

# Quibble

## A testrunner for MediaWiki

Antoine « Hashar » Musso  
Barcelona Hackathon  
May 2018



**WIKIMEDIA**  
FOUNDATION

**But it works for me!**



**WIKIMEDIA**  
FOUNDATION

# CI failures are ~~hard~~ impossible to reproduce

mwext-testextension-php7.0 FAILURE



WIKIMEDIA  
FOUNDATION

# Too many requirements



WIKIMEDIA  
FOUNDATION

# 1) Environment

Debian Jessie/Stretch

Wikimedia patched up **HHVM/PHP** packages

Custom HHVM/PHP extensions

Old composer/npm versions

NodeJs 6

Chromium

Chromedriver

Firefox

MariaDB/MySQL 5.x (no utf8mb4)



# 2) Source / config

mediawiki/vendor which requires installing dev dependencies

Extensions and skins dependencies (and git submodules)

zuul-cloner for branch checkout

LocalSettings.php tweaks

Depends-On

Remember patch under test is merged against tip of branch before being tested!

# 3) Test commands

composer test

npm test

php tests/phpunit.php --exclude-groups Broken --debug-tests

Run chromedriver then:

npm run qunit

npm run selenium-test

....

# Examples



**WIKIMEDIA**  
FOUNDATION



# Test with vendor.git

Misses phpunit hence we need to:

- Read vendor/package.json
- composer require the dev dependencies



# composer & extensions

- Glob extensions/\*/composer.json
- Generate /composer.local.json
- composer merge plugin



Lets factor out

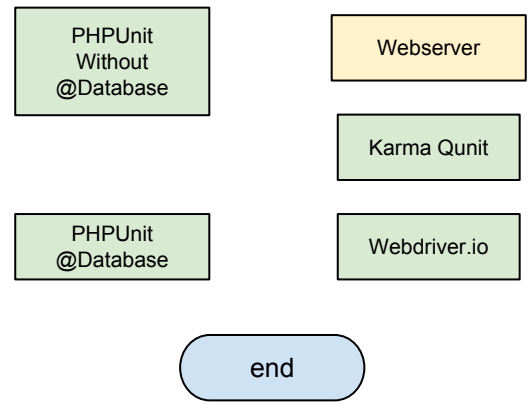
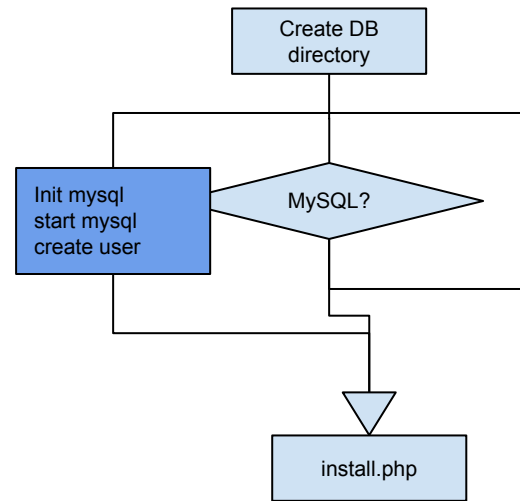
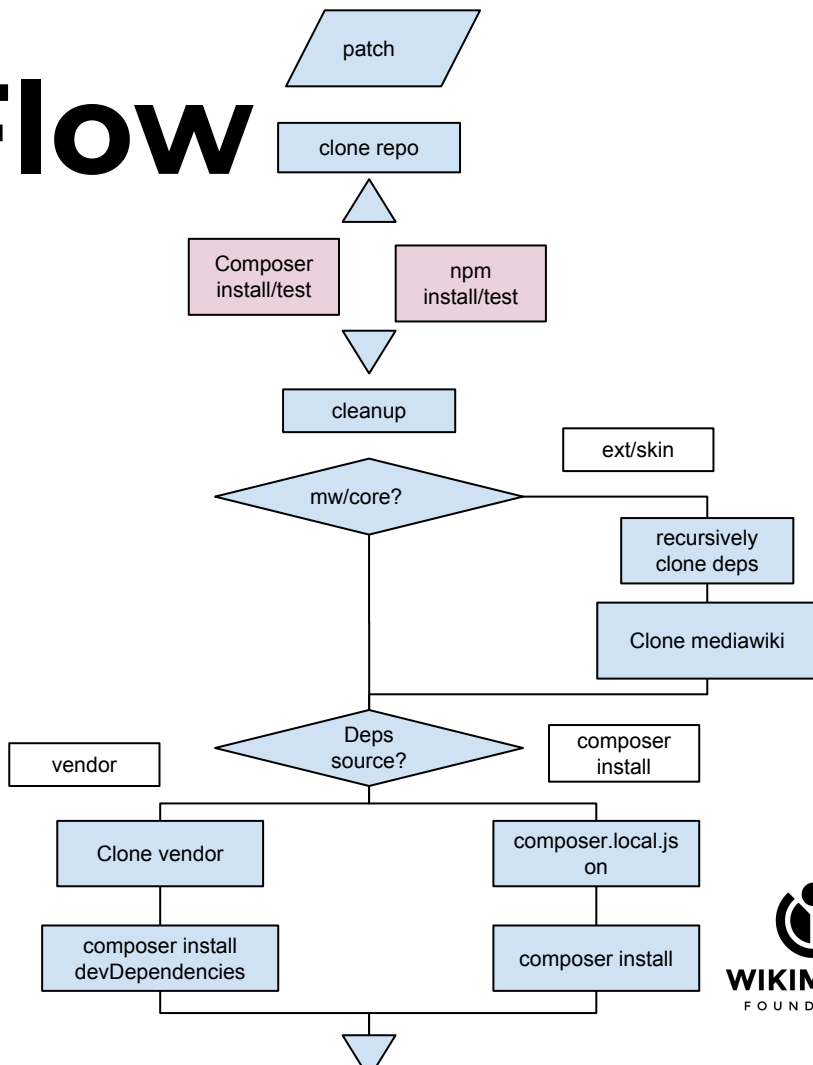


**WIKIMEDIA**  
FOUNDATION

# Before

<b>Jenkins job</b>	Commands to run in Jenkins In integration/config.git ./jjob/
<b>Shell scripts</b>	Helpers In integration/jenkins.git
<b>Zuul cloner</b>	Clone repo, fetch patch, checkout branches
<b>LocalSettings tweaks</b>	integration/jenkins.git (now in mediawiki/core DevSettings.php)
<b>Parameters tweaks</b>	Injected by Zuul In integration/config.git ./zuul/

# Flow



# Quibble Flows!

- Clone sources
- Get dependencies
- Install MediaWiki
- Start backends
- Run test
- Profit!

¥ € \$

# Minimal use case

```
$ pip install git+https://gerrit.wikimedia.org/r/p/integration/quibble  
$ quibble
```

Clones mediawiki/core, set it up, starts mysql / webserver, run tests

# Extra repo

Add an extension to the loop:

```
$ quibble mediawiki/extensions/BoilerPlate
```



# Extra repo + deps

Add an extension to the loop:

```
$ quibble mediawiki/extensions/CodeEditor \  
mediawiki/extensions/WikiEditor
```

*Dependencies will be extracted automatically later on*

# Skip clone/deps

```
$ quibble --skip-zuul --skip-deps
```

Assumes ./src/ has the repos and the composer or vendor dependencies.

WAY faster when iterating

# Moaar

\$ quibble --help

<https://doc.wikimedia.org/quibble/usage.html>

# Environment



**WIKIMEDIA**  
FOUNDATION

# Up to you

Quibble does not setup your environment (php/mysql/composer/npm)

Unix philosophy is to restrict to doing a single thing but doing it right

Quibble just glue commands to runs. That is all.

# But Docker!



# But Docker!

#1 Docker is terrible

# But Docker!

#1 Docker is terrible

#2 read my book “**Docker the bad parts**”



# Docker (1/3)

It is just a tarball of a file system, a snapshot of app + dependencies:

- Libraries ( libc libregex ... )
- Commands ( php , chromium )

# Docker (2/3)

- chroot on steroid
- Or think of a process running with different libraries and isolated from the rest of your host
- Somehow safe

# Docker (3/3)

- integration/quibble has an example Dockerfile
- CI uses images from integration/config dockerfiles:
  - [docker-registry.wikimedia.org/releeng/quibble-jessie-php55](https://docker-registry.wikimedia.org/releeng/quibble-jessie-php55)
  - [docker-registry.wikimedia.org/releeng/quibble-stretch](https://docker-registry.wikimedia.org/releeng/quibble-stretch)
  - [docker-registry.wikimedia.org/releeng/quibble-jessie-hhvm](https://docker-registry.wikimedia.org/releeng/quibble-jessie-hhvm)

# Run it

```
docker pull docker-registry.wikimedia.org/releeng/quibble-stretch
```

```
docker run -it --rm docker-registry.wikimedia.org/releeng/quibble-stretch
```

Equivalent to having a Debian Stretch locally with the proper packages installed and running:

```
$ quibble
```

# Reuse local code

Mount a local directory inside the container as `/workspace`:

```
docker run -it --rm \  
  -v "$pwd"/workspace:/workspace \  
  docker-registry.wikimedia.org/releeng/quibble-stretch
```

Quible runs git commands which populate `./workspace/src` on your host machine.

# Reuse local code

Next run: instruct Quibble to skip git operations (`--skip-zuul`)

while still mounting the workspace:

```
docker run -it --rm \
```

```
-v "$pwd"/workspace:/workspace \
```

```
docker-registry.wikimedia.org/releeng/quibble-stretch \
```

```
--skip-zuul
```

**80% speed up  
Guarantee  
As seen on TV \***



\*

Lorem ipsum dolor sit amet, movet pertinax molestiae te qui, id vel omnes nostrud accumsan, pri ut enim natum ullam. Maiorum fuisset mea ad. Eruditi vituperata cum ea, est ex detracto contentiones. Convenire signiferumque usu an, et molestie deserunt sea, duo atqui imperdiet ut. Singulis scribentur at quo, est epicuri detraxit eu, atqui scaevola iudicabit sea in. In duo tritani antiopam. Yeah that is a boring legal statemet nobody ever read. You have won a beer or tow. Lorem ipsum.

# Reuse cache

Mount a local directory inside the container as /cache:

```
mkdir cache && chmod 777 cache
```

```
docker run -it --rm \
```

```
  -v "$pwd"/cache:/cache \
```

```
  docker-registry.wikimedia.org/releng/quibble-stretch
```

composer / npm writes downloaded packages to /cache which is then kept being run. Magic!

# Moar tricks

<https://doc.wikimedia.org/quibble/>

Doc requests in Phabricator are much welcome. #quibble



# Usage on CI



**WIKIMEDIA**  
FOUNDATION

# Parameters

Zuul crafts a patch / reference for each repositories and pass:

- ZUUL\_URL ZUUL\_BRANCH ZUUL\_COMMIT etc

Zuul holds extensions/skins dependencies tree (TO BE FIXED).

# Combo time

<b>PHP</b>	5.5	7.0	HHVM
<b>Source</b>	vendor.git	composer.install	
<b>Database</b>	SQLite	MySQL	PostgreSQL

# Combo used

mediawiki/core

- 1) **composer** + mysql + php7.0
- 2) vendor + mysql + **hhvm**
- 3) vendor + mysql + php7.0

(tests break with sqlite)

# Written in Python

Python language makes it easy

Built-in modules for logging / spawning process ...

No other requirements beside the usual python modules

CI maintainers are all into python to some degree

Straight forward to read



WIKIMEDIA  
FOUNDATION

# Refs

Source: <https://gerrit.wikimedia.org/g/integration/quibble>

Clone: <https://gerrit.wikimedia.org/r/p/integration/quibble>

Doc: <https://doc.wikimedia.org/quibble/>

Phab: <https://phabricator.wikimedia.org/project/profile/2772/>

<https://docs.google.com/presentation/d/1V6b3aUkAM0FS605bYOQB9iyQroMYcKhzSZXJUStxIhQ/edit?usp=sharing>