

[_\(\)](#)

PUBLIC | AUTOMATED BUILD

[opinkerfi \(/u/opinkerfi/\)/adagios-git \(/r/opinkerfi/adagios-git/\)](#)

Last pushed: 3 months ago

[Repo Info \(/r/opinkerfi/adagios-git/\)](#)

Short Description

Adagios + Nagios 4 - built from Git

Full Description

docker-adagios-git

Docker image for Adagios built from git (Nagios 4-Adagios)

docker stars **2** [_\(\)](#)docker pulls **359** [_\(\)](#)tag **none** [_\(\)](#)release **undefined** [_\(\)](#)

Usage

```
docker create \  
  --name=my-adagios \  
  -p 80:80 \  
  opinkerfi/adagios-git
```

Log in with user `thrukadmin` and password `thrukadmin`

Parameters

The parameters are split into two halves, separated by a colon, the left hand side representing the host and the right the container side.

For example with a port `-p external:internal` - what this shows is the port mapping from internal

to external of the container.

So -p 8080:80 would expose port 80 from inside the container to be accessible from the host's IP on port 8080

<http://192.168.x.x:8080> (<http://192.168.x.x:8080>) would show you what's running INSIDE the container on port 80.

- -p 80 - Port for Adagios webui

For shell access whilst the container is running do `docker exec -it my-adagios /bin/bash`.

To access Adagios

:80/adagios

To access Nagios 4 UI

:80/nagios

Local development of Adagios with adagios-git docker image

Via docker run command

```
docker run -it -p 8080:80 \
-v ~/code/adagios:/opt/adagios \
-v ~/code/pynag:/opt/pynag \
-v ~/code/logs:/var/log/nagios \
--name adagios opinkerfi/adagios-git:latest
```

Via docker-compose

```
docker-compose up
```

where the contents of docker-compose.yml is similar to the example provided

```
version: '3.1'

services:
  adagios:
    image: opinkerfi/adagios-git:latest
    ports:
      - 8080:80
    volumes:
      - ./adagios:/opt/adagios
      - ./pynag:/opt/pynag
      - ./logs:/var/log/nagios

volumes:
  adagios:
  pynag:
  logs:
```

Basic development environment setup via shell script

We have created a small script that you can use to setup the local development environment

```
cd ~/my/adagios/env/dev/folder # Folder that you want to keep the code
bash -c "$(curl -L https://raw.githubusercontent.com/opinkerfi/docker-
```

To access your Adagios development container, direct your browser to <http://0.0.0.0:8080> (<http://0.0.0.0:8080>).

Now do some changes with your Adagios or Pynag code inside your local volumes that you created before and restart your container.

```
docker-compose restart
```

When the container restarts it will install Adagios and Pynag from your local volumes. Your new code changes should now reflect on your Adagios development container.

Docker Pull Command



```
docker pull opinkerfi/adagios-git
```

Owner



opinkerfi

Source Repository

🔗 opinkerfi/docker-adagios-git (<https://github.com/opinkerfi/docker-adagios-git>).