# Index



# Topics

## Faraday Server

Faraday Server is the interface between CouchDB and Faraday Client sessions. The server's responsibility is to transmit information between the client and CouchDB, and make sure that they are kept in sync. It also serves the Web UI client, which allows you to handle enormous workspaces from your favorite web browser.

**Important:** You should keep in mind that the Faraday server must be installed on the same machine as CouchDB.

**Also Important:** make sure to use version 1.7.1 of CouchDB, as Faraday doesn't support CouchDB version 2.0.

Unfortunately, in various Debian-based systems (Kali, potentially others), CouchDB 1.7.1 does not work (due to upgrades to a new version of Erlang). If this is the case with your OS, we recommend use Couchdb through a Docker container.

Because of this, we are currently working on migrating to another database engine.

### Downloading

After the purchase you will receive an email with your credentials and a link to our **Customers Portal**. Use those credentials to log in to the site and you will get two links:

- Download License - this is the tarball for the **Faraday License**
- Download Faraday - this is the tarball for the actual **Faraday Code**

Download both those packages and then:

1. Create a new directory and unpack the **Faraday tarball** there. For example, `/home/user/Infobyte/faraday`.
2. Unpack the **License Package** and place its contents in the `doc` directory. For example, using the path from **Step 1**, you should place the **License files** in `/home/user/Infobyte/faraday/doc`.

After doing so, make sure to install system dependencies, install Python dependencies and configure the Server.

### Requirements

Faraday Server is built with minimum requirements. This is by design, so you can install it even on

the most bare-bones machine you can think of.

The Python requirements for the server are stored in the `requirements_server.txt` file.

| Dependency | Version |
|---|---|
| CouchDB | 1.6 |
| Python | 2.6 or 2.7 |
| flask | >= 0.10.1 |
| twisted | >= 16.1.1 |
| sqlalchemy | >=1.0.12 |
| pyopenssl | >16.0.0 |
| couchdbkit | >=0.6.5 |
| restkit | >=4.2.2 |
| requests | >=2.10.0 |
| flask | >=0.10.1 |
| twisted | >=16.1.1 |
| sqlalchemy | >=1.0.12 |
| pyopenssl | >=16.0.0 |
| service_identity | >=16.0.0 |
| python-docx | >=0.8.5 |
| docxtpl | >=0.2.2 |
| six | >=1.10.0 |
| matplotlib | >=1.4.3 |

### Installing system dependencies

#### Debian based distributions (Debian, Ubuntu, Backtrack, etc)

You can run the following command to install the required dependencies on any Debian based distribution.

```
$ sudo apt-get update
$ sudo apt-get install build-essential ipython python-setuptools \
                python-pip python-dev libssl-dev libffi-dev couchdb \
                pkg-config libssl-dev libffi-dev libxml2-dev \
                libxslt1-dev libfreetype6-dev libpng12-dev
```

#### Kali Linux

If you are running Kali, please run the following commands:

```
$ sudo apt-get update
$ sudo apt-get install build-essential ipython python-setuptools \
                python-pip python-dev libssl-dev libffi-dev couchdb \
                pkg-config libssl-dev libffi-dev libxml2-dev \
                libxslt1-dev libfreetype6-dev libpng-dev
```

**Others**

Please consult with your distribution documentation to install the dependencies listed above.

### Installing Python 2 dependencies

Once you have the required system dependencies, you just have to install the Python modules needed to run the server using `pip`:

```
$ pip2 install -r requirements_server.txt
```

### Configuration

Once you have installed the additional dependencies and **CouchDB is running**, you will need to execute the setup script, which will create the admin user with the configured password, and create a backup cronjob for CouchDB. Just execute the following command and answer the questions asked.

```
$ sudo python2 setup-server.pyc
```

If you want more fine grained control over what the setup script does, you can see the available options by executing:

```
$ python2 setup-server.pyc --help
```

---

**This step is optional:** If you want, you can upload the included CWE database to CouchDB to enable better searching and autocomplete features. To do so, you only need to execute the following command:

```
$ ./helpers/pushCwe.py
```

You will be asked to pass as an argument the CouchDB URL so that a connection can be established. For this, use the -c parameter along with your credentials. For example: `sudo ./helpers/pushCwe.py -c http://faraday:password@127.0.0.1:5984`

---

After running the setup script run the server once using `python2 faraday-server.pyc` and don't worry if it fails.

Edit the file located in `~/.faraday/config/server.ini` and add the username and password created using the setup script in the `[couchdb]` section. Assuming the selected password is **changeme**, the file should look something like:

```
[couchdb]
user=faraday
password=changeme
```

#### Authentication

You can create different types of users through the web UI. Those users can login though the same web UI or though a Faraday client using the `--login` flag (Faraday will ask for the credentials later)

#### Exposing the Server

If you wish to access the Server form a different box you need to expose the service. In order to do so, edit the server configuration file and set the `bind_address` param to `0.0.0.0`.

Edit the file located in `~/.faraday/config/server.ini` and under the section `[faraday-server]` set the param, it should look something like this:

```
[faraday-server]
...
bind_address=0.0.0.0
```

Then restart the server if you had it running and reload your browser in case you were already trying to access the Web UI form a different IP.

#### Running

Once everything is installed you need to configure your server properly. Read about Server Configuration.

After configuring, you can proceed to run the Faraday Server script:

```
$ python2 faraday-server.pyc
```

If you want to run the server in background mode, you should use the `--start` option:

```
$ python2 faraday-server.pyc --start
```

This is the recommended way to do this. Other methods like using the bash `&` could cause unexpected IOErrors and other related exceptions.

#### Web UI

Once the server is running, you can access Faraday's Web UI using any browser: just point it to `http://SERVER_IP:SERVER_PORT/_ui` and you can start playing with Faraday.

## Faraday Client

Faraday Client is the software which will allow you to work with your favorite security tools and capture their output in an organized manner. It works under a GTK+3 interface with the popular VTE terminal with a custom ZSH shell that respects the user's configuration (yes, that means you get to keep your exact ZSH terminal inside Faraday, even if you use ZPrezto or Oh My ZSH).

From the client you can also create and delete workspaces, specify plugin configuration, view information about your hosts, resolve conflics that may arise and much more.

It's also a responsibility of the client to send all of the collected information to the server, which will then process it and format it in an friendly way for you to view, edit, and confirm.

The client is bundled in the same package as the server, so if you have already downloaded Faraday, you can skip the next step.

#### Downloading

After the purchase you will receive an email with your credentials and a link to our **Customers Portal**. Use those credentials to log in to the site and you will get two links:

- Download License - this is the tarball for the **Faraday License**
- Download Faraday - this is the tarball for the actual **Faraday Code**

Download both those packages and then:

1. Create a new directory and unpack the **Faraday tarball** there. For example, `/home/user/Infobyte/faraday`.
2. Unpack the **License Package** and place its contents in the `doc` directory. For example, using the path from **Step 1**, you should place the **License files** in `/home/user/Infobyte/faraday/doc`.

After doing so, make sure to install system dependencies, install Python dependencies and configure the Client.

#### Requirements

Faraday Client works under any modern Linux distribution or Mac OS X, and needs the following dependencies:

| Dependency | Version |
| --- | --- |
| CouchDB | 1.6 |
| Python | 2.6 or 2.7 |
| GTK3 | |

| Dependency | Version |
| --- | --- |
| PyGobject | 3.12.0 |
| Vte | API >= 2.91 |
| zsh | |
| CURL | |
| couchdbkit | |
| mockito | |
| whoosh | |
| argparse | |
| IPy | |
| restkit | |
| requests | |
| tornado | |
| flask | |
| colorama | |

The Python requirements for the client are stored in the `requirements.txt` file. Some additional requirements are necessary for specific features to work, these are stored in the `requirements_extras.txt` file.

Out tests include Debian, Ubuntu, Kali, Backtrack and OSX Sierra.

If instead of installing you want to take a quick look at Faraday you can also use Docker.

### Installing system dependencies

#### Debian and derivatives

You can run the following command to install the required dependencies on any Debian based distribution.

```
$ sudo apt-get update
```

If you are running Ubuntu 12.04 LTS, or Ubuntu 14.04 LTS, please execute this command:

```
$ sudo apt-get install libpq-dev python-pip python-dev gir1.2-gtk-3.0
gir1.2-vte-2.91 python-gobject zsh curl
```

If you are any other version, please execute the following command:

```
$ sudo apt-get install libpq-dev python-pip python-dev gir1.2-gtk-3.0
gir1.2-vte-2.91 python-gobject zsh curl
```

#### Kali

Faraday comes pre-installed in Kali Rolling. However, that version is **incompatible** with the **Pro License**. Follow the Debian install steps to install.

#### ArchLinux

Before installing Faraday you will need to get some user-contributed packages. In order to do this quickly we need an AUR wrapper, in this case we will use Yaourt. After installing Yaourt run:

```
$ yaourt -S python2-dateutil python2-pip mime-types python2-gobject gtk3 vte3
postgresql-libs
```

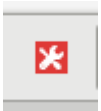
### Installing Python 2 dependencies

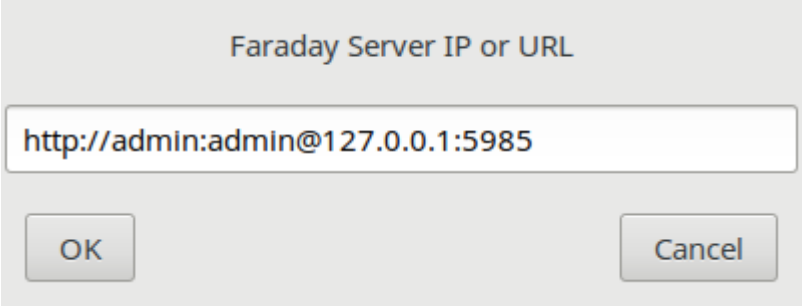
Once you have the required system dependencies, you just have to install the Python modules needed to run the client using `pip`:

```
$ pip2 install -r requirements.txt
```

### Configuration

Now you need to configure every Faraday instance so it can connect to the server.

- If you're using the GTK interface click on the Preferences icon  and fill in the server URL, for example **http://127.0.0.1:5985**




- If you are using the ***--gui=no-gui*** option

Edit the file: `~/.faraday/config/user.xml` And search for the following **couch_uri** tag and set it to the server URL, for example:

```
<couch_uri>http://127.0.0.1:5985</couch_uri>
```

### Running

Once you have already configured the client and have Faraday Server running, you simply have to run:

```
$ python2 faraday.pyc
```

Some distributions or installations require additional steps, so look down below if you are using something different than Debian or Ubuntu, or if you need to apply some configuration to the client.

#### Docker

You can find instructions on how to run the client inside a Docker container here.

#### OSX

You can find instructions on how to run the client under Mac OSX here.

#### Chef

If you want to deploy Faraday using Chef, Sliim made a cookbook for it! You can find it here.