

Contrail Summer School

June 2011, France

Building Clouds with OpenNebula 2.2 and StratusLab

Javier Fontán Muiños

dsa-research.org | OpenNebula.org

Distributed Systems Architecture Research Group
Universidad Complutense de Madrid



StratusLab Project

Grid aware cloud stack



- MarketPlace with pregenerated Grid images
- Claudia service manager
- User friendly CLI to manage VMs and create new images
- Prepackaged software for common Linux distributions
- Quattor recipes to easily install and configure the software in a cluster
- X509/Grid credentials support
- VM management based on OpenNebula 2.2

StratusLab Project

Grid aware cloud stack

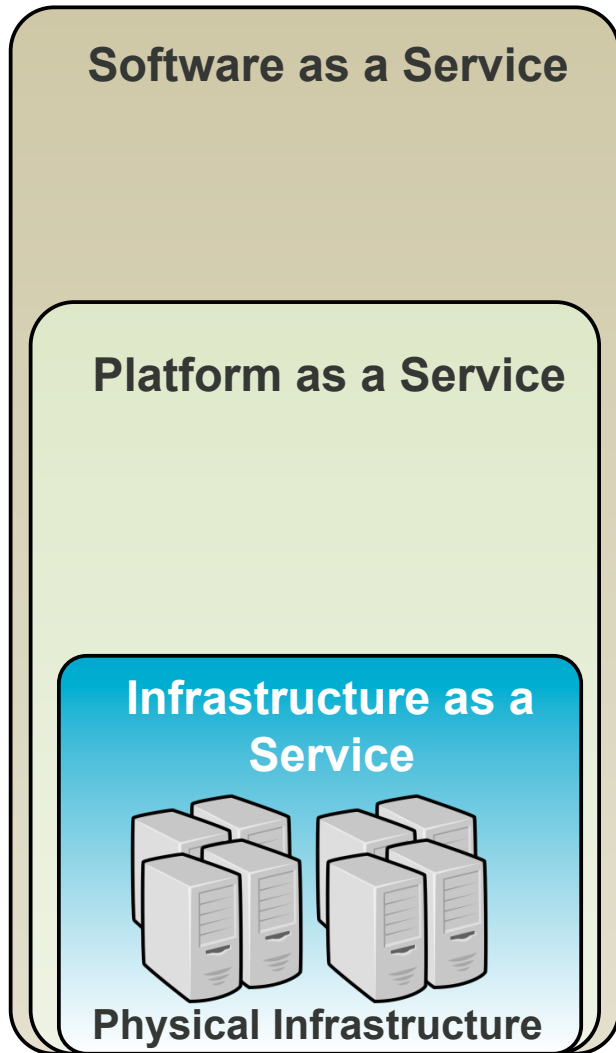


The version 0.4 of the software is available at <http://stratuslab.eu>

Version 1.0 is coming soon with utilities to upgrade previous installed versions.

Cloud Computing Disciplines

An Introduction to Cloud Computing



What

Who

On-demand access to any application

End-user
(does not care about hw or sw)



Platform for building and delivering web applications

Developer
(no managing of the underlying hw & sw layers)



Delivery of a *raw* computer infrastructure

System Administrator
(complete management of the computer infrastructure)



Infrastructure as a Service (IaaS)

An Introduction to Cloud Computing

Public Cloud

- Simple Web Interface
- Raw *Infrastructure* Resources
- Pay-as-you-go (On-demand access)
- Elastic & “infinite” Capacity



Infrastructure as a Service (IaaS)

An Introduction to Cloud Computing

Public Cloud

- Simple Web Interface
- Raw *Infrastructure* Resources
- Pay-as-you-go (On-demand access)
- Elastic & “infinite” Capacity



Private Cloud

A “*Public Cloud behind the firewall*”

- Simplify internal operations
- Dynamic allocation of resources
- Higher utilization & operational savings
- Security concerns

Infrastructure as a Service (IaaS)

An Introduction to Cloud Computing

Public Cloud

- Simple Web Interface
- Raw *Infrastructure* Resources
- Pay-as-you-go (On-demand access)
- Elastic & “infinite” Capacity



Private Cloud

A “Public Cloud behind the firewall”

- Simplify internal operations
- Dynamic allocation of resources
- Higher utilization & operational savings
- Security concerns

Hybrid Cloud

- Supplement the capacity of the Private Cloud
- Utility Computing dream made a reality!

Infrastructure as a Service (IaaS)

An Introduction to Cloud Computing

Public Cloud

- Simple Web Interface
- Raw *Infrastructure* Resources
- Pay-as-you-go (On-demand access)
- Elastic & “infinite” Capacity



Private Cloud

A “Public Cloud behind the firewall”

- Simplify internal operations
- Dynamic Allocation of resources
- Higher utilization & operational savings
- Security concerns

OpenNebula

Hybrid Cloud

- Supplement the capacity of the Private Cloud
- Utility Computing dream made a reality!

Challenges of an IaaS Cloud

An Introduction to Cloud Computing

I'm using virtualization/cloud, and plan a private Cloud (BUT's)

Where do/did I put my web server VM?

Monitoring & Scheduling

How do I provision a new VM?

Image Management & Context

Who have access to cloud (and What)?

User & Role Management

How do I create a new disk?

Storage

How do I set up networking for a multitier service?

Network & VLANs

How can I manage the distributed infrastructure?

Interfaces & APIs

Can I use hypervisor X?

Virtualization

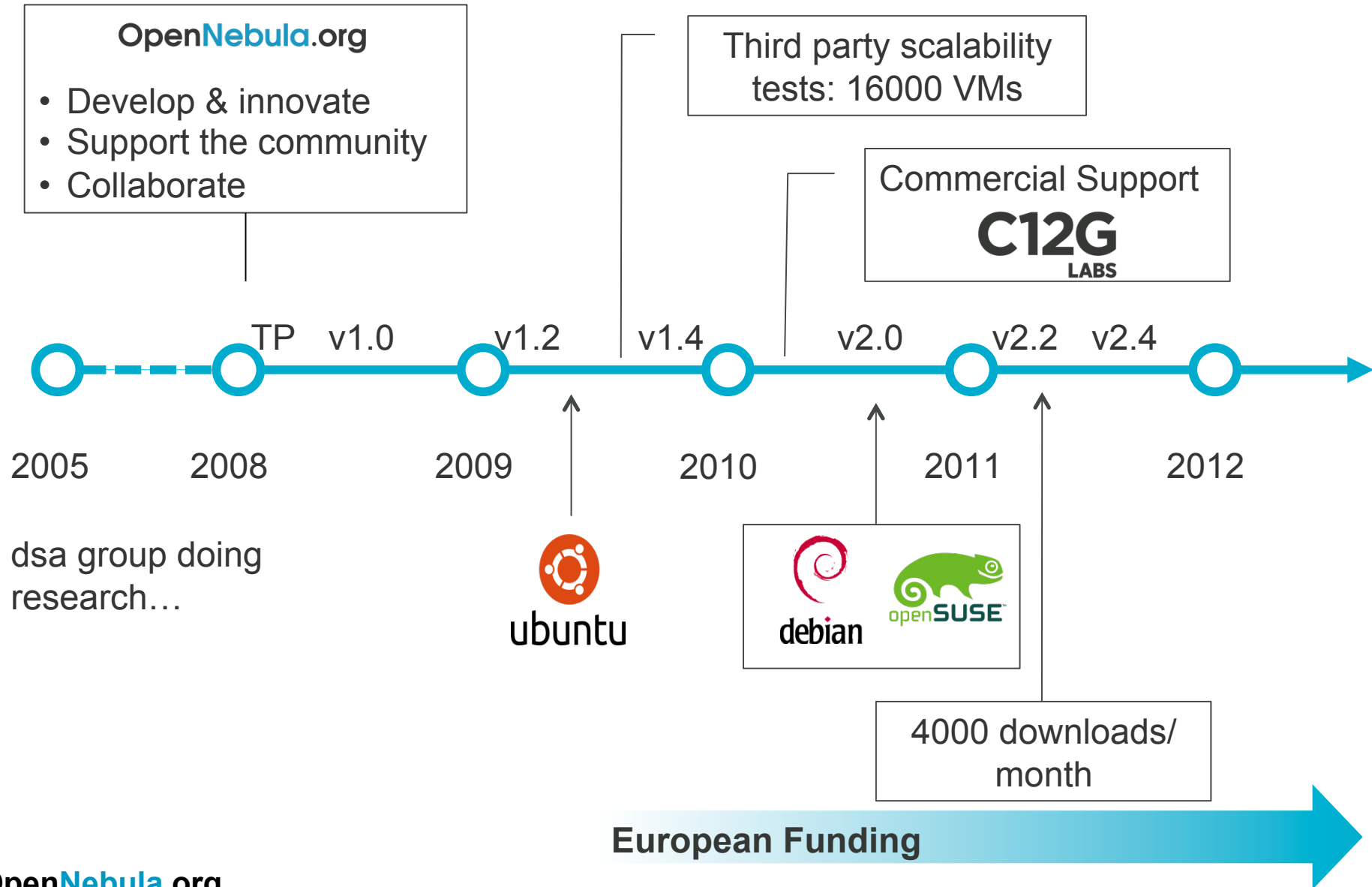
Uniform management layer that orchestrates multiple technologies



OpenNebula

History of OpenNebula.org

An Introduction to Cloud Computing



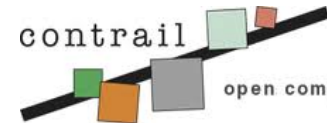
History of OpenNebula.org: Sample Users

An Introduction to Cloud Computing

Organizations Building Clouds for Development, Testing and Production



Projects Building an Open Cloud Ecosystem Around OpenNebula



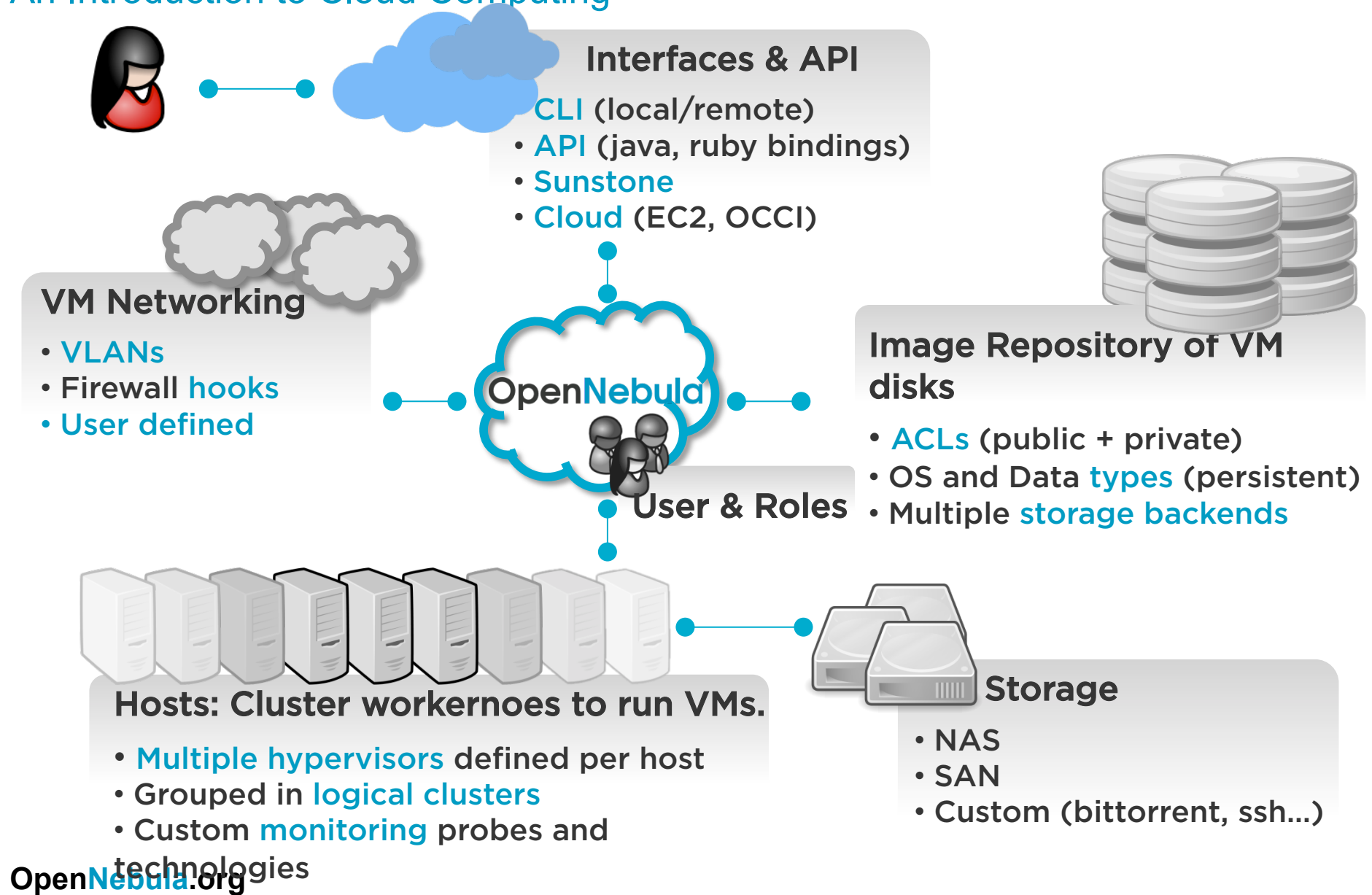
Technical Overview of OpenNebula: Vision & Design Philosophy

An Introduction to Cloud Computing

- **One solution can not fit all** data-center, requirements and constraints
- **Open, flexible and extensible** architecture that allows multiple components to be orchestrated
- **Ready for production**
- **Massively scalable** deployments
- **Open Source** - Apache License v2.0
- Provide basic components, but allow them to be **easily replaceable**

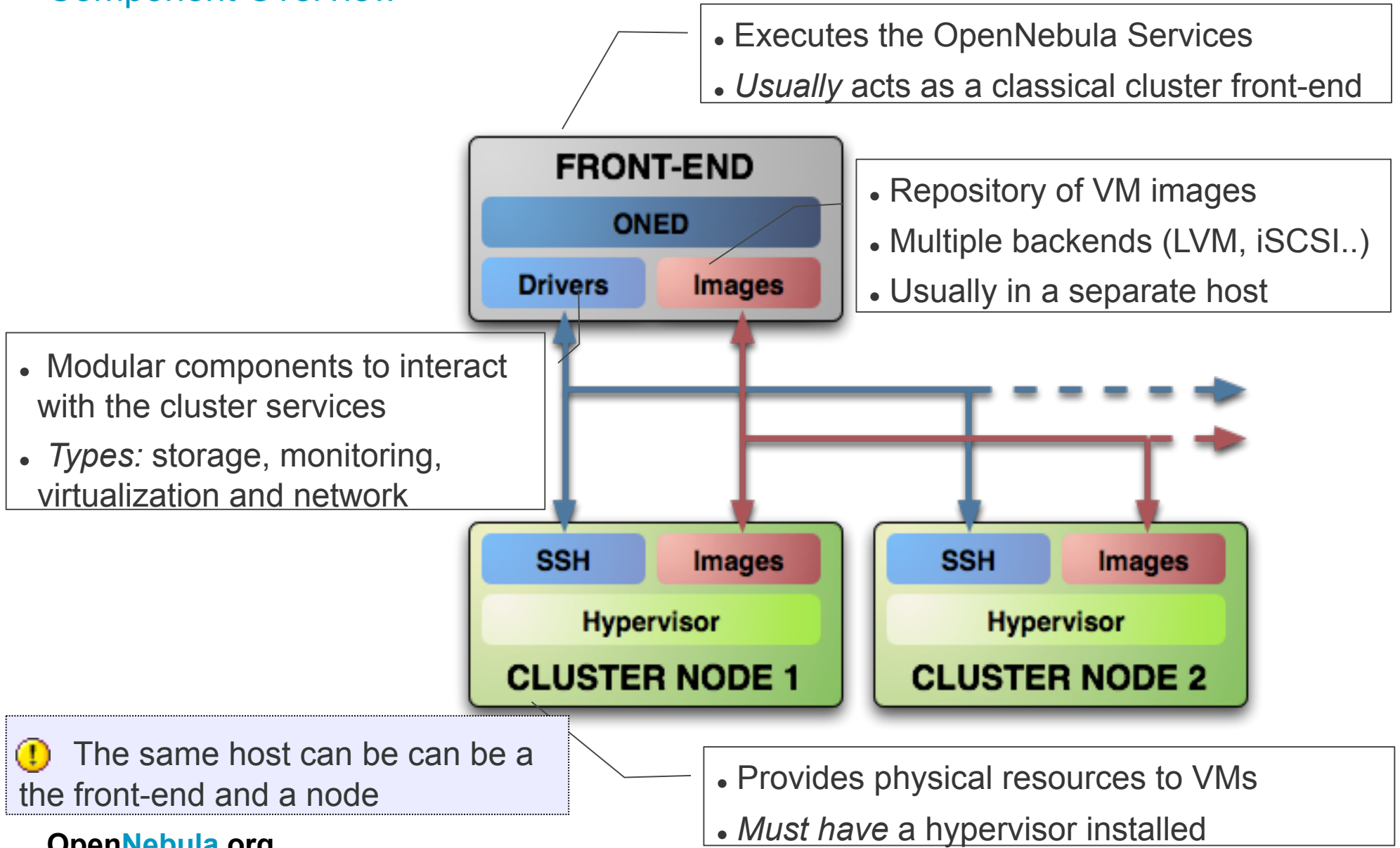
Technical Overview of OpenNebula: Key Components

An Introduction to Cloud Computing



Component Overview

Component Overview



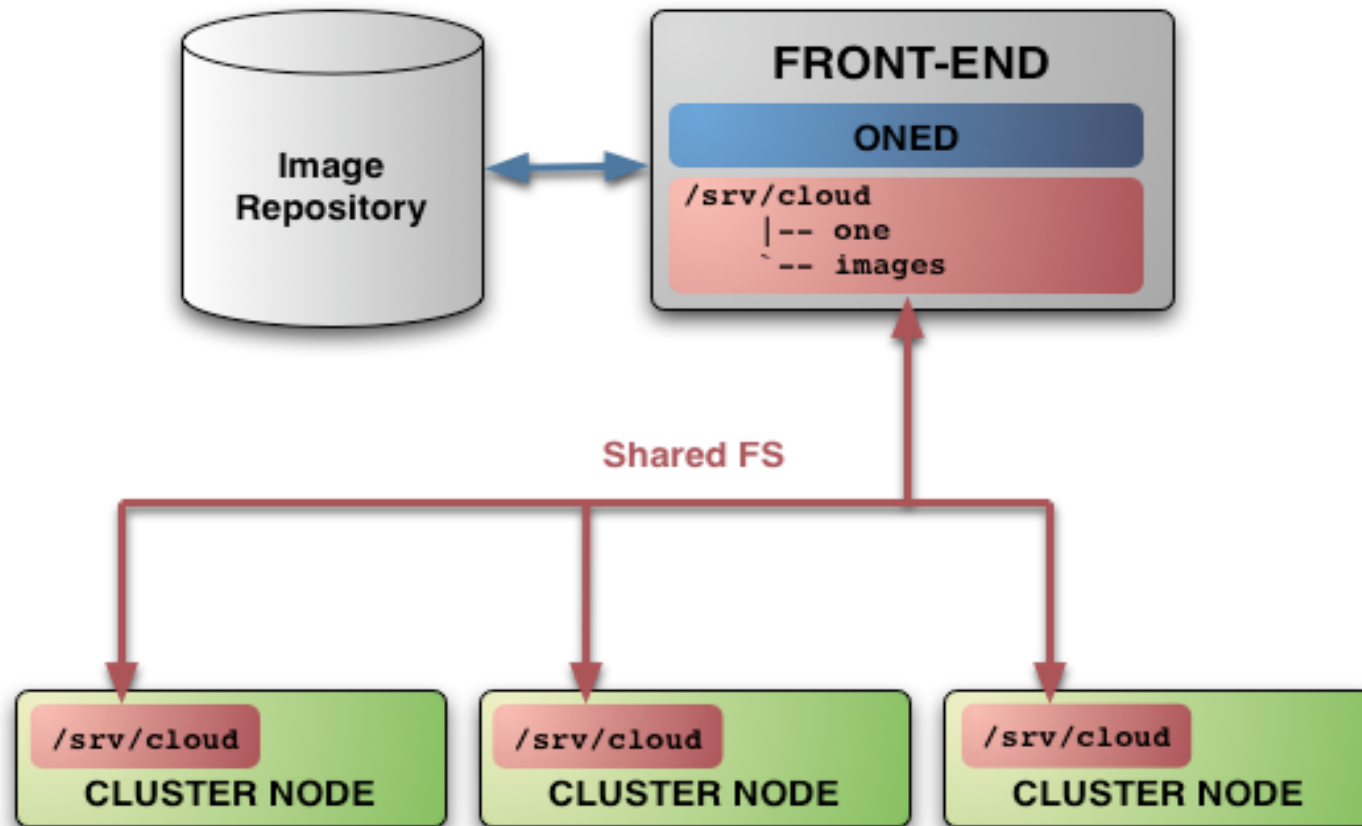
Storage for the Private Cloud

Component Overview

- **Image Repository:** Any storage medium for the VM images (usually a high performing SAN).
- **Cluster Storage**
 - OpenNebula supports multiple back-ends (e.g. LVM for fast cloning)
- **VM Directory:** The home of the VM in the cluster node
 - Stores checkpoints, description files and VM disks
 - Actual operations over the VM directory depends on the storage medium
 - Should be shared for live-migrations
 - You can go on without a shared FS and use the SSH back-end
 - Defaults to `$ONE_LOCATION/var/$VM_ID`

Storage for the Private Cloud

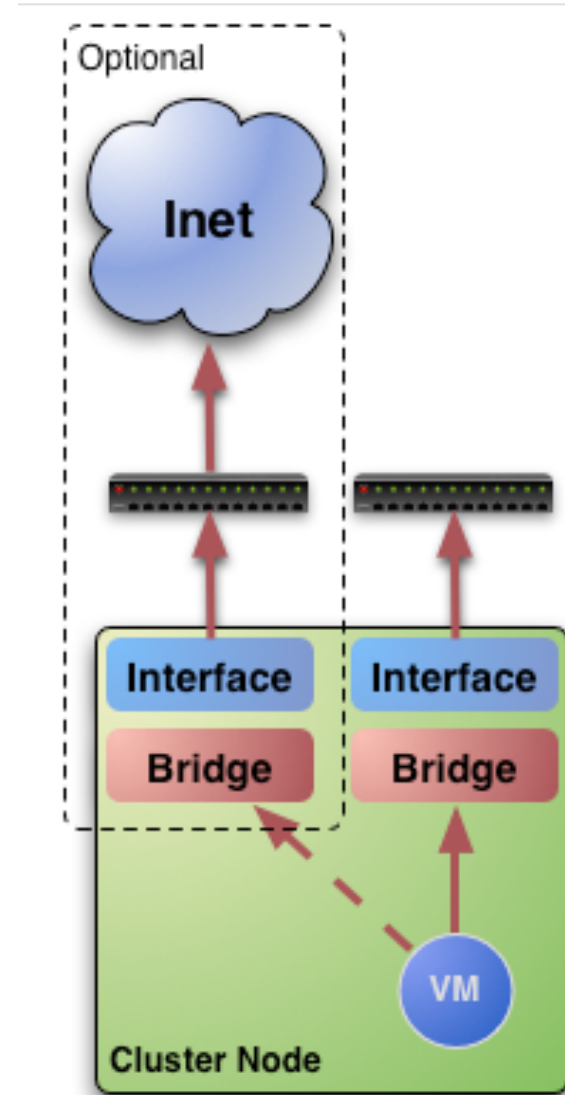
Example, a shared FS architecture



Networking for the Private Cloud

Component Overview

- OpenNebula management operations uses a ssh connections, it does not require a performing NIC
- **Image traffic**, may require the movement of heavy files (VM images, checkpoints). Dedicated storage links may be a good idea
- **VM demands**, consider the typical requirements of your VMs. Several NICs to support the VM traffic may be a good idea
- OpenNebula relies on bridge networking for the VMs

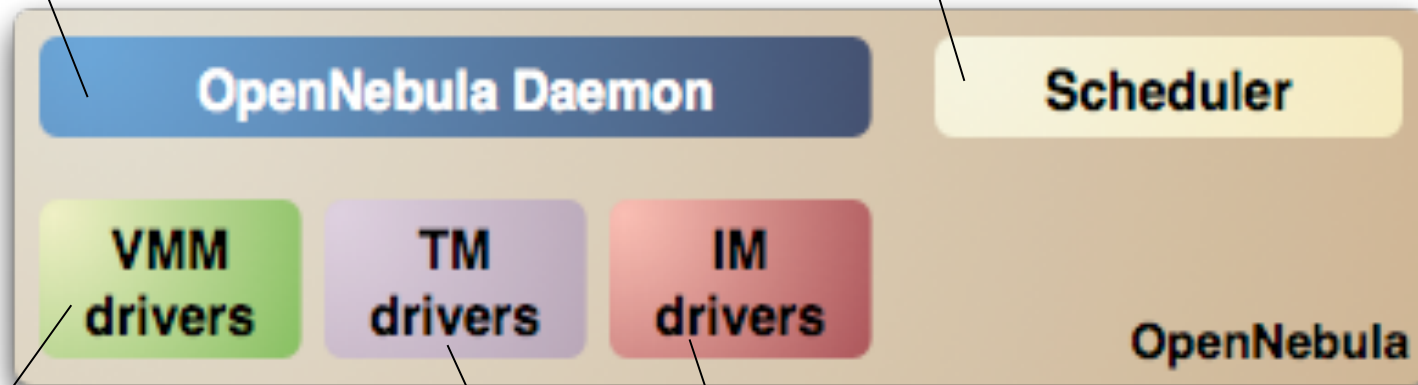


Configuring OpenNebula 2.2

Configuration Interface

- **`$ONE_LOCATION/etc/oned.conf`**
- General configuration
- Defines the drivers used in the private cloud

- Match-making scheduler (default)
- Placement policies configured per VM



- **`$ONE_LOCATION/etc/vmm_*/vmm_*.conf`**
- Defaults values for the hypervisor

- **`$ONE_LOCATION/etc/im_*/im_*.conf`**
- Defines monitoring probes

- **`$ONE_LOCATION/etc/tm_*/tm_*.conf`**
- Defines action for generic storage operations

Configuring OpenNebula

The oned.conf file

- **General configuration attributes**

- Monitoring intervals:
 - HOST_MONITORING_INTERVAL
 - VM_POLLING_INTERVAL
- Global Paths
 - VM_DIR: Path to the VM directory **in the** cluster nodes.
 - SCRIPTS_REMOTE_DIR: to store driver actions in the cluster nodes
- PORT : Port where oned will listen for xml-rpc calls
- DEBUG_LEVEL
- DB, configuration for the DB backend driver:
 - Sqlite
 - MySQL
- VNC_BASE_PORT, for VNC port generation (BASE + ID)

Configuring OpenNebula

The oned.conf file

- **Information Drivers**, to monitor cluster nodes
 - name: identifies the driver
 - executable: absolute or relative to \$ONE_LOCATION/lib/mads
 - arguments:
 - hypervisor probe set (remotes dir)
 - Number of retries (-r)
 - Concurrency (-t number of threads)

```
IM_MAD = [  
    name          = "im_kvm",  
    executable    = "one_im_ssh",  
    arguments     = "-r 0 -t 15 kvm" ]
```

Configuring OpenNebula

The oned.conf file

- **Transfer Drivers**, to interface with the storage
 - name: identifies the driver
 - executable: path to driver executable
 - arguments: storage commands configuration file

```
TM_MAD = [  
    name          = "tm_nfs",  
    executable    = "one_tm",  
    arguments     = "tm_nfs/tm_nfs.conf" ]
```

Configuring OpenNebula

The oned.conf file

- **Virtualization Drivers**, to interface the hypervisors
 - name: identifies the driver
 - executable: absolute or relative to \$ONE_LOCATION/lib/mads
 - arguments: same as Information Drivers
 - default: default values for the hypervisor
 - type: format of the VM description used by the driver: xen, kvm or xml

```
VM_MAD = [  
    name          = "vmm_kvm",  
    executable    = "one_vmm_ssh",  
    arguments     = "-t 15 -r 0 kvm",  
    default       = "vmm_ssh/vmm_ssh_kvm.conf",  
    type          = "kvm" ]
```

Configuring the Private Cloud

Managing hosts and clusters

- **Hosts** are cluster worker-nodes defined with
 - *Hostname* of the node or IP
 - *Information Driver* to be used to monitor the host
 - *Storage Driver* to clone, delete, move or copy images into the host
 - *Virtualization Driver* to boot, stop, resume VMs in the host
- Hosts are managed with the **onehost** utility
 - Create & delete hosts
 - List the hosts
 - Show detailed information from a host
 - Enable/Disable a host

Configuring the Private Cloud

Managing Users

- **Users** are defined within OpenNebula by:
 - *ID* unique identifier for the user
 - *Name* of the user, used for authentication
 - *Password* used for authentication
- Users are managed with the **oneuser** utility
 - Create & delete users
 - List the users in the cluster

Virtual Networks

Overview

- A **Virtual Network (vnet)** in OpenNebula
 - Defines a separated MAC/IP address space to be used by VMs
 - A vnet is associated with a physical network through a bridge
 - Virtual Networks can be isolated (at layer 2 level)
- Virtual Network **definition**
 - **Name**, of the network
 - **Type**
 - **Fixed**, a set of IP/MAC leases
 - **Ranged**, defines a network range
 - **Bridge**, name of the physical bridge in the physical host where the VM should connect its network interface.
- Virtual Networks are managed with the **onevnet** utility

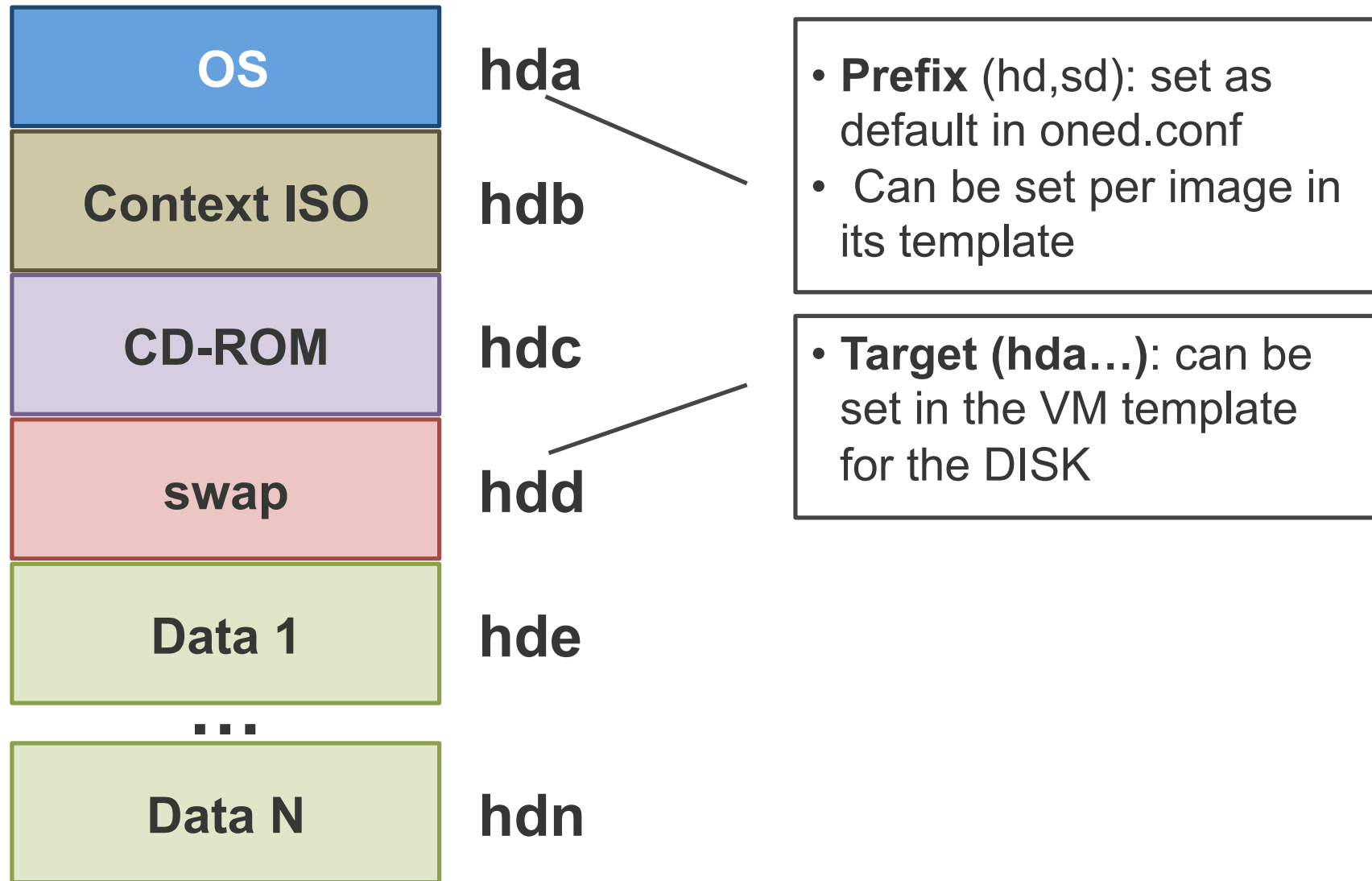
Images

Overview

- An **Image** in OpenNebula's repository
 - A virtual machine disk to be used as OS or DATA device.
 - Images can be **persistent** and/or **public**
 - Images modifications can be saved as another image
- **Image Types:**
 - **OS:** contains a working operative system
 - **CDROM:** readonly data
 - **DATABLOCK:** A storage for data. Can be created either from previous existing data, or as an empty drive.
- Images are *stored* in the repository

Images

Automatic Disk Layout for Images



Virtual Machines

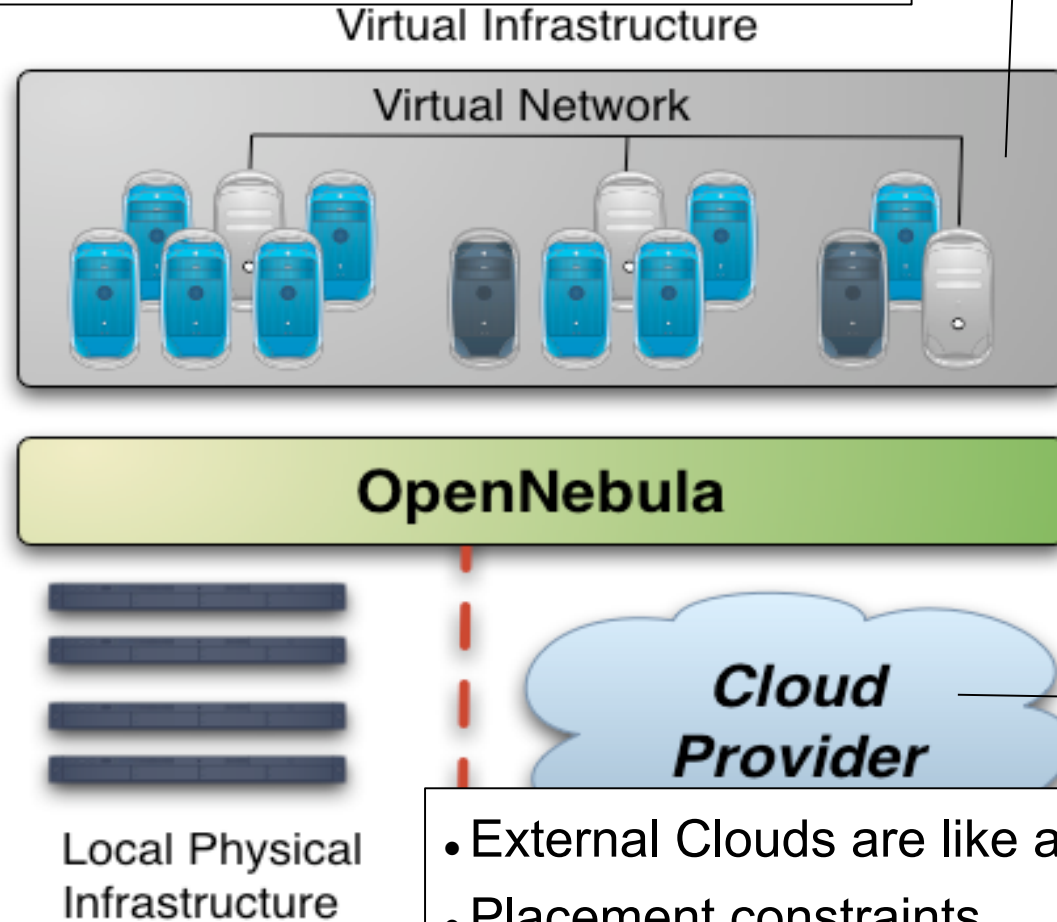
Overview

- A **Virtual Machine** in OpenNebula
 - A **capacity** in terms memory and CPU
 - A set of **NICs** attached to one or more virtual networks
 - A set of **disk images**, to be “*transferred*” to/from the execution host.
 - A **state file** (optional) or recovery file, with the memory image of a running VM plus some hypervisor specific information.
- Virtual Machines are defined in a **VM template**
- Each VM has an unique ID in OpenNebula the VMID
- All the files (logs, images, state files...) are stored in **`$ONE_LOCATION/var/<VMID>`**

Hybrid Cloud Computing

Overview

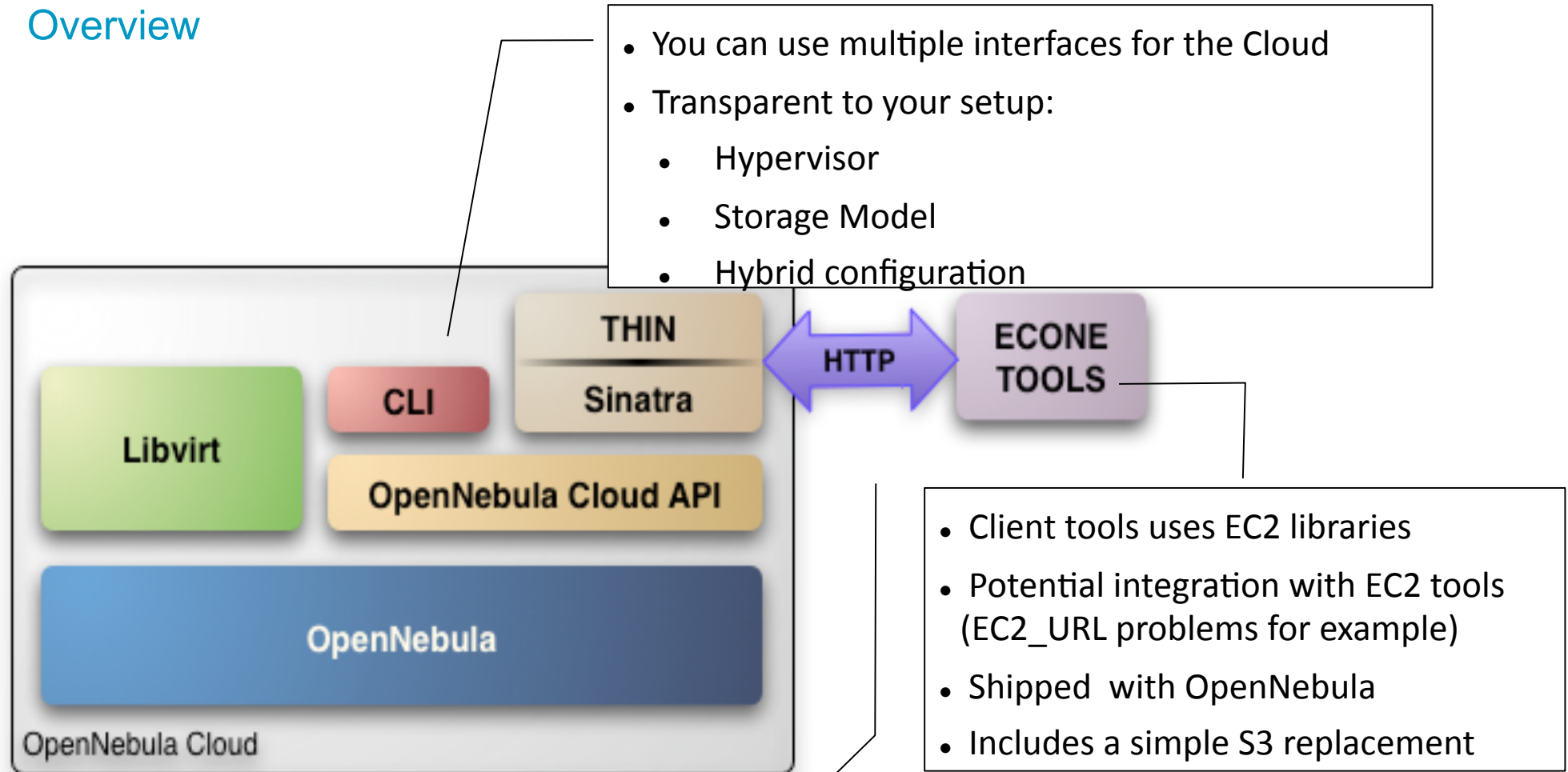
- VMs can be local or remote
- VM connectivity has to be configured, usually VPNs



- External Clouds are like any other host
- Placement constraints
- OpenNebula distribution includes EC2 drivers

Public Cloud Computing with OpenNebula

Overview



- You can use multiple interfaces for the Cloud
- Transparent to your setup:
 - Hypervisor
 - Storage Model
 - Hybrid configuration

- Client tools uses EC2 libraries
- Potential integration with EC2 tools (EC2_URL problems for example)
- Shipped with OpenNebula
- Includes a simple S3 replacement

- Supports HTTP and HTTPS protocols
- *EC2 authentication* based on OpenNebula credentials
- Public Cloud users need an OpenNebula account

Configuring the Public Cloud

Define the Instances

- You have to define the correspondence between **types** (simple) and **local instantiation of VMs** (hard, you should be fine by now)
 - Capacity allocated by this VM type (CPU, MEMORY)
 - Your cloud requirements, e.g. force to use a given kernel (OS) or place public VMs in a given set of cluster nodes (REQUIREMENTS)
 - The network used by Public VMs (NIC)
- VM Types are **defined in `econe.conf`**. Templates for the VM templates are in `$ONE_LOCATION/etc/ec2query_templates`
- Templates for VM Types are erb files `<% Ruby code here %>`, you should not need to modify that.

Using the Public Cloud

The econe Toolset

- The **econe-tools** are a subset of the functionality provided by the onevm utility, and resembles the ec2-* cli
- EC2 ecosystem *can* be used (e.g. elasticfox, euca2ools...)
- **Image** related commands are:
 - **econe-upload**, place an image in the Cloud repo and returns ID
 - **econe-describe-images**, lists the images
 - **econe-register**, register an image
- **Instance** related commands are:
 - **econe-run-instances**, starts a VM using an image ID
 - **econe-describe-instances**, lists the VMs
 - **econe-terminate-instances**, shutdowns a VM

Using the Public Cloud

The econe Toolset

- **User authentication** is based in the OpenNebula credentials
 - AWSAccessKeyId is OpenNebula's username
 - AWSSecretAccessKey is OpenNebula's password
- Pass **your credentials** to the econe-tools by (in this order)
 - Command arguments (-K <username>, -S <pass>)
 - Environment EC2_ACCESS_KEY and EC2_SECRET_KEY
 - Environment ONE_AUTH
- Point econe-tools to your **target cloud**
 - Command arguments (-U <http|https>://<fqdn>:<port>) port needed if not the default for the protocol
 - EC2_URL environment

Hands-on Session

Sunstone web interface

<https://cloud.opennebula.org>

USER: tutorial

PASSWORD: lero