

How to build a FreeBSD CI/CD environment based on pot containers

Luca Pizzamiglio
pizzamig@FreeBSD.org
FOSDEM 2019

whoami(1)

- **Luca Pizzamiglio aka pizzamig@**
- **FreeBSD enthusiast**
- **Port committer since August 2017**
- **Building packages at trivago**

Motivations

CI/CD is a well established best practices in any modern software development process

Growing interest on build/test software on FreeBSD

- Improve portability

- Provide FreeBSD support

- Provide artifacts for FreeBSD

Lack of FreeBSD support on all major CI systems

- Exception for Cirrus CI

 - VM based support to FreeBSD

Ugly workarounds

rust-lang / libc

Watch 27

Star 455

Fork 356

Code

Issues 49

Pull requests 15

Projects 0

Insights

Tree: b46b6e2e72 - libc / ci / docker / x86_64-unknown-freebsd / Dockerfile

Find file

Copy path

wezm Update FreeBSD docker CI to use FreeBSD 11.1 image

c1fa4b6 on Mar 15, 2018

3 contributors

14 lines (10 sloc) | 448 Bytes

Raw

Blame

History



```
1 FROM wezm/port-prebuilt-freebsd11@sha256:43553e2265ec702ec72a63a765df333f50b1858b896e69385749e96d8624e9b0
2
3 RUN apt-get update
4 RUN apt-get install -y --no-install-recommends \
5     qemu genext2fs xz-utils
6 RUN apt-get install -y curl ca-certificates gcc
7
8 ENTRYPOINT ["sh"]
9
10 ENV PATH=$PATH:/rust/bin \
11     QEMU=2018-03-15/FreeBSD-11.1-RELEASE-amd64.qcow2.xz \
12     CAN_CROSS=1 \
13     CARGO_TARGET_X86_64_UNKNOWN_FREEBSD_LINKER=x86_64-unknown-freebsd11-gcc
```

My motivation 1/2

The screenshot shows the GitHub interface for the repository `pizzamig / ci-test`, which is marked as `Private`. At the top right, there are buttons for `Unwatch` (1), `Star` (0), and `Fork` (0). Below this is a navigation bar with `Code` (selected), `Issues` (0), `Pull requests` (0), `Actions`, `Projects` (0), `Wiki`, `Insights`, and `Settings`. The main content area has `Releases` (selected) and `Tags` tabs, along with `Edit release` and `Delete` buttons. The current release is `Beta`, labeled as the `Latest release`. It was released by `pizzamig` and has a version number of `0.3` with commit hash `7b80ed9`. Under the `Assets` section, there are two items: `Source code (zip)` and `Source code (tar.gz)`. The release description contains the text `beta beta!`.

My Motivation 2/2

The screenshot shows a GitHub repository page for `pizzamig / ci-test`, which is marked as Private. The repository has 1 Unwatch, 0 Stars, and 0 Forks. The navigation bar includes links for Code, Issues (0), Pull requests (0), Actions, Projects (0), Wiki, Insights, and Settings. The 'Releases' tab is active, showing a release named 'Beta' by pizzamig, released 5 minutes ago. The release version is 0.3 with commit hash 7b80ed9. There are 4 assets: 'FreeBSD-11.2-ci-test.tar.gz' (619 KB), 'FreeBSD-12.0-ci-test.tar.gz' (696 KB), 'Source code (zip)', and 'Source code (tar.gz)'. The release description is 'beta beta!'.

`pizzamig / ci-test` Private

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Actions Projects 0 Wiki Insights Settings

Releases Tags Edit release Delete

Latest release

0.3
7b80ed9

pizzamig released this 5 minutes ago

Assets 4

- FreeBSD-11.2-ci-test.tar.gz 619 KB
- FreeBSD-12.0-ci-test.tar.gz 696 KB
- Source code (zip)
- Source code (tar.gz)

beta beta!

Then, what?

<https://github.com/pizzamig/freebsd-ci>

A command line tool to locally run a build process on FreeBSD

Written in Rust

Still a lot limitations:

Supports projects stored in github

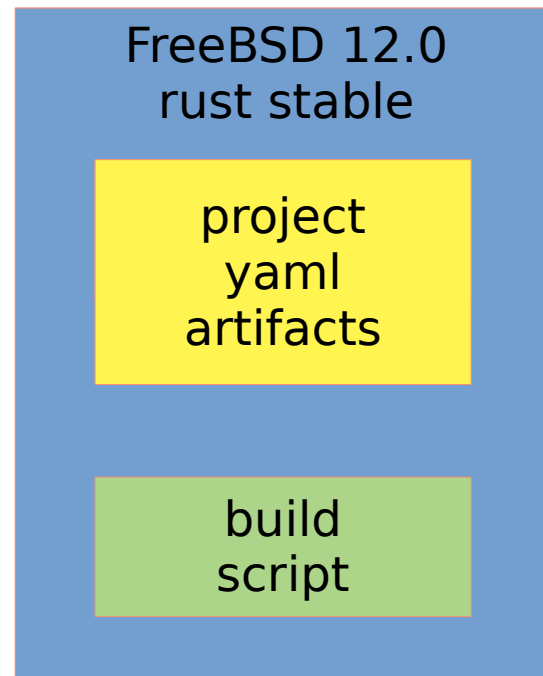
Supports rust projects

```
freebsd-ci -U username -P project [-T 1.0]
```

FreeBSD-ci

- 1. Download a github project and put it in a ZFS dataset**
- 2. It parses the yaml file with instructions**
- 3. From the images catalog, it clones the appropriate image and attach the ZFS dataset**
- 4. It generates the build script**
- 5. Run the build**
- 6. Destroy the image**
- 7. Revert the ZFS dataset and go back to point 3.**

FreeBSD-ci



FreeBSD 11.2
rust stable

FreeBSD 11.2
rust beta

FreeBSD 11.2
rust nightly

FreeBSD 12.0
rust stable

FreeBSD 12.0
rust beta

FreeBSD 12.0
rust nightly

The yaml file

```
os: FreeBSD                # the operating system
                           # FreeBSD is the only one supported
FreeBSD:                   # The FreeBSD version to use to build
  - '11.2'
  - '12.0'

update: true               # Run an update of the image before to build
                           # Update packages, toolchain, and so on

language: rust             # The project language

rust:                      # Which rust variant use to build
  - stable
  - nightly

no_deploy:                 # Which combination shouldn't be deployed
  rust:
    - nightly
```

The build script template

```
#!/bin/sh
export HOME=/root
export PATH=/sbin:/bin:/usr/sbin:/usr/bin
PATH=$PATH:/usr/local/sbin:/usr/local/bin
PATH=/root/.cargo/bin:$PATH

if {{ update }} ; then
    rustup update
    pkg upgrade -y
fi

cd /mnt
if ! cargo clippy --release ; then
    exit 1
fi
if ! cargo build --release ; then
    exit 1
fi
if ! cargo test --release ; then
    exit 1
fi

if {{ upload }} ; then
    cargo install --path . -f
    tgt_dir="{{ os_family }}-{{ os_version }}-{{ project }}"
    tarball="{{ tarball }}"
    mkdir $tgt_dir
    mv $HOME/.cargo/bin/{{ project }} $tgt_dir
    tar zcf ${tarball} $tgt_dir
    if {{ delete_asset }} ; then
        curl -H "Authorization: bearer {{ token }}" \
            -X DELETE \
                https://api.github.com/repos/{{user}}/
                {{project}}/releases/assets/{{asset_id}}
    fi
    curl -H "Authorization: bearer {{ token }}" \
        -H "Content-Type: application/gzip" \
        -X POST \
            --data-binary @${tarball} \
                https://uploads.github.com/repos/{{user}}/
                {{project}}/releases/{{release_id}}/assets?name=${tarball}
    fi
exit 0
```

Images catalog: the challenge

We provide some pot flavors to generate images

```
#!/bin/sh
```

```
[ -w /etc/pkg/FreeBSD.conf ] && sed -i '' 's/quarterly/latest/' /etc/pkg/FreeBSD.conf  
ASSUME_ALWAYS_YES=yes pkg bootstrap  
touch /etc/rc.conf  
sysrc sendmail_enable="NONE"
```

```
pkg install -y ca_root_nss curl
```

```
fetch -o /root/rustup.sh https://sh.rustup.rs  
sh /root/rustup.sh -y --default-toolchain stable  
export PATH="$HOME/.cargo/bin:$PATH"  
echo setenv PATH $HOME/.cargo/bin:'$PATH' >> $HOME/.cshrc
```

```
rustup component add clippy-preview  
rustup component add rustfmt
```

```
pkg clean -aqy
```

Next steps

Extend support

More languages

More platforms

Better logging

Remote image catalog

A system to download images from a remote catalog

Adopt an orchestration framework

Nomad is a good candidate

Thanks!

Thanks a lot!

Questions?