Hands on Virtualization with Ganeti

Lance Albertson

Peter Krenesky

http://is.gd/osconganeti | http://is.gd/osconganetipdf

About us

OSU Open Source Lab

Server hosting for Open Source Projects

Open Source development projects

Lance / Lead Systems Administrator

Peter / Lead Software Engineer

How we use Ganeti

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

Tutorial Overview

- Ganeti Architecture
- Installation
- Virtual machine deployment
- Cluster Management
- Dealing with failures
- Ganeti Web Manager

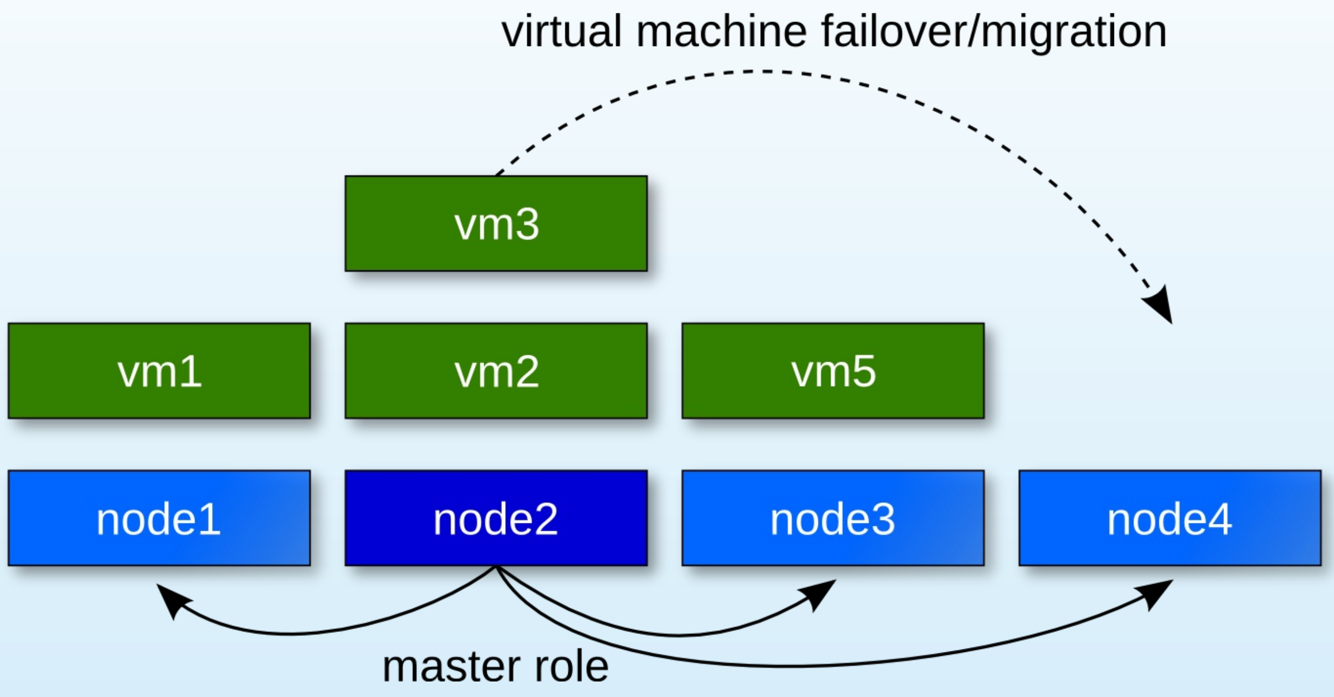
Hands-on Tutorial

- *Debian* VMs with VirtualBox
- Pre-setup already using *Puppet*
- Setup Guide PDF
- Hands-on is *optional*

Importing VMs

- Install *VirtualBox*
- Import *node1/2* (node3 is optional)
- *USB drives* are available with images

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

- 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

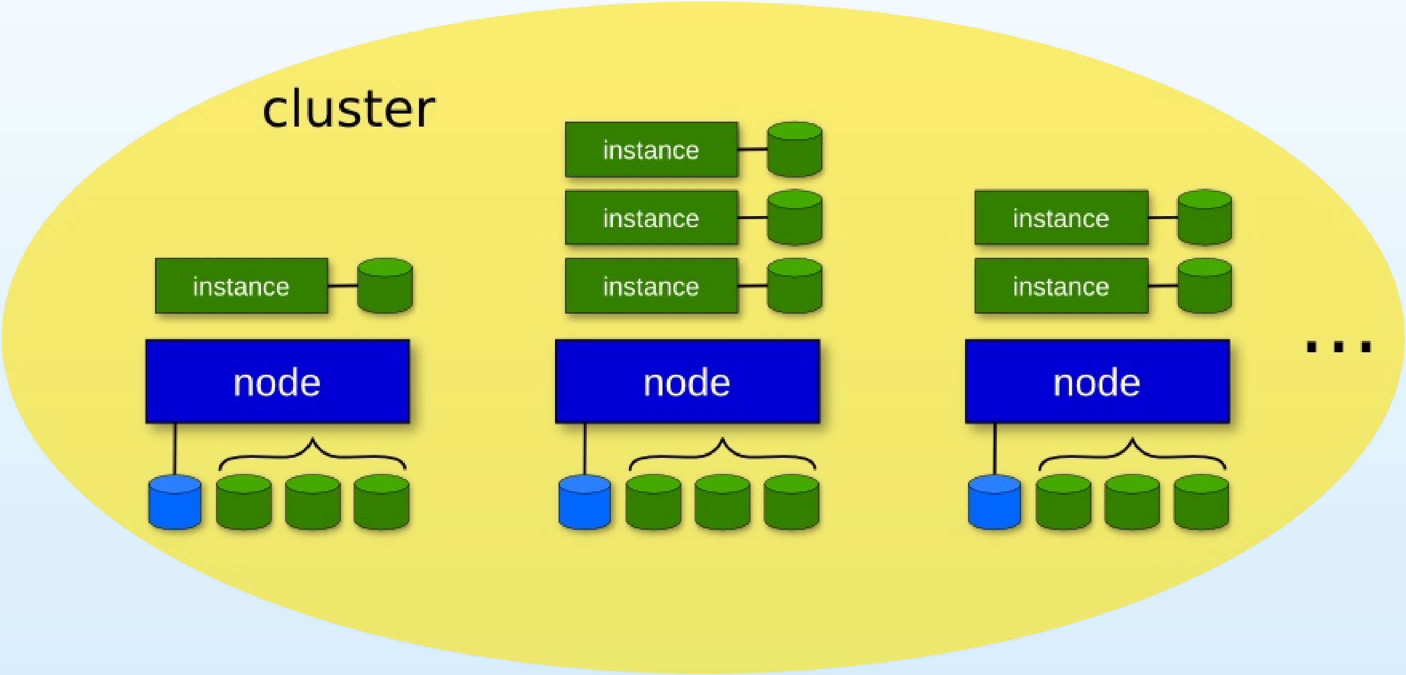
Haskell

DRBD

LVM

Hypervisor

Architecture



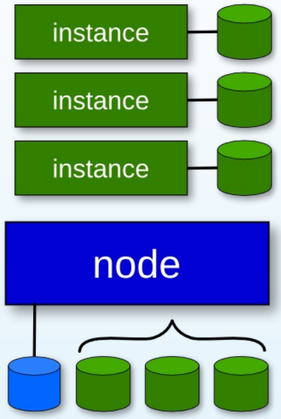
Nodes

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

Node Daemons

`ganeti-noded`	control hardware resources, runs on all
`ganeti-confd`	only functional on master, runs on all
`ganeti-rapi`	offers HTTP-based API for cluster, runs on master
`ganeti-masterd`	allows control of cluster, runs on master

Instances



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

Instance Parameters

- Hypervisor (called `hvparams`)
- General (called `beparams`)
- Networking (called `nicparams`)
- *Modified* via instance or cluster defaults

hvparams

- Boot order, CDROM Image
- NIC Type, Disk Type
- VNC Parameters, Serial console
- Kernel Path, initrd, args
- Other Hypervisor specific parameters

beparams
nicparams

- Memory / Virtual CPUs
- MAC
- NIC mode (routed or bridged)
- Link

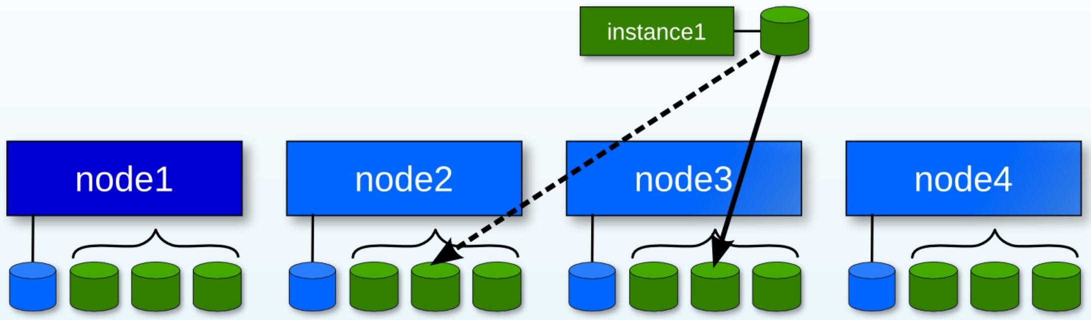
Disk template

- **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)

Planning your cluster

Hardware Planning

Disks

Types: SAS vs SATA

Speed: Faster = better

Number: More = better

Hardware Planning

CPU

Cores: More = better

Speed: Depends on your uses

Brand: AMD vs Intel

Hardware Planning

RAM

Amount: More = better

Use case: Types of services

Other considerations

RAID

Redundant Power

Higher **Density**

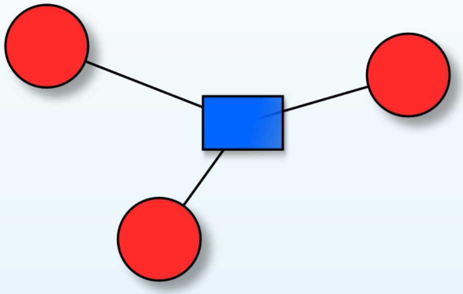
More **nodes**

Network topology

Operating System Planning

- **Debian** - most supported upstream
- **Gentoo** - great support
- **Ubuntu** - should work great
- CentOS - works but a few setup issues

Networking



Bridging is most widely used

Routed networking also supported

Nodes on *private NAT/VLAN*

Hands-on Setup

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*

Installing the Hypervisor

Hypervisor requirements

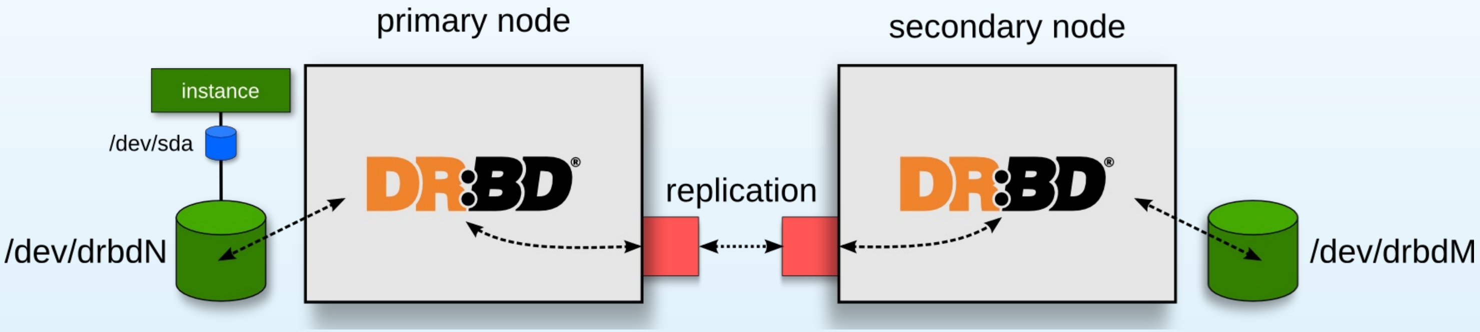
Mandatory on all nodes

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*

DRBD Setup

Installation

```
$ apt-get install drbd8-utils
```

Via modules

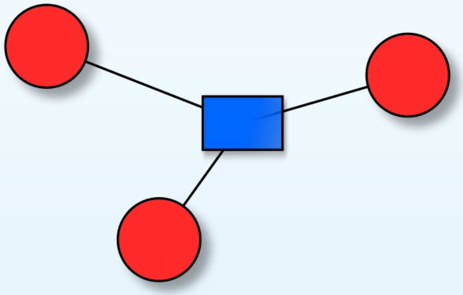
```
$ echo drbd minor_count=255 usermode_helper=/bin/true >> /etc/modules
$ depmod -a
$ modprobe drbd minor_count=255 usermode_helper=/bin/true
```

Via Grub

```
# Kernel Commands
drbd.minor_count=255 drbd.usermode_helper=/bin/true
```

Network Setup

Interface Layout



eth0 - trunked VLANs

eth1 - private DRBD network

VLAN setup

for Debian/Ubuntu

```
allow-hotplug eth0
allow-hotplug eth1
allow-hotplug vlan100
allow-hotplug vlan42

auto vlan100
iface vlan100 inet manual
    vlan_raw_device eth0

auto vlan42
iface vlan42 inet manual
    vlan_raw_device eth0
```

Bridge setup

for Debian/Ubuntu

```
allow-hotplug br42
allow-hotplug br10

auto br42
iface br42 inet static
   address 10.1.0.140
   netmask 255.255.254.0
   network 10.1.0.0
   broadcast 10.1.1.255
   gateway 10.1.0.1
   dns-nameservers 10.1.0.130
   dns-search example.org
   bridge_ports vlan42
   bridge_stp off
   bridge_fd 0

auto br100
iface br100 inet manual
   bridge_ports vlan100
   bridge_stp off
   bridge_fd 0
```

DRBD Network setup

for Debian/Ubuntu

```
iface eth1 inet static
    address 192.168.16.140
    netmask 255.255.255.0
    network 192.168.16.0
    broadcast 192.168.16.255
```

Configuring LVM

```
$ pvcreate /dev/sda3
$ vgcreate ganeti /dev/sda3
```

lvm.conf changes

Ignore drbd devices

```
filter = ["r|/dev/cdrom|", "r|/dev/drbd[0-9]+|" ]
```

Installing Ganeti

Installation Options

Via package manager

Via source

Installing Ganeti Dependencies

via source

```
$ apt-get install lvm2 ssh bridge-utils \
    iproute iputils-arping ndisc6 python \
    python-pyopenssl openssl \
    python-pyparsing python-simplejson \
    python-pyinotify python-pycurl socat
```

Htools Dependencies

provides IAllocator *hail*

```
$ apt-get install ghc6 libghc6-json-dev \
    libghc6-network-dev \
    libghc6-parallel-dev libghc6-curl-dev
```

Install Ganeti

Note: this is for >=ganeti-2.5

```
$ ./configure --localstatedir=/var \
    --sysconfdir=/etc \
    --enable-htools
$ make
$ make install
```

Startup Scripts

Installed into /usr/local/

```
$ cp doc/examples/ganeti.initd /etc/init.d/ganeti
$ update-rc.d ganeti defaults 20 80
```

ganeti-watcher

```
$ cp doc/examples/ganeti.cron /etc/cron.d/ganeti
```

- *Automatically* restarts failed instances
- Restarts *failed* secondary storage

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 Deboostrap* - 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

Install Instance Image Dependencies

```
$ apt-get install dump qemu-kvm kpartx
```

Install Instance Image

```
$ ./configure --prefix=/usr \
    --localstatedir=/var \
    --sysconfdir=/etc \
    --with-os-dir=/srv/ganeti/os
$ make
$ make install
```

Creating images

Manually install/setup guest

Shutdown guest

Create filesystem *dump* or *tarball*

Place in `IMAGE_DIR`

Hands on

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`

Initialization

KVM example

```
$ gnt-cluster init \
    --master-netdev=br0 \
    --vg-name ganeti \
    --secondary-ip 192.168.16.16 \
    --enabled-hypervisors=kvm \
    --nic-parameters link=br0 \
    --backend-parameters \
        vcpus=1,memory=128M \
    --hypervisor-parameters \
        kvm:kernel_path=/boot/vmlinuz-2.6-kvmU \
        vnc_bind_address=0.0.0.0 \
    ganeti.example.org
```

Cluster init args

Master Network Device

```
--master-netdev=br0
```

Volume Group Name

```
--vg-name ganeti
```

DRBD Interface

```
--secondary-ip 192.168.16.16
```

Enabled Hypervisors

```
--enabled-hypervisors=kvm
```

Cluster init args

Default NIC

```
--nic-parameters link=br0
```

Default Backend parameters

```
--backend-parameters vcpus=1,memory=128M
```

Default Hypervisor Parameters

```
--hypervisor-parameters \
    kvm:kernel_path=/boot/vmlinuz-2.6-kvmU, \
    vnc_bind_address=0.0.0.0 \
```

Cluster hostname

```
ganeti.example.org
```

Hands-on

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 burnin 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*)

Hands-on

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

among others

- 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
```

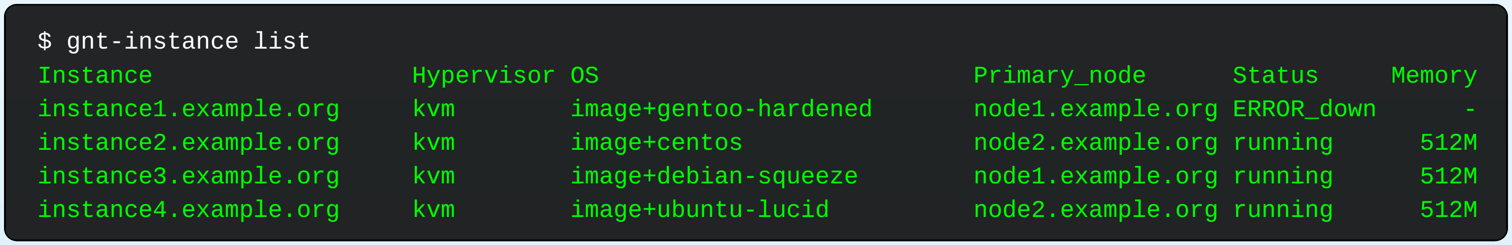
Started automatically

Do not use hypervisor directly

Querying Instances

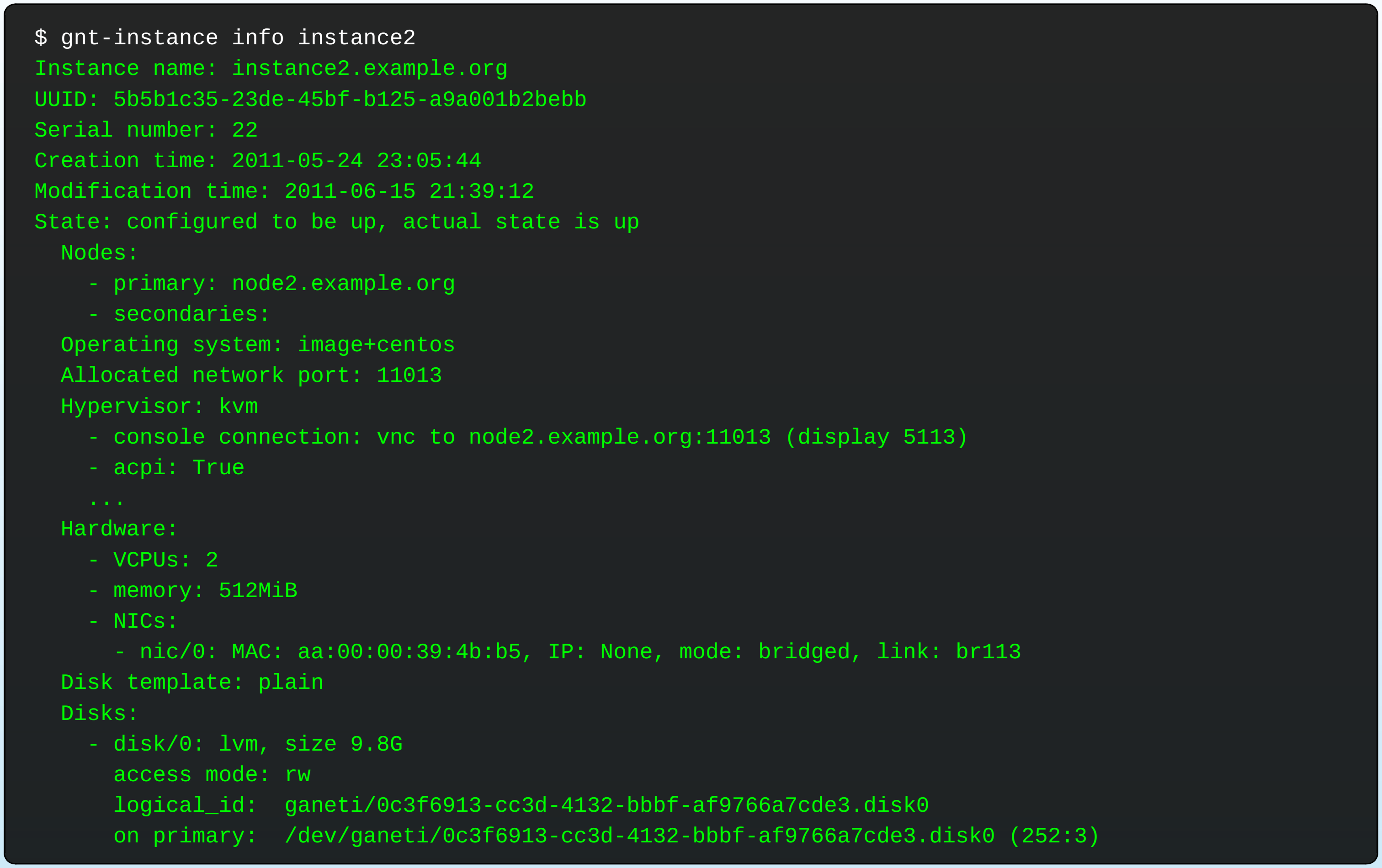
- **Two methods:**
  - listing instances
  - detailed instance information
- One useful for grep
- Other has more details, slower

Listing instances



```
$ gnt-instance list
Instance                  Hypervisor OS                          Primary_node      Status     Memory
instance1.example.org     kvm        image+gentoo-hardened       node1.example.org ERROR_down      -
instance2.example.org     kvm        image+centos                node2.example.org running      512M
instance3.example.org     kvm        image+debian-squeeze        node1.example.org running      512M
instance4.example.org     kvm        image+ubuntu-lucid          node2.example.org running      512M
```

Detailed Instance Info



```
$ gnt-instance info instance2
Instance name: instance2.example.org
UUID: 5b5b1c35-23de-45bf-b125-a9a001b2bebb
Serial number: 22
Creation time: 2011-05-24 23:05:44
Modification time: 2011-06-15 21:39:12
State: configured to be up, actual state is up
  Nodes:
    - primary: node2.example.org
    - secondaries:
  Operating system: image+centos
  Allocated network port: 11013
  Hypervisor: kvm
    - console connection: vnc to node2.example.org:11013 (display 5113)
    - acpi: True
    ...
  Hardware:
    - VCPUs: 2
    - memory: 512MiB
    - NICs:
      - nic/0: MAC: aa:00:00:39:4b:b5, IP: None, mode: bridged, link: br113
  Disk template: plain
  Disks:
    - disk/0: lvm, size 9.8G
      access mode: rw
      logical_id:  ganeti/0c3f6913-cc3d-4132-bbbf-af9766a7cde3.disk0
      on primary:  /dev/ganeti/0c3f6913-cc3d-4132-bbbf-af9766a7cde3.disk0 (252:3)
```

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

Instance Console

```
$ gnt-instance console INSTANCE_NAME
```

Type ^] when done, to exit.

Hands-on

Instance HA Features

Changing the Primary node

Failing over an instance

```
$ gnt-instance failover INSTANCE_NAME
```

Live migrating an instance

```
$ gnt-instance migrate INSTANCE_NAME
```

Restoring redundancy for DRBD-based instances

- *Primary* node storage failed
    - Re-create disks on it
- *Secondary* node storage failed
    - Re-create disks on secondary node
    - Change secondary

Replacing disks

```
$ # re-create disks on the primary node
gnt-instance replace-disks -p INSTANCE_NAME

$ # re-create disks on the current secondary
gnt-instance replace-disks -s INSTANCE_NAME

$ # change the secondary node, via manual
$ # specification
gnt-instance replace-disks -n NODE INSTANCE_NAME

$ # change the secondary node, via an iallocator
$ # script
gnt-instance replace-disks -I SCRIPT INSTANCE_NAME

$ # automatically fix the primary or secondary node
gnt-instance replace-disks -a INSTANCE_NAME
```

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 Operations

Add/Re-add

```
$ gnt-node add NEW_NODE
```

May need to pass `-s REPLICATION_IP` parameter

```
$ gnt-node add --readd EXISTING_NODE
```

`-s` parameter *not* required

Master fail-over

```
$ gnt-cluster master-failover
```

On a non-master, master-capable node

Evacuating nodes

- Moving the *primary* instances
- Moving *secondary* instances

Primary Instance conversion

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

Node Removal

```
$ gnt-node remove NODE_NAME
```

Deconfigure node

Stop ganeti daemons

Node in *clean* state

Hands-on

Job Operations

Listing Jobs

```
$ gnt-job list
17771 success INSTANCE_QUERY_DATA
17773 success CLUSTER_VERIFY_DISKS
17775 success CLUSTER_REPAIR_DISK_SIZES
17776 error   CLUSTER_RENAME(cluster.example.com)
17780 success CLUSTER_REDIST_CONF
17792 success INSTANCE_REBOOT(instance1.example.com)
```

Detailed Info

```
$ gnt-job info 17776
Job ID: 17776
  Status: error
  Received:         2009-10-25 23:18:02.180569
  Processing start: 2009-10-25 23:18:02.200335 (delta 0.019766s)
  Processing end:   2009-10-25 23:18:02.279743 (delta 0.079408s)
  Total processing time: 0.099174 seconds
  Opcodes:
    OP_CLUSTER_RENAME
      Status: error
      Processing start: 2009-10-25 23:18:02.200335
      Processing end:   2009-10-25 23:18:02.252282
      Input fields:
        name: cluster.example.com
      Result:
        OpPrereqError
        [Neither the name nor the IP address of the cluster has changed]
      Execution log:
```

Watching a job

```
$ gnt-instance add --submit … instance1
JobID: 17818
$ gnt-job watch 17818
Output from job 17818 follows
-----------------------------
Mon Oct 26 2009  - INFO: Selected nodes for instance instance1 via iallocator dumb: node1, node2
Mon Oct 26 2009 * creating instance disks...
Mon Oct 26 2009 adding instance instance1 to cluster config
Mon Oct 26 2009  - INFO: Waiting for instance instance1 to sync disks.
…
Mon Oct 26 2009 creating os for instance instance1 on node node1
Mon Oct 26 2009 * running the instance OS create scripts...
Mon Oct 26 2009 * starting instance...
```

30min break

Be back at 3:00pm

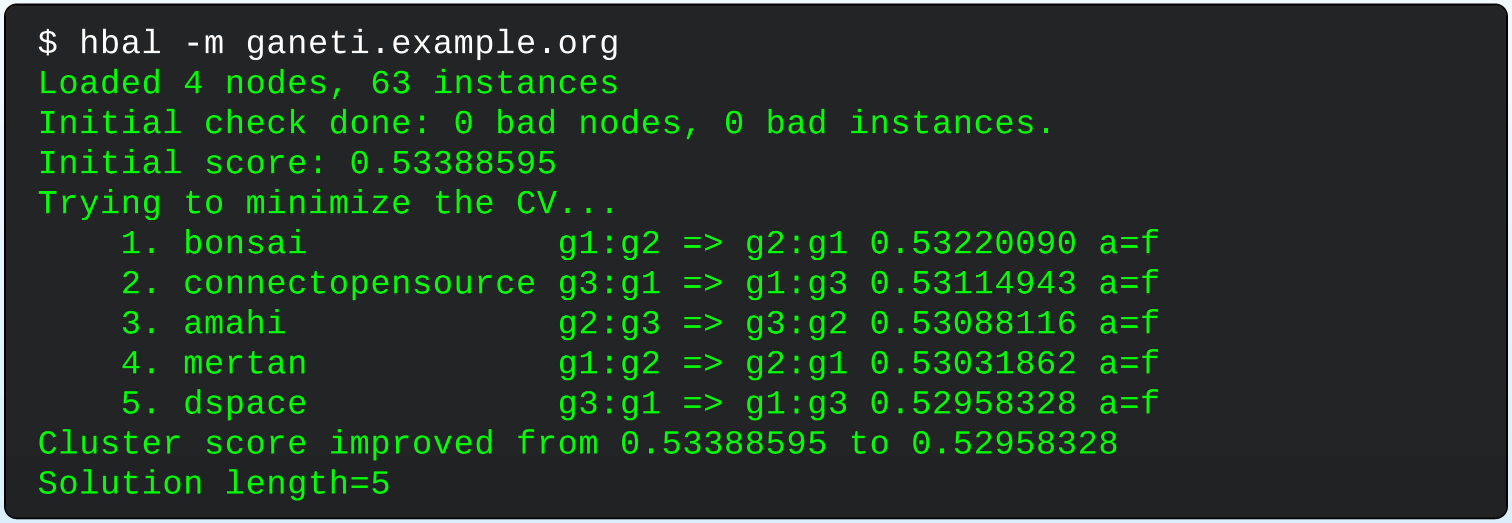
Hands-on

Using htools

Components

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

hbal



```
$ hbal -m ganeti.example.org
Loaded 4 nodes, 63 instances
Initial check done: 0 bad nodes, 0 bad instances.
Initial score: 0.53388595
Trying to minimize the CV...
    1. bonsai            g1:g2 => g2:g1 0.53220090 a=f
    2. connectopensource g3:g1 => g1:g3 0.53114943 a=f
    3. amahi             g2:g3 => g3:g2 0.53088116 a=f
    4. mertan            g1:g2 => g2:g1 0.53031862 a=f
    5. dspace            g3:g1 => g1:g3 0.52958328 a=f
Cluster score improved from 0.53388595 to 0.52958328
Solution length=5
```

Useful for cluster re-balancing

hbal

```
$ hbal -C -m ganeti.example.org
Loaded 4 nodes, 71 instances
Initial check done: 0 bad nodes, 0 bad instances.
Initial score: 2.10591985
Trying to minimize the CV...
    1. linuxfund            g4:g3 => g4:g2 2.09981699 a=r:g2
Cluster score improved from 2.10591985 to 2.09981699
Solution length=1

Commands to run to reach the above solution:

  echo jobset 1, 1 jobs
    echo job 1/1
    gnt-instance replace-disks -n g2 linuxfund
```

hspace

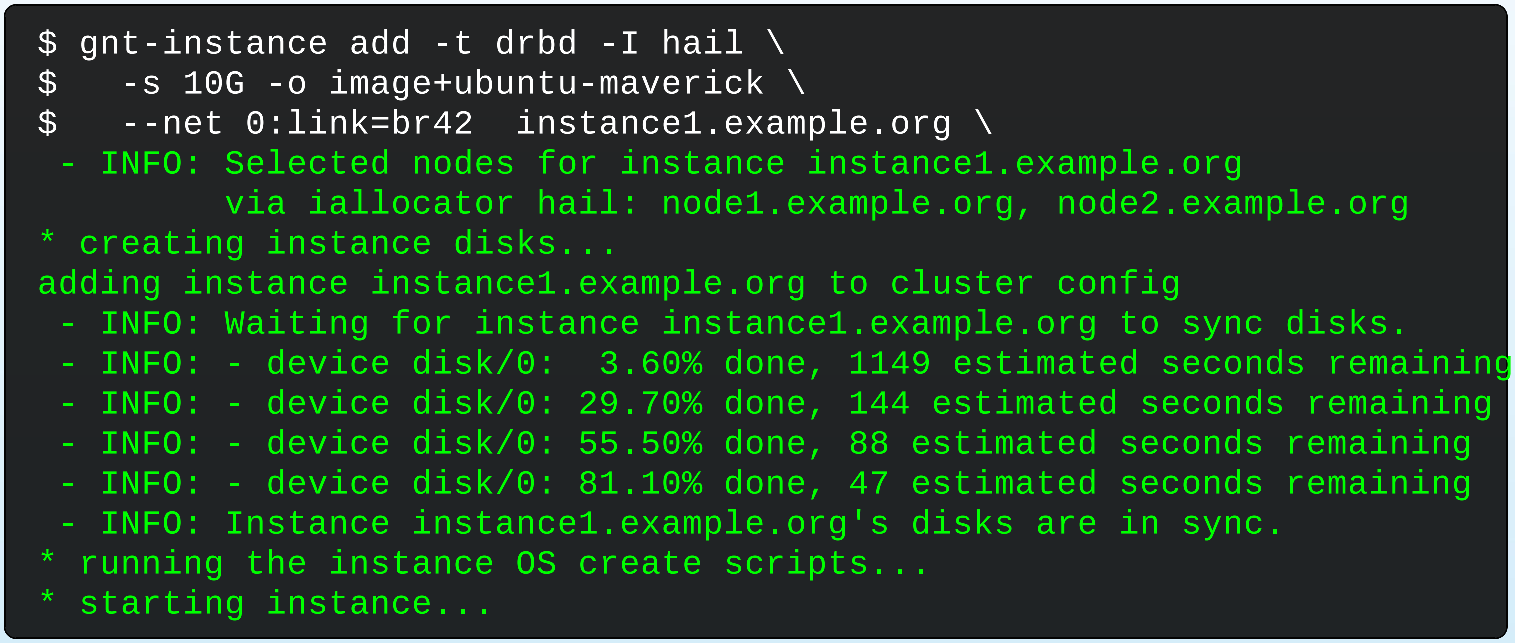
Cluster planning

```
$ hspace --memory 512 --disk 10240 \
$    -m ganeti.example.org
HTS_INI_INST_CNT=63

HTS_FIN_INST_CNT=101

HTS_ALLOC_INSTANCES=38
HTS_ALLOC_FAIL_REASON=FAILDISK
```

hail



```
$ gnt-instance add -t drbd -I hail \
$   -s 10G -o image+ubuntu-maverick \
$   --net 0:link=br42  instance1.example.org \
 - INFO: Selected nodes for instance instance1.example.org
         via iallocator hail: node1.example.org, node2.example.org
* creating instance disks...
adding instance instance1.example.org to cluster config
 - INFO: Waiting for instance instance1.example.org to sync disks.
 - INFO: - device disk/0:  3.60% done, 1149 estimated seconds remaining
 - INFO: - device disk/0: 29.70% done, 144 estimated seconds remaining
 - INFO: - device disk/0: 55.50% done, 88 estimated seconds remaining
 - INFO: - device disk/0: 81.10% done, 47 estimated seconds remaining
 - INFO: Instance instance1.example.org's disks are in sync.
* running the instance OS create scripts...
* starting instance...
```

Hands-on

Handling Node Failures

Node Groups

- All nodes in same *pool*
- Nodes not equally *connected* sometimes
- Cluster-wide *job locking*

Node Group Attributes

- At least *one* group
- `alloc_policy`: unallocable, last_resort, & preferred
- P/S nodes must be in the *same group* for an instance
- Group *moves* are possible

Node Group Management

```
# add a new node group
gnt-group add <group>

# delete an empty node group
gnt-group remove <group>

# list node groups
gnt-group list

# rename a node group
gnt-group rename <oldname> <newname>
```

Node Group Management

```
# list only nodes belonging to a node group
gnt-node {list,info} -g <group>

$ gnt-group list
Group   Nodes Instances AllocPolicy NDParams
default     5        74 preferred   (empty)

# assign a node to a node group
gnt-node modify -g <group>
```

OOB Management

- Emergency Power Off
- Repairs
- Crashes
- `gnt-cluster modify --oob-program <script>`

Remote API

Remote API

- *External* tools
- Retrieve cluster state
- *Execute* commands
- *JSON* over HTTP via *REST*

RAPI Security

- Users & Passwords
- RFC 2617 *HTTP Authentication*
- Read-only or Read-write

RAPI Example use-cases

- Web-based GUI (see *Ganeti Web Manager*)
- Automate cluster tasks via scripts
- Custom reporting tools

Project Roadmap

Project Details

- http://code.google.com/p/ganeti/
- License: *GPL v2*
- Ganeti 1.2.0 - December 2007
- Ganeti 2.0.0 - May 2009
- Ganeti 2.4.0 - Mar 2011 / *2.4.2* current
- Ganeti 2.5.0 - *July 2011?*

Upcoming features

- Merge htools
- CPU Pinning
- Replacing internal HTTP server
- Import/export version 2
- Moving instance across node groups
- Network management
- Shared storage support

Ganeti Web Manager

Conclusion

Questions?

Lance Albertson	Peter Krenesky
lance@osuosl.org	peter@osuosl.org
@ramereth	@kreneskyp
http://www.lancealbertson.com	http://blogs.osuosl.org/kreneskyp/

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

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

Presentation made with ***showoff***
http://github.com/ramereth/presentation-ganeti-tutorial
http://is.gd/osconganeti | http://is.gd/osconganetipdf