TOOLS FOR CLOUD COMPUTING

Eucalyptus, OpenNebula and Tashi

Gustavo Ansaldi Oliva
goliva@ime.usp.br
golivax@gmail.com

Agenda

- Basic Concepts
- Eucalyptus
- OpenNebula
- Tashi
- Tools Comparison
- Extra: Amazon EC2 API

Basic Concepts

- Virtualization
  - Hypervisor
    - A.K.A Virtual Machine Monitor (VMM)
    - Allows multiple operating systems to run concurrently on a host computer— a feature called hardware virtualization
    - The hypervisor presents the guest OSs with a virtual platform and monitors the execution of the guest OSs
    - Type 1/native/baremetal: Xen
    - Type 2/hosted: VMWare Server, Sun VirtualBox

Basic Concepts

- Virtualization
  - Hardware-assisted virtualization
    - Processor instruction set extensions that provide hardware assistance to virtual machines
    - These extensions address the parts of x86 that are difficult or inefficient to virtualize, providing additional support to the hypervisor.
    - Enables simpler virtualization code and a higher performance for full virtualization.
    - Intel VT-x, AMD-V

Basic Concepts

- Virtualization
  - KVM
    - Kernel-based Virtual Machine
    - Full virtualization solution for Linux on x86 hardware
    - Requires processor with Intel VT or AMD-V
    - KVM does not perform any emulation by itself.
    - User-space program uses the /dev/kvm interface to set up the guest VM's address space, feeds it simulated I/O and maps its video display back onto the host's
    - KVM uses QEMU for its device emulation

Basic Concepts

- Clouds
  - Public/External Cloud
    - Commercial cloud providers that offer a publicly-accessible remote interface to create and manage virtual machine instances within their proprietary infrastructure
  - Private/Internal Cloud
    - Provides local users with a flexible and agile private infrastructure to run service workloads within their administrative domain
  - Hybrid Cloud
    - Supplements local infrastructure with computing capacity from an external public cloud

Eucalyptus

- Owner
  - Eucalyptus Systems
- License
  - GNU GPL v3
- Latest Version
  - 1.6.2 (February, 2010)

Eucalyptus

- Eucalyptus implements what is commonly referred to as Infrastructure as a Service (IaaS)
  - (Virtualization + Storage + Network) as a Service
- Eucalyptus is an open-source software infrastructure for the implementation of cloud computing on computer clusters which provides an interface that is compatible with the Amazon EC2 service
- Eucalyptus is not fault-tolerant
  - http://open.eucalyptus.com/forum/eucalyptus-specs-scalability-fault-tolerance-and-slightl

Eucalyptus

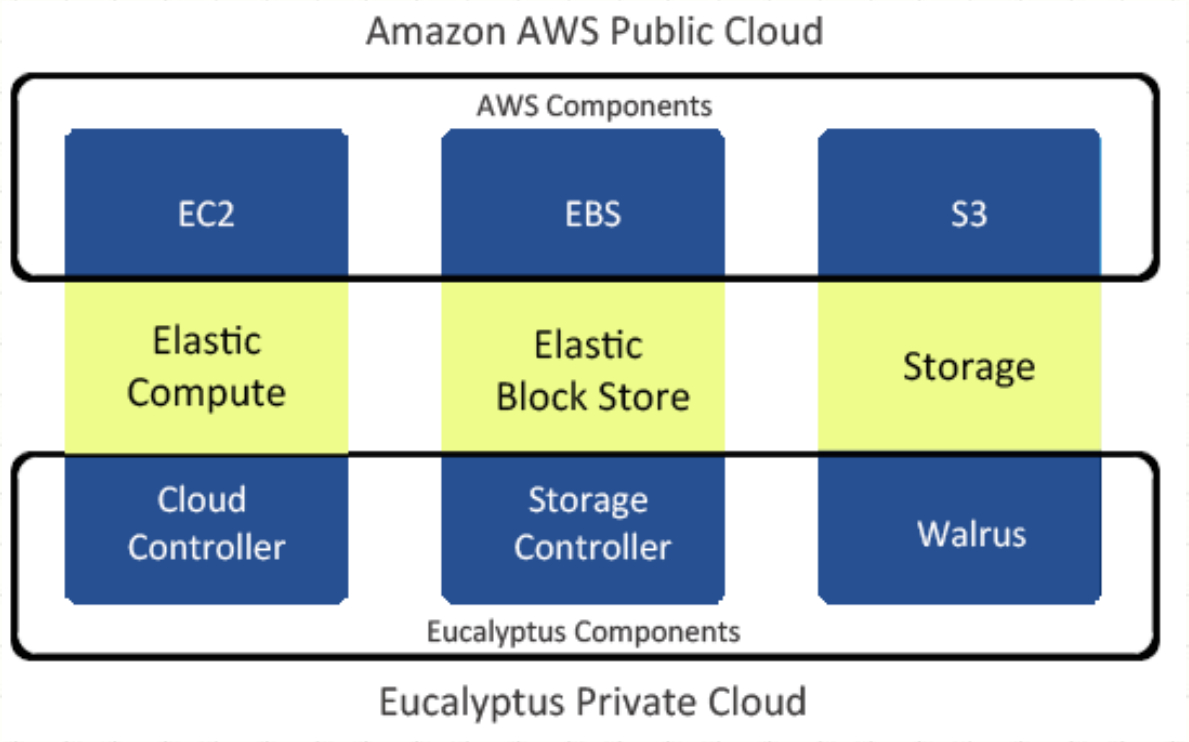
- Eucalyptus works with most of the currently available Linux distributions
- Eucalyptus can use a variety of Virtualization technologies including VMware, Xen and KVM
- Eucalyptus offers a community cloud (ECC)
  - Sandbox environment in which community members can testdrive and experiment with Eucalyptus

Eucalyptus

- Main features
  - Compatibility with Amazon Web Services API
  - Installation and deployment from source or DEB and RPM packages.
  - Secure communication between internal processes via SOAP and WS-Security
  - Basic administration tools
  - Able to configure multiple clusters as a single cloud
  - Added support for elastic IP assignment
  - Configurable scheduling policies and SLAs

Eucalyptus

- Hybrid Cloud with Eucalyptus



Eucalyptus API

- Eucalyptus uses Euca2ools
- Euca2ools are command-line tools for interacting with Web services that export a REST/Query-based API compatible with Amazon EC2 and S3 services.
- Euca2ools can be used with both Amazon's services and with installations of the Eucalyptus open-source cloud-computing infrastructure.
- Euca2ools were inspired by command-line tools distributed by Amazon (api-tools and ami-tools) and largely accept the same options and environment variables.

Euca2ools

- Query of availability zones (i.e. clusters in Eucalyptus)
- SSH key management (add, list, delete)
- VM management (start, list, stop, reboot, get console output)
- Security group management
- Volume and snapshot management (attach, list, detach, create, bundle, delete)
- Image management (bundle, upload, register, list, deregister)
- IP address management (allocate, associate, list, release)

Euca2ools

```
view plain  copy to clipboard  print  ?
01.  cd .euca
02.  source eucarc
03.  euca-describe-availability-zones verbose
```

```
AVAILABILITYZONE rosh-cluster1 192.168.0.114
AVAILABILITYZONE |- vm types free / max cpu ram disk
AVAILABILITYZONE |- m1.small 0004 / 0004 1 128 2
AVAILABILITYZONE |- c1.medium 0004 / 0004 1 256 5
AVAILABILITYZONE |- m1.large 0002 / 0002 2 512 10
AVAILABILITYZONE |- m1.xlarge 0000 / 0000 2 1024 20
AVAILABILITYZONE |- c1.xlarge 0000 / 0000 4 2048 20
```

Euca2ools

euca-add-group	euca-unbundle
euca_conf	euca-attach-volume
euca-delete-volume	euca-delete-bundle
euca-describe-keypairs	euca-describe-groups
euca_killall	euca-detach-volume
euca-run-instances	euca-register
euca-add-keypair	euca-upload-bundle
euca-confirm-product-instance	euca-authorize
euca-deregister	euca-delete-group
euca-describe-regions	euca-describe-image-attribute
eucalyptus-cloud	euca-disassociate-address
euca_sync_key	euca-release-address
euca-allocate-address	euca-version
euca-create-snapshot	euca-bundle-image
euca-describe-addresses	euca-delete-keypair
euca-describe-snapshots	euca-describe-images
euca-modify-image-attribute	euca-download-bundle
euca-terminate-instances	euca-reset-image-attribute
euca-associate-address	euca-bundle-vol
euca-create-volume	euca-delete-snapshot
euca-describe-availability-zones	euca-describe-instances
euca-describe-volumes	euca-get-console-output
euca-reboot-instances	euca-revoke

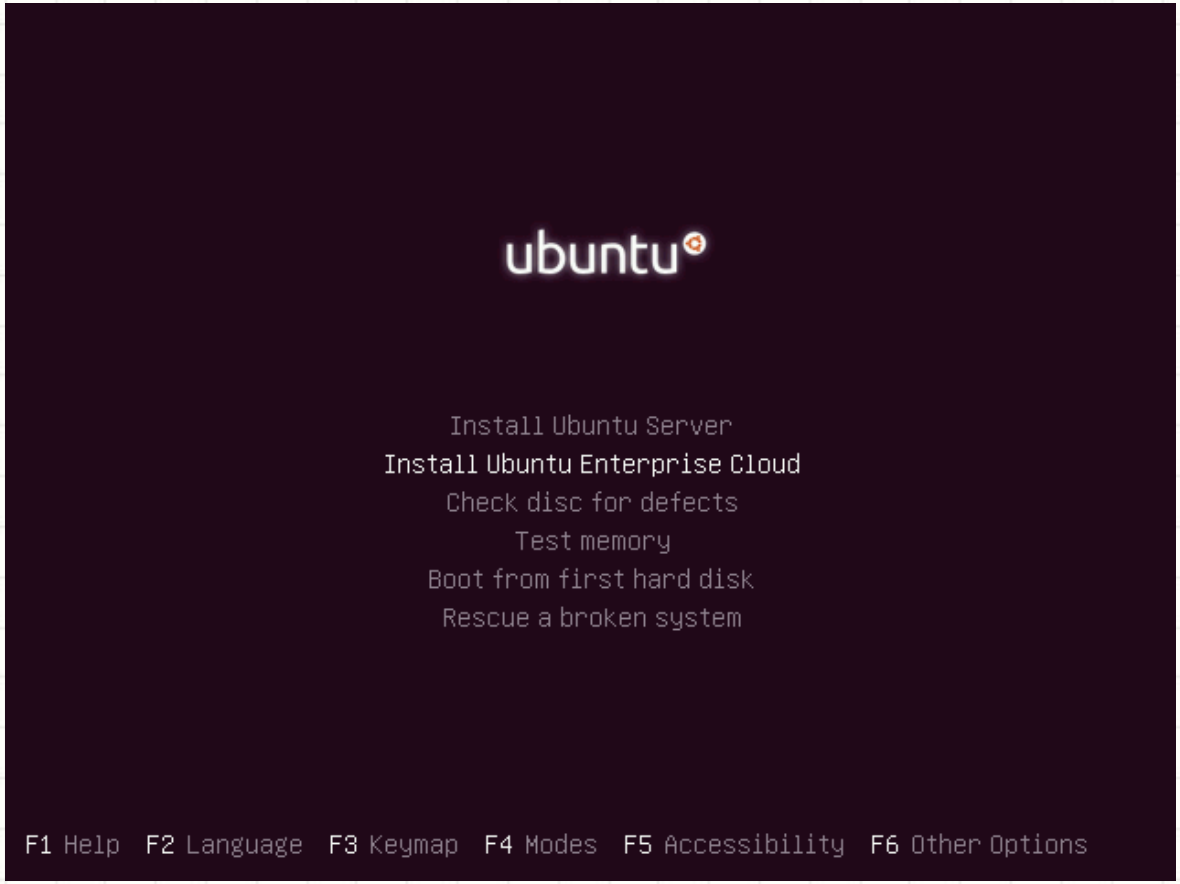
Eucalyptus - UEC

- Core element of the Ubuntu Enterprise Cloud (UEC) cloud computing module
  - Included in Ubuntu Server Edition
  - Wizards to build private and public clouds
  - http://www.ubuntu.com/cloud/why-ubuntu

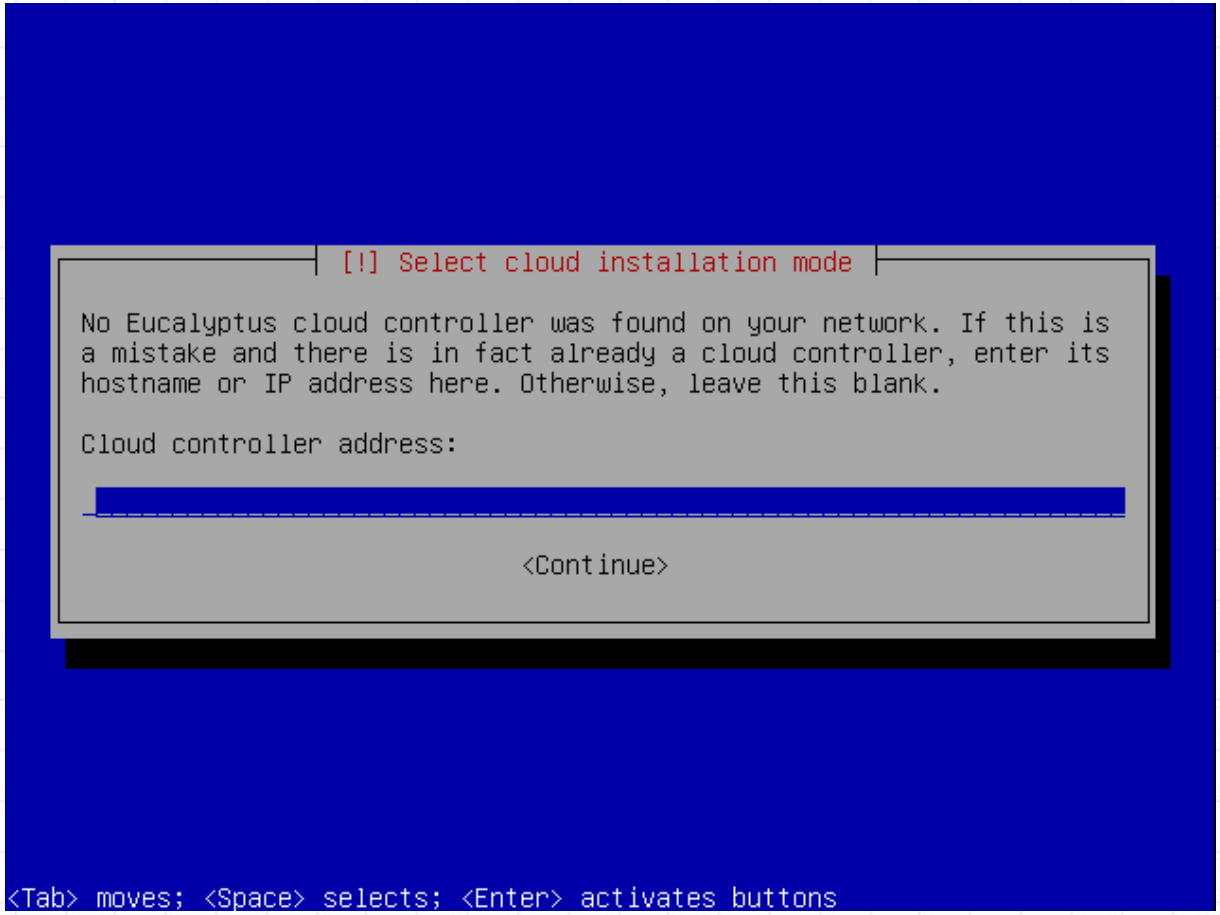
Setting Up Private Cloud in UEC

- All-in-one controller
- One or more working nodes (run VM instances)
- 1) Prerequisites
  - Front end
    - Cloud controller (clc)
    - Cluster controller (cc)
    - Walrus (the S3-like storage service)
    - The storage controller (sc)
  - One or more nodes
    - Node controller (nc)
- 2) Install the Cloud/Cluster/Storage/Walrus Front End Server

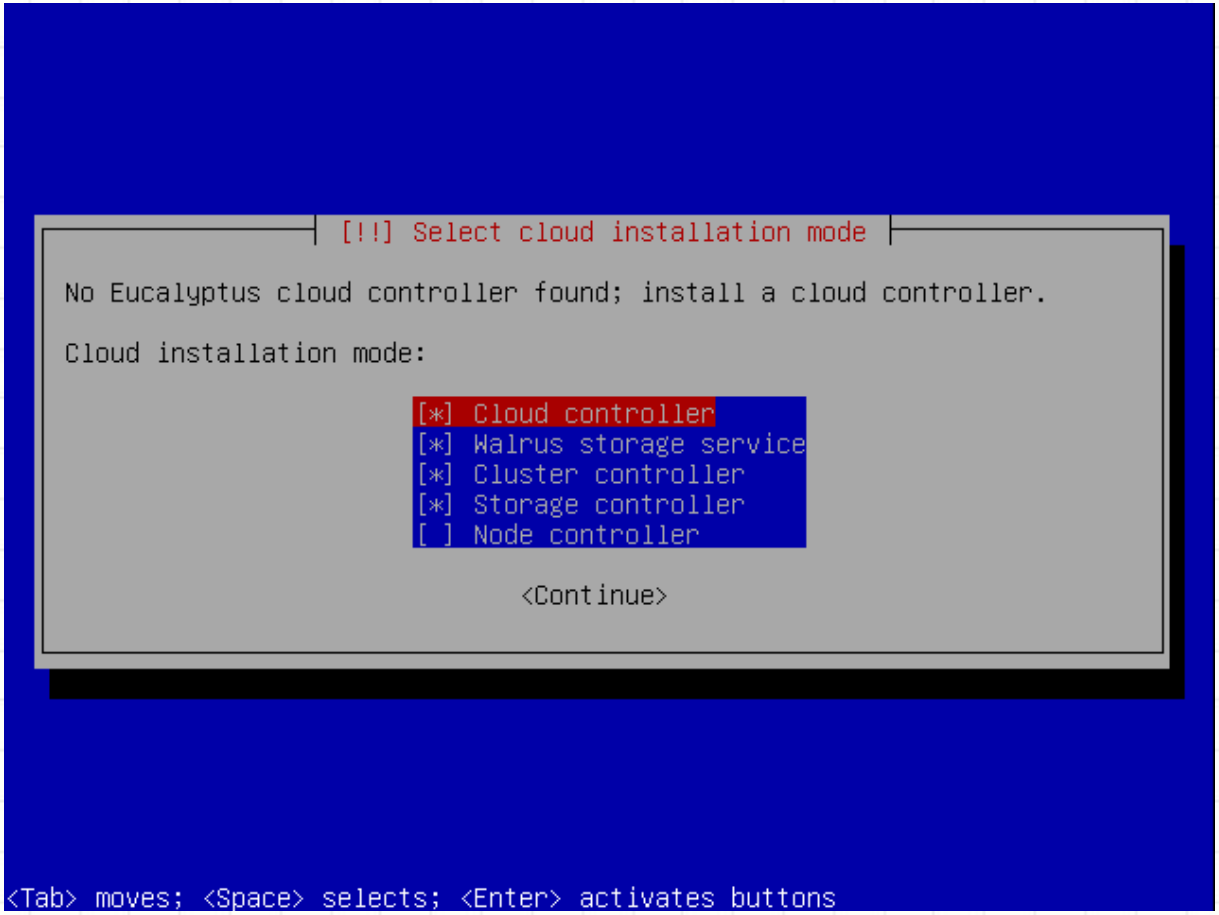
Setting Up Private Cloud in UEC



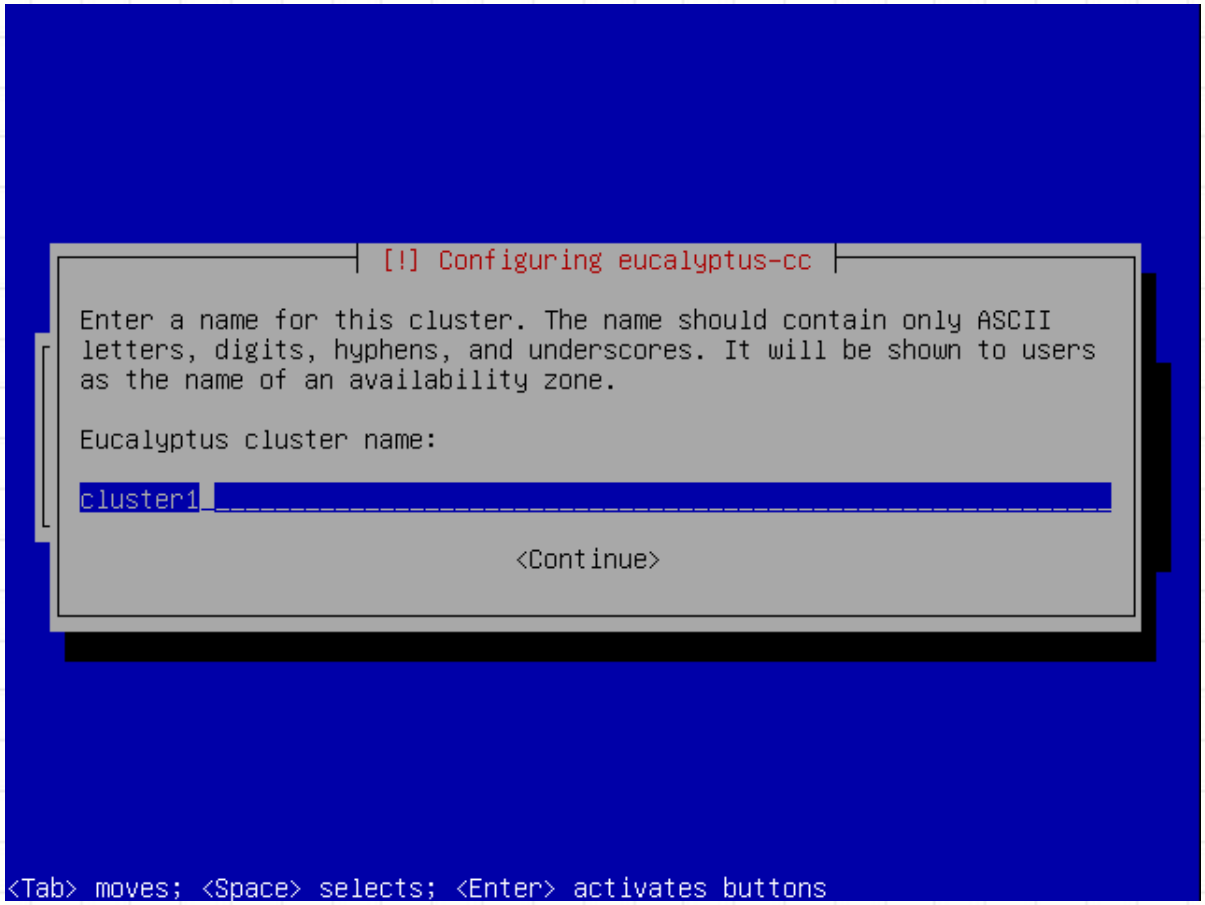
Setting Up Private Cloud in UEC



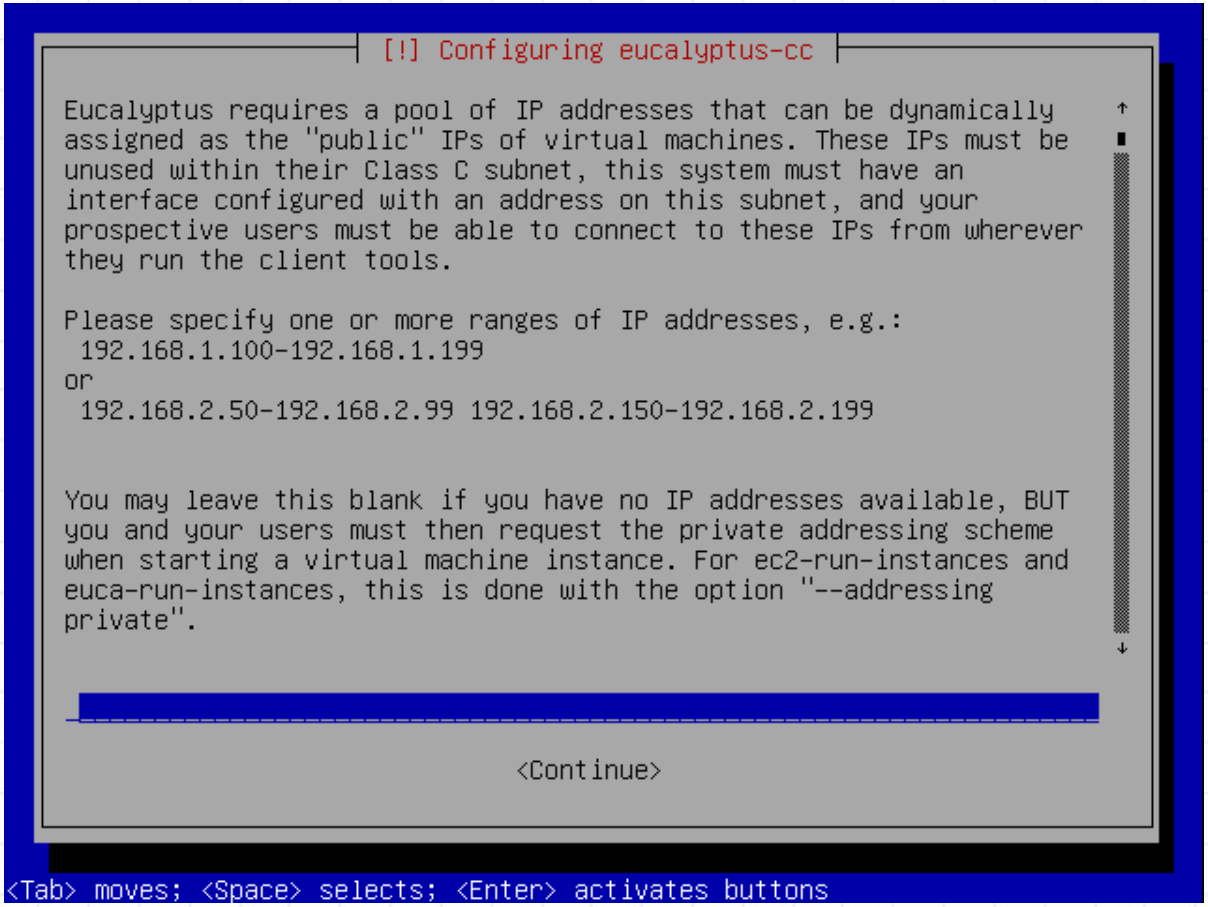
Setting Up Private Cloud in UEC



Setting Up Private Cloud in UEC



Setting Up Private Cloud in UEC



Setting Up Private Cloud in UEC

- 3) Install the Node Controller
  - Make sure that the node is connected to the network on which the cloud/cluster controller is already running
  - Boot from the same ISO on the node(s)
  - Select “Install Ubuntu Enterprise Cloud”
  - It should detect the Cluster and preselect “Node” install for you
  - Confirm the partitioning scheme
  - The rest of the installation should proceed uninterrupted; complete the installation and reboot the node

Setting Up Private Cloud in UEC

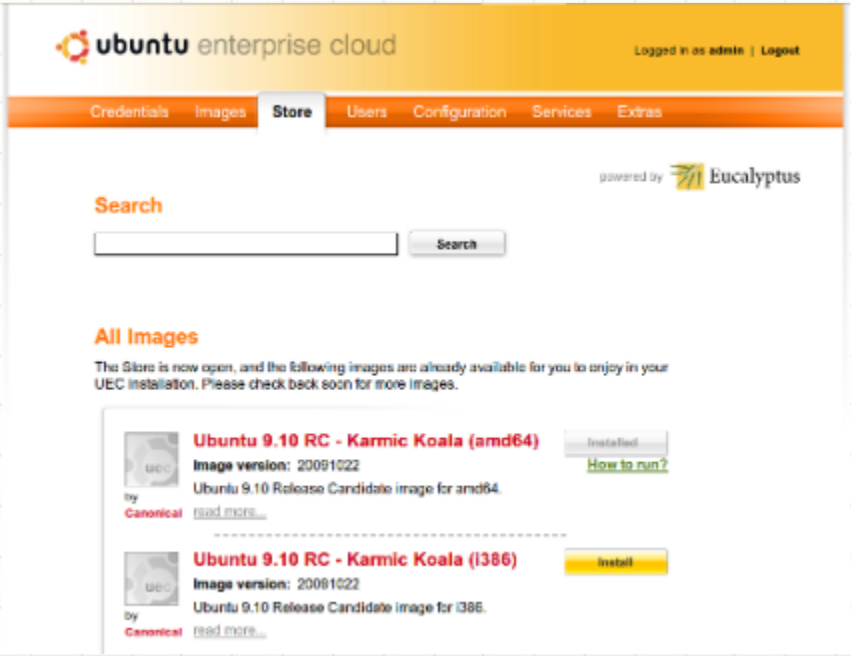
- 4) Register the nodes
  - In Ubuntu 10.04 all registration is automatic with the UEC CD Install
    - Public SSH keys have been exchanged properly
    - The services are configured properly
    - The services are publishing their existence
    - The appropriate uec-component-listener is running
  - Refer to the following link when doing a package install
    - https://help.ubuntu.com/community/UEC/CDInstall

Setting Up Private Cloud in UEC

- 5) Obtain credentials
  - After installing and booting the Cloud Controller, users of the cloud will need to retrieve their *credentials*
  - This can be done either through a web browser, or at the command line
    - https://<cloud-controller-ip-address>:8443/
    - Use username 'admin' and password 'admin' for the first time login

Setting Up Private Cloud in UEC

- 6) Install an image from the store
  - Image Store on the UEC web interface



Setting Up Private Cloud in UEC

- 7) Run an Image
  - There are multiple ways to instantiate an image in UEC:
    - Use the command line
    - Use one of the UEC compatible management tools such as Landscape
    - Use the ElasticFox extension to Firefox
  - More info on https://help.ubuntu.com/community/UEC/CDInstall

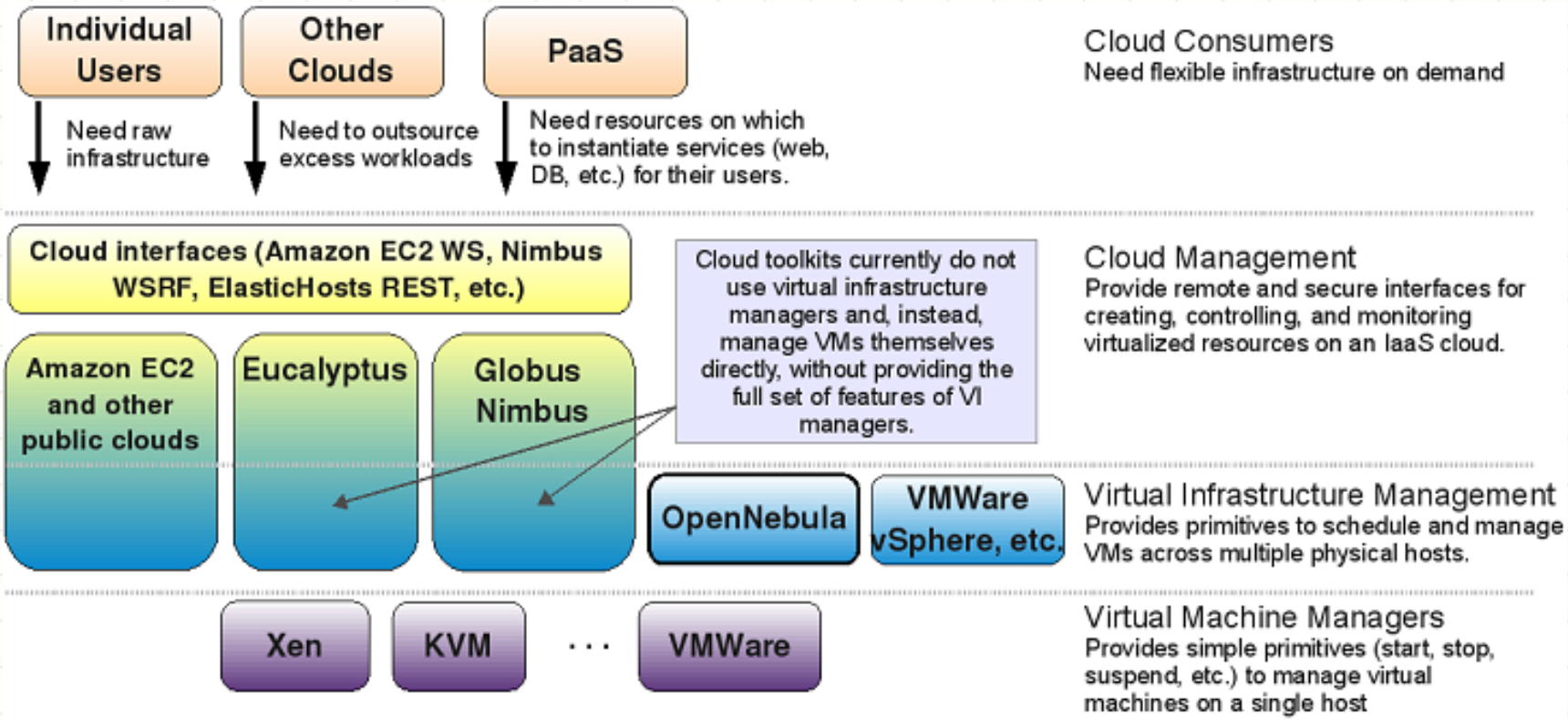
OpenNebula

- Main Contributors
  - DSA (Distributed Systems Architecture) Research at UCM (Universidad Complutense de Madrid)
  - http://dsa-research.org/doku.php
- License
  - Apache License v2 (2004)
- Latest Version
  - 1.4 (December, 2009)

OpenNebula

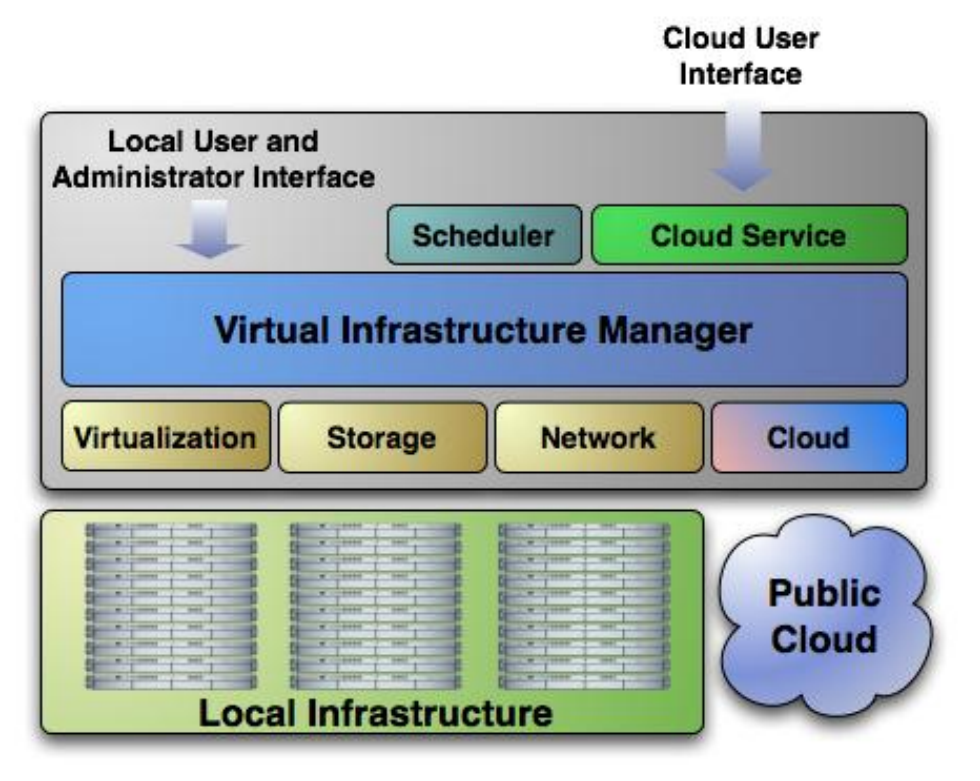
- OpenNebula is an open-source toolkit to easily build any type of cloud
- OpenNebula has been designed to be integrated with any networking and storage solution and so to fit into any existing data center.
  - Private cloud with Xen, KVM and VMWare
  - Hybrid cloud with Amazon EC2 and ElasticHosts
  - Public cloud supporting EC2 Query, OGF OCCI and vCloud APIs
- Apache License v2 (2004)

OpenNebula


Individual Users
Other Clouds
PaaS
Need raw infrastructure
Need to outsource excess workloads
Need resources on which to instantiate services (web, DB, etc.) for their users.
Cloud Consumers
Need flexible infrastructure on demand
Cloud interfaces (Amazon EC2 WS, Nimbus WSRF, ElasticHosts REST, etc.)
Amazon EC2 and other public clouds
Eucalyptus
Globus Nimbus
Cloud toolkits currently do not use virtual infrastructure managers and, instead, manage VMs themselves directly, without providing the full set of features of VI managers.
Cloud Management
Provide remote and secure interfaces for creating, controlling, and monitoring virtualized resources on an IaaS cloud.
OpenNebula
VMWare vSphere, etc.
Virtual Infrastructure Management
Provides primitives to schedule and manage VMs across multiple physical hosts.
Xen
KVM
. . .
VMWare
Virtual Machine Managers
Provides simple primitives (start, stop, suspend, etc.) to manage virtual machines on a single host

OpenNebula

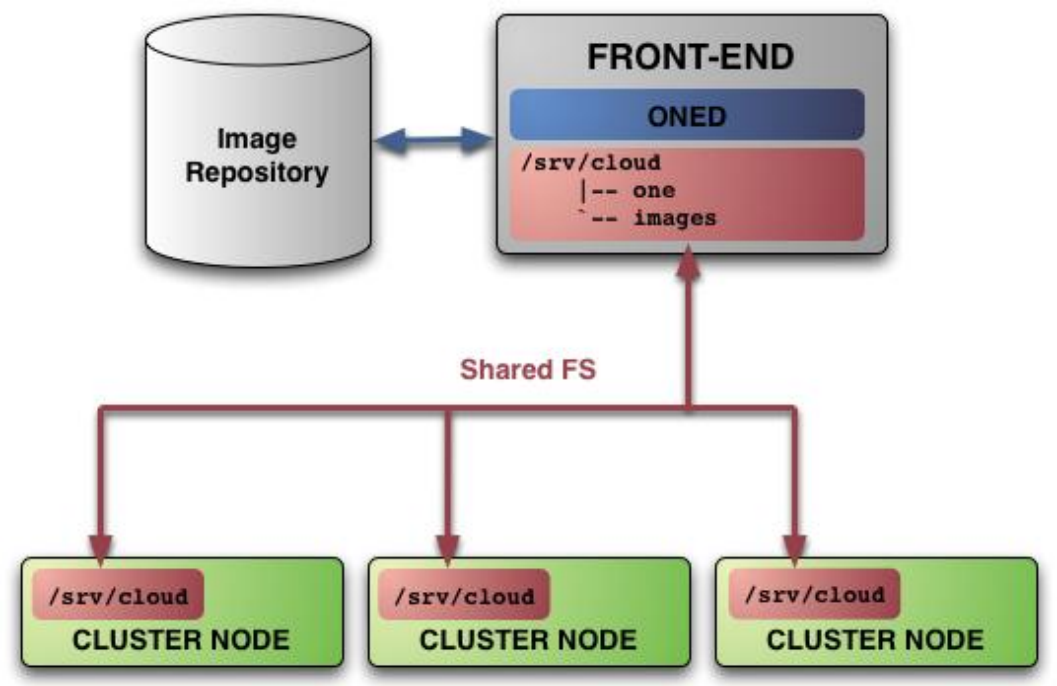
- Focus on the Virtual Infrastructure Management



OpenNebula – Fault Tolerance

- Fault tolerance is managed by a persistent back-end database (sqlite3), that stores the hosts and their VM information
- All virtual machine images are stored in a Image Repository accessible from the OpenNebula front-end
- When a VM start request is made, the VM is copied (cloned) to the cluster node which will then run the VM. When the image is closed, its copied back to the front-end

OpenNebula – Fault Tolerance



- /srv/cloud/one → hold the OpenNebula installation and the clones for the running VMs
- /srv/cloud/images → will hold the master images and the repository
- http://www.opennebula.org/documentation:rel1.4:plan#storage

OpenNebula – EC2 Query API

- Implements a subset of the EC2 Query interface, enabling the creation of public clouds managed by OpenNebula
  - **upload image**: Uploads an image to the repository manager
  - **register image**: Registers an image (previously uploaded in the repository manager) in order to be launched
  - **describe images**: Lists all registered images belonging to one particular user.
  - **run instances**: Runs an instance of a particular image (that needs to be referenced)
  - **terminate instances**: Shutdown a virtual machine(or cancel, depending on its state)

OpenNebula – EC2 Query API

- Starting with a working installation of an OS residing on an **.img** file, with three steps a user can launch it in the cloud
- 1) Upload image to the cloud image repository

```
$ ./econe-upload /images/gentoo.img
Success: ImageId 872ce740-5904-012c-08e0-0017f231be96
```

- 2) Register the image to be used in the cloud

```
$ ./econe-register 872ce740-5904-012c-08e0-0017f231be96
Success: ImageId 872ce740-5904-012c-08e0-0017f231be96
```

- 3) Launch the registered image to be run in the cloud

```
$ ./econe-run-instances -H 872ce740-5904-012c-08e0-0017f231be96
Owner       ImageId                                 InstanceId InstanceType
-------------------------------------------------------------------------
helen       872ce740-5904-012c-08e0-0017f231be96    15         m1.small
```

OpenNebula – EC2 Query API

- Additionally, the instance can be **monitored** with:

```
$ ./econe-describe-instances  -H
Owner       Id    ImageId                                 State         IP             Type
-------------------------------------------------------------------------------------------
helen       15    872ce740-5904-012c-08e0-0017f231be96  pending       147.96.80.33    m1.small
```

OpenNebula – OCCI API

- The OpenNebula OCCI API is a RESTful service to create, control and monitor cloud resources based on the latest draftof the OGF OCCI API specification
- OCCI stands for “Open Cloud Computing Interface”
- Resources are: Storage, Network, Compute

OpenNebula – OCCI API

- Storage:
  - Upload: using a multi-part HTTP POST.
  - Retrieve: using a HTTP GET.
- Network:
  - Upload: using a HTTP POST.
  - Retrieve: using a HTTP GET.
- Compute:
  - Upload: using a HTTP POST.
  - Update: using a HTTP PUT.
  - Retrieve: using a HTTP GET.

OpenNebula – OCCI API

- Upload an image

```
<DISK>
    <NAME>GentooImage</NAME>
    <URL>file:////images/gentoo.img</URL>
</DISK>
```

- $ ./occi-storage --url http://localhost:4567 --username oneadmin --password opennebula create imagexml

  <DISK><ID>ab5c9770-7ade-012c-f1d5-00254bd6f386</ID><NAME>GentooImage</NAME><SIZE>1000</SIZE><URL>file://images/gentoo.img<URL></DISK>

OpenNebula – OCCI API

- Create a network resource

```
<NETWORK>
        <NAME>MyServiceNetwork</NAME>
        <ADDRESS>192.168.1.1</ADDRESS>
        <SIZE>200</SIZE>
</NETWORK>
```

- $ ./occi-network --url http://localhost:4567 --username oneadmin --password opennebula create vnxml

  <NIC><ID>23</ID><NAME>MyServiceNetwork</NAME><ADDRESS>192.168.1.1</ADDRESS><SIZE>200</SIZE></NIC>

OpenNebula – OCCI API

- Create a compute resource (VM)

```
<COMPUTE>
        <NAME>MyCompute</NAME>
        <STORAGE>
                <SWAP size="1024" dev="sda2"/>
                <DISK image="ab5c9770-7ade-012c-f1d5-00254bd6f386" dev="sda1"/>
                <FS size="512" format="ext3" dev="sda3"/>
        </STORAGE>
        <NETWORK>
                <NIC network="23" ip="192.168.0.9"/>
        </NETWORK>
        <INSTANCE_TYPE>small</INSTANCE_TYPE>
</COMPUTE>
```

OpenNebula – OCCI API

- $ ./occi-compute --url http://localhost:4567 --username tinova --password opennebula create ~/vmxml

  <COMPUTE><ID>17</ID><NAME>MyCompute</NAME><STATE>PENDING</STATE><STORAGE><DISK image="ab5c9770-7ade-012c-f1d5-00254bd6f386" dev="sda1"/><FS size="512" format="ext3" dev="sda3"/><SWAP size="1024" dev="sda2"/></STORAGE><NETWORK><NIC network="23" ip="192.168.0.9"/><NIC network="23" ip="192.168.1.1"/></NETWORK></COMPUTE>

Setting Up Private Cloud with OpenNebula and Ubuntu

- Six steps
  - 1) Front-end installation
  - 2) Add the cluster nodes to the system
  - 3) Configure ssh access
  - 4) Install the nodes → Done on working nodes
  - 5) Setting authorization → Done on working nodes
  - 6) Prepare Virtual Network and Launch VMs
- Please refer to https://help.ubuntu.com/community/OpenNebula

Tashi

- Key Contributors
  - Intel Labs Pittsburgh
  - Carnegie Mellon University
- License
  - Apache License v2 (2004)
- Latest Version
  - No formal release (code must be checked out from SVN)

Tashi

- An infrastructure through which service providers are able to build applications that harness cluster computing resources to efficiently access repositories of “Big Data”
  - Cluster management system for cloud computing on big data
- Project incubated at Apache
  - Non-trivial installation and setup

Tashi

- Tashi is primarily a system for managing Virtual Machines (VMs)
  - Request the creation, destruction, and manipulation of VMs (Xen, KVM/QEMU)
- Tashi also has a web client interface that is compatible with a subset of Amazon Web Services
- Boot 100 copies of an operating system in 2 minutes
- Tashi is a key software component that enables the Big Data cluster to participate in the HP OpenCirrus cluster testbed as a Center of Excellence

Tashi

- Main features
  - Provide high-performance execution over Big Data repositories

    Enable multiple services to access a repository concurrently
  - Enable low-latency scaling of services
  - Enable each service to leverage its own software stack
    - Virtualization, file-system protections
  - Enable slow resource scaling for growth
  - Enable rapid resource scaling for power/demand
    - Scaling-aware storage

Tashi – Example Applications

Application	Big Data	Algorithms	Compute Style
Video search	Video data	Object/gesture identification, face recognition, …	MapReduce
Internet library search	Historic web snapshots	Data mining	MapReduce
Virtual world analysis	Virtual world database	Data mining	*To be defined*
Earth study	Ground model	Earthquake simulation, thermal conduction, …	HPC (high performance computing)
Language translation	Text corpuses, audio archives,…	Speech recognition, machine translation, text-to-speech, …	MapReduce & HPC

Tashi – Organization

- Each cluster contains one **Tashi Cluster Manager (CM)**
- The CM maintains a database of:
  - Available physical resources (nodes)
  - Active virtual machines
  - Pending requests for virtual machines
  - Virtual networks
- Users submit requests to the CM through a **Tashi Client**
- The **Tashi Scheduler** uses the CM databases to invoke actions, such as VM creation, through the CM
- Each node contains a **Node Manager** that carries out actions, such as invoking the local **Virtual Machine Manager (VMM)**, to create a new VM, and monitoring the performance of VMs

Tashi – Native API

- Users invoke Tashi actions through a Tashi client
- VM Life-cycle
  - createVm [--userId <value>] --name <value> [--cores <value>] [--memory <value>] --disks <value> [--nics <value>] [--hints <value>]
  - destroyVm --instance <value>
  - shutdownVm --instance <value>

Tashi – Native API

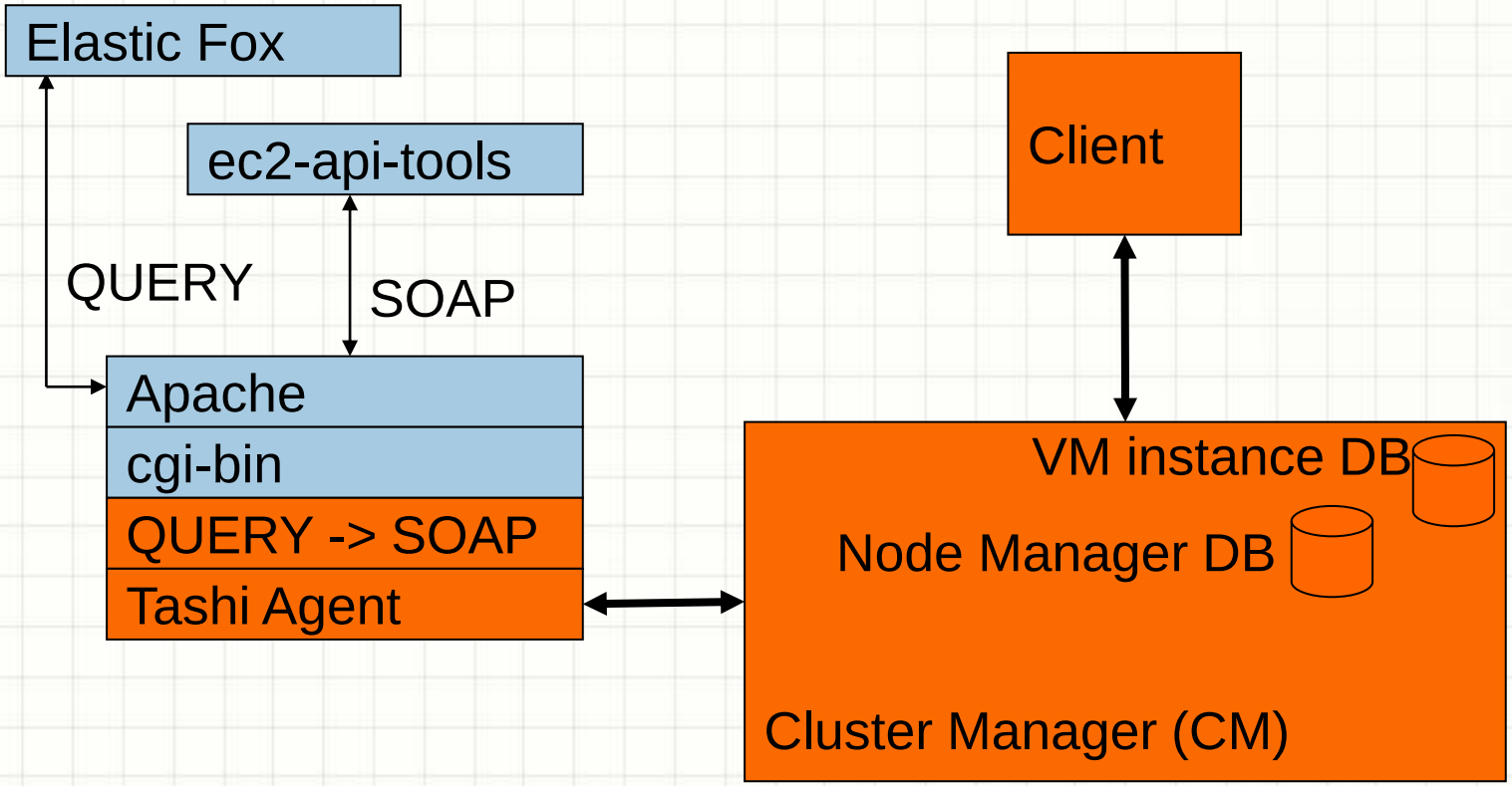
- VM Management Calls
  - suspendVm --instance <value>
  - resumeVm --instance <value>
  - pauseVm --instance <value>
  - unpauseVm --instance <value>
  - migrateVm --instance <value> --targetHostId <value>
  - vmmSpecificCall --instance <value> --arg <value>

Tashi – Native API

- “Bookkeeping” Calls
  - getMyInstances
  - getInstances
  - getVmLayout
  - getUsers
  - getNetworks
  - getHosts

Tashi – Using EC2 API Tools

Tashi is compatible with a subset of Amazon Web Services



Tashi

- More information
  - Presentation: www.pittsburgh.intel-research.net/~rgass/projects/sc09/SC09-Tashi-CR.ppt
    - Includes architecture diagrams of Tashi
  - Tashi on a single test machine
    - http://incubator.apache.org/tashi/documentation-single.html

Tools Comparison

- Report done by Universiteit van Amsterdam
- Compares Cloud Computing Solutions and offers guidance according to specific scenarios
- http://staff.science.uva.nl/~delaat/sne-2009-2010/p31/report.pdf

Eucalyptus vs OpenNebula

	Eucalyptus 1.6.1	OpenNebula 1.4
Deployment Model		
Private cloud	yes	yes
Offload to private (same platform)[1]	no	yes
Offload to private (other platforms)	no	Eucalyptus & UEC
Offload to public[2]	yes	yes
API[6]	EC2	EC2 & OGC OCCI
Availability		
Redundant Front-End	no[3]	no
Fault tolerance/fail-over[4]	no	yes
Features		
Availability Zones	yes	yes
(Auto) resource scaling	Manual	Dynamic[5]
Live migration	no	only with Shared FS (SAN/-NAS)
Management		
User Management	web interface	cli 3rd party GUIs
Monitoring	support for 3rd party	support for 3rd party
Licenses and Standards		
open source	yes	yes
Enterprise	yes	no
Open Virtualization Format (OVF)	no	recommended[8]
Product Potential		
Marketshare	+	+
Future development	++	++

Extra: Amazon EC2 API

- http://docs.amazonwebservices.com/AWSEC2/latest/APIReference/
  - API Reference guide with usage examples
    - EC2 Query API
    - EC2 SOAP API

Amazon EC2 API – Query API

Example Request

This example starts the i-10a64379 instance.

```
https://ec2.amazonaws.com/?Action=StartInstances&InstanceId.1=i-10a64379"&AuthParams
```

Example Response

```
<StartInstancesResponse xmlns="http://ec2.amazonaws.com/doc/2009-11-30/">
  <instancesSet>
    <item>
      <instanceId>i-10a64379</instanceId>
      <currentState>
          <code>0</code>
          <name>pending</name>
      </currentState>
      <previousState>
          <code>80</code>
          <name>stopped</name>
      </previousState>
    </item>
  </instancesSet>
  </StartInstancesResponse>
```

Amazon EC2 API – SOAP API

Example Request

This example starts the i-10a64379 instance.

```
<StartInstances xmlns="http://ec2.amazonaws.com/doc/2009-11-30/">
  <instancesSet>
    <item>
                <instanceId>i-10a64379</instanceId>
    </item>
  </instancesSet>
  </StartInstances>
```

Example Response

```
<StartInstancesResponse xmlns="http://ec2.amazonaws.com/doc/2009-11-30/">
  <instancesSet>
    <item>
      <instanceId>i-10a64379</instanceId>
      <currentState>
          <code>0</code>
          <name>pending</name>
      </currentState>
      <previousState>
          <code>80</code>
          <name>stopped</name>
      </previousState>
    </item>
  </instancesSet>
  </StartInstancesResponse>
```

References

- Eucalyptus
  - http://open.eucalyptus.com/
- Eucalyptus and Fault-Tolerance
  - http://open.eucalyptus.com/forum/eucalyptus-specs-scalability-fault-tolerance-and-slightl
- Euca2ools User Guide
  - http://open.eucalyptus.com/wiki/Euca2oolsGuide_v1.1
- Euca2ools Blog Entry
  - http://blogs.plexibus.com/2010/06/17/eucalyptus-euca2ools/
- Ubuntu Enterprise Cloud
  - http://www.ubuntu.com/cloud/why-ubuntu

References

- OpenNebula
  - http://www.opennebula.org/
  - http://dsa-research.org/doku.php
- OpenNebula and Fault-Tolerance
  - http://www.opennebula.org/documentation:rel1.4:plan#storage
- Private Cloud with OpenNebula and Ubuntu
  - https://help.ubuntu.com/community/OpenNebula
- An Open Source Solution for Virtual Infrastructure Management in Private and Hybrid Clouds – IEEE Special Issue on Cloud Computing
  - http://www.mcs.anl.gov/uploads/cels/papers/P1649.pdf
- OGF OCCI Implementation on OpenNebula
  - http://62.149.240.97/uploads/Tutorials/ogf_occi_implementation_on_opennebula.pdf

References

- Tashi
  - http://incubator.apache.org/tashi/index.html
  - http://www.pittsburgh.intel-research.net/projects/tashi/
- Tashi Presentation by Michael Ryan (Intel)
  - www.pittsburgh.intel-research.net/~rgass/projects/sc09/SC09-Tashi-CR.ppt
- SURFnet Cloud Computing Solutions
  - http://staff.science.uva.nl/~delaat/sne-2009-2010/p31/report.pdf

References

- Amazon EC2 API Reference
  - http://docs.amazonwebservices.com/AWSEC2/latest/APIReference/
- Amazon EC2 API Tools
  - http://developer.amazonwebservices.com/connect/entry.jspa?externalID=351&categoryID=88
- Elasticfox Firefox Extension for Amazon EC2
  - http://developer.amazonwebservices.com/connect/entry.jspa?externalID=609&categoryID=88

That’s all

Questions?