
gnadi-cookbook Documentation

Release 0.0.1

azediv

February 07, 2016

1	IaaS	3
1.1	CoreOS on Gandi IaaS vm	3
1.2	Howto Luks Debian and Gandi	6
2	PaaS	11
2.1	virtualenv for Django CMS	11
2.2	auto deploy from GitHub to Simple Hosting	12
3	Mail	15
3.1	Gandi Mail and Gmail	15
3.2	Gandi Mail on iPhone	16
4	Support Channel	19
4.1	Offical Support	19
4.2	Community Support	19

Welcome to Gandi cookbook's documentation!

Here's a list of community driven Howto's

- Installation guides

1.1 CoreOS on Gandi IaaS vm

Installation script for CoreOS on Gandi server with gandi.cli

Original idea “fork” from jmbarbier idea to create a cluster of coreos on Gandi vm :
<https://gist.github.com/jmbarbier/ab06cf23735845a0167a>

Requirements

- gandi.cli with credits on iaas account
- nano text editor (vi(m) fan, sed script to replace nano by vi(m))

Howto

Download on local computer and make it executable :

```
$ wget https://raw.githubusercontent.com/azediv/gnadi-coreos/master/install-core.sh && chmod +x
```

Edit install-core.sh to change hostname, disk name, user :

hostname and disk name must be short for now, 7 or 8 max.

```
$ $EDITOR install-core.sh
```

Run :

```
$ ./install-core.sh
```

Automatic script

install-core.sh is used locally with gandi.cli to create vm and install coreos with /gandi/config json file

json config is download and used to create cloud-config.yml

cloud-config.yml is scp to temp vm before install

details in script, need to edit : VM_USER VM and DC

TODO : gandi/config file provisionned in swap during first boot and use in cloud-config

Step by step process

Resume

Creation of a Debian vm with a 10GB data disk.

Retrieve config of vm from /gandi/config.

Creation of cloud-config.yml file.

Installation of CoreOS from Debian vm on data disk.

Stop vm and detach of Debian disk.

Define kernel as raw on CoreOS disk

Attach CoreOS disk as system disk to vm.

Boot and enjoy

Details:

\$HOSTNAME define hostname of coreos vm. (\$VM in script)

\$VM_USER define username for coreos vm.

\$DC define datacenter for vm and disk

Creation of Debian vm (512Mo at least, 256Mo isn't enough to install packages) :

```
$ gandi vm create --datacenter $DC --memory 512 --cores 1 --ip-version 4 --login $VM_USER --hos
```

Creation of data disk (target of CoreOS install) and attachment to VM :

```
$ gandi disk create --name core_sys --size 10G --datacenter $DC -vm $HOSTNAME
```

SSH to Debian :

```
$ gandi vm ssh $HOSTNAME
```

Unmont data disk before installation :

```
# umount /dev/sdc
```

Installation of wget

```
# apt-get update && apt-get install -y wget
```

Download of coreos install script :

```
# wget https://raw.githubusercontent.com/coreos/init/master/bin/coreos-install
```

Define as executable :

```
# chmod +x coreos-install
```

Grab username, hashed password, sshkey of vm from /gandi/config json file :

```
VM_USER = cat /gandi/config | grep -Po '(?<="user": ")[^"]*' | head -1
PASS : Hashed password can be used = cat /gandi/config | grep -Po '(?<="password": ")[^"]*'
SSH = cat /gandi/config | grep -Po '(?<="ssh_key": ")[^"]*'
```

Grab network config of vm from /gandi/config json file :

```
HOSTNAME = cat /gandi/config | grep -Po '(?<="vm_hostname": ")[^"]*'
IP = cat /gandi/config | grep -Po '(?<="pna_address": ")[^"]*' | head -1
ROUTE = cat /gandi/config | grep -Po '(?<="pbn_gateway": ")[^"]*' | head -1
DNS = gandi vm ssh $VM 'cat /etc/resolv.conf' | awk '{ print $2 }' | grep -E -o "[0-9]{1,3}[\.]"
```


Creation of cloud-config.yml file for CoreOS installation.

Contains static network, user, sshkey, units, etc...

Help : <https://coreos.com/os/docs/latest/cloud-config.html#users>

```
# nano cloud-config.yml

#cloud-config
coreos:
  units:
    - name: systemd-networkd.service
      command: stop
    - name: 00-eth0.network
      runtime: true
      content: |
        [Match]
        Name=eth0

        [Network]
        DNS=$DNS
        Address=$IP
        Gateway=$ROUTE
    - name: systemd-networkd.service
      command: start
hostname: $HOSTNAME
users:
  - name: $VM_USER
    passwd: $PASS
    groups:
      - sudo
      - docker

  ssh_authorized_keys:
    - $SSH
```

Starting CoreOS installation with cloud-config.yml

```
# ./coreos-install -d /dev/sdc -C alpha -c cloud-config.yml
```

When installation is successfull, logout of Debian :

```
# exit
```

Stop the Debian vm :

```
$ gandi vm stop $HOSTNAME
```

Detach of system disk of Debian :

```
$ gandi disk detach sys_$HOSTNAME
```

Detach of data disk, CoreOS :

```
$ gandi disk detach core_sys
```

Update kernel to raw of data disk of CoreOS :

```
$ gandi disk update --kernel raw core_sys
```

Attachment as system disk (-p 0) to vm :

```
$ gandi disk attach -p 0 core_sys $HOSTNAME
```

First start of CoreOS on Gandi vm ! :

```
$ gandi vm start $HOSTNAME
```

Remove previous SSH fingerprint for IP :

```
$ ssh-keygen -f "/home/$USER/.ssh/known_hosts" -R $IP
```

Login to CoreOS :

```
$ gandi vm ssh --login $VM_USER $HOSTNAME
```

Ping us ! :

```
CoreOS alpha (870.2.0)
$VM_USER@$HOSTNAME ~ $ ping gandi.net
PING gandi.net (217.70.184.1) 56(84) bytes of data.
64 bytes from website.vip.gandi.net (217.70.184.1): icmp_seq=1 ttl=60 time=104 ms
64 bytes from website.vip.gandi.net (217.70.184.1): icmp_seq=2 ttl=60 time=104 ms
^C
--- gandi.net ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1000ms
rtt min/avg/max/mdev = 104.245/104.318/104.392/0.331 ms
$VM_USER@$HOSTNAME ~ $
```

1.2 Howto Luks Debian and Gandi

This Howto describe a process to create a raw vm of Debian with an encrypted root partition on Gandi servers. Step 1, 2 and 5 are common to other distribution. As we are looking for a way to improve new distribution deployment, this process might be declined for other use case.

1. creation of a vm with all tools to start minimal installation of “any” distribution on Gandi IaaS VPS/servers
2. creation of target disk that will be use as target disk and is needed for “any” distribution.
3. specific to Debian installation with LVM encrypted in kvm vm with raw format
4. specific to encrypted volume and Grub bootloader configuration
5. configure Gandi VPS with new disk and is for “any” distribution

1 : Gandi vm creation, tools install and raw vm creation.

We will proceed to a Debian vm creation and tools required to create a qemu/kvm raw vm inside of this Gandi vm. (sexy isn't it ?) Then we will download desired boot medium for an install. A netinstall iso of latest debian version in this example.

1.1 : Creation of temp vm:

```
local $ gandi vm create --hostname vdl --datacenter LU --ip-version 4 --login user --password --
local $ gandi vm ssh vdl
```

1.2 : Installation of requirements on temp vm:

```
vm # apt-get update && apt-get upgrade -y && apt-get install kvm qemu-kvm libvirt-bin virtinst b
```

1.3 : Creation of working directory:

```
vm # mkdir /home/vm
vm # cd /home/vm
```

1.4 : Download of debian netinstall iso:

```
vm # wget http://cdimage.debian.org/debian-cd/8.2.0/amd64/iso-cd/debian-8.2.0-amd64-netinst.iso
```

1.5 : Creation of raw image for installation of raw vm

```
vm # qemu-img create -f raw deb.raw 3G
```

1.6 : Configuration of network on temp vm for raw vm

Edit gandi config file :

```
vm # nano/vi(m) /etc/default/gandi
```

and change :

```
"CONFIG_NETWORK=1"
```

into :

```
"CONFIG_NETWORK=0"
```

then configure bridge network :

```
vm # virsh iface-bridge eth0 br0
```

Here you'll be disconnected so :

```
local $ gandi vm reboot vdl
local $ gandi vm ssh vdl
```

1.7 : Creation of raw vm with iso as boot disk

```
vm # cd /home/vm
vm # virt-install --connect qemu:///system -n vm -r 2048 --vcpus=1 --disk path=/home/vm/deb.raw
```

1.8 : Connexion to raw vm, 2 solutions :

1.8.1 : Install virt-manager on local computer

```
local $ sudo apt-get install virt-manager
local $ virt-manager
```

Choose in Menu : File => Add a connexion

Hypervisor : QEMU/KVM tick "Connexion to a remote host" Method : SSH Username : root Hostname : IP of temp VM Select Automatic connexion and click "Connect"

In virt-manager, right-click on new connexion and click "Connect" You will then be able to select raw vm and click "Open" in menu.

1.8.2 : Creation of a tunnel to initiate vnc connection from local computer

```
local $ ssh user@VDL_IP -L 5900:127.0.0.1:5900
local $ vncviewer 127.0.0.1
```

select "Install" in viewer and you'll be disconnected so connect again

```
local $ vncviewer 127.0.0.1
```

1.9 : Manual config of network during installation with vm public network settings

from temp vm :

```
IP : cat /gandi/config | grep -Po '(?<="pna_address": ")[^"]*' | head -1
GATEWAY : cat /gandi/config | grep -Po '(?<="pbn_gateway": ")[^"]*' | head -1
DNS : cat /etc/resolv.conf | awk '{ print $2}' | grep -E -o "([0-9]{1,3}[\.]){3}[0-9]{1,3}" | head -1
HOSTNAME : cat /gandi/config | grep -Po '(?<="vm_hostname": ")[^"]*' | head -1
```

2 : Target disk creation

This section is for any distribution

Creation of target disk

```
local : $ gandi disk create --size 4G --name vdluks --vm vdl
vm : # umount /dev/sdc
```

3 : Debian install with LVM encrypted

This section is specific to Debian install and should be adapted for another distribution

3.1 : Partitionning

select assisted all in one disk with encrypted LVM

all in one partition (recommend for beginner)

3.2 : Networking

important note

after apt mirror selection, when it fail to find network mirrors : choose “Go Back” and continue without network mirrors with “Yes”

irl : grab a coffee

3.3 : Bootloader

at the end of installation, choose to install Grub “Yes” and on /dev/sda

When install is over, kvm vm will stop due to default settings “no autostart”

4 : Configuraiton of Grub bootloader

Copy to disk and mount with cryptsetup to reinstall Grub

4.1 : Copy image to disk and mount partitions to chroot :

Copy of raw image with dd to target disk :

```
vm # dd if=deb.raw of=/dev/sdc
```

Open with cryptsetup root partition and name it vm :

```
vm # cryptsetup luksOpen /dev/sdc5 vm
```

Create mount directory :

```
vm # mkdir /srv/vm
```

Scan all pysical volume :

```
vm # vgscan
vm # vgchange -ay
```

Scan all logical volume :

```
vm # lvscan
```

Note logical volume path to mount it.

Mount {root,boot,dev,proc,sys} partitions :

```
vm # mount /dev/deb-vdl/root /srv/vm
vm # mount /dev/sdc1 /srv/vm/boot/
vm # mount -o bind /dev/ /srv/vm/dev/
vm # mount -o bind /proc/ /srv/vm/proc/
vm # mount -o bind /sys /srv/vm/sys/
```

Chroot inside new debian vm :

```
vm # chroot /srv/vm
```

4.2 : Edit grub :

```
chroot # vi(m)/nano /etc/default/grub
```

and change :

```
# GRUB_CMDLINE_LINUX=""
```

to :

```
# GRUB_CMDLINE_LINUX="console=ttyS0"
```

4.3 : Reinstall Grub :

```
chroot # grub-install /dev/sdc
chroot # update-grub
chroot # exit

vm # exit
```

5 : The end

5.1 : Detach, update, attach and start vm :

```
local $ gandi vm stop vdl
local $ gandi disk detach sys_vdl -f
local $ gandi disk detach vdluks -f
local $ gandi disk update vdluks --kernel raw
local $ gandi disk attach vdluks -p 0 vdl -f
local $ gandi vm start vdl
```

5.2 : Decrypt Partition with console :

```
local $ gandi vm console vdl

Asking for console, please wait
Connected

Grabbing terminal
Ok
```

```
...  
Please unlock disk sda5_crypt:
```

Unlock !!!

Login as root for next step.

vm has network, please ping us !

```
raw_vm # ping gandi.net
```

Optimization :

Edit sources :

```
raw_vm # # vi(m)/nano /etc/apt/sources.list
```

And add

```
deb ftp://ftp.fr.debian.org/debian/ jessie main contrib non-free  
deb ftp://ftp.fr.debian.org/debian/ jessie-updates main contrib non-free
```

Update

```
raw_vm # apt-get update
```

Install gandi-hosting-vm2 :

```
raw_vm # wget http://mirrors.gandi.net/gandi/debian/pool/gandi-hosting-vm2_2.6_all.deb  
raw_vm # dpkg -i gandi-hosting-vm2_2.6_all.deb  
raw_vm # apt-get install -f
```

Install OpenSSH server :

```
raw_vm # apt-get install openssh-server  
local $ ssh-keygen -f "/home/$USER/.ssh/known_hosts" -R VM_IP  
local $ gandi vm ssh --login user vdl
```

Ready !

2.1 virtualenv for Django CMS

If you haven't already, create a virtual environment:

```
$ virtualenv env $ source env/bin/activate
```

Or activate it:

```
$ source env/bin/activate
```

Django CMS and Dependencies

Run the following command to Install Django CMS using *pip*:

```
$ pip install.djangocms-installer
```

This provides you with a new command, *djangocms*.

Creating a New Project

Create your project's folder and *cd* into it:

```
$ mkdir tutorial-project $ cd tutorial-project
```

For the purposes of this tutorial, it's recommended that you answer the installer's questions as follows - where our suggestions differ from the default, they're highlighted below:

```
$ sudo djangocms -p . mysite
```

- Database configuration (in URL format) [default sqlite://localhost/project.db]:
- django CMS version (choices: 2.4, 3.0, stable, develop) [default stable]: stable
- Django version (choices: 1.4, 1.5, 1.6, 1.7, stable) [default stable]: **1.6**
- Activate Django I18N / L10N setting (choices: yes, no) [default yes]: yes
- Install and configure reversion support (choices: yes, no) [default yes]: yes
- Languages to enable. Option can be provided multiple times, or as a comma separated list. Only language codes supported by Django can be used here: en,de
- Optional default time zone [default America/Chicago]:
- Activate Django timezone support (choices: yes, no) [default yes]: yes
- Activate CMS permission management (choices: yes, no) [default yes]: yes
- Use Twitter Bootstrap Theme (choices: yes, no) [default no]: **yes**

- Use custom template set [default no]: no
- Load a starting page with examples after installation. Choose “no” if you use a custom template set. (choices: yes, no) [default no]: yes

```
## create wsgi.py
```

```
$ nano wsgi.py
```

Add the following contents:

```
import sys import os import os.path

sys.path.insert(0, os.path.abspath(os.path.join(os.path.dirname(__file__), 'mysite')))
os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'mysite.settings') from django.core.wsgi
import get_wsgi_application application = get_wsgi_application()
```

```
## create requirements.txt
```

```
$ pip freeze > requirements.txt
```

```
## create git repository, push, and deploy
```

```
$ git init $ git remote add origin ssh+git://146585@git.dc1.gpaas.net/default.git $ git add . $ git com-
mit -m "testing django" $ git push origin master $ ssh 146585@git.dc1.gpaas.net 'deploy default.git'
```

References:

1. <http://docs.django-cms.org/en/support-3.0.x/introduction/install.html>
2. <http://wiki.gandi.net/en/simple/instance/python>

Author : soulshake Github : <https://github.com/soulshake> Doc Type : Community

2.2 auto deploy from GitHub to Simple Hosting

This tutorial will be useful if you:

- have a website with static content
- want to host the source code for your website on GitHub or BitBucket
- want the site to be hosted on a Gandi Simple Hosting instance, and
- you want your site to be updated automatically any time changed are pushed to the repo.

```
## Requirements
```

- A Gandi Simple Hosting (PHP/MySQL) (or create one below)
- A GitHub or BitBucket account
- A public git repo with static web content

We’re using the [Gandi CLI](<http://cli.gandi.net>) for this tutorial, but you can also perform these steps within your Gandi account.

```
## Warning
```

Simple Hosting is not designed for use of git from within the instance. It only partially works and even the things we’re doing below are very fragile. Do not make any changes to files directly on your instance, or you will get stuck and have to start over.

```
### Create a PHP/MySQL Simple Hosting Instance
```

```
$ gandi paas create --name squirrels --type phpmysql --vhosts squirrels.gandi.xyz
```


Connect to the instance via SSH

The following command will activate the console and automatically open a connection to it.

```
$ gandi paas console squirrels
```

Clone the repo to the instance

Clone the repo into the *htdocs/* directory:

```
$ cd /srv/data/web/vhosts/squirrels.gandi.xyz/htdocs $ rm index.html # Because the directory has to be empty first $ git clone https://github.com/soulshake/squirrels.git . # Note the dot at the end $ ls -la
```

In my case, the output of *ls -la* looks like this:

```
hosting-user@squirrels:~/web/vhosts/squirrels.gandi.xyz/htdocs$ ls -la total 9 drwxrwxr-x 4 hosting-user hosting-user 7 Dec 14 22:52 . drwxrwxr-x 3 root hosting-user 3 Dec 14 07:36 .. drwxr-xr-x 8 hosting-user hosting-user 15 Dec 14 22:52 .git -rw-r--r- 1 hosting-user hosting-user 79 Dec 14 22:48 README.md drwxr-xr-x 2 hosting-user hosting-user 3 Dec 14 22:48 images -rw-r--r- 1 hosting-user hosting-user 101 Dec 14 22:52 index.html -rwxr-xr-x 1 hosting-user hosting-user 318 Dec 14 22:49 pull.php
```

See *pull.php* above? It's there in the sample repo we're using for this tutorial, but you'll need to make one. Create a PHP file (we'll call ours *pull.php*) with the following contents:

```
` <?php `git pull`; // This will execute the `git pull` command on your instance header("Cache-Control: max-age=1"); // Lower the cache while we're here so the changes take effect faster echo "hello!"; // So you can confirm the file is in the right place by browsing to the URL ?> `
```

Make sure the file is executable:

```
$ chmod +x pull.php
```

Check that everything is working correctly by browsing to the URL corresponding to *pull.php*. In our case, it looks like this:

```
http://squirrels.gandi.xyz/pull.php
```

![[screenshot](http://ss.squirrel.li/image/3N0v3n1I2S1o "pull.php")]

Next, go to the settings page of your GitHub repo, then choose “Webhooks & Services” from the menu on the left.

Click “Add webhook” and paste the path to your .php file in the “Payload URL” field:

![[screenshot](http://ss.squirrel.li/image/1U1G3l133O3O)]

It may take a few minutes to start working due to the Varnish cache. Thereafter, note that changes will still take about a minute to be visible unless you purge the cache manually.

Now, every time your repo is updated, your website will be, too!

Author : soulshake Github : <https://github.com/soulshake> Doc Type : Community

3.1 Gandi Mail and Gmail

Configuring Gandi Mail for Gmail

This page contains instructions for setting up Gmail (aka Mail Fetcher) to send and receive from your Gandi mailboxes.

Note: This will allow you to import your GandiMail to your current “@gmail.com” inbox – not to be confused with [Google Apps a.k.a. Gmail for your domain](<http://wiki.gandi.net/en/domains:management:googleapps>).

Step 1: Receiving (POP/IMAP)

- In your Gmail account, go to settings (click the gear icon to the upper right of your inbox, then click “settings”)
- Click the “Accounts” tab at the top.
- Under “Check mail from other accounts (using POP3)” click “Add a POP3 mail account you own”
- **Email address:** Enter the email address you want to be able to check from Gmail, i.e. *admin@example.com*, then click “next step”
- **Username:** Enter your **email address** again, i.e. *admin@example.com* (*admin* by itself won’t work).
- **Password:** Enter the password you chose when you created the *admin@* mailbox for *example.com*. (This must be a mailbox and not a forwarding address).
- **POP Server:** Change it to *mail.gandi.net*
- **Port:** 110 for POP or 995 for POP SSL

If successful, you’ll be given the opportunity on the following screen to be able to send mail from the same address.

Step 2: Sending (SMTP)

- **Name:** Enter the name you want to appear in the “From:” field of your emails
- **Email address:** Enter the same email address as in Step 1. Make your other selections and click Next.
- **Send mail through your SMTP server?** You can send through Gandi SMTP, if you want: - **SMTP Server:** *mail.gandi.net* (this will probably pre-populate incorrectly) - **Port:** 25, 465 (with SSL) or 587 (try one or the other) - **Username:** Same email address as before, ie *admin@example.com*

If successful, the following screen will request you either click the confirmation link in the email or enter a confirmation code to be able to send mail from the above address.

Possible errors

“Authentication failed”

```
` Server denied POP3 access for the given username and password.
Server returned error: "Authentication failed." `
```

- You may have only entered your username, whereas you need to enter the whole email address.
- You may be entering the wrong password.
- You may be trying to add a forwarding address instead of a mailbox.

“You already have the maximum number of accounts allowed.”

You can only set up 5 external POP accounts in Gmail. Consider using a [forwarding address](<http://wiki.gandi.net/en:mail:email-mailboxes-and-forwarding-addresses#how-can-i-set-up-e-mail-forwarding>) instead.

“We were unable to locate the other domain.”

```
` There was a problem connecting to mail.yourdomain.tld Server
returned error: "We were unable to locate the other domain.
Please contact your other provider." `
```

Replace *mail.yourdomain.tld* with *mail.gandi.net*.

“SSL protocol error.”

```
` There was a problem connecting to mail.gandi.net Server returned
error: "SSL protocol error. Please try disabling SSL, or contact
your other provider to verify the correct port settings." `
```

You may have selected “Always use a secure connection (SSL) when retrieving mail” but did not select the correct port for using SSL.

“Unable to process the account info.”

```
` Server denied POP3 access for the given username and password.
Server returned error: "Unable to process the account info." `
```

This error can occur when the wrong port is selected for a given configuration.

“There was a problem connecting to mail.gandi.net”

```
` There was a problem connecting to mail.gandi.net Server returned
error: "Missing +OK response upon connecting to the server: * OK
[CAPABILITY IMAP4rev1 LITERAL+ SASL-IR LOGIN-REFERRALS ID ENABLE
AUTH=PLAIN] Dovecot ready." `
```

You may be trying to use an IMAP port. Gmail does not support IMAP import.

See also

- [Google/Gmail/Mail Fetcher support: Get mail from other accounts](<http://support.google.com/mail/bin/answer.py?hl=en&answer=21288>)

Author : soulshake Github : <https://github.com/soulshake> Doc Type : Community

3.2 Gandi Mail on iPhone

Setting up your iPhone for Gandi mail just takes a minute. Here’s what to do:

1. Go to **Settings**, then

2. **Mail, Contacts, Calendars**, then
3. Click **Add Account...**
4. Choose **Other**
5. Select **Add Mail Account**” under **Mail**

Here, you will just enter the information as requested. I am providing some example values below, you will need to replace them with your own:

- Name: **‘Jon Doe‘**
 - Address: **‘jon@example.com‘**
 - Password: **‘yourpasswd‘**
 - Description: will be filled in automatically, but can be anything
6. Click **“Next“**
 7. Choose IMAP or POP.

Instructions for POP and IMAP ### Incoming Mail Server

- Host Name: *mail.gandi.net*
- User Name: **‘joe@example.com‘**
- Password: **‘yourpasswd‘**

Outgoing Mail Server

- Host Name: *mail.gandi.net*
- Username: **‘joe@example.com‘**
- Password: **‘yourpasswd‘**

Click **“Save”**.

It will now check the configuration and will display the message “Verifying” at the top. This process can take about 30 seconds. When complete, it will bring you back to the screen that shows your email accounts.

Troubleshooting

Not all providers will let you send emails from a third-party SMTP server. If you encounter problems sending mail (eg. a message saying that the SMTP server mail.gandi.net is not responding), just use the SMTP server of your local internet provider.

You can verify the standard settings (port, etc.) at <http://wiki.gandi.net/en/mail/standard-settings>

Author : soulshake Github : <https://github.com/soulshake> Doc Type : Community

- Official support channels

Support Channel

4.1 Official Support

4.1.1 Support ticket

Contact us via mail

4.1.2 Live Chat

While logged in on [Gandi](#) click on ‘?’ icon on the right.

4.2 Community Support

4.2.1 IRC

- Server : [irc.freenode.net](#)
- Channel : [#gandi](#)

4.2.2 Github

Open issue on any Gandi project on [Github](#)