Google ganeti

Ganeti

Ganeti Core Team - Google
LISA '13 - 5 Nov 2013

Instances

Everything you wanted to know about instances

- Guido Trotter <ultrotter@google.com>
- Helga Velroyen <helgav@google.com>

Latest version of these slides

Please find the latest version of these slides at:

https://code.google.com/p/ganeti/wiki/LISA2013

The importance of instances

Running instances is the only purpose of Ganeti. As such Instances are the most important entity that we care about: everything else exists purely for their purpose.

What makes an instance

In order to define an instance we need the following:

- Virtual memory
- Virtual cpus
- Backend Storage
- Network interfaces
- Operating System
- Console access

Virtual memory and cpus

These are provided by the hypervisor (Xen or KVM). With Ganeti we just configure the memory size and the number of CPUs.

Depending on the hypervisor, also utilizing other advanced features like ballooning is possible.

Backend storage

Ganeti will manage the backend storage for instances. The most common backend storages are:

- plain (LVM logical volume device)
- drbd (DRBD device insisting on an LVM logical volume, with redundancy on a secondary node)

Ganeti also supports:

- file (image file on a node)
- sharedfile (image file on a shared mount point)
- RBD (radon block device)

We are also working on better Ceph support and on Gluster support.

Network interfaces

Each instance nic has:

- a mode: routed, bridged or openvswitch
- a link: the network entity on the host node it is connected to
- a MAC address, which is unique on the cluster (or more, if you set it so)
- an IP address (optional), which can be used by the network setup code or by hooks

New in 2.7: retrieve mode, link and ip from pre-configured networks.

Operating system

The Operating System is what Ganeti uses to initialize the instance. It

- implements the Ganeti OS API,
- is made up as a series of executables that format the instance disks and initialize them appropriately.

At minimum it can do nothing (for an instance that will later be installed from the network), and at most it sets up partitions, system, and configuration for an instance.

Currently OS scripts run as root on the target node, and so must be "trusted". Work is ongoing to allow untrusted OS images.

Main OS implementations

- `debootstrap`
  - reference OS implementation
  - glorified call of `debootstrap(8)`
  - `git://git.ganeti.org/instance-debootstrap.git`
- `ganeti-instance-image`
  - implementation that installs arbitrary images
  - images created with `tar(1)` or `dump(8)`
  - `https://code.osuosl.org/projects/ganeti-image`
- `snf-image`
  - implementation that provides an insulated environment for the installation
  - `http://www.synnefo.org/docs/synnefo/latest/snf-image.html`
- `ganeti-os-defs`
  - `http://sourceforge.net/p/ganeti-os-defs/home/Home/`

OS API

A definition of an OS that can be installed on an instance

- `create`, `import`, `export`, `rename`: instance operations
- `ganeti_api_version`: the OS conforms to (current one: `20`)
- `variants.list`: supported variants of the os
    - (eg. `+squeeze`, for `debootstrap`)
- `parameters.list`: extra parameters supported (name description)
- `verify`: check the OS parameters

Console Access

Instance console access is provided through the following means:

- Serial/Xen console (via `gnt-instance console <name>`)
- VNC (if set up instances listen on a node port for VNC access)
- Spice (KVM supports the spice remote desktop protocol)

Thank You!

Questions?

Survey at https://www.usenix.org/lisa13/training/survey