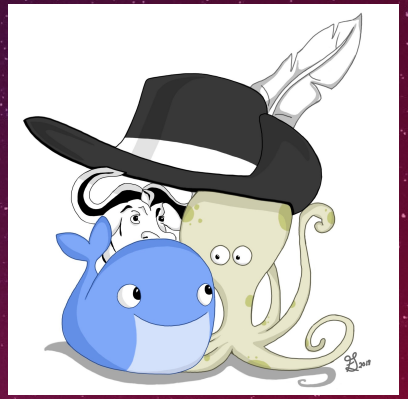
CONTINO

3 Musketeers



Caio Trevisan - Cloud Lead

https://github.com/caiocezart/3musketeers

https://3musketeers.io

3 Musketeers

Test, build, and deploy your apps from anywhere, the same way.

Get Started →

Consistency

Run the same commands no matter where you are: Linux, MacOS, Windows, CI/CD tools that supports Docker like GitHub Actions, Travis CI, CircleCI, and GitLab CI.

Control

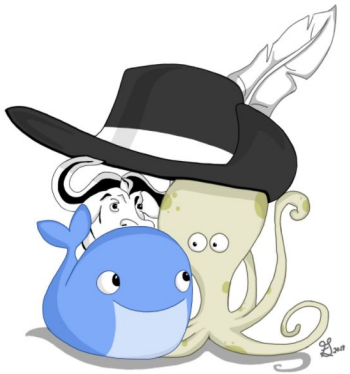
Take control of languages, versions, and tools you need, and version source control your pipelines with your preferred VCS like GitHub and GitLab.

Confidence

Test your code and pipelines locally before your CI/CD tool runs it. Feel confident that if it works locally, it will work in your CI/CD server.

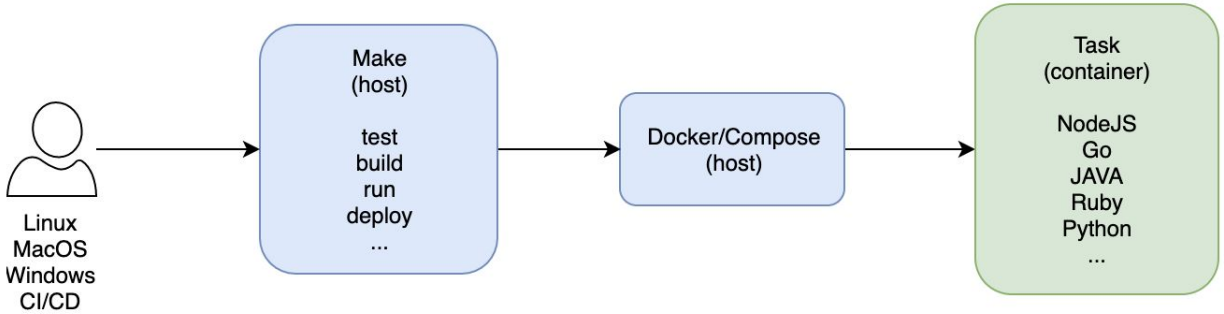
What is 3 Musketeers (3M)?

- It's not a tool!



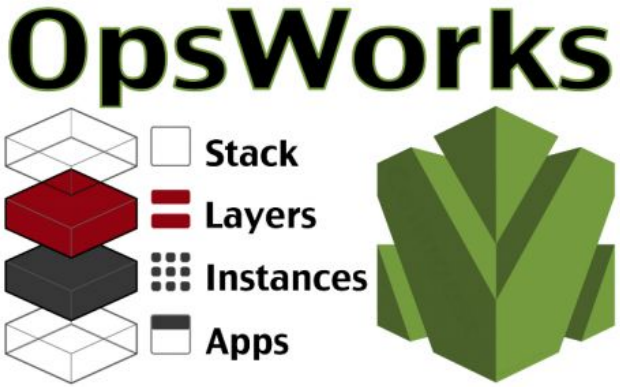
docker + docker-compose + make

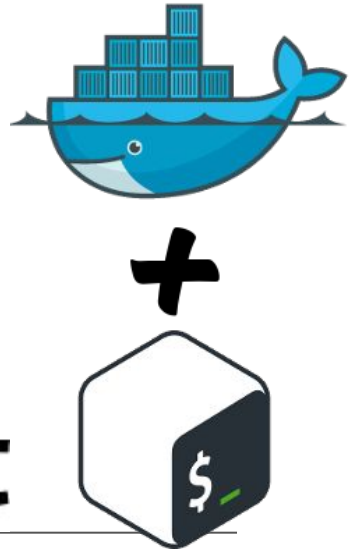
An approach/strategy for organising code in a way that can be easily reproducible in any environment



Other Alternatives

- VMs - too bulky, tackling problem at a desktop level
- Vagrant - tackle problem at a application level but way too bulky for app level
- 2 Musketeers https://2musketeers.sh/ - tackles problem at app level, but contains 100+ lines of bash maintained by one guy. Also can't assume everyone has bash
- **NO musketeers -** install node 4 but definitely not version 3, what 5? But I'm on windows. Can't find anything on this 20 page docs





Make

- Commonly used to compile and build applications from its source code
- Runs sequence of commands (like a recipe) based on targets
- Ability to run in multiple platforms and shells (bash/zsh, linux/macos)

simple target

```
caio@yoda make % cat Makefile
hello:
	echo Hello world
caio@yoda make % make hello
echo Hello world
Hello world
caio@yoda make %
```

variables

```
caio@yoda make % cat Makefile
NAME ?= caio

hello:
	echo Hello $(NAME)
caio@yoda make % make hello
echo Hello caio
Hello caio
caio@yoda make % NAME=trevisan make hello
echo Hello trevisan
Hello trevisan
caio@yoda make %
```

multiple targets / phony

```
caio@yoda make % ls
Makefile
caio@yoda make % cat Makefile
LOG_FILE ?= output.log

.PHONY: file_exists
file_exists:
	test ! -f $(LOG_FILE) && touch $(LOG_FILE)

.PHONY: log
log: start_log file_exists
	echo "error" > $(LOG_FILE)

.PHONY: start_log
start_log:
	echo "Start logging"
caio@yoda make % make log
echo "Start logging"
Start logging
test ! -f output.log && touch output.log
echo "error" > output.log
caio@yoda make % ls
Makefile	output.log
caio@yoda make % cat output.log
error
```

Docker

- Able to run multiple tooling versions without worrying with dependencies
- Same code works across platforms
- Combined with Make abstracts the need of explaining steps to reproduce automations
    - I.e: make build / make deploy / make run

wordpress + mariadb locally

```
# run mariadb
docker run --name db \
    -e MYSQL_ROOT_PASSWORD=example \
    -d mariadb
```

```
# run wordpress
docker run --name wordpress \
    --link db:mysql \
    -p 8080:80 \
    -d wordpress
```

multiple cli tools versions

```
# terraform
docker run --rm -it hashicorp/terraform:0.12.24 version
docker run --rm -it hashicorp/terraform:0.11.14 version
```

docker-compose (Compose)

- Docker commands can be verbose (see example below)
- Declarative (via code) rather than imperative (run commands)
- Reduce margin for human error
- Can version control updates on commands

```
docker run -d -t -i -e REDIS_NAMESPACE='staging' \
    -e POSTGRES_ENV_POSTGRES_PASSWORD='foo' \
    -e POSTGRES_ENV_POSTGRES_USER='bar' \
    -e POSTGRES_ENV_DB_NAME='mysite_staging' \
    -e POSTGRES_PORT_5432_TCP_ADDR='example.amazonaws.com' \
    -e SITE_URL='staging.mysite.com' \
    -v ${PWD}/:/work \
    -p 80:80 \
    --link redis:redis \
    --name container_name
    dockerhub_id/image_name
```

```
version: '3.1'

services:

  wordpress:
    image: wordpress
    restart: always
    ports:
      - 8080:80
    environment:
      WORDPRESS_DB_HOST: db
      WORDPRESS_DB_USER: exampleuser
      WORDPRESS_DB_PASSWORD: examplepass
      WORDPRESS_DB_NAME: exampledb
    volumes:
      - wordpress:/var/www/html

  db:
    image: mysql:5.7
    restart: always
    environment:
      MYSQL_DATABASE: exampledb
      MYSQL_USER: exampleuser
      MYSQL_PASSWORD: examplepass
      MYSQL_RANDOM_ROOT_PASSWORD: '1'
    volumes:
      - db:/var/lib/mysql

volumes:
  wordpress:
  db:
```

Putting all together

```
TAG ?= 1.0.0
REPOSITORY ?= caiocezart

.SILENT:

.PHONY: build
build:
    docker-compose build app
    docker tag app:latest $(REPOSITORY)/app:$(TAG)

.PHONY: push
push:
    docker push $(REPOSITORY)/app:$(TAG)

.PHONY: deploy
deploy: build push

.PHONY: run
run:
    docker-compose up
```

Resources

3Musketeers sources

- https://3musketeers.io/
- https://github.com/flemay/3musketeers

DevOps Academy

- https://github.com/devopsacademyau/academy/tree/master/classes/05class

Make

- https://www.gnu.org/software/make/manual/html_node/index.html#toc-Overview-of-make

Docker

- https://docs.docker.com/

Docker-compose

- https://docs.docker.com/compose/


Questions