

# README

---

## UPD89

---

upd89 is a system update management for debian based systems. It consists of two components: the control center (<https://github.com/upd89/controlcenter>) and agents (<https://github.com/upd89/agent>).

### Control Center

Central web application for managing the connected agents and creating tasks for the systems.

### Agent

Python-based component that runs as a daemon on each managed server and notifies the control center about new updates and receives tasks from the control center to install.

## Dependencies

---

This application is based on "Ruby on Rails" (RoR).

Tested environment:

- ruby-2.2.3
- Rails 4.2.4
- passenger 5.0.27 / Apache/2.4.7 (Ubuntu 14.04)
- postgresql-9.3

To make sure you have the correct versions of ruby and rails as well as git and postgres installed, run the following:

```
$ ruby -v
$ rails -v
$ git --version
$ which psql
```

The result should ideally look like this:

```
$ ruby -v
ruby 2.2.3p173 (2015-08-18 revision 51636) [x86_64-linux]
$ rails -v
Rails 4.2.4
$ git --version
git version 2.5.0
$ which psql
/usr/bin/psql
```

If you don't have a recent version of RoR, please have a look at the following installation guides:

- [Setup Ruby On Rails on Ubuntu](#)
- [How To Deploy a Rails App with Passenger and Apache on Ubuntu 14.04](#)

## Installation

---

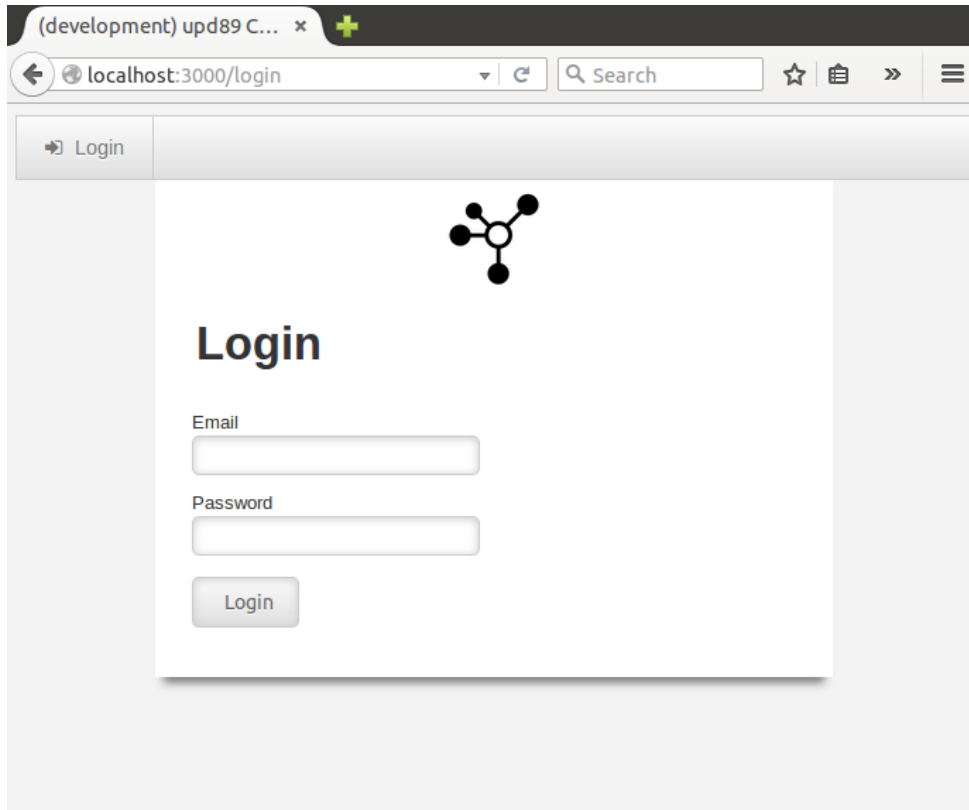
```
git clone https://github.com/upd89/controlcenter.git
cd controlcenter
bundle install
```

```
rake db:create
rake db:migrate
rake db:base_data
```

## Start the Server locally

```
rails server
```

You should now be able to visit a local instance on `https://localhost:3000`



The initial rake task created a couple of users, most importantly `admin` (also doubles as email) with the password `RF9wRF9w`, with which you can log in.

## Deployment on Apache

Setup Certificate authority:

```
apt install easy-rsa
make-cadir ca
```

Change the following entries (recommended) in `ca/vars` (with your own settings, of course):

```
export KEY_SIZE=4096
export KEY_COUNTRY="CH"
export KEY_PROVINCE="ZH"
export KEY_CITY="Zuerich"
export KEY_ORG="UPD89"
export KEY_EMAIL="hello@upd89.org"
export KEY_OU="Dev"
```

in `ca/openssl-1.0.0.cnf`, set the usage to both client and server so it can be used in both directions (there are 2 entries, so do this twice):

```
extendedKeyUsage=serverAuth,clientAuth
```

Afterwards, set up your CA:

```
cd ca
export EASY_RSA="${EASY_RSA:-.}"
. vars
./clean-all
"$EASY_RSA/pkitool" --initca
"$EASY_RSA/pkitool" --server cc.upd89.org
"$EASY_RSA/pkitool" agent1.upd89.org
"$EASY_RSA/pkitool" agent2.upd89.org
"$EASY_RSA/pkitool" agent3.upd89.org
```

Follow this guide <https://www.digitalocean.com/community/tutorials/how-to-deploy-a-rails-app-with-passenger-and-apache-on-ubuntu-14-04>

But: Use <https://raw.githubusercontent.com/upd89/controlcenter/master/apache.conf.sample> instead of the suggested content for the conf file and replace the following variables:

```
$_HOSTNAME_ with your desired hostname (or localhost),
$_ROOTDIR_ with the installation directory of the rails application,
$_RAILSENV_ with the desired rails environment (e.g. development, production),
$_SSLCERTFILE_ with the SSL certificate file (for the web interface)*,
$_SSLKEYFILE_ with the SSL key file (for the web interface)*,
$_SSLCHAINFILE_ with the SSL chain file (for the web interface)*,
$_UPD89CA_ with the absolute path to ca/keys/ca.crt
$_SSLAPICERTFILE_ with the SSL certificate file for the API (ca/keys/cc.upd89.org.crt),
$_SSLAPIKEYFILE_ with the SSL key file for the API (ca/keys/cc.upd89.org.key)
```

\* Recommendation: use letsencrypt ()

```
sudo a2enmod rewrite
sudo a2enmod ssl
sudo a2enmod passenger
apache2ctl configtest
sudo service apache2 restart
```

Now you should have a functioning web server, congratulations!

## Configuration

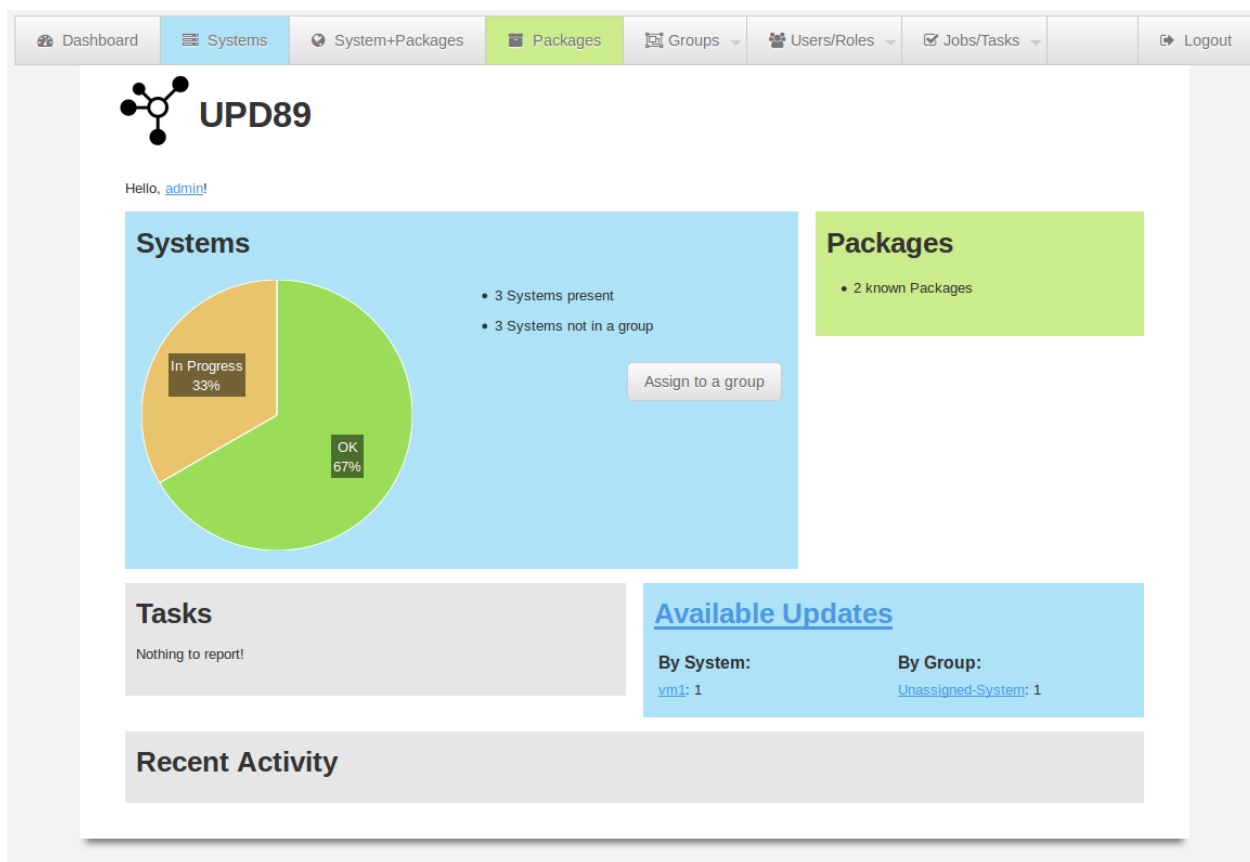
To configure some view-related settings, you can change some variables in `config/settings.yml`. Each setting is commented and should be self-explanatory.

Database-related settings can be changed in `config/database.yml`

If you want some more example entries, you can run

```
rake db:sample_data
```

This will create a number of basic entries for systems, packages and others for testing purposes.



## Stuck?

If you're stuck, need help, have a problem or need some other assistance, there's always the [how-to](#) where some more detailed instructions can be found.