

**OSDC 2012**  
24<sup>th</sup> April, Nürnberg

# Building Clouds with OpenNebula 3.4

**Constantino Vázquez Blanco**  
**[dsa-research.org](http://dsa-research.org) | [OpenNebula.org](http://OpenNebula.org)**

Distributed Systems Architecture Research Group  
Universidad Complutense de Madrid

# Building Clouds with OpenNebula 3.4

## *Customizing your Cloud*

**Constantino Vázquez Blanco**

**[dsa-research.org](http://dsa-research.org) | [OpenNebula.org](http://OpenNebula.org)**

Distributed Systems Architecture Research Group

Universidad Complutense de Madrid



- Adapting & Customizing OpenNebula
- The Storage Subsystem
- The Information Subsystem
- Using Hooks
- Fault Tolerance
- Network Isolation
- More Customization

# Customizing & Extending OpenNebula

---

## Overview

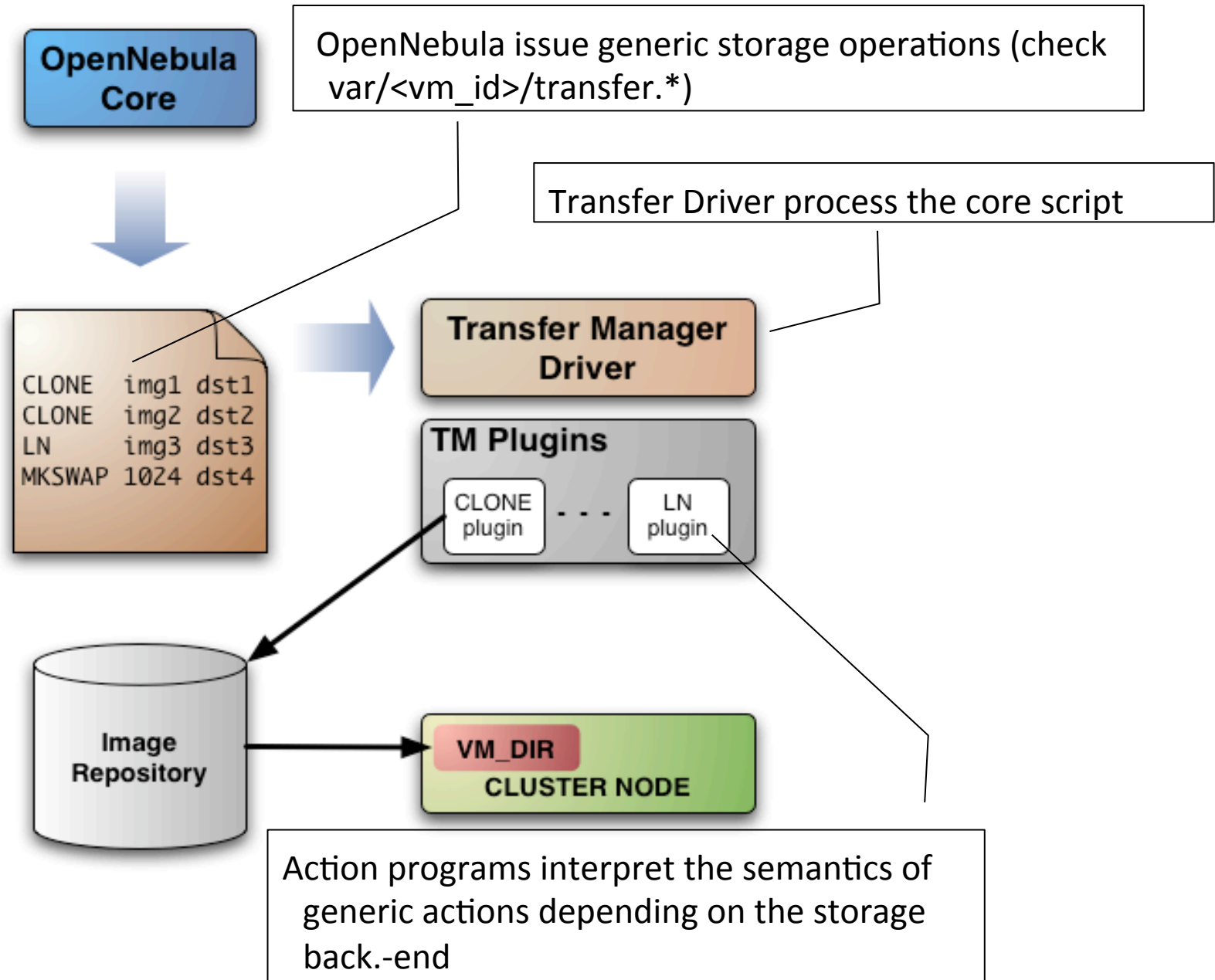
- You can **customize** your cloud by:
  - Tuning or adapting the transfer operations to your **storage back-end**
  - Adding new **monitorization** probes to improve the VM placement
  - Adjusting VM operations to your hypervisor installation
  - Trigger **custom actions** on specific VM events (e.g. “on VM creation update the accounting DB” or “on VM shutdown send an email”)
- You can **extend** your cloud by:
  - Developing new drivers for other **hypervisors**
  - Developing new drivers for other **storage back-ends**
  - Developing **Cloud applications** using the OpenNebula API or the Cloud APIs



OpenNebula is very scripting friendly, drivers can be written in any language. You can modify the current ones or use them as templates for new ones.

# The Storage Backend

## Overview



# The Storage Backend

---

## Transfer Manager Operations

- OpenNebula requests the following **abstract operations** over a VM image
  - **CLONE**: This action will basically make a copy of the image from ORIGIN to DESTINATION.
  - **LN**: Creates a symbolic link in DESTINATION that points to ORIGIN
  - **MKSWAP**: Generates a swap image in DESTINATION. The size is given in ORIGIN in MB.
  - **MKIMAGE**: Creates a disk image in DESTINATION and populates it with the files inside ORIGIN directory.
  - **DELETE**: Deletes ORIGIN file or directory.
  - **MVDS**: Move files to/from datastore
  - **MV**: Moves ORIGIN to DESTINATION.

# The Storage Backend

---

## The Transfer Manager Action Scripts

- Actions are contained in scripts under  
`$ONE_LOCATION/var/remotes/tm/ssh`

```
$ find $ONE_LOCATION/var/remotes/tm/ssh  
./clone  
./context  
./delete  
./ln  
./mkimage  
./mkswap  
./mv  
./mvds
```

# The Storage Backend

---

Example, Customize your storage

- **Hands on!**

- Study the **clone** script
- Check the semantics of Storage operations for the ssh TM
- Delete and create a host using ssh backend
- Example 1: Make swap partitions in local storage
  - Take a look to **mkswap** from ssh
  - Link the swap partition to original DST path
- **Example 2:** Make the clone script aware of compressed images (\*.gz)

# The Storage Backend

## Example, Customize your storage

```
$ cat $ONE_LOCATION/var/remotes/tm/ssh/clone

ssh_make_path $DST_HOST $DST_DIR

#-----
# Copy files to the remote host
#-----
case $SRC in
http://*)
    log "Downloading $SRC"
    RMT_CMD="$WGET -O $DST_PATH $SRC"
    ssh_exec_and_log "$DST_HOST" "$RMT_CMD" "Error downloading $SRC"
    ;;

*)
    log "Cloning $SRC in $DST_PATH"
    exec_and_log "$SCP -r $SRC $DST" "Error copying $SRC to $DST"
    ;;
```



# The Information System

---

## Overview

- OpenNebula gets host information by executing an arbitrary number of **probes**:
  - Program that returns a monitor metric (**METRIC\_NAME = VALUE**)
  - **Placed** in `$ONE_LOCATION/var/remotes/im/<hypervisor>.d`
- Monitor probes can be **executed**:
  - **Remotely** using the `one_im_ssh` driver
  - **Locally** to integrate it with an external monitor system (Ganglia, Nagios...)
- **Remote probes** are cached at the cluster nodes and sync with `onehost sync` command (next time the host is monitored)
- Probe information is mainly used for **VM placement**

# The Information System

---

## Example, customize the Monitor Information

- **Hands on!**

- Study & execute (run\_probes kvm) the probes for kvm at \$ONE\_LOCATION/var/remotes/im/kvm.d
- Create a new monitor probe:
  - Compute the number of running VMS (e.g. MY\_RVMS), use virsh list, pgrep kvm...
  - Sync the cluster nodes
  - Use the new metric for VM placement

# Hooks

---

## Overview

- Hooks **are custom programs** executed:
  - Locally (front-end) or remotely (target host)
  - Upon VM related events or Host related events
- OpenNebula **includes**:
  - **Fault Tolerance** Hooks, to restart VMs when a host crashes or restart VMs when it fails
  - **Network Isolation** Hooks, to setup VLANs
- Hooks are **defined in oned.conf**
  - **Name**, of the hook
  - **command**, to be executed
  - **arguments**
  - **on**, event that triggers the hook
  - **remote**

# Hooks: Fault Tolerance

## Configuring Fault Tolerance Hooks

- Host Hook (on ERROR)
  - resubmit (-r) or delete (-d) VMs in the host
  - Including suspended (y) VMs or not (n)

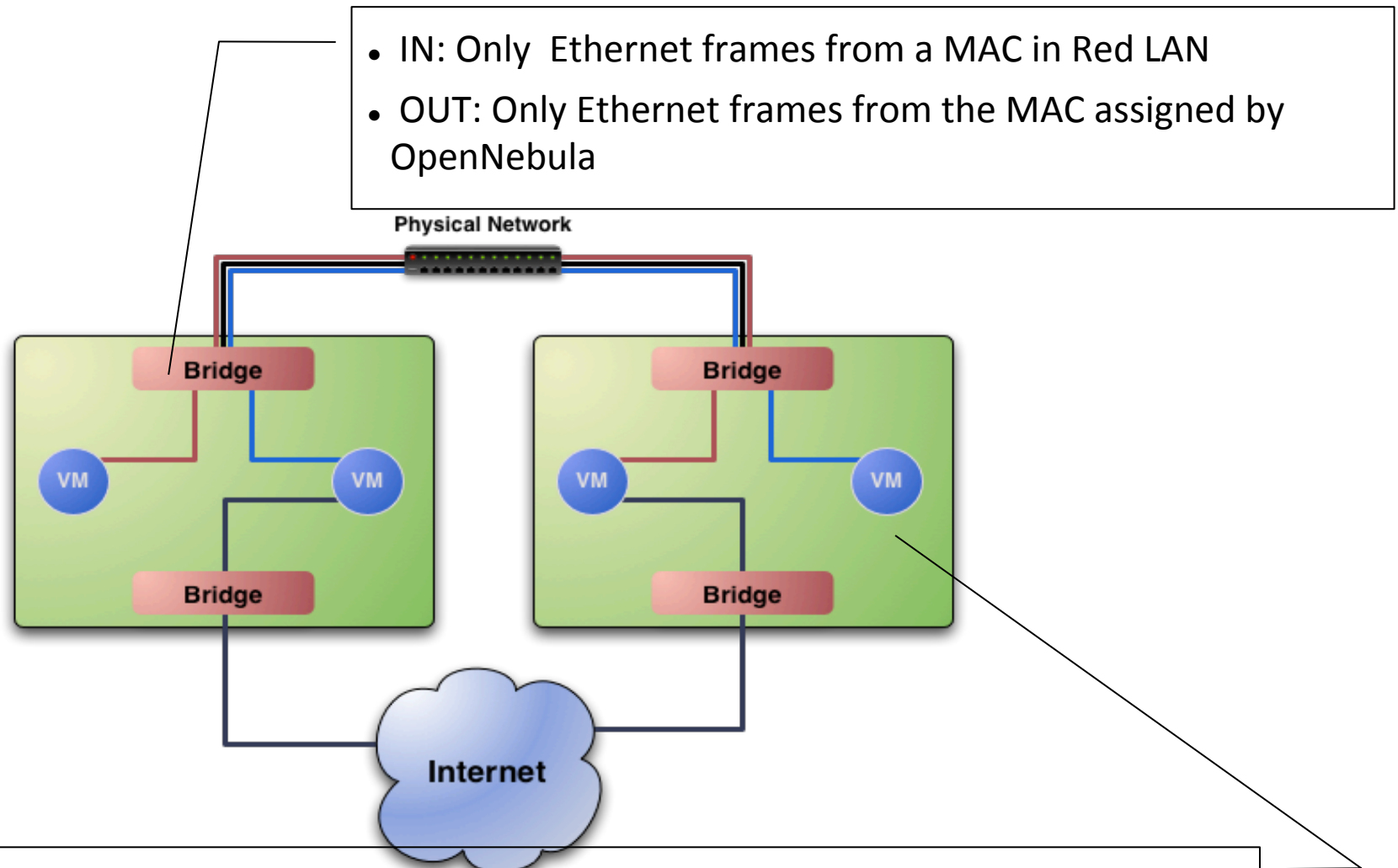
```
HOST_HOOK = [  
  name      = "error",  
  on        = "ERROR",  
  command   = "host_error.rb",  
  arguments = "$HID -r n",  
  remote    = no ]
```

- VM Hook (on FAILURE)

```
VM_HOOK = [  
  name      = "on_failure_resubmit",  
  on        = "FAILURE",  
  command   = "onevm resubmit",  
  arguments = "$VMID" ]
```

# Hooks: Network Isolation

## Overview



- Networks are isolated at layer 2
- You can put any TCP/IP service as part of the VMs (e.g. DHCP, nagios...)

# Hooks: Network Isolation

## Configuration

- Requirements (this has to be done in all the cluster nodes)
  - Check that ebtables package is installed
  - Allow oneadmin to use the ebtables command through sudo

```
#visudo
...
oneadmin    ALL=(ALL) NOPASSWD: /sbin/ebtables *
...
```

- Configure the hooks for OpenNebula

```
VM_HOOK = [
  name      = "ebtables-start",
  on        = "running",
  command   = "/srv/cloud/one/share/hooks/ebtables-kvm",
  arguments = "one-$VMID",
  remote    = "yes" ]
VM_HOOK = [
  name      = "ebtables-flush",
  ...
```

# Hooks: Network Isolation

---

## Practical example

### Hands on!

- Start a couple of VMs in Networks Red and Blue.
  - Check the ebtables rules in the hosts
  - Check connectivity between VMs
  - Change the network mask of the VMs and check connectivity
  - Shutdown and check the ebtables rules

# Network Isolation

---

## With the virtual network manager

- **dummy**: Default driver that doesn't perform any network operation. Firewalling rules are also ignored.
- **fw**: Firewall rules are applied, but networking isolation is ignored.
- **802.1Q**: restrict network access through VLAN tagging, which also requires support from the hardware switches.
- **ebtables**: restrict network access through Ebtables rules. No special hardware configuration required.
- **ovswitch**: restrict network access with Open vSwitch Virtual Switch.
- **VMware**: uses the VMware networking infrastructure to provide an isolated and 802.1Q compatible network for VMs launched with the VMware hypervisor.



# More Customization

---

## Other Components that can be adapted

- **Authorization & Authentication**, can be performed with external drivers
  - ssh-keys based authentication
  - ldap based authentication
- **DB Backend**, use MySQL for more performing setups
- **Use Ganglia**, for better scalability
- **Develop** your own components/applications (PaaS,SaaS)
  - XML-RPC interface
  - OCA (Ruby, JAVA and Python – contributed - bindings)
  - EC2 or OCCI APIs

# Building Clouds with OpenNebula 3.4

## *Customizing your Cloud*

**Constantino Vázquez Blanco**

**[dsa-research.org](http://dsa-research.org) | [OpenNebula.org](http://OpenNebula.org)**

Distributed Systems Architecture Research Group

Universidad Complutense de Madrid



- Adapting & Customizing OpenNebula
- The Storage Subsystem
- The Information Subsystem
- Using Hooks
- Fault Tolerance
- Network Isolation
- More Customization

# Building Clouds with OpenNebula 3.4

---

Summary, You should be able by now to ....

- Describe the benefits and characteristics of virtual infrastructures and IaaS clouds
- Describe the characteristics and architecture of the different clouds that can be deployed with OpenNebula 2.2
- Plan and architect a private cloud
- Design, Use and Manage Virtual infrastructures (cloud applications)
- Build public and hybrid clouds
- Adapt OpenNebula 3.4 to your datacenter