## Contents

1 What is ganetimgr? ........................................ 1
2 Installation ............................................... 3
3 Upgrading ............................................... 5
4 Compatibility ........................................... 7

5 Table of Contents ......................................... 9
   5.1 Installation Instructions .......................... 9
   5.2 Administrator guide .............................. 15
   5.3 Ganeti Modifications .............................. 17
   5.4 Upgrading .......................................... 19
   5.5 Interface Guide .................................. 21
ganetimgr is a Django project that serves as an administration frontend for (multiple) Ganeti clusters. It is developed as the frontend of a VPS service.
Installation

You can go through the installation at the Install ganetimgr section.
Upgrading

If running an older version, look through the Upgrade Notes before upgrading to a new one.
ganetimgr has been tested with ganeti versions 2.4-2.10. Due to the nature of the Ganeti RAPI, ganetimgr should be able to communicate with any Ganeti v.2.X cluster.

Some ganeti changes that are required for some of ganetimgr’s features to work, see Ganeti Patches
5.1 Installation Instructions

We test (and use) ganetimgr on the latest stable version of Debian. We also prefer using the Debian packages for Django and any python dependencies instead of using pip and virtualenv. That way we don’t have to worry about any of the upstream projects breaking anything and we have quicker/easier security updates.

This guide documents how to install ganetimgr with the following software:

- Debian Stable, the base OS
- gunicorn with gevent, it runs the Django project
- Nginx, the web server that serves the static content and proxies to gunicorn
- Mysql, the database backend
- Redis, as Django’s caching backend. Stores session info and caches data
- Beanstalkd, used by worker.py

Any feedback on how to install under different circumstances is welcome.

5.1.1 Install packages

Update and install the required packages:

```
apt-get install git nginx mysql-server gunicorn python-gevent redis-server beanstalkd
apt-get install python-mysqldb python-django python-redis python-django-south python-django-registration python-paramiko ...
```

Fabric script

We have created a fabric script in order to set up and deploy ganetimgr. It is included under “contrib/fabric/”. One can use it by running:

```
fab deploy:tag='v1.6' -H ganetimgr.example.com -u user
```

You will need to have fabric installed though.

This script will connect to the specified server and try to set up ganetimgr under “/srv/ganetimgr” which will be a symlink to the actual directory.

In general it performs the following steps:
• stop redis, beanstalk, touch “/srv/maintenance.on”
• git clone, git archive under “/tmp” and move to “/srv/ganetimgr<year><month><day><hour><minute>”
• check if there is an old installation under /srv/ganetimgr and get all the dist files in order to compare them with the newer version
• create a backup of the database
• If no differences have been found between the two versions of ganetimgr, the old configuration files (whatever has also a dist file) will be copied to the new installation.
• “/srv/ganetimgr” will be point to the new installation
• management commands (migrate, collectstatic) will be run
• fabric will ask your permission to remove old installations
• restart nginx, gunicorn, redis, beanstalk, rm “maintenance.on”
• in case something goes wrong it will try to make a rollback
• in case no older installations exist or the dist files, it will ask you to log in the server and edit the settings, while waiting for your input.

5.1.2 Beanstalkd

Edit /etc/default/beanstalkd and uncomment the following line:

START=yes

and then start the daemon with:

service beanstalkd start

5.1.3 Database Setup

Create a mysql user for ganetimgr.

Note: This is only defined on the project’s settings.py so use a strong random password.

Login to the mysql server:

mysql -u root -p

Create database and user:

mysql> CREATE DATABASE ganetimgr CHARACTER SET utf8;
mysql> CREATE USER 'ganetimgr'@'localhost' IDENTIFIED BY <PASSWORD>;
mysql> GRANT ALL PRIVILEGES ON ganetimgr.* TO 'ganetimgr';
mysql> flush privileges;

5.1.4 Requirements.txt

A requirements.txt file is included in order to help you install the required python dependencies. You can do that by running:
**5.1.5 Pre-Setup**

Get the source and checkout to latest stable:

```
mkdir /srv/
mkdir /var/log/ganetimgr
cd /srv/
git clone https://code.grnet.gr/git/ganetimgr
cd ganetimgr
git checkout stable
```

Create the required `settings.py` files for the example files:

```
cd ganetimgr
cp settings.py.dist settings.py
```

**5.1.6 Settings.py**

There are a lot of parts of ganetimgr that are customizable. Most of them are changed from the `settings.py` file. Below are explanations for most of the settings:

- Fill the default `DATABASES` dictionary with the credentials and info about the database you created before.
- Set `CACHES` to the backend you want to use, take a look at: [https://docs.djangoproject.com/en/1.4/topics/cache/](https://docs.djangoproject.com/en/1.4/topics/cache/)
- Set `STATIC_URL` to the relative URL where Django expects the static resources (e.g. ‘/static/’)
- The `BRANDING` dictionary allows you to customize stuff like logos and social profiles. You can create your own logo starting with the static/branding/logo.* files.
- `FEED_URL` is an RSS feed that is displayed in the user login page.
- `SHOW_ADMINISTRATIVE_FORM` toggles the admin info panel for the instance application form.
- `SHOW_ORGANIZATION_FORM` does the same for the Organization dropdown menu.
- You can use use an analytics service (Piwik, Google Analytics) by editing `templates/analytics.html` and adding the JS code that is generated for you by the service. This is sourced from all the project’s pages.
- `AUDIT_ENTRIES_LAST_X_DAYS` (not required, default is None) determines if an audit entry will be shown depending on the date it was created. It’s only applied for the admin and is used in order to prevent ganetimgr from being slow. ‘0’ is forever.
- `GANETI_TAG_PREFIX` (Default is ‘ganetimgr’) sets the prefix ganetimgr will use in order to handle tags in instances. eg in order to define an owner it sets ‘ganeti_tag_prefix:users:testuser’ as a tag in an instance owned by `testuser`, assuming the GANETI_TAG_PREFIX is equal to ‘ganeti_tag_prefix’.

**External Services**

You can use Google re-CAPTCHA during registration to avoid spam accounts. Generate a key pair from [here](here) and fill these settings:

```
RECAPTCHA_PUBLIC_KEY = '<key>'
RECAPTCHA_PRIVATE_KEY = '<key>'
```
You can use LDAP as an authentication backend. The package python-ldap needs to be installed. You need to uncomment the LDAPBackend entry in the AUTHENTICATION_BACKENDS and uncomment and edit accordingly the AUTH_LDAP_* variables. LDAP authentication works simultaneously with normal account auth.

SERVER_MONITORING_URL is used to link ganeti node information with ganetimgr. This URL with the hostname appended is used to create a link for every node. We use servermon for node information.

If you deploy a Jira installation then you can append a tab on the left of ganetimgr web interface via an issue collection plugin that can be setup via:

| HELPDESK_INTEGRATION_JAVASCRIPT_URL |
| HELPDESK_INTEGRATION_JAVASCRIPT_PARAMS |

If you want to embed collectd statistics in ganetimgr instance view fill the:

COLLECTD_URL

If COLLECTD_URL is not null, then the graphs section can be used in order to show graphs for each instance. One can define a NODATA_IMAGE if the default is not good enough.

There is a VM isolation feature for VMs that are suspect of having been compromised. The admin or the user can define a subnet from which the VM will remain accessible for further investigation. The next settings limit the subnet width (v4 and v6 accordingly) that is allowed to be used:

| WHITELIST_IP_MAX_SUBNET_V4 |
| WHITELIST_IP_MAX_SUBNET_V6 |

### Instance Images

There are two ways to define available images:

From the OPERATING_SYSTEMS dictionary (e.g. for a Debian Wheezy image):

```
OPERATING_SYSTEMS = {
    "debian-wheezy": {
        "description": "Debian Wheezy 64 bit",
        "provider": "image+default",
        "osparams": {
            "img_id": "debian-wheezy",
            "img_format": "tarball",
        },
        "ssh_key_param": "img_ssh_key_url",
    },
}
```

As of v.1.5.0 there is an autodiscovery mechanism for the images. You just have to insert the following settings variable:

```
OPERATING_SYSTEMS_URLS = ['http://repo.noc.grnet.gr/images/', 'http://example.com/images/']
```

All the given HTTP URLs from OPERATING_SYSTEMS_URLS will be searched for images. This discovers all images found on these URLs and makes them available for usage.

The description of the images can be automatically fetched from the contents of a .dsc file with the same name as the image. For example, if an image named debian-wheezy-x86_64.tar.gz, ganetimgr will look for a debian-wheezy-x86_64.tar.gz.dsc file in the same directory and read its contents (e.g. Debian Wheezy) and display it accordingly.

You also need to set OPERATING_SYSTEMS_PROVIDER and OPERATING_SYSTEMS_SSH_KEY_PARAM:
OPERATING_SYSTEMS_PROVIDER = 'image+default'
OPERATING_SYSTEMS_SSH_KEY_PARAM = 'img_ssh_key_url'

GanntiMgr will look for available images both from both sources. None of the above settings is required.
There is also an autodiscovery mechanism for snf images, by setting snf-image url in settings.py as such:

SNF_OPERATING_SYSTEMS_URLS = ['http://repo.noc.grnet.gr/images/snf-image/']

The process is identical with that above.

**FLATPAGES**

GanntiMgr provides 3 flatpages - Service Info, Terms of Service and FAQ. Flatpages can be enabled or disabled via the:

**FLATPAGES**

dictionary.

We provide 6 flatpages placeholders (3 flatpages x 2 languages - English and Greek) for the flatpages mentioned. By invoking the command:

```bash
python manage.py loaddata flatpages.json
```

the flatpages placeholders are inserted in the database and become available for editing via the admin interface (Flat Pages).

**VNC**

We provide 2 VNC options for the users.

- For the Java VNC applet to work, vncauthproxy must be running on the server. Setup instructions can be found [here](#).
- For setup instructions for the Websocker VNC applet, check [here](#).

There are three relevant options here:

- WEBSOCK_VNC_ENABLED enables/disabled the options for the noVNC console.
- NOVNC_PROXY defines the host vncauthproxy is running (default is ‘localhost:8888’).
- NOVNC_USE_TLS specifies the use or not of TLS in the connection. For cert info look at [here](#).

### 5.1.7 Install

**Attention:** When running the syncdb command that follows DO NOT create a superuser yet!

Run the following commands to create the database entries:

```bash
python manage.py syncdb --noinput
python manage.py migrate
```

and create the superuser manually:
python manage.py createsuperuser

To get the admin interface files, invoke collectstatic:

python manage.py collectstatic

Run the watcher.py:

./watcher.py

### 5.1.8 Gunicorn Setup

Create the gunicorn configuration file (/etc/gunicorn.d/ganetimgr):

```python
CONFIG = {
    'mode': 'django',
    'working_dir': '/srv/ganetimgr',
    'user': 'www-data',
    'group': 'www-data',
    'args': ('--bind=127.0.0.1:8088',
             '--workers=2',
             '--worker-class=egg:gunicorn#gevent',
             '--timeout=30',
             '--log-file=/var/log/ganetimgr/ganetimgr.log'),
}
```

**Attention:** A logrotate script is recommended from keeping the logfile from getting too big.

Restart the service:

```bash
service gunicorn restart
```

### 5.1.9 Web Server Setup

Create (or edit) an nginx vhost with the following:

```nginx
location /static {
    root /srv/www/ganetimgr;
}

location / {
    proxy_pass http://127.0.0.1:8088;
}
```

Restart nginx:

```bash
service nginx restart
```

### 5.1.10 End

This installation is finished. If you visit your webserver’s address you should see the ganetimgr welcome page.

Now it’s time to go through the Admin guide to setup your clusters.
5.2 Administrator guide

5.2.1 Cluster Setup

Ganetimgr needs a set of RAPI credentials to communicate with a cluster. These need to be created manually. The next steps need to be repeated for every cluster you wish to administer with ganetimgr.

Create (or edit if it already exists) the /var/lib/ganeti/rapi/users file on every node at the cluster like this:

```
<user> <pass> write
```

You can replace `write` with `read` so that ganetimgr can only view the cluster resources, but most of the ganetimgr functionality will be disabled.

Login to the ganetimgr platform and go to the admin interface. You can do so from the sidebar:

- My Profile
- FAQ
- Admin
- Language: English
- Admin

Select the cluster option from the Ganeti section.

From here you can manage the cluster pool. Normally this list is empty now. Select the “Add” cluster option:

An explanation about some of the settings:

- **Hostname** is the fully qualified domain name of the cluster
- **Slug** is a friendly name for the cluster
- **Port** is the port the RAPI daemon listens to on the master node. Unless you manually changed it this should be 5080
• **Username/Password** are the credentials created earlier for the cluster.

• **Fast instance creation** is an option to submit instance creation requests through the admin interface instead of going through the normal application procedure.

• **Cluster uses gnt-network** is a soon to be deprecated option about network options for new instances. If you use routed networks (though gnt-network) this should be on.

### 5.2.2 Network Setup

Ganetimgr autodiscovers any network available through gnt-network on the cluster during the instance creation. You can also hardcode any other networks (e.g. bridged vlans) from the admin interface.

- **Link** is the name of the network device found on the cluster.
- **Mode** is the network mode for the interface can be routed or bridged.
- **Groups** ties the network to a specific user group. When a user from that group submits an application this network is autoselected.

### 5.2.3 Using Oauth2

We have integrated oauth2 support. API +++ The endpoints the api provides are:

`/instances/list/`

which fetches all the instances of the current user. Optionally the api user can pass a get parameter named ‘tag’ which filters all instances by the specific tag.

E.g:

`/instances/list/?access_token=CUST0MT0k3N&tag=ganetimgr:filterby:thistag`

will fetch us only the instances tagged with “ganetimgr:filterby:thistag”

And:

`/user/details/`

which lists user details (username, email, password, id)

### 5.2.4 Configuring oauth2

In order to add oauth2 you have to install:
• django-cors-headers==1.1.0
• django-oauth-toolkit==0.5.0

And insert in installed apps:
• oauth2_provider
• corsheaders

5.2.5 Usage

• Hit /o/applications and create a new one.
• Create a new application with implicit authorization grant type
• Set the redirect urls to the desired ones
• by hitting /o/authorize?redirect_uri=<redirect_uri>&client_id=<client_id>&response_type=token&scope=read you get a response with the token.
• by adding the access_token in the request to /instances/lists/?access_token=<token> you get a list of all the user instances.

5.2.6 Translations

Ganetimgr uses Django’s translation system. We have created translations in Greek. In order to create translations to another language you have to follow these instructions.

5.3 Ganeti Modifications

For the time being, ganetimgr requires the use of some patched packages. We are working on merging those changes upstream so that it works on tha vanilla software. This software needs to be install on the ganet clusters.

5.3.1 Repository

We provide Debian packages for all the different software listed here. To get them you need to use our public repository.

Add our repository to your sources.list:

```bash
echo 'deb http://repo.noc.grnet.gr/ wheezy main backports' > /etc/apt/sources.list.d/grnet.list
```

Add our gpg key to apt's keyring:

```bash
wget -O - http://repo.noc.grnet.gr/grnet.gpg.key | apt-key add -
```

And refresh the package list to discover the new packages:

```bash
apt-get update
```
5.3.2 ganeti-instance-image

This is a forked version of the [ganeti-instance-image](https://code.osuosl.org/projects/ganeti-image) OS provider written by UOSL:

```
apt-get install ganeti-instance-image
```

It uses the Ganeti OS API v20 to specify runtime osparams so we can specify the instance os during instance creation. It also injects the ssh key of the user inside the instance. The code from which the package is build can be found [here](https://github.com/grnet/ganeti-instance-image). You can find a sample Debian Wheezy image [here](http://repo.noc.grnet.gr/debian-wheezy-x86_64.tgz).

5.3.3 Ganeti

If you want to use the `boot from url` feature of ganetimgr, you will need our ganeti package:

```
apt-get install ganeti ganeti-htools
```

Our package version has `+grnet` appended to the version string.

5.3.4 Java VNC applet

The package `vncauthproxy` is required to run on the host ganetimgr is running for the Java VNC applet to work. You can install it with:

```
apt-get install vncauthproxy
```

An example config that needs to be placed on `/etc/default/vncauthproxy`:

```text
DAEMON_OPTS="-p11000 -P15000 -s /var/run/vncauthproxy/vncproxy.sock"
CHUID="nobody:www-data"
```

11000-15000 is the (hardcoded, it seems) port range that ganeti uses for vnc binding, so you will need to open your firewall on the nodes for these ports.

5.3.5 WebSocketsVNC

To enable WebSocket support you will need to install [VNCAuthProxy](https://github.com/osuosl/twisted_vncauthproxy) following this [guide](https://code.osuosl.org/projects/ganeti-webmgr/wiki/VNC#VNC-AuthProxy) from OSL.

You will also need at least the following packages: python-twisted, python-openssl

Start your twisted-vncauthproxy with:

```
twistd --pidfile=/tmp/proxy.pid -n vncap -c tcp:8888:interface=0.0.0.0
```

Make sure your setup fulfills all the required [firewall rules](https://code.osuosl.org/projects/ganeti-webmgr/wiki/VNC#Firewall-Rules)

Modern browsers block `ws://` connections initiated from HTTPS websites, so if you want to open `wss://` connections and encrypt your noVNC sessions you need to enable TLS noVNC. You will also need signed a certificate for the ‘example.domain.com’ host and place it under twisted-vncauthproxy/keys directory. The paths are currently hardcoded so one needs to install these 2 files (keep the filenames):
IPv6 Warning

Since twisted (at least until version 12) does not support IPv6, make sure the host running twisted-vncauthproxy does not advertise any AAAA records, else your clients won’t be able to connect.

5.4 Upgrading

This article describes the actions that are needed when upgrading from a previous version of Gametimgr.

5.4.1 Migrating to v.1.6

- **Perform south migration**: python manage.py migrate
- **Collect static files**: python manage.py collectstatic
- Update settings.py to settings.py.dist

Requirements

Ganetimgr v.1.6 comes with a requirements.txt file, which lists all requirements. Make sure all of these requirements have been installed on your system.

5.4.2 Migrating to v.1.5.3

- **Perform south migration**: python manage.py migrate
- **Collect static files**: python manage.py collectstatic
- Update settings.py to settings.py.dist. A new context processor is deployed
- Update urls.py to urls.py.dist. The graph url has been updated
- New dependencies: python-bs4, python-requests and python-markdown. All packaged in Debian.
- We recommend changing gunicorn logging to the system logging path (i.e. /var/log/) and not /tmp.
- Also create a logrotate script for the logfile.
- After upgrade is done make sure to restart the watcher process

5.4.3 Migrating to v.1.4.1

Bugfix/Feature Enhancements release

settings.py: - Copy the FLATPAGES dict from settings.py.dist to allow handling of flatpages
5.4.4 Migrating to v.1.4.0

Debian wheezy/Django 1.4 compatibility

settings.py:

- If migrating from a squeeze installation pay attention to Django 1.4 changes as depicted in settings.py file, especially the introduction of the staticfiles django app
- Set the FEED_URL to an RSS news feed if desired
- Setup WebSockets VNCAuthProxy if desired
- If WebSockets NoVNC is setup, set the WEBSOCK_VNC_ENABLED to True and the NOVNC_PROXY and NOVNC_USE_TLS to match your setup
- Update the BRANDING dict to match your needs
- Perform south migration

5.4.5 Migrating to v.1.3.0

- Set the WHITELIST_IP_MAX_SUBNET_V4/V6 to desired max whitelist IP subnets in settings.py
- Perform south migration

5.4.6 Migrating to v.1.2.3

- Make sure to include HELPDESK_INTEGRATION_JAVASCRIPT_PARAMS in settings.py.

If you deploy Jira and want to set custom javascript parameters, set

```python
HELPDESK_INTEGRATION_JAVASCRIPT_PARAMS = { 'key': 'value' # eg. 'customfield_23123': '1stline' } ' In any other case set
HELPDESK_INTEGRATION_JAVASCRIPT_PARAMS = False '```

5.4.7 Migrating to v.1.2.2

- Make sure to restart watcher.py

5.4.8 Migrating to >= v1.2

- Make sure to:
  - Set the RAPI_TIMEOUT in settings.py (see .dist)
  - Set the NODATA_IMAGE path in settings.py.dist
  - Update urls.py to urls.py.dist
  - Copy templates/analytics.html.dist to templates/analytics.html.
5.4.9 Migrating to v1.0:

- install python-ipaddr lib
- update settings.py and urls.py with latest changes from dist files

Run:  manage.py migrate

If your web server is nginx, consider placing:

`proxy_set_header X-Forwarded-Host <hostname>;
USE_X_FORWARDED_HOST = True`

in your settings.py. The above ensures that i18n operates properly when switching between languages.

5.5 Interface Guide

This document showcases some of ganetimgr’s features from the user’s and administrator’s perspective.

ganetimgr is the frontend to a VPS service, albeit without a pricing component. A regular user has access to administer only the vms assigned to them. An administrator can view and administer every resource. There is a permissions system that can be configured to provide separate access levels. A user can’t create a vm by themselves, they need to submit an instance application which needs to be approved by an administrator for the vm to be created.

5.5.1 User View

Main view

The Home view presents a table with every vm owned by the user with some basic information for quick sorting. There are also actions shortcuts for basic interaction.

Statistics

There is a statistics view that shows graphs and data about the user (stats, accumulated resources, instance applications, etc...)
Profile

This is the profile settings page. The user can change the profile full name, the password or the registered email address. Also SSH key management. The SSH keys defined here are auto-injected to the vms when an instance application is approved by the admins.

New Application

The instance application form. The user requests the resources needed and contact information.

History

User can view a timeline of the events for all his vms.
5.5.2 Permissions

The are three extra permissions for non-administrative users:

**Can view all instances**

When a user has this permission, they can view all instances.

**Can Lock**

Combined with the above permission, they can lock an instance.

**Can Isolate**

Combined with the ‘can view all instances’ permission, they can isolate an instance.

5.5.3 Admin view

The administrator has access to quite a few more views.

**Main view**

The admin main view lists all the vms from all the clusters ganetimgr knows about. The interaction with the vm is the same as a regular users.

**Cluster Info**

Real-time lookup of cluster-wide properties of a specific cluster.
Nodes view

List of all available hardware nodes with important resources for easy sorting. Ability to filter based on Ganeti node status (master, master-candidate, regular, drained, offline).

Batch Graphs

List all instances Cpu and network graphs, per cluster or per node. You can collapse and expand the graphs per instance and one can select graphs between a specific time period.

Jobs

Real-time view of the cluster(s) job queue. Ability to sort based on job status (running, success, failed, waiting) and per cluster.
Audit Log

Logging of vm-related user initiated events such as Instance shutdown/reboot/startup/

Applications

Statistics

The statistics page shows data from all the clusters that ganetimgr knows about. There is a view for administering applications (approving/denying the pending ones, viewing past ones), viewing interesting users (users and vm associations, inactive users, etc...).
Instance Details

There is an instance details page, which shows the basic information of an instance to a user. A user has permission to make some actions to the instances he or the groups he is in owns.

- Shutdown Instance
- Reboot Instance
- **Can have console access (VNC/NoVNC):** A vnc console (java applet) and a novnc console can be given to the user in order to use the vm remotely through ganetimgr. There is also a text area (NoVNC) in case one needs to paste some commands and easily send them to the virtual machine
- Rename Instance
- Reinstall Instance
- Destroy

Fatal actions require email verification.

Notifications

An administrator can directly send mail to vm owners. The Recipient field is dynamic. It can lookup a usernames email address dynamically and also fetch all the emails associated with a vm or even all the vms the reside on a cluster. We have added the variable `instances` in the context of the message (where applicable, eg clusters) in order to show
the names of a users affected instances. In this section there is also the archive where one can see all the previous notifications sent by the admins.