

# Prerequisites

- [Docker Swarm \(https://docs.docker.com/get-started/part4/\)](https://docs.docker.com/get-started/part4/)
- [pip \(https://pip.pypa.io/en/stable/installing/\)](https://pip.pypa.io/en/stable/installing/) (optional)
- [git \(https://git-scm.com/book/en/v2/Getting-Started-Installing-Git\)](https://git-scm.com/book/en/v2/Getting-Started-Installing-Git)

## Setup

Clone the cion repository

```
$ git clone https://github.com/cionkubes/cion
$ cd cion
```

Create secrets that you will need

- [Token for webhook url \(secrets.md#token\)](#)
- [Login details for docker repositories that are private. \(secrets.md#dockerhub\)](#)
- [TLS certificates for any docker swarms with services you want to update \(secrets.md#docker\)](#)
- [Kubernetes service user for any kubernetes cluster with deployments you want to update \(secrets.md#kubernetes\)](#)

## Configuring the stack

It is recommended you have some experience with docker compose or docker stack, you can read about it [here](https://docs.docker.com/compose/compose-file/#service-configuration-reference).  
(<https://docs.docker.com/compose/compose-file/#service-configuration-reference>)

You may create your own compose file ([example \(https://github.com/cionkubes/cion/blob/master/docker/cion-compose.yml\)](https://github.com/cionkubes/cion/blob/master/docker/cion-compose.yml)), or you may follow these instructions.

**Install the docker stack deploy helper**

```
$ pip install git+https://github.com/cionkubes/dsd
```

**Create a compose file.**

You can choose zero or more profiles. For production it is recommended to run without any profiles, for development the `local` and `live` profiles are recommended.

- `local`: This profile exposes all the ports necessary for accessing the cion web interface, the rethinkdb web interface and the catalyst. It also mounts the docker daemon socket so that cion can update services in the swarm in which it runs.
- `live`: Mounts all source code from the host to the containers, to avoid rebuilding the container images on every change. This profile requires that the environment variable `CION_ROOT` is pointing to a directory containing the github repositories [workq](https://github.com/cionkubes/workq) (<https://github.com/cionkubes/workq>), [interface](https://github.com/cionkubes/cion-interface) (<https://github.com/cionkubes/cion-interface>), [rethink-wrapper](https://github.com/cionkubes/rethink-wrapper) (<https://github.com/cionkubes/rethink-wrapper>), [worker](https://github.com/cionkubes/cion-worker) (<https://github.com/cionkubes/cion-worker>), [orchestrator](https://github.com/cionkubes/cion-orchestrator) (<https://github.com/cionkubes/cion-orchestrator>), [web](https://github.com/cionkubes/cion-web) (<https://github.com/cionkubes/cion-web>), [api](https://github.com/cionkubes/cion-api) (<https://github.com/cionkubes/cion-api>) and [catalyst](https://github.com/cionkubes/cion-catalyst) (<https://github.com/cionkubes/cion-catalyst>)
- `expose-rdb`: Exposes the rethink api port 28015 to the host.
- `expose-orchestrator`: Exposes the orchestrator port 8890 to the host so that external workers can connect to it.

```
$ dsd docker/cion-compose.yml --out [profiles...] > my-stack.yml
```

**Modify the compose file**

Open `my-stack.yml` in a text editor, and edit the file to suit your needs. At a bare minimum you need to expose the catalyst and web interface services (the local profile exposes the ports to the host for you). You also need to add all of your docker secrets, except the url token, to the worker container.

You can expose the services through a proxy like [docker flow proxy \(http://proxy.dockerflow.com/swarm-mode-stack/\)](http://proxy.dockerflow.com/swarm-mode-stack/), or, like shown below, you can map the ports to the host. The catalyst needs to be accessible by the image hosting solution, e.g. [dockerhub \(hub.docker.com\)](https://hub.docker.com) or your [docker registry \(https://docs.docker.com/registry/\)](https://docs.docker.com/registry/), while the web service needs to be accessible by the end users. Both services use port 80 internally.

```
services:
  web:
    ports:
      - 80:80
  catalyst:
    ports:
      - 8080:80
```

Adding the secrets simply requires you to add them to the `services.worker.secrets` list like so.

```
services:
  worker:
    secrets:
      - secret1
      - secret2

secrets:
  secret1:
    external: true
  secret2:
    external: true
```

## Starting the stack

Now that we have configured our compose file we can start it up using docker.

```
$ docker stack --compose-file my-stack.yml [stack-name]
```

Eventually you should see the services starting.

```
$ docker service ls
> ID                NAME                MODE                REPLICAS            IMAGE
PORTS
> hio6xfarzscf      cion_api            replicated          1/1                 cion/api:latest
> ylsqt6l7s51b      cion_catalyst       replicated          1/1                 cion/catalyst:latest
*:8080->80/tcp
> el0hdwlbbk6z      cion_orchestrator   replicated          1/1                 cion/orchestrator:latest
> kriljv6mnwzm      cion_rethink        replicated          1/1                 rethinkdb:latest
> v5rzpi9v8fn0      cion_rethink-shard  replicated          1/1                 rethinkdb:latest
> iu3292szxf6u      cion_web            replicated          1/1                 cion/web:latest
*:80->80/tcp
> a2v2ea9uzp27      cion_worker         replicated          3/3                 cion/worker:latest
```

## Next steps

You should now be able to access the cion web interface and [configure \(configure.md\)](#) cion