# Hands on Virtualization with Ganeti

Lance Albertson
Associate Director of Operations
OSU Open Source Lab

# About us

- OSU Open Source Lab
- Server hosting for Open Source Projects
- Open Source development projects

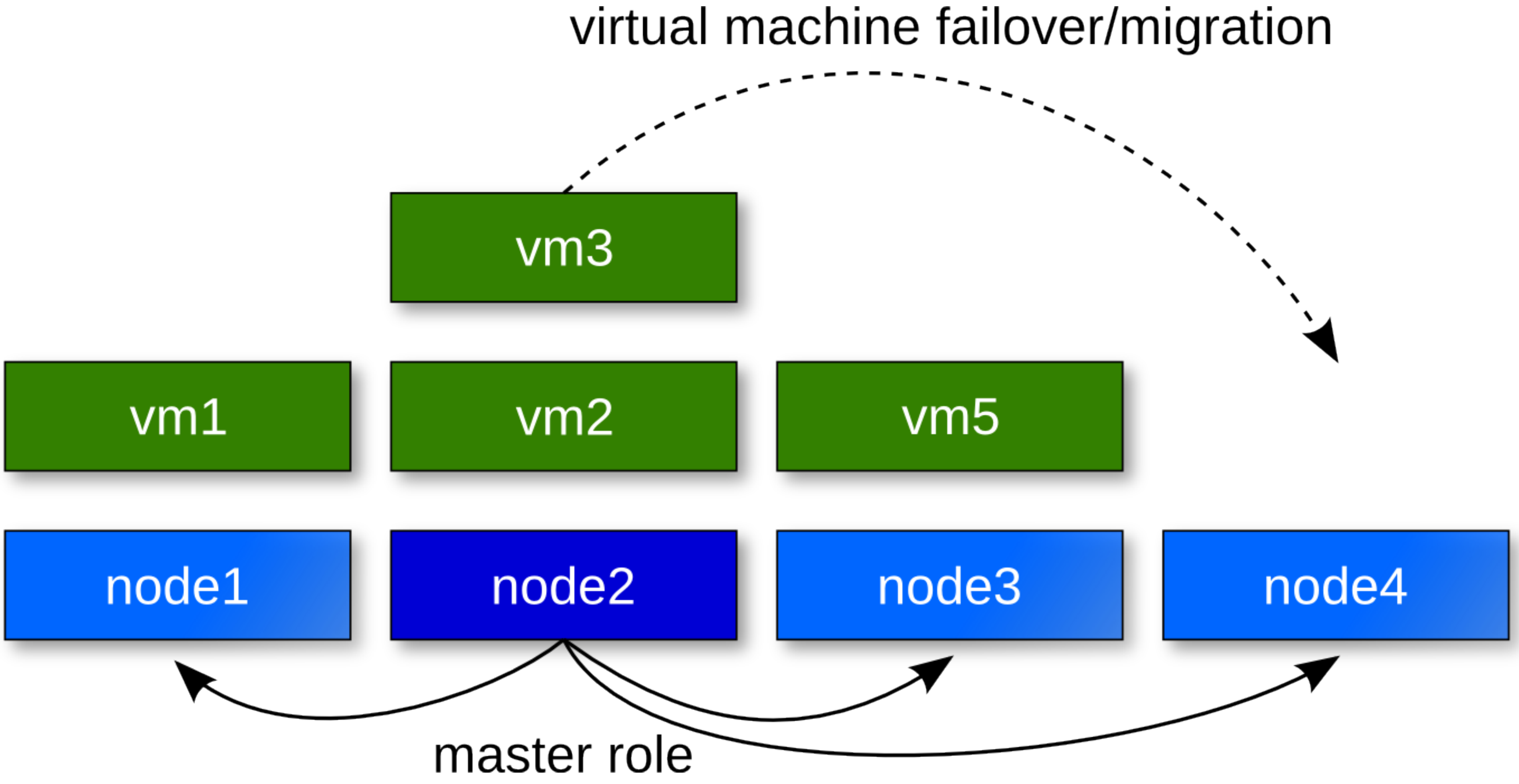
# How we use Ganeti

- *Powers* all OSUOSL virtualization
- Project hosting
- *KVM* based
- *Hundreds* of VMs
- Web hosts, code hosting, etc

# Talk Overview

- Ganeti Architecture
- Demo
- Cluster Management
- Dealing with failures
- Ganeti Web Manager

# Ganeti Cluster



# What is Ganeti?

- *Cluster* virtual server management software tool
- Built on top of *existing* OSS hypervisors
- Fast & simple *recovery* after physical failures
- Using *cheap* commodity hardware
- Private *IaaS*

# Comparing Ganeti

- Primarily utilizes *local* storage
- Built to deal with *hardware failures*
- *Mature* project
- Low package requirements
- Easily *pluggable* via hooks & RAPI

# Project Background

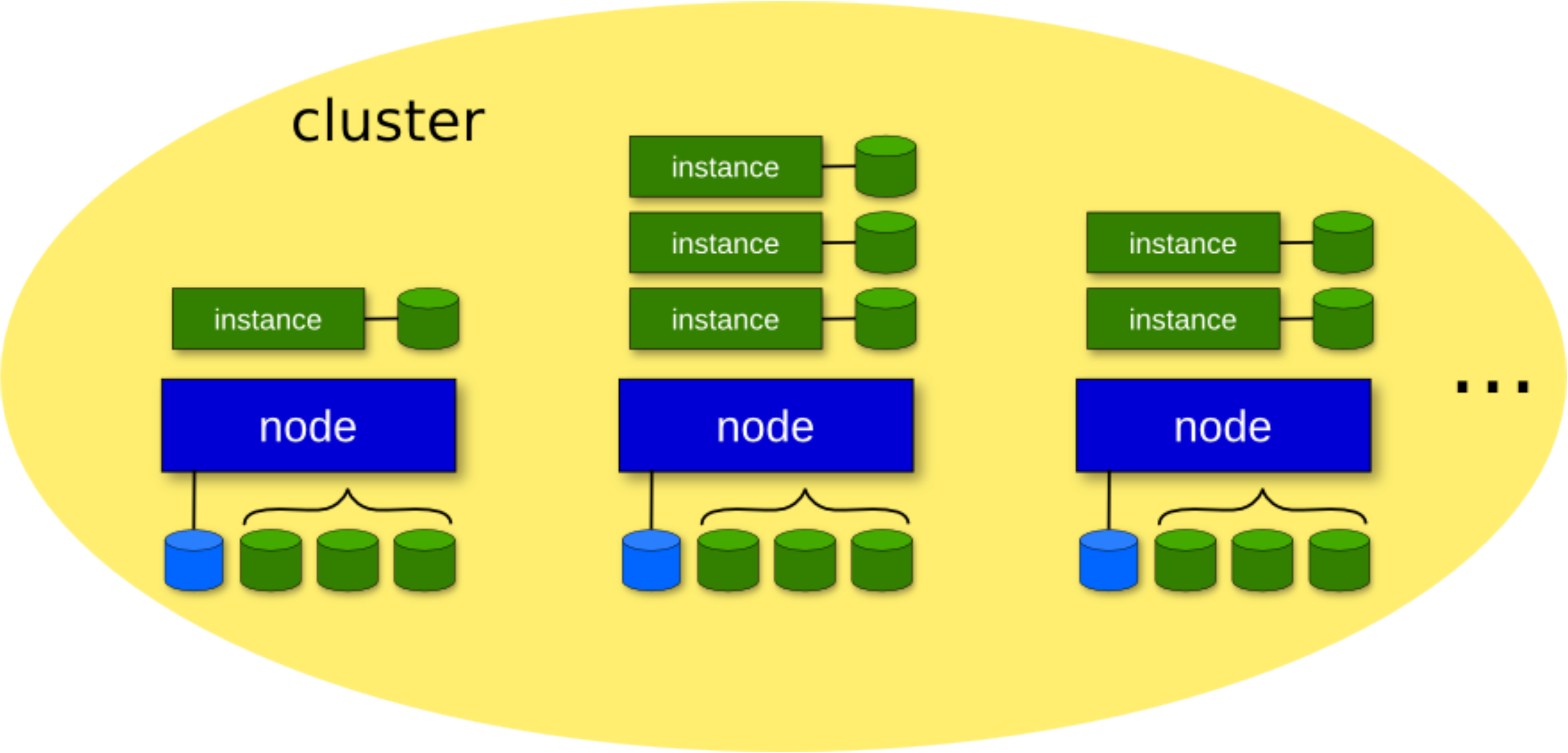
- *Google* funded project
- Used in internal corporate env
- Open Sourced in 2007 *GPLv2*
- Team based in Google Switzerland
- Active mailing list & IRC channel
- Started internally before *libvirt*

# Terminology

# Components

Python
Haskell
DRBD
LVM
Hypervisor

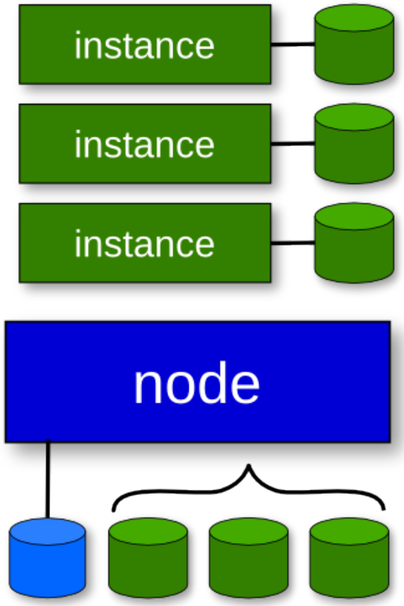
# Architecture



# Nodes

- *Physical* machine
- Fault tolerance not *required*
- Added/removed *at will* from cluster
- No *data loss* with loss of node

# Instances



- Virtual machine that *runs* on the cluster
- *fault tolerant/HA* entity within cluster

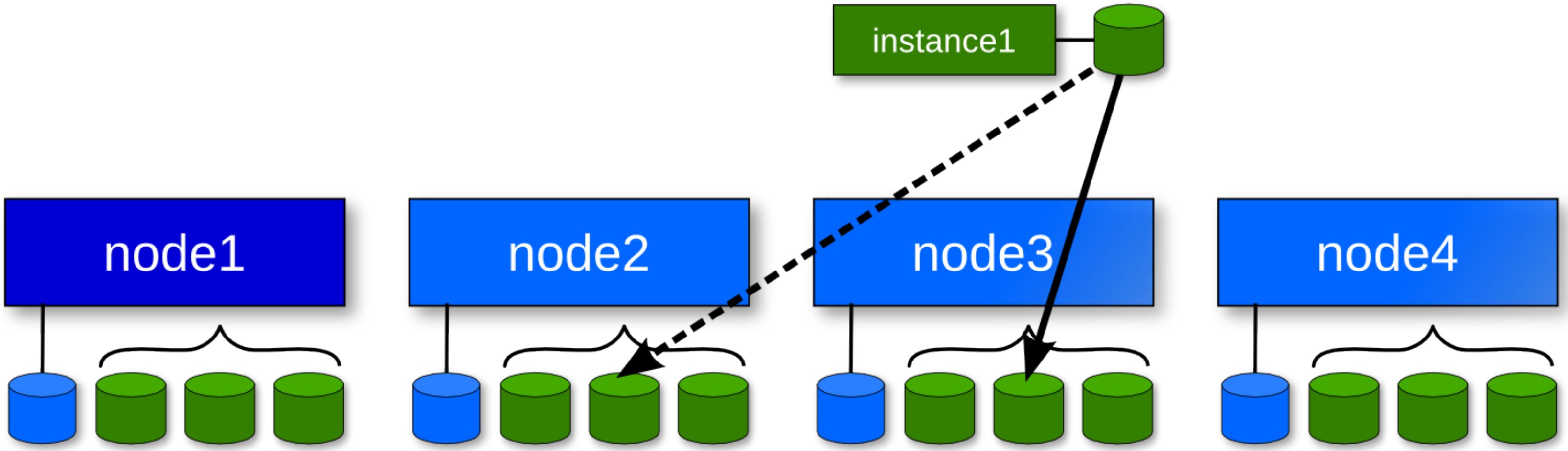
# Disk Template

- **drbd** : LVM + DRBD between 2 nodes
- **plain** : LVM w/ no redundancy
- **file** : Plain files, no redundancy
- **diskless** : Special purposes

# IAllocator

- Automatic placement of instances
- Eliminates manual node specification
- htools
- External scripts used to compute

# Primary & Secondary Concepts



- Instances always runs on *primary*
- Uses secondary node for *disk replication*
- Depends on *disk template* (i.e. drbd)

# Pre-installation Steps

# Operating System Setup

- Clean, minimal system install
- Minimum *20GB* system volume
- *Single* LVM Volume Group for instances
- 64bit is preferred
- *Similar* hardware/software configuration across nodes

# Partition Setup

## typical layout

| | | |
|---|---|---|
| /dev/sda1 | /boot | 200M |
| /dev/sda2 | / | 10-20G |
| /dev/sda3 | LVM | rest, named ganeti |

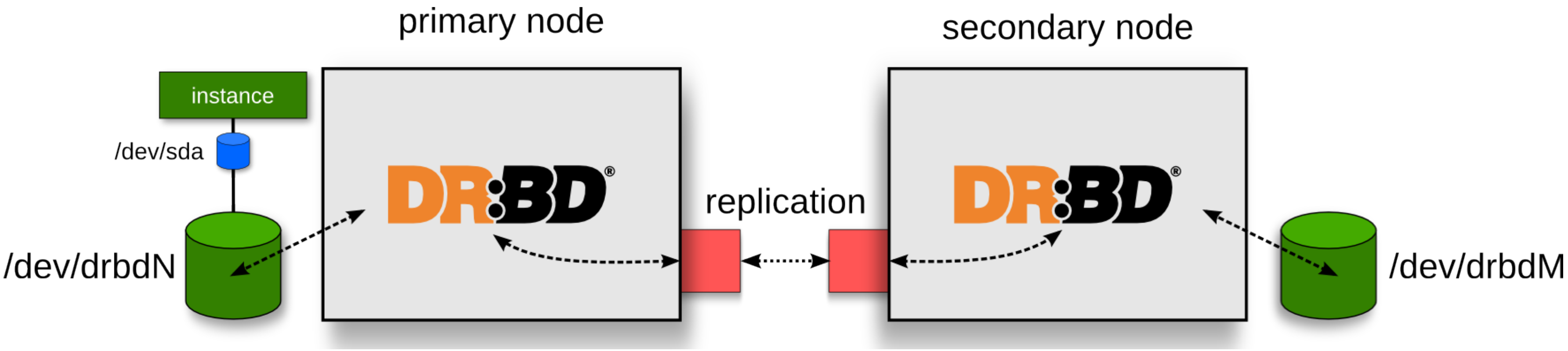
# Hostname Issues

- Requires hostname to be the **FQDN**
- i.e. *node1.example.com* instead of *node1*
- **`hostname --fqdn`** requires resolver library
- Reduce dependency on DNS and *guessing*

# Hypervisor requirements

- **Mandatory** on all nodes
- Xen 3.0 and above
- KVM 0.11 and above
- Install via your distro

# DRBD Architecture

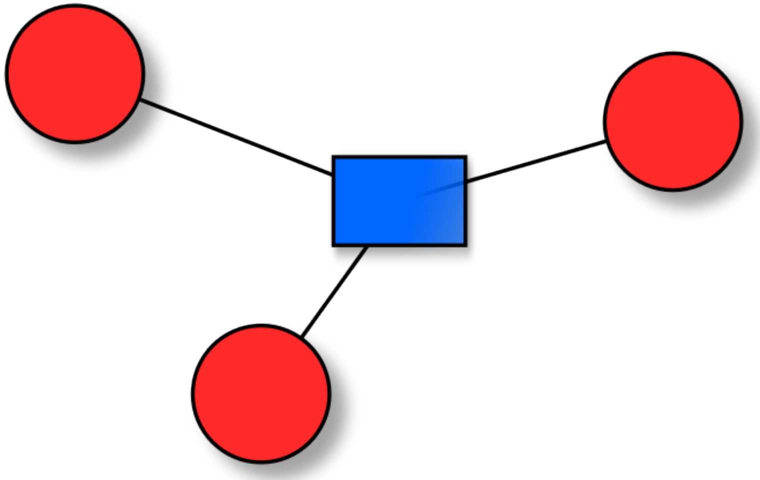


RAID1 over the network

# Installing DRBD

- Required for *high availability*
- Can *upgrade* non-HA to DRBD later
- Need at least *>=drbd-8.0.12*
- Depends on distro Support
- Included in *mainline*

# Interface Layout



- **eth0** - trunked VLANs
- **eth1** - private DRBD network

# What gets installed

- Python libraries under the *ganeti* namespace
- Set of programs under /usr/local/sbin or /usr/sbin
- Set of tools under lib/ganeti/tools directory
- IAllocator scripts under lib/ganeti/tools directory
- *Cron job* needed for cluster maintenance
- *Init script* for Ganeti daemons

# Install OS Definition

# Instance creation scripts

*also known as OS Definitions*

- Requires Operating System installation script
- Provide scripts to deploy various operating systems
- *Ganeti Instance Debootstrap* - upstream supported
- *Ganeti Instance Image* - written by me

# OS Variants

- *Variants* of the OS Definition
- Used for *defining* guest operating system
- Types of deployment settings:
  - Filesystem
  - Image directory
  - Image Name

# Ganeti Initialization

# Cluster name

**Mandatory** once per cluster, on the first node.

- Cluster hostname *resolvable* by all nodes
- IP reserved **exclusively** for the cluster
- Used by *master* node
- i.e.: ganeti.example.org

# Testing Ganeti

# Testing / Viewing the nodes

```
$ gnt-node list
Node              DTotal  DFree MTotal MNode MFree Pinst Sinst
node1.example.org 223.4G 223.4G   7.8G  300M  7.5G     0     0
node2.example.org 223.4G 223.4G   7.8G  300M  7.5G     0     0
```

- Ganeti damons can talk to each other
- Ganeti can examine storage on the nodes *(DTotal/DFree)*
- Ganeti can talk to the selected hypervisor *(MTotal/MNode/MFree)*

# Cluster burn-in testing

```
$ /usr/lib/ganeti/tools/burnin -o image -p instance{1..5}
```

- Does the *hardware* work?
- Can the *Hypervisor* create instances?
- Does each *operation* work properly?

# Adding an instance

Requires at least 5 params

- OS for the instance (`gnt-os list`)
- Disk template
- Disk count & size
- Node or iallocator
- Instance name (*resolvable*)

# Deploying VMs

# Add Command

```
$ gnt-instance add \
    -n TARGET_NODE:SECONDARY_NODE \
    -o OS_TYPE \
    -t DISK_TEMPLATE -s DISK_SIZE \
    INSTANCE_NAME
```

# Other options

- Memory size (`-B memory=1GB`)
- Number of virtual CPUs (`-B vcpus=4`)
- NIC settings (`--nic 0:link=br100`)
- `batch-create`
- See `gnt-instance` manpage for others

# Instance Removal

```
$ gnt-instance remove INSTANCE_NAME
```

# Startup / Shutdown

```
$ gnt-instance startup INSTANCE_NAME
$ gnt-instance shutdown INSTANCE_NAME
```

- Started automatically
- Do not use hypervisor directly

# Querying Instances

- Two methods:
  - listing instances
  - detailed instance information
- One useful for grep
- Other has more details, slower

# Export / Import

```
$ gnt-backup export -n TARGET_NODE INSTANCE_NAME
```

- Create *snapshot* of disk & configuration
- Backup, or import into another cluster
- *One* snapshot for an instance

# Importing an instance

```
$ gnt-backup import \
    -n TARGET_NODE \
    --src-node=NODE \
    --src-dir=DIR INSTANCE_NAME
```

# Import of foreign instances

```
$ gnt-instance add -t plain -n HOME_NODE ... \
    --disk 0:adopt=lv_name[,vg=vg_name] \
    INSTANCE_NAME
```

- Already stored as LVM volumes
- Ensure non-managed instance is stopped
- Take over given logical volumes
- Better transition

# Conversion of an instance's disk type

```
# start with a non-redundant instance
gnt-instance add -t plain ... INSTANCE

# later convert it to redundant
gnt-instance stop INSTANCE
gnt-instance modify -t drbd \
    -n NEW_SECONDARY INSTANCE
gnt-instance start INSTANCE

# and convert it back
gnt-instance stop INSTANCE
gnt-instance modify -t plain INSTANCE
gnt-instance start INSTANCE
```

# Node level operations

```
gnt-node migrate NODE
gnt-node evacuate NODE
```

# Instance Console

```
gnt-instance console INSTANCE_NAME
```

Type ^] when done, to exit.

# Using Htools

# Htools Components

- Automatic allocation
- **hbal** : Cluster rebalancer
- **hail** : IAllocator script
- **hspace** : Cluster capacity estimator

# Other topics...

- Node groups
- OOB Management
- Remote API

# Hands-on Demo

# Ganeti Web Manager

# Conclusion

# Questions?

Lance Albertson
lance@osuosl.org
@ramereth
http://lancealbertson.com

http://code.google.com/p/ganeti/

http://code.osuosl.org/projects/ganeti-webmgr

https://github.com/ramereth/vagrant-ganeti