# BUILDING A CHE STACK

The art of...

# An Example

# The SPECS

Languages support:

- Python
- JavaScript

Runtime:

- Front: Angular
- Back: Python Service
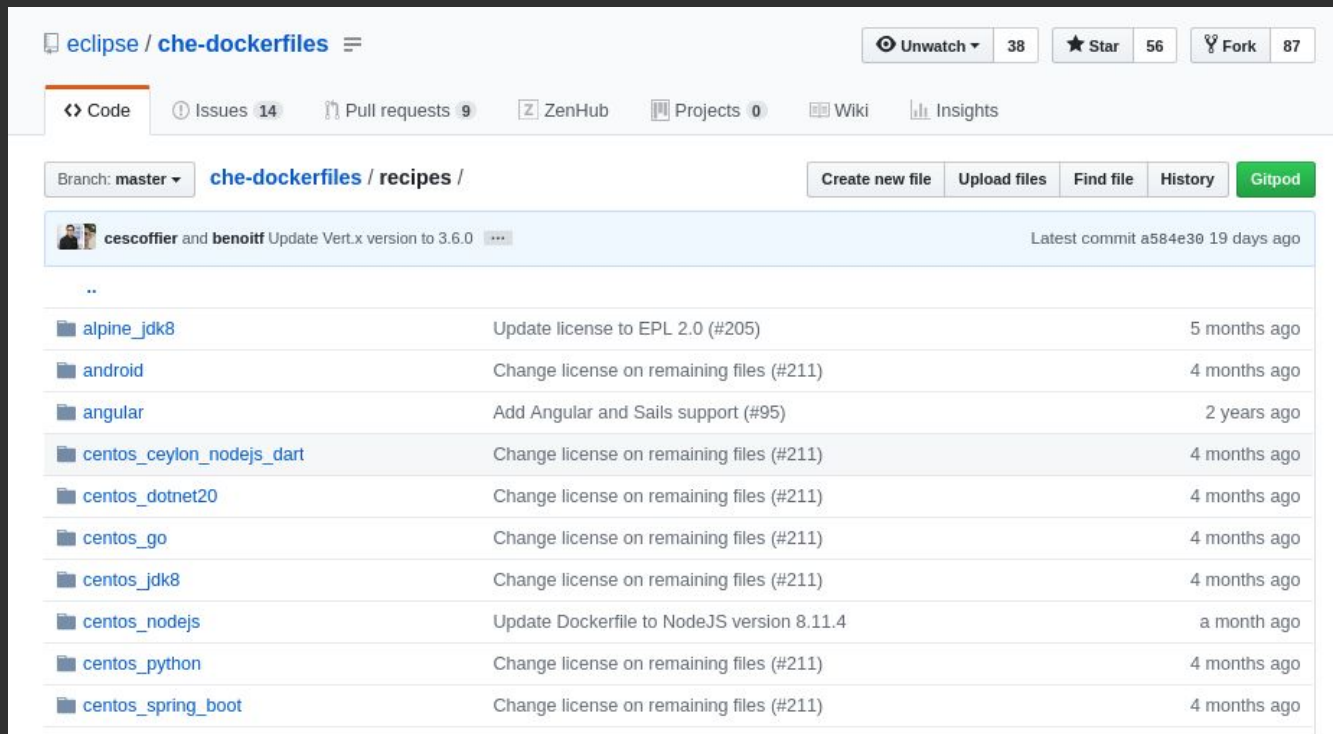
DataBase:

- PostgreSQL

Source Code:

Frontend: gothinkster/angular-realworld-example-app

Backend: gothinkster/flask-realworld-example-app

RED HAT
DEVELOPER

# STEPS

Choose a base image for the "DEV" container
Customize the "DEV" container base image
Select an existing Che stack
Customize the Che stack

RED HAT
DEVELOPER

# STEP 1 - Choose a base image for the "DEV" container

# STEP 1 - Choose a base image for the "DEV" container

📁 angular       Add Angular and Sails support (#95)       2 years ago

📁 centos_nodejs       Update Dockerfile to NodeJS version 8.11.4       a month ago

📁 centos_python       Change license on remaining files (#211)       4 months ago

📁 stack-base       Change license on remaining files (#211)       4 months ago

RED HAT
DEVELOPER

# STEP 1 - Choose a base image for the "DEV" container

| | | |
|---|---|---|
| 📁 angular | Add Angular and Sails support (#95) | 2 years ago |
| 📁 centos_nodejs | Update Dockerfile to NodeJS version 8.11.4 | a month ago |
| 📁 centos_python | Change license on remaining files (#211) | 4 months ago |
| 📁 stack-base | Change license on remaining files (#211) | 4 months ago |

# STEP 2 - Customize the image

```
1    FROM registry.centos.org/che-stacks/centos-nodejs
2
3    USER root
4    RUN npm install -g @angular/cli
5    USER user
6    |
```

```
$ docker build -t mariolet/che-stack-angular-python .
(...)
Successfully built ae9cb1b9a415
$ docker push mariolet/che-stack-angular-python
(...)
c1533a28b1a9: Pushed
```

# STEP 3 - Select an existing Che stack

| NAME | DESCRIPTION | COMPONENTS | ACTIONS |
|---|---|---|---|
| .NET | Default .NET 2.0.0 Stack with .NET Core SDK | Ubuntu, Dotnet | 🗑 ⧉ |
| Android | Default Android Stack with Java 1.8 and Android SDK | Centos, JDK, Maven, Android API | 🗑 ⧉ |
| Apache Camel based projects | Stack with environment ready to develop Integration projects with Apache Camel ... | CentOS, OpenJDK, RH-Maven | 🗑 ⧉ |
| Blank | Default Blank Stack. | Ubuntu, JDK, Maven, Tomcat | 🗑 ⧉ |
| C++ | Default C++ Stack with C++, gcc 4.8.4, GNU Make 3.81. | Ubuntu, JDK, G++, GCC, Make | 🗑 ⧉ |
| CentOS blank | CentOS Blank Stack | Centos, OpenJDK | 🗑 ⧉ |
| CentOS Go | CentOS based Go Stack | Centos, Golang | 🗑 ⧉ |
| CentOS nodejs | CentOS Node Stack | CentOS, OpenJDK, RH-Maven, NodeJS, NPM | 🗑 ⧉ |
| CentOS WildFly Swarm | Eclipse WildFly Swarm Stack on CentOS. | CentOS, OpenJDK, RH-Maven | 🗑 ⧉ |
| Ceylon with Java JavaScript Dart on CentOS | Ceylon stack with Java, JavaScript and Dart backends on CentOS. | Centos, Ceylon | 🗑 ⧉ |
| Che 7 | Workspace.next sidecars and Theia as IDE | Centos, OpenJDK, NodeJS, NPM | 🗑 ⧉ |
| Che 7 Dev | Workspace.next sidecars and Theia as IDE with the tooling for plugin development | Centos, OpenJDK, NodeJS, NPM | 🗑 ⧉ |
| Eclipse Che | Utilities to build Che in Che with JDK 8 and Maven. | JDK, Maven | 🗑 ⧉ |
| Eclipse Vert.x | Eclipse Vert.x Stack on CentOS with OpenJDK and Maven | CentOS, OpenJDK, RH-Maven | 🗑 ⧉ |
| Go | Default Go Stack with Go 1.10.2 | Ubuntu, Go | 🗑 ⧉ |
| Java | Default Java Stack with JDK 8, Maven and Tomcat. | Ubuntu, JDK, Maven, Tomcat | 🗑 ⧉ |
| Java and MySQL with Theia IDE on Kubernetes | Multi Container Workspace - Java with MySQL database in Theia IDE | | 🗑 ⧉ |

RED HAT DEVELOPER

# STEP 3 - Select an existing Che stack

| Nodejs and Postgres Stack | Nodejs stack with Postgres | Node.js, NPM, Postgres | |

RED HAT DEVELOPER

# STEP 4 - Customize the Che stack

- Duplicate the original stack
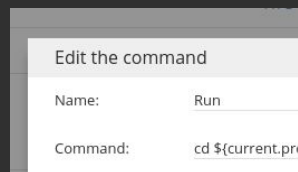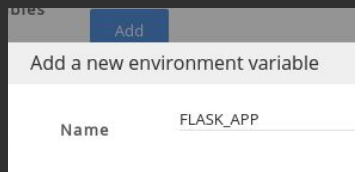- Modify the original recipe



```
containers:
 -
   image: quay.io/openshiftio/che-centos-nodejs
   name: nodejs
```

```
containers:
 -
   image: mariolet/che-stack-angular-python
   name: nodejs
```

- Add language support (activate agents)



| | Python language server | 1.0.4 | Python intellisense |
| | TypeScript language server | 1.0.1 | TypeScript intellisense |

- And other configurations...



Add

Add a new environment variable

Name        FLASK_APP

Edit the command

Name:      Run

Command:   cd ${current.pro

RED HAT
DEVELOPER

# STEP 5 - Create the Workspace

# When you want to start from scratch

# "DEV" Container Prerequisites

- Packages to install:
    - Always: JDK 8, Git, Wget, cURL, sudo, unzip
    - Based on the agents activated in the Workspace:

| Workspace API | Java LS | TypeScript LS | Python LS | PHP LS | GO LS |
|---|---|---|---|---|---|
| JDK 8 | - JDK<br>- maven | - NodeJS | - Python 3.6<br>- pip | | |

- The entrypoint should be non terminating
- Ready to run as an arbitrary user
    - Set the right folder permissions
    - Dynamically patch /etc/passwd and /etc/groups when the container starts
- In general have a look at the base stacks

# Gotchas - Exposing a Service

# Gotchas - Memory Tweaking

-

# Gotchas - Volumes

-