Private Cloud Computing as in Infrastructure as a Service (IaaS) model using OpenStack

OpenStack/Havana
Prakashan Korambath
Virtual Computing Laboratory*
Institute for Digital Research and Education (IDRE), UCLA
January 29, 2014
ppk@idre.ucla.edu

Havana on
Ubuntu 12.04 LTS/CentOS 6.5

* Funded by UCLA Cyberinfrastructure Grant https://cyber.oit.ucla.edu
Attributes of Cloud

• On-demand self-service - A cloud should enable self-service, so that users can provision servers and networks with little human intervention.
• Network access - Any computing capabilities are available over the network and you can use many different devices through standardized mechanisms.
• Resource pooling - Clouds can serve multiple consumers according to demand.
• Elasticity - Provisioning is rapid and scales out or in based on need.
• Metered or measured service - Just like utilities that are paid for by the hour, clouds should optimize resource use and control it for the level of service or type of servers such as storage or processing.

SaaS, PaaS, IaaS

Examples: Salesforce, Google Docs (SaaS), Google AppEngine, Vmware Cloud Foundry (PaaS), EC2, OpenNebula, OpenStack, Eucalyptus, CloudStack (IaaS)
Gartner Hype Cycle for Emerging Technologies, 2013

Expectations:
- Big Data
- Natural-Language Question Answering
- Internet of Things
- Speech-to-Speech Translation
- Mobile Robots
- 3D Scanners
- Neurobusiness
- Biochips
- Autonomous Vehicles
- Prescriptive Analytics
- Affective Computing
- Electrovibration
- Volumetric and Holographic Displays
- Human Augmentation
- Brain-Computer Interface
- 3D Bioprinting
- Quantum Computing
- Smart Dust
- Bioacoustic Sensing

Peak of Inflated Expectations:
- Consumer 3D Printing
- Gamification
- Wearable User Interfaces
- Complex-Event Processing
- Content Analytics
- In-Memory Database Management Systems
- Virtual Assistants

Plateau of Productivity:
- Predictive Analytics
- Speech Recognition
- Location Intelligence
- Consumer Telematics

As of July 2013

Time:
- Innovation Trigger
- Peak of Inflated Expectations
- Trough of Disillusionment
- Slope of Enlightenment
- Plateau of Productivity

Plateau will be reached in:
- less than 2 years
- 2 to 5 years
- 5 to 10 years
- more than 10 years
- obsolete before plateau
IaaS Software Stack

Data

Applications

Middleware

Guest OS

Hypervisor & Host OS

Hardware

User Controls

Service Provider controls
What is happening during Instance deployment?
(nova-volume has a similar function to EBS)

Initial State

Instance Launching
Network Traffic in the Cloud
nova-network
Smart Manufacturing Project

Calling another Kepler Workflow to Execute Matlab/Octave plus Ansys Fluent Jobs (bottom Workflow)

Job Execution Kepler Workflow on the Controller Node (Called from Above)

Stage I Infrared Image Analysis
Octave/Matlab Execution Script - Stage
SSH Status Log - Stage 1
SSH File Copy to Compute Node
Stage II Reduced Order Model (ROM) Analysis
Octave/Matlab Execution Script - Stage
WF Status - Stage 2
SSH Execution of Octave/Matlab Stage
Boolean Switch
Wait for Stage 1 completion
Status == 0 ? True : False
Stage III Validation by CFD simulation
Ansys Fluent Execution Script - Stage 3
Boolean Switch
Wait for Stage 2 completion
Status == 0 ? True : False
Remote Output File
Local Output File

Variable Input Parameters
- IdentityFile: /Users/ppk/bls/sa_rsa
- cmdfile0: /u/local/apps/kepler-2.4/kepler.sh -runf -requl /home/sim...
- targetHost: sm1c@grizzly.vc.uc.edu
- sourceFile: /Users/ppk/smlc/opstack/inputs/*
- targetFile: sm1c@grizzly.vc.uc.edu/home/smlc/workdir
- outLocalFile: /Users/ppk/smlc/opstack/outputs
- sourceOutFile: sm1c@grizzly.vc.uc.edu/home/smlc/workdir/*

Smart	Manufacturing	Project
OpenStack


• Technology
  – Web
  – REST
  – HTTP
  – Python

• Services
  – Compute (Nova)
  – Object Storage (Swift)
  – Image Service (Glance)
  – Identity (Keystone)
  – Networking (Neutron)
  – Cinder (block storage)
  – Dashboard (Horizon)
  – Orchestration (Heat)
  – Telemetry (Ceilometer)
    • Each service is a webApp
    • REST API server (front end)
    • One or more back end servers
    • Messaging interface between them
  – API’s use HTTP + json (or xml)
    • Use curl or wget or browser plugins
    • Any programming language via HTTP libraries
    • Use the Python novaclient library
Components

- **Keystone**
  - Policy, catalog, token and authentication, backends include LDAP, KVS
- **Dashboard (“Horizon”)**
  - Django, APIs to communicate with OpenStack services
- **Object Storage (Swift)**
- **Glance**
  - Glance-api accept image API calls for image discovery, retrieval and storage, database to storage image metadata
- **Compute (“Nova”)**
  - Nova-api accepts and responds to end user compute API calls
  - Nova-compute Creates virtual machine instances via hypervisors’ API libvirt for KVM, XenAPI for XenServer, Vmware API for Vmware etc.
- **Nova Scheduler**
- **Block Storage (“Cinder”)**
  - Cinder-api accepts requests and routes them to cinder-volume which in turn interacts with cinder-scheduler.
- **Networking (Neutron)**
  - Neutron-server accepts API requests and then routes them to appropriate quantum plugins like Nicira NVP, Cisco, Open vSwitch
- **Ceilometer**
  - A metering project, pricing, billing etc
- **Heat**
  - REST API ot orchestrate multiple cloud applications and implementing standards such as AWS Cloud Formation
Conceptual Architecture

Heat
- Orchestrates cloud

Horizon
- Provides UI

Neutron
- Provides network connectivity for
  - Provides volumes for

Cinder

Nova
- Provisions

Glance
- Provides images
  - Stores images in

Swift

Ceilometer
- Monitors
  - Provides Auth for

Keystone
- Backups volumes in
Concepts

• Users and Projects, Cloud Admins, Project Manager etc. (Role Based Access)
  – access to images is limited by project
  – access/secret are per user
  – keypairs are per user
  – quotas are per project
• Virtualization - Hypervisors
  – KVM
  – Xen
  – HyperV
  – LXC (User space chrooted install)
  – Qemu
• Images and Instances
  – A virtual machine running inside a cloud environment
• System Architecture – Shared Nothing, messaging
  – Cloud Controller
  – Compute Controller
  – Object Store
  – Volume Controller
  – Auth Manager
  – Network Controller
  – Scheduler
• Communication through HTTP GET POST
Resources

Compute Nodes
- 16 IBM iDataPlex 2U units – Intel Nehalem – E5530
  - 2 sockets per node
  - 4 cores per socket
  - 8 cores per node
  - 24 GB memory per node
- Total 256 cores and 768 GB memory

Controller Node
- AMD Opteron
- Six Core Opteron 2431
- 12 cores in total and 32 GB memory

Network Node
Network Switches = 2

Swift Object Storage = 120 TB triplicated.

- Auxiliary nodes for NAT and OpenVPN, GIT, Software License Managers
- High Availability ??
Architecture

• **The Cloud Controller**
  – also run portions of the Compute service such as the API server, the scheduler, conductor, console authenticator, and VNC service.
  – hosts the API endpoint for the Network service.

• **The Network Controller**
  – Network services such as DHCP, layer 2 switching, layer 3 routing, floating Ips and metadata connectivity.

• **Compute Node**
  – Compute service as well as the OpenStack Network service agent (in this case, the Open vSwitch plugin agent). This server also manages an OpenStack-compatible hypervisor such as KVM or Xen. This server will host the actual virtual machines (instances).
**Software-defined networking using Neutron**

- **neutron-server** => implements REST API
- **neutron-*plugin-agent** => connect instances to network port
- **neutron-dhcp-agent** => start/stop dhcp server
- **neutron-l3-agent** => implement floating IPs, NAT
Network configuration

- Controller – 2 NICS 192.168.0.10, 10.10.10.10
- Network – 3 NICS 192.168.0.9, 10.10.10.9, 10.0.0.9
- Compute – 2 NICS 10.10.10.11, 10.0.0.11
- API network 192.168.0.0/24 (reachable on the internet)
- Management network 10.10.10.0/24
- Data Network 10.0.0.0/24

Network Node IP configuration

- eth0 10.10.10.9, eth1 10.0.0.9, br-ex 192.168.0.9, eth2 in promisc mode
- ovs-vsctl add-br br-ex
- ovs-vsctl add-port br-ex eth2
VM booting workflow

- nova boot calls compute driver -> compute driver calls neutron API to create a port id
- neutron-server creates the port and allocates an IP address from the subnet -> notifies neutron-dhcp-agent with port id
- neutron-dhcp-agent configures the port id with IP, MAC address, gateway and router
- compute-driver-gets the network information -> creates a port on br-int software switch and boots the VM with a TAP device attached on the soft-switch port
- soft-neutron-agent detects the port, set up the port and network flow so that VM can get the address
- VM gets the IP address using dhcp
OpenVswitch Network configuration

Scenario 1: one tenant, two networks, one router

Has two private networks (net01, and net02), each with one subnet (net01_subnet01: 192.168.101.0/24, net02_subnet01, 192.168.102.0/24). Both private networks are attached to a router that connects them to the public network (10.64.201.0/24).
OpenVswitch Network configuration **Scenario 2: two tenants, two networks, two routers**
Neutron Networking with GRE tunneling
Linux network namespaces

Namespaces are many independent networks (routers, interfaces iptables) in the same linux machine. They sit on top of global namespace

Tenant-1 (namespace-1, net interface, routing table, iptables),
Tenant-1 (namespace-2, net interface, routing table, iptables),
.
.
Tenant-n (namespace-n, net interface, routing table, iptables),

ifconfig

ifconfig <interface> up/down
ifconfig <interface> <ip> netmask <netmask>
netstat -rn
route add -net <net> netmask <netmask> gw <gateway>

ip netns exec <namesapce id> commands (Eg: ping <ip>, route –n etc.)
TAP (simulates link layer) devices, TUN (network TUNnel simulates a network layer) veth pairs (directly connected to virtual network), Linux bridges (behaves like a hub), Open vSwitch bridges (behave like a virtual switch).
br-int OpenVSwitch bridge is the integration bridge:
ovs-ofctl dump-flows br-tun
ip netns show
ip netns exec qrouter-d8b2ef39-90c3-42cf-b893-253e936cd194 route
ip netns exec qrouter-d8b2ef39-90c3-42cf-b893-253e936cd194 ifconfig
ip netns exec qrouter-85e8bd92-ff1e-49f8-89a8-7ed3808f21ee iptables -L -n -t nat
ip netns exec qrouter-d8b2ef39-90c3-42cf-b893-253e936cd194 ip addr list
ip netns exec qrouter-d8b2ef39-90c3-42cf-b893-253e936cd194 ping <fixedip>
Network on the compute node
Network on the network node
OpenVswitch/OpenFlow commands

ovs-ofctl show br-int
ovs-ofctl show br-tun
ovs-ofctl dump-flows br-tun
ovs-ofctl dump-ports br-tun
ovs-dpctl show
ovs-dpctl dump-flows ovs-system
ovs-vsctl show
ovs-vsctl list-br
ovs-vsctl list-ports br-ex
ovs-vsctl list-ports br-int
ovsdb-client list-tables
Neutron - CLI

• Create a new tenant and User
  – keystone tenant-create --name project_one  
  – keystone user-create --name=user_one --pass=user_one --tenant-id <id_of_project_one>  
    --email=user_one@domain.com  
  – keystone user-role-add --tenant-id <id_of_project_one> --user-id <id_of_user_one> --role-id <id_of_member_role>

• Public Network:
  – neutron subnet-create ext-net --allocation-pool start=192.168.0.180,end=192.168.0.250 --gateway=192.168.0.1 --enable_dhcp=False 192.168.0.0/24

• Create a new network for the tenant:
  – neutron net-create --tenant-id <id_of_project_one> net_proj_one  
  – neutron subnet-create --tenant-id <id_of_project_one> --name subnet_proj_one net_proj_one 50.50.1.0/24  
  – neutron router-create --tenant-id <put_id_of_project_one> router_proj_one  
  – neutron router-gateway-set <id of router_proj_one> <id of ext-net>  
  – neutron router-interface-add <id of net_proj_one> <id of subnet_proj_one>
Neutron CLI

• Floating IP
  ▪ neutron net-list -- --router:external=True
  ▪ nova list
  ▪ neutron floatingip-create ext_net --port_id $VM_PORT_ID
  ▪ neutron floatingip-list
  ▪ neutron floatingip-dissassociate <floatingip_id>
  ▪ neutron floatingip-associate <floating_ip_id> $VM_PORT_ID
cirros@192.168.0.189's password: 

ppk@control:~$ nova list
+-------------------------------------+---------+-----------------+-----------------+---------------------+---------------------+
| ID | Name | Status | Task State | Power State | Networks |
+-------------------------------------+---------+-----------------+-----------------+---------------------+---------------------+
| 5829f28e-e638-4aee-a68c-6f281d5c6fcd | insttest | ACTIVE | None | Running | net_dmo pjct=10.6.6.2 |
+-------------------------------------+---------+-----------------+-----------------+---------------------+---------------------+

ppk@control:~$ neutron net-list --router:external True
+-------------------------------------+---------+---------+----------+
| id | name | subnets |          |
+-------------------------------------+---------+---------+----------+
| 155759c9-0aa1-481a-8d67-4bbf1a4720ff | ext-net | c55a8dd1-3695-4948-92a0-261525b73dc5 |          |
+-------------------------------------+---------+---------+----------+

ppk@control:~$ neutron port-list --device_id 5829f28e-e638-4aee-a68c-6f281d5c6fcd
+-------------------------------------+---------+---------+----------+
| id | name | mac_address | fixed_ips |
+-------------------------------------+---------+---------+----------+
| 3bc0078d-82fa-4601-8b21-9f60b40f993f | fa:16:3e:ea:8e:72 | {"subnet_id": "7972aa48-adfd-4042-88dd-2256a832b981", "ip_address": "10.6.6.2"} | |
+-------------------------------------+---------+---------+----------+

ppk@control:~$ neutron floatingip-create --port_id 3bc0078d-82fa-4601-8b21-9f60b40f993f 155759c9-0aa1-481a-8d67-4bbf1a4720ff
Created a new floatingip:
+-------------------------------------+---------+---------+----------+
| Field | Value |          |
+-------------------------------------+---------+---------+----------+
| floating_ip_address | 10.6.6.2 |          |
| floating_ip_address | 192.168.0.189 | |
+-------------------------------------+---------+---------+----------+

ppk@control:~$ nova list
+-------------------------------------+---------+-----------------+-----------------+---------------------+---------------------+
| ID | Name | Status | Task State | Power State | Networks |
+-------------------------------------+---------+-----------------+-----------------+---------------------+---------------------+
| 5829f28e-e638-4aee-a68c-6f281d5c6fcd | insttest | ACTIVE | None | Running | net_dmo pjct=10.6.6.2, 192.168.0.189 |
+-------------------------------------+---------+-----------------+-----------------+---------------------+---------------------+

ppk@control:~$ ssh 192.168.0.189 -l cirros
cirros@192.168.0.189's password:
$
Underlying Technologies

Circles are Linux services that are part of OpenStack compute.
Rectangles are external.

All Compute services interact with RabbitMQ and MySQL.

All network components interact through the Linux networking stack.
Installing from DevStack for a Single Node Testing

• Follow the instructions at http://devstack.org/ for Ubuntu 12.04
• sudo apt-get install git
• git clone https://github.com/openstack-dev/devstack.git
• cd devstack; ./stack.sh
• Answer few questions:
  ENTER A PASSWORD TO USE FOR RABBIT.
  ENTER A SERVICE_TOKEN TO USE FOR THE SERVICE ADMIN TOKEN.
  ENTER A SERVICE_PASSWORD TO USE FOR THE SERVICE AUTHENTICATION.
  ENTER A PASSWORD TO USE FOR HORIZON AND KEYSTONE (20 CHARs OR LESS).

• Default Account is ‘admin’ and the project is ‘demo’. Use the Horizon/Keystone password to login.
• https://sites.google.com/site/hpccloud/devstack_installation
Cloud Provider Conceptual Architecture
Hardware for Building a IaaS Cloud Platform
Hardware for Building a IaaS Cloud Platform
Hardware for Building a IaaS Cloud Platform III
For the Developers

• Paste HTTP server
  – HTTP protocol + networking
    • (http://pythonpaste.org)

• WebOb requests and responses
  – Wrappers for HTTP requests and responses
    • http://www.webob.org

• OpenStack code for Nova, Glance Keystone etc.
  • http://www.openstack.org

• Web Service Gateway Interface (WSGI)
  – http://www.python.org/dev/peps/pep-3333/
  – http://www.wsgi.org

• RESTful Web Services
  – Book by L. Richardson and S. Ruby
Demo

- [https://sites.google.com/site/hpccloud/](https://sites.google.com/site/hpccloud/)
- GUI and CLI
Overview

Limit Summary

- Instances: Used 1 of 50
- VCPUs: Used 1 of 20
- RAM: Used 512.0 MB of 50.0 GB
- Floating IPs: Used 4 of 50
- Security Groups: Used 1 of 10

Select a period of time to query its usage:

From: 2014-01-01  To: 2014-01-27  Submit  The date should be in YYYY-mm-dd format.

Active Instances: 1  Active RAM: 512MB  This Period's VCPU-Hours: 1.71  This Period's GB-Hours: 1.71

Usage Summary

<table>
<thead>
<tr>
<th>Instance Name</th>
<th>VCPUs</th>
<th>Disk</th>
<th>RAM</th>
<th>Uptime</th>
</tr>
</thead>
<tbody>
<tr>
<td>instest</td>
<td>1</td>
<td>1</td>
<td>512MB</td>
<td>1 hour, 42 minutes</td>
</tr>
</tbody>
</table>

Displaying 1 item
Network Topology

Mozilla Firefox is free and open source software from the non-profit Mozilla Foundation.