
Drups.io Engine

Release 0.1

Drups.io

May 28, 2021

CONTENTS

- 1 What is Drups.io Engine 3**
- 2 Installation 5**
 - 2.1 Requirements 5
 - 2.2 Production 5
 - 2.3 Development 5
- 3 Usage 7**
 - 3.1 Production 7
 - 3.1.1 Using init.d 7
 - 3.1.2 Using celery 8
 - 3.1.3 Using drups itself 8
 - 3.2 Development 8

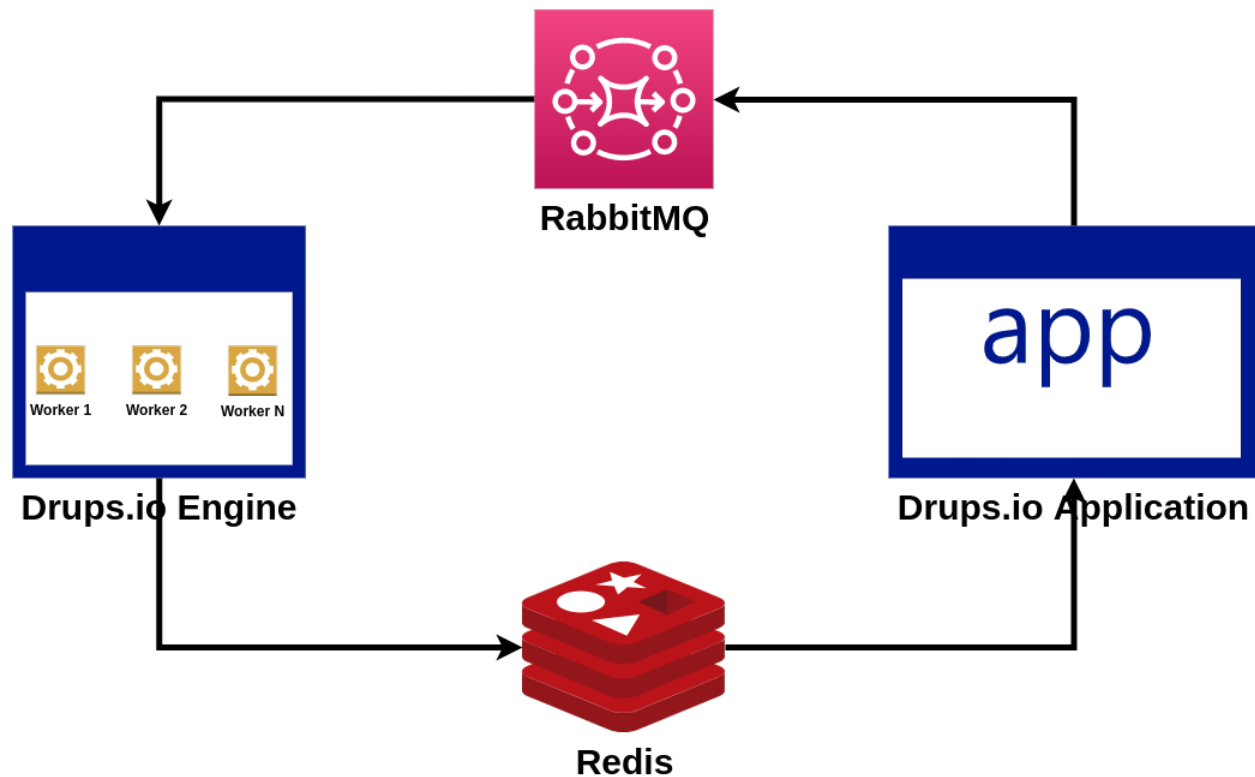
Contents:

WHAT IS DRUPS.IO ENGINE

Drups.io Engine is a bridge between [Drups.io Application](#) and DevOps stuff. It takes care about Cloning, Building and Deploying the user projects.

The connection between Engine and Application is implemented by using the [Message Broker pattern](#). We use [RabbitMQ](#) as our broker and [Redis](#) as the Celery Result Backend. [Celery](#) is our Task Queue provider.

Connection between Drups.io Engine and application



INSTALLATION

2.1 Requirements

- Python `>= 3.6`
- `pip`
- `venv` (recommended)
- RabbitMQ Server
- Redis Server

2.2 Production

```
$ pip install drups
```

2.3 Development

- Clone the repository: `git clone git@github.com:drupsio/engine.git`
- Go to the project directory: `cd engine`
- Install pip dependencies: `pip install -r requirements/dev.txt`

3.1 Production

First you need to define the environment variables:

```
# RabbitMQ URL.
$ export BROKER_URL="amqp://some_user:some_password:127.0.0.1"

# Redis URL.
$ export RESULT_BACKEND="redis://127.0.0.1"
```

3.1.1 Using init.d

- Copy the `init.d/drupsd` file into `/etc/init.d/drupsd`
- Modify and copy the `init.d/drupsd.config.example` file to `/etc/default/drups` (or `/usr/local/etc/drups` on BSD)
- Run the daemon `sudo /etc/init.d/drupsd start`. It is important to run it with the `root` user

The default user and group for running the `drupsd` daemon is `drups:drups`. You should create it first (or change the default user in `/etc/default/drups` -> `DRUPSD_USER` and `DRUPSD_GROUP`).

Available command for drups daemon

- `start` - Start the daemon
- `stop` - Stop the daemon
- `restart` - Restart the daemon
- `status` - Get the daemon status
- `kill` - Kill the daemon
- `dryrun` - Start the daemon in verbose mode

3.1.2 Using celery

```
$ celery -A drups worker --loglevel=INFO -E
```

3.1.3 Using drups itself

```
$ drups worker --loglevel=INFO -E
```

3.2 Development

- Create the `.env.local` file and set the environment variables:

```
# Example .env.local

# RabbitMQ URL.
BROKER_URL = 'amqp://some_user:some_password:127.0.0.1'

# Redis URL.
RESULT_BACKEND = 'redis://127.0.0.1'
```

- Run the Celery worker

```
$ celery -A drups.app worker --loglevel=INFO -E
```