

# Installation guide

For production, we provide deb packages for Ubuntu 14.04 Trusty and Ubuntu 18.04 Bionic on the AMD64 platform. We recommend using the package for Ubuntu 18.04.

The package can be installed inside a Docker container, however at the time of this writing there is no official image yet and running the package in a container requires some tweaks as there is no systemd inside a Docker container (and there shouldn't be).

It is also possible to use our old package version for Ubuntu 14.04 Trusty distribution on the AMD64 platform.

## Ubuntu package - bionic and newer - recommended

This is a quick introduction on how to install Ralph on Ubuntu 18.04 Bionic.

We introduced some changes in the Ubuntu 18.04 Bionic package:

- Ralph now uses Python 3.6
- settings are located in /etc/ralph
- database settings are configured via debconf prompts during a fresh installation
- ralphctl command has been introduced for Ralph management
- Ralph runs as a systemd service

The settings are just environment variables that are passed to Ralph and then used as Django settings.

## Ralph installation

The steps below can be executed on any clean installation Ubuntu 18.04 Bionic.

```
sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv-keys E2D0F3764B54797F
sudo sh -c "echo 'deb https://dl.bintray.com/allegro/debng bionic main' > /etc/apt/sources.list.d/ralph.list"
sudo apt-get update
sudo apt-get install mysql-server nginx ralph-core
```

When prompted, input Ralph database settings. For testing purposes, choosing the default settings will be fine. You can review the settings later in `/etc/ralph/conf.d/database.conf`.

## Nginx configuration

Configure nginx by editing `/etc/nginx/sites-available/default` file and pasting the following:

```
server {

    listen 80;
    client_max_body_size 512M;

    proxy_set_header Connection "";
    proxy_http_version 1.1;
    proxy_connect_timeout 300;
    proxy_read_timeout 300;

    access_log /var/log/nginx/ralph-access.log;
    error_log /var/log/nginx/ralph-error.log;

    location /static {
        alias /usr/share/ralph/static;
        access_log off;
        log_not_found off;
        expires 1M;
    }

    #location /media {
    #    alias /var/local/ralph/media;
    #    add_header Content-disposition "attachment";
    #}

    location / {
```

```
        proxy_pass http://127.0.0.1:8000;
        include /etc/nginx/uwsgi_params;
        proxy_set_header Host $http_host;
        proxy_set_header X-Forwarded-For
$proxy_add_x_forwarded_for;
    }
}
```

After that, restart nginx:

```
sudo systemctl restart nginx.service
```

## Database configuration

Once Ralph is installed, you can create Ralph database and a database user:

```
sudo mysql
mysql> create user 'ralph_ng'@'127.0.0.1' identified by 'ralph_ng';
mysql> grant all privileges on ralph_ng.* to
'ralph_ng'@'127.0.0.1';
mysql> create database ralph_ng;
```

Create the database schema, create Ralph superuser and populate the database with some data:

```
sudo ralphctl migrate
sudo ralphctl createsuperuser
sudo ralphctl demodata
```

## Starting Ralph

Now just a finishing touch:

```
sudo ralphctl sitetree_resync_apps
sudo systemctl enable ralph.service
sudo systemctl start ralph.service
```

And you are all set. Navigate to your new Ralph installation. Just follow this link: <http://localhost> [http://localhost].

## Troubleshooting

If something goes wrong, you can take a peek at these log files:

```
/var/log/ralph/ralph.log
/var/log/ralph/gunicorn.error.log
/var/log/ralph/gunicorn.access.log
/var/log/nginx/ralph-error.log
/var/log/nginx/ralph-access.log
```

## Next steps

Once you familiarize yourself with Ralph, you can use the example above as an inspiration for creating the configuration that suits your needs.

It would probably be a good idea to have the database located on a different host with a password different than the default one and use a load balancer for ssl traffic termination (or just to configure nginx to use ssl).

Don't forget to read our quick start: <https://ralph-ng.readthedocs.io/en/latest/user/quickstart/> [https://ralph-ng.readthedocs.io/en/latest/user/quickstart/]!

## Debian/Ubuntu package - trusty/jessie

Make sure, your installation is clean Ubuntu 14.04, without any other packages installed, and `apt-transport-https` installed.

```
sudo apt-get update && sudo apt-get install apt-transport-https
```

Now, add our official ralph repository:

```
sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv-keys 379CE192D401AB61
sudo sh -c "echo 'deb https://dl.bintray.com/vi4m/ralph wheezy main' > /etc/apt/sources.list.d/vi4m_ralph.list"
```

Then, just install ralph the traditional way:

```
sudo apt-get update
sudo apt-get install ralph-core redis-server mysql-server
```

Note: Main www instance of Ralph requires redis and mysql server installed. If you want to install only ralph agent somewhere, just install `ralph-core` and point it to the particular mysql and redis instance somewhere on your network.

## Configuration

Create the database.

```
> mysql -u root
> CREATE database ralph default character set 'utf8';
```

## Settings

We are working on some sane configuration management files. Currently, we just read some environment variables, so just paste somewhere in your `~/.profile` following environment variables customizing it to your needs.

cat `~/.profile`

```
export DATABASE_NAME=ralph
export DATABASE_USER=someuser
export DATABASE_PASSWORD=somepassword
export DATABASE_HOST=127.0.0.1
export PATH=/opt/ralph/ralph-core/bin/:$PATH
export RALPH_DEBUG=1
```

## Initialization

1. Type `ralph migrate` to create tables in your database.
2. Type `ralph sitetree_resync_apps` to reload menu.
3. Type `ralph createsuperuser` to add new user.

Run your ralph instance with `ralph runserver 0.0.0.0:8000`

Now, point your browser to the `http://localhost` and log in. Happy Ralphing!

## Docker installation (experimental)

You can find experimental docker-compose configuration in

<https://github.com/allegro/ralph/tree/ng/contrib>

[<https://github.com/allegro/ralph/tree/ng/contrib>] directory. Be aware, it is still a beta.

### Install

Install docker and [docker-compose](http://docs.docker.com/compose/install/) [http://docs.docker.com/compose/install/] first.

### Create compose configuration

Copy `docker-compose.yml.tpl` outside ralph sources to `docker-compose.yml` and tweak it.

### Build

Then build ralph:

```
docker-compose build
```

To initialize database run:

```
docker-compose run --rm web /root/init.sh
```

Notice that this command should be executed only once, at the very beginning.

If you need to populate Ralph with some demonstration data run:

```
docker-compose run --rm web ralph demodata
```

### Run

Run ralph at the end:

```
docker-compose up -d
```

Ralph should be accessible at `http://127.0.0.1` (or if you are using `boot2docker` at `$(boot2docker ip)`). Documentation is available at `http://127.0.0.1/docs`.

If you are upgrading ralph image (source code) run:

```
docker-compose run --rm web /root/upgrade.sh
```

## Migration from Ralph 2

If you used Ralph 2 before and want to save all your data see [Migration from Ralph 2 guide](#) [`../data_migration/#migration_ralph2`]