

---

# **Ganeti Web Manager Documentation**

*Release 0.10.2*

**Oregon State University Open Source Lab**

September 09, 2013



# CONTENTS



Ganeti Web Manager is a [Django](#) based web frontend for managing [Ganeti](#) virtualization clusters. Since Ganeti only provides a command-line interface, Ganeti Web Manager's goal is to provide a user friendly interface to Ganeti by being feature complete with Ganeti's *RAPI*. On top of Ganeti it provides a permission system for managing access to clusters and virtual machines and an in browser VNC console.

If you don't already have a Ganeti cluster setup, these [directions](#) can help you get started. If you are looking for support, please contact us through these [channels](#). If you are looking to deploy Ganeti Web Manager for the first time, check out our *Installation* guide. If you already have a Ganeti Web Manager instance running it might be time to *upgrade*.

Ganeti Web Manager is licensed under the *GPLv2*. It is currently developed and maintained by the Oregon State University Open Source Lab and a handful of volunteers. If you would like to get involved in development see our *development* guide.



# GETTING STARTED

## 1.1 Installation

Currently we have use *Fabric*, a tool for streamlining administration tasks, to deploy Ganeti Web Manager. Before installing Ganeti Web Manager, make sure you have all the required *Dependencies* installed.

### 1.1.1 Installing

1. Download and unpack the [latest release](#), currently this is 0.10.2.
2. Change to the project directory.

```
cd ganeti_webmgr
```

3. Run Fabric to automatically create a python virtual environment and install required dependencies. This may take a few minutes.

```
# Deploy a production environment
fab deploy
```

Changed in version 0.10: *fab prod deploy* is now *fab deploy*. *fab dev deploy* is still the same.

---

**Note:** If you would like a more noisy output, adding *v*, as in *fab v deploy*, will provide more verbosity.

---

4. While in the project root, copy the default settings file **settings.py.dist** to **settings.py**:

```
cp settings.py.dist settings.py
```

### 1.1.2 Minimum Configuration

Getting Ganeti Web Manager up and running requires a minimum configuration of a database server. If you don't have a database server available, and are fine using SQLite, you can skip this step.

1. Edit **settings.py** and change the database backend to your preferred database along with filling any any relevant details relating to your database setup.

```
'default': {
    # Add 'postgresql_psycopg2', 'postgresql', 'mysql',
    # 'sqlite3' or 'oracle'.
    'ENGINE': 'django.db.backends.',
```

```
# Or path to database file if using sqlite3.
'NAME': 'ganeti.db',

# Not used with sqlite3.
'USER': '',

# Not used with sqlite3.
'PASSWORD': '',

# Set to empty string for localhost. Not used with sqlite3.
'HOST': '',

# Set to empty string for default. Not used with sqlite3.
'PORT': '',
}
```

### 1.1.3 Initializing

1. Activate the Python Virtualenv:

```
source venv/bin/activate
```

2. Initialize Database:

MySQL/SQLite:

```
# Create new tables and migrate all apps using southdb
./manage.py syncdb --migrate
```

Postgres:

---

**Note:** This assumes you're doing a fresh install of Ganeti Web Manager on a new Postgres database.

---

```
./manage.py syncdb --all
./manage.py migrate --fake
```

3. Build the search indexes

```
./manage.py rebuild_index
```

---

**Note:** Running `./manage.py update_index` on a regular basis ensures that the search indexes stay up-to-date when models change in Ganeti Web Manager.

---

### 1.1.4 Next Steps

Congratulations! Ganeti Web Manager is now installed and initialized. Next, you'll want to look into *Configuring* and *Deployment*, if you are going to be setting up a production instance. Otherwise, if you just want to play around with Ganeti Web Manager, or are *developing*, take a look at *Testing*.



## 1.2 Deployment

If you haven't already *configured* Ganeti Web Manager, now would be a good time to do so.

Now that you have a Ganeti Web Manager instance setup and configured, you will want to deploy it to somewhere that can be accessed by a web browser.

### 1.2.1 Testing

If you are just testing Ganeti Web Manager out, run:

```
$ python manage.py runserver
```

Then open a web browser, and navigate to *http://localhost:8000*.

```
firefox http://localhost:8000
```

### 1.2.2 Apache + mod\_wsgi

Follow the *django* guide to deploy with *apache*. Here is an example *mod\_wsgi* file:

```
import os
import sys

path = '/var/lib/django/ganeti_webmgr'

# activate virtualenv
activate_this = '%s/venv/bin/activate_this.py' % path
execfile(activate_this, dict(__file__=activate_this))

# add project to path
if path not in sys.path:
    sys.path.append(path)

# configure django environment
os.environ['DJANGO_SETTINGS_MODULE'] = 'settings'

import django.core.handlers.wsgi
application = django.core.handlers.wsgi.WSGIHandler()
```

#### Virtualenv

The virtual environment must be activated for use with *mod\_wsgi*. This is done by executing the **activate\_this** script generated when a *virtualenv* is created. The following code should be in the **django.wsgi** file *apache* is configured to use.

```
# activate virtual environment
activate_this = '%s/venv/bin/activate_this.py' % PATH_TO_GANETI_WEBMGR
execfile(activate_this, dict(__file__=activate_this))
```

### 1.2.3 Nginx

### 1.2.4 Gunicorn

### 1.2.5 uWSGI

## 1.3 Importing a Cluster

1. Log in as an admin user.
2. Navigate *Clusters* -> *Add Cluster*
3. Fill out properties and click *save*

When the Cluster is imported into Ganeti Web Manager it will automatically synchronize. Virtual Machine objects will be created to match what is found on the Ganeti Cluster. *Permission Tags* will also be parsed to automatically add permissions for virtual machines.

A cluster can be added with only its hostname and port, but a username and password for the cluster are required if you want to make changes to it. Clusters added without a valid username and password appear in *Cluster Read Only Mode* where you can only change aspects of the cluster that are local to Ganeti Web Manager's database.

If you're logged in as a cluster admin or superuser you can edit properties of a cluster by using the "edit" button on the cluster detail page.

**Warning:** If a cluster is in *Cluster Read Only Mode*, there will be errors if you try to modify virtual machines or create new ones.

### 1.3.1 Synchronizing Clusters

Ganeti Web Manager stores some information about clusters in its database. Cluster and virtual machine information will *refresh automatically*, but the list of virtual machines must be synchronized manually. This can be done by via the orphans view

1. Main Menu -> Orphans

Clusters are synchronized when the orphans view is visited.

### 1.3.2 Adding Virtual Machines

To add a virtual machine, select "Create VM" in the toolbar. Only fields with multiple options will be selectable. For example, if you are unable to change the cluster to which a VM gets added, it means that there is only one valid option and cluster is a mandatory field.

- If the user creating the VM has permissions to do so, the owner will be that user. If the user does not have create permissions but is a member of a group that can create VMs, ownership defaults to that group.
- Cluster can be chosen from those that the user creating the VM has access to.
- The Hypervisor will generally be dictated by the cluster that you choose.
- The instance name must be a fully qualified domain name (FQDN). (e.g. hostname.example.org)
- If you uncheck "Start up after creation", you can start the VM manually on its virtual machine detail page. (click Virtual Machines in the sidebar, then the VM's name)

- DNS name check: if checked, sends the name you selected for the VM to the resolver (e.g. in DNS or /etc/hosts, depending on your setup). Since the name check is used to compute the IP address this also enables/disables IP checks (e.g. if the IP is pingable). Uncheck if using dynamic DNS.
- Disk Template chooses a layout template from these options:
  - plain - Disk devices will be logical volumes (e.g. LVM)
  - drbd - Disk devices will be DRBD (version8.x) on top of LVM volumes
    - \* If drbd is selected, a primary and secondary node will need to be chosen unless automatic allocation has been selection. DRBD will allow the virtual machine to use live migration and failover in case one of the nodes goes offline.
  - file - Disk devices will be regular files (e.g. qcow2)
  - diskless - This creates a virtual machine with no disks. Its useful for testing only (or other special cases).
- Operating system to install on the virtual machine. Your choices are limited to the images configured on the cluster.

General Parameters:

- Virtual CPUs will be deducted from owner's quota. If the owner field appears blank and is not selectable, the default owner has been chosen.
- Memory is the amount of RAM to give this VM. If no units are given, megabytes is assumed.
- Disk size is the amount of owner's disk quota to allot this VM. If no units are given, megabytes is assumed.
- Disk type determines the way the disks are presented to the virtual machine. Options may vary based on cluster's hypervisor settings.
- More information about NIC Mode, NIC Link, and NIC Type can be found [here](#)

Hypervisor parameters:

*TODO finish this part*

### 1.3.3 Orphaned Virtual Machines

*Permission Tags* are parsed by virtual machine objects, but sometimes virtual machines will have no tags. To quickly identify virtual machines with no admin users, use the orphans view

1. Main Menu -> Orphans

Visiting the orphans view will force a synchronization of all clusters and display VirtualMachines that do not have any permissions assigned. You only need to grant permissions directly on virtual machines if you are granting access to non-admin users.



# FEATURES

## 2.1 Permissions

Permissions may be granted to both clusters and virtual machines. The permissions system is intended to allow users to manage themselves. Any object that can have its permissions edited will have a *Users* tab.

For a high level description of how permissions can be used in various scenarios, read this [blog post](#).

### 2.1.1 Adding users to objects.

1. navigate to Group, Cluster, or VirtualMachine detail page
2. click *Add New User*
3. select user or group
4. select permissions
5. *save*

### 2.1.2 Updating permissions

1. navigate to Group, Cluster, or VirtualMachine detail page
2. click *Users* tab
3. click permissions column
4. select permissions and *save*

### 2.1.3 Deleting permissions

1. navigate to Group, Cluster, or VirtualMachine detail page
2. click *Users* tab
3. click the *delete* icon

Deleting a user will remove all permissions, and other properties associated with the user such as cluster quotas.

## Groups

Groups may be created so that permissions. This allows permissions structures where you are granting permissions to different organizations. Users may belong to unlimited number of groups. They will inherit the permissions of any group they belong to.

Groups are a persona that user's may act on behalf of. When creating virtual machines, the user must choose whether they are acting on behalf of themselves or a group they are a member of. When acting on behalf of a group, the group's permissions and quota used.

### 2.1.4 Group Permissions

- **admin** - Grants the ability to see the member list, and edit permissions

## Cluster

These permissions can be granted to either a user or a group. A user who is part of a group with a permission does not automatically have that permission individually. For instance, a user who is part of a group that has VM create permission can create a VM, but can only assign ownership to the group, not to himself. To grant permissions on a cluster, click *add user* on the Users tab of the cluster detail page. Cluster permissions can also be added by clicking *Add Cluster* in the Permissions tab of the user detail page.

- **admin** - Grants full access to the cluster. Includes ability to set permissions and quotas, and full access to all virtual machines.
- **create\_vm** - Grants ability to create virtual machines on the cluster.
- **tags** - Grants ability to set tags on the cluster.
- **replace disks** - Ability to replace disks of VMs on the cluster.
- **migrate** - Can migrate a VM to another node
- **export** - Can export a virtual machine

## Quotas

Quotas restrict the usage of cluster resources by users and groups. Default quotas can be set by editing clusters, if no quota is set unlimited access is allowed. This will affect all users and groups.

The default quota can be overridden on the cluster users page:

1. *Clusters -> Cluster -> Users*
2. click quota value.
3. edit values, and click *save*

Leaving a value empty specifies unlimited access for that resource.

## Virtual Machines

To grant a user permissions on a VM, click *Add VirtualMachine* in the Permissions tab of the User detail page. To grant permissions to a user or group, use the *Add User* button on the Users tab of the VM detail page.

- **admin** - Grants full access to the virtual machine, including granting permissions.
- **Modify** - Allows user to modify VM's settings, including reinstallation of OS

- **Remove** - Permission to delete this VM
- **Power** - Permission to start, stop, reboot, and access console
- **Tags** - Can set tags for this VM

### 2.1.5 Permission Tags

Permissions for virtual machines are also registered as tags on the virtual machine object. This allows the permissions to be viewed and set via the command line tool. Tags will be parsed when creating virtual machines, and will be updated when the object is refreshed (#387). When permissions are granted tags will be set on the virtual machine (#393).

Tags use the pattern: `GANETI_WEB_MANAGER:<permission>:[G|U]:<user_id>`

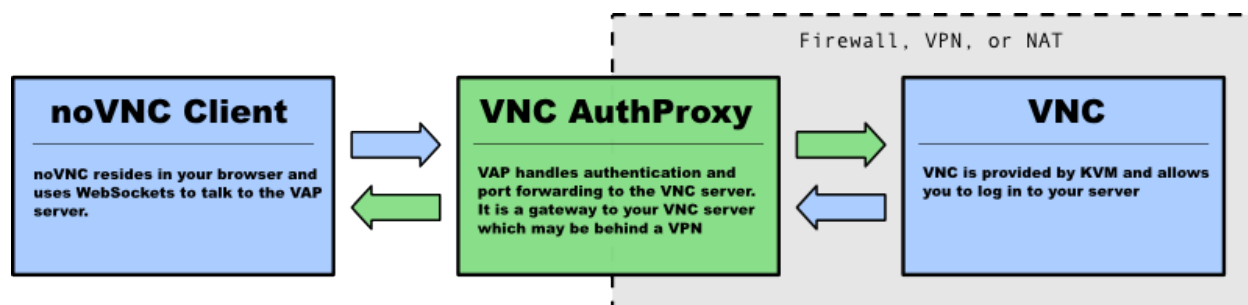
- **GANETI\_WEB\_MANAGER:admin:U:2** - admin permission for User with id 2
- **GANETI\_WEB\_MANAGER:admin:G:4** - admin permission for Group with id 4
- **GANETI\_WEB\_MANAGER:start:U:2** - start permission for User with id 2

## 2.2 Object Log

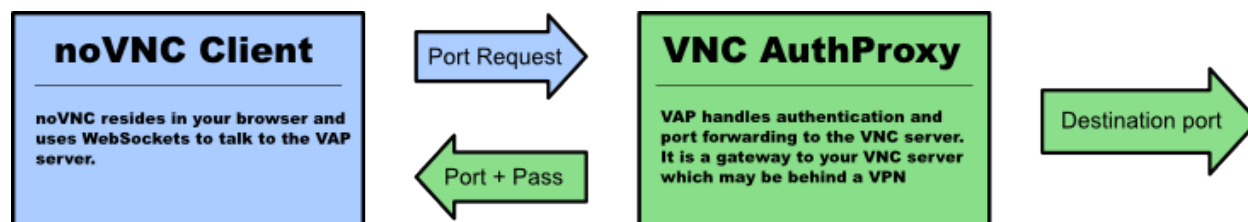
## 2.3 VNC

Ganeti Web Manager provides an in browser console using noVNC, an HTML5 client. noVNC requires WebSockets to function. Support for older browsers is provided through a flash applet that is used transparently in the absence of WebSockets.

### 2.3.1 VNC AuthProxy



VNC AuthProxy is required for the console tab to function. VNC servers do not speak websockets and our proxy allows your ganeti cluster to sit behind a firewall, VPN, or NAT.



VNCAuthProxy has a control channel that is used to request port forwarding to a specific VNC machine. It will respond with a local port and temporary password that must be used within a short period. This allows a secure connection with the VNCAuthProxy, without compromising the vnc password, and without leaving the port open to anyone with a port scanner.

## 2.3.2 Configuring VNC AuthProxy

Set the host and port that the proxy uses in `settings.py` with the `VNC_PROXY` setting.

Syntax is `HOST:CONTROL_PORT`, for example: “localhost:8888”.

If the host is localhost then the proxy will only be accessible to clients and browsers on the same machine as the proxy. Production servers should use a public hostname or IP.

```
# located in settings.py
VNC_PROXY='localhost:8888'
```

### Starting the Daemon

Twisted VNC Authproxy is started with `twistd`, the twisted daemon. Eventually we will include `init.d` scripts for better managing the daemon.

```
twistd --pidfile=/tmp/proxy.pid -n vncap
```

### Starting Flash Policy Server

Browsers that do not support WebSockets natively are supported through the use of a flash applet. Flash applets that make use of sockets must retrieve a policy file from the server they are connecting to. Twisted VNCAuthProxy includes a policy server. It must be run separately since it requires a root port. You may want to open port 843 in your firewall for production systems.

Start the policy server with `twistd`

```
sudo twistd --pidfile=/tmp/policy.pid -n flashpolicy
```

### Firewall Rules

The following ports are used by default

- **8888**: Control port used to request vnc forwarding. Should be open between **Ganeti Web Manager** and **Proxy**
- **12000+**: Internal VNC Ports assigned by **Ganeti**. Should be open between **Proxy** and **Ganeti Nodes**.
- **7000-8000**: External VNC Ports assigned by **Proxy**. Should be open between **Proxy** and **Clients/Web Browsers**.
- **843**: Flash policy server. Required to support browsers without native websocket support. Should be open between **Proxy** and **Clients/Web Browsers**.

## 2.3.3 Debugging Help

### Python Path for flash policy server

The following error indicates that your python path is not set or the proxy is not installed.



```
/usr/bin/twistd: Unknown command: flashpolicy
```

Ensure that your virtualenv is active

```
source venv/bin/activate
```

If not using a virtualenv, then you must manually set the **PYTHONPATH** environment variable as root.

```
export set PYTHONPATH=.
```

## 2.4 SSH Keys

Ganeti Web Manager allows users to store SSH Keys. Each virtual machine has a view that will return SSH keys for users with access.

### 2.4.1 Configuring User SSH Keys

#### As an User

1. click your **username** in the menu sidebar
2. use the Add, Edit, and Delete buttons to manage your keys

#### As an Admin

1. click **Users** in the menu sidebar
2. click the edit button for the user you want to edit
3. use the Add, Edit, and Delete buttons to manage your keys

### 2.4.2 SSH Keys script

Ganeti Web Manager provides a script that will automatically generate an `authorized_keys` files

```
$ python util/sshkeys.py [-c CLUSTER [-i INSTANCE]] API_KEY URL
```

- **API\_KEY** is the value set in `settings.py`
- **URL** is a URL pointing to the GWM server
- **CLUSTER** is the identifier of a cluster
- **INSTANCE** is the hostname of an instance

The GWM server URL has some flexibility in how it may be specified; HTTP and HTTPS are supported, as well as custom port numbers. The following are all valid URLs:

- `http://example.com/`
- `https://example.com/`
- `http://example.com:8080/`

**CLUSTER** and **INSTANCE** are optional. Including them will narrow the list of users to either a **Cluster** or a **VirtualMachine**.

### 2.4.3 SSH Keys Ganeti hook

If you want your VMs to automatically copy the ssh keys from GWM, then you can use the included ssh keys ganeti hook found in `util/hooks/sshkeys.sh`. Copy that file onto every node in your cluster in the hooks directory for the instance definition you're using (i.e. `ganeti-debootstrap`). Copy and set the variables in `util/hooks/sshkeys.conf` into the variant config and/or the instance definition config file. Make sure that the hook is executable and all the variables are set include changing the API Key.

## 2.5 LDAP

New in version 0.10. Ganeti Web Manager supports LDAP authentication through the use of `django-auth-ldap` and `python-ldap`. A fabric command has been written to easily handle enabling and disabling LDAP support.

### 2.5.1 Dependencies

In order to use `python-ldap` a couple of system level packages need to be installed first.

**For a Debian based systems:**

- `libldap2-dev`
- `libsasl2-dev`

**For a Red Hat based systems:**

- `openldap-devel`

### 2.5.2 Deploying

To deploy Ganeti Web Manager with LDAP

1. Copy `ldap_settings.py.dist` to `ldap_settings.py`.

```
$ cp ldap_settings.py.dist ldap_settings.py
```

2. Change `ldap_settings.py` to fit your LDAP configuration.

```
$ vi ldap_settings.py
```

---

**Note:** `ldap_settings.py.dist` has been thoroughly commented so that external documentation shouldn't be needed. If you have specific questions about options or want an overview of the package, please consult the `django-auth-ldap` documentation.

---

3. Run the fabric command to enable LDAP in settings:

```
$ fab ldap
```

`fab ldap` installs `django-auth-ldap` and `python-ldap` and takes care of the commenting and uncommenting the lines in `settings.py` that handle LDAP imports.

### 2.5.3 Disabling

If you would like to later disable LDAP support, all that is required is to run:

```
$ fab ldap:disable
```

---

**Note:** This will remove `django-auth-ldap` and `python-ldap` but will not remove the system specific dependencies.

---

## 2.6 Virtual Machine Templates

A new feature of Ganeti Web Manager 0.8 is the ability to create Templates.

## 2.7 Managing Clusters

### 2.7.1 Ganeti RAPI users and passwords

Before you can start using Ganeti Web Manager you will need to create a user and password on the Ganeti cluster.

#### Create MD5 hash

Here is an example with a user **jack** and password **abc123**

```
echo -n 'jack:Ganeti Remote API:abc123' | openssl md5
```

#### Add user to Ganeti cluster

Add the hash to `/var/lib/ganeti/rapi_users` on all the nodes in the cluster and restart `ganeti-rapi`. Here's an example using above: For ganeti 2.4 and above you need use file `/var/lib/ganeti/rapi/users`

```
# Hashed password for jack
jack {HA1}54c12257ee9be413f2f3182435514aae write
```

For more information on adding users, please check the [Ganeti RAPI documentation](#)

## 2.8 Cluster Read Only Mode

It is possible to add a cluster with only its hostname and port number, and no username and password credentials. This creates a copy of the cluster and its VMs in your local Ganeti Web Manager database without giving you the ability to change the cluster itself.

### 2.8.1 In Read-Only mode, you CAN:

- Assign ownership of VMs to GWM users from the Orphans page
- Delete VMs from your Ganeti Webmanager database from the Missing VMs page
- Import nodes to your database or delete nodes from it

- Assign permissions to users on the cluster or VM (note that although you can assign VM create permission to a user or group, they cannot actually create a VM in read-only mode)
- Edit the cluster, so that you can go back and add username/password credentials and gain full privileges on it later.
- Delete the record of the cluster from your database (Note: This does not affect the actual cluster)
- Record a default quotas for Virtual CPUs, Disk Space, and Memory
- Change the cluster’s slug (the name of the cluster as it appears in the url: <hostname>/cluster/<slug>/<vm>)

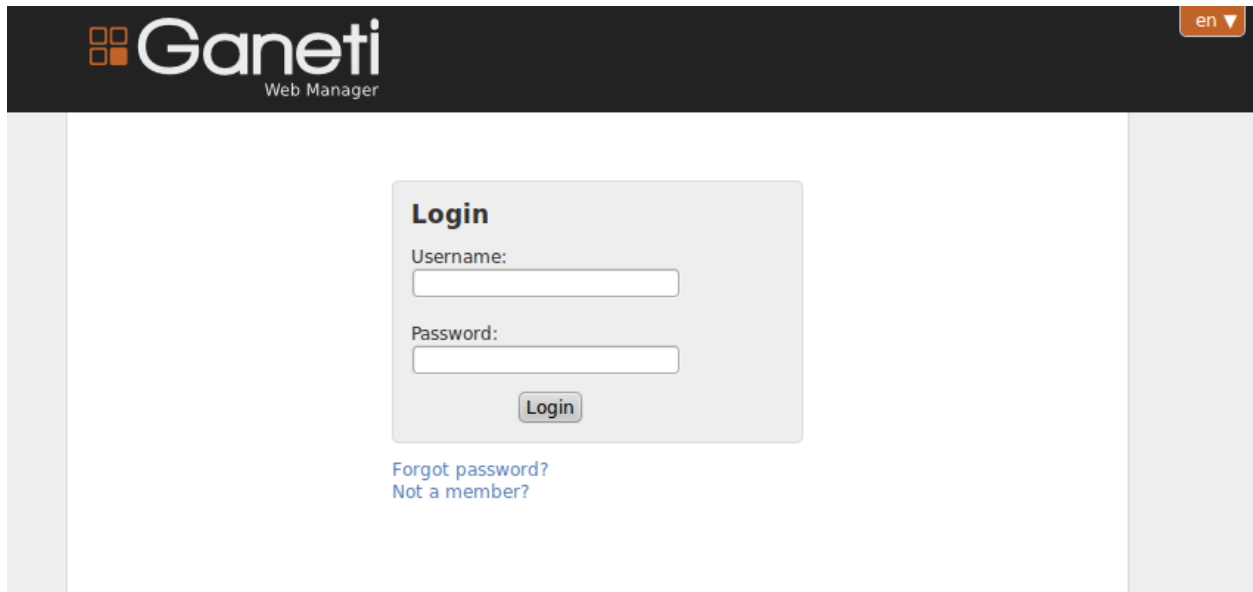
## 2.8.2 In Read-Only mode, you can NOT:

- Redistribute the cluster’s configuration
- Start, stop, or reinstall a VM
- Migrate or change disks
- Access a VM’s console
- Create a new VM on the cluster

## 2.9 Open Registration

Ganeti Web Manager versions 0.8 and above allow you to choose whether users can create their own accounts, or need to be added by an administrator.

The default setting for registration is open, which means that visitors to your site’s login page can follow a link from the login page to create their own accounts.



The “Not a member?” link takes the user to the registration page:

The user is emailed a password and a confirmation link, then has an account on your site. Users can also be added by a site admin by selecting the users link in the admin toolbar, then using the Add User button to reach the user creation form.

### 2.9.1 Closing Registration

In some contexts, users should not be able to create their own accounts. To implement this, simply change the `ALLOW_OPEN_REGISTRATION` setting in your `settings.py` file to `False`:

```
# Whether users should be able to create their own accounts.
# False if accounts can only be created by admins.
ALLOW_OPEN_REGISTRATION = False
```

This setting is currently located near line 203, below the registration email settings.

#### Result of closed registration

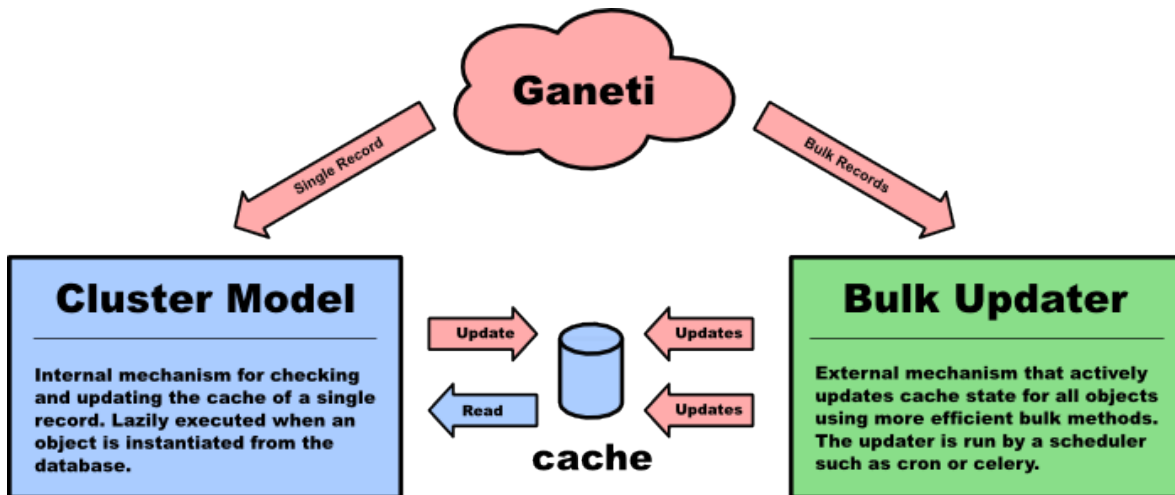
The “Not a member?” link is hidden from users on the login page. If they navigate to the `<SITE_ROOT>/accounts/register` page, they will see this message instead of the account creation form:

## 2.10 Nope! This page is defunct (Cache System)

Ganeti Web Manager uses a cache system that stores information about ganeti clusters in the database. This allows the following:

- Permissions are stored in the database and are associated to the cached objects
- The cached data can be searched and or filtered
- Limits the amount of traffic between the webserver and ganeti cluster.

The cache system is transparent and will load cached data automatically when the object is initialized.



### 2.10.1 Lazy Cache Refresh

Cached objects will refresh themselves transparently when they are out of date. This happens transparently when objects are queried from the ORM. Lazy cache refreshing is inefficient, it will cause multiple calls to the ganeti RAPI to fetch information. For this reason the lazy refresh mechanism is intended to only be used for testing, and as a backup to ensure that objects will always be refreshed.

### 2.10.2 CachedClusterObject

The functionality for lazy caching is built into an abstract model, `CachedClusterObject`. Extending this model will enable caching for the object. It requires that `_refresh()` be implemented with an object specific method for querying fresh info from ganeti. Currently only `Cluster` and `VirtualMachine` are cached, but this may extend to `Node` and `Job` objects in the future.

`parse_persistent_info()` can be overridden to parse object specific properties that should be stored in the database. This allows properties to be used as query filters, without requiring the entire object to be loaded.

### Bypassing The Cache Refresh

It is not currently possible to bypass the automatic cache refresh in a simple way since it is part of the models `*init*`. Currently the only way to bypass the cache is to query the object with a `values` or `values_list` query, and copy the values into a new object.

```
values = VirtualMachine.objects.get(id=id)
vm = VirtualMachine()
for k, v in values.items():
    setattr(vm, k, v)
```

### 2.10.3 RAPI Client Cache

Ganeti remote API clients are also cached. This reduces the number of database calls to retrieve a client capable of connecting to a cluster. This is a deterministic cache based off connection credentials. The keys are a hash of hostname, port, user, and password. This allows changes in settings to be easily detected. Cached objects should store the hash as part of its model and use it to look up existing clients without querying the cluster for the full set of connection credentials.

## 2.11 Caching

Ganeti Web Manager caches objects for performance reasons.

### 2.11.1 Why are things cached?

Ganeti is a bottleneck when accessing data. In tests, over 97% of time taken to render a normal page in Ganeti Web Manager is spent waiting for Ganeti to respond to queries. Thus, Ganeti Web Manager caches some of Ganeti's data.

### 2.11.2 Manual Updates

Sometimes it is necessary to refresh objects manually. To do this, navigate to the detail page for the cluster of the object that needs to be refreshed, and click the "Refresh" button. This will refresh the cluster and all of its objects.

### 2.11.3 Cached Cluster Objects

Some database-bound objects cache Ganeti data automatically. The functionality for this caching is encapsulated in the `CachedClusterObject` class. Any models which inherit from this class will gain this functionality.

#### Bypassing the Cache

The cache cannot currently be bypassed reasonably. `CachedClusterObject` uses `__init__()` to do part of its work. An unreasonable, albeit working, technique is to abuse the ORM:

```
values = VirtualMachine.objects.get(id=id)
vm = VirtualMachine()
for k, v in values.items():
    setattr(vm, k, v)
```

### 2.11.4 RAPI Cache

RAPI clients are cached in memory, and a hash of cluster information is stored in order to locate them quickly. The entire scheme is no longer necessary since RAPI clients are no longer expensive to allocate, and will be removed soon.





# USAGE

## 3.1 Configuring

Deploying a production server requires additional setup steps.

### 3.1.1 Required

1. Change your **SECRET\_KEY** and **WEB\_MGR\_API\_KEY** to unique (and hopefully unguessable) strings in your settings.py.
2. Change the ownership of the `whoosh_index` directory to apache

```
chown apache:apache whoosh_index/
```

3. Ensure the server has the ability to send emails or you have access to an SMTP server. Set **“EMAIL\_HOST”**, **“EMAIL\_PORT”**, and **“DEFAULT\_FROM\_EMAIL”** in settings.py. For more complicated outgoing mail setups, please refer to the [django email documentation](#).
4. Configure the [Django Cache Framework](#) to use a production capable backend in **settings.py**. By default Ganeti Web Manager is configured to use the **LocMemCache** but it is not recommended for production. Use Memcached or a similar backend.

```
CACHES = {
    'default': {
        'BACKEND': 'django.core.cache.backends.locmem.LocMemCache',
    }
}
```

5. For versions  $\geq 0.5$  you may need to add the full filesystem path to your templates directory to **“TEMPLATE\_DIRS”** and remove the relative reference to **“templates”**. We’ve had issues using wsgi not working correctly unless this change has been made.

### 3.1.2 Optional

### 3.1.3 VNC

1. Set **VNC\_PROXY** to the hostname of your VNC AuthProxy server in **settings.py**. The VNC AuthProxy does not need to run on the same server as Ganeti Web Manager.

```
VNC_PROXY = 'my.server.org:8888'
```

### 3.1.4 SSH Keys

## 3.2 Upgrading

---

**Note:** Please read the instructions fully before starting. The order of operations is important. The upgrade may fail if done out of order.

---

This guide will walk you through upgrading Ganeti Web Manager. Our upgrade process uses [South](#), a database migration tool that will update your database.

1. Backup the database
2. Download the latest code
3. Save a copy of **settings.py**
4. Deploy code to your existing directory
5. Copy **settings.py** back into the directory

Follow the guide for your version.

### 3.2.1 Upgrading From Version 0.4

If you are upgrading from version 0.4 you will be required to convert your installation to use South. Version 0.4 did not track the database with South, so South must be informed that your installation is already partially migrated. Read the [South documentation](#) for more information about converting apps.

1. Backup your database
2. [install python-django-south](#).
3. Add “south” to the list of **INSTALLED\_APPS** inside **settings.py**
4. Make sure you add any new settings to **settings.py** that are listed in Settings Changes
5. Synchronize the database with **./manage.py syncdb**

```
$ ./manage.py syncdb
```

```
/usr/lib/pymodules/python2.6/registration/models.py:4: DeprecationWarning: the sha module is deprecated
  import sha
Syncing...
Creating table south_migrationhistory
No fixtures found.
```

```
Synced:
```

```
> django.contrib.auth
> django.contrib.admin
> django.contrib.contenttypes
> django.contrib.sessions
> django.contrib.sites
> registration
> logs
> object_permissions
> south
```

```
Not synced (use migrations):
```

```
- ganeti
- logs
(use ./manage.py migrate to migrate these)
```

#### 6. Convert the ganeti app to use South for future migrations.

```
$ ./manage.py migrate ganeti 0001 --fake

/usr/lib/pymodules/python2.6/registration/models.py:4: DeprecationWarning: the sha module is deprecated
import sha
- Soft matched migration 0001 to 0001_version_0_4.
Running migrations for ganeti:
- Migrating forwards to 0001_version_0_4.
> ganeti:0001_version_0_4
(faked)
```

#### 7. Convert the logs app to use South for future migrations.

```
$ ./manage.py migrate logs 0001 --fake

/usr/lib/pymodules/python2.6/registration/models.py:4: DeprecationWarning: the sha module is deprecated
import sha
- Soft matched migration 0001 to 0001_version_0_4.
Running migrations for logs:
- Migrating forwards to 0001_version_0_4.
> logs:0001_version_0_4
(faked)
```

#### 8. Run South migration

```
$ ./manage.py migrate

/usr/lib/pymodules/python2.6/registration/models.py:4: DeprecationWarning: the sha module is deprecated
import sha
Running migrations for ganeti:
- Migrating forwards to 0002_version_0_5.
> ganeti:0002_version_0_5
- Loading initial data for ganeti.
No fixtures found.
Running migrations for logs:
- Nothing to migrate.
```

### 3.2.2 Upgrading from >=0.5

1. **Backup** your database

#### Pre-0.8

1. Run South migration.

```
$ ./manage.py migrate
```

#### 0.8 and later

1. Delete ghost migrations while running migrations.

```
$ ./manage.py migrate --delete-ghost-migrations
```

2. Update `settings.py` following the guide below

### 3.2.3 Settings Changes

The following settings have been added or changed. Please modify `settings.py` with these new values.

#### Version 0.5

##### TESTING

```
# XXX - Django sets DEBUG to False when running unittests. They want to ensure
# that you test as if it were a production environment. Unfortunately we have
# some models and other settings used only for testing. We use the TESTING flag
# to enable or disable these items.
#
# If you run the unittests without this set to TRUE, you will get many errors!
TESTING = False
```

##### ITEMS\_PER\_PAGE

```
# default items per page
ITEMS_PER_PAGE = 20
```

##### VNC\_PROXY

```
# Enable the VNC proxy. When enabled this will use the proxy to create local
# ports that are forwarded to the virtual machines. It allows you to control
# access to the VNC servers. When disabled, the console tab will connect
# directly to the VNC server running on the virtual machine.
#
# Expected values: False if no proxy, string with proxy host and port otherwise
# String syntax: "HOST:PORT", for example: "localhost:8888"
#
# Note: you will probably have to open more ports in firewall. For proxy's default
# settings, it uses port 8888 for listening for requests and ports 7000..8000
# for serving proxy.
#
# To run proxy (in 'util' directory):
# $ python vncauthproxy.py --websockets
# If you want to use encryption, then:
# $ python vncauthproxy.py --websockets --cert=FILE.pem
VNC_PROXY=False
```

##### Messages Framework

- Add `django.contrib.messages.middleware.MessageMiddleware` to `MIDDLEWARE_CLASSES`
- Add `django.contrib.messages` to `INSTALLED_APPS` after `django.contrib.contenttypes`

## Version 0.6

### Rename Logs App

The `logs` app has been renamed `object_log`. Update `INSTALLED_APPS` to reflect this change.

## Version 0.7

### South

```
# Disable South during unittests. This is optional, but will likely cause unittests
# to fail if these are not set properly.
SOUTH_TESTS_MIGRATE = False
SKIP_SOUTH_TESTS = True
```

### Haystack

```
# haystack search engine config
HAYSTACK_SITECONF = 'search_sites'
HAYSTACK_SEARCH_ENGINE = 'whoosh'
HAYSTACK_WHOOSH_PATH = os.path.join(DOC_ROOT, 'whoosh_index')
```

## Version 0.8

**Remember that it is absolutely critical to back up your database before making any changes.**

### User Registration

```
# Whether users should be able to create their own accounts.
# False if accounts can only be created by admins.
ALLOW_OPEN_REGISTRATION = True
```

More documentation for registration can be found at *Open Registration*.

## 3.2.4 Ganeti Version

### Ganeti Web Manager version 0.8

Users have experienced problems with Ganeti version 2.1, because it does not support some of the new RAPI features available in version 0.8 of Ganeti Web Manager. (see Issue #8973). To avoid these problems, use GWM 0.8 with Ganeti version 2.4 or better.

### **3.3 Clusters**

### **3.4 Virtual Machines**

### **3.5 Nodes**

### **3.6 Templates**

# PROJECT INFORMATION

## 4.1 Dependencies

### 4.1.1 Base

**Python** `>= 2.5`

Python-dev

```
sudo apt-get install python-dev
```

---

**Note:** Python-dev is required because some pip packages need it to build dependencies

---

### 4.1.2 Databases

All databases need their required python binding installed in order for Django to connect. Please refer to Django database [documentation](#) if you have any issues.

**MySQL** `python-mysql`

```
pip install MySQL-python
```

**PostgreSQL** `postgresql-psycopg2`

```
pip install psycopg2
```

### 4.1.3 LDAP

LDAP dependencies can be found on the *Dependencies* page.

## 4.2 Compatibility

Ganeti Web Manager is compatible with the following:

**Ganeti 2.4.x–2.6.0.**

Earlier versions are unsupported; they may occasionally work, but should not be relied upon.

**Browsers** Mozilla Firefox >= 3.x, current Google Chrome/Google Chromium.

Other contemporary browsers may also work, but are not supported.

The web-based VNC console requires browser support of WebSockets and HTML5.

**Databases** SQLite, MySQL. New in version 0.10: PostgreSQL has limited support

**Operating Systems** Ubuntu 11.10, Ubuntu 12.04, CentOS 6.

Known to work on Debian 7 and CentOS 5.

Debian 6 should work, provided that pip, virtualenv and fabric are the latest version managed through pip.

## 4.3 Changelog

### 4.3.1 v0.10.2

**Notable Changes:**

- Assigning owners to a VM has been changed slightly.
  - Must have admin/create\_vm permissions to be an owner
  - Groups can be owners
  - Superusers can assign owner to anyone
  - Owner assignment dropdown is now sorted by name (finally)
- Users without perms on any clusters now get a 403 error if they try to go to the VM Creation page. Before they would get to the page but have no clusters to choose from.
- The 5th step of the VM Wizard composing of HVParams is now properly submitting the data to the RAPI
- Refresh button now refreshes data for nodes and vms instead of just the cluster from the RAPI.
- Hostnames are now stored in the database using all lowercase
- More validation on data retrieved from the RAPI
- Updated sshkeys scripts to be more redundant
- Fixed missing CSRF token on password reset page
- VM List pages should be consistent between the global VM list and the cluster VM lists

### 4.3.2 v0.10.1

**Notable Changes:**

- Cluster defaults are now used for all steps in VM Wizard. Previously NIC settings and Disk size had no defaults.
- Pinned Versions of dependencies
- Fixed bug for KVM where kernel path was required, now optional. (KVM only)
- Fixed exception when owner of a VM was a group
- During VM Creation the form now properly raises a validation error if primary node is the same as the secondary node



### 4.3.3 v0.10

#### Notable Changes:

- Ganeti 2.6 Support
- VM Wizard
- Job List - Cluster
- LDAP Support
- Manual Refresh Button
- Notice on Read-Only Clusters
- Sharedfile Disk Template Added
- Docs now ship with product
- Fabfile cleaned up and simplified

### 4.3.4 v0.9.2

- Pinned requirements at Django 1.4. Project not reviewed for Django 1.5 compatibility.

### 4.3.5 v0.9.1

- New Error list page
- Pagination links now correctly show up on the Virtual Machines page
- Migrate button disabled for non-drbd VMs on VM detail page
- VM template fields correctly set NIC and DNS defaults for new VM
- Fix network devices not copied back to new VM page, when deployment fails
- Account password reset form fixed
- Error messages on VMs clearable again

### 4.3.6 v0.9

#### Notable changes:

- Django 1.4
- Ganeti 2.5 support
- Pip 1.0+ support
- Remove PyCurl dependency
- Immediate Shutdown button
- Improved installation process and documentation
- Improved RAPI functionality
- Simplified layout infrastructure
- Fix CSRF Token errors

- Transaction middleware
- Check VM hostnames for illegal characters
- New Help Tips
- Many back-end fixes to improve standards compliance and Django best practices
- Many user interface fixes and improvements

### 4.3.7 v0.8.1

Bugfix release.

#### Bugs fixed:

- CsrfResponseMiddleware removed from settings.py.dist

### 4.3.8 v0.8

#### Notable Changes:

- VM Templates
- Multiple Disks and Nics for VM Creation
- ‘No Install’ option for VM Creation
- CDROM2 Image Path for KVM
- User auto-complete for all username fields
- Rework and stabilisation of Jobs
- User registration is now optional
- CPU info added to node list and detail pages
- Ability to replace disks for a VM on DRBD clusters
- Cached AJAX calls
- Unified json package use (django.utils.simplejson)
- Reduced name collisions with directory reorganizing
- Cache refresh migration moved to post\_migrate hook
- Unified use of CSRF tokens

### 4.3.9 v0.7.2

- Fixed HAYSTACK\_SITECONF default setting
- Updated README to include virtualenv for mod\_wsgi script

### 4.3.10 v0.7.1

- Updated Fabric dependency: Django Object Permissions 1.4.1
- Overview: Used resources was not displaying clusters when used did not permissions

### 4.3.11 v0.7

#### Notable Changes:

- Xen Support
- Internationalization Support (only greek translations.)
- Fabric & Virtual Environment deployment.
- **Improved Navigation:**
  - Search
  - Contextual links added to more pages
  - Breadcrumbs available on most pages
- Object log upgraded to 0.6 includes scalability improvements
- **Object permissions upgraded to 1.4**
  - speed improvements
  - contextual links added to generic views
  - user/group selection widget added for permission editor.
- noVNC updated to latest head, includes better support for future revisions
- Node Evacuation now works properly
- VirtualMachine owner can now be edited
- **Periodic Cache updater**
  - now synchronizes Nodes
  - now runs using twistd
- Nodes can now be imported through the user interface
- Various UI fixes
- Various optimizations to views to improve load times.

### 4.3.12 v0.6.2

- fixing packaging issue with object log

### 4.3.13 v0.6.1

- updating object log to 0.5.1

### 4.3.14 v0.6

#### Notable Changes:

- Nodes are now cached in the database:
- Node detail views are now available, including some admin methods
- VirtualMachines may now be edited, renamed, and migrated.

- Errors while creating virtual machines are now handled better, and can be recovered from
- Django Object Log is now providing logs for all objects tracked by GWM
- Admins can now add ssh keys for other users
- Virtual machine detail page has had its layout updated to be more readable and add more
- fixed bugs preventing syncdb working with postgresql

### 4.3.15 v0.5

#### Notable Changes:

- **Status Dashboard is now the front page for GWM**
  - lists cluster status for admins.
  - lists summary of virtual machines status for users.
  - lists resource usage for the user and groups.
  - error list including job failures and ganeti errors.
- Integrated NoVNC, an HTML5 + WebSockets VNC viewer
- Super users can now view resource usage and permissions for users and groups.
- Virtual machine lists are now paginated for quicker loading
- Ram and CPU quota is now based off running virtual machines
- Improved layout
- Virtual Machines list now properly works for cluster admins

### 4.3.16 v0.4

#### Initial Release

- Caching system
- **Permissions system:**
  - user & group management
  - per cluster/vm permissions
- basic VM management: Create, Delete, Start, Stop, Reboot
- ssh key feed
- basic quota system
- Import tools

## 4.4 History

## 4.5 Design

### 4.5.1 Next

### 4.5.2 1.0

Celery

## 4.6 Contact Information

Here are a couple of ways to get in touch with us.

---

**Note:** Before asking a support related question, please make sure it has not already been answered by searching the mailing list [archive](#) and current [issues](#).

---

- IRC - Join our IRC channel [#ganeti-webmgr](#) on Freenode
- [Mailing List](#) - Send an email to the mailing list
- [Twitter](#) - Tweet us!

## 4.7 Screenshots

Here's a few screenshots of Ganeti Web Manager:

### 4.7.1 Status Dashboard

### 4.7.2 Viewing virtual machines in a cluster

### 4.7.3 Creating a new Virtual Machine

### 4.7.4 Virtual Machine Console

### 4.7.5 Virtual Machine Detail Page

Ganeti Web Manager is licensed under the **GPLv2 or later**. Ganeti Web Manager releases also includes other libraries with separately licensed

- Django Object Permissions - MIT
- NoVNC - LPGL

The screenshot displays the Ganeti Web Manager interface. At the top, the Ganeti logo and 'Web Manager' text are visible. A navigation sidebar on the left includes 'Overview', 'Clusters', 'Virtual Machines', 'Create VM', and an 'Admin' section with links for 'Orphan VMs', 'Import VMs', 'Mooing VMs', 'Users', and 'Groups'. The main content area is titled 'Overview' and contains three sections:

- Cluster Status:** A table listing clusters with columns for Cluster, Version, Memory [GB], Disk [GB], Nodes, and VMs.
 

Cluster	Version	Memory [GB]	Disk [GB]	Nodes	VMs
ganeti-csv	2.2.2	6.67 / 15.7	1007 / 1201	10	06
ganeti-special	2.2.2	243 / 252	2716 / 2799	10	10
ganeti	2.2.2	39.4 / 94.4	558 / 2696	44	1870
gwn	2.2.2	1.07 / 1.96	43.6 / 58.6	10	26
- Resource Usage:** A section for the 'gwn' cluster showing 'Your VMs' (2/5), 'Disk' (13292 / 20000), 'RAM' (2552 / 10000), and 'Virtual CPUs' (12 / 20) with corresponding progress bars.
- Administration:** A section for 'Orphaned VMs' with a link to 'Adopt 32'.

At the bottom of the page, it states 'A Project by the Oregon State University Open Source Lab.'

## 4.8 GPL License

GNU GENERAL PUBLIC LICENSE  
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it

The screenshot shows the Ganeti Web Manager interface. The top navigation bar includes the Ganeti logo and a user login status. The main content area displays the cluster name 'ganeti-dev.osuosl.bak' and a list of virtual machines. The table below shows the details of these VMs.

Name	Node	OS	RAM	Disk Space	vCPUs
✓ dfohlu.osuosl.org	gdev3	Manual (image)	512 MB	20.00 GB	1
✓ dhs-test.osuosl.bak	gdev3	Ubuntu Lucid (image)	512 MB	10.00 GB	1
✓ drp4ip.drupal.org	gdev3	Centos CI (image)	512 MB	10.00 GB	1
✓ helpdesk-dev.osuosl.org	gdev5	Centos Hardened CI (image)	512 MB	10.00 GB	1
✓ nutberry.osuosl.bak	gdev3	Centos CI (image)	512 MB	5.00 GB	1
✓ onod-dev.onod.org	gdev5	Centos Hardened CI (image)	1.00 GB	10.00 GB	1
✓ rtdvva.osuosl.bak	gdev3	Centos Hardened CI (image)	768 MB	10.00 GB	1
✓ stajright.drupal.org	gdev5	Centos CI (image)	2.00 GB	80.00 GB	1
✓ stajright.drupal.org	gdev2	Ubuntu Lucid (image)	2.00 GB	30.00 GB	1

in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the

You are logged in as **peter**. [Logout](#)


Ganeti  
Web Manager

Overview  
Clusters  
Virtual Machines  
Create VM

---

**Admin**

Refresh VMs  
Import VMs  
Missing VMs  
Users  
Groups

### New Virtual Machine

Owner:

Cluster:

Instance Name:

Start up after creation:

DNS Reverse Check:

Disk Template:

Primary Node:

Operating System:

---

#### General Parameters

Virtual CPUs:

Memory:

Disk Size:

Disk Type:

NIC Mode:

NIC Link:

NIC Type:

---

#### Hypervisor Parameters

Kernel Path:

Root Path:

Enable Serial Console:

Boot Device:

CD-ROM Image Path:

**Disk Template**

Disk layout template for the virtual machine on the cluster node.

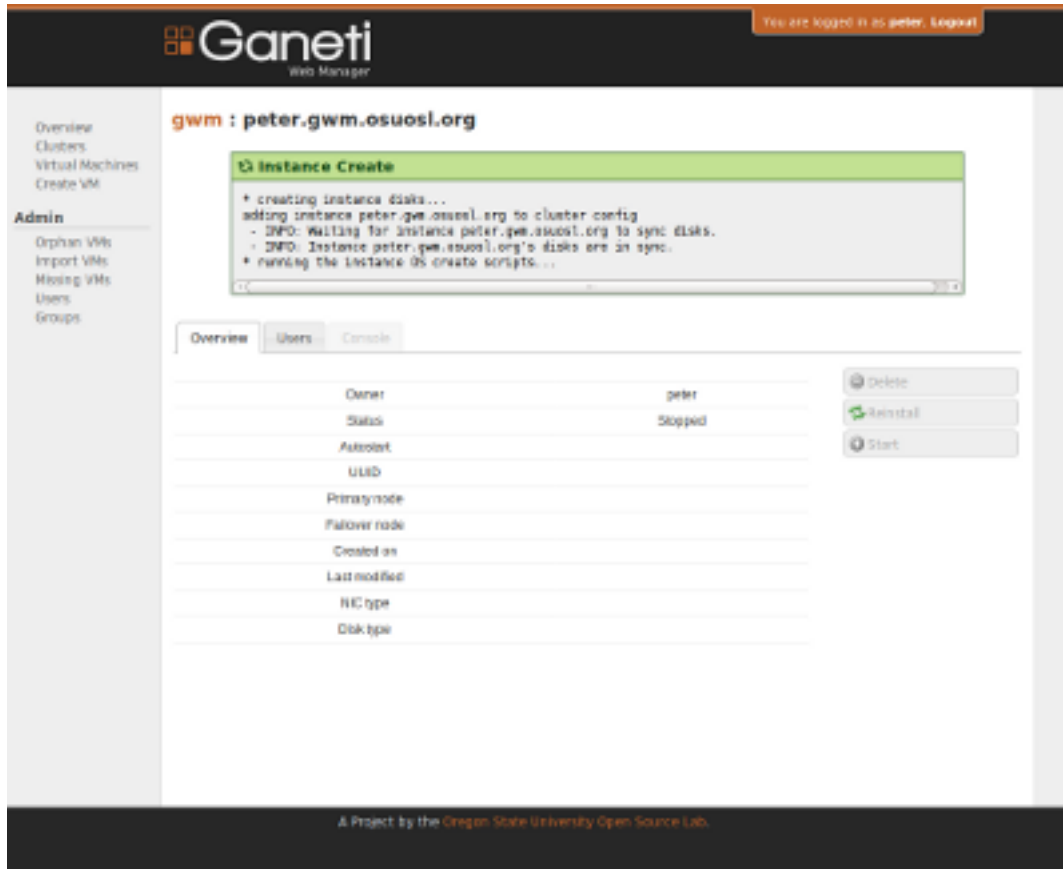
The available choices are:

- **plain** - Disk devices will be logical volumes (e.g. LVM)
- **drbd** - Disk devices will be DRBD (version 8.x) on top of LVM volumes.
- **file** - Disk devices will be regular files (e.g. qcow2)
- **diskless** - This creates a virtual machine with no disks. Its useful for testing only (or other special cases).

If drbd is selected, then a primary and secondary node will need to be chosen unless automatic allocation has been selected. DRBD will allow the virtual machine to use live migration and failover in case one of the nodes goes offline.

A Project by the Oregon State University Open Source Lab.





program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.


GNU GENERAL PUBLIC LICENSE  
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's





en ▼ | You are logged in as **tester**, [Logout](#)

Overview  
Clusters  
Virtual Machines  
Create VM

**Search**

**Admin**  
Orphan VMs  
Import VMs  
Missing VMs  
Import Nodes  
Missing Nodes  
Users  
Groups

### gwm : testvm

Overview
Users
Log
Console

Owner	
Created on	05/07/2011 10:12
Last modified	05/07/2011 10:12
Status	Running
Primary node	<a href="#">gwm1.osuosl.org</a>
Secondary node	
Operating System	Ubuntu Natty ( <i>image</i> )
Autostart	✔
VNC	<a href="#">gwm1.osuosl.org::11191</a>

Change Owner
Edit
Rename
Delete
Reinstall
Shutdown
Reboot

#### Hardware

VCPUs	2
Memory	512 MiB
NIC type	paravirtual

#### Disks

Disk	Type	Size
disk0	paravirtual	2.00 GiB

#### Network Devices

Nic	Mac	IP	Mode	Link
nic0	aa:00:00:b5:e7:eb	None	bridged	br0

#### Hypervisor

ACPI	✔
Boot Order	disk
CDROM Image Path	
Disk Cache	default
Initrd Path	

source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals

of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>
```

```
This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.
```

```
This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.
```

```
You should have received a copy of the GNU General Public License along
with this program; if not, write to the Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.
```

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this

when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w`.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c` for details.
```

The hypothetical commands `'show w'` and `'show c'` should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `'show w'` and `'show c'`; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.
```

```
<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

## 4.9 FAQ

Here are some frequently asked questions, and their answers. If your question isn't answered here, ask on the Freenode IRC network in channel `#ganeti-webmgr` or on the [GWM Google Group](#).

### 4.9.1 I added a virtual machine using the `gnt-instance` command-line tool, and I don't see it in GWM!

Use the "Import VM" page (linked from the admin sidebar) to add those virtual machines to GWM.

### 4.9.2 How do I limit the resources available to a user?

Change the user's quota on the cluster. Default quotas for virtual CPUs, disk space, and memory can be set when adding the cluster. After adding a user to the cluster, their quota will be listed in the "Users" tab of the cluster detail page. Their quota will be listed if it was set from default, or shown as an infinity sign if there is no existing quota. Click on the user's quota to add or edit the amounts of disk space, memory, and CPUs available to that user.

### 4.9.3 What does "Autostart" do?

When the "Autostart" mark on a virtual machine's detail page is a green check mark, it means that the virtual machine will be automatically started if the node reboots. Otherwise, if the mark is a red cross, the virtual machine will only start when a user manually starts it.



### I get the error: “Whoosh\_index not writable for current user/group”

When running GWM through Apache, it is required that the apache user (or www-data) and group can write to the whoosh\_index directory.

```
chown apache:apache whoosh_index/
```



# DEVELOPMENT

## 5.1 Release Schedule

## 5.2 Development Process

### 5.2.1 Git Workflow

### 5.2.2 Issue Triaging

## 5.3 Filing Issues

Please follow this guide when filing new issues for Ganeti Web Manager

### 5.3.1 Trackers

#### Bugs

Bugs are well...bugs. Things that are broken. Pieces of the project that don't work as expected.

#### Features

Features are full blown enhancements to the project that vastly improve the usability experience. They are generally referenced with a phrase like "The ability to..."

#### Enhancements

Enhancements are minor updates to features. They are not errors encountered in features. They are an extension to a feature.

### 5.3.2 Status

Status reflects where a ticket stands in the filing process.

The available status' are:

- New

- In Progress
- Won't Fix
- Duplicate
- Blocked
- Feedback
- Resolved
- Upstream
- Closed
- Rejected
- Can't Reproduce

### 5.3.3 Priorities

- Low
- Normal
- High
- Urgent
- Blocker

### 5.3.4 Categories

### 5.3.5 Difficulty

An estimation of how hard a bug, feature, or enhancement will be and how long it may take.

- Easy
- Medium
- Hard

## 5.4 Development Tools

## 5.5 Documentation

This article is pretty meta.

### 5.5.1 Todo

All of these items are things the documentation is lacking and any contributions will be happily accepted

## 5.6 Current Contributors

Below is a list of contributions made to the Ganeti Web Manager project.

### 5.6.1 Main Contributors

- **Peter Krenesky** (*kreneskyp*) Sep. 2010 - Present
- **Corbin Simpson** (*MostAwesomeDude*) Oct. 2010 - Present
- **Trevor Bramwell** (*bramwelt*) Sep. 2010 - Present
- **Scott White** (*whiteau*) Oct. 2010 - Mar. 2011
- **Emily Dunham** (*edunham*) Apr. 2011 - Present

### 5.6.2 Google Code In Contributors, 2010

- **Jonathan Smith** (jaydez) - j.jaydez@gmail.com
- **Kyle Cesare** (kylc) - kcesare@gmail.com
- **Kapeel Sable** (kapeel) - kapeelsable+p@gmail.com
- **Samuel Dolean** (samm) - selony24@gmail.com
- **David Zderic** - dzderic@gmail.com
- **Crypt Wizard** - cryptwizard@gmail.com
- **Piotr Banaszekiewicz** (pbnan) - piotr@banaszekiewicz.org

### 5.6.3 Other Contributors

- **Apollon** (apoikos) - apollon@noc.grnet.gr
- **Faidon Liambotis** (paravoid) - faidon@admin.grnet.gr
- **Leonidas Pouloupoulos** - leopoul@noc.grnet.gr

## 5.7 Developers

### 5.7.1 Git Access

#### Read-only Git Access

```
git clone git://git.osuosl.org/gitolite/ganeti/ganeti_webmgr
```

### Commit Access

Commit access requires an account on our gitolite server. You must Submitting Patches and establish a relationship with us prior to being granted commit access.

```
git clone git@git.osuosl.org:ganeti/ganeti_webmgr
```

### 5.7.2 Branch Layout

We use [Git-flow](#) for managing our branches. Read about the [branching model](#) and why you should use it too.

- **master** - Releases only.
- **develop** - Mostly stable development branch. Small changes only. It is acceptable that this branch have bugs, but should remain mostly stable.
- **feature/<name>** - Large code changes and new features.

### 5.7.3 Coding Standards

We follow [PEP 8](#), “the guide for python style”.

In addition to PEP 8:

- Do not use backslash continuations. If a line must be broken up, use parenthetical continuations instead.

### Units

Try to write modular code. Focus on isolating units of code that can be easily analyzed and tested. For sanity purposes, please try to avoid mutually recursive objects.

### JSON

If you need a JSON library, the import for this codebase is “`from django.utils import simplejson as json`”. See [#6579](#) for more information.

### Testing

Ganeti Web Manager has a fairly complete test suite. New code should have matching tests. Before committing code, run the suite for Ganeti Web Manager and [Object Permissions](#)

```
./manage.py test ganeti_web
./manage.py test object_permissions
```

### Clean up after yourself

Remember to tear down any resources you set up in your tests. Don't use “`YourModel.objects.all().delete()`” to clean up your objects; it could be hiding bugs. Clean up exactly the resources you created.

## Test your setups and teardowns

To speed up analysis of broken tests, if you have a `setUp()` or `tearDown()` in a `TestCase`, add a `test_trivial()` method which is empty. It will pass if your `setUp()` and `tearDown()` work.

## Views

All views should be thoroughly tested for security, checking to ensure that the proper http codes are returned.

- Test Anonymous User access
- Test Permission based access
- Test Superuser based access

Check for invalid input.

- missing fields
- invalid data for field

## Templates & Javascript

The test suite does not yet include selenium tests for verifying javascript functionality. Some basic tests can be performed using django's test suite:

- Check objects in the context: forms, lists of objects, etc.
- Check for existence of values in forms.

## Adding features

When adding a feature to GWM, please remember to include:

## Help tips

The gray box with a green title bar that appears on the right side of the page when you focus on a form field is a help tip. To add one for a new field, add it to the file which corresponds to your field's form in the `ganeti_web/templates/ganeti/helptips/` directory.

## Internationalization

Ganeti Web Manager is designed to support translation to other languages using Django's `i18n` machinery. If you add text that will be displayed to the user, please remember to format it for translation:

```
{% trans "this text will be displayed in the correct language" %}

{% blocktrans %}
    Here is a some text that will be displayed
    in the correct language but would not
    fit well in a single line
{% endblocktrans %}
```

[Django's i18n page](#) has more information about this.

## 5.7.4 Submitting Patches

Patches should either be attached to issues, or emailed to the mailing list. If a patch is relevant to an issue, then please attach the patch to the issue to prevent it from getting lost.

Patches must be in git patch format, as generated by `git format-patch`.

```
git commit
git format-patch HEAD^
```

To create patches for all changes made from the origin's master branch, try:

```
git format-patch origin/master
```

For more information, see the man page for `git-format-patch`.

Sending emails to the list can be made easier with `git send-mail`; see the man page for `git-send-email` for instructions on getting your email system to work with git.

If there are multiple patches comprising a series which should be applied all at once, git pull requests are fine. Send a rationale for the pull request, along with a git pull URL and branch name, to the mailing list.

## 5.8 Writing Tests

Any assert statement will take the optional kwarg of `'msg'`. This kwarg will be output instead of an error in the test. It is a very useful argument for debugging.

```
self.assertTrue(False, msg="And what else floats? A duck!")
```

### 5.8.1 Forms

Here are a few pointers for testing forms.

If there is an error in the test for `form.is_valid()` a good way to find out what form fields are giving trouble is by checking `form.errors`.

```
form = MyFormWithErrors()
self.assertTrue(form.is_valid(), msg=form.errors)
```

All Django forms generally accept `'initial'`, `'instance'`, and `'data'` keyword arguments, but forms will behave differently depending on which of these arguments you pass in.

#### **data**

This is probably the easiest of all the form tests to work with, because it is what you are so used to seeing.

```
# data is a dictionary
form = MyForm(data)
self.assertTrue(form.is_bound)
self.assertTrue(form.is_valid())
for field in form.fields:
    self.assertEqual(data[field], form.cleaned_data[field])
```

#### **initial**

Initial is a unique case in which I have not found an easy way to check if the initial values of the form were properly set. Checking against `form.fields[field].initial` will not work as this is set to the value of the initial keyword argument originally passed into the form field.



```
# initial is a dictionary
form = MyForm(initial=initial)
self.assertFalse(form.is_bound)
self.assertFalse(form.is_valid())
for field in form.fields:
    self.assertEqual(initial[field], form.fields[field].initial) # This will *NOT* work!
```

### instance

Forms that are passed the instance keyword argument will have set the ‘instance’ property on form. Thus you can test form.instance against the instance you passed in.

```
# instance is an instance of a model
form = MyModelForm(instance=instance)
self.assertEqual(instance, form.instance)
```

## 5.9 About the search system

The search system in Ganeti Web Manager utilizes three main technologies:

- [Haystack](#) - A model-based search system for Django
- [Whoosh](#) - A pure-Python indexing and searching library
- [jQuery UI Autocomplete widget](#) - Displays suggestions to input boxes as the user types

Below, I will discuss the different components of the search system.

### 5.9.1 Haystack

Haystack is the meat ‘n’ potatoes of the search system, involved in every aspect. Mainly it does the following:

- Defines the search set in terms of GWM models in `/ganeti_web/search_indexes.py`
- Manages making queries and indexing the search back-end (currently Whoosh).
- Provides search-specific forms and templates (GWM only uses one search result template: `/templates/search/search.html`)

Find out more about defining search indexes with the [Haystack SearchIndex API](#). To find out more about Haystack in general, see [its documentation](#).

### 5.9.2 Whoosh

Whoosh is a pure-Python indexing and searching library that Haystack uses as a search back-end. The developer actually doesn’t need to interact with Whoosh directly.

### 5.9.3 Of indexing and DB performance

Haystack is currently set to do live indexing. This means that the search index gets updated every time an included model is updated in the database. This means the index will always be up-to-date, but has the potential to severely hamper performance when dealing with a lot of database changes.

Indexing behavior is set when the search set is defined in `/ganeti_web/search_indexes.py` If database performance starts to become an issue, try using `SearchIndex` instead of `RealTimeSearchIndex`, and run:

```
$ ./manage.py update_index
```

from time-to-time. For more information, please see the [Haystack documentation on the subject](#).

### 5.9.4 jQuery UI Autocomplete widget

The Autocomplete widget suggests search results in real-time as the user types a query. This is facilitated through two main components:

- jQuery UI Autocomplete widget itself
- An autocomplete Django view `ganeti_web/views/search.py` that supplies Autocomplete with suggestion data to display

Basically, the Autocomplete widget calls the autocomplete view as the user types, and fills a pop-up box underneath the input box with search suggestions. The JavaScript logic can be found in `/static/js/autocomplete_search.js`, and the search view can be found in `ganeti_web/views/search.py`. Both of these files contain details about how the suggestion data is structured, sent, and processed.

## 5.10 Selenium test specs

Since the closest that GWM has to a formal specification seems to be the bug list, this document is a brainstorm of the behaviors which are expected of Ganeti Web Manager and can be tested using Selenium. The specs will be listed here in two categories: Testing that it does what it should, and testing that it doesn't do what it shouldn't.

*italicized sections aren't specific enough*

When I say that B should happen “immediately” after A, I mean that one should not have to refresh the page or wait more than a second or two for B to show up after A occurs. Selenium tests can run really fast, so it would probably be acceptable (if necessary) to build in a half-second or so of wait time between A and B, since the succession of events would still seem “immediate” to a user.

### 5.10.1 gwm SHOULD:

#### User

–ANYONE–

- Create New User: From login page “Not a member” link or (admin only) users page Add User button
  - Display correct error messages for each invalid field in create form
  - Display correct success message when valid form is submitted

–ADMIN–

- New user appears in user list page accessed from sidebar link
  - Correct email address is displayed
  - Correct user creation time displayed
  - Username links to user detail page
- Detail page displays correct username, email, join date, active, and admin fields
- New user appears in user list dropdown menu for adding permissions

- Edit link leads to edit form pre-populated with user’s info
- Edit form shows correct error when invalid options are entered
  - Catches mismatched passwords
  - Forbids changing username to one that’s already taken
  - If ‘active’ status changed, this change must appear on user detail & user list pages
  - *If superuser status changed, this change must be reflected by detail/list pages and in the availability of adding permissions from user detail page.*
- Changes made via user edit form are shown in user list and user dropdown lists
- Remove User link displays popup saying “Remove user: <Username>” popup... Clicking “ok” deletes user, “cancel” returns to unaltered user list page

–OTHER TESTS RELATED TO USER–

- Resource usage displayed in tab of detail page should change when used resources change
- When logged in as user, should be able to change own email address/password
- ADMIN should be able to add cluster or add VM to user’s permissions from tab of detail page, and it should show up in the list immediately after being added
- A test which creates a log (probably done while logged in as user) should check that the logged action is displayed correctly in the log tab of the user detail page (same with User Actions)

## VM:

These tests should be run after the cluster is added. Ideally the test cluster will have more than one page of VMs.

- VM Detail Page shows list of all the VMs that should be there in the test cluster
  - Clicking any heading in the list causes entire list to be sorted by that attribute
  - If >1 page of VMs, number buttons cause correct page of VMs to be shown
- Add VM button on detail page links to same add page as Create VM link in sidebar..
- Owner, Cluster, & other dropdown menus only offer valid options (don’t allow VM to be created with inactive user as owner?)
- Help Tip displays text appropriate to box most recently clicked
- Fails gracefully with bad data
  - Non-generic error message if invalid DNS name and DNS Name Check box was checked

## Group

- Clicking Add Group button on Groups page opens Add Group dialog
  - Typing a name in the dialog and either hitting enter or clicking save creates the group with that name
  - Group is displayed in group detail page with name as a link to group, 0 members, 0 admins.
  - Edit link on detail page lets you change group’s name
    - \* New name is reflected in list immediately if you change name from edit link
- Link in groups list takes you to the group detail page
  - Log tab lists group’s creation and the user who created it

- Users page:
  - When user added via Add User link,

–OTHER TESTS RELATED TO GROUP–

- Permissions on cluster/VM
- Correct display of resource usage

## Cluster

- *If cluster is added with no username/password credentials? If quota given but no credentials? What should errors be?*

### 5.10.2 gwm should NOT:

- *Show admin things to non-admins*
  - *Search Results*
  - access admin-only url by typing the url in, if not logged in as admin
  - *edit user page?*

## 5.11 References

### 5.11.1 Working With The RAPI

There are several places to view documentation detailing how to interface with the python RAPI, though none of them are comprehensive.

The following is a list of five different sources where information relating to the RAPI can be found.

1. The Ganeti RAPI [PyDocs](#)
2. The Ganeti RAPI [HTML Docs](#)
3. The `gnt-instance` [man-page](#)
4. The `rapi client` code contained in the [upstream](#) ganeti project
5. The `rapi-tests` which are also contained in the [upstream](#) ganeti project

### RAPI in a Python Shell

Start up a python shell using the `manage django` script.

```
$ ./manage.py shell
```

In the python shell import `client.py` from `util`.

```
>>> from util import client
```

Assign a variable to the `rapi client` and pass in the name of the cluster as a string to the `GanetiRapiClient` object.

```
>>> rapi = client.GanetiRapiClient('my.test.cluster')
```

- Note - For R/W access to the cluster you need to pass in 'username' and 'password' as kwargs to the Ganeti-RapiClient object. Replace USERNAME and PASSWORD with the correct cluster R/W credentials

```
>>> rapi = client.GanetiRapiClient('my.test.cluster', username='USERNAME', password='PASSWORD')
```

- Optional - Setup PrettyPrinter to prettify the output of RAPI functions that return dictionaries.

```
>>> import pprint
>>> pp = pprint.PrettyPrinter(indent=4).pprint
```

Now you are able to prettify output like this:

```
>>> pp(rapi.GetInfo())
```

- RAPI commands can now be accessed as functions of the rapi variable.

```
>>> rapi.GetInstance('my.test.instance')
```

### 5.11.2 REST API

Development of the RESTful API for GWM started as a Google Summer of Code 2011 project. The developer, Bojan Suzic, can be reached at [bojan.suzic@gmail.com](mailto:bojan.suzic@gmail.com). The mentor for this project is Peter Krenesky.

The current version of the page describes its status as of ~end August 2011. Currently it supports a fair amount of the underlying functionality.

Additionally, this page is intended as a development resource and documentation source for the project.

The project code can be reached at <https://github.com/bsu/GWM2>. The additional code (changes in other related modules) can be found at the following locations:

- [https://github.com/bsu/django\\_obj\\_perm](https://github.com/bsu/django_obj_perm)
- [https://github.com/bsu/django\\_obj\\_log](https://github.com/bsu/django_obj_log).

Users and developers are encouraged to try this addon and send their suggestions/bug reports/improvements either directly to the author, using the [GWM mailing list](#) or [issue tracker](#). Please put the line **\*\*[REST-API]\*\*** in the subject if you are sending email message.

REST API for GWM can now be considered as a beta software. For the further development the following is proposed:

Roadmap:

- completion of unit tests
- relocate parts of endpoints which may contain long answers - that variables should be accessed separately (such as object logs)
- further refinement of the code and documentation
- work on further integration (like cloud driver)

The **users/developers/visitors are advised to test** the code and **submit the comments/notices/wishes**. Comments can be submitted either directly to the author, here on this wiki or using the [Redmine ticket](#)

The version containing significant changes to this version may be expected in **November 2011**.

#### About this documentation

This documentation covers basic functionality of the REST API. It consists of the subsections, referring to particular endpoints forming the API. As an endpoints referred are application resources exposed as URIs through appropriate

hierarchy. Currently, the system exposes the following resources as REST API endpoints: **User, Group, Virtual Machine, Cluster, Node, Job**. These are accessible in the form of CRUD operations using HTTP protocol.

By default, each API endpoint returns a list of resources.

For example:

```
/api/vm/
```

would return a list of Virtual Machines in the system, while the particular VM resource is accessed through:

```
/api/vm/1/
```

where 1 represents identifier of particular Virtual Machine. This way only that particular resource is returned. Each resource contains fields related to particular instance in the system. The field types and their properties are described further in the documentation, as a part of related subsections.

For each endpoint described is basic scheme of its resource representation. The scheme includes name, type and description of the particular fields, Additional properties described are possibility for the field to be modified and whether the field returned may be excluded from the representation e.g. nullable. The later one might be useful for clients to prevent unexpected behavior.

While **type** field represents basic data types, it should be noted that **related** type points to other resource in the system. For example:

```
...
<cluster>
...
<virtualmachine>
/api/vm/5
</virtualmachine>
</cluster>
...
```

says that particular cluster resource includes virtualmachine resource, described by related URI. Therefore, if necessary, the complete resource referred at that point may be obtained through provided URI.

This documentation trends to provide as much as complete list of resources and their schematic description. However, due to the level of deepness and limitations of the current wiki system, in some cases this representation is simplified and explained in words rather than in tabular form. It should be noted that all these descriptions are already included in the system.

Using:

```
http://somesite.com/api/?format=xml
```

user is able to get the list of resources exposed, while with the help of:

```
http://somesite.com/api/resource/schema/?format=xml
```

the system returns detailed schema of resource representation and field properties in XML format. Therefore, the user is always able to take a look and check detailed description about a resource, if the one provided here in documentation is not detailed and clear enough.

### Design principles

This API aims to expose the resources of Ganeti Web Manager, making suitable for integration into other systems or just performing of simple operations on resources. It does not aim to expose all resources and functions contained in the software, but only the set deemed necessary in order to support its business functions. Currently, it means that the most of the functionality present in the web interface is available for usage and manipulation also using this REST API.

## Installation

The most of the code of this add-on comes under `/api` directory of GWM distribution. Other code changes are done in some of views and dependent modules (like `django_object_log` and `django_object_permissions`). Its inclusion in the GWM is done in `/urls.py` via:

```
urlpatterns = patterns('',
    ...
    (r'^$', include('api.urls')),
    ...
)
```

The prerequisite for running RESTful API over Ganeti Web Manager is to have **django-tastypie** installed. The latest active version/commit of **tastypie** should be used in order to support **ApiKeys** based authentication. That means, as of time of writing this documentation, that **tastypie** should be installed manually. Additionally, the following line in `tastypie/authentication.py`:

```
username = request.GET.get('username') or request.POST.get('username')
```

should be changed to:

```
username = request.GET.get('username') or request.POST.get('username')
or request.GET.get('user') or request.POST.get('user')
```

This is the known issue with **tastypie** already reported in its bug system. If not changed, the part **username** in `/api/user/?api_key=xxx&username=xxx` will produce error message during browsing the main user endpoint. This change makes **tastypie** to accept **user** for authentication instead of **username**. Later produces collision with the field of the same name under **User** model class.

The next change related to the installation of the module is inclusion of **'tastypie'** in `INSTALLED_APPS` of `settings.py`. This will produce necessary tables during installation/migration.

## Development

The code is prepared as a part of GSoC activities, and therefore by person not being a part of narrowed GWM development team before. As a such, the main principle to be followed is to try not to interfere too much with existing code. It implies further that the resulting code should be considered as a simple to install add-on. The core business logic of the GWM have not been changed or altered. The most changes done on GWM code are of technical nature, trying to make functions/views accessible to REST backend interface additionally. The code has been committed to separate repository. I tried mostly to perform smaller commits in size, in order to make the code and changes easily readable and trackable.

The framework used to introduce RESTful interface is **django-tastypie**. It has been selected after initial research and testing of several popular Python/Django/REST frameworks. The system supports both XML and JSON as input/output serializations.

## Authentication and Authorization

The authentication is done using **API keys**. For each user the appropriate API key is generated automatically. The key can be renewed/recreated using **POST** request and appropriate action inside API. The access to the system looks like in the following example:

```
http://localhost:8000/api/?format=xml&api_key=381a5987a611fb1f8c68ffad49d2cd2b9f92db71&user=test
```

---

**Note:** **username** initially supported by **tastypie** has been replaced with **user** in the example query above. The changes and reasons are described in the installation section of this document.

Authorization is completely dependent on Django's authorization system. The existing views from the GWM have been used to expose the most of resources available. Those views are already integrated in Django's authorization system. Therefore, this API should not contain critical security flaws or problems and should be easier to maintenance.

### REST API endpoints

#### /api/user

This endpoint exposes data and operations related to the user management. The following table provides the descriptions of the fields:

#### Explanations for particular list elements **Container: ssh\_keys**

The elements of the list are denoted as **value** nodes, containing particular ssh key for the user in the form of **string hash**

Example:

```
<ssh_keys type="list">
<object type="hash">
<id type="integer">1</id>
<key>ssh-rsa
A398kuwNzaC1yc2EAAAADAQABAAQDI2oqyrleSvAg4CV5A/4ZZ2fTEFAYU1W2i8373zspaJCS00eHI1+v4fGeIzH7CFokbM98:
/E9ucXR4xcxo77sxGSGH8hiS89aUcHmPKyRY1Yj5TwqkZopxYTFmeUhhkP9e5Yr1TRXMdhMsIXqXAKRu j j y SycQ45QLqdYOHbfOh
/aq2vvOoyiGo9vaiIfqbtLjqk jwecDGykesw1c9d07vH53myiLLLkAGGk4KudKSWV6ZxK0ap3/olzzJ3HZpk5MAe15ELX6XuT8Vm
xxx@gmail.com
</key>
</object>
<object>
<id type="integer">2</id>
<key>ssh-rsa
7398kuwNzaC1yc2EAAAADAQABAAQDI2oqyrleSvAg4CVjska jsla jwFAYU1W2i8373zspaJCS00eHI1+v4fGeIzH7CFokbM98:
/E9ucXR4xcxo77sxGSGH8hiS89aUcHmPKyRY1Yj5TwqkZopxYTFmeUhhkP9e5Yr1TRXMdhMsIXqXAKRu j j y SycQ45QLqdYOHbfOh
/aq2vvOoyiGo9vaiIfqbtLjqk jwecDGykesw1c9d07vH53myiLLLkAGGk4KudKSWV6ZxK0ap3/olzzJ3HZpk5MAe15ELX6XuT8Vm
yyy@gmail.com
</key>
</object>
</ssh_keys>
```

#### Containers: user\_actions and actions\_on\_users

This is the list of **objects**, each object consisting of nullable fields denoted as **obj1**, **obj2**, **user**, **action\_name**. The both containers share the representation. The difference between these is the fact that first describes actions performed by user, while the second one describes actions performed on user (by administrator, for instance). The both containers provide read only information.

Example:

```
<user_actions type="list">
<object type="hash">
<obj1>/api/vm/3/</obj1>
<timestamp>2011-07-31T15:23:45.533479</timestamp>
<obj2>/api/job/68/</obj2>
<user>/api/user/2/</user>
<action_name>VM_REBOOT</action_name>
</object>
```



```

<object type="hash">
<obj1>/api/vm/3/</obj1>
<timestamp>2011-07-31T17:04:02.333061</timestamp>
<user>/api/user/2/</user>
<action_name>EDIT</action_name>
</object>

```

### Container used\_resources

This list consists of **object** elements, each containing **resource**, **object** and **type**. The field **object** represents related resource for which the system resources consumption is given. The **type** is **string** describing the object type using internal descriptions (like **VirtualMachine** for virtual machine). The **resource** contains subfields **virtual\_cpus**, **disk** and **ram**, each of type **integer** and representing actual consumption of the particular system resource in system's default dimension (e.g. MBs for RAM consumption).

Example:

```

<used_resources type="list">
<object type="hash">
<resource type="hash">
<virtual_cpus type="integer">0</virtual_cpus>
<disk type="integer">0</disk>
<ram type="integer">0</ram>
</resource>
<object>/api/vm/3/</object><
type>VirtualMachine</type>
</object>
<object type="hash">
<resource type="hash">
<virtual_cpus type="integer">0</virtual_cpus>
<disk type="integer">0</disk>
<ram type="integer">0</ram></resource>
<object>/api/vm/11/</object>
<type>VirtualMachine</type>
</object>
</used_resources>

```

### Container permissions

**Permissions** contains elements describing particular resource type, each further containing a list of resources. The primary **elements** are described as **Cluster**, **VirtualMachine**, **Group**. Their list member main nodes are described as **object**, containing **object** reference (related resource) for which the permissions are set, and the **permissions** list containing list of **values** as strings, describing permission type in internal format (like **create\_vm**).

Example:

```

<permissions type="hash">
<Cluster type="list"/>
<Group type="list"/>
<VirtualMachine type="list">
<object type="hash">
<object>/api/vm/3/</object>
<permissions type="list">
<value>admin</value>
<value>power</value>
<value>tags</value>
</permissions>
</object>
<object type="hash">
<object>/api/vm/11/</object>

```

```
<permissions type="list">
<value>admin</value>
</permissions></object>
</VirtualMachine>
</permissions>
```

**Manipulation and operations using POST/PUT/DELETE methods** The fields marked as non-readonly (table above) can be subject of further manipulation and operations. **The same applies to the rest of the document - those fields can be automatically updated or deleted by performing analog request.** In order to maintain consistency with REST approach, the **PUT** method is used on currently available resources with purpose to change or update them. On another side, **POST** method is used either to generate new resources, or to perform predefined actions on currently existing resources.

The following example demonstrates changing of users lastname and status in system (disabling its account). Request URI:

```
PUT /api/user/1/?api_key=xxxxx&username=yyyyy
```

Request header:

```
Content-Type: application/json
Accept: application/json
```

Request payload:

```
{"last_name": "New LastName", "is_active": false}
```

Server response:

```
HTTP/1.1 204 NO CONTENT
Date: Sat, 06 Aug 2011 11:18:25 GMT
Server: WSGIServer/0.1 Python/2.7.1+
Vary: Accept-Language, Cookie
Content-Length: 0
Content-Type: text/html; charset=utf-8
Content-Language: en
```

The next example demonstrates generating of new Api key for the user:

Request URI:

```
POST /api/user/2/?api_key=xxxxx&username=yyyyy
```

Request header:

```
Content-Type: application/json
Accept: application/xml
```

Request payload:

```
{"action": "generate_api_key"}
```

Server response:

```
HTTP/1.1 201 CREATED
Date: Sat, 06 Aug 2011 11:21:56 GMT
Server: WSGIServer/0.1 Python/2.7.1+
Vary: Accept-Language, Cookie
Content-Type: text/html; charset=utf-8
Content-Language: en
```

Response body:

```
<?xml version='1.0' encoding='utf-8'?>
<object>
<api_key>de0a57db0ce43d0f3c52f83eaf33387750ac9953</api_key>
<userid>2</userid>
</object>
```

For the API Key manipulation under `/api/user/` endpoint implemented are two POST actions: **generate\_api\_key**, as demonstrated in the example above, and **clean\_api\_key**. The former generates a new API key for the user and returns it in the response, while the later one cleans user's API key. This way its access to the system using REST API is disabled, but the standard access using web interface is untouched.

Additionally, two POST actions are implemented for user-group membership manipulation.

### `/api/group`

This endpoint exposes data and operations related to the group management. The following table summarizes supported fields.

#### Container: `actions_on_group`

This is the list of **objects**, each object consisting of nullable fields denoted as **obj1**, **obj2**, **user**, **action\_name**. This container describes actions performed on the group (by administrator, for instance) in the form of read-only information. Please note that inclusion of **obj1** and **obj2** depends on the action type, e.g. one of these may be omitted.

Example:

```
<actions_on_group type="list">
<object type="hash">
<obj1>/api/group/1/</obj1>
<timestamp>2011-07-29T08:28:24.566903</timestamp>
<user>/api/user/1/</user>
<action_name>CREATE</action_name>
</object>
<object type="hash">
<obj1>/api/cluster/1/</obj1>
<timestamp>2011-07-29T08:28:59.854791</timestamp>
<obj2>/api/group/1/</obj2>
<user>/api/user/1/</user>
<action_name>ADD_USER</action_name>
</object>
</actions_on_group>
```

#### Field: `users`

This simple field contains a list of users belonging to the group. The type of the resource is **related**, which means that it points to the URI representing the resource. Example:

```
<users type="list">
<value>/api/user/2/</value>
<value>/api/user/3/</value>
</users>
```

#### Container `used_resources`

The syntax used here is the same as used in the `<object>User</object>` resource. For more information and example, please refer to the user section of this document.

#### Container `permissions`

The syntax used here is the same as used in the `<object>User</object>` resource. For more information and example, please refer to the user section of this document.

### Manipulation actions

#### `/api/vm`

This endpoint exposes methods for VirtualMachine inspection and manipulation.

**Important:** as the attributes exposing VM object are related to many other objects and many calls are done on different views, here the somewhat different approach to attribute exposure is used. At the main point `/api/vm/`, which provides a list of virtual machines, only the basic attributes of VM are provided. However, when the particular object is called, sad `/api/vm/3/`, the system returns additional set of its attributes. This is due to need to perform additional calls which introduce network latency. Performing all those calls at once for all virtual machines could produce unnecessary overhead.

Fields exposed (main endpoint):

Fields exposed (additionally, particular object):

#### **Containers: actions\_on\_vm and permissions**

The format and members of those lists are similar to previous mentioned fields, e.g. in `cluster` endpoint. For detailed description, please refer to those.

The field `actions_on_vm` contains object(s) taking part in action, user initiated the action, timestamp and the internal description of the action in form of the string. The field `permissions` lists users and groups (as related fields) which have any form of permissions on virtual machine.

#### **Operations supported**

Operations on VM are accomplished in form of action. Action is initiated using POST request. Example:

```
POST /api/vm/7
{"action": "shutdown"}
```

In this example, user initiates `@POST@` request on Virtual Machine described with `@id=7@`. The action type is described in field `@action@` in request header.

After the action is initiated, server sends back response. Example:

Header:

```
HTTP/1.1 200 OK
Date: Wed, 27 Jul 2011 18:39:31 GMT
Server: WSGIServer/0.1 Python/2.7.1+
Vary: Accept-Language, Cookie
Content-Type: application/json
Content-Language: en
```

Body:

```
{"end_ts": null, "id": "138722", "oplog": [], "opresult": [null],
"ops": [{"OP_ID": "OP_INSTANCE_SHUTDOWN", "debug_level": 0, "dry_run":
false, "ignore_offline_nodes": false, "instance_name":
"ooga.osuosl.org", "priority": 0, "timeout": 120}], "opstatus":
["running"], "received_ts": [1311791966, 837045], "start_ts":
[1311791966, 870332], "status": "running", "summary":
["INSTANCE_SHUTDOWN(ooga.osuosl.org)"]}
```

The following actions and parameters are supported:

### `/api/cluster`

This endpoint describes fields and operations available for the Cluster.

#### **Containers: available\_ram and available\_disk**

The first container provides information about status of the RAM in the cluster. Analogously, the second one provides information about disk space in the cluster.

Example:

```
<available_ram type="hash">
<allocated type="integer">1024</allocated>
<total type="integer">2004</total>
<used type="integer">874</used>
<free type="integer">980</free>
</available_ram>
```

#### **Containers: missing\_ganeti and missing\_db**

Here the names of the missing machines are provided in the simple form. The former container describes machines missing in the Ganeti, while the former contains the machines missing in the database.

Example:

```
<missing_db type="list">
<value>3429_test</value>
<value>breakthis.gwm.osuosl.org</value>
<value>brookjon.gwm.osuosl.org</value>
<value>noinstall2.gwm.osuosl.org</value>
</missing_db>
```

#### **Container: quota and default\_quota**

This container returns the quotas for the user performing request. If the user is not found or do not have a quotas assigned, default quota is returned. If there are no values for the specific quota element, null is returned. Default\_quota container is additionally returned for the case that quota for the user if found.

Example:

```
<quota type="hash">
<default type="integer">1</default>
<virtual_cpus type="null"/>
<ram type="null"/>
<disk type="null"/>
</quota>
```

#### **Container: info**

This element provides extensive information related to the cluster. These information are used internally in Ganeti Web Manager to render specific pages. As of level of detail used, the field contained here will not be described but just provided in detail in example.

```
<info type="hash">
<default_iallocator/>
<maintain_node_health type="boolean">False</maintain_node_health>
<hvparams type="hash">
<kvm type="hash">
<nic_type>paravirtual</nic_type>
```

```
<use_chroot type="boolean">False</use_chroot>
<migration_port type="integer">8102</migration_port>
<vnc_bind_address>0.0.0.0</vnc_bind_address>
<cdrom2_image_path/>
<usb_mouse/>
<migration_downtime type="integer">30</migration_downtime>
<floppy_image_path/>
<kernel_args>ro</kernel_args>
<cdrom_image_path/>
<boot_order>disk</boot_order>
<vhost_net type="boolean">False</vhost_net>
<disk_cache>default</disk_cache>
<kernel_path/>
<initrd_path/>
<vnc_x509_verify type="boolean">False</vnc_x509_verify>
<vnc_tls type="boolean">False</vnc_tls>
<cdrom_disk_type/>
<use_localtime type="boolean">False</use_localtime>
<security_domain/>
<serial_console type="boolean">False</serial_console>
<kvm_flag/>
<vnc_password_file/>
<migration_bandwidth type="integer">32</migration_bandwidth>
<disk_type>paravirtual</disk_type>
<migration_mode>live</migration_mode>
<security_model>pool</security_model>
<root_path>/dev/vda3</root_path>
<vnc_x509_path/>
<acpi type="boolean">True</acpi>
<mem_path/>
</kvm>
</hvparams>
<default_hypervisor>kvm</default_hypervisor>
<uid_pool type="list">
<objects>
<value type="integer">8001</value>
<value type="integer">8030</value>
</objects>
</uid_pool>
<prealloc_wipe_disks type="boolean">False</prealloc_wipe_disks>
<primary_ip_version type="integer">4</primary_ip_version>
<mtime type="float">1308862451.98</mtime>
<os_hvp type="hash"/>
<osparams type="hash"/>
<uuid>0b3b2432-a8e1-4c17-a99b-87303841cb95</uuid>
<export_version type="integer">0</export_version>
<hidden_os type="list"/>
<master>gwml.osuosl.org</master>
<nicparams type="hash">
<default type="hash">
<link>br0</link>
<mode>bridged</mode>
</default>
</nicparams>
<protocol_version type="integer">2040000</protocol_version>
<config_version type="integer">2040000</config_version>
<software_version>2.4.2</software_version>
<tags type="list"/>
```

```

<os_api_version type="integer">20</os_api_version>
<candidate_pool_size type="integer">10</candidate_pool_size>
<file_storage_dir>/var/lib/ganeti-storage/file</file_storage_dir>
<blacklisted_os type="list"/>
<enabled_hypervisors type="list">
<value>kvm</value>
</enabled_hypervisors>
<drbd_usermode_helper>/bin/true</drbd_usermode_helper>
<reserved_lvs type="list"/>
<ctime type="float">1292887189.41</ctime>
<name>gwm.osuosl.org</name>
<master_netdev>eth0</master_netdev>
<ndparams type="hash">
<oob_program type="null"/>
</ndparams>
<architecture type="list">
<value>64bit</value>
<value>x86_64</value>
</architecture>
<volume_group_name>ganeti</volume_group_name>
<beparams type="hash">
<default type="hash">
<auto_balance type="boolean">True</auto_balance>
<vcpus type="integer">2</vcpus>
<memory type="integer">512</memory>
</default>
</beparams>
</info>

```

### **/api/node**

In this endpoint exposed are the attributes and operations on the Cluster.

#### **Container: actions\_on\_node**

This container provides the actions done on the node in form of the log. It is similar in the form to the other actions\_on\_X containers in other endpoints. For more info please take a look there.

#### **Container: primary\_list and secondary\_list**

These containers provide the list of virtual machines existing on the node in primary and secondary node mode. The list is simple and includes object hostname and related link. Example:

```

<primary_list type="list">
<object type="hash">
<hostname>3429</hostname>
<resource>/api/vm/1/</resource>
</object>
<object type="hash">
<hostname>breakthis.gwm.osuosl.org</hostname>
<resource>/api/vm/2/</resource>
</object>
</primary_list>

```

#### **Container: info**

This element provides extensive information related to the node. These information are used internally in Ganeti Web Manager to render specific pages. As of level of detail used, the field contained here will be described partially only. It should be noted that the elements in the table may be nullable. The full example output is included after the table.

```
<info type="hash">
<dfree type="integer">30336</dfree>
<cnodes type="integer">1</cnodes>
<serial_no type="integer">1</serial_no>
<dtotal type="integer">60012</dtotal>
<sinst_cnt type="integer">0</sinst_cnt>
<mtime type="null"/>
<pip>140.211.15.61</pip>
<mfree type="integer">1310</mfree>
<sip>140.211.15.61</sip>
<uuid>4a0e9df5-0b59-4643-b156-c133edb035bc</uuid>
<drained type="boolean">False</drained>
<sinst_list type="list"/>
<csockets type="integer">1</csockets>
<role>M</role>
<ctotal type="integer">2</ctotal>
<offline type="boolean">False</offline>
<vm_capable type="boolean">True</vm_capable>
<pinst_cnt type="integer">15</pinst_cnt>
<mtotal type="integer">2004</mtotal>
<tags type="list"/>
<group.uuid>e318906a-40cd-4702-813b-c2185abaf8ec</group.uuid>
<master_capable type="boolean">True</master_capable>
<ctime type="null"/>
<master_candidate type="boolean">True</master_candidate>
<name>gwm1.osuosl.org</name>
<mnode type="integer">730</mnode>
<pinst_list type="list">
<value>3429</value>
<value>noinstall2.gwm.osuosl.org</value>
<value>failed</value>
<value>success</value>
<value>derpers.gwm.osuosl.org</value>
<value>testttest</value>
<value>breakthis.gwm.osuosl.org</value>
<value>foobarherpderp.gwm</value>
<value>brookjon.gwm.osuosl.org</value>
<value>orphanme</value>
<value>foobar352</value>
<value>testcdrom2.gwm.osuosl.org</value>
<value>ooga.osuosl.org</value>
<value>diskless3</value>
<value>noinstall.gwm.osuosl.org</value>
</pinst_list>
</info>
```

### **/api/job**

This endpoint exposes information related to the job execution in the system.

#### **Container: opresult**

This field contains a detailed description of error encountered during job execution. The fields included are the following:

Example:



```
<opresult type="hash">
<error_type>OpPrereqError</error_type>
<error_message>The given name (owelwjqe) does not resolve: Name or
service not known</error_message>
<error_family>resolver_error</error_family>
</opresult>
```

**Container: ops** This field contains information about the job executed. There may be many subfields included, spanned through several levels.

The following excerpts provide two typical example outputs:

```
<pre>
<ops type="list">
<object type="hash">
<hvparams type="hash">
<nic_type>paravirtual</nic_type>
<boot_order>disk</boot_order>
<root_path>/dev/vda3</root_path>
<serial_console type="boolean">False</serial_console>
<cdrom_image_path/>
<disk_type>paravirtual</disk_type>
<kernel_path/>
</hvparams>
<debug_level type="integer">0</debug_level>
<disk_template>plain</disk_template>
<name_check type="boolean">True</name_check>
<osparams type="hash"/>
<src_node type="null"/>
<source_x509_ca type="null"/>
<dry_run type="boolean">False</dry_run>
<pnode>gwml.osuosl.org</pnode>
<nics type="list">
<object type="hash">
<link>br0</link>
<mode>bridged</mode>
</object>
</nics>
<wait_for_sync type="boolean">True</wait_for_sync>
<priority type="integer">0</priority>
<start type="boolean">True</start>
<ip_check type="boolean">True</ip_check>
<source_shutdown_timeout type="integer">120</source_shutdown_timeout>
<file_storage_dir type="null"/>
<no_install type="boolean">False</no_install>
<src_path type="null"/>
<snode type="null"/>
<identify_defaults type="boolean">False</identify_defaults>
<OP_ID>OP_INSTANCE_CREATE</OP_ID>
<source_instance_name type="null"/>
<source_handshake type="null"/>
<hypervisor>kvm</hypervisor>
<force_variant type="boolean">False</force_variant>
<disks type="list">
<object type="hash">
<size type="integer">408</size>
</object>
</disks>
<instance_name>owelwjqe</instance_name>
```

```
<mode>create</mode>
<iallocator type="null"/>
<file_driver>loop</file_driver>
<os_type>image+debian-squeeze</os_type>
<beparams type="hash">
<vcpus type="integer">2</vcpus>
<memory type="integer">512</memory>
</beparams>
</object>
</ops>
</pre>

<pre>
<ops type="list">
<object type="hash">
<instance_name>brookjon.gwm.osuosl.org</instance_name>
<ignore_secondaries type="boolean">False</ignore_secondaries>
<dry_run type="boolean">False</dry_run>
<priority type="integer">0</priority>
<debug_level type="integer">0</debug_level>
<OP_ID>OP_INSTANCE_REBOOT</OP_ID>
<reboot_type>hard</reboot_type>
<shutdown_timeout type="integer">120</shutdown_timeout>
</object>
</ops>
</pre>
```

### 5.11.3 Working With Git

This page holds a list of some helpful git commands. Several of them were borrowed from Mislav's [Blog](#)

#### **git remote show origin**

Find out what branches are tracked, configured for pull, configured for push, and are 'up to date'.

#### **git fetch**

Update local references to remote branches.

#### **git log -p**

View each commits changes.

#### **git log -m -S"search text"**

Search through commits for commit messages containing the text = "search text".

#### **git reset --soft HEAD**

Accidentally added the wrong file to the index? This command will reset the current head to HEAD and keep your changes.

#### **git reset --hard HEAD**

Remove all your changes and reset the current head to HEAD.

#### **git commit --amend**

Change the commit message of the most recent commit.

#### **git add (FILENAME or DIRECTORY)**

Add a single file or directory to the index. If a directory is specified all files that are new or have been changed under that directory are added.

#### **git commit**

Open up the editor specified by `git.editor` to review changes and add a message. If no message is given, then the commit is aborted. Commit takes effect once the file is written.

#### **git commit -m "COMMIT MESSAGE"**

Commit changes with the message "COMMIT MESSAGE".

#### **git cherry -v BRANCH**

Will list the commits that are part of the current branch, and not part of BRANCH

### 5.11.4 Logos



The Ganeti Web Manager logo by Oregon State University's Open Source Lab are licensed under a Creative Commons Attribution-NoDerivs 3.0 Unported License.



### 5.11.5 Versions

Date GWM Object Log Object Permissions Sep 16th, 2010 INIT -- Oct 8th, 2010 -- INIT Nov 16th, 2010 -- 0.9 Nov 23rd, 2010 -- 1.0 Dec 10th, 2010 -- 1.1 Dec 20th, 2010 -- 1.2 Jan 26th, 2011 0.4 - 1.2 Feb 3rd, 2011 0.5 - 1.3 Feb 25th, 2011 0.5 INIT 1.3 Mar 4th, 2011 0.6 0.5 1.3.1 Mar 6th, 2011 0.6 0.5.1 1.3.1 Mar 10th, 2011 0.6.2 0.5.1 1.3.1 Jun 17th, 2011 0.7 0.6 1.4.1 Jun 17th, 2011 0.7.1 0.6 1.4.1 Jun 19th, 2011 0.7.2 0.6



# INDICES AND TABLES

- *genindex*
- *modindex*
- *search*