Building a private cloud with OpenStack
OpenStack

Introduction

- Software for creating private and public clouds (Infrastructure-as-a-Service), released as Open Source (Apache license)
  - Initiated by Rackspace and NASA in 2010
    - NASA contributes compute platform Nebula
    - Rackspace contributes storage platform Cloud Files
  - Today OpenStack Foundation is supported by >500 companies
    - Current Platinum members:
      - AT&T
      - Canonical (Ubuntu Linux)
      - HP Enterprise
      - IBM
      - Intel
      - Rackspace
      - Red Hat
      - SUSE
  - "OpenStack is the Linux of cloud computing."
OpenStack

History

- 2010-07 Rackspace and NASA launch OpenStack Open Source cloud software initiative

- 2010-11 first release, Austin

- 2011-04 Canonical starts to distribute OpenStack in Ubuntu 11.04

- 2012-04 Rackspace starts offering public cloud service based on OpenStack

- 2012-09 OpenStack Foundation is launched as an independent body

- 2012-12 HP Enterprise starts offering public cloud service based on OpenStack

- 2013-03 IBM/Softlayer start offering public cloud service based on OpenStack

- 2013-07 Red Hat starts to offer commercial support for OpenStack

- 2014-09 Oracle starts to distribute OpenStack in Oracle Linux
OpenStack and the competition
OpenStack vs. OpenNebula vs. Eucalyptus vs. CloudStack

Figure 12 -- Monthly Git Contributors

Top 10 contributing orgs for OpenStack in 2015Q1 by email domain

- 9.6% gmail.com
- 7.3% redhat.com
- 5.0% ibm.com
- 4.7% mirantis.com
- 4.6% hp.com
- 1.6% rackspace.com
- 1.4% intel.com
- 1.2% yahoo-inc.com
- 1.1% doughellmann.com
- 0.8% cisco.com

OpenStack

Deployment options

- **Deploy your own private cloud**
  - Many options available from do-it-yourself to remotely operated

- **Use a public cloud**
  - Rackspace Public Cloud (since 2012-04)
  - HP Enterprise (since 2012-12)
  - IBM/Softlayer (since 2013-03)
  - and many others

- Open Source nature of OpenStack avoids vendor lock-in!

**Private cloud deployment options**

- **Do-it-yourself**
  - Plan: Install / configure
  - Operate / maintain

- **Get help with planning**
  - Plan: Install / configure
  - Operate / maintain

- **Let service provider plan and install, then hand over to own sysadmins**
  - Plan: Install / configure
  - Operate / maintain

- **Remotely operated cloud**
  - Plan: Install / configure
  - Operate / maintain

**Responsibilities**

- **Client has responsibility / control**
- **Service provider has responsibility / control**
OpenStack

Users

- Among the most notable users of OpenStack
  - AT&T
  - Bloomberg
  - CERN
  - Cisco Webex
  - Disney
  - Fidelity
  - Swisscom
  - Walmart
OpenStack

Main components

- OpenStack components are developed in their own sub-projects called **programs**
  - **Compute** — Nova
  - **Storage**
    - Object storage — Swift
    - Block storage — Cinder
  - **Networking** — Neutron
  - **Dashboard** — Horizon
  - **Identity** (authentication and authorization) — Keystone
OpenStack
Integrated programs and incubation programs

Integrated programs
- Compute (Nova)
- Networking (Neutron)
- Object store (Swift)
- Block storage (Cinder)
- Image service (Glance)
- Identity (Keystone)
- Dashboard (Horizon)
- Metering service (Ceilometer)
- Orchestration service (Heat)

Core
- Database service (Trove)
- Data processing (Savanna)
- Queue service (Marconi)
- Bare metal (Ironic)

Shared services
OpenStack
Not all projects are created equal

What is your **most** favorite OpenStack project?

- Nova:responses
- Swift:responses
- Heat:responses
- Keystone:responses
- Neutron:responses
- Ironic:responses
- Designate:responses
- Ceilometer:responses
- Trove:responses
- Barbican:responses
- Glance:responses
- Horizon:responses
- Manila:responses
- Oslo:responses
- Sahara:responses
- TripleO:responses
- Zaqar:responses
- Other:responses

What is your **least** favorite OpenStack project?

- Ceilometer:responses
- Neutron:responses
- TripleO:responses
- Cinder:responses
- Horizon:responses
- Oslo:responses
- Glance:responses
- Keystone:responses
- Nova:responses
- Heat:responses
- Ironic:responses
- Sahara:responses
- Swift:responses
- Trove:responses
- Zaqar:responses
- Barbican:responses
- Designate:responses
- Manila:responses
- Other:responses
## OpenStack / Amazon Web Services translation chart

<table>
<thead>
<tr>
<th>Generic term</th>
<th>OpenStack term</th>
<th>AWS term</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compute service</td>
<td>Nova</td>
<td>EC2 (includes compute service)</td>
</tr>
<tr>
<td>Object storage</td>
<td>Swift</td>
<td>S3</td>
</tr>
<tr>
<td>Block storage</td>
<td>Cinder</td>
<td>Elastic Block Store (EBS)</td>
</tr>
<tr>
<td>Image service</td>
<td>Glance</td>
<td>Amazon Machine Image (AMI)</td>
</tr>
<tr>
<td>Virtual machine</td>
<td>Instance</td>
<td>Instance</td>
</tr>
<tr>
<td>Virtual disk</td>
<td>Volume</td>
<td>Volume</td>
</tr>
<tr>
<td>Firewall configuration</td>
<td>Security group</td>
<td>Security group</td>
</tr>
<tr>
<td>Fixed IP address</td>
<td>Floating IP address</td>
<td>Elastic IP address</td>
</tr>
</tbody>
</table>
OpenStack
Compute — Nova

- Manages virtual machines in compute servers
  - Receives requests from users
  - Creates / starts / stops / releases virtual machines
  - Determines on which server to create a VM requested by a user (scheduling)

- Relies on hypervisors to implement the virtual machines
  - Can make use of KVM, Xen, ESX, Hyper-V, …

OpenStack
Block Storage — Cinder

- Service for offering virtual disks (volumes) that can be attached to virtual machines.
  - The operating system in the VM sees a block device and puts a file system on it.

- Functions:
  - Create volume
  - Create volume from snapshot
  - Create volume from VM image
  - Save volume in VM image
  - Attach / detach volume to / from VM
  - Create / delete snapshot

- Can use different storage technologies underneath
  - Local disks
  - SAN (Fiber Channel, iSCSI)
  - NAS (NFS, CIFS)

OpenStack
Object Storage — Swift

- Service for storing “objects”
  - Files without structure
  - Directories are not files

- Distributed architecture
  - Deployed on a cluster of servers
  - Three server roles
    - **Proxy server**: Receives user requests and coordinates storage servers.
    - **Storage servers**: Offer disk space. Three types of storage servers:
      - Account server: Stores user accounts
      - Container server: Stores containers
      - Object server: Stores objects
    - **Consistency server**: Responsible for looking for and finding errors (hardware or software failures) and correcting them.

Source: Creationline http://www.creationline.com/lab/772
OpenStack
Logical deployment topology

Based on: Mirantis – OpenStack Overview and History, http://www.slideshare.net/mirantis/overview-43176920
OpenStack architecture
Beginning (2011)

OpenStack architecture
Two years later (2013)

Source: OpenStack Grizzly logical architecture
OpenStack architecture
Today (2016)

Source: OpenStack Liberty logical architecture, http://docs.openstack.org/admin-guide-cloud/common/get_started_logical_architecture.html
OpenStack
Release schedule

- OpenStack is developed and released around 6-month cycles.
  - Every 6 months a new release
  - Each release gets security updates for only 12 months, then it is declared end-of-life

- Much shorter support timespan than Linux distributions (3 - 10 years). Hard to keep up!

<table>
<thead>
<tr>
<th>Series</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ocata</td>
<td>Future</td>
</tr>
<tr>
<td>Newton</td>
<td>Future</td>
</tr>
<tr>
<td>Mitaka</td>
<td>Under Development</td>
</tr>
<tr>
<td>Liberty</td>
<td>Current stable release, security-supported (EOL: 2016-11-17)</td>
</tr>
<tr>
<td>Kilo</td>
<td>Security-supported (EOL: 2016-05-02)</td>
</tr>
<tr>
<td>Juno</td>
<td>EOL: 2015-12-07</td>
</tr>
<tr>
<td>Icehouse</td>
<td>EOL: 2015-07-02</td>
</tr>
<tr>
<td>Havana</td>
<td>EOL: 2014-09-30</td>
</tr>
<tr>
<td>Grizzly</td>
<td>EOL: 2014-03-29</td>
</tr>
<tr>
<td>Folsom</td>
<td>EOL: 2013-11-19</td>
</tr>
<tr>
<td>Essex</td>
<td>EOL: 2013-05-06</td>
</tr>
<tr>
<td>Diablo</td>
<td>EOL: 2013-05-06</td>
</tr>
<tr>
<td>Cactus</td>
<td>Deprecated</td>
</tr>
<tr>
<td>Bexar</td>
<td>Deprecated</td>
</tr>
<tr>
<td>Austin</td>
<td>Deprecated</td>
</tr>
</tbody>
</table>
OpenStack
Automated installation options

- Red Hat
  - RDO — Install proof of concept on single node, extend to more nodes later
  - TripleO — Deploy a production cloud
  - TryStack — Play with OpenStack in a sandbox. Sandbox is provided as a cloud service

- Ubuntu
  - OpenStack Autopilot — Canonical OpenStack installer for production and test environments
  - BootStack — Installation service by Canonical

- SUSE
  - SUSE OpenStack Cloud Admin

- Mirantis
  - Fuel — Deployment and management tool for OpenStack

- Rackspace
  - Openstack-Ansible — Ansible scripts and extensions for deploying production clouds

- ...
HEIG-Cloud

- Goal: Deploy a private OpenStack cloud at HEIG-VD for research and teaching
  - Want to run clusters for Big Data analysis (Hadoop, Spark)
  - Want to deploy a private PaaS (CloudFoundry)
- Hardware: 13 Dell PowerEdge servers
- Had previously installed Havanna
- In August 2015 started to install Kilo
HEIG-Cloud
Deployment architecture

- Dell PowerEdge
  - 2 CPU sockets w/ 4 cores
  - 96 GB RAM
  - 4 T disk

- Dell PowerEdge
  - 2 CPU sockets w/ 12 cores
  - 256 GB RAM
  - 4 T disk

Cloud total:
- 168 VCPUs
- 944 GB RAM
- 16 T disk

Intranet
Internet

Internal users
External users

Cloud administration

Compute
Compute
Compute
Compute
Compute
Compute
Storage
Storage
Net-working
Controller
Configuration management

Introduction

- When the number of managed machines exceeds a handful, manual configuration becomes too cumbersome.
- Configuration management tools automate the installation and configuration of software.
Configuration management

Tools

- Some popular configuration management tools:
  - CFEngine
    - Started in 1993 by Mark Burgess at Oslo University
    - CFEngine 3 released 2009
  - Puppet
    - Started in 2005 by Luke Kanies
    - Written in Ruby, uses Ruby domain-specific language
  - Chef
    - Started in 2009 by Adam Jacob and people from Amazon
    - Written in Ruby and Erlang, uses Ruby domain-specific language
  - Ansible
    - Started in 2012 by Michael DeHaan
    - Written in Python
Configuration management
Client-server architecture (for example Puppet)

Puppet has a client-server architecture:
Puppet agent = client
Puppetmaster = server

Managed machines must be prepared by installing Puppet agent.
Configuration management
Serverless architecture (Ansible)

Run `ansible-playbook` command
There is no server or database

System administrator

SSH protocol

Server
Ansible script
Load balancer

Server
Ansible script
Web server

Server
Ansible script
Application server

Server
Ansible script
Database

No special software must be installed on managed machines. No agents need to run. Uses existing Python interpreter and SSH.
Installing OpenStack with Ansible

- Unfortunately the installation procedure changes significantly with each new release of OpenStack
- Difficult to re-use Ansible scripts for previous versions
  - Developed new scripts from scratch
<table>
<thead>
<tr>
<th>Instance Name</th>
<th>Image Name</th>
<th>IP Address</th>
<th>Size</th>
<th>Key Pair</th>
<th>Status</th>
<th>Availability Zone</th>
<th>Task</th>
<th>Power State</th>
<th>Time since created</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>spark-hdfs-master</td>
<td></td>
<td>192.168.1.167</td>
<td>m1.spark</td>
<td>bbo_key</td>
<td>Active</td>
<td>nova</td>
<td>None</td>
<td>Running</td>
<td>1 week, 3 days</td>
<td>Create Snapshot</td>
</tr>
<tr>
<td>spark-slave_14</td>
<td></td>
<td>192.168.1.163</td>
<td>m1.spark</td>
<td>bbo_key</td>
<td>Active</td>
<td>nova</td>
<td>None</td>
<td>Running</td>
<td>3 months, 1 week</td>
<td>Create Snapshot</td>
</tr>
<tr>
<td>spark-slave_13</td>
<td></td>
<td>192.168.1.162</td>
<td>m1.spark</td>
<td>bbo_key</td>
<td>Active</td>
<td>nova</td>
<td>None</td>
<td>Running</td>
<td>3 months, 1 week</td>
<td>Create Snapshot</td>
</tr>
<tr>
<td>spark-slave_12</td>
<td></td>
<td>192.168.1.161</td>
<td>m1.spark</td>
<td>bbo_key</td>
<td>Active</td>
<td>nova</td>
<td>None</td>
<td>Running</td>
<td>3 months, 1 week</td>
<td>Create Snapshot</td>
</tr>
<tr>
<td>spark-slave_11</td>
<td></td>
<td>192.168.1.51</td>
<td>m1.spark</td>
<td>bbo_key</td>
<td>Active</td>
<td>nova</td>
<td>None</td>
<td>Running</td>
<td>3 months, 3 weeks</td>
<td>Create Snapshot</td>
</tr>
<tr>
<td>spark-slave_9</td>
<td></td>
<td>192.168.1.49</td>
<td>m1.spark</td>
<td>bbo_key</td>
<td>Active</td>
<td>nova</td>
<td>None</td>
<td>Running</td>
<td>3 months, 3 weeks</td>
<td>Create Snapshot</td>
</tr>
<tr>
<td>spark-slave_8</td>
<td></td>
<td>192.168.1.48</td>
<td>m1.spark</td>
<td>bbo_key</td>
<td>Active</td>
<td>nova</td>
<td>None</td>
<td>Running</td>
<td>3 months, 3 weeks</td>
<td>Create Snapshot</td>
</tr>
<tr>
<td>spark-slave_7</td>
<td></td>
<td>192.168.1.47</td>
<td>m1.spark</td>
<td>bbo_key</td>
<td>Active</td>
<td>nova</td>
<td>None</td>
<td>Running</td>
<td>3 months, 3 weeks</td>
<td>Create Snapshot</td>
</tr>
<tr>
<td>spark-slave_6</td>
<td></td>
<td>192.168.1.46</td>
<td>m1.spark</td>
<td>bbo_key</td>
<td>Active</td>
<td>nova</td>
<td>None</td>
<td>Running</td>
<td>3 months, 3 weeks</td>
<td>Create Snapshot</td>
</tr>
</tbody>
</table>
Multi-domain Keystone

- Keystone performs authentication and authorization of users.
- For authentication the user population can be divided into different domains.
  - Cloud resources in different domains are completely separated from each other.
  - Each domain can be configured differently.
  - Authentication can be delegated to a backend, for example an LDAP server.

- Multiple domains are available in Keystone v3 API
  - Not all OpenStack projects support the v3 API, or they support it only partially!
Multi-domain Keystone

Log In

User Name

Password

Domain

Sign In

Domains

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>heig-vd</td>
<td>HEIG-VD domain.</td>
</tr>
<tr>
<td>Default</td>
<td>Owns users and tenants (i.e. projects) available on Identity API v2.</td>
</tr>
</tbody>
</table>

Displaying 2 items
Troubleshooting tools
Elasticsearch + LogStash + Kibana (ELK)

- **Elasticsearch** — Search server based on Lucene
- **Logstash** — General-purpose log management tool to gather logs from multiple sources, process/parse them to a required format and push them to multiple outputs
- **Kibana** - Data visualization plug-in for Elasticsearch

Monitoring

Ganglia

- Ganglia is a scalable distributed monitoring system for clusters
- Components:
  - **gmond** — Monitoring daemon installed on every server to be monitored
  - **gmetad** — Daemon on the master node that collects data from all the gmond daemons
  - **RDDtool** — (Round-robin database tool) Creates a database with circular buffer
  - **gweb** — Web-based user interface

Source: Matt Massie — Monitoring with Ganglia — O'Reilly Media
Overview of iict_cloud @ 2015-12-21 15:08

- CPUs Total: 192
- Hosts up: 9
- Hosts down: 0

Current Load Avg (15, 5, 1m): 13%, 12%, 12%
Avg Utilization (last hour): 12%

Server Load Distribution

- iict_cloud Cluster Load last hour
- iict_cloud Cluster Memory last hour
- iict_cloud Cluster CPU last hour
- iict_cloud Cluster Network last hour
Controller Load Last Hour

Controller Memory Last Hour

Controller CPU Last Hour

Controller Network Last Hour

Controller graphs [23] last hour sorted by name
Columns: 2 | Size: small
Expand All Metric Groups | Collapse All Metric Groups | Timeshift Overlay | Jump To Metric Group...
Thank you for your time

More information at the HEIG-Cloud blog at http://heig-cloud.github.io/