
Drups.io Docker

Release 0.1

Drups.io

May 28, 2021

CONTENTS

- 1 What is Drups.io Docker 3**
 - 1.1 Stack 3
 - 1.2 Architecture 5
- 2 Installation 7**
 - 2.1 Requirements 7
 - 2.2 Installation Process 7
- 3 Usage 9**
 - 3.1 Basic Make commands 9
 - 3.2 Project Settings 9
 - 3.2.1 Basic variables from .env 9

Contents:

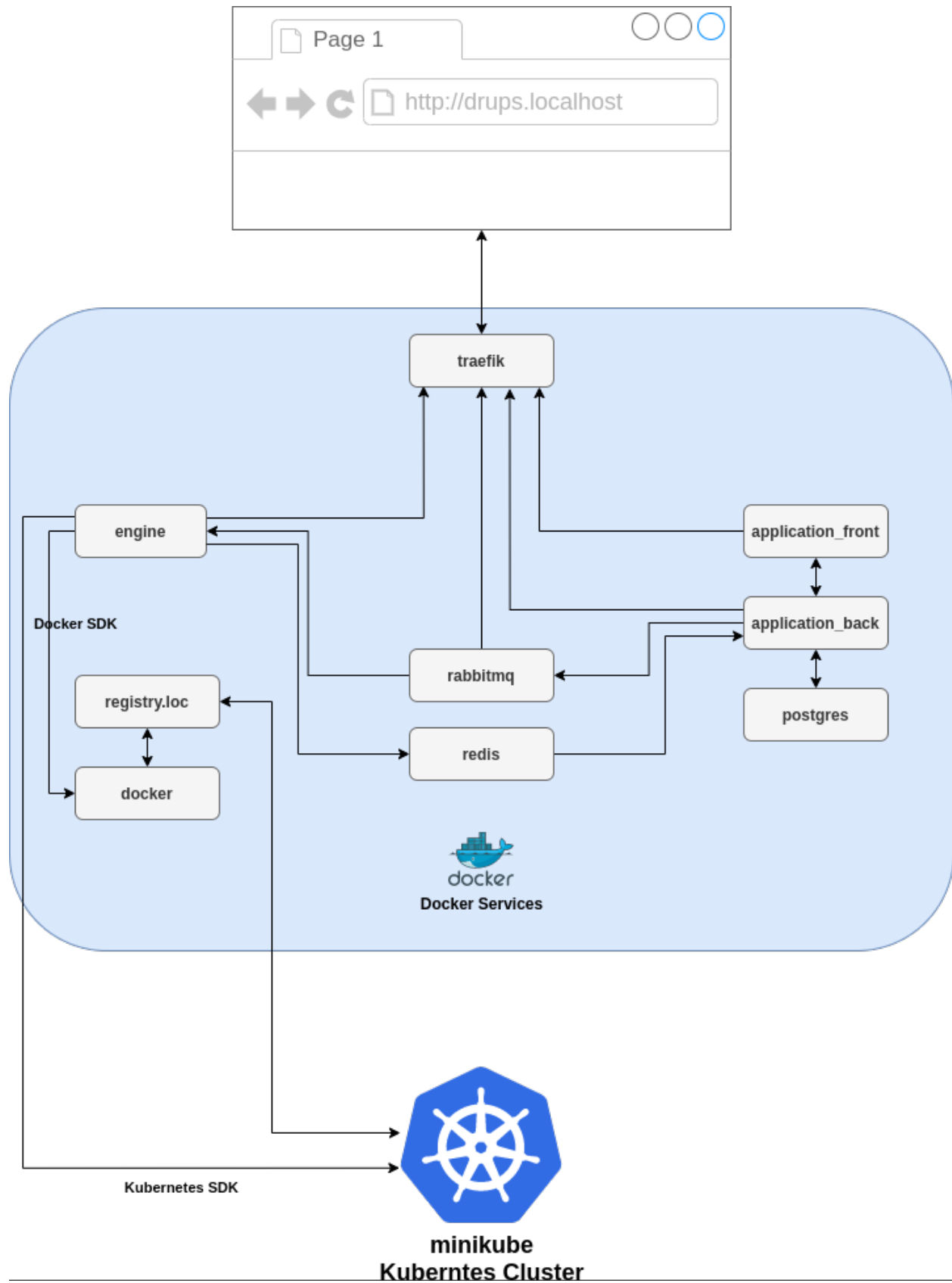
WHAT IS DRUPS.IO DOCKER

Drups.io Docker is a development environment for Drups.io Dev and DevOps teams. It creates production-similar environment for developing [Drups.io Engine](#) and [Drups.io Application](#) using Docker containers and custom building tools.

1.1 Stack

Container	Service name	Image
Engine	engine	_/python
Application Back	application_back	_/node
Application Front	application_front	_/node
RabbitMQ	rabbitmq	_/rabbitmq
Redis	redis	_/redis
Postgres	postgres	_/postgres
Docker	docker	_/docker
Docker Registry	registry.loc	_/registry
Traefik	traefik	_/traefik

1.2 Architecture



INSTALLATION

2.1 Requirements

- Docker
- Docker Compose
- GNU Make
- OpenSSL
- kubectl
- minikube (Recommended)

You need to have a running [Kubernetes](#) Cluster on your machine. We recommend using [minikube](#). *minikube is local Kubernetes, focusing on making it easy to learn and develop for Kubernetes.* You can install minikube using the [official guide](#).

2.2 Installation Process

- Clone the repository: `git clone git@github.com:drupsio/docker.git`
- Go to the project directory: `cd docker`
- Run the project installer: `make install`

The installer script will clone the application parts into `/apps` directory, install them and run servers and daemons. You can get information about running stack (Application URLs, Credentials, etc.) by running `make info`.

3.1 Basic Make commands

- `make up`- Pull and start up containers.
- `make stop`- Stop containers.
- `make start`- Start containers without updating.
- `make shell`- Access engine container via shell. You can optionally pass an argument with a service name to open a shell on the specified container (e.g. `make shell application_back`, `make shell rabbitmq`).

Note: You can see all available commands by running `make help`.

3.2 Project Settings

Project settings are stored in the `.env` file. You can modify them before running `make up` or `make install`. e.g. you can change the version of Redis in the `redis` container by updating the `REDIS_TAG` variable.

3.2.1 Basic variables from `.env`

- `ENGINE_GIT_REPO/APPLICATION_GIT_REPO` - Git repository of Engine/Application to clone.
- `ENGINE_GIT_BRANCH/APPLICATION_GIT_BRANCH` - Git branch of Engine/Application to clone.
- `POSTGRES_USER` - Username of Postgres default user.
- `POSTGRES_PASSWORD` - Password of Postgres default user.