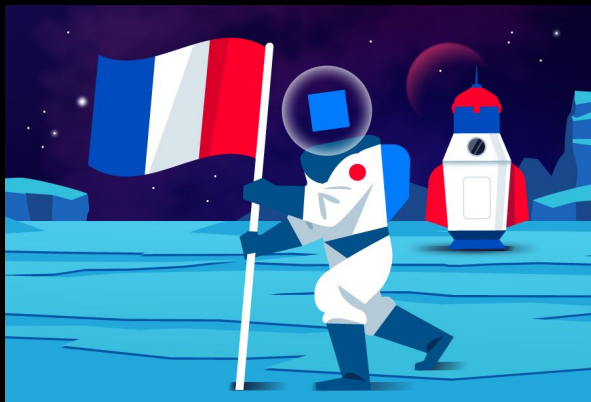




Automatically tag your images with DockerHub hooks!





Julien Landuré



Julien Landuré

CTO Zenika

GDG Cloud & DevFest Nantes Organizer

GDE Cloud

Technical Board of Skillsmatter

 @jlandure

#1 Why?



Back in...2017

Jul 30, 2017 – May 26, 2021

Contributions: Commits ▾

Contributions to master, excluding merge commits and bot accounts





Crafting the perfect container to play with a Headless Chrome



Julien Landuré • May 26 '20





How to measure success?

stars 838

docker build passing

docker pulls 15M

167.8MB 12 layers

all contributors 39

How to measure success?





My background

- Know about CI:
Jenkins & CircleCI
- Official Docker Trainer
- Fan of Open Source



dockerHub

#2 Step by step



Prerequisite:
1/ Zenika Organisation
2/ Upload from my
laptop



Zenika



Community Organization



Zenika



Paris



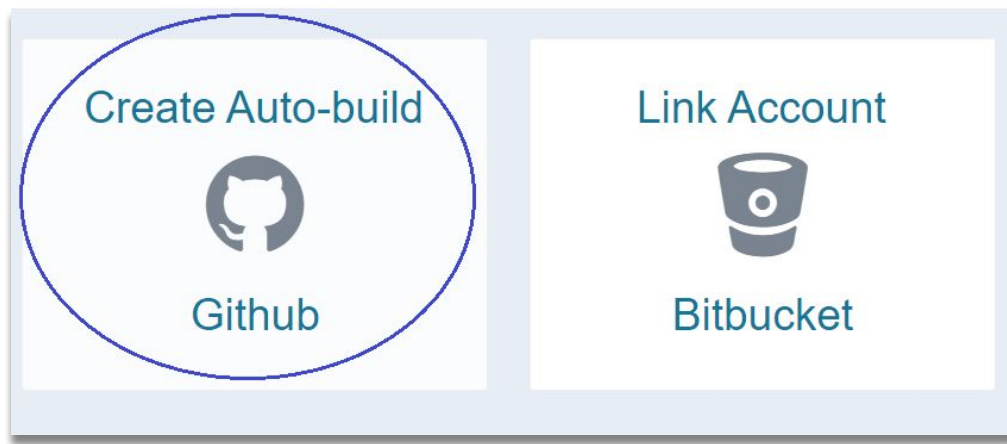
<https://www.zenika.com/>



Joined January 31, 2015



GitHub + DockerHub = AutoBuild ❤️





Microbadger

MicroBadger.com will shutdown on 1st July 2021. More details [here](#)

Manage your Docker container images

e.g. centos or puppet/puppetserver

View image

Shields IO

Shields IO

Pixel-perfect Retina-ready Fast Consistent Hackable No tracking

docker|

Love Shields? Please consider [donating](#) to sustain our activities

Build

Docker Automated build: `docker build` `automated` `/docker/automated/:user/:repo`

Docker Cloud Automated build: `docker build` `automated` `/docker/cloud/automated/:user/:repo`

Docker Cloud Build Status: `docker build` `passing` `/docker/cloud/build/:user/:repo`

Downloads

Docker Pulls: `docker pulls` `765M` `/docker/pulls/:user/:repo`

Size

Docker Image Size (latest by date): `image size` `126 MB` `/docker/image-size/:user/:repo?sort=date`

Docker Image Size (latest semver): `image size` `136 MB` `/docker/image-size/:user/:repo?sort=semver`

Docker Image Size (tag): `image size` `103 MB` `/docker/image-size/:user/:repo/:tag`

MicroBadger Layers: `layers` `15` `/microbadger/layers/:user/:repo`

MicroBadger Layers (tag): `layers` `12` `/microbadger/layers/:user/:repo/:tag`

MicroBadger Size: `image size` `126 MB` `/microbadger/image-size/:user/:repo`

#3 DockerHub hooks



 docker docs

Home

Guides

Product manuals

Reference

Samples

[Home](#) / [Product manuals](#) / [Docker Hub](#) / [Automated builds](#) / [Advanced automated builds](#)

Docker Desktop

Docker Engine

Docker Compose

Docker Hub

- Quickstart
- Docker ID accounts
- Repositories
- Official images
- Automated builds
 - Set up automated builds
 - Testing in automated builds
 - Advanced automated builds**
 - Link to GitHub and BitBucket
- Webhooks

Override build, test or push commands

Docker Hub allows you to override and customize the `build`, `test` and `push` commands during automated build and test processes using hooks. For example, you might use a build hook to set build arguments used only during the build process. (You can also set up [custom build phase hooks](#) to perform actions in between these commands.)

Use these hooks with caution. The contents of these hook files replace the basic `docker` commands, so you must include a similar build, test or push command in the hook or your automated process does not complete.

To override these phases, create a folder called `hooks` in your source code repository at the same directory level as your Dockerfile. Create a file called `hooks/build`, `hooks/test`, or `hooks/push` and include commands that the builder process can execute, such as `docker` and `bash` commands (prefixed appropriately with `#!/bin/bash`).

These hooks will be running on an instance of [Amazon Linux 2](#), a distro based on Ubuntu, which includes interpreters such as Perl and Python and utilities such as `git` or `curl`. Please check the link above for the full list.

Custom build phase hooks

You can run custom commands between phases of the build process by creating hooks. Hooks allow you to provide extra instructions to the autobuild and autotest processes.



```
5
6 LABEL org.label-schema.build-date=$BUILD_DATE \
7     org.label-schema.description="Chrome running in headless mode in a tiny Alpine image" \
8     org.label-schema.name="alpine-chrome" \
9     org.label-schema.schema-version="1.0.0-rc1" \
10    org.label-schema.usage="https://github.com/Zenika/alpine-chrome/blob/master/README.md" \
11    org.label-schema.vcs-url="https://github.com/Zenika/alpine-chrome" \
12    org.label-schema.vcs-ref=$VCS_REF \
13    org.label-schema.vendor="Zenika" \
14    org.label-schema.version="latest"
15
```

🔗 master ▾

[alpine-chrome](#) / [hooks](#) / [build](#)

Go to file

...

Executable File | 4 lines (3 sloc) | 164 Bytes

Raw

Blame



```
1 #!/bin/bash
2 set -e
3
4 docker image build --build-arg VCS_REF=`git rev-parse --short HEAD` --build-arg BUILD_DATE=`date -u +"%Y-%m-%dT%H:%M:%SZ"` -t $IMAGE_NAME .
```



0510c13ba8 ▾

alpine-chrome / hooks / post_push

Go to file

...

Executable File | 13 lines (11 sloc) | 413 Bytes

Raw

Blame



```
1  #!/bin/bash
2  set -e
3
4  CURRENT_VERSION_CHROMIUM=`docker container run --rm --entrypoint "" zenika/alpine-chrome chromium-browser --version`
5  echo ${CURRENT_VERSION_CHROMIUM:9:2}
6
7  for tag in ${CURRENT_VERSION_CHROMIUM:9:2}; do
8      echo $tag
9      tagStart=$(expr index "$IMAGE_NAME" :)
10     repoName=${IMAGE_NAME:0:tagStart-1}
11     docker tag $IMAGE_NAME ${repoName}:${tag}
12     docker push ${repoName}:${tag}
13  done
```



master ▾

alpine-chrome / with-node / hooks / post_push

Executable File | 17 lines (15 sloc) | 737 Bytes

```
1  #!/bin/bash
2  set -e
3
4  CURRENT_VERSION_CHROMIUM=`docker container run --rm --entrypoint "" zenika/alpine-chrome:with-node chromium-browser --version`
5  echo ${CURRENT_VERSION_CHROMIUM:9:2}
6  CURRENT_NODE_VERSION=`docker container run --rm --entrypoint "" zenika/alpine-chrome:with-node node -v`
7  echo ${CURRENT_NODE_VERSION:1:2}
8
9  for tag in ${CURRENT_VERSION_CHROMIUM:9:2}; do
10     echo $tag
11     tagStart=$(expr index "$IMAGE_NAME" :)
12     repoName=${IMAGE_NAME:0:tagStart-1}
13     docker tag $IMAGE_NAME ${repoName}:${tag}-with-node
14     docker push ${repoName}:${tag}-with-node
15     docker tag $IMAGE_NAME ${repoName}:${tag}-with-node-${CURRENT_NODE_VERSION:1:2}
16     docker push ${repoName}:${tag}-with-node-${CURRENT_NODE_VERSION:1:2}
17  done
```

#4 More info



Instance of Amazon Linux 2

> a distro based on Ubuntu (Perl and Python and utilities such as git or curl)

Main hooks

- hooks/build
- hooks/test
- hooks/push

How

The following hooks are available:

- `hooks/post_checkout`
- `hooks/pre_build`
- `hooks/post_build`
- `hooks/pre_test`
- `hooks/post_test`
- `hooks/pre_push` (only used when executing a build rule or [automated build](#))
- `hooks/post_push` (only used when executing a build rule or [automated build](#))

#5 And then...

GCR.IO



```
for published_tag in "latest" $tag; do
  docker tag "$IMAGE_NAME" "$registry/$name:$published_tag"
  docker push "$registry/$name:$published_tag"
  for region in "us" "eu" "asia"; do
    #publish in the different regions
    docker login -u _json_key -p "$(cat key.json)" "https://$region.$registry"
    docker tag "$registry/$name" "$region.$registry/$name:$published_tag"
    docker push "$region.$registry/$name:$published_tag"
  done
done
```

Ideas

Fill the README automatically

Announce a new version on Twitter



Next?

master [alpine-chrome](#) / [hooks](#) / test

Go to file

...

Executable File | 10 lines (7 sloc) | 251 Bytes

Raw

Blame



```
1  #!/bin/bash
2  set -e
3
4  docker image build -t $IMAGE_NAME .
5
6  # Alpine version
7  docker container run --rm --entrypoint "" $IMAGE_NAME cat /etc/alpine-release
8
9  # Chrome version
10 docker container run --rm --entrypoint "" $IMAGE_NAME chromium-browser --version
```

Cons

Lack of documentation

Difficult to debug

Not a real CI/CD system

Pros

Transparent `AutoBuild`

Quality `Autotest`

Auto Updates `FROM`

To sum up

**# Don't spend time
building docker images**

**# Focus on Open Source
value: your project!**

Thanks



Julien Landuré

@jlandure

Zenika/**alpine-chrome**



Chrome Headless docker images built upon alpine official image

19

Contributors

21

Issues

833

Stars

152

Forks



  [.to/alpine-chrome](https://hub.docker.com/r/zenika/alpine-chrome)

Questions?