

Ganeti administrator's guide

Documents Ganeti version 1.2

1. Introduction

Ganeti is a virtualization cluster management software. You are expected to be a system administrator familiar with your Linux distribution and the Xen virtualization environment before using it.

The various components of Ganeti all have man pages and interactive help. This manual though will help you getting familiar with the system by explaining the most common operations, grouped by related use.

After a terminology glossary and a section on the prerequisites needed to use this manual, the rest of this document is divided in three main sections, which group different features of Ganeti:

- Instance Management
- High Availability Features
- Debugging Features

1.1. Ganeti terminology

This section provides a small introduction to Ganeti terminology, which might be useful to read the rest of the document.

Cluster

A set of machines (nodes) that cooperate to offer a coherent highly available virtualization service.

Node

A physical machine which is member of a cluster. Nodes are the basic cluster infrastructure, and are not fault tolerant.

Master node

The node which controls the Cluster, from which all Ganeti commands must be given.

Instance

A virtual machine which runs on a cluster. It can be a fault tolerant highly available entity.

Pool

A pool is a set of clusters sharing the same network.

Meta-Cluster

Anything that concerns more than one cluster.

1.2. Prerequisites

You need to have your Ganeti cluster installed and configured before you try any of the commands in this document. Please follow the *Ganeti installation tutorial* for instructions on how to do that.

2. Managing Instances

2.1. Adding/Removing an instance

Adding a new virtual instance to your Ganeti cluster is really easy. The command is:

```
gnt-instance add -n TARGET_NODE -o OS_TYPE -t DISK_TEMPLATE INSTANCE_NAME
```

The instance name must be resolvable (e.g. exist in DNS) and of course map to an address in the same subnet as the cluster itself. Options you can give to this command include:

- The disk size (`-s`)
- The swap size (`--swap-size`)
- The memory size (`-m`)

- The number of virtual CPUs (`-p`)
- The instance ip address (`-i`) (use the value `auto` to make Ganeti record the address from dns)
- The bridge to connect the instance to (`-b`), if you don't want to use the default one

There are four types of disk template you can choose from:

diskless

The instance has no disks. Only used for special purpose operating systems or for testing.

plain

The instance will use LVM devices as backend for its disks. No redundancy is provided.

local_raid1

A local mirror is set between LVM devices to back the instance. This provides some redundancy for the instance's data.

remote_raid1

Note: This is only valid for multi-node clusters using drbd 0.7.

A mirror is set between the local node and a remote one, which must be specified with the second value of the `--node` option. Use this option to obtain a highly available instance that can be failed over to a remote node should the primary one fail.

drbd

Note: This is only valid for multi-node clusters using drbd 8.0.

This is similar to the `remote_raid1` option, but uses new features in drbd 8 to simplify the device stack. From a user's point of view, this will improve the speed of the **replace-disks** command and (in future versions) provide more functionality.

For example if you want to create an highly available instance use the `remote_raid1` or `drbd` disk templates:

```
gnt-instance add -n TARGET_NODE[:SECONDARY_NODE] -o OS_TYPE -t remote_raid1 \  
    INSTANCE_NAME
```

To know which operating systems your cluster supports you can use

```
gnt-os list
```

Removing an instance is even easier than creating one. This operation is non-reversible and destroys all the contents of your instance. Use with care:

```
gnt-instance remove INSTANCE_NAME
```

2.2. Starting/Stopping an instance

Instances are automatically started at instance creation time. To manually start one which is currently stopped you can run:

```
gnt-instance startup INSTANCE_NAME
```

While the command to stop one is:

```
gnt-instance shutdown INSTANCE_NAME
```

The command to see all the instances configured and their status is:

```
gnt-instance list
```

Do not use the xen commands to stop instances. If you run for example `xm shutdown` or `xm destroy` on an instance Ganeti will automatically restart it (via the `ganeti-watcher(8)`)

2.3. Exporting/Importing an instance

You can create a snapshot of an instance disk and Ganeti configuration, which then you can backup, or import into another cluster. The way to export an instance is:

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

The target node can be any node in the cluster with enough space under `/srv/ganeti` to hold the instance image. Use the `--noshutdown` option to snapshot an instance without rebooting it. Any previous snapshot of the same instance existing cluster-wide under `/srv/ganeti` will be removed by this operation: if you want to keep them move them out of the Ganeti exports directory.

Importing an instance is similar to creating a new one. The command is:

```
gnt-backup import -n TARGET_NODE -t DISK_TEMPLATE --src-node=NODE --src-dir=DIR INSTANCE_NAME
```

Most of the options available for the command `gnt-instance add` are supported here too.

3. High availability features

Note: This section only applies to multi-node clusters.

3.1. Failing over an instance

If an instance is built in highly available mode you can at any time fail it over to its secondary node, even if the primary has somehow failed and it's not up anymore. Doing it is really easy, on the master node you can just run:

```
gnt-instance failover INSTANCE_NAME
```

That's it. After the command completes the secondary node is now the primary, and vice versa.

3.2. Replacing an instance disks

So what if instead the secondary node for an instance has failed, or you plan to remove a node from your cluster, and you failed over all its instances, but it's still secondary for some? The solution here is to replace the instance disks, changing the secondary node. This is done in two ways, depending on the disk template type. For `remote_raid1`:

```
gnt-instance replace-disks -n NEW_SECONDARY INSTANCE_NAME
```

and for `drbd`:

```
gnt-instance replace-disks -s -n NEW_SECONDARY INSTANCE_NAME
```

This process is a bit longer, but involves no instance downtime, and at the end of it the instance has changed its secondary node, to which it can if necessary be failed over.

3.3. Failing over the master node

This is all good as long as the Ganeti Master Node is up. Should it go down, or should you wish to decommission it, just run on any other node the command:

```
gnt-cluster masterfailover
```

and the node you ran it on is now the new master.

3.4. Adding/Removing nodes

And of course, now that you know how to move instances around, it's easy to free up a node, and then you can remove it from the cluster:

```
gnt-node remove NODE_NAME
```

and maybe add a new one:

```
gnt-node add [--secondary-ip=ADDRESS] NODE_NAME
```

4. Debugging Features

At some point you might need to do some debugging operations on your cluster or on your instances. This section will help you with the most used debugging functionalities.

4.1. Accessing an instance's disks

From an instance's primary node you have access to its disks. Never ever mount the underlying logical volume manually on a fault tolerant instance, or you risk breaking replication. The correct way to access them is to run the command:

```
gnt-instance activate-disks INSTANCE_NAME
```

And then access the device that gets created. After you've finished you can deactivate them with the deactivate-disks command, which works in the same way.

4.2. Accessing an instance's console

The command to access a running instance's console is:

```
gnt-instance console INSTANCE_NAME
```

Use the console normally and then type `^]` when done, to exit.

4.3. Instance OS definitions Debugging

Should you have any problems with operating systems support the command to ran to see a complete status for all your nodes is:

```
gnt-os diagnose
```

4.4. Cluster-wide debugging

The `gnt-cluster` command offers several options to run tests or execute cluster-wide operations. For example:

```
gnt-cluster command  
gnt-cluster copyfile  
gnt-cluster verify  
gnt-cluster getmaster  
gnt-cluster version
```

See the man page `gnt-cluster(8)` to know more about their usage.