Open Grid Forum 28
March 15-18, 2010
Munich, Germany

Deployment of Private and Hybrid Clouds Using OpenNebula/RESERVOIR

Constantino Vázquez Blanco (tinova@fdi.ucm.es)
Rubén Santiago Montero (rubensm@dacya.ucm.es)

dsa-research.org

Distributed Systems Architecture Research Group
Universidad Complutense de Madrid

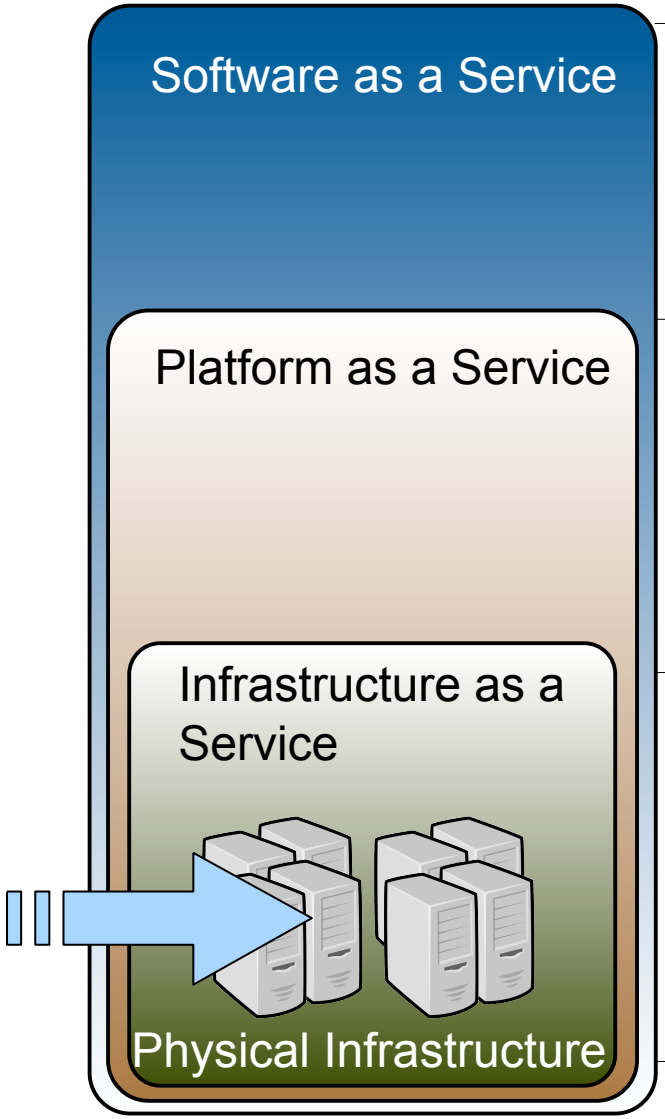
Outline

- Cloud Computing in a Nutshell
- What is OpenNebula?
- Private Cloud :: Deploying OpenNebula
- Private Cloud :: Using OpenNebula
- Hybrid Cloud

Cloud Computing in a Nutshell

Cloud Computing in a Nutshell

Overview



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)

Cloud Computing in a Nutshell

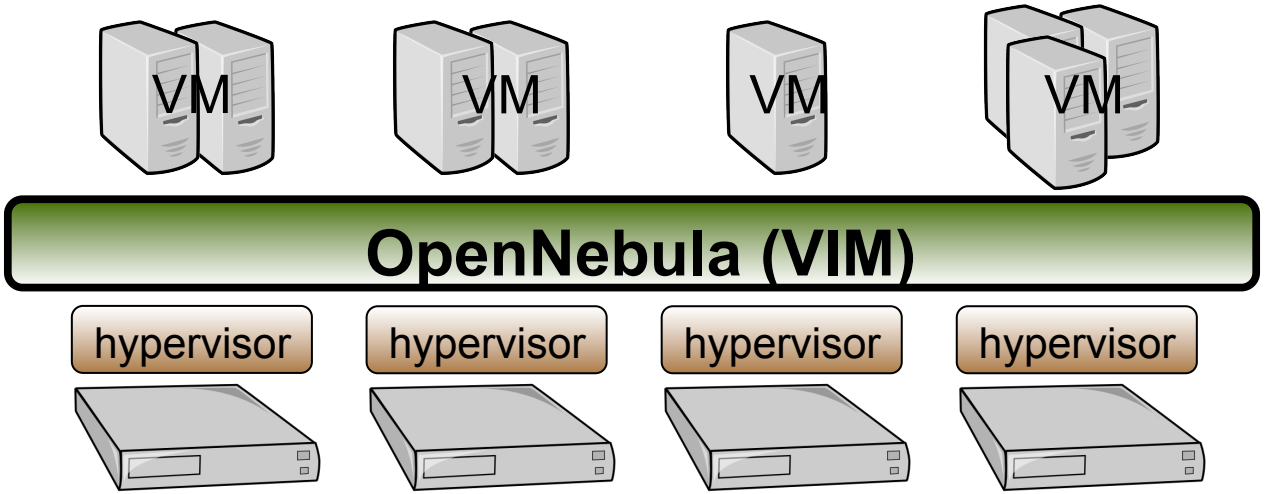
The IaaS Clouds: A Four Point Check List

- Simple Interface
- Raw *Infrastructure* Resources
  - Total control of the resources
  - Capacity leased in the form of VMs
  - Complete Service-HW decoupling
- Pay-as-you-go
- Elastic & *"infinite"* Capacity

Cloud Computing in a Nutshell

Why a Virtual Infrastructure Manager?

- VMs are great!!...but something more is needed
  - Where did/do I put my VM? (***scheduling & monitoring***)
  - How do I provision a new cluster node? (***clone & context***)
  - What MAC addresses are available? (***networking***)
- Provides a ***uniform view*** of the resource pool
- ***Life-cycle management*** and monitoring of VM
- The VIM ***integrates*** Image, Network and Virtualization



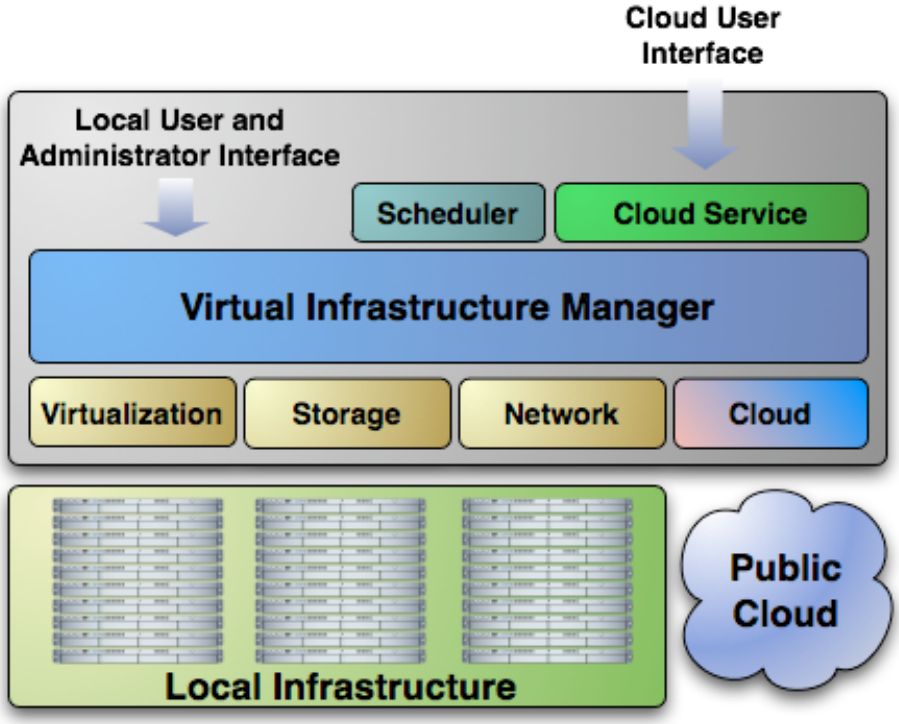
What is OpenNebula?

What is OpenNebula?

The OpenNebula Virtual Infrastructure Engine

Extending the Benefits of Virtualization to Clusters

- Dynamic deployment and re-placement of virtual machines on a pool of physical resources
- Transform a rigid distributed physical infrastructure into a flexible and agile virtual infrastructure



Backend of Public Cloud: Internal management of the infrastructure

Private Cloud: Virtualization of cluster or data-center for internal users

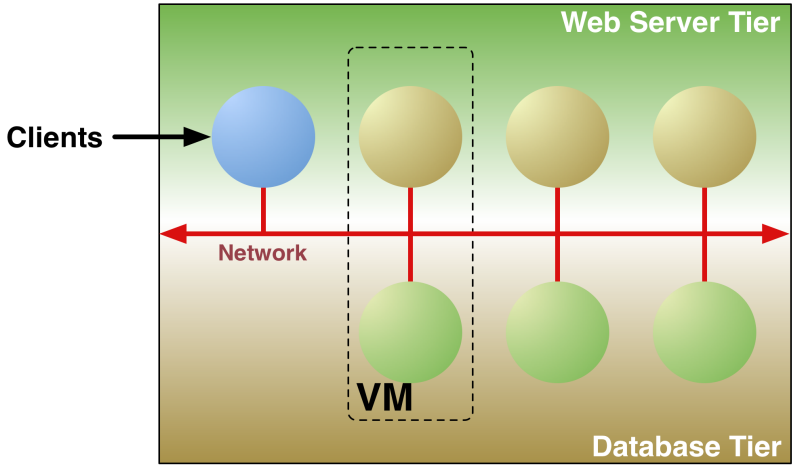
Cloud Interoperation: On-demand access to public clouds

What is OpenNebula?

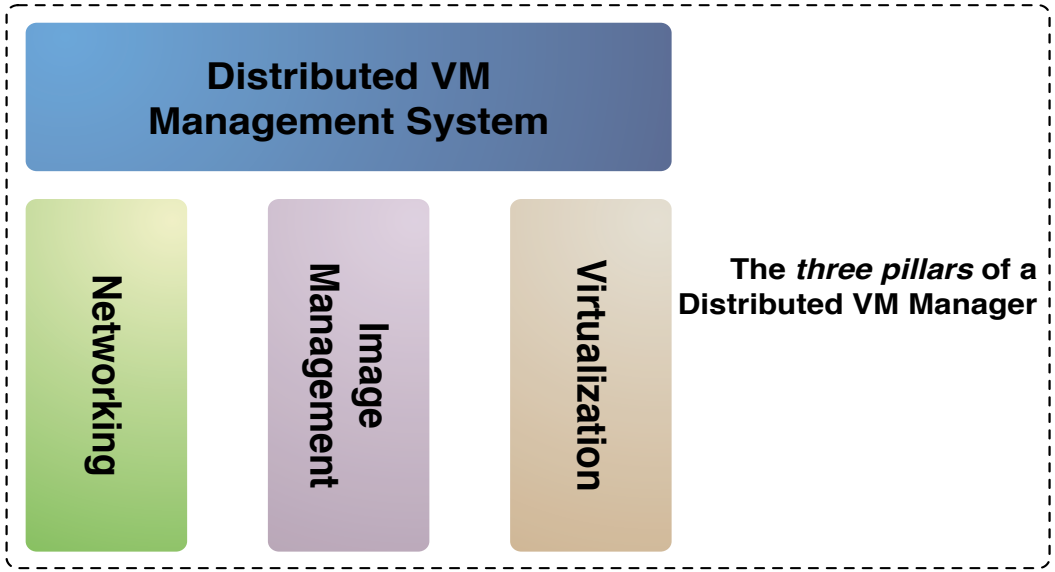
Virtual Machine Management Model

Service as Management Entity

- Service structure
  - Service components run in VMs
  - Inter-connection relationship
  - Placement constraints
- The VM Manager is service agnostic
- Provide infrastructure context



Distributed VM Management Model



dsa-research.org

What is OpenNebula?

Benefits

For the Infrastructure Manager

- Centralized management of VM workload and distributed infrastructures
- Support for VM placement policies: balance of workload, server consolidation…
- Dynamic resizing of the infrastructure
- Dynamic partition and isolation of clusters
- Dynamic scaling of private infrastructure to meet fluctuating demands
- Lower infrastructure expenses combining local and remote Cloud resources

For the Infrastructure User

- Faster delivery and scalability of services
- Support for heterogeneous execution environments
- Full control of the lifecycle of virtualized services management

dsa-research.org

What is OpenNebula?

Benefits

For System Integrators

- Fits into any existing data center, due to its open, flexible and extensible interfaces, architecture and components
- Builds any type of Cloud deployment
- Open source software, Apache license
- Seamless integration with any product and service in the cloud ecosystem and management tool in the data center, such as
  - cloud providers
  - VM managers
  - virtual image managers
  - service managers
  - management tools
  - schedulers

dsa-research.org

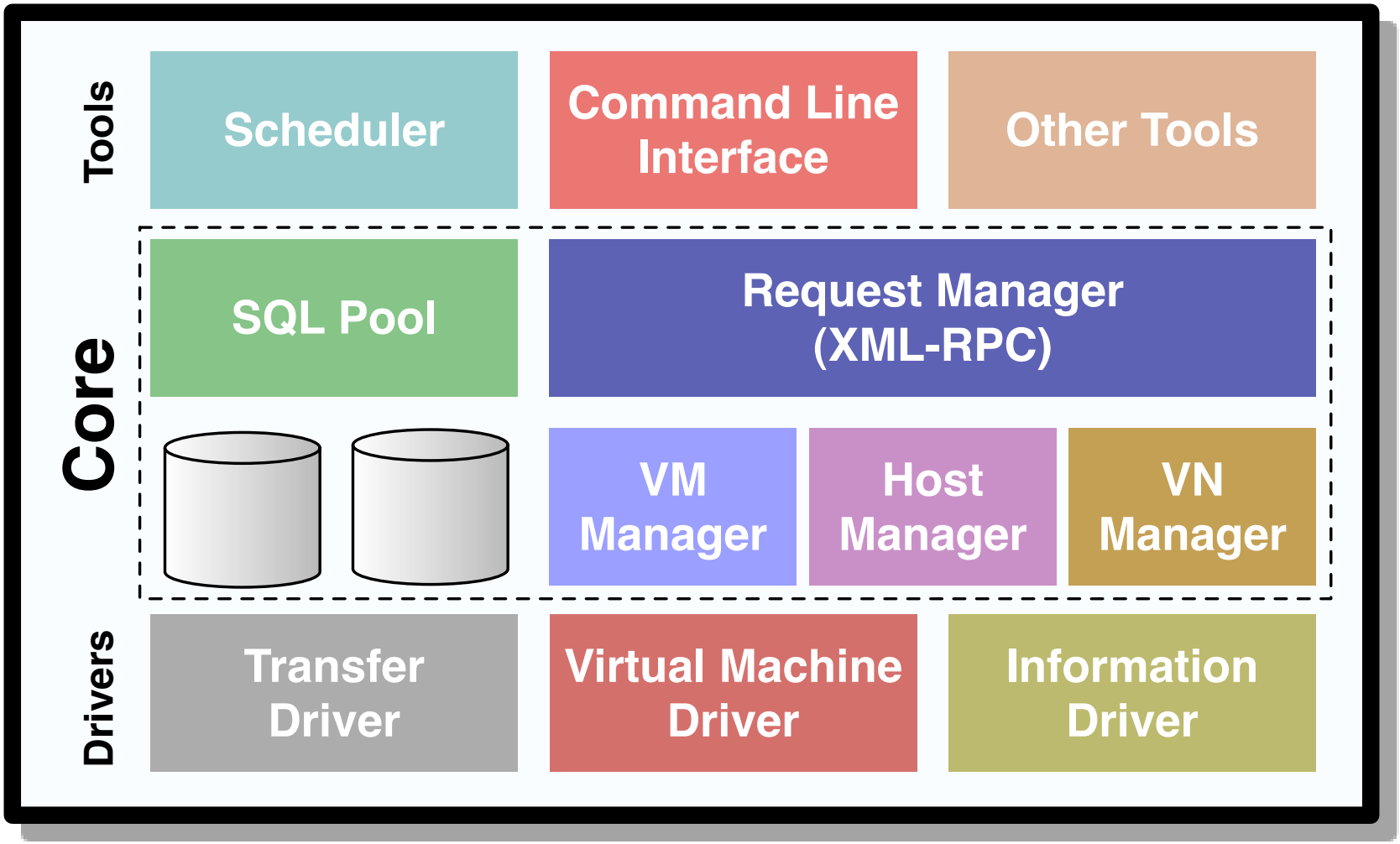
What is OpenNebula?

Features

Feature	Function
Internal Interface	• Unix-like CLI for fully management of VM life-cycle and physical boxes • XML-RPC API and libvirt virtualization API
Scheduler	• Requirement/rank matchmaker allowing the definition of workload and resource-aware allocation policies • Support for advance reservation of capacity through Haizea
Virtualization Management	• Xen, KVM, and VMware • Generic libvirt connector (VirtualBox planned for 1.4.2)
Image Management	• General mechanisms to transfer and clone VM images
Network Management	• Definition of isolated virtual networks to interconnect VMs
Service Management and Contextualization	• Support for multi-tier services consisting of groups of inter-connected VMs, and their auto-configuration at boot time
Security	• Management of users by the infrastructure administrator
Fault Tolerance	• Persistent database backend to store host and VM information
Scalability	• Tested in the management of medium scale infrastructures with hundreds of servers and VMs (no scalability issues has been reported)
Installation	• Installation on a UNIX cluster front-end without requiring new services • Distributed in Ubuntu 9.04 (Jaunty Jackalope)
Flexibility and Extensibility	• Open, flexible and extensible architecture, interfaces and components, allowing its integration with any product or tool

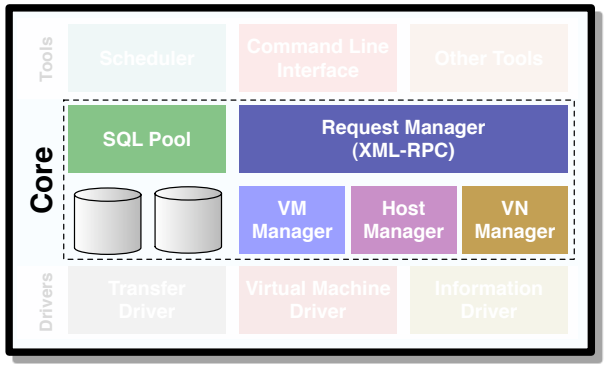
What is OpenNebula?

OpenNebula Architecture



What is OpenNebula?

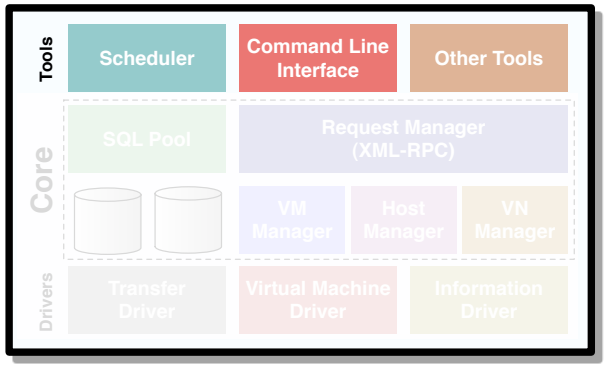
OpenNebula Architecture :: Core



- Request manager: Provides a XML-RPC interface to manage and get information about ONE entities.
- SQL Pool: Database that holds the state of ONE entities.
- VM Manager (virtual machine): Takes care of the VM life cycle.
- Host Manager: Holds the information about hosts and how to interact with them.
- VN Manager (virtual network): This component is in charge of generating MAC and IP addresses.

What is OpenNebula?

OpenNebula Architecture :: Tools

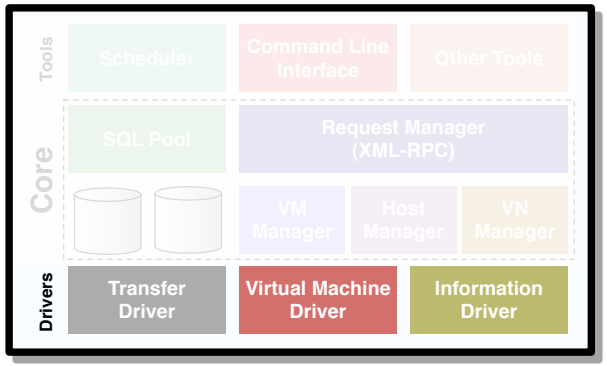


- Scheduler: This component searches for physical hosts to deploy newly defined VMs

- Command Line Interface: Commands used to manage OpenNebula entities.
  - onevm: Virtual Machines
    - *create, list, migrate…*
  - onehost: Hosts
    - *create, list, disable…*
  - onevnet: Virtual Networks
    - *create, list, delete…*
  - oneuser: Users
    - *create, list, delete…*

What is OpenNebula?

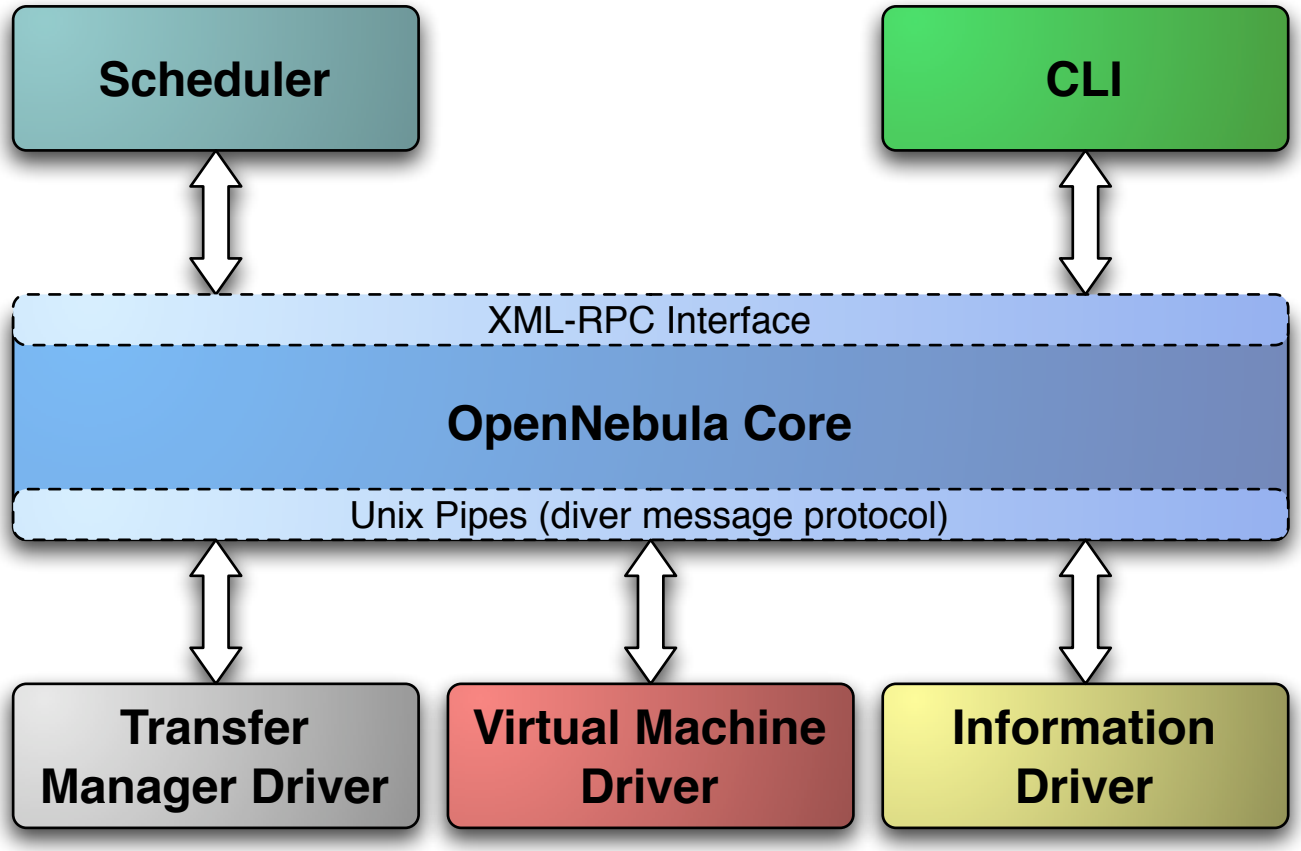
OpenNebula Architecture :: Drivers



- Transfer Driver: Takes care of the images.
  - *cloning, deleting, creating swap image…*
- Virtual Machine Driver: Manager of the lifecycle of a virtual machine
  - *deploy, shutdown, poll, migrate…*
- Information Driver: Executes scripts in physical hosts to gather information about them
  - *total memory, free memory, total cpus, cpu consumed…*

What is OpenNebula?

OpenNebula Architecture :: Process separation

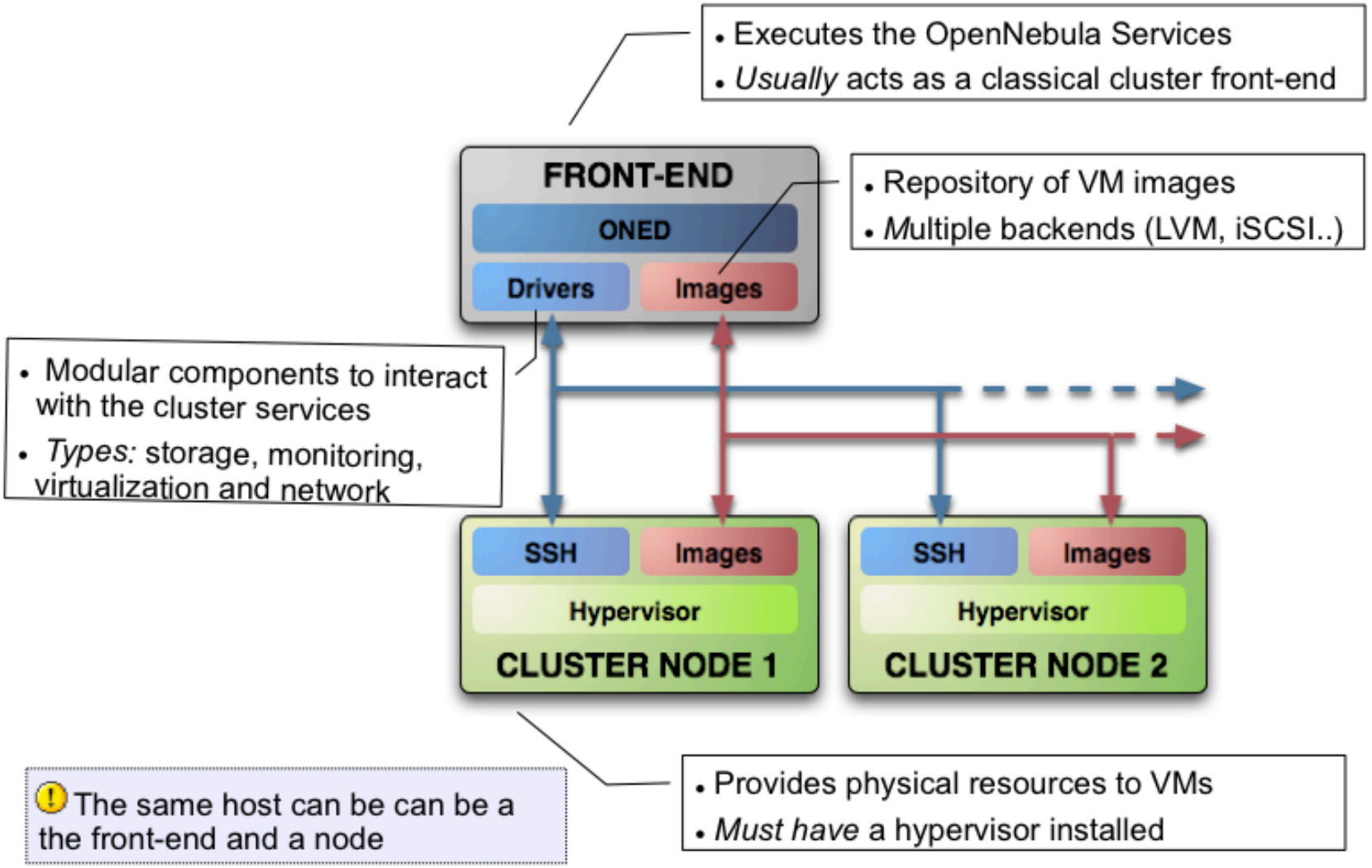


- Scheduler is a separated process, just like command line interface.
- Drivers are also separated processes using a simple text messaging protocol to communicate with OpenNebula Core Daemon (oned)

Private Cloud :: Deploying OpenNebula

Private Cloud :: Deploying OpenNebula

System Overview



Private Cloud :: Deploying OpenNebula

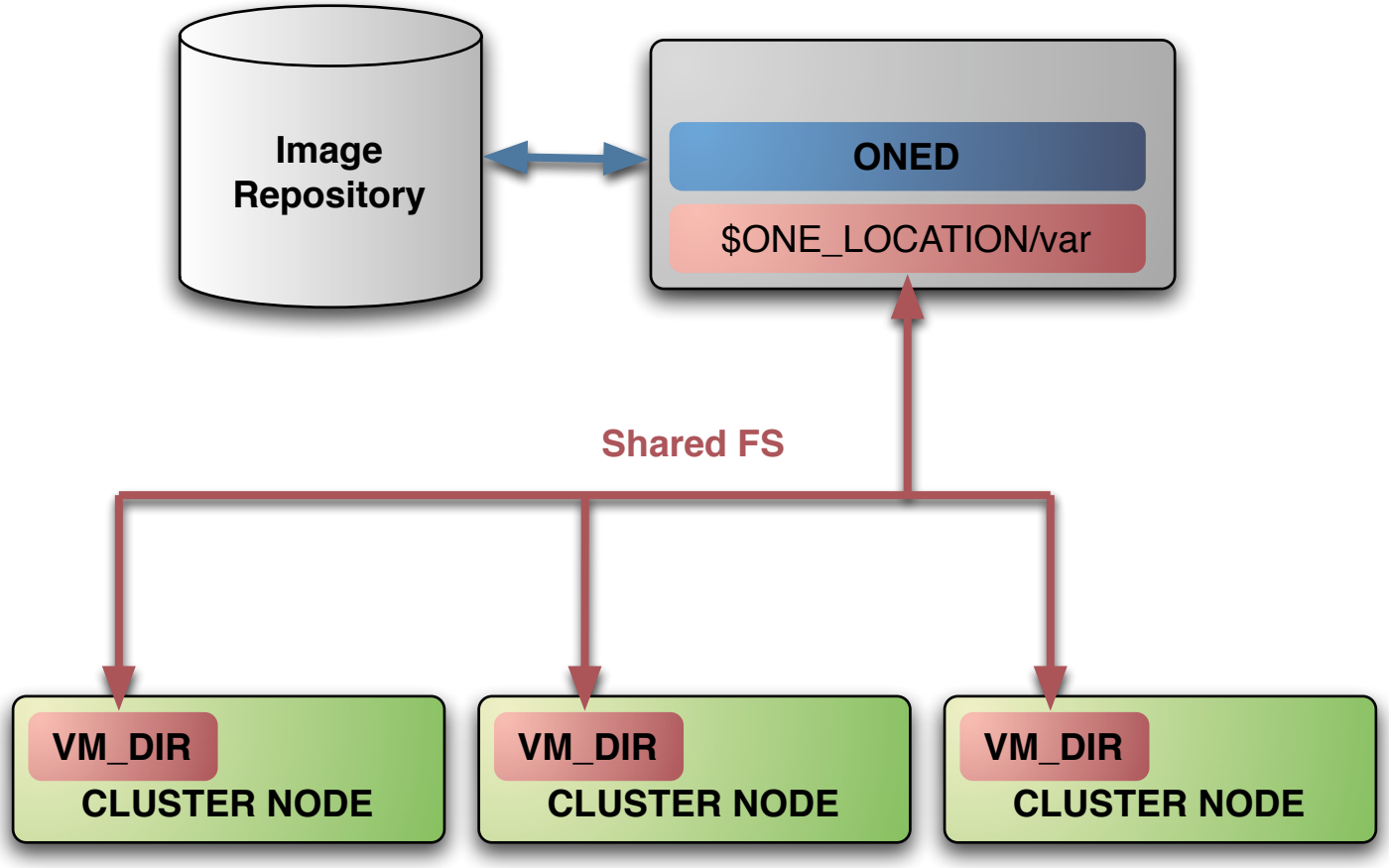
Storage

- Preparing the storage for the private cloud...
  - ***Image Repository:*** Any storage medium for the VM images (usually a high performing SAN)
    - OpenNebula supports multiple back-ends (e.g. LVM for fast cloning)
    - The front-end must have access to the repository
  - ***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

Dimensioning the Storage... Example: A 64 core cluster will typically run around 80VMs, each VM will require an average of 10GB of disk space. So you will need ~800GB for /srv/cloud/one, you will also want to store 10-15 master images so ~200GB for /srv/cloud/images. A 1TB /srv/cloud will be enough for this example setup.

Private Cloud :: Deploying OpenNebula

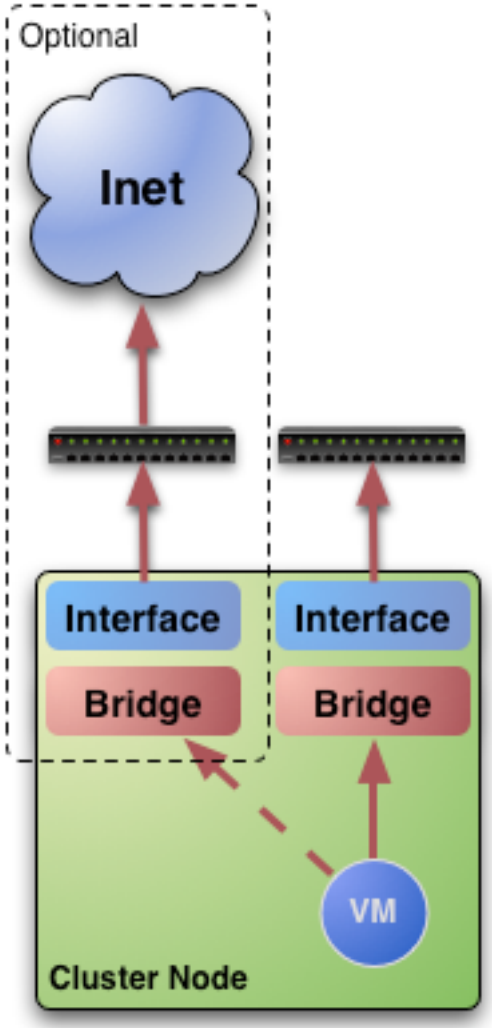
Storage



Private Cloud :: Deploying OpenNebula

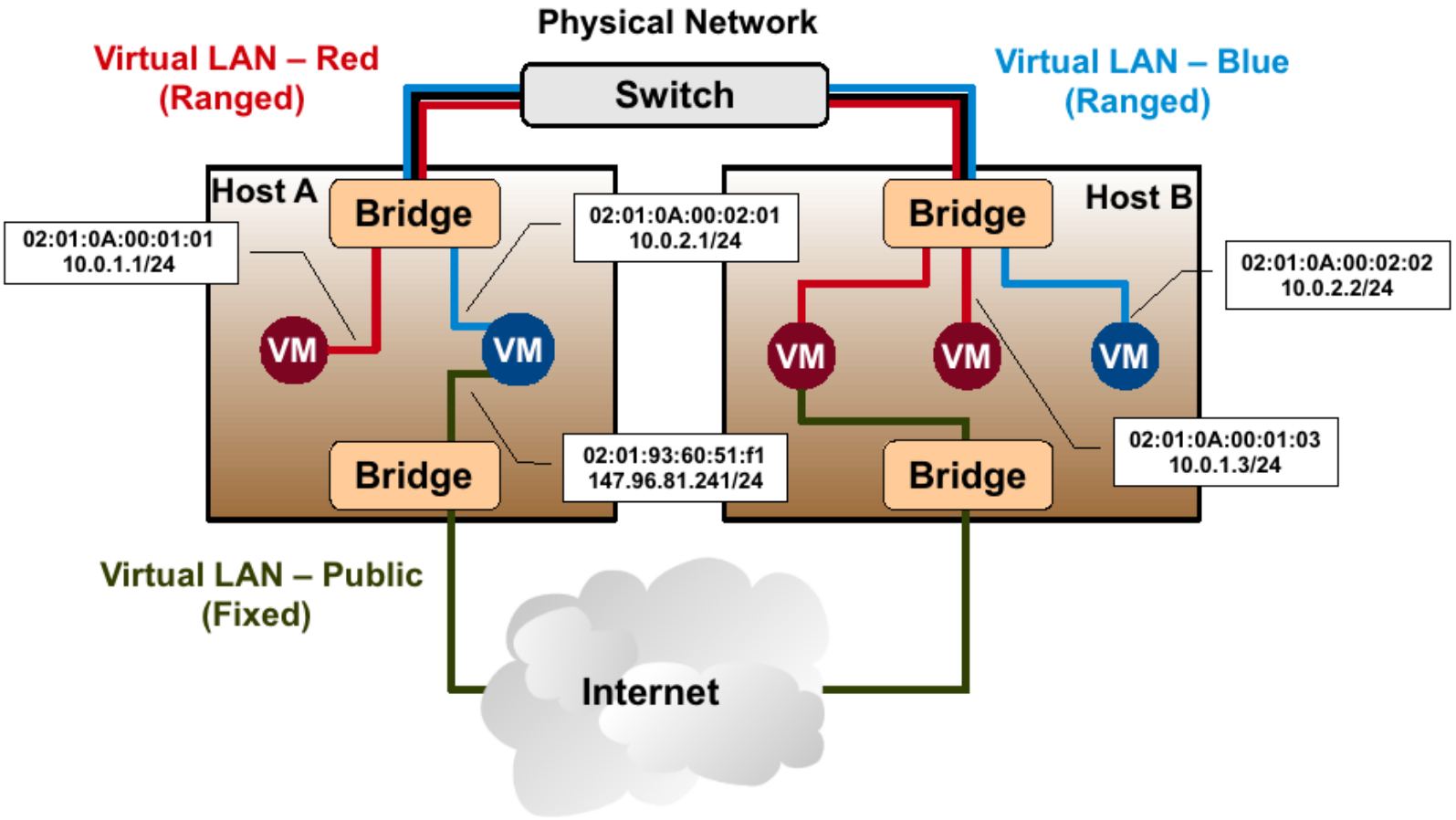
Networking

- Networking for the private cloud
  - 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



Private Cloud :: Deploying OpenNebula

Networking



Private Cloud :: Deploying OpenNebula

User Management

- Native user support since v1.4
- oneadmin: privileged account
- networks and VMs (storage in v1.6)
- SHA1 passwords (AA module in v1.6)
  - Stored in FS
  - Alternatively in environment

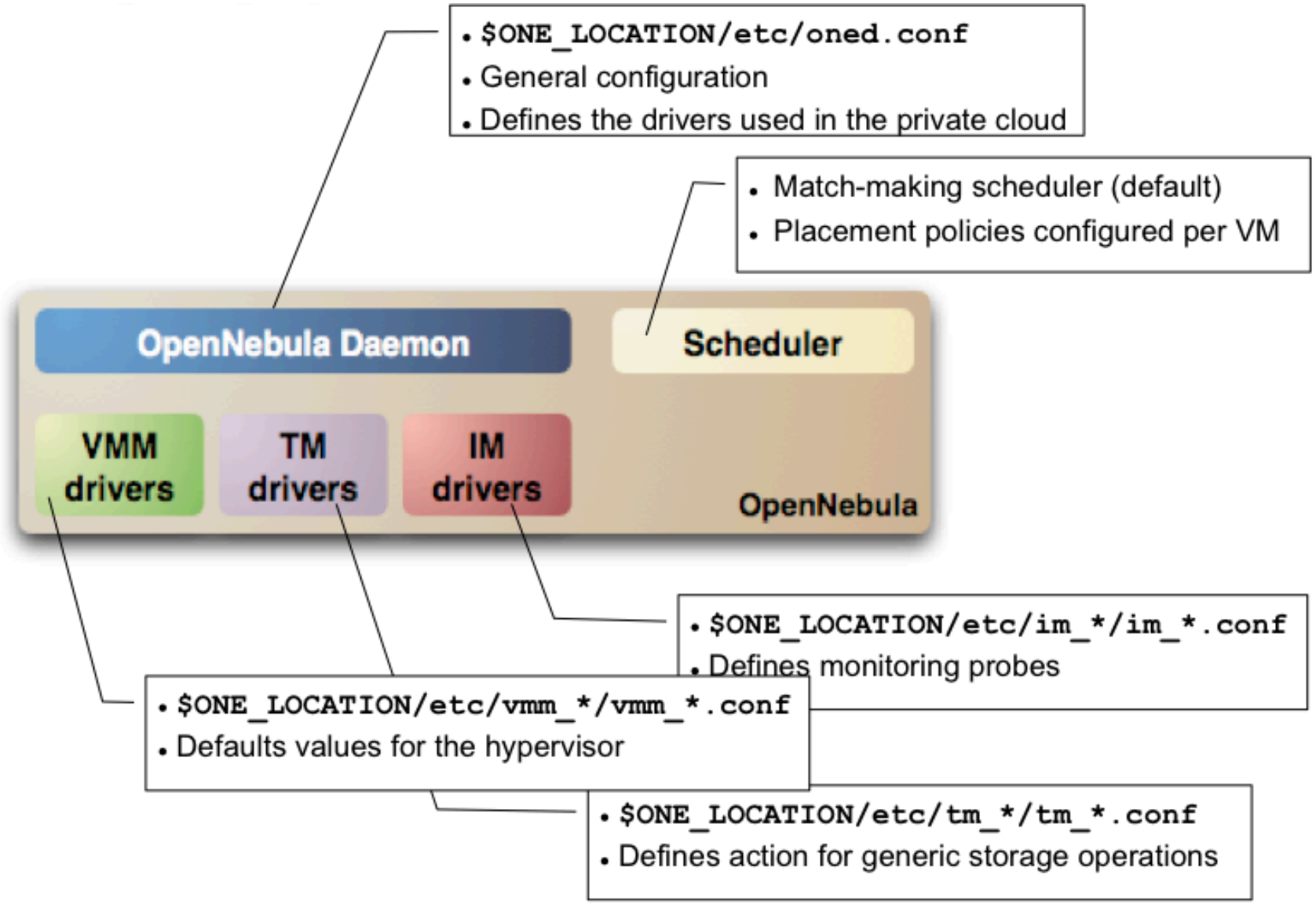
Private Cloud :: Deploying OpenNebula

Virtualization

- OpenNebula can interface various hypervisors
  - Xen
  - *KVM*
  - *VMware*
- Different functionality offered depending of the HV
  - Careful with versions
  - Libvirt abstraction used for kvm

Private Cloud :: Deploying OpenNebula

Configuration



Private Cloud :: Using OpenNebula

dsa-research.org

Private Cloud :: Using OpenNebula

Storage

- Support for Catalog due in 1.6
  - ACLs
  - Metadata
  - Native support, offered by XMLRPC API
- Meanwhile…
  - Image repository in FS
  - Plugins for
    - SSH – always clone
    - NFS – quick, possible bottleneck?

Storage

Disk description options

```
DISK = [
  type     = “floppy|disk|cdrom|swap”,
  source   = “path_to_disk_image_file|physical_dev”,
  size     = “size_in_GB”,
  target   = “device_to_map_disk”,
  bus      = “ide|scsi|virtio|xen”,
  readonly = “yes|no”,
  clone    = “yes|no”,
  save     = “path_to_disk_image_file” ]
```

Disk description examples

```
DISK = [
  source   = “/images/etch/disk.img”,
  target   = “sda” ]

DISK = [
  type     = swap,
  size     = 1024,
  target   = “sdb” ]
```

Private Cloud :: Using OpenNebula

Virtual Networks

- A Virtual Network in OpenNebula
  - Defines a separated MAC/IP address space to be used by VMs
  - Each virtual network is associated with a physical network through a bridge
  - Virtual Networks can be isolated (at layer 2 level) with ebtables and hooks
- Virtual Networks are managed with the `onevnet` utility

Networks created by oneadmin are *public, i.e.* can be used by VMs of any other user

Private Cloud :: Using OpenNebula

Virtual Networks

Ranged network definition

```
NAME            = “Private LAN”
TYPE            = RANGED
BRIDGE          = eth0
NETWORK_SIZE    = 250
NETWORK_ADDRESS= 10.0.0.0
```

Fixed network definition

```
NAME  = “Public LAN”
TYPE  = FIXED
BRIDGE= eth1
LEASES= [IP=130.10.0.1,MAC=50:20:20:20:20:20]
LEASES= [IP=130.10.0.2]
```

Network information in VM description

```
NIC = [
  network = “name_of_the_virtual_network”,
  ip      = “ip_address”,
  bridge  = “name_of_bridge_to_bind_if”,
  target  = “device_name_to_map_if”,
  mac     = “HW_address”,
  script  = “path_to_script_to_bring_up_if” ]
```

Private Cloud :: Using OpenNebula

Users

- A User in OpenNebula
  - Is a pair of username:password
- Only oneadmin can add/delete users
- Users are managed with the `oneuser` utility

Private Cloud :: Using OpenNebula

Virtual Machines

- Preparing a VM to be used with OpenNebula
  - You can use any VM prepared for the target hypervisor
  - **Hint I**: Place the vmcontext.sh script in the boot process to make better use of vlans
  - **Hint II**: Do not pack useless information in the VM images:
    - swap. OpenNebula can create swap partitions on-the-fly in the target host
    - Scratch or volatile storage. OpenNebula can create plain FS on-the-fly in the target host
  - **Hint III:** Install once and deploy many; prepare master images
  - **Hint IV:** Do not put private information (e.g. ssh keys) in the master images, use the `CONTEXT`
  - **Hint V:** Pass arbitrary data to a master image using `CONTEXT`
- Virtual Machines are managed with the `onevm` utility

Private Cloud :: Using OpenNebula

Virtual Machines

Option	Description
NAME	• Name that the VM will get for description purposes.
CPU	• Percentage of CPU divided by 100 required for the Virtual Machine.
OS (KERNEL, INITRD)	• Path of the kernel and initrd files to boot from.
DISK (SOURCE, TARGET, CLONE, TYPE)	• Description of a disk image to attach to the VM.
NIC (NETWORK)	• Definition of a virtual network the VM will be attached to.

- Multiple disk an network interfaces can be specified just adding more disk/nic statements.
- To create swap images you can specify TYPE=swap, SIZE=<*size in MB*>.
- By default disk images are cloned, if you do not want that to happen CLONE=no can be specified and the VM will attach the original image.

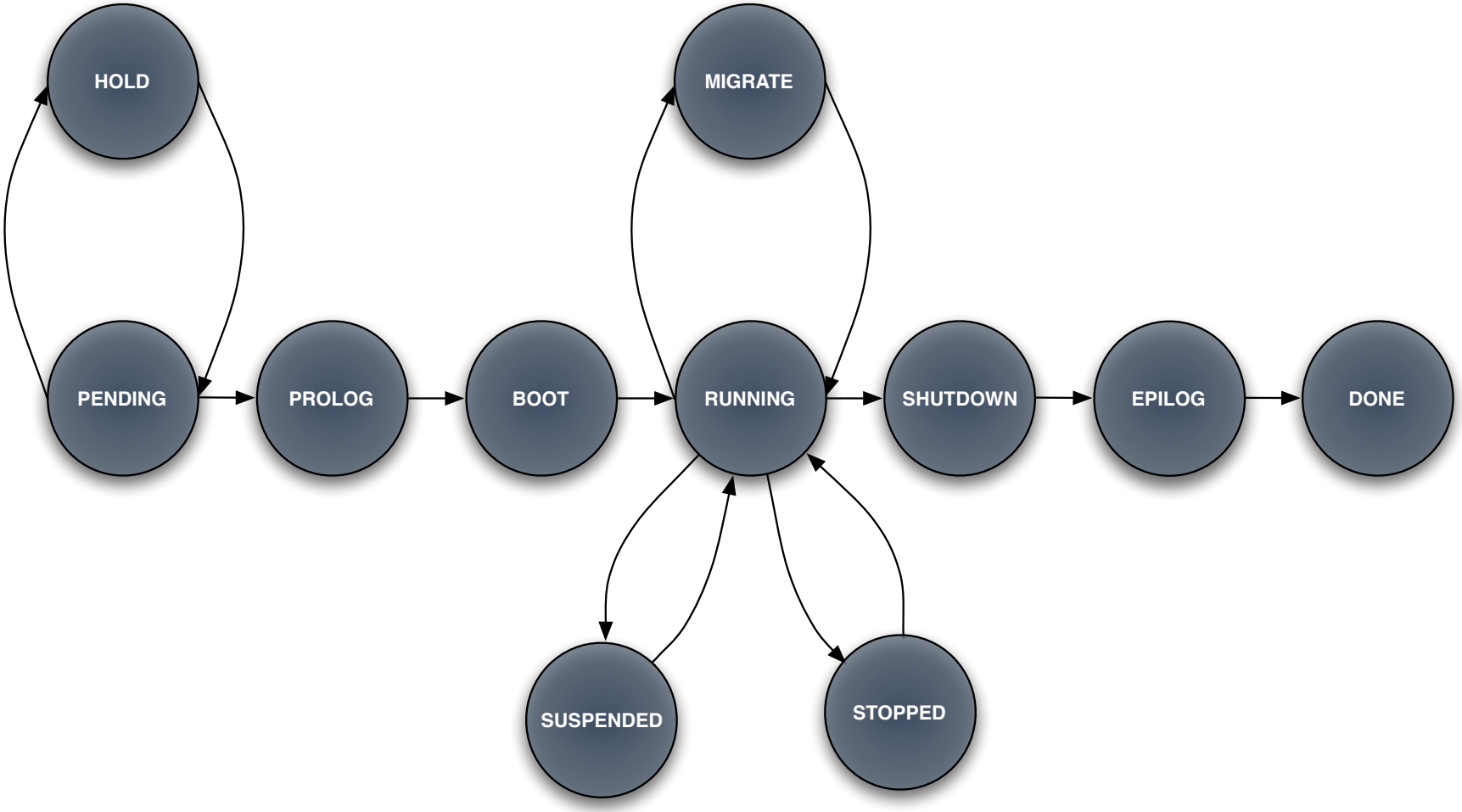
Private Cloud :: Using OpenNebula

Virtual Machines

```
NAME = vm-example
CPU    = 1
MEMORY = 512

# --- kernel & boot device ---
OS = [
  kernel   = “/vmlinuz”,
  initrd   = “/initrd.img”,
  root     = “sda” ]

# --- 2 disks ---
DISK = [
  source   = “/images/etch/disk.img”,
  target   = “sda” ]

DISK = [
  type     = swap,
  size     = 1024,
  target   = “sdb” ]

# --- 1 NIC ---
NIC = [ network=“public” ]
```

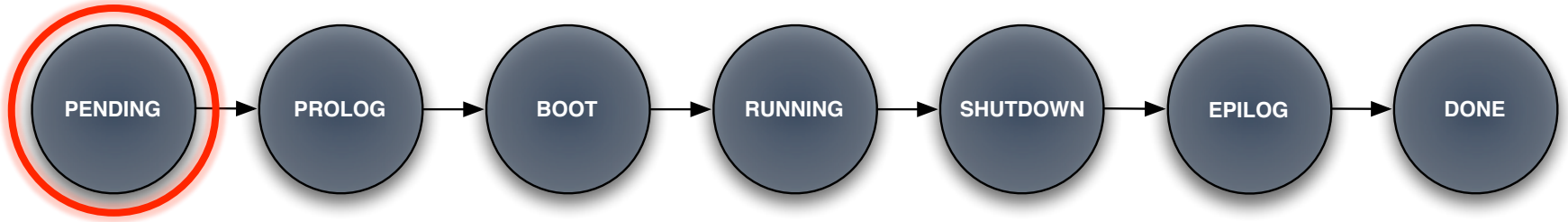
Private Cloud :: Using OpenNebula

Virtual Machines



Private Cloud :: Using OpenNebula

Pending State

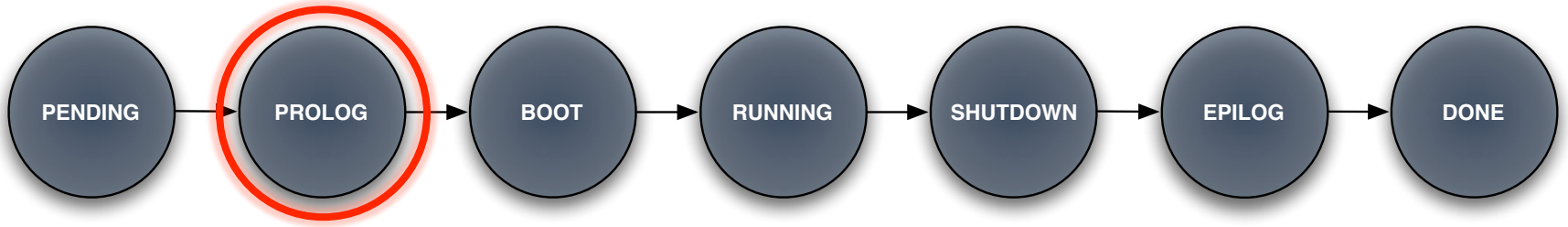


- After submitting a VM description to ONE it is added to the database and its state is set to PENDING.
- In this state IP and MAC addresses are also chosen if they are not explicitly defined.
- The scheduler awakes every 30 seconds and looks for VM descriptions in PENDING state and searches for a physical node that meets its requirements. Then a deploy XML-RPC message is sent to *oned* to make it run in the selected node.
- Deployment can be also made manually using the Command Line Interface:

```
⇒onevm deploy <vmid> <hostid>
```

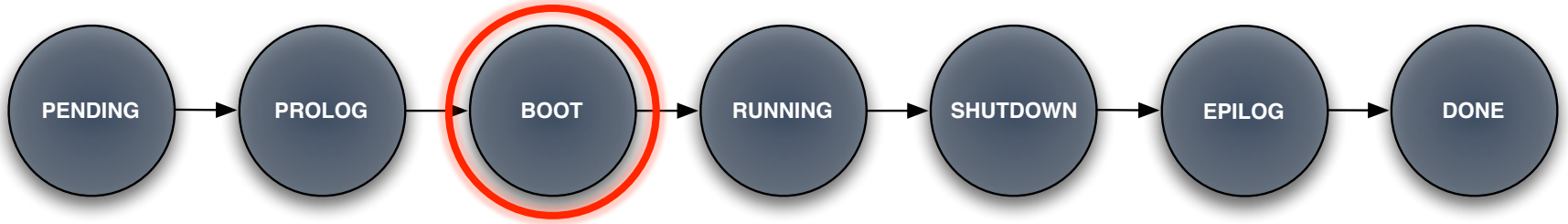
Private Cloud :: Using OpenNebula

Prolog State



- In PROLOG state the Transfer Driver prepares the images to be used by the VM.
- Transfer actions:
  - **CLONE**: Makes a copy of a disk image file to be used by the VM. If Clone option for that file is set to false and the Transfer Driver is configured for NFS then a symbolic link is created.
  - **MKSWAP**: Creates a swap disk image on the fly to be used by the VM if it is specified in the VM description.

Boot State



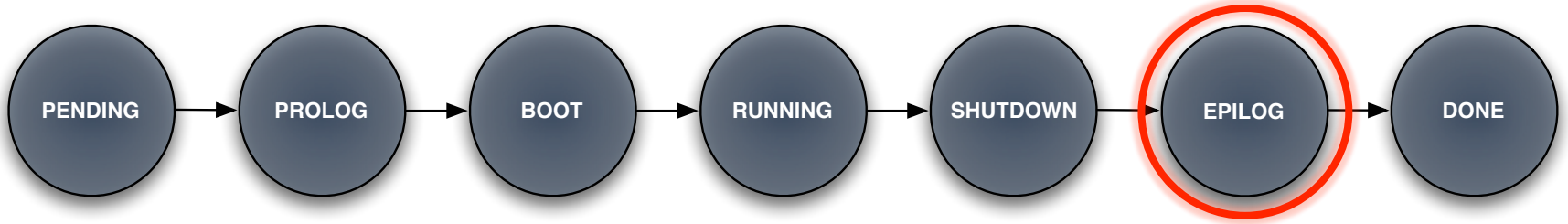
- In this state a deployment file specific for the virtualization technology configured for the physical host is generated using the information provided in the VM description file. Then Virtual Machine Driver sends deploy command to the virtual host to start the VM.
- The VM will be in this state until deployment finishes or fails.

Private Cloud :: Using OpenNebula

Running and Shutdown States

- While the VM is in RUNNING state it will be periodically polled to get its consumption and state.
- In SHUTDOWN state Virtual Machine Driver will send the shutdown command to the underlying virtual infrastructure.

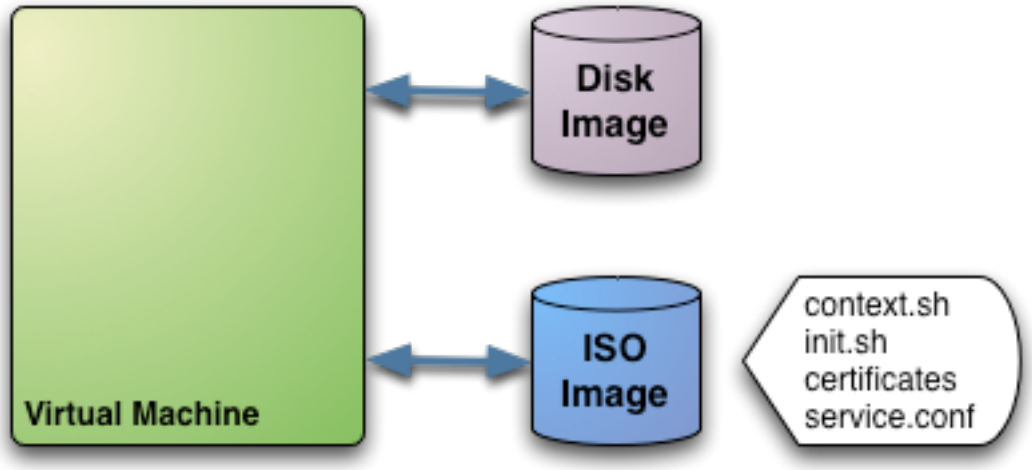
Epilog State



- In EPILOG state the Transfer Manager Driver is called again to perform this actions:
  - Copy back the images that have ***SAVE=yes*** option.
  - Delete images that were cloned or generated by **MKSWAP**.

Private Cloud :: Using OpenNebula

Contextualization



The ISO image has the contextualization for that VM
Example:

- **context.sh**: contains configuration variables
- **init.sh**: script called by VM at start to configure specific services
- **certificates**: directory that contains certificates for some service
- **service.conf**: service configuration

Private Cloud :: Using OpenNebula

Contextualization

```
CONTEXT = [
  hostname   = "$NAME",
  ip_private = '$NIC[IP, NETWORK="Private LAN"]',
  ip_gen     = "10.0.0.$VM_ID",
  files      = "/service/init.sh /service/
service.conf /service/certificates",
  target     = "sdc"
]
```

• **files**: Files and directories that will be included in the contextualization image

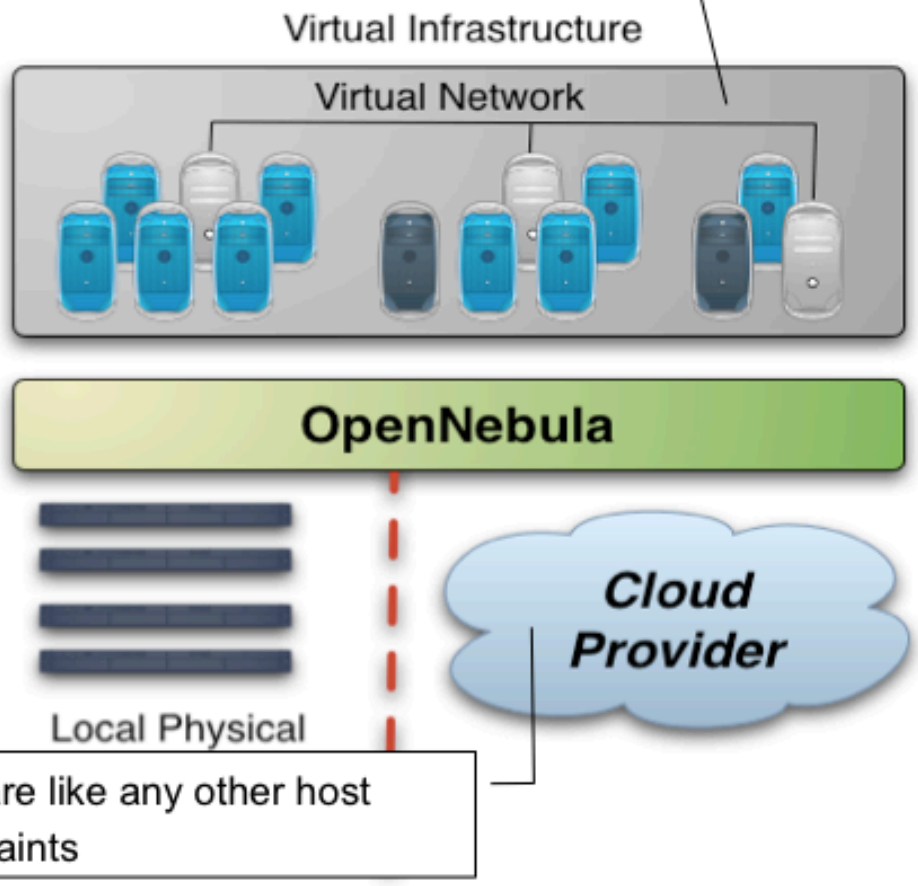
• **target:** device where the contextualization image will be available to the VM instance

Hybrid Cloud

Hybrid Cloud

Overview

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



- External Clouds are like any other host
- Placement constraints

dsa-research.org

Hybrid Cloud

Configuration

```
IM_MAD = [
  name       = "im_ec2",
  executable = "one_im_ec2",
  arguments  = "im_ec2/im_ec2.conf" ]


VM_MAD = [
  name       = "vmm_ec2",
  executable = "one_vmm_ec2",
  arguments  = "vmm_ec2/vmm_ec2.conf",
  type       = "xml" ]

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

Hybrid Cloud

Use

- Amazon EC2 cloud is managed by OpenNebula as any other cluster node
  - You can use **several accounts** by adding a driver for each account (use the arguments attribute, `-k` and `-c` options). Then create a host that uses the driver
  - You can use **multiple EC2 zones**, add a driver for each zone (use the arguments attribute, `-u` option), and a host that uses that driver
  - You can limit the use of EC2 instances by modifying the IM file

Hybrid Cloud

Use

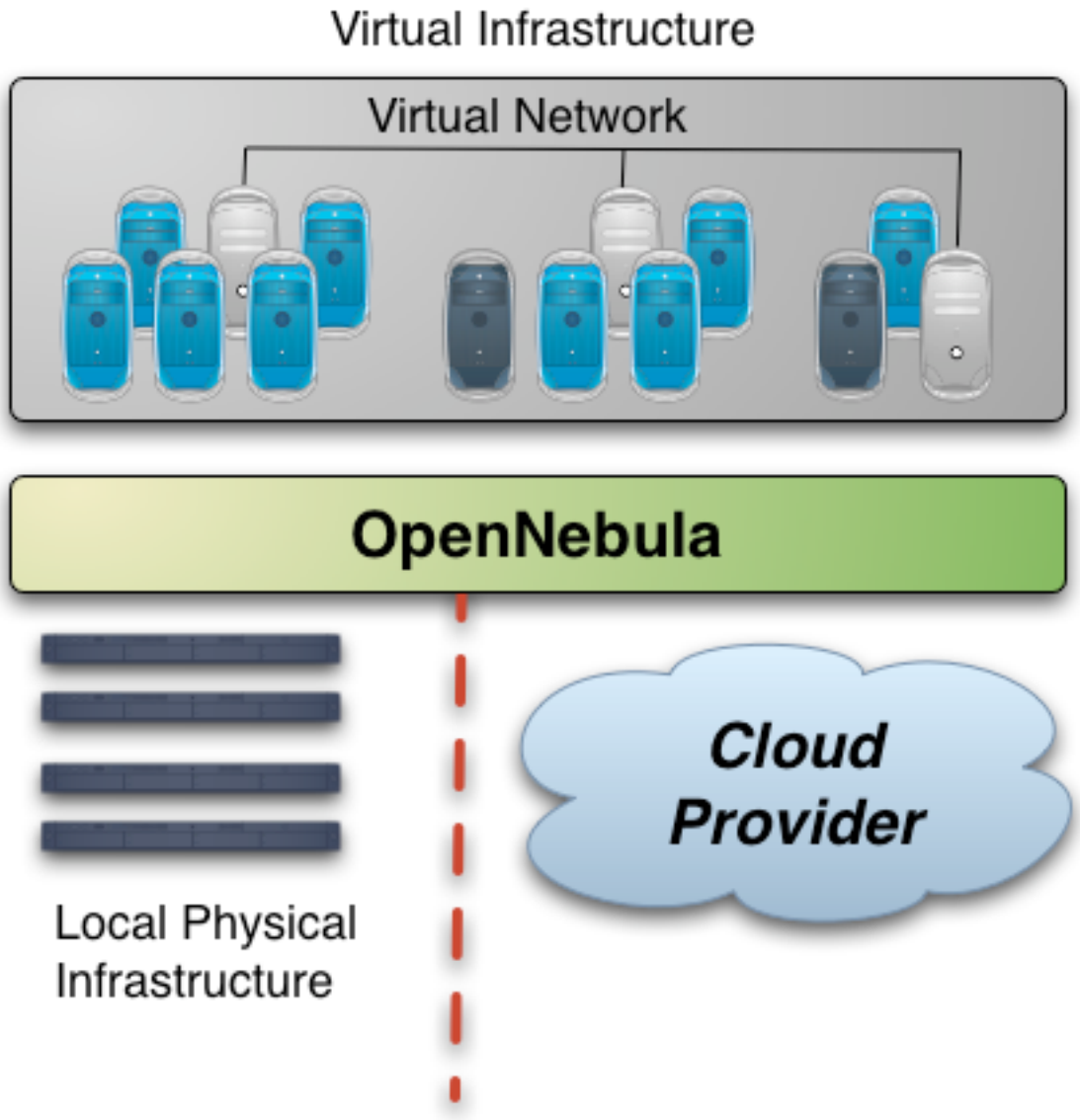
- Virtual Machines can be instantiated locally or in EC2
  - The template must provide a description for both instantiation methods.
  - The EC2 counterpart of your VM (`AMI_ID`) must be available for the driver account
  - The EC2 VM template attribute:

```
EC2 = [
  AMI             = "ami_id for this VM",
  KEYPAIR         = "the keypair to use the instance",
  AUTHORIZED_PORTS = "ports to access the instance",
  INSTANCETYPE    = "m1.small...",
  ELASTICIP       = "the elastic ip for this instance",
  CLOUD           = "host (EC2 cloud) to use this
description with"
]
```

Hybrid Cloud

Service Execution

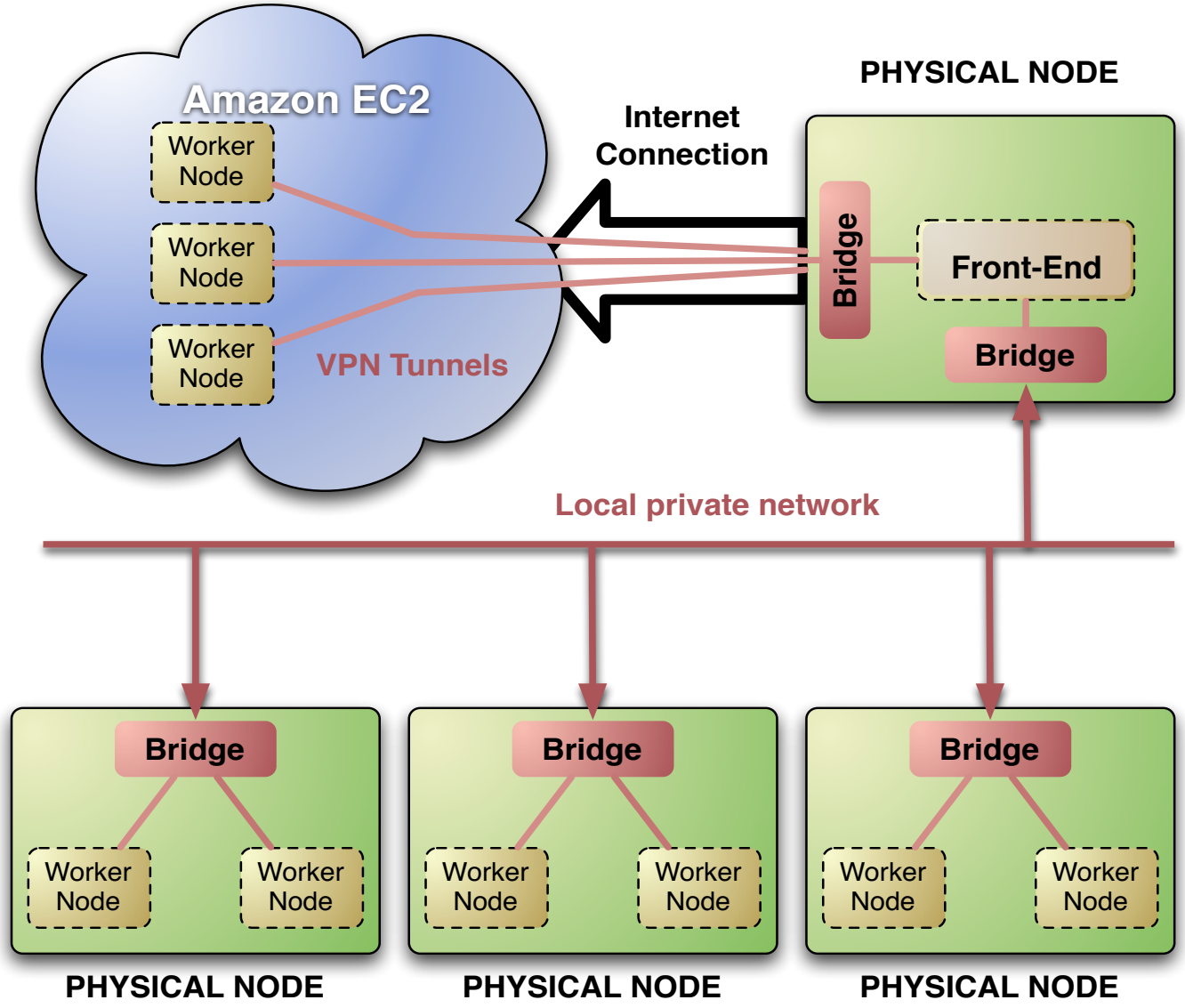
Infrastructure Perspective



Hybrid Cloud

Service Execution

Service Perspective

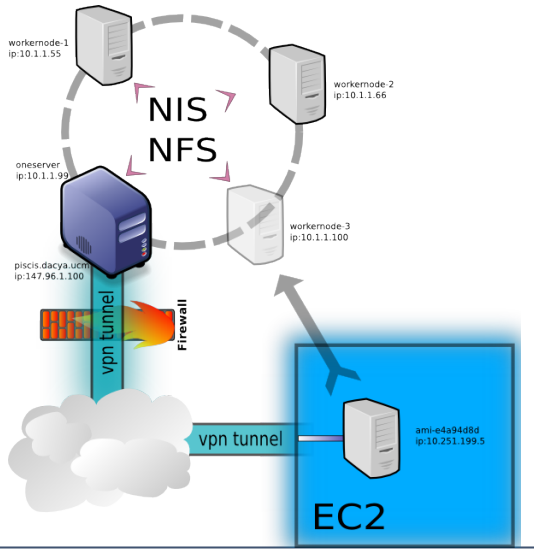


Hybrid Cloud

Use Cases

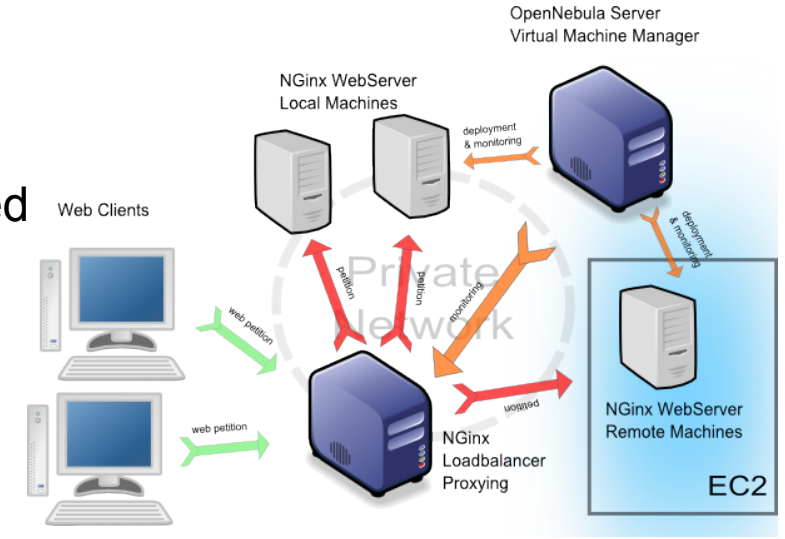
On-demand Scaling of Computing Clusters

- Elastic execution of a SGE computing cluster
- Dynamic growth of the number of worker nodes to meet demands using EC2
- Private network with NIS and NFS
- EC2 worker nodes connect via VPN



On-demand Scaling of Web Servers

- Elastic execution of the NGinx web server
- The capacity of the elastic web application can be dynamically increased or decreased by adding or removing NGinx instances



THANK YOU FOR YOUR ATTENTION!!!
More info, downloads, mailing lists at
www.OpenNebula.org

OpenNebula is partially funded by the “RESERVOIR– Resources and Services Virtualization without Barriers” project
EU grant agreement 215605

www.reservoir-fp7.eu/

Acknowledgements

- Ignacio M. Llorente
- Rubén S. Montero
- Javier Fontán
- Jaime Melis
- Rafael Moreno