



Ghiro Documentation

Release 0.3-dev

September 15, 2016

1	Introduction	3
1.1	Why Ghiro?	3
1.2	Use Case	3
1.3	Supported image types	3
1.4	Architecture	4
1.5	Download Ghiro	5
1.6	Verifying Signatures	6
2	Techniques	9
2.1	Techniques	9
2.2	ELA (Error Level Analysis)	9
2.3	GPS Localization	9
2.4	Hash digest generation	9
2.5	Perceptual Hash	10
2.6	Hash list matching	10
2.7	Metadata information	10
2.8	MIME information	11
2.9	Nude detection	11
2.10	Preview thumbnail	12
2.11	Preview thumbnail consistency	12
2.12	Signature engine	12
2.13	Strings extraction	12
3	Setup	13
3.1	Setup Ghiro	13
3.2	Requirements	13
3.3	Getting started	14
3.4	Configuration	15
3.5	Running Ghiro as service	18
4	Usage	21
4.1	Usage	21
4.2	Administration	22
4.3	Virtual Appliance	25
4.4	API Usage	39
4.5	FAQ	41
5	Notes	43

5.1	Keep in touch with us	43
5.2	Contribute, bugs and feature requests	43
5.3	Disclaimer	44
5.4	License	44

Ghiro is an *Open Source* software for digital photo and digital image forensics. The forensic analysis is fully automated, report data can be searched or aggregated in different perspectives. Ghiro is designed to assist you and your team in the process of analyzing a massive amount of images, it could become an essential tool in your forensic lab.

This documentation will explain what Ghiro is, which techniques it uses for forensic analysis, how to set up it and how to use it.

Introduction

1.1 Why Ghiro?

Sometime forensic investigators need to process digital images as evidence. There are some tools around, most of them are little scripts, otherwise it is difficult to deal with forensic analysis with many images involved. Digital images contain tons of information, it is a tedious work to manually extract all of them, Ghiro extracts these information from provided images and display them in a nicely formatted report. Dealing with tons of images has never been so easy, Ghiro is designed to scale to support gigs of images. All tasks are totally automated, you have just to upload your images and let Ghiro do the work. Understandable reports and great search capabilities allows you to find a needle in a haystack. Ghiro is a multi user environment, different permissions can be assigned to each user. Cases allow you to group image analyses by topic, you can choose which user allow to see your case with a permission schema. Every team in your forensic lab could work in their own cases with privileges separation.

1.2 Use Case

Ghiro can be used in many scenarios, forensic investigators could use it on daily basis in their analysis lab but also people interested to undercover secrets hidden in images could benefit. Some use case examples are the following:

- If you need to extract all data and metadata hidden in an image in a fully automated way
- If you need to analyze a lot of images and you have not much time to read the report for all them
- If you need to search a bunch of images for some metadata
- If you need to geolocate a bunch of images and see them in a map
- If you have an hash list of “special” images and you want to search for them

Anyway Ghiro is designed to be used in many other scenarios, the imagination is the only limit.

1.3 Supported image types

The following file type are supported:

- Windows bitmap .bmp
- Raw Canon .cr2
- Raw Canon .crw
- Encapsulated PostScript .eps

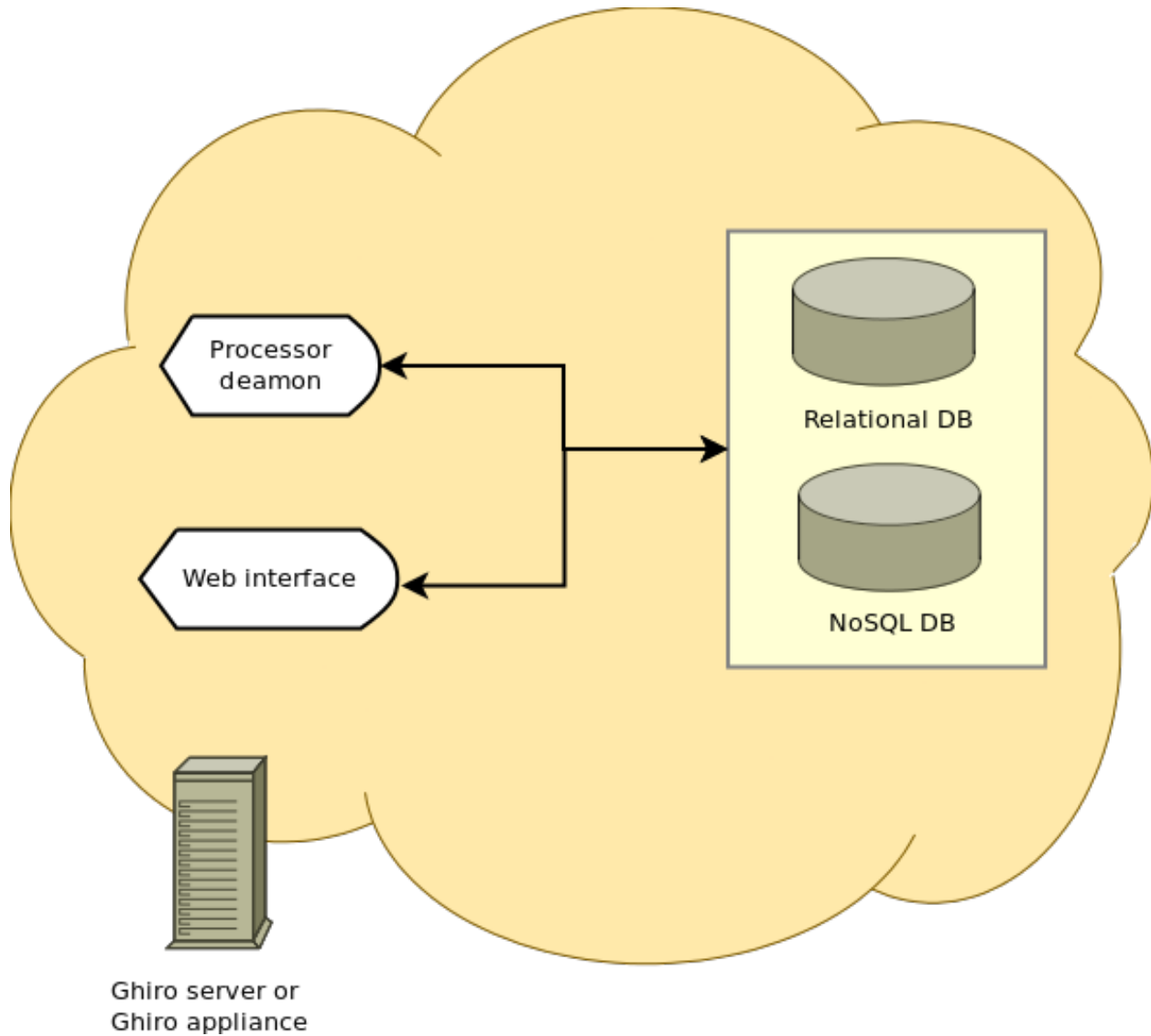
- Graphics Interchange Format .gif
- JPEG File Interchange Format .jpg or .jpeg
- Raw Minolta .mrw
- Raw Olympus .orf
- Portable Network Graphics .png
- Raw Photoshop .psd
- Raw Fujifilm .raf
- Raw Panasonic .rw2
- Raw TARGA .tga
- Tagged Image File Format .tiff

1.4 Architecture

Ghiro is composed by the following components:

- The web interface: to interact with all features, this is the component used by users to work with Ghiro
- The processor daemon: it fetches waiting tasks from the queue, process and analyze images
- The SQL database: it stores relational data, you can choose between MySQL, PostgreSQL and SQLite3
- The MongoDB database: it stores analysis data

Following the architecture in a simple schema:



1.5 Download Ghiro

Ghiro can be downloaded from the [official website](#), where the stable and packaged releases are distributed. Stable package is available in both .zip and .tar.gz format. The package above is strongly suggested for all users. Some people need to keep updated with Ghiro's changes, they can download (git clone) from our [official GitHub page](#). There are two different releases available.

1.5.1 Development Branch

The development branch is where the next Ghiro's release is developed. You can download Ghiro from here if you need to keep it always at the cutting edge or if you want to hack on Ghiro. You can download it with the following command:

```
git clone https://github.com/ghirensics/ghiro.git
```

Warning: While being more updated, the development branch should be considered an *under development*. Therefore its stability is not guaranteed.

1.6 Verifying Signatures

Every release published by the Ghiro Developers is digitally signed by the [Ghiro Master Signing key](#) or by one of the developers (each such key is signed by the [Ghiro Master Signing key](#)).

The first step is to import the Ghiro Master Signing public key, you can download it from a keyserver with this command:

```
$ gpg --keyserver pool.sks-keyservers.net --recv-keys 0xafda03a581c21ee9
```

You can add it directly from Ghiro website too:

```
$ gpg --fetch-keys http://getghiro.org/keys/ghiro_master_signing_key.asc
```

The fingerprint of [Ghiro Master Signing key](#) is published here (for additional security):

```
9DD9 3A61 39A4 A72D 2467 378D AFDA 03A5 81C2 1EE9
```

Now you can download the signature for Ghiro package or appliance from Ghiro website, you can verify it with the following:

```
$ gpg --verify ghiro-0.2.zip.sig
gpg: Signature made Sun Mar 15 17:55:51 2015 CET using RSA key ID 81C21EE9
gpg: Good signature from "Ghiro Master Signing key (Ghiro Master Signing key)" [ultimate]
```

If you get an output like this one, the package you got is good and you can trust it, if you get a different output you are facing a security risk, you should contact Ghiro's developers and never use the downloaded package.

It is also a good, although optional, practice to set its trust level to "ultimate", so that it can be used to automatically verify all the keys signed by the Ghiro developers:

```
$ gpg --edit-key 0x81C21EE9
```

Now trust the key, and set trust to ultimate level with:

```
gpg> trust pub 4096R/81C21EE9
```

As example the full output follows:

```
$ gpg --keyserver pool.sks-keyservers.net --recv-keys 0xafda03a581c21ee9
gpg: requesting key 81C21EE9 from hkps server pool.sks-keyservers.net
gpg: /home/jekil/.gnupg/trustdb.gpg: trustdb created
gpg: key 81C21EE9: public key "Ghiro Master Signing key (Ghiro Master Signing key)" imported
gpg: no ultimately trusted keys found
gpg: Total number processed: 1
gpg:          imported: 1 (RSA: 1)

$ gpg --edit-key 0x81C21EE9
gpg (GnuPG) 1.4.16; Copyright (C) 2013 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

pub 4096R/81C21EE9  created: 2015-03-15  expires: 2021-03-15  usage: SC
```

```
trust: unknown      validity: unknown
sub 4096R/E51F5BBD  created: 2015-03-15 expires: 2021-03-15 usage: E
[ unknown] (1). Ghiro Master Signing key (Ghiro Master Signing key)

gpg> trust pub 4096R/81C21EE9
pub 4096R/81C21EE9  created: 2015-03-15 expires: 2021-03-15 usage: SC
trust: unknown      validity: unknown
sub 4096R/E51F5BBD  created: 2015-03-15 expires: 2021-03-15 usage: E
[ unknown] (1). Ghiro Master Signing key (Ghiro Master Signing key)

Please decide how far you trust this user to correctly verify other users' keys
(by looking at passports, checking fingerprints from different sources, etc.)

  1 = I don't know or won't say
  2 = I do NOT trust
  3 = I trust marginally
  4 = I trust fully
  5 = I trust ultimately
  m = back to the main menu

Your decision? 5
Do you really want to set this key to ultimate trust? (y/N) y

pub 4096R/81C21EE9  created: 2015-03-15 expires: 2021-03-15 usage: SC
trust: ultimate      validity: unknown
sub 4096R/E51F5BBD  created: 2015-03-15 expires: 2021-03-15 usage: E
[ unknown] (1). Ghiro Master Signing key (Ghiro Master Signing key)
Please note that the shown key validity is not necessarily correct
unless you restart the program.

gpg> quit
```

Techniques

2.1 Techniques

Several techniques are used to extract all data and metadata hidden in digital images. They are briefly described in this chapter.

2.2 ELA (Error Level Analysis)

Error Level Analysis (ELA) is a technique aimed to detect if an image is edited or not. It can be applied to compressed images only, i.e. JPEG or PNG. The main idea is that an image in his original form has unique levels of compression. The analyzed image is resaved and differences in compression levels are calculated, if differences are detected a probability of edits is high. Ghirelli calculates error levels and detects differences between them.

2.3 GPS Localization

Embedded in the image metadata sometimes there is a geotag, a bit of GPS data providing the longitude and latitude of where the photo was taken. Geotagging is useful when a device such as an iPhone, Android smartphone or digital camera stores your location or geographical information, such as your GPS coordinates, within a photo. A geotagged photograph is a photograph which is associated with a geographical location by geotagging. Geotags are useful in helping people find a wide variety of location-specific information. For example, one can find images taken near a given location by entering latitude and longitude coordinates into a suitable image search engine. The geotag inside image metadata is read and the position where the photo was taken is displayed on a map.

2.4 Hash digest generation

Most common hash are calculated for the image, to create an unique identifier of it. The calculated hashes are:

- CRC32
- MD5
- SHA1
- SHA224
- SHA256

- SHA384
- SHA512

2.5 Perceptual Hash

Regular hashing techniques usually don't work well with images. Images are different from usual binary files, like executables, if you change only a bit of an image, it could remain pretty much the same. Perceptual hash or image hash are algorithms designed to work with images despite of the little differences.

2.6 Hash list matching

Suppose you are searching for an image and you have only the hash. You can provide a list of hashes and all images matching are reported.

2.7 Metadata information

Metadata may be written into a digital photo file that will identify who owns it, copyright and contact information, what camera created the file, along with exposure information and descriptive information such as keywords about the photo, making the file searchable on the computer and/or the Internet. Some metadata are written by the camera and some is input by the photographer and/or software after downloading to a computer. Metadata are divided in several categories depending on the standard they come from. The following categories are extracted and analyzed:

- **EXIF metadata extraction**
 - Standard Exif tags
 - Canon MakerNote tags
 - Fujifilm MakerNote tags
 - Minolta MakerNote tags
 - Nikon MakerNote tags
 - Olympus MakerNote tags
 - Panasonic MakerNote tags
 - Pentax MakerNote tags
 - Samsung MakerNote tags
 - Sigma/Foveon MakerNote tags
 - Sony MakerNote tags
- **IPTC metadata extraction**
 - IPTC datasets
- **XMP metadata extraction**
 - Dublin Core schema (dc)
 - XMP Basic schema (xmp)
 - XMP Rights Management schema (xmpRights)

- XMP Media Management schema (xmpMM)
- XMP Basic Job Ticket schema (xmpBJ)
- XMP Paged-Text schema (xmpTPg)
- XMP Dynamic Media schema (xmpDM)
- Adobe PDF schema (pdf)
- Photoshop schema (photoshop)
- Camera Raw schema (crs)
- Exif schema for TIFF Properties (tiff)
- Exif schema for Exif-specific Properties (exif)
- Exif schema for Additional Exif Properties (aux)
- IPTC Core schema (Iptc4xmpCore)
- IPTC Extension schema (Iptc4xmpExt)
- PLUS License Data Format schema (plus)
- digiKam Photo Management schema (digiKam)
- KDE Image Program Interface schema (kipi)
- Microsoft Photo schema (MicrosoftPhoto)
- iView Media Pro schema (mediapro)
- Microsoft Expression Media schema (expressionmedia)
- Microsoft Photo 1.2 schema (MP)
- Microsoft Photo RegionInfo schema (MPRI)
- Microsoft Photo Region schema (MPReg)
- Metadata Working Group Regions schema (mwg-rs)

2.8 MIME information

Multipurpose Internet Mail Extensions (MIME) is a standard to describe content type of a file, MIME is detected using magic number inside the image. Magic numbers implement strongly typed data and are a form of in-band signaling to the controlling program that reads the data type(s) at program run-time. Many files have such constants that identify the contained data. Detecting such constants in files is a simple and effective way of distinguishing between many file formats and can yield further run-time information. The image MIME type is detected to know the image type you are dealing with, in both contacted (example: image/jpeg) and extended form.

2.9 Nude detection

Nude detector is based on NudePy library and should detect images with nude figures due to their skin tone.

2.10 Preview thumbnail

Most digital camera and phones write a preview, called thumbnail, in image metadata. The thumbnails and data related to them are extracted from image metadata and stored for review.

2.11 Preview thumbnail consistency

This check compares an image with the embedded thumbnail to spot differences. Sometimes when a photo is edited, if the image editing software does not support embedded thumbnail automatic editing, the original image is edited but the thumbnail not. A simple comparison between the original image and the thumbnail could detect image alteration or show the original image.

2.12 Signature engine

Signature provides evidence about most critical data to highlight focal points and common exposures. Signature engine to highlight common exposure on over 120 signatures.

2.13 Strings extraction

All text strings contained in the analyzed image are extracted, like in the unix strings tool. The more interesting (i.e. URLs) are highlighted.

3.1 Setup Ghiro

Ghiro is supposed to run on a *GNU/Linux* native system. For the purpose of this documentation, we choose **latest Ubuntu LTS Server** as reference system for the commands examples, although Ghira works on any GNU/Linux distribution. Probably Ghira could work on other systems like MacOSX but this is not tested and out of scope of this documentation.

3.2 Requirements

Ghiro has the following requirements:

- MongoDB: you need to run a MongoDB database (at least release 2.0)
- Python: that's how we roll (it is required Python 2.7, although Ghira is written to work also in Python 3 some third party libraries aren't)
- Python-magic: for MIME extraction
- Python 2.x bindings for gobject-introspection libraries, required by Gexiv2
- Gexiv2: for metadata extraction (at least release 0.6.1)
- Pillow (Python Imaging library - PIL fork): for image manipulation
- Python-dateutil: for datetime manipulation
- Pymongo: driver for MongoDB (at least release 2.5)
- Django: for web interface (at least release 1.5, suggested django 1.6.x)
- Chardet: for text encoding detection
- Pdftk: used for PDF report generation (at least release 0.4)
- Wkhtmltopdf: used by pdftk
- Requests: used for HTTP requests
- NudePy: used for nude detection
- ImageHash: to calculate perceptual image hash

If you choose MySQL or PostgreSQL as database you have to install their additional drivers.

Ghiro web application is tested and working on the following browsers:

- Internet Explorer 11
- Mozilla Firefox starting from 35
- Google Chrome starting from 39
- Opera starting from 26
- Safari 8 and 9
- IOS 7 for Ipad and Iphone

3.3 Getting started

3.3.1 Download and extract

Download Ghiro as explained in this documentation, if you download the stable package extract it. Enter in the Ghiro folder.

3.3.2 Preparing

If you don't have already it, install MongoDB with the following command (run as root or with sudo):

```
apt-get install mongodb
```

Ghiro works with SQLite although it is strongly suggested to use MySQL or PostgreSQL as database. If SQLite is used, Ghiro will automatically decrease processing parallelism to one because SQLite does not support concurrent operations. Optionally, as an example, you can install MySQL with the following command (run as root or with sudo):

```
apt-get install mysql-server
```

Install required libraries with the following commands (run as root or with sudo):

```
apt-get install python-pip build-essential python-dev python-gi
apt-get install libgexiv2-2 gexiv2-0.10 wkhtmltopdf
apt-get install libtiff5-dev libjpeg-dev zlib1g-dev libfreetype6-dev
apt-get install liblcms2-dev libwebp-dev tcl8.5-dev tk8.5-dev python-tk
```

The wkhtmltopdf tool used for PDF report generation needs a X server running, if you don't have one just install XFVB and configure wkhtmltopdf to use it with:

```
apt-get install xvfb
printf '#!/bin/bash\nxvfb-run --server-args="-screen 0, 1024x768x24" /usr/bin/wkhtmltopdf $*' > /usr/
chmod a+x /usr/bin/wkhtmltopdf.sh
ln -s /usr/bin/wkhtmltopdf.sh /usr/local/bin/wkhtmltopdf
```

Install updated libraries via pip with the following commands (run as root or with sudo):

```
pip install -r requirements.txt
```

3.3.3 Preparing

The default databases are SQLite3 and MongoDB (you need to have it listening on localhost). If you need to change this see the configuration chapter below.

First of all you need to create an empty database with the following command (inside Ghiro's root):

```
python manage.py migrate
```

Create a superuser for administration, you should provide an username and a password, use the following command (inside Ghiro's root):

```
python manage.py createsuperuser
```

3.3.4 Running

To start the web interface run the following command (inside Ghiro's root):

```
python manage.py runserver
```

A web server running Ghiro will be available on <http://127.0.0.1:8000/> If you need to listen expose Ghiro to all addresses or change the port (in this example is 9000) run the following command (inside Ghiro's root):

```
python manage.py runserver 0.0.0.0:9000
```

To start processing images you have to start the processing daemon, run the following command (inside Ghiro's root):

```
python manage.py process
```

3.4 Configuration

Ghiro works pretty well with default options, which are SQLite3 as relational database and use MongoDB installed and listening on local host. If you want to change any setting the configuration file is located in *ghiro/local_settings.py*. The default settings will fit all common user needs.

Following is the default *ghiro/local_settings.py* file:

```
LOCAL_SETTINGS = True
from .settings import *

DATABASES = {
    'default': {
        # Engine type. Ends with 'postgresql_psycopg2', 'mysql', 'sqlite3' or 'oracle'.
        'ENGINE': 'django.db.backends.sqlite3',
        # Database name or path to database file if using sqlite3.
        'NAME': 'db.sqlite3',
        # Credentials. The following settings are not used with sqlite3.
        'USER': '',
        'PASSWORD': '',
        # Empty for localhost through domain sockets or '127.0.0.1' for localhost through TCP.
        'HOST': '',
        # Set to empty string for default port.
        'PORT': '',
        # Set timeout (avoids SQLite "database is locked" errors).
        'timeout': 300,
        # The lifetime of a database persistent connection, in seconds.
        'CONN_MAX_AGE': 60,
    }
}

# MySQL tuning.
#DATABASE_OPTIONS = {
```

```
# "init_command": "SET storage_engine=INNODB",
#}

# Mongo database settings
MONGO_URI = "mongodb://localhost/"
MONGO_DB = "ghirodb"

# Max uploaded image size (in bytes).
# Default is 150MB.
MAX_FILE_UPLOAD = 157286400

# Allowed file types.
ALLOWED_EXT = ['image/bmp', 'image/x-canon-cr2', 'image/jpeg', 'image/png',
               'image/x-canon-crw', 'image/x-eps', 'image/x-nikon-nef',
               'application/postscript', 'image/gif', 'image/x-minolta-mrw',
               'image/x-olympus-orf', 'image/x-photoshop', 'image/x-fuji-raf',
               'image/x-panasonic-raw2', 'image/x-tga', 'image/tiff', 'image/pjpeg',
               'image/x-x3f', 'image/x-portable-pixmap']

# Override default secret key stored in secret_key.py
# Make this unique, and don't share it with anybody.
# SECRET_KEY = "YOUR_RANDOM_KEY"

# Language code for this installation. All choices can be found here:
# http://www.i18nguy.com/unicode/language-identifiers.html
LANGUAGE_CODE = "en-us"

ADMINS = (
    # ("Your Name", "your_email@example.com"),
)

MANAGERS = ADMINS

# Allow verbose debug error message in case of application fault.
# It's strongly suggested to set it to False if you are serving the
# web application from a web server front-end (i.e. Apache).
DEBUG = True

# A list of strings representing the host/domain names that this Django site
# can serve.
# Values in this list can be fully qualified names (e.g. 'www.example.com').
# When DEBUG is True or when running tests, host validation is disabled; any
# host will be accepted. Thus it's usually only necessary to set it in production.
ALLOWED_HOSTS = ["*"]

# Automatically checks once a day for updates.
# Set it to False to disable update check.
UPDATE_CHECK = True

# Auto upload is used to upload and analyze files from a directory, monitoring
# it for changes.
# It is usually used to upload images via a shared folder or FTP.
# It should be an absolute path.
# Example: "/home/ghiro_share"
AUTO_UPLOAD_DIR = None
# Delete a file after upload and submission.
# The default behaviour is True.
# WARNING: It is not suggested to set it to False, because you will re-submit images
```

```
# each startup.
AUTO_UPLOAD_DEL_ORIGINAL = True
# Clean up AUTO_UPLOAD_DIR when startup.
# The default behaviour is True.
# WARNING: It is not suggested to set it to False, because you will re-submit images
# each startup.
AUTO_UPLOAD_STARTUP_CLEANUP = True

# Auditing.
# Logs all user actions.
AUDITING_ENABLED = True

# Log directory. Here is where Ghiro puts all logs.
LOG_DIR = os.path.join(PROJECT_DIR, "log")
# File name used for image processor log.
LOG_PROCESSING_NAME = "processing.log"
# Processor log maximum size.
LOG_PROCESSING_SIZE = 1024*1024*16 # 16 megabytes
# How many copies of processor log keep while rotating logs.
LOG_PROCESSING_NUM = 3 # keep 3 copies
# File name used for audit log.
LOG_AUDIT_NAME = "audit.log"
# Audit log maximum size.
LOG_AUDIT_SIZE = 1024*1024*16 # 16 megabytes
# How many copies of audit log keep while rotating logs.
LOG_AUDIT_NUM = 3 # keep 3 copies

# Enable JSON export to file for analysis results.
# This will create a JSON file for each analysis.
JSON_EXPORT = False
```

If you change the configuration after the first setup, before editing this file you have to stop both Ghiro's web interface and processing daemon, you may restart them after the edit.

If you changed any setting related to the database configuration you have to re-build your database with the command (inside Ghiro's root):

```
python manage.py syncdb
```

3.4.1 Logging

Ghiro provides several types of logging divided in the following categories:

- Audit logging: tracks all users actions for audit purposes.
- Processing logging: logs all steps of image analysis, this helps debugging stuff.

You can change the behavior of logging editing *ghiro/local_settings.py*:

- LOG_DIR: log directory. Here is where Ghiro puts all logs.

Audit log

The audit log contains all users actions (i.e. case creation, image analysis actions) to keep track of user activity. By default it is located in *audit.log* in the Ghiro log directory. You can change the behavior of this log editing *ghiro/local_settings.py*:

- LOG_AUDIT_NAME: audit log file name.

- LOG_AUDIT_SIZE: audit log maximum size.
- LOG_AUDIT_NUM: how many copies of audit log keep while rotating logs.

Processing log

The processing log contains image analysis logs, it is of great help when debugging Ghiro or trying to understand what happen under the hood. By default it is located in *processing.log* in the Ghiro log directory. You can change the behavior of this log editing *ghiro/local_settings.py*:

- LOG_PROCESSING_NAME: processing log file name.
- LOG_PROCESSING_SIZE: processing log maximum size.
- LOG_PROCESSING_NUM: how many copies of processing log keep while rotating logs.

3.5 Running Ghiro as service

If you want to run Ghiro as a service you have to get rid of Django web server and run Ghiro inside a web server (i.e. Apache).

3.5.1 Database

We do not suggest SQLite3 for production environment, please go for MySQL or PostgreSQL. In this example we are going to show you how to configure Ghiro with MySQL.

Setup MySQL and Python drivers with the following command (run as root or with sudo):

```
apt-get install mysql-server python-mysqldb
```

Go through the wizard and set MySQL password. Configure Ghiro to use MySQL as explained in configuration paragraph.

3.5.2 Apache as a front-end

Now we are going to configure Apache as a front end for Ghiro's django application.

Setup Apache and mod_wsgi with the following command (run as root or with sudo):

```
apt-get install apache2 libapache2-mod-wsgi
```

An example of virtual host configuration is the following (Ghiro is extracted in /var/www/ghiro/ in this example):

```
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    WSGIProcessGroup ghiro
    WSGIDaemonProcess ghiro processes=5 threads=10 user=nobody group=nogroup python-path=/var/www/ghiro
    WSGIScriptAlias / /var/www/ghiro/ghiro/wsgi.py
    Alias /static/ /var/www/ghiro/static/
    <Location "/static/">
        Options -Indexes
    </Location>

    ErrorLog ${APACHE_LOG_DIR}/error.log
```

```
# Possible values include: debug, info, notice, warn, error, crit,  
# alert, emerg.  
LogLevel warn  
  
CustomLog ${APACHE_LOG_DIR}/access.log combined  
</VirtualHost>
```

Restart apache. Now the web application is listening on port 80/tcp, just put the IP address in your browser.

3.5.3 Run the processor with upstart

You can automatically run the processor with upstart.

Create the file `ghiro.conf` in `/etc/init/` with the following content:

```
description      "Ghiro"  
  
start on started mysql  
stop on shutdown  
script  
    exec start-stop-daemon --start -d /var/www/ghiro \  
        --exec /usr/bin/python manage.py process  
end script
```

To stop the processor use the following command (run as root or with `sudo`):

```
service ghiro stop
```

To start the processor use the following command (run as root or with `sudo`):

```
service ghiro start
```

Usage

4.1 Usage

Ghiro's web application is composed by several parts to organize information and analysis data.

4.1.1 Dashboard

This is the summary of all Ghiro activities, here you can figure what is going on, which are the last cases and analysis, and take a look to analysis trend.

4.1.2 Cases

Image analysis are grouped in cases. Different users and permissions can be assigned to each case. You can upload images via an upload in many ways:

- You can add some images using your browser using the “Add image” method
- You can add an image from an URL using the “Add URL” method
- You can add images from a folder in Ghiro's server giving his path with the “Add folder” method
- You can add images from a folder in Ghiro's server via command line

4.1.3 Images

Here you can see all image analysis in the system (all images you have permission to see).

4.1.4 Search

You can search for several image properties or for image location. Search is available also inside a case, to have the search scope restricted to the case.

4.1.5 Hashes

Sometimes hash lists are used to search and match a special kind of images you already have an hash. A text file with a list, one per line, of image's hashes can be loaded in Ghiro with a label and a description. If Ghiro detects an image with an hash matching your hash list, it will trigger a signature and warn you.

4.1.6 Administration

In Ghiro's administration panel you can:

- Administer all Ghiro's users
- See user's activity log
- Check for required dependencies

4.2 Administration

Some hints about Ghiro administration.

4.2.1 Auto Upload

Auto Upload is a feature to automatically upload images written in a default directory; you can share the default directory over FTP, Samba (Windows shared folder) or other file sharing technologies, writing a file here you can simply launch a Ghiro analysis copying a file in a shared folder.

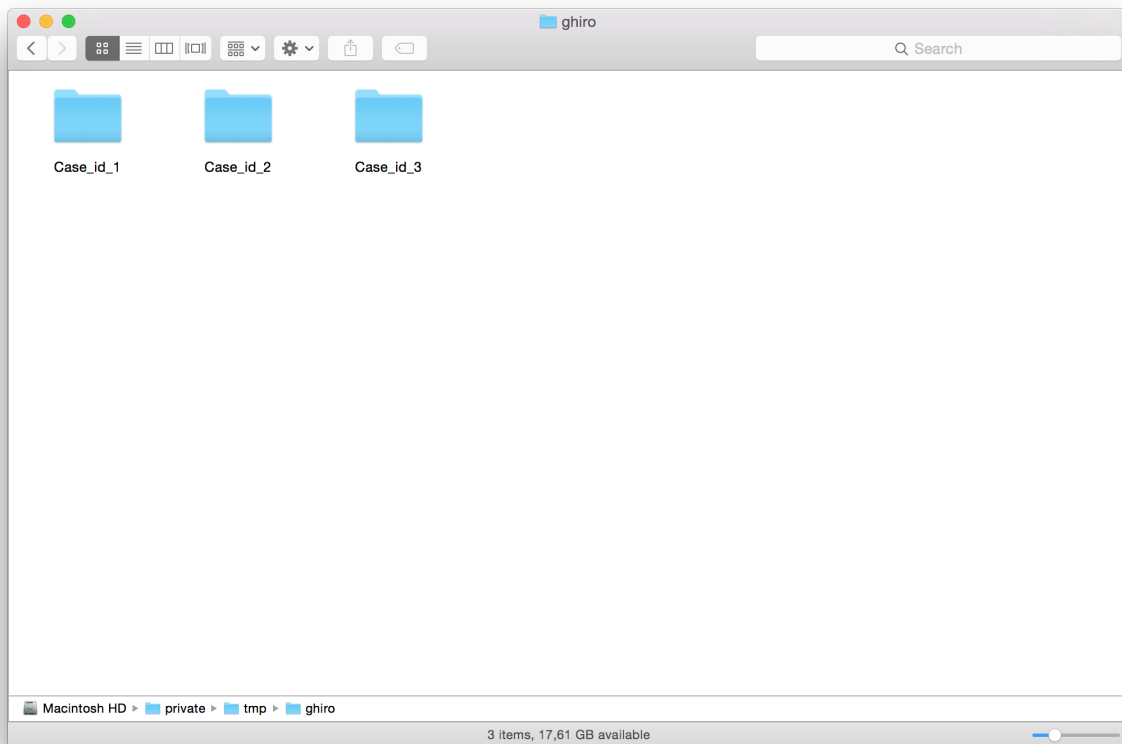
To configure Auto Upload you have to edit *local_settings.py* and configure:

- *AUTO_UPLOAD_DIR* to the directory you want to use for Ghiro Auto Upload
- (optional) *AUTO_UPLOAD_DEL_ORIGINAL* if you want to keep the original files (it is suggested to keep the default value, files will be deleted after being submitted to Ghiro)
- (optional) *AUTO_UPLOAD_STARTUP_CLEANUP* if you don't want to clean *AUTO_UPLOAD_DIR* at startup (it is suggested to keep the default value, *AUTO_UPLOAD_DIR* will be cleaned up at startup)

After configuration, you have to start the directory monitoring with the following command:

```
python manage.py auto_upload
```

Now directory monitoring is running, a tree of folders related to cases will be created, for example:



As you can see there are many folders, one for each case stored, with the case id in the folder name. You should only put the images you want to analyze in a case folder, ghiro will automatically process them, and remove the original file from disk.

This feature is designed to help in many situations:

- When you need to provide people an easy way to submit images to Ghiro, you could just setup a shared folder.
- If you need to analyze a large amount of data, you could submit with a copy files command.

Warning: Don't use the Auto Upload directory configured in *AUTO_UPLOAD_DIR* as a permanent storage! Depending on the configuration, it will be cleaned up each time you run *manage.py auto_upload*. Auto Upload directory is designed as temporary storage to submit images only.

4.2.2 Run processor in debug mode

If you need to run the image processor daemon in debug mode with tracebacks enabled run the following command (inside Ghiro's root):

```
python manage.py process --traceback
```

4.2.3 Create a superuser

If you need to create a new superuser from the command line (at least an user is mandatory to use Ghiro), run the following command (inside Ghiro's root):

```
python manage.py createsuperuser
```

4.2.4 Upload images via command line utility

You can analyze images from command line with the submit utility. It can load and submit for analysis: an image, a folder containing images, a folder containing images and other folders, and recurse inside them.

If you want to add the image located at /target/image.jpg to case with id 2 and owner user name “foobar” run the following command (inside Ghiro’s root):

```
python manage.py submit -u foobar -c 2 -t /target/image.jpg
```

If you want to add all images in folder /target/images to case with id 2 and owner user name “foobar” run the following command (inside Ghiro’s root):

```
python manage.py submit -u foobar -c 2 -t /target/images
```

If you want to add all images in folder /target/images and all subfolders to case with id 2 and owner user name “foobar” run the following command (inside Ghiro’s root):

```
python manage.py submit -u foobar -c 2 -t /target/images -r
```

If you need to load tons of images this utility is designed for you, all images could be loaded in a single batch.

4.2.5 Check for updates

Ghiro automatically checks for new updates every day, if a new release is found a message is shown in the web interface (if you don’t disable the update check as described in this documentation). Anyway a command line command is available to manually check for updates:

```
python manage.py update_check
```

Updates are fetched once day, if you run the command two times the same day, a cached result will be used. If you want to force the update fetch every command you run, use the *force* option:

```
python manage.py update_check --force
```

4.2.6 Save all images

If you need to dump all images in Ghiro’s database, in their original format, to disk, you can save all to disk with:

```
python manage.py images_save_all -p /path/to/disk/
```

4.2.7 Auditing

Ghiro comes with an auditing feature enabled by default. The auditing support is designed to meet the requirements about logging, for example a law enforcement organization or forensics lab could require to have a track of who did what. The logs are stored in your local database and are written in the auditing log file (if you enabled logging on files). Auditing is logging the following information:

- Message describing event.
- Category of the event.

- User who requested the event.
- Client IP address.
- Timestamp.

If you want to disable auditing, set to false `AUDITING_ENABLED` inside `local_settings.py` as follows:

```
AUDITING_ENABLED = False
```

4.2.8 Auditing purge

Auditing logs are stored in Ghiro's database. If you need to purge all audit trails, for example because you want to free some space, there is a management command for that:

```
$ python manage.py purge_auditing
Audit log purge
WARNING: this will permanently delete all your audit logs!
Do you want to continue? [y/n]y
Purging audit log... (it could take several minutes)
Done.
```

Warning: This command wil PERMANENTLY delete all audit data!

4.3 Virtual Appliance

The fastest way to start playing with Ghiro is to download the Ghiro Virtual Appliance (it is a copy of Ubuntu Linux with all you need already installed, you can run on your host) to help people get a running Ghiro in few steps. In few minutes you will have a fully functional Ghiro setup, running in a virtual machine, to start to analyze your images. It is an OVA file, you have to import in your virtualization software (like VirtualBox or VMWare) and configure the networking as explained in the documentation.

4.3.1 Quick start guide

The following steps will guide you through appliance setup.

Step 1. Ready for Virtualization

You can run Ghiro Appliance in any host (Mac, Windows or Linux), only a virtualization software is requested. There are many out there, free and commercial, for example [Vmware](#) or [VirtualBox](#).

VirtualBox is a free and open source virtualization software, so for the sake of this guide we are going to use it, although you can use any other software to run Ghiro Appliance.

You need to have [VirtualBox](#) working, so download and install [VirtualBox](#) following the instruction on his website.

Step 2. Get Ghiro Appliance

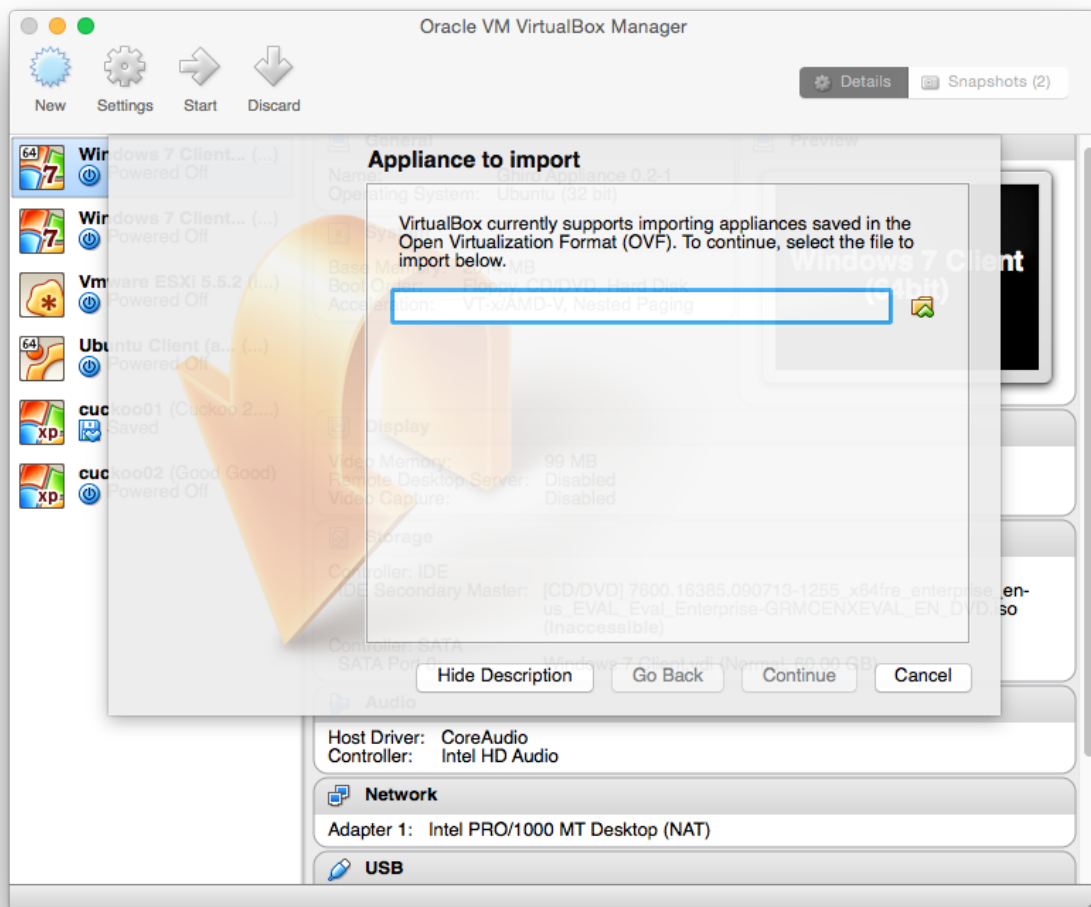
Download Ghiro Appliance from [Ghiro website](#) in OVA format and uncompress it, it is around 600Mb (an image for Vmware ESXi is also available).

You will explode an .OVA file (the appliance), and a readme file with setup instructions.

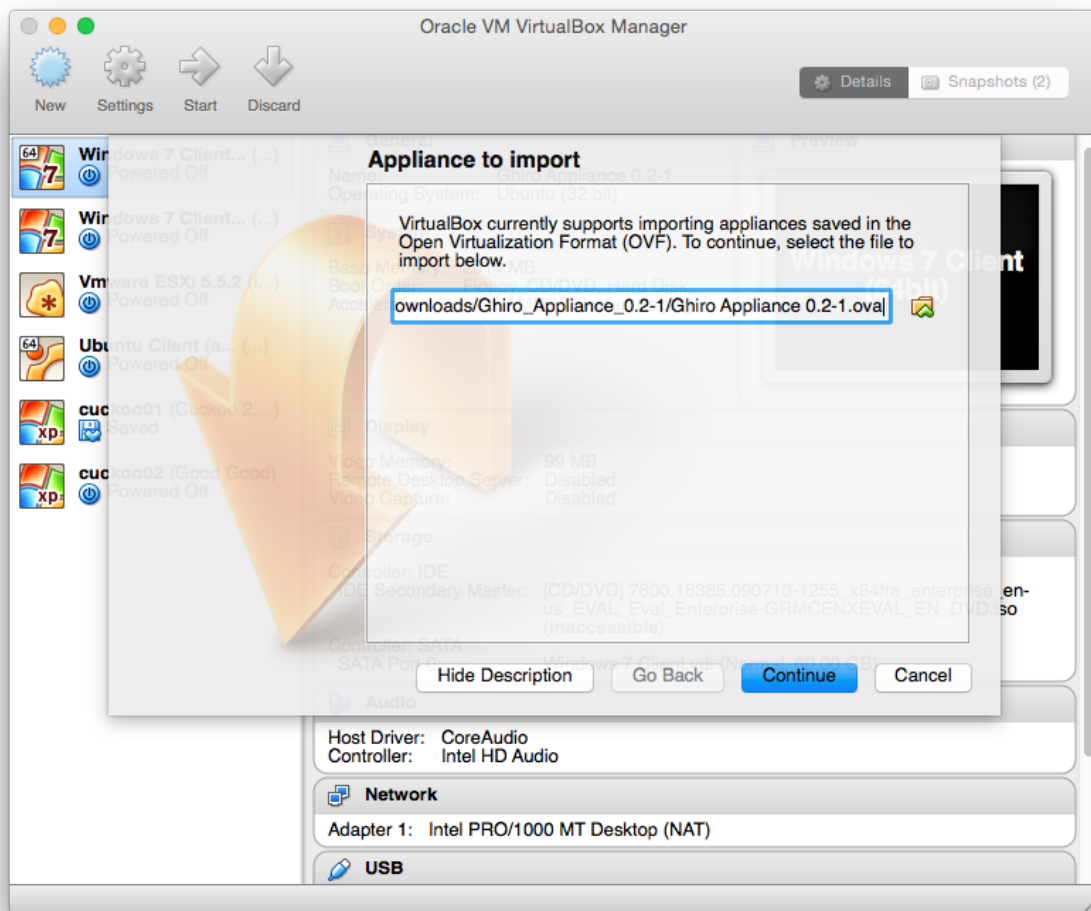


Step 3. Import Appliance

Now you can import the .OVA file inside VirtualBox. Open VirtualBox, go in the menu File and click on “Import Appliance...”, a screen like the following will popup:



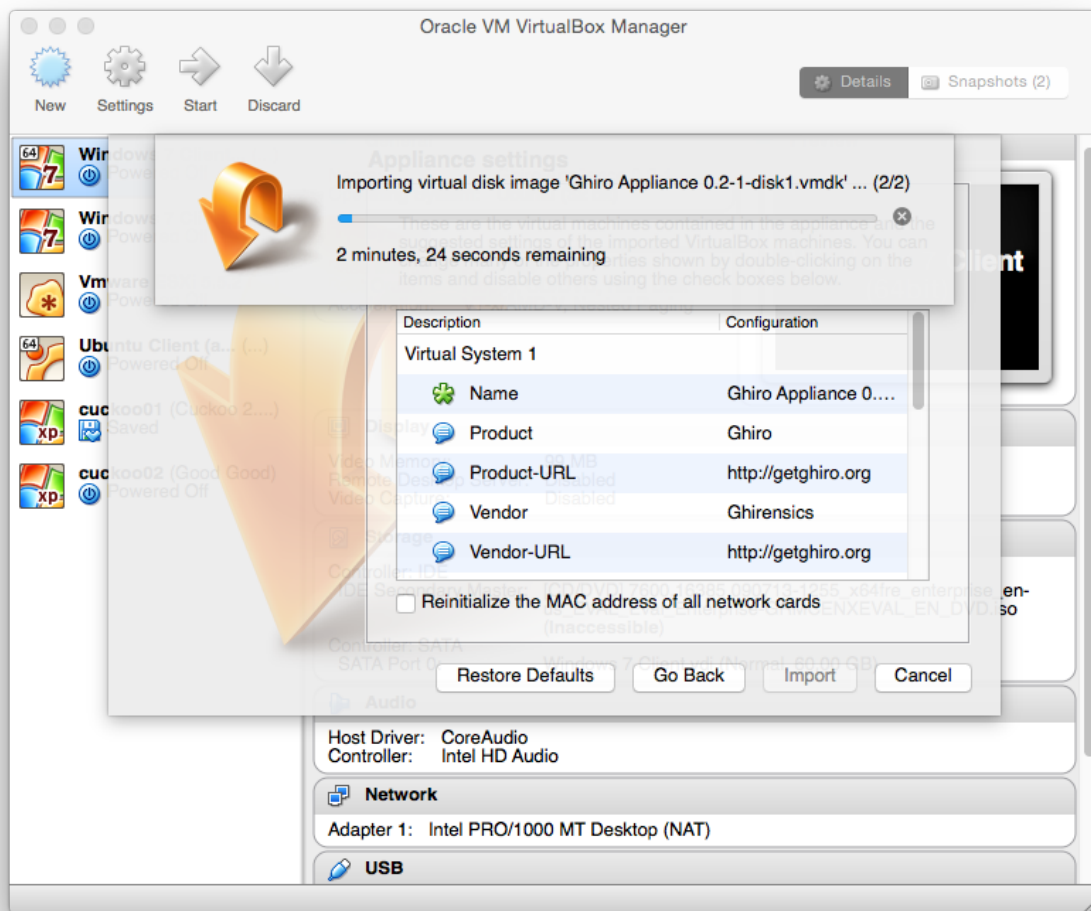
Select the .OVA file and then click “Continue”:



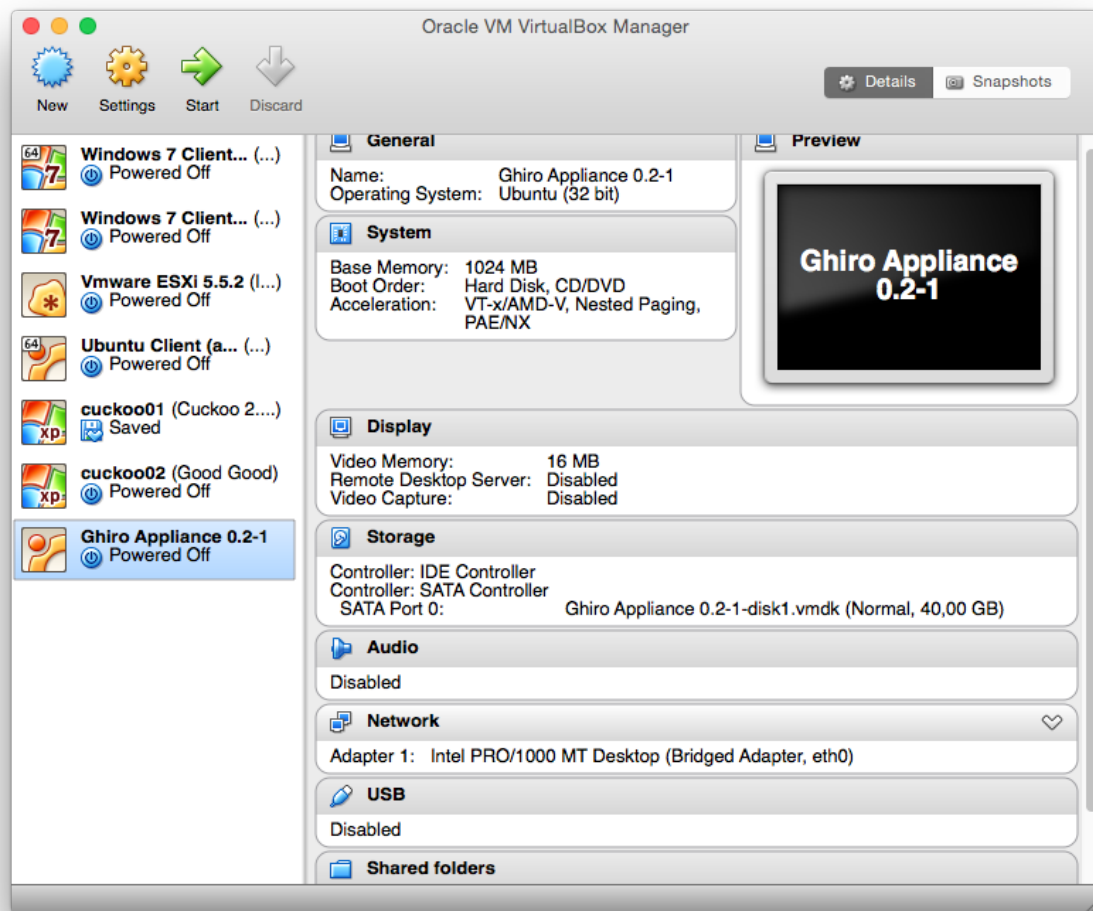
Now a default setting page is proposed, just hit “Import”:



After clicking “Import” the import process will start and in a couple of minutes it will be ready:



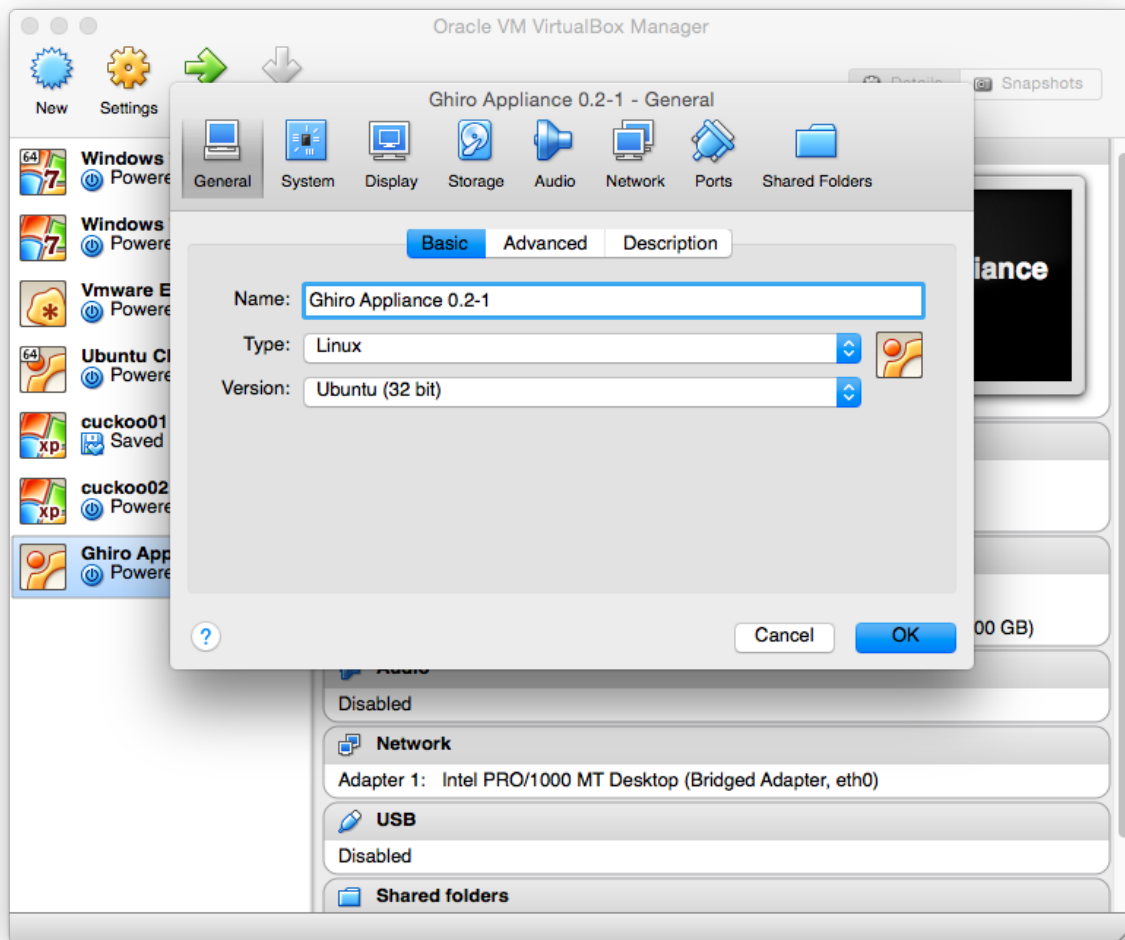
When the appliance is imported you will see it in virtual machines list (don't worry if you don't have all the machines listed in the screenshots, I am sorry but I have many):



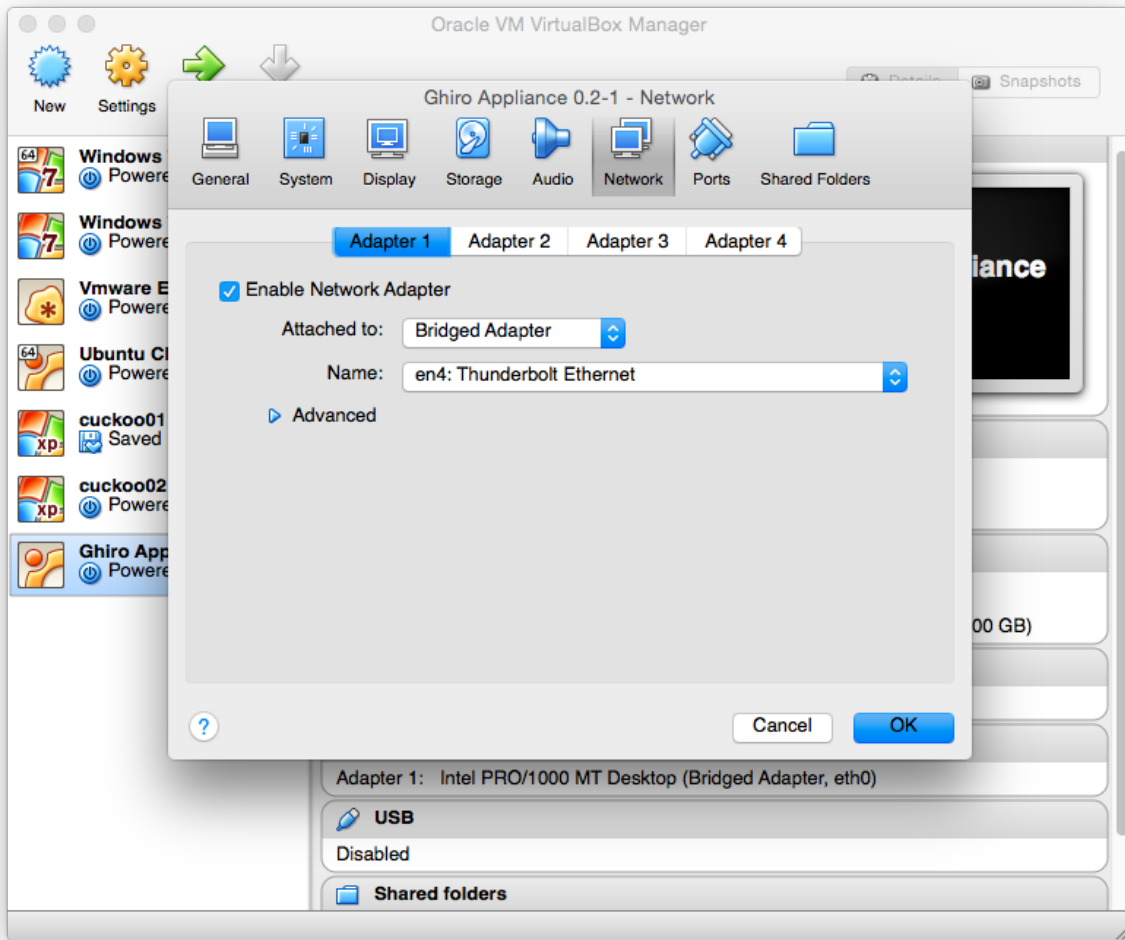
Step 4. Network Configuration

Most people fail configuring the network, so please pay attention.

Right click on your Ghiro Appliance on the Virtual Box Manager window and click on “Settings”.



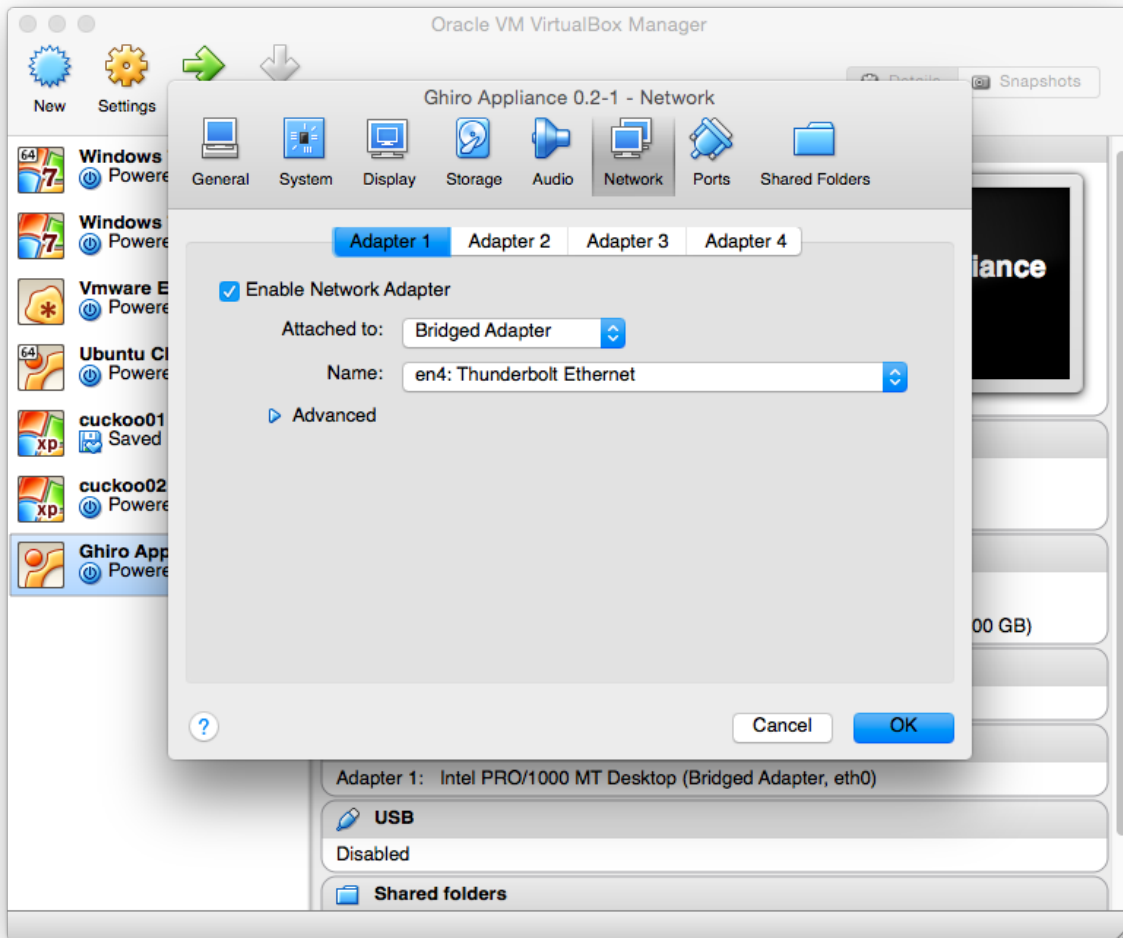
Then choose the “Network” tab.



You have to configure how the virtual machine can connect to your network, so now you are asked to select the network interface you are using and the type of link (bridged or host only).

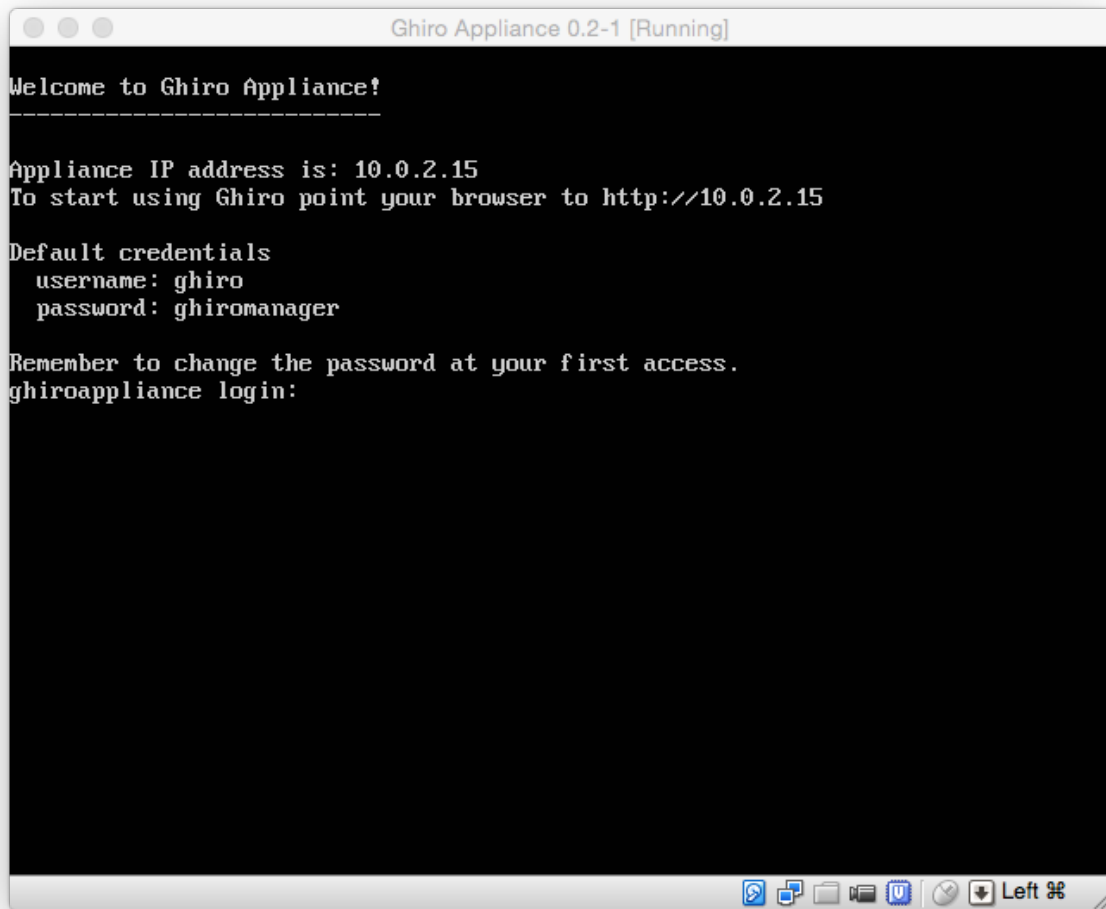
In most cases you need to set “Attached to:” to “Bridged Adapter” and you have to set the “Name” of the network card you are using your for network, for example if you are using your wired interface named “eth0”, select “eth0” on the name drop down menu.

Remember to always set “Attached to:” to “Bridged Adapter” or “Host-only Adapter”, never use NAT or any other option, it will not work due to how networking is implemented in VirtualBox. For more information about connectivity see the [VirtualBox documentation](#).

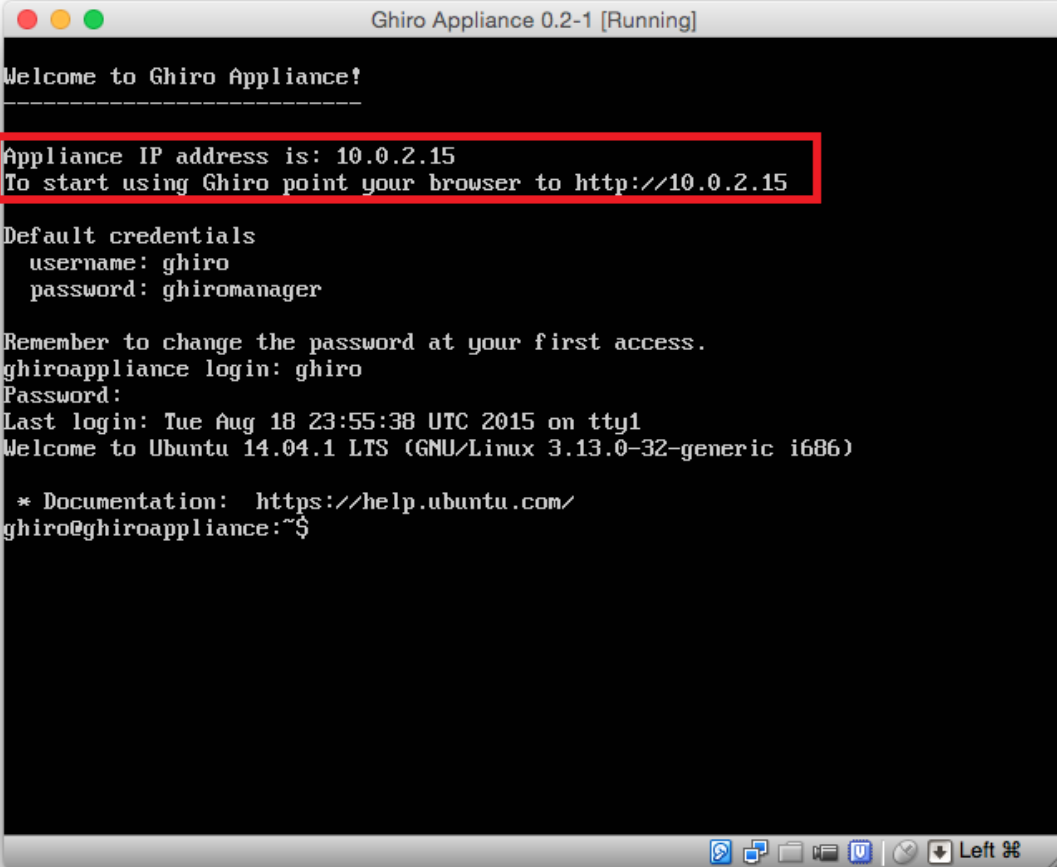


Step 5. Start & Play

Start the Ghiri Appliance selecting it and clicking on “Start”. The boot will start, when the appliance is ready you will see a screen like this one.



The appliance IP address is printed on the screen, as highlighted:



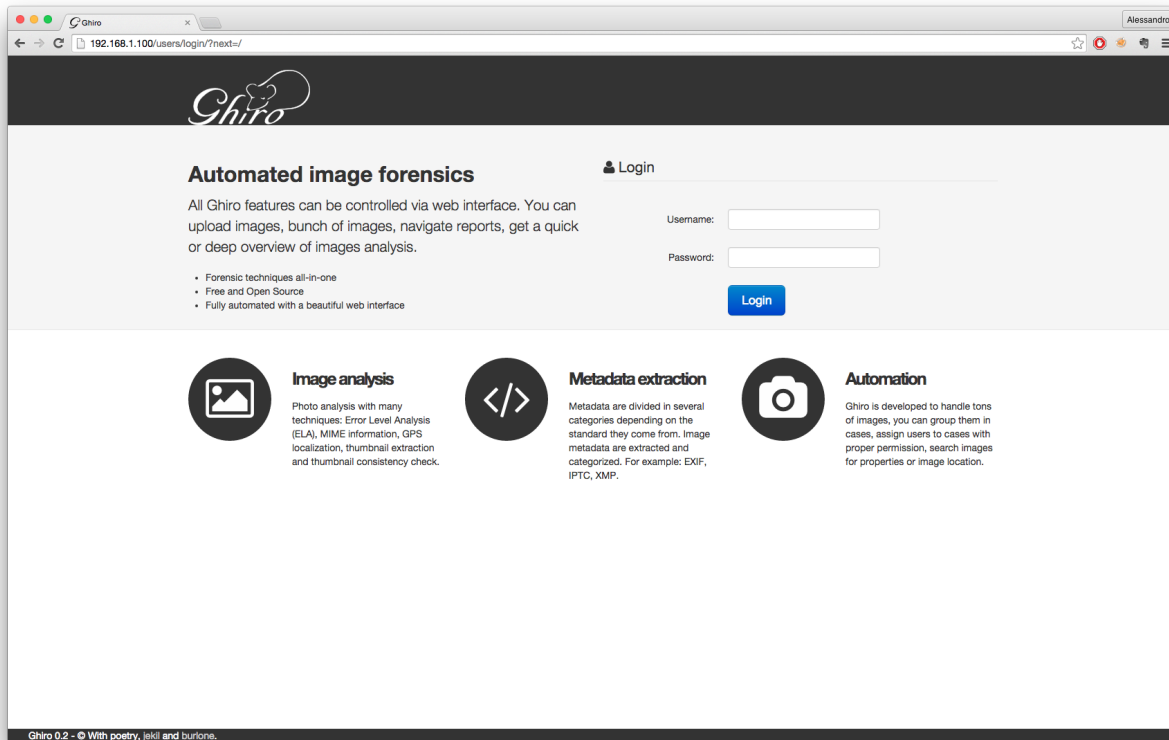
```
Ghiro Appliance 0.2-1 [Running]
Welcome to Ghiro Appliance!
-----
Appliance IP address is: 10.0.2.15
To start using Ghiro point your browser to http://10.0.2.15

Default credentials
  username: ghiro
  password: ghiromanager

Remember to change the password at your first access.
ghiroappliance login: ghiro
Password:
Last login: Tue Aug 18 23:55:38 UTC 2015 on tty1
Welcome to Ubuntu 14.04.1 LTS (GNU/Linux 3.13.0-32-generic i686)

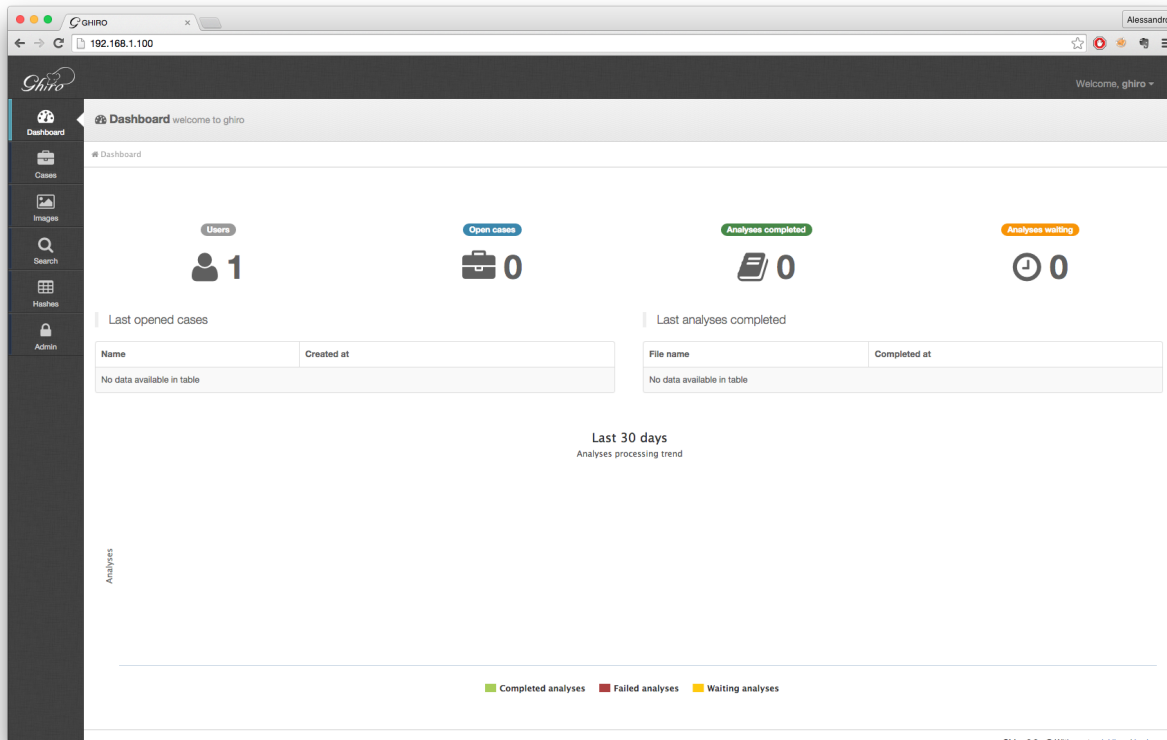
 * Documentation:  https://help.ubuntu.com/
ghiro@ghiroappliance:~$
```

Now just put that address in your browser and the Ghiro interface will appear.



Now login in your browser with the same credentials and you will be ready to play.

- Login: ghiro
- Password: ghiromanager



Now appliance is running, enjoy! You can start analyzing images! Go in the “Cases” panel, create your first case, and add your images with the add button. For usage help please refer to the documentation at: <http://www.getghiro.org/docs/latest/usage/index.html>

4.3.2 How to login

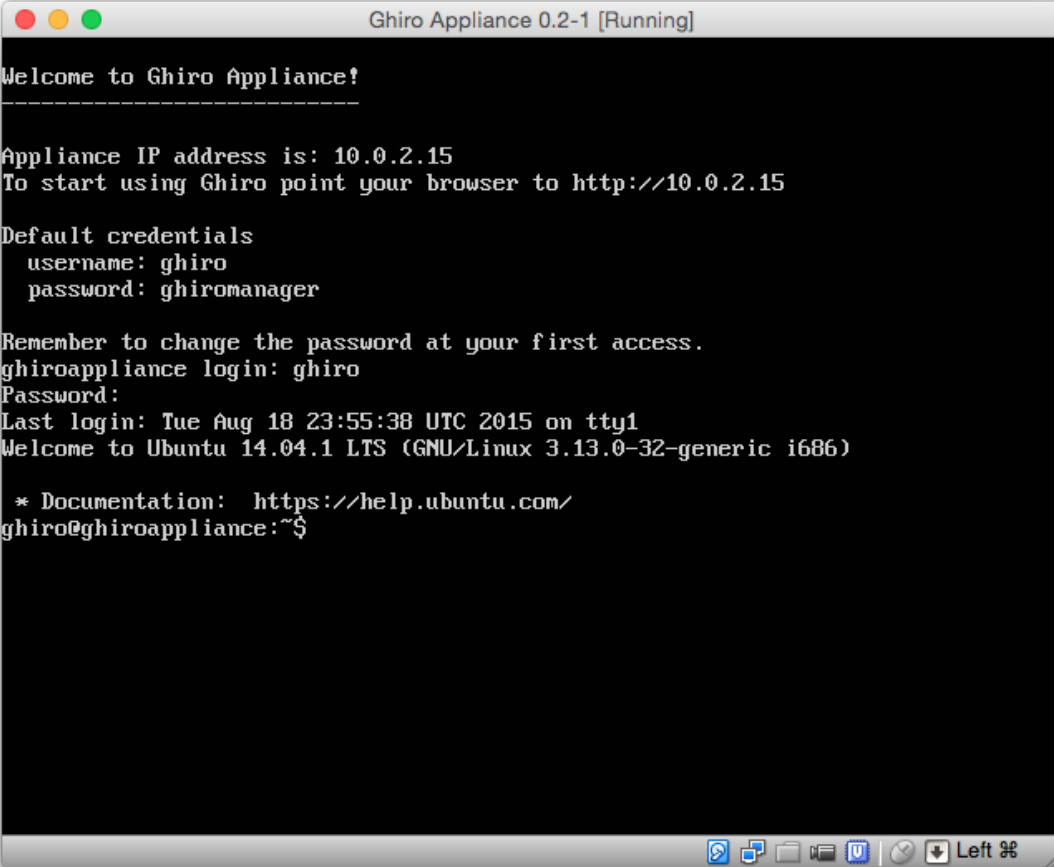
What you see after Ghiro Appliance boot is a Linux login screen, you have just to enter the credentials to login if you need to access the Appliance linux OS.

Login as explained with:

- Login: ghiro
- Password: ghiromanager

Note that when you type the password it will not be shown.

If you login successfully you will see the command prompt, with the blinking cursor after a “\$” symbol.



```
Ghiro Appliance 0.2-1 [Running]
Welcome to Ghiro Appliance!
-----
Appliance IP address is: 10.0.2.15
To start using Ghiro point your browser to http://10.0.2.15

Default credentials
  username: ghiro
  password: ghiromanager

Remember to change the password at your first access.
ghiroappliance login: ghiro
Password:
Last login: Tue Aug 18 23:55:38 UTC 2015 on tty1
Welcome to Ubuntu 14.04.1 LTS (GNU/Linux 3.13.0-32-generic i686)

 * Documentation:  https://help.ubuntu.com/
ghiro@ghiroappliance:~$
```

4.3.3 Uploading files

You can upload images using the following methods:

- Using Web interface * Go to cases page * Create a new case * Upload your images to the case using the add images button
- **Via SMB share (windows shared folders)**
 - Go to cases page
 - Create a new case
 - List the SMB (windows shared folders) shares on your network: Ghiro appliance is exposing all cases in a shared folder, you can upload images related to a case his folder
- **Via FTP share**
 - Go to cases page
 - Create a new case

- Access to FTP listing on Ghiro appliance IP address: Ghiro appliance is exposing all cases in a FTP folder, you can upload images related to a case his folder

4.3.4 Extra Configuration

For extra security, remember to change the default password at your first access for both web application and Linux console access.

If you need to manually configure your IP address: login in, and configure the networking card with your desired IP, for example to give the IP 192.168.0.10 use the following command:

```
sudo ifconfig eth0 192.168.0.10 up
```

If you need to access remotely to the appliance you can use SSH. The appliance is shipped with a default disk of 50GB, if is not enough you can create another virtual disk and add that to the root volume using LVM.

4.3.5 Appliance building

This chapter explains how Ghiro Appliance is created, probably you could skip this if you are not interested in customizing the appliance.

The appliance building script is open source and available under a project dubbed [ghiro-appliance](#) on Github.

Ghiro appliance builder is a [packer.io](#) script to automagically create a Ghiro appliance ready to be used, based on Ubuntu Linux.

Using this script you should be able to create your onw Ghiro appliance updated to latest Ghiro's development branch. You can easily customize the appliance building script to create your own customized appliance.

4.4 API Usage

External systems might like to interact with Ghiro, use their features and share data with it. For example you might be interested in integrating Ghiro with your analysis environment or your scripts, to integrate Ghiro with your analysis pipeline. For example you would have a system sending images to Ghiro and a script to fetch analysis results and process them to send an email only if some circumstances are met. Ghiro comes with a set of JSON API to help automation and integration with external systems. All major functionalities are available through API, although API will be expanded and enriched in future releases.

The available API methods are:

- Create a new case
- Show case contents
- Image upload for analysis (create a new analysis)
- Fetch report of an image analysis

You can use these API as follows.

4.4.1 /api/cases/new

POST /api/cases/new

Adds a new case with name, description optional. Returns the ID of the newly created case.

Example request:

```
curl -kis -F name=foo -F description=bar -F api_key=YOUR_API_KEY http://127.0.0.1:8000/api/c
```

Example response:

```
{"id": 6}
```

Form parameters:

- name (*required*) - case name
- api_key (*required*) - your API key (get it in your profile page)
- description (*optional*) - case description

Status codes:

- 200 - success
- 400 - failure (see the message for error description)

4.4.2 /api/cases/show

POST /api/cases/show

Shows a case. Including image ids for this case.

Example request (requesting case with ID 3):

```
curl -kis -F case_id=3 -F api_key=YOUR_API_KEY http://127.0.0.1:8000/api/cases/show
```

Example response:

```
{"status": "O", "images": [2, 1], "description": "Test case.", "id": 3, "name": "Test"}
```

Form parameters:

- case_id (*required*) - case ID
- api_key (*required*) - your API key (get it in your profile page)

Status codes:

- 200 - success
- 400 - failure (see the message for error description)

4.4.3 /api/images/new

POST /api/images/new

Adds a new image and enqueue it for analysis. Returns the ID of the newly created analysis. This is an example, you should put your Ghiro's server IP address and port in the url.

Example request (image not added to a case):

```
curl -kis -F image=@path_to_image.jpg -F api_key=YOUR_API_KEY http://127.0.0.1:8000/api/imag
```

Example request (image added to case ID 1):

```
curl -kis -F case_id=1 -F image=@path_to_image.jpg -F api_key=YOUR_API_KEY http://127.0.0.1:
```

Example response:

```
{"id": 6}
```

Form parameters:

- `case_id` (*optional*) - case ID
- `image` (*required*) - image file to upload for analysis
- `api_key` (*required*) - your API key (get it in your profile page)

Status codes:

- 200 - success
- 400 - failure (see the message for error description)

4.4.4 /api/images/report

POST /api/images/report

Get image analysis report.

Example request (requesting image analysis with ID 1):

```
curl -kis -F task_id=1 -F api_key=YOUR_API_KEY http://127.0.0.1:8000/api/images/report
```

Example response of completed analysis (truncated):

```
{"id": 1, "status": "C", "data": {"signatures": [], "hash": {"sha1\": "fda88a5aa ..snip..
```

Example response of not completed analysis (truncated):

```
{"id": 1, "status": "P"}
```

Form parameters:

- `task_id` (*required*) - analysis id
- `api_key` (*required*) - your API key (get it in your profile page)

Status codes:

- 200 - success
- 400 - failure (see the message for error description)

4.5 FAQ

4.5.1 How to install GExiv2 on a virtualenv?

First of all setup GExiv2 on your system installing it the usual way, for example with apt-get. Now you have two ways to get it working on a virtualenv. The first is create your virtualenv starting from system packages. The `--system-site-packages` option links the public packages installed on the system to the new virtual environment. The second one is to symlink the GExiv2 library from your virtualenv with:

```
$ cd virtualenv
$ cd lib/python2.7/
$ ln -s /usr/lib/python2.7/dist-packages/gi
```

4.5.2 How to submit a bug report

If you encounter an issue with Ghiro or if you want to suggest a new feature, you are welcome to [submit a new issue](#).

If you are filling a bug report you should always follow this template:

1. What were you doing?
2. What did you expect to happen?
3. What happened instead?
4. Ghiro release of Ghiro Appliance release.
5. Browser and Version of Browser, Operating System running Browser.
6. If you are reporting a UI issue screenshot(s) showing the problem.

4.5.3 Is Ghiro uploading my data somewhere?

Absolutely no. Ghiro is a tool designed for forensic's professionals, law enforcement, and generally people working with critical data related to an investigation. We put great attention in protecting users privacy and security. Ghiro does not upload or disclose your data in any way. Your data are stored in your database and never leave it. Ghiro only contacts, once per day, the update server (located at update.getghiro.org) with a simple HTTP request to check if a new release is available. If you don't like this, you can disable the update check editing `local_settings.py` and setting:

```
UPDATE_CHECK = False
```

If you set this to false no connection will be opened.

5.1 Keep in touch with us

Join the discussion to talk about Ghiro, we want you! Please share your opinion with us, we are happy to get your feedback, we setup many different ways to communicate with us and to keep in touch with the community.

5.1.1 Ghiro & Image Forensics Forum

[Ghiro & Image Forensics Forum](#) is a bulletin board for discussions about image forensics and Ghiro. The forum has several categories, local forums where you can talk in your native language (if a forum for your country is missing, just ask and it will be added), registration with third party account (i.e. google, github), and many other features. Register your account and enjoy!

5.1.2 Mailing list - Google Group

Join our mailing list to the [Ghiro Google Group](#), you will receive updates and discussions by mail (low rate).

5.1.3 IRC Chat

You are welcome in our IRC chat on [Freenode](#):

Server: <code>irc.freenode.net</code> Channel: <code>#ghiro</code>

Join using your favorite IRC client or using the Freenode [web client](#).

5.1.4 WTF mail

If you have something super secret to discuss privately with us just drop an email to: wtf@getghiro.org

5.2 Contribute, bugs and feature requests

We appreciate every feedback, question, bug report, pull requests and contributions. You can help Ghiro in many ways: contributing with code, hunting bugs, submitting feature request, donating money or in many other ways. To

report bugs or request for new features you can use the ticketing system in [official GitHub page](#). or just get in touch with us.

Ghiro is an Open Source project, the users and the community are the most valuable and most important value. We are a group of volunteers and all project's expenses are covered by us, with our personal funds. Your financial contribution will support the maintenance, improvement, and promotion of Ghiro and it is a way of ensuring the longevity and overall health of the project. We accept both sponsorship (a sponsorship is an annual commitment of funds to the project) and donations. We allow the following forms of payment but feel free to contact us for any question.

5.2.1 PayPal

You can donate via a PayPal account or using all credit cards at the [donation page](#).

5.2.2 BitCoin

You can donate using BitCoin to this address: 15TWFTDA3c3LpjzCXGY4WZFXANXjydfBtSN



5.3 Disclaimer

Ghiro 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 license for more details.

Whatever you do with this tool is uniquely your own responsibility.

5.4 License

Ghiro's license is shipped with it and contained in "LICENSE.txt" file inside "docs" folder.

Ghiro is licensed as follows.

```
Ghiro is copyrighted by Ghiro Developers and released under the following
license:
```

```
GNU GENERAL PUBLIC LICENSE
Version 3, 29 June 2007
```

```
Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.
```


Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. 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 them 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 in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. 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.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that

patents cannot be used to render the program non-free.

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

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other

than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological

measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".
- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.
- d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not

used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

- a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.
- b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.
- c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.
- d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.
- e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any

tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of

it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
- e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work

conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If 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 convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU 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 that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

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.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS 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.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

Components shipped with Ghiro are licensed as following.

jQuery is copyrighted by jQuery Foundation, Inc. and released under the following license: GNU GPL, MIT License.

fancyBox is copyrighted by Janis Skarnelis and released under the following license: Creative Commons Attribution-NonCommercial 3.0 License.

datatables is copyrighted by Allan Jardine and released under the following license: GPL v2 license or a BSD style License.

HTML5 Shiv is copyrighted by @afarkas @jddalton @jon_neal @rem and released under the following license: MIT/GPL2 License.

Highcharts is copyrighted by Torstein Hønsi and released under the following license: Creative Commons Attribution-NonCommercial 3.0 License.

Bootstrap is copyrighted by Twitter and released under the following license: Apache License Version 2.0.

Font Awesome is copyrighted by Dave Gandy and released under the following license: SIL OFL 1.1 - <http://scripts.sil.org/OFL>.

Plupload is copyrighted by Moxiecode Systems AB and released under the following

```
license: GPL2 License.
```