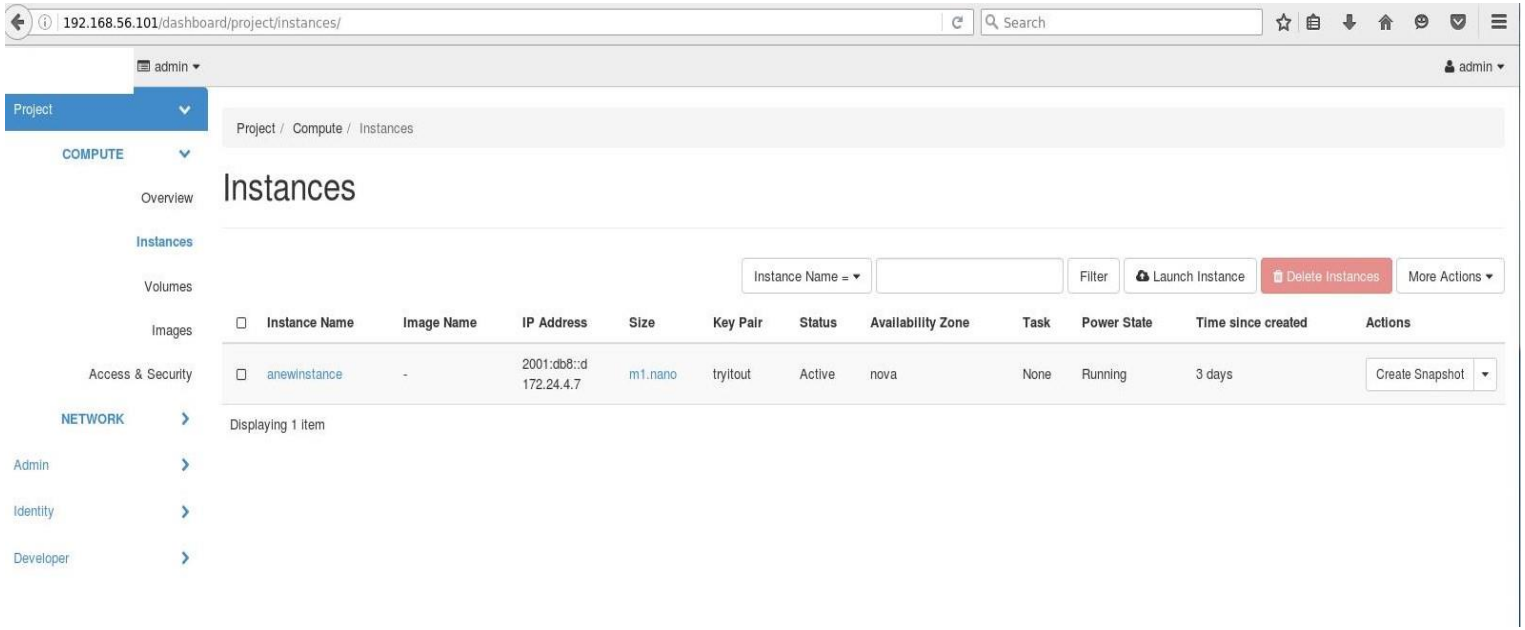
How to use OpenStack Dashboard (cont'd)—Select Project



How to use OpenStack Dashboard (cont'd)—Admin Project Tab



How to use OpenStack Dashboard (cont'd)—Instances in Admin Project



How to use OpenStack Dashboard (cont'd)—Instances in Admin Project (cont'd)

- You can see from the previous figure that a an instance of a cirros operating system was created.
- The CirrOS OS is a test OS, that does very little but also doesn't require much memory to run:
  - cirros-0.3.2-x86_64-uec
- The instance is called "anewinstance"
- By default using private IP address 172.24.4.7
- Using m1.nano size
- Using a previously created keypair named "tryitout"

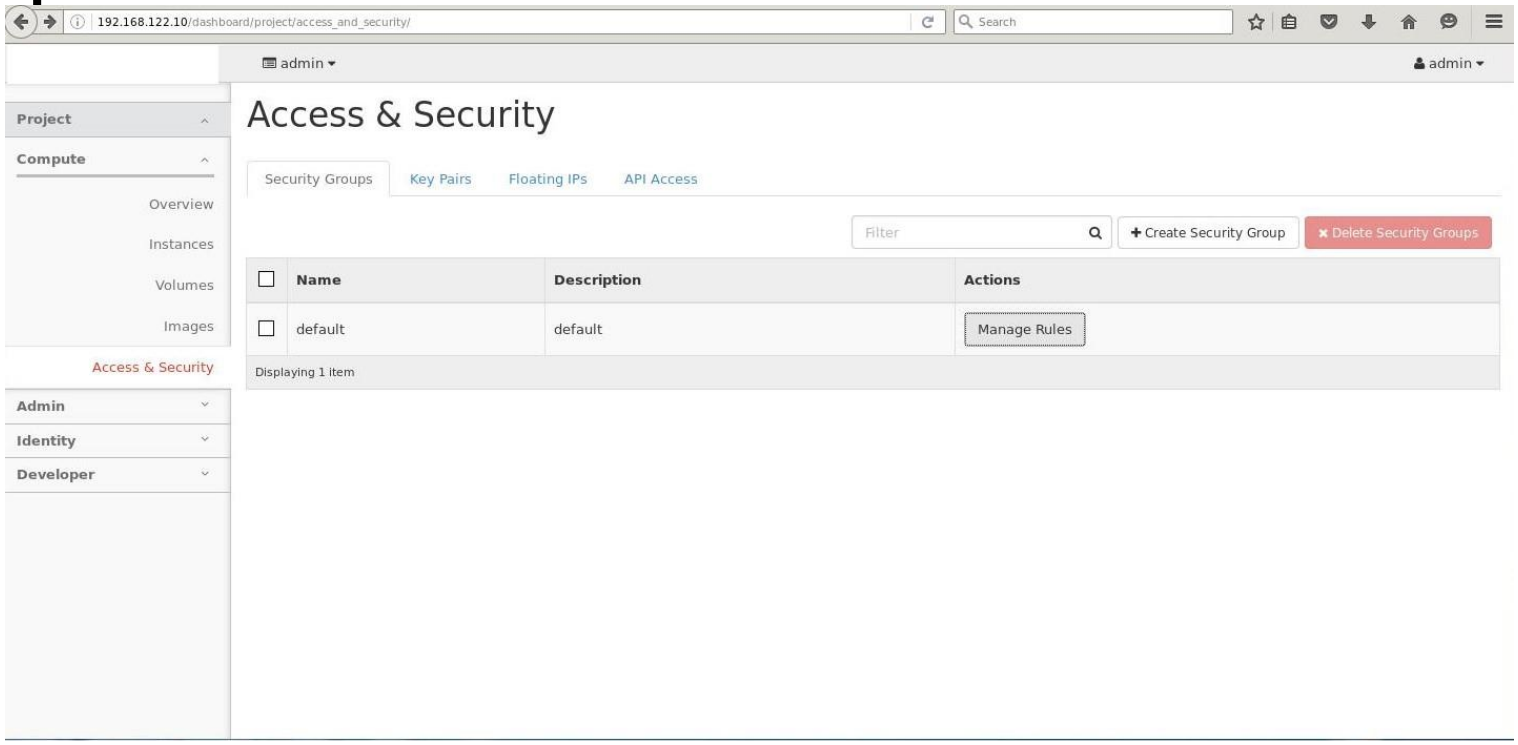
How to use OpenStack Dashboard (cont'd)—Instances in Admin Project (cont'd)

- An "instance" is what OpenStack calls a running virtual machine
- An image is a copy of an operating system.
- When an instance is started, it must be given some kind of OS to run
- OpenStack can run many different versions of Linux (Ubuntu, RedHat, Fedora, SUSE, Debian) and it can run Windows
- Several tested images are available or you can create your own image

How to use OpenStack Dashboard (cont'd)—Instances in Admin Project (cont'd)

- The keypair associated with an OpenStack instance consists of a public key and a private key
- You can use the OpenStack dashboard to create these, or you can create them other ways (such as with ssh-keygen)
- The public key must be registered with OpenStack
- You must download the private key to keep it safe:
  - You will use your private key later on to access your OpenStack instance

How to use OpenStack Dashboard (cont'd)—Key Pairs Tab



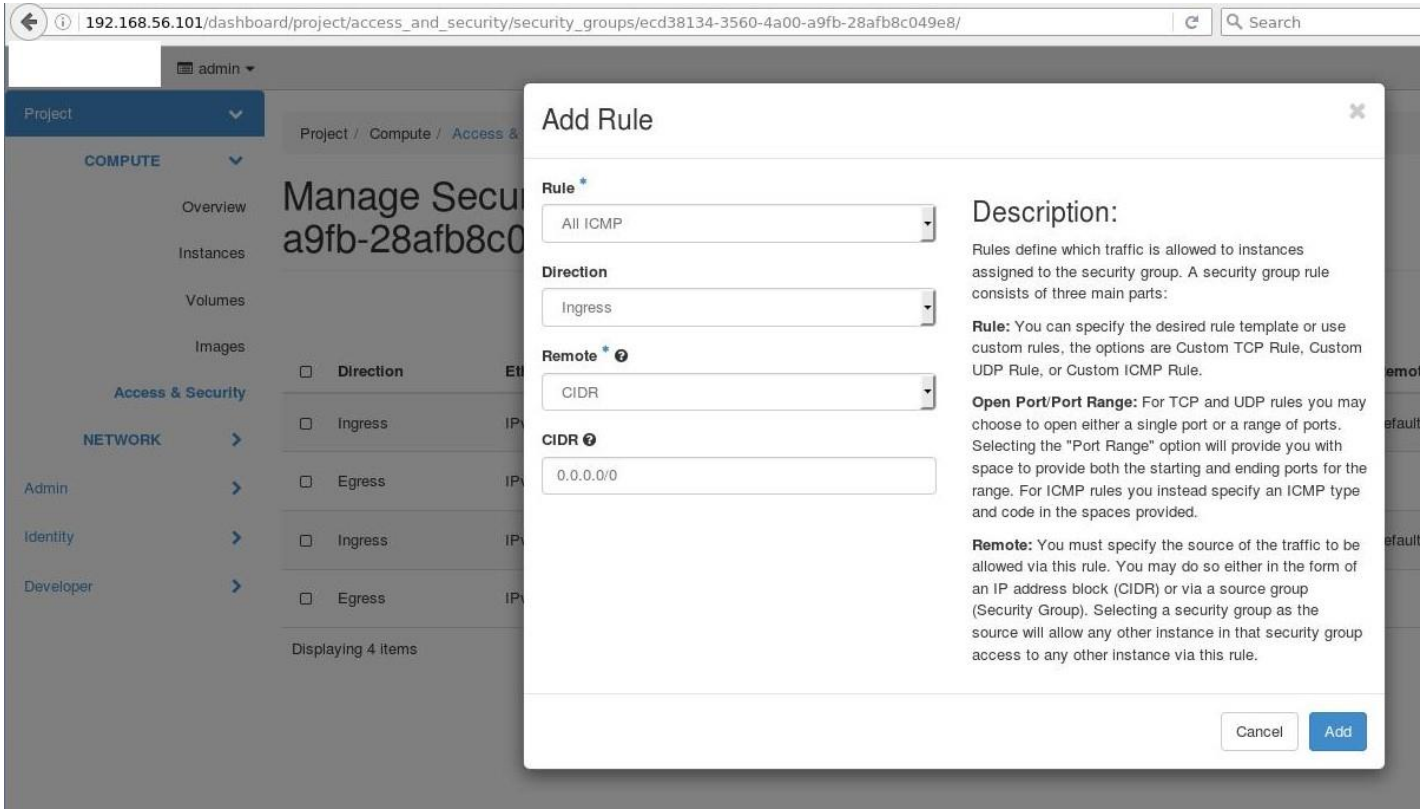
How to use OpenStack Dashboard (cont'd)—Create Key Pair (cont'd)

- Create a keypair named "myveryownkeypair"
- you will get a notification that a file named "myveryownkeypair.pem" has automatically downloaded
- This .pem file will contain your private key
  - The ".pem" extension specifies a container file that can store private keys, public keys, or digital certificates
  - It originally stood for "***p***rivacy ***e***nhanced e***m***ail"
    - That particular email technology was never heavily used, but the .pem files remain

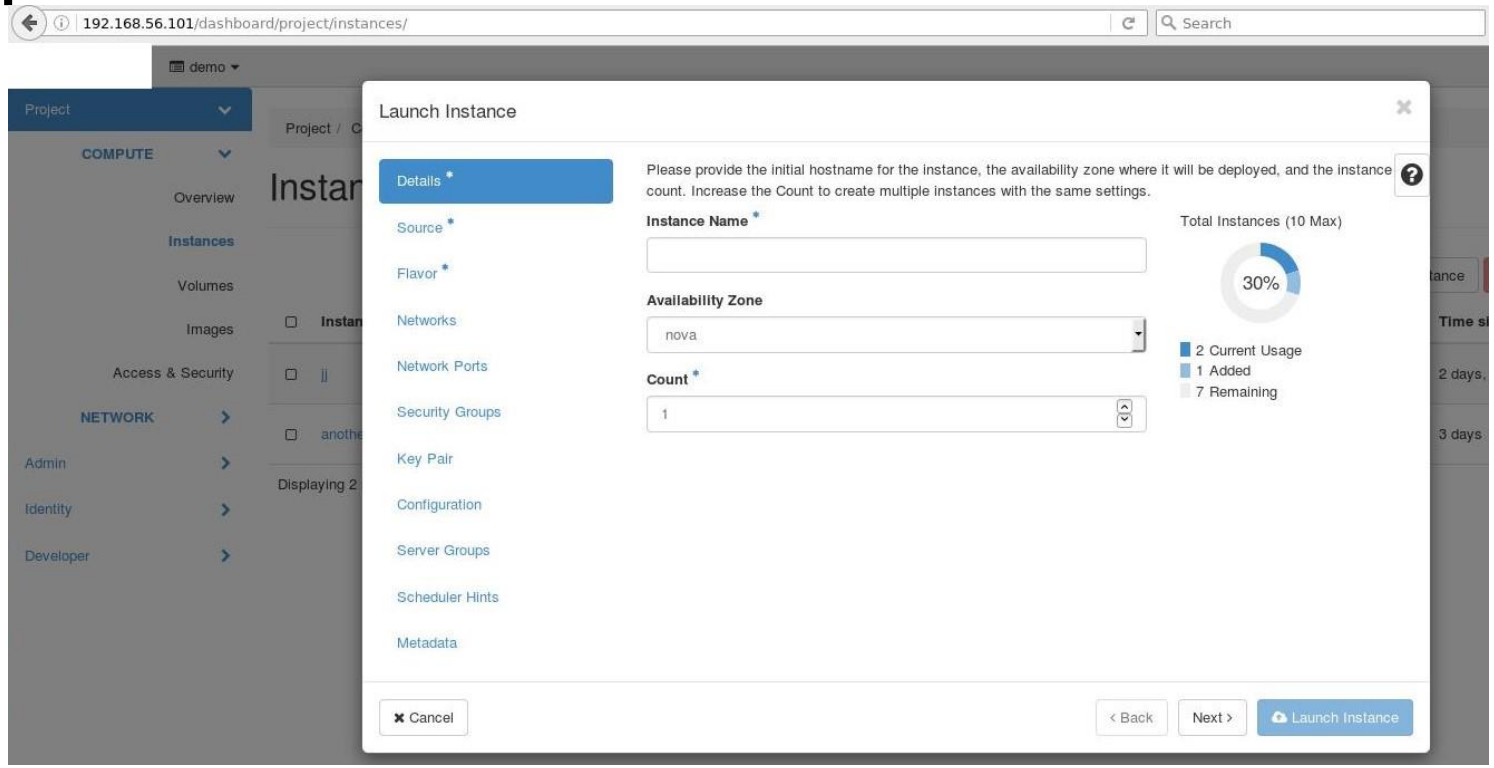
How to use OpenStack Dashboard (cont'd)—Add Rules to Security Group

- Here you have to add rules to the default security group for:
  - Allow SSH
  - All all ICMP

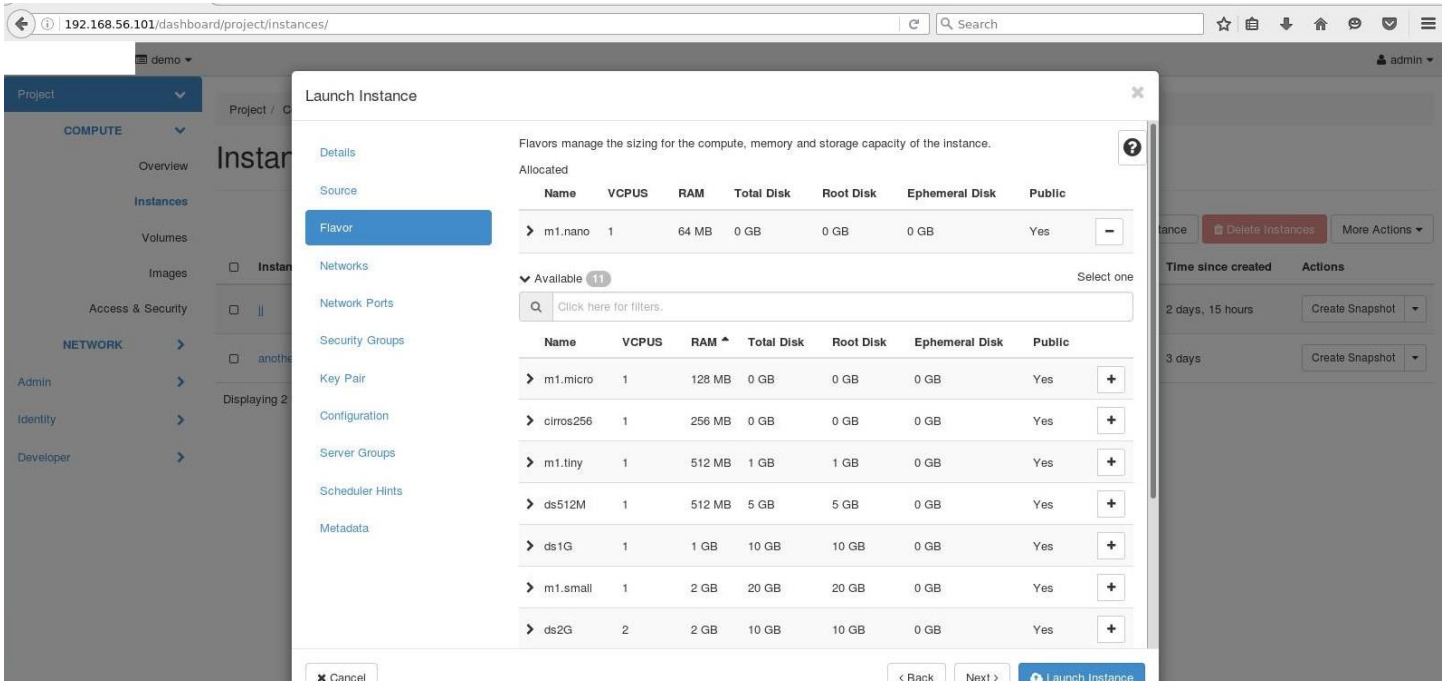
How to use OpenStack Dashboard (cont'd)—Add Rules



How to use OpenStack Dashboard (cont'd)—Launch Instance



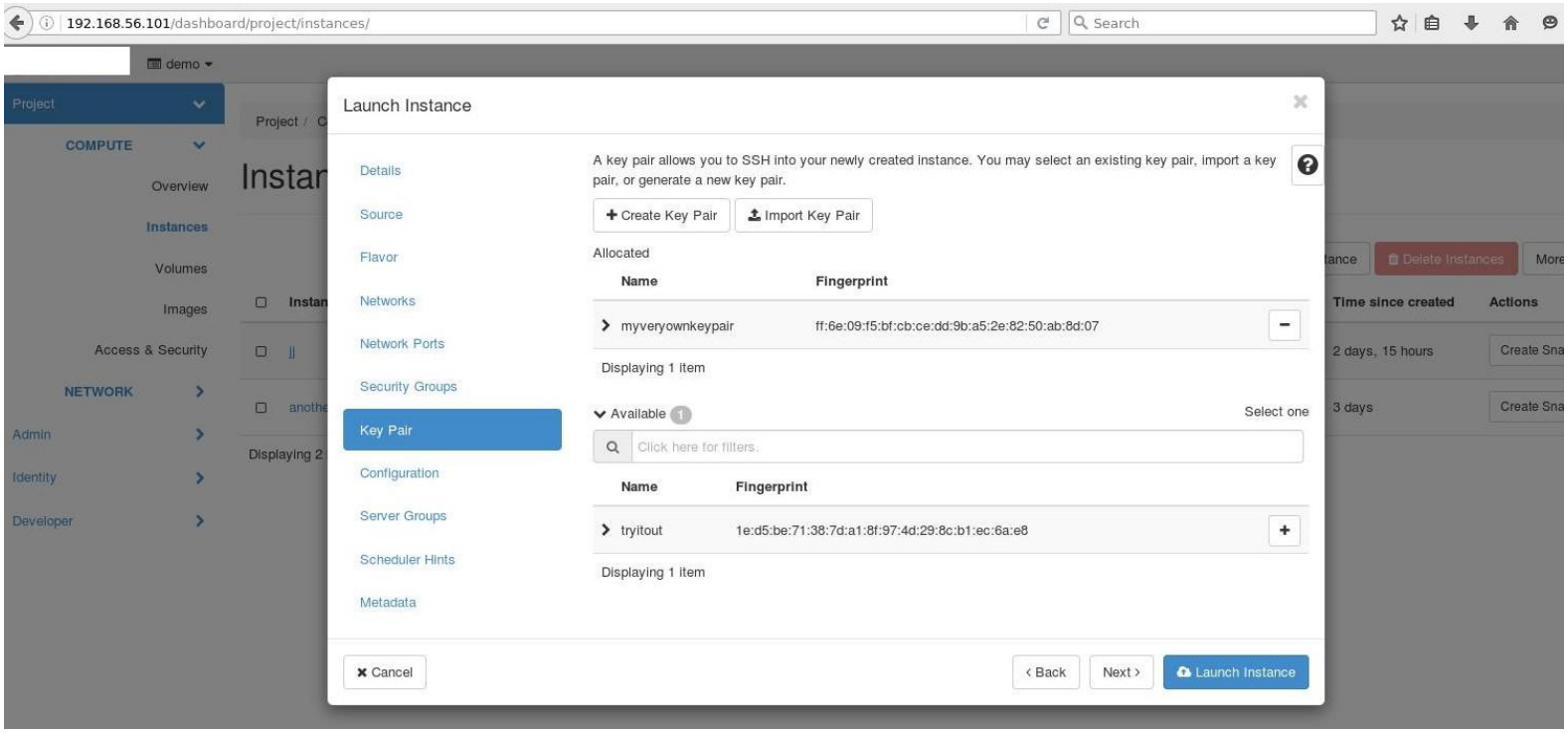
How to use OpenStack Dashboard (cont'd)—Flavors



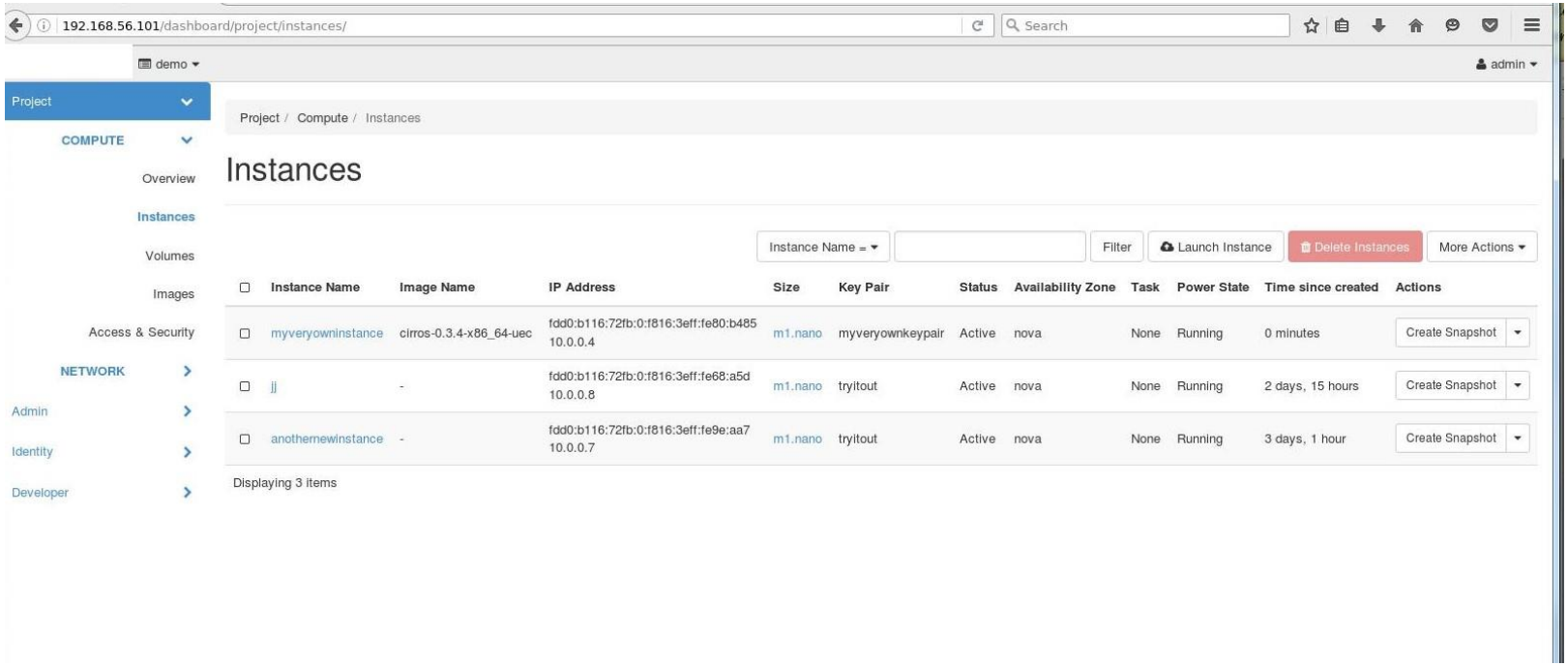
How to use OpenStack Dashboard (cont'd)—Flavors (cont'd)

- The flavor tab provides flavors that correspond to choices about:
  - Number of virtual CPUs
  - How much disk memory (different kinds of disks)
  - How much RAM

How to use OpenStack Dashboard (cont'd)—Launch Instance (cont'd)—Use Keypair



How to use OpenStack Dashboard (cont'd)—Running Instance



How to Log into OpenStack Instance with ssh

- Assume you're located on the Manager node (so the private IP addresses will work). Type:

ssh –i myveryownkeypair.pem cirros@10.0.0.4

Note:

-i means private key is read from the given file

- The first time you do this, ssh will ask you if you want to add the remote host to a list of known hosts. Type "yes"

How to Log into OpenStack Instance with ssh (cont'd)

If your key was already in known_hosts, you will receive something like the following:

```
letzkorn@letzkorn:~/testitout$ ssh -i myveryownkeypair.pem cirros@10.0.0.4
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that a host key has just been changed.
The fingerprint for the RSA key sent by the remote host is
fe:3b:d3:8e:3a:ad:11:1d:0a:a7:af:80:ad:43:bd:fa.
Please contact your system administrator.
Add correct host key in /home/letzkorn/.ssh/known_hosts to get rid of this message.
Offending RSA key in /home/letzkorn/.ssh/known_hosts:1
  remove with: ssh-keygen -f "/home/letzkorn/.ssh/known_hosts" -R 10.0.0.4
RSA host key for 10.0.0.5 has changed and you have requested strict checking.
Host key verification failed.
letzkorn@letzkorn:~/testitout$ rm /home/letzkorn/.ssh/known_hosts
```

How to Log into OpenStack Instance with ssh (cont'd)

- This could be a man in the middle attack, as the warning says.
- However, since you've been playing around starting up instances with different key pairs, the chances are this is just you having changed things.
- As the message says, you can remove the host from the known_hosts file as follows:
  - ssh-keygen -f "/home/letzkorn/.ssh/known_hosts" -R 10.0.0.4
  - Or, alternately, you can just delete the known_hosts file from the .ssh directory using the "rm" command.

How to Log into OpenStack Instance with ssh (cont'd)

letzkorn@letzkorn:~/testitout$ ssh -i myveryownkeypair.pem cirros@10.0.0.4

The authenticity of host '10.0.0.4 (10.0.0.4)' can't be established.

RSA key fingerprint is fe:3b:d3:8e:3a:ad:11:1d:0a:a7:af:80:ad:43:bd:fa.

Are you sure you want to continue connecting (yes/no)? yes

Warning: Permanently added '10.0.0.4' (RSA) to the list of known hosts.

@@@

@   WARNING: UNPROTECTED PRIVATE KEY FILE!   @

@@@

Permissions 0644 for 'myveryownkeypair.pem' are too open.

It is required that your private key files are NOT accessible by others.

This private key will be ignored.

bad permissions: ignore key: myveryownkeypair.pem

cirros@10.0.0.4's password:

How to Log into OpenStack Instance with ssh (cont'd)

- Here it tells you that your permissions are too open for your private key file. This is bad, you don't want other people to be able to read your private key. Do the following:
  - chmod 600 myveryownkeypair.pem
  - This sets permissions on the private key such that only the owner (you) can read it and write it.

How to Log into OpenStack Instance with ssh (cont'd)

If all goes well, you will get something like the following:

```
letzkorn@letzkorn:~/testitout$ chmod 600 *.pem
letzkorn@letzkorn:~/testitout$ ssh -i myveryownkeypair.pem cirros@10.0.0.4
$ pwd
/home/cirros
$ .
```

- Congratulations! You are now logged into your instance!