

super_unkoを
スーパーキレイにした

発表者：次郎

目次

- 概要
- 自己紹介
- 経緯
- やったこと
 - 静的解析 (shellcheck)
 - フォーマッタ (shfmt)
 - テストフレームワーク (bats)
 - テスト・CI環境の整備 (docker)
- まとめ
- 謝辞

概要

今日話すこと

概要

super_unkoを
スーパーキレイにし
た

概要

そもそも

super_unkoとは

概要

- ぐれさんさんにより発足したシェルスクリプトだけで書かれたコマンド郡、およびそのリポジトリ
- シェル芸勉強会中に爆誕したsuper_unkoリポジトリを巡る悲喜交々 - Togetter
- いろんなシェル芸人のシェルが詰まった宝庫(肥溜め)
- リポジトリ: https://github.com/uncontributors/super_unko

概要

シェル芸botでおなじみの
こういうのがいっぱいある



概要

そんなsuper_unkoを
スーパーキレイ
にした

概要

スーパーキレイ

概要

スーパーキレイ

→とは ???

概要

スーパーキレイ

概要

スーパーキレイ

→コード品質の向上

概要

super_unkoを
スーパーキレイにし
た

概要

言い換えると


概要

シェルスクリプトのみの
OSSのコード品質を
向上させるために何をしたか

概要

シェルスクリプトのみの
OSSのコード品質を
向上させるために何をしたか
について話す

目次

-  概要
- 自己紹介
- 経緯
- やったこと
 - 静的解析 (shellcheck)
 - フォーマッタ (shfmt)
 - テストフレームワーク (bats)
 - テスト・CI環境の整備 (docker)
- まとめ
- 謝辞

10秒で終わる自己紹介

発表者：次郎



10秒で終わる自己紹介

発表者：次郎

サーバサイド

エンジニア (Go)



目次

- ✓ 概要
- ✓ 自己紹介
- 経緯
- やったこと
 - 静的解析 (shellcheck)
 - フォーマッタ (shfmt)
 - テストフレームワーク (bats)
 - テスト・CI環境の整備 (docker)
- まとめ
- 謝辞

経緯

スーパーキレイに
しようと思った経緯

経緯



super_unkoにコミット
をしていた
ある日

経緯

READMEの 一文が気になった

経緯

super_unko

license  build 



super_unko project is the one of the awesome, clean and sophisticated OSS project in the world. Let's create shit commands!

super_unko プロジェクトは世界で最もクリーンで洗練されたOSSプロジェクトの一つです。 うんこなコマンドを作りましょう。

Command	Description
---------	-------------

経緯

super_unko

license  build 



super_unko project is the one of the awesome, clean and sophisticated OSS project in the world. Let's create shit commands!

super_unko プロジェクトは世界で最もクリーンで洗練されたOSSプロジェクトの一つです。 うんこなコマンドを作りましょう。

Command	Description
---------	-------------

経緯

super_unko

license  build 

super_unko project is the one of the awesome, clean and sophisticated OSS project in the world. Let's create shit commands!

super_unko プロジェクトは世界で最もクリーンで洗練されたOSSプロジェクトの一つです。うんこなコマンドを作りましょう。



Command

Description

super_unkoは世界で最もクリーンで洗練されたOSSプロジェクト

経緯

super_unko

license  build 

super_unko project is the one of the awesome, clean and sophisticated OSS project in the world. Let's create shit commands!

super_unko プロジェクトは世界で最もクリーンで洗練されたOSSプロジェクトの一つです。うんこなコマンドを作りましょう。

Command

Description

世界で最もクリーンで洗練された

経緯


出力がクリーンで
洗練されていることは自
明

経緯

しかし

ソースコードは・・・？ 

経緯

 [unkontributors](#) / [super_unko](#)

Watch ▾

7

★ U

<> Code

! Issues 2

🔗 Pull requests 2

📁 Projects 0

📖 Wiki

🛡 Security

📊 Insights

⚙ Settings


Super Unko project is the one of the awesome, clean and sophisticated OSS project in the world.

[Manage topics](#)

🕒 272 commits

🌿 3 branches

🏷 0 releases

 **14 contributors**

Branch: master ▾


New pull request

Create new file

Upload files

Find

経緯

 jiro4989  数値のみ受け付けるように修正 #56		Latest commit 92a09bc 3 days ago
..		
 bigunko.show	 bigunko.showにshfmtとshellcheckをかけて通るようにした	2 months ago
 super_unko	 フォーマッタの指摘を修正 (#41)	2 months ago
 unko.any	 sedの置換式内で参照する文字列に / (置換式のセパレータ)が含まれていた時にエラーが発生する不具...	21 days ago
 unko.encode	unko.encodeのsedのzオプションを削除(BSD系列対応)	last month
 unko.grep	 unko.grepにshfmtをshellcheckをかけて通るようにした	2 months ago
 unko.king	 数値のみ受け付けるように修正 #56	3 days ago
 unko.life	 unko.lifeをshfmtをshellcheckにかけて通るようにした	2 months ago
 unko.ls	 unko.lsをshfmtとshellcheckにかけて通るようにした	2 months ago
 unko.printpnm	 ドキュメンテーションコメントの追加、不要なshiftの削除	2 months ago
 unko.puzzle	 フォーマッタの指摘を修正 (#41)	2 months ago
 unko.question	cleaning! 2nd	2 months ago
 unko.say	 unko.sayにshfmtとshellcheckをかけて通るようにした	2 months ago
 unko.shout	Apply lint	2 months ago
 unko.think	 unko.thinkにshfmtとshellcheckをかけて通るようにした	2 months ago
 unko.toilet	 引数渡せてなかったので修正	2 months ago
 unko.tower	 awk処理の部分を修正 #47	20 days ago
 unko.tr	 unko.trにshfmtとshellcheckをかけて通るようにした	2 months ago
 unko.yes	Change directory structure	7 months ago

経緯

jiro4989 数値のみ受け付けるように修正 #56		Latest commit 92a09bc 3 days ago
..		
bigunko.show	bigunko.showにshfmtとshellcheckを付けて通るようにした	2 months ago
super_unko	フォーマッタの指摘を修正 (#41)	2 months ago
unko.any	sedの置換式内で参照する文字列に / (置換式の区切り文字) が含まれているとエラーが発生する不具合を修正	21 days ago
unko.encode	unko.encodeのsedのzオプションを削除(BSD系列対応)	last month
unko.grep	unko.grepにshfmtとshellcheckを付けて通る	2 months ago
unko.king	数値のみ受け付けるように修正 #56	3 days ago
unko.life	unko.lifeをshfmtとshellcheckに付けて通るようにした	2 months ago
unko.ls	unko.lsをshfmtとshellcheckに付けて通るようにした	2 months ago
unko.printpnm	ドキュメンテーションコメントの追加、不要なshfmtを削除	2 months ago
unko.puzzle	フォーマッタの指摘を修正 (#41)	2 months ago
unko.question	cleaning! 2nd	2 months ago
unko.say	unko.sayにshfmtとshellcheckを付けて通るようにした	2 months ago
unko.shout	Apply lint	2 months ago
unko.think	unko.thinkにshfmtとshellcheckを付けて通る	2 months ago
unko.toilet	引数渡せてなかったのを修正	2 months ago
unko.tower	awk処理の部分を修正	20 days ago
unko.tr	unko.trにshfmtとshellcheckを付けて通るようにした	2 months ago
unko.yes	Change directory structure	7 months ago

経緯

シェルの書き方は
多種多様

経緯

各人に各人の
スタイルがある

経緯

どれが良い
どれが悪いは
決められない

経緯

皆違って

皆良い

経緯

とはいえ、

経緯

とはいえ、

super_unkoは色々な人
がメンテするOSS

経緯

ある程度、
コードスタイルを
統一して

経緯

危険なコードを
機械的に検出して

経緯

内面も外面も
クリーンな
OSSにしたい

目次

- ✓ 概要
- ✓ 自己紹介
- ✓ 経緯
- やったこと
 - 静的解析 (shellcheck)
 - フォーマッタ (shfmt)
 - テストフレームワーク (bats)
 - テスト・CI環境の整備 (docker)
- まとめ
- 謝辞

やったこと

やったこと

- 静的解析 (shellcheck)
- フォーマッタ (shfmt)
- テストフレームワーク (bats)
- テスト・CI環境の整備 (docker)

やったこと / shellcheck

(1)

やったこと / shellcheck

静的解析

shellcheck

やったこと / shellcheck

危険なシェルの実装を検
出し、修正方法を提示して
くれる

やったこと / shellcheck

```
vidar@vidarholen ~$ shellcheck myscript
```

```
In myscript line 7:
```

```
if (( $n > 3.5 ))
```

```
^-- Don't use $ on variables in (( )).
```

```
^-- (( )) doesn't support decimals. Use bc or awk.
```

```
In myscript line 16:
```

```
[[ $1 == $result ]] && mode=lookup
```

```
^-- Quote the rhs of = in [[ ]] to prevent glob interpretation.
```

```
vidar@vidarholen ~$
```


やったこと / shellcheck

shellcheckの

イイトコロ

やったこと / shellcheck

豊富な

ドキュメント

やったこと / shellcheck

指摘された箇所がなぜ問題なのか

やったこと / shellcheck

IDと共に

すべてドキュメント化され
ている

やったこと / shellcheck

例

やったこと / shellcheck

```
jiro4989-pc:[0] 1:nvim* wifi = on [2019-10-10(土) 03:07]
1 #!/bin/bash
2 $
3 error_count=0$
4 $
5 ls *.sh | while read -r shell_script; do$
6   bash "$shell_script"$
7   ret=$?$
8   if [[ "$ret" -ne 0 ]]; then$
9     error_count=$((error_count + 1))$
10  fi$
11 done$
12 $
13 echo "Error count is $error_count"$
~
~
~
a.sh [utf-8][unix] (1,1)/13 全て
```

やったこと / shellcheck

```
jiro4989-pc:[0] 1:nvim* wifi = on [2019-10-10(土) 03:07]
1 #!/bin/bash
2 $
3 error_count=0$
4 $
5 ls *.sh | while read -r shell_script; do$
6   bash "$shell_script"$
7   ret=$?$
8   if [[ "$ret" -ne 0 ]]; then$
9     error_count=$((error_count + 1))$
10  fi$
11 done$
12 $
13 echo "Error count is $error_count"$
~
~
~
a.sh [utf-8][unix] (1,1)/13 全て
```

バグ
あり

tile vertically

やったこと / shellcheck

shellcheck

してみる

3:07:41 jiro4989@jiro4989-pc [/tmp]

(*'-')! < % shellcheck a.sh

In a.sh line 5:

```
ls *.sh | while read -r shell_script; do
```

^-----^ SC2012: Use find instead of ls to better handle non-alphanumeric filenames.

^-- SC2035: Use ./*glob* or -- *glob* so names with dashes won't become options.

In a.sh line 9:

```
error_count=$((error_count + 1))
```

^-----^ SC2030: Modification of error_count is local (to subshell caused by pipeline).

In a.sh line 13:

```
echo "Error count is $error_count"
```

^-----^ SC2031: error_count was modified in a subshell. That change might be lost.

For more information:

<https://www.shellcheck.net/wiki/SC2012> -- Use find instead of ls to better ...

<https://www.shellcheck.net/wiki/SC2030> -- Modification of error_count is lo...

<https://www.shellcheck.net/wiki/SC2031> -- error_count was modified in a sub...

3:07:47 jiro4989@jiro4989-pc [/tmp]

(;^q^)? < %

やったこと / shellcheck

In a.sh line 13:

```
echo "Error count is $error_count"
```

^-----^ SC2031: error_count was m

ost.

やったこと / shellcheck



shellcheck SC2031




 [すべて](#)

 ショッピング

 地図

 動画

 画像

 もっと見る

設定

ツール

約 541 件 (0.43 秒)

[SC2031](#) · [koalaman/shellcheck Wiki](#) · [GitHub](#)

<https://github.com> › [koalaman](#) › [wiki](#) › [SC2031](#) ▼ [このページを訳す](#)

Here are some constructs that cause subshells (**shellcheck** may not warn about all of them). In each case, you can replace `subshell1` by a command or function that sets a variable, e.g. simply `var=foo` , and the variable will appear to be unset ...

やったこ

SC2031

Ivan Gonzalez-Polanco edited this page on 29 Aug · 7 revisions

var was modified in a subshell. That change might be lost.

Problematic code:

There are many ways of accidentally creating subshells, but a common one is piping to a loop:

```
n=0
printf "%s\n" {1..10} | while read i; do (( n+=1 )); done
echo $n
```

Correct code:

```
# Bash specific: process substitution. Also try shopts like lastpipe.
n=0
```

やったこと / shellcheck

かなり細かいところまで

やったこと / shellcheck

かなり細かいところまで
指摘をくれる

やったこと / shellcheck


shellcheckの指摘を読んで
で修正するだけで

やったこと / shellcheck

shellcheckの指摘を読んで
で修正するだけで
シェルの理解が深まる

やったこと / shellcheck

shellcheck

神だ... 

やったこと

- 静的解析 (shellcheck)
- フォーマッタ (shfmt)
- テストフレームワーク (bats)
- テスト・CI環境の整備 (docker)

やったこと / shfmt

(2)

やったこと / shfmt

フォーマッタ

shfmt

やったこと / shfmt

インデントなどのコー
ドスタイル
を統一する

やったこと / shfmt

例

```
1 #!/bin/bash$
2 $
3 set -eu$
4 $
5 if [[ 1 = 2 ]];      then$
6                     echo 1 == 2$
7 fi$
8 $
9 info    ()    {$
10        echo -e "[ INFO ] $*"    >&2$
11 }$
12 $
13 info "うんこ"$
14 $
15 for i in {1..10}$
16 do$
17 >->-echo $i$
18 done$
```

~
~
~
~
~

```
(;^q^)? < % ~/bin/shfmt -i 2 -ci -sr -d b.sh  
diff -u b.sh.orig b.sh  
@@ -2,17 +2,16 @@
```

✖ set -eu

```
-if [[ 1 = 2 ]];      then  
-                      echo 1 == 2
```

```
+if [[ 1 = 2 ]]; then  
+ echo 1 == 2  
fi
```

```
-info    ()    {  
-        echo -e "[ INFO ] $"      >&2  
+info() {  
+ echo -e "[ INFO ] $" >&2  
}
```

```
- info "うんこ"  
+info "うんこ"
```

```
-for i in {1..10}  
-do  
-        echo $i  
+for i in {1..10}; do  
+ echo $i  
done
```

```
3:42:07 jiro4989@jiro4989-pc [/tmp]
```

```
(;^q^)? < %
```



```
(;^q^)? < % ~/bin/shfmt -i 2 -ci -sr -d b.sh  
diff -u b.sh.orig b.sh  
@@ -2,17 +2,16 @@
```

※

```
set -eu
```

```
-if [[ 1 = 2 ]]; then  
-    echo 1 == 2
```

```
+if [[ 1 = 2 ]]; then  
+    echo 1 == 2  
fi
```

```
-info () {  
-    echo -e "[ INFO ] $" >&2  
+info() {  
+    echo -e "[ INFO ] $" >&2  
}
```

```
- info "うんこ"  
+info "うんこ"
```

```
-for i in {1..10}  
-do  
-    echo $i  
+for i in {1..10}; do  
+    echo $i  
done
```

```
3:42:07 jiro4989@jiro4989-pc [/tmp]  
(;^q^)? < %
```

※ -w で

フォーマットして 上書きできる

やったこと / shfmt

shfmt

すごすぎる...

やったこと / shfmt

shfmtの

イイトコロ

やったこと / shfmt

機械的に
コードスタイルを
統一できる

やったこと / shfmt

shfmtコマンドに
スクリプトを
渡すだけ

やったこと / shfmt

コマンドもGo製
バイナリを配置するだけ

やったこと / shfmt

手間いらず
導入も簡単

やったこと / shfmt

つまり神

やったこと

- 静的解析 (shellcheck)
- フォーマッタ (shfmt)
- テストフレームワーク (bats)
- テスト・CI環境の整備 (docker)

やったこと / bats

(3)

やったこと / bats

テストフレームワーク bats

やったこと / bats

シェルのコマンドや
関数をテストできる

やったこと / bats

```
32  @test 'unko.towerでメッセージを埋め込む' {
33      run "$TARGET_COMMAND" -s あい
34      [ "$status" -eq 0 ]
35      [ "${lines[0]}" = "          人" ]
36      [ "${lines[1]}" = "        (あいあ) " ]
37      [ "${lines[2]}" = "      (いあいあい) " ]
38      [ "${lines[3]}" = "    (あいあいあいあ) " ]
39  }
```

40

```
:02:41 jiro4989@jiro4989-pc [~/src/github.com/unkontributors/super_unko/test][master]
```

```
*'-')! < % ./unko.tower-test.bats
```

✓unko.towerは引数なしのときは3段

✓unko.towerで引数2

✓unko.towerで引数4

✓unko.towerでメッセージを埋め込む

✓unko.towerで先頭から文字数分だけメッセージを埋め込む

✓unko.towerで半角文字は全角文字1つ分で置換される

✓unko.tower -mで置換文字に/が含まれていても置換できる

✓unko.tower -sで置換文字に/が含まれていても置換できる

✓unko.tower -sで置換文字に\\が含まれていても置換できる

✓unko.tower -sASCIIコード表33~127までの全ての文字は半角文字として扱う (\\は除外)

0 tests, 0 failures

```
:02:47 jiro4989@jiro4989-pc [~/src/github.com/unkontributors/super_unko/test][master]
```

```
*'-')! < % █
```

やったこと / bats

batsの

イイトコロ

やったこと / bats

① 基本的なテストAPI が全部ある

やったこと / bats

自前でいいいち
実装しなくていい

やったこと / bats

②標準出力も

エラー出力もテストで
きる

やったこと / bats

③関数も

テストできる

やったこと / bats

すなわち神

やったこと

- 静的解析 (shellcheck)
- フォーマッタ (shfmt)
- テストフレームワーク (bats)
- テスト・CI環境の整備 (docker)

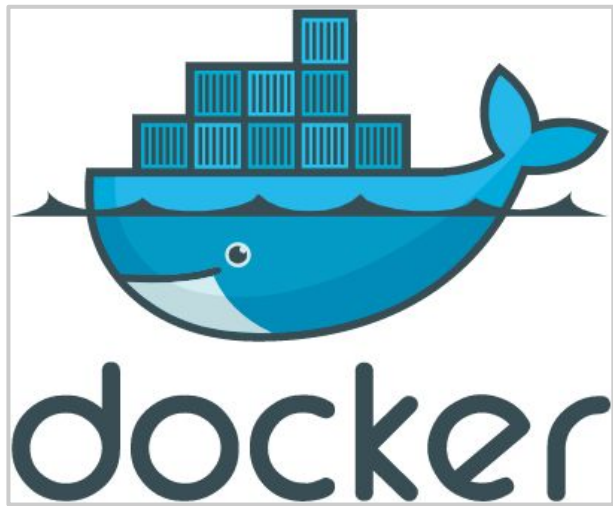
やったこと / docker

(4)

やったこと / docker

テスト・CI環境の 整備

やったこと / docker



でローカルPCに
複数のBash環
境を構築

やったこと / docker

経緯

やったこと / docker

経緯

2つの課題

やったこと / docker

①テストに時間
がかかる

やったこと / docker

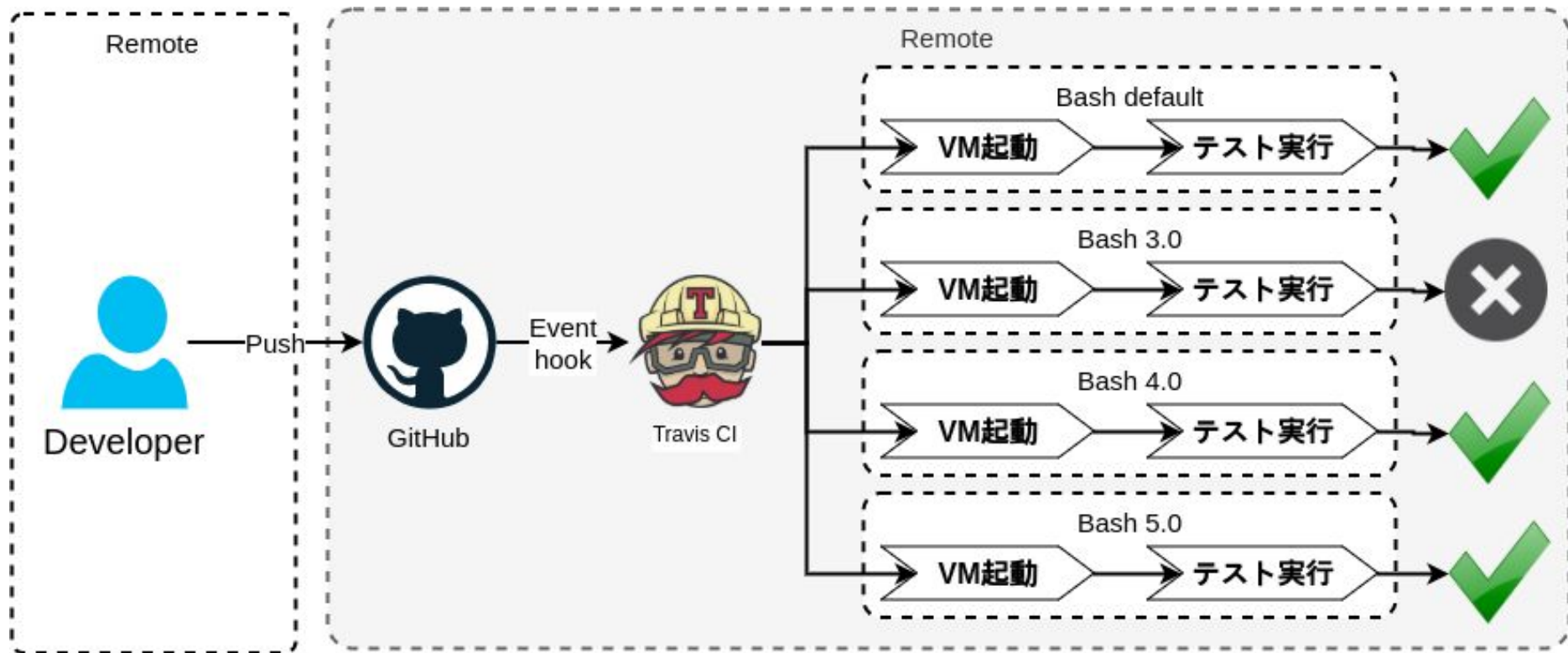
Docker導入前

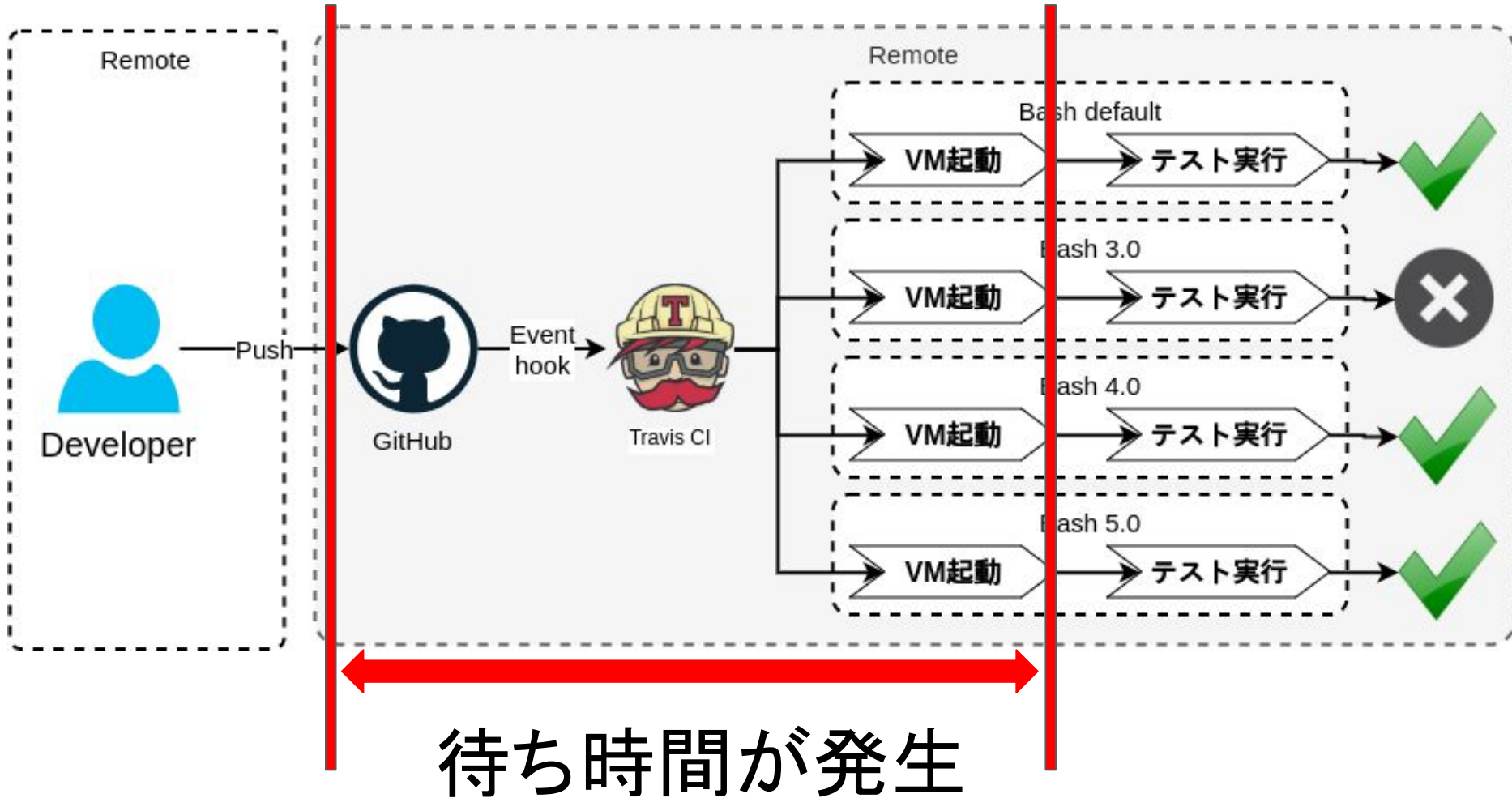
やったこと / docker

Travis CI環境で 複数のBashでの テストしていた

やったこと / docker

Pushしないと
テストができない





やったこと / docker

②環境差異による テスト失敗

やったこと / docker

docker導入前

やったこと / docker

Travis CI環境で 直接batsを実行

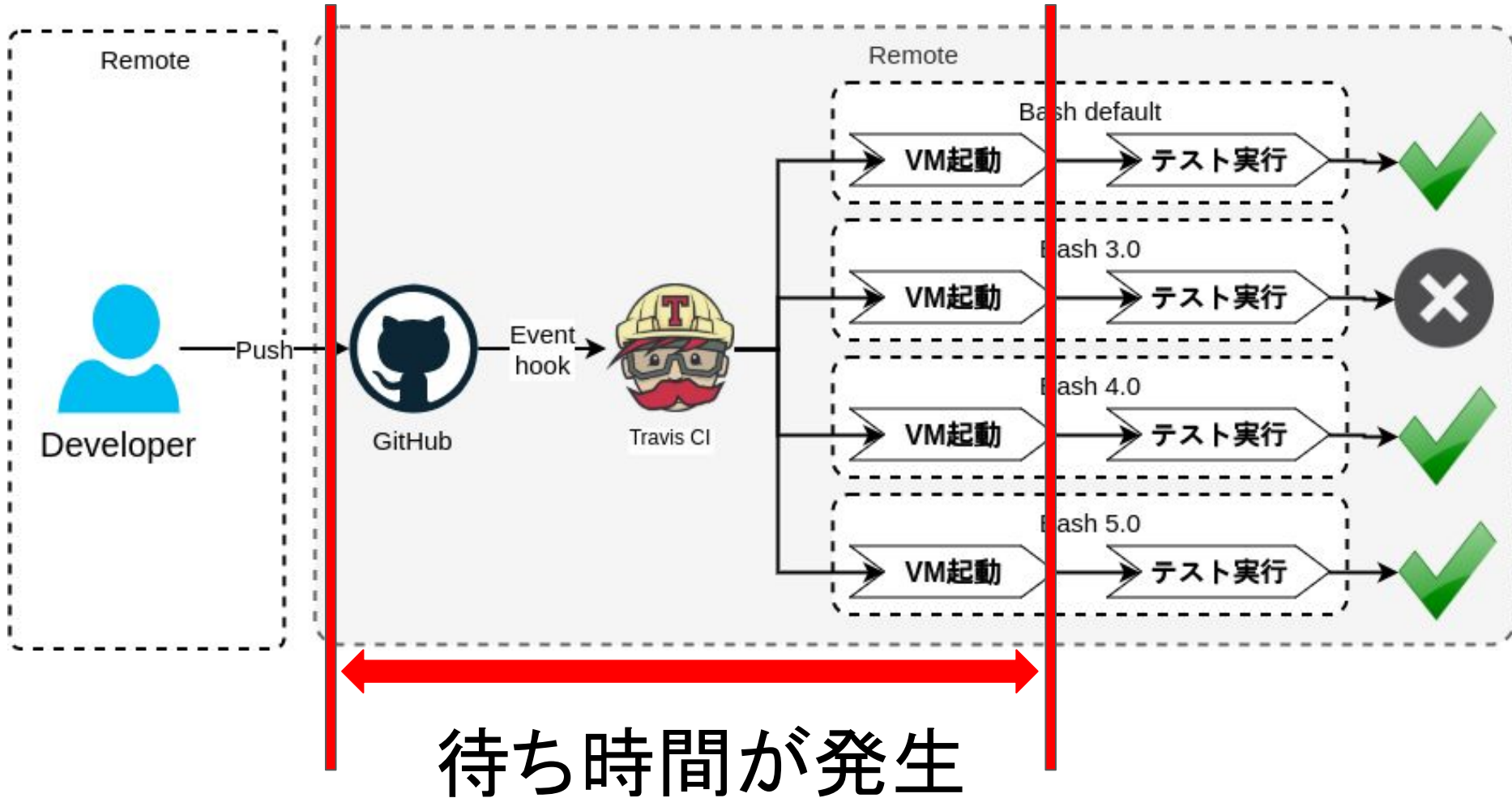
やったこと / docker

TravisCI環境に 日本語環境入れ忘れ てテストが失敗

やったこと / docker

Pushしないと

調査ができず



やったこと / docker

原因究明に難航

(主に僕が)

やったこと / docker

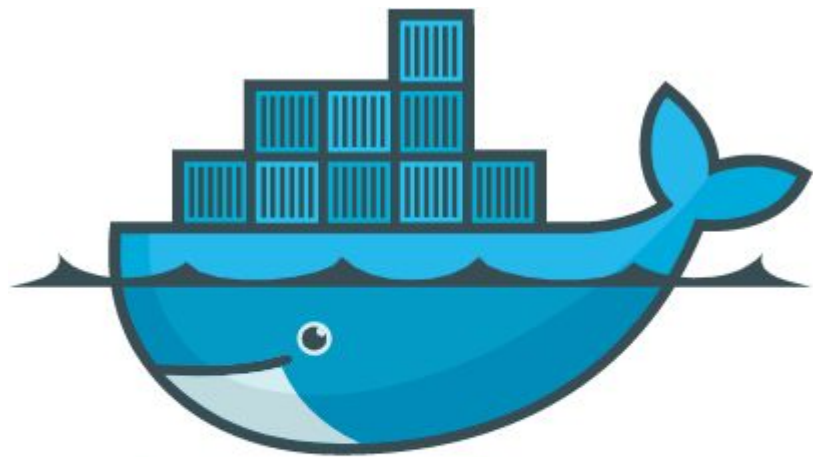
ローカルとTravis CI
のテスト環境を
統一したい

やったこと / docker

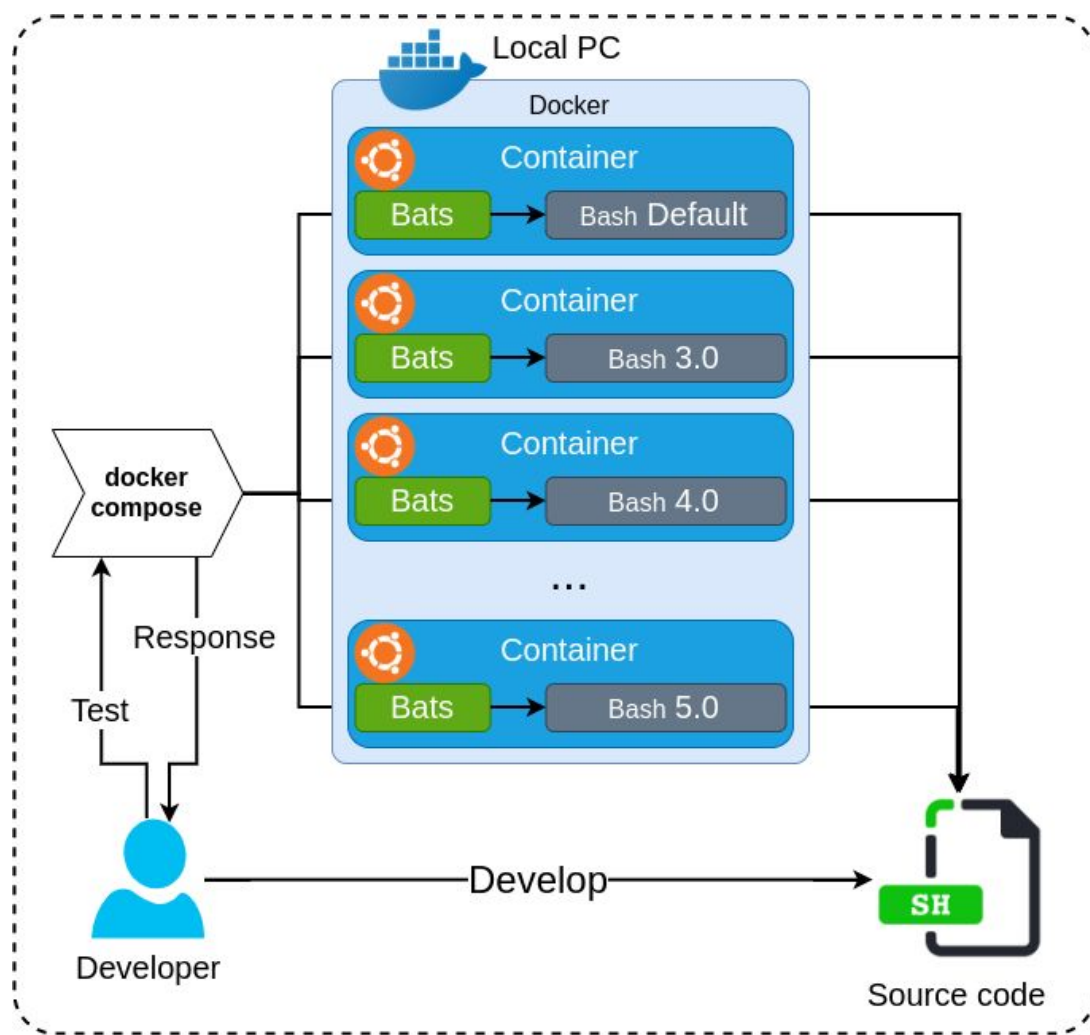
そのための

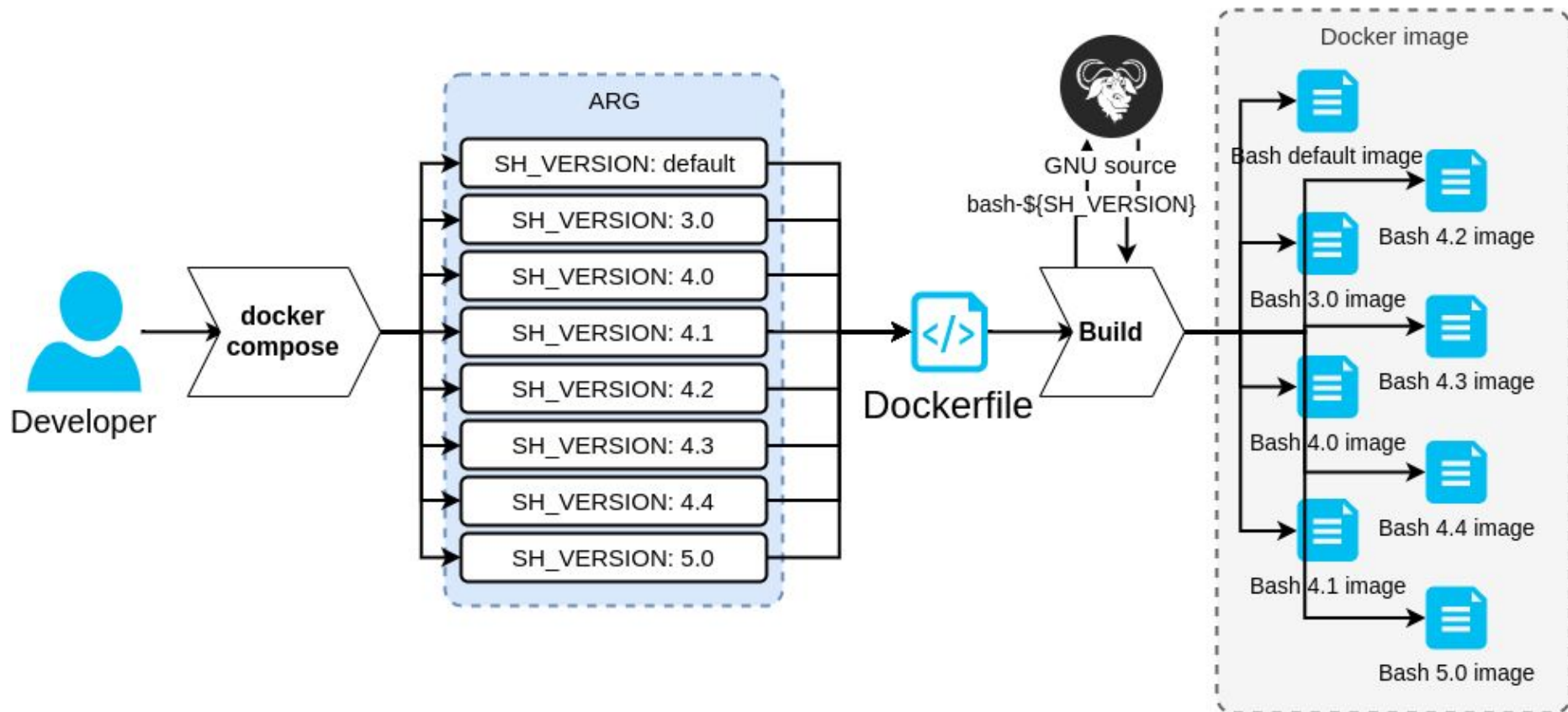
やったこと / docker

そのための



docker





やったこと / docker

あとshellcheckと

shfmtと

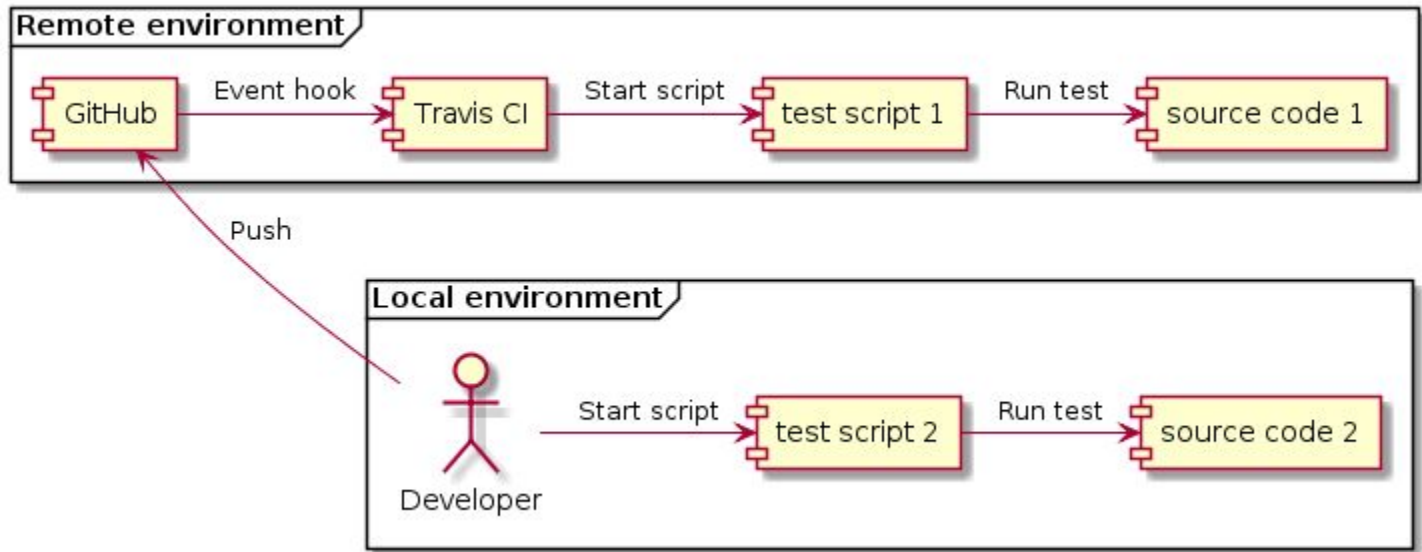
batsも

Dockerコンテナを使って環境を用意するようになった

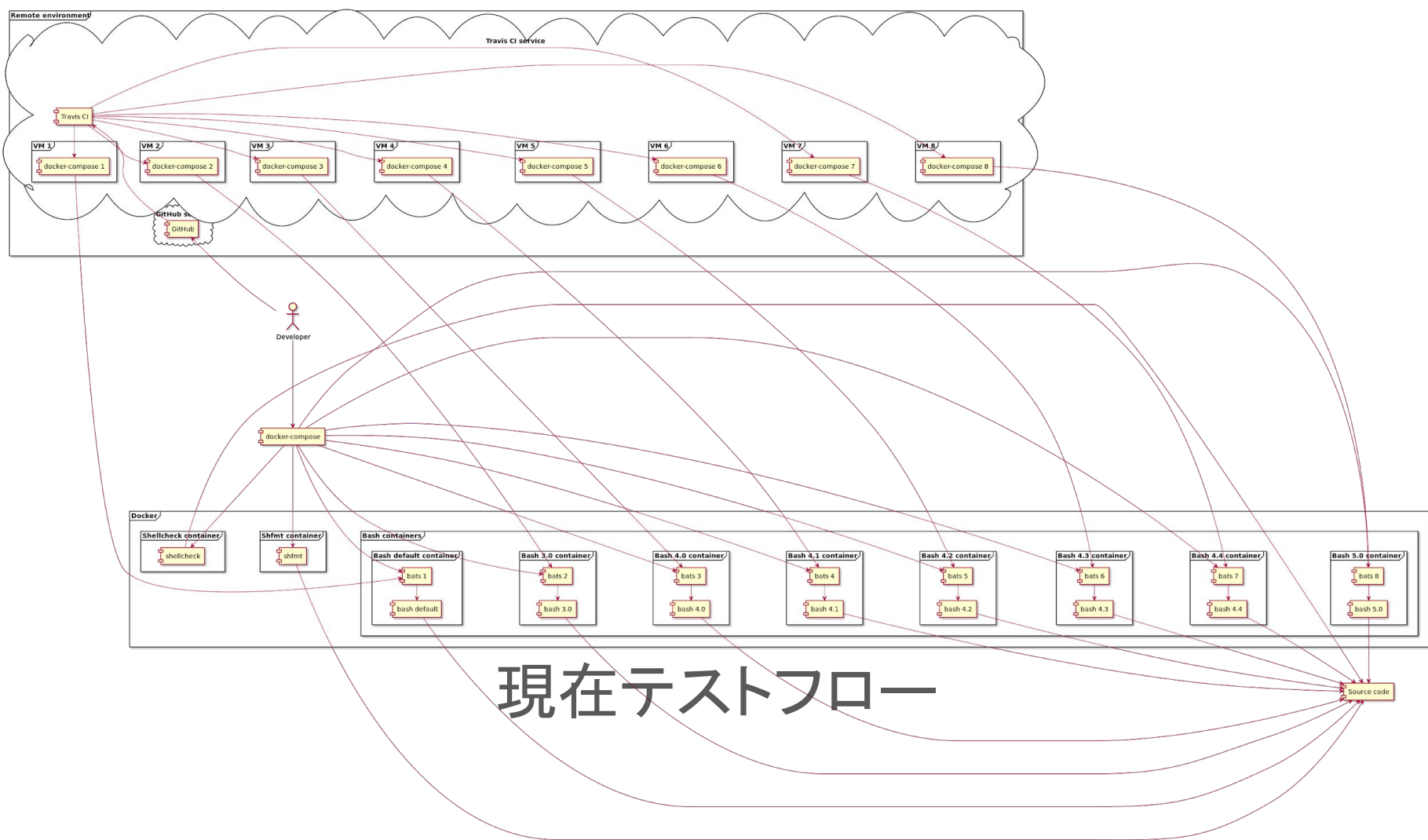
やったこと / docker

デモ

やったこと / docker



変更前のテストフロー

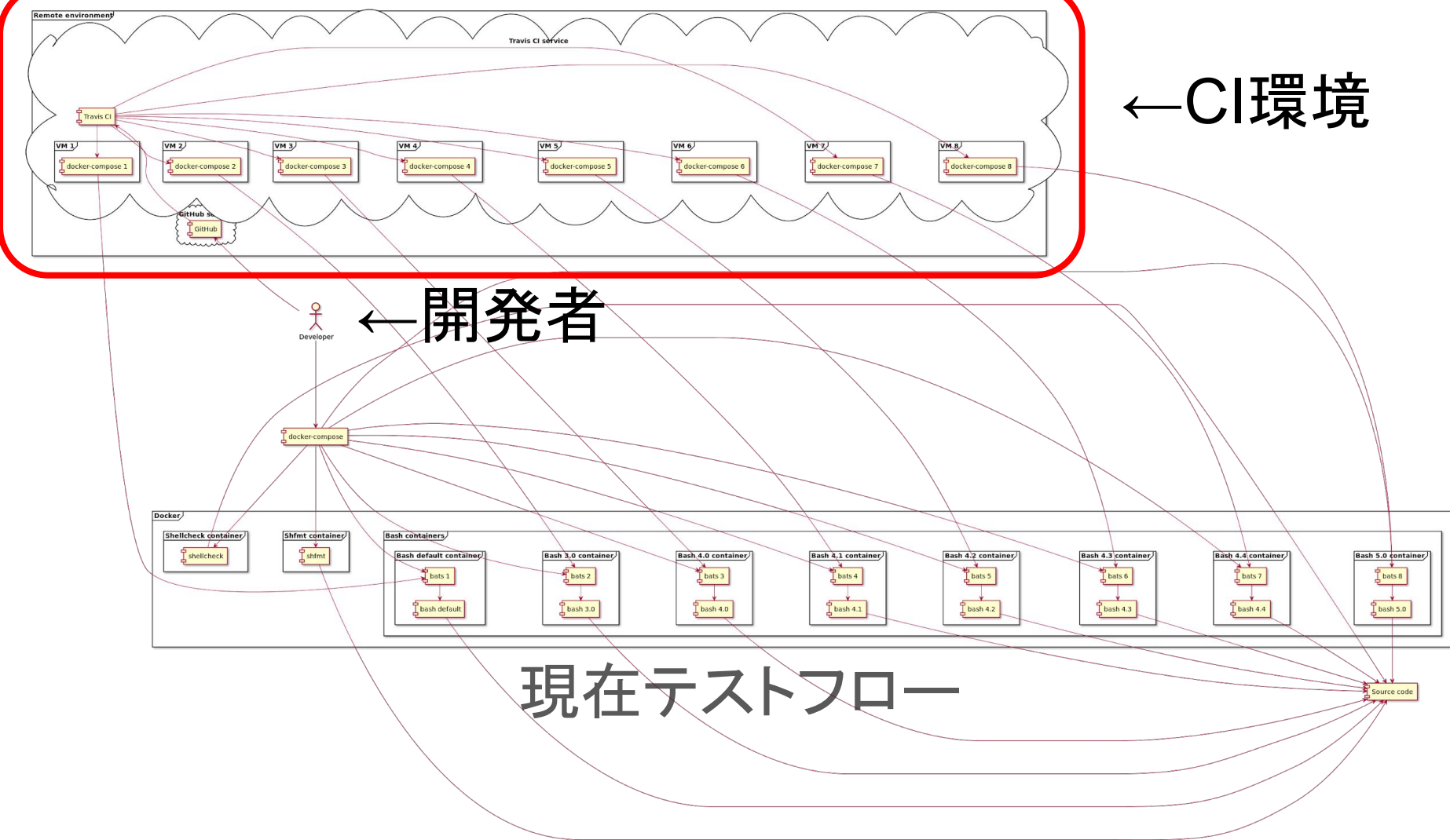


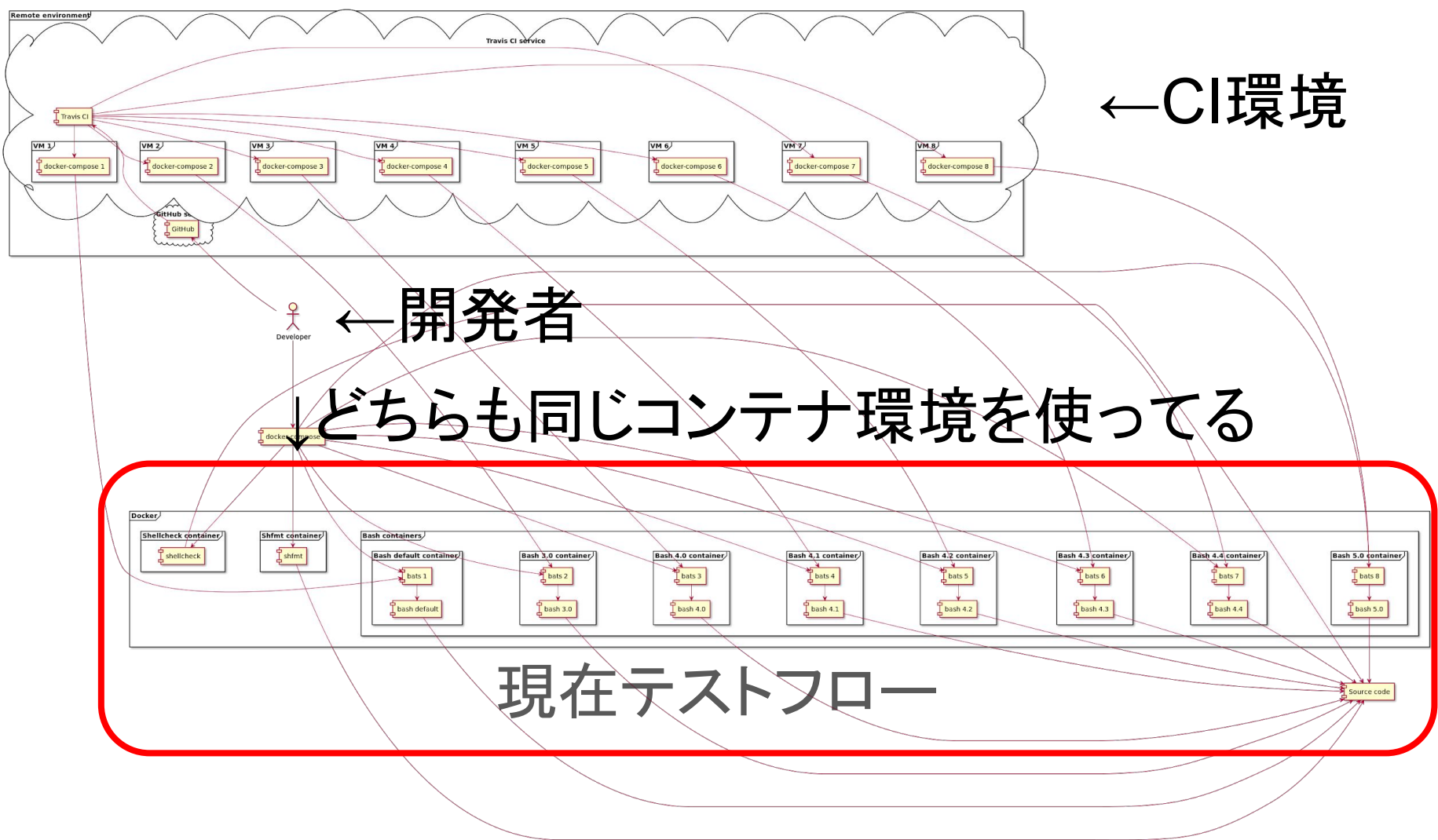
現在テストフロー

← CI環境

← 開発者

現在テストフロー





目次

- ✓ 概要
- ✓ 自己紹介
- ✓ 経緯
- ✓ やったこと
 - 静的解析 (shellcheck)
 - フォーマッタ (shfmt)
 - テストフレームワーク (bats)
 - テスト・CI環境の整備 (docker)
- まとめ
- 謝辞

まとめ

まとめ

まとめ/やったこと

- 静的解析 (shellcheck)
- フォーマッタ (shfmt)
- テストフレームワーク (bats)
- テスト・CI環境の整備 (docker)

まとめ/感想

感想

まとめ/感想

①シエルの勉強に なった

まとめ/感想

shellcheckの指摘 は有用

まとめ/感想

シェルの細かい挙動 の理解につながる

まとめ/感想

②コードが

読みやすくなった

まとめ/感想

インデントが
揃ってるだけでも
だいぶ読みやすい

まとめ/感想

他の方のコマンドを修正する時に効果を感じた

まとめ/感想

③テストしやすくなっ
た

まとめ/感想

shellcheckやbatsと
かも全部Dockerコン
テナで使う

まとめ/感想

dockerだけあれば
環境が整う

まとめ/感想

すぐ修正して
すぐ確認できる

まとめ/反省

反省

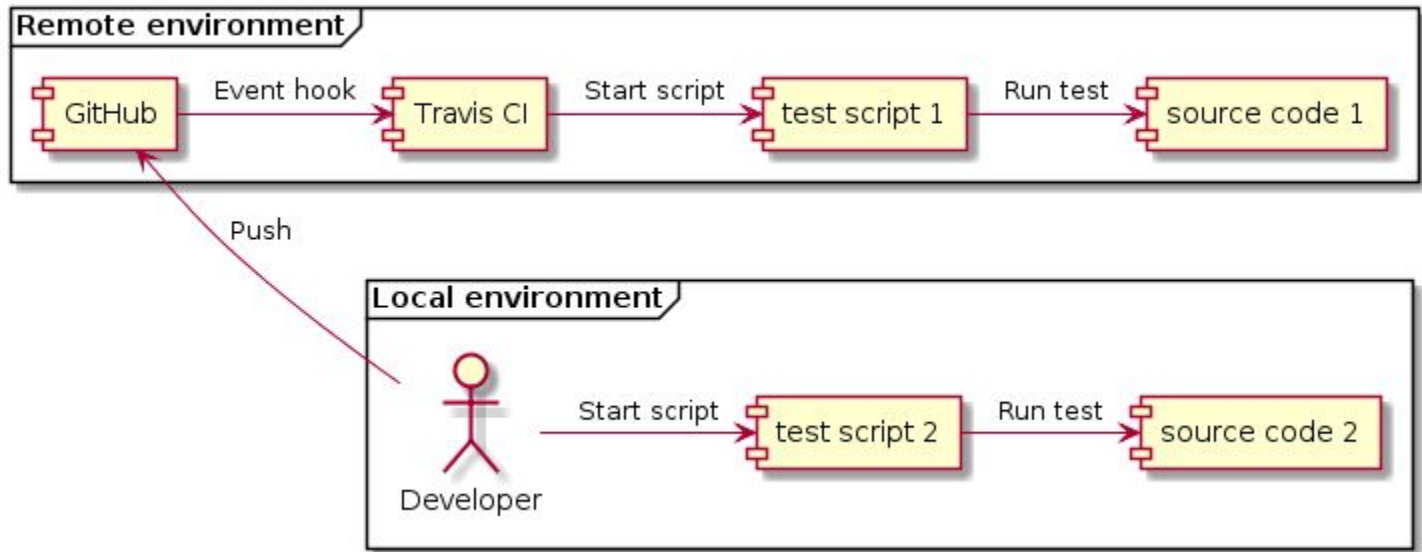
まとめ/反省

やりすぎた

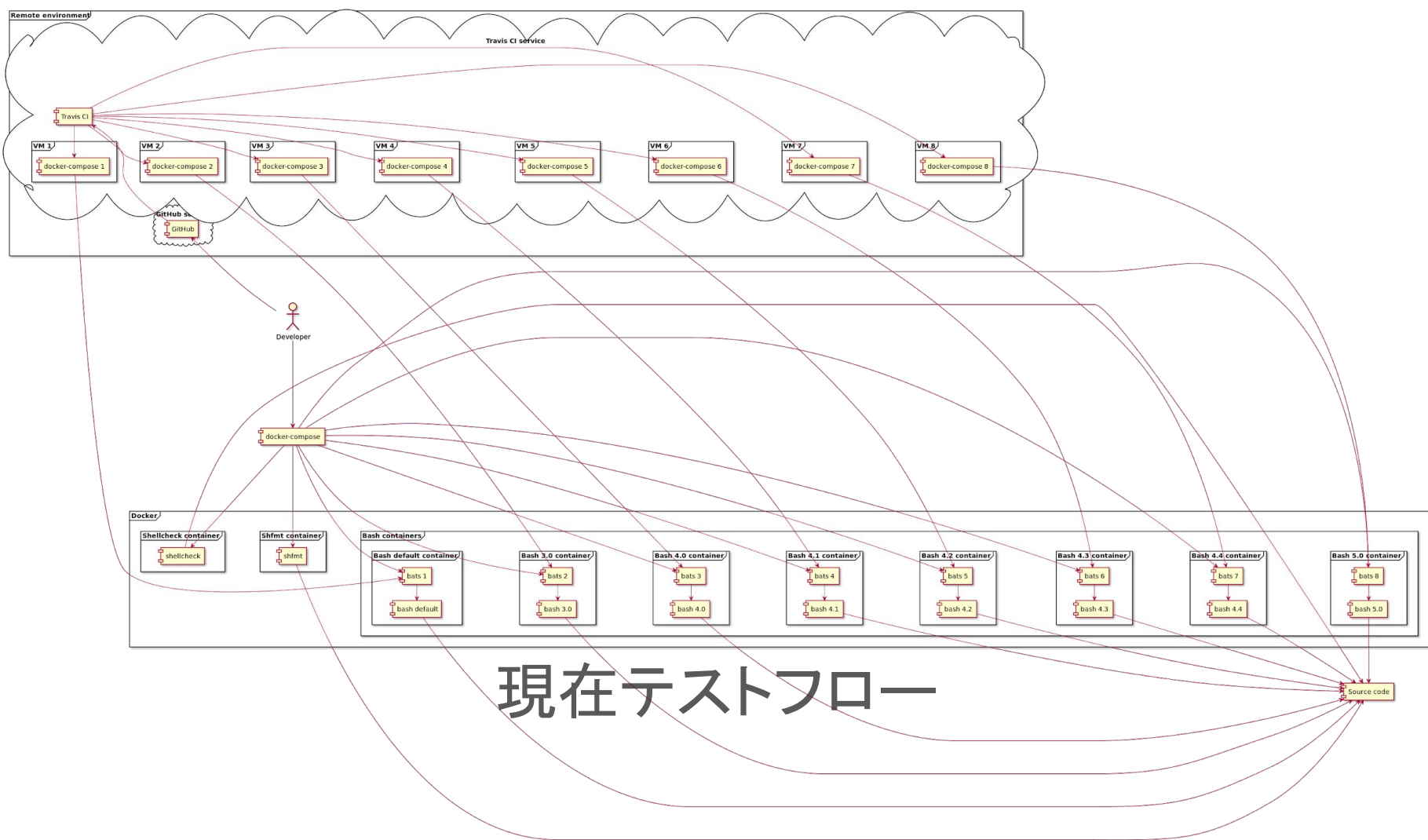
まとめ/やったこと

- 静的解析 (shellcheck)
- フォーマッタ (shfmt)
- テストフレームワーク (bats)
- テスト・CI環境の整備 (docker)

やったこと / docker



変更前のテストフロー



まとめ/反省

着手する前から
やるべきかは
悩んでた

まとめ/反省

次郎「super_unkoを
静的解析したいです！」

まとめ/反省

次郎「super_unkoを
静的解析したいです！」

なんて言おうものなら

まとめ/反省

お前は何を言っているんだ

まとめ/反省

これは
うんこのOSSだぞ

まとめ/反省

お前は何
を言っでいい

まとめ/反省

とか言われるのでは
と思っていた

まとめ/反省

結果的には 無事マージして いただいた





無理言ってごめんなさい！

目次

- ✓ 概要
- ✓ 自己紹介
- ✓ 経緯
- ✓ やったこと
 - 静的解析 (shellcheck)
 - フォーマッタ (shfmt)
 - テストフレームワーク (bats)
 - テスト・CI環境の整備 (docker)
- ✓ まとめ
- 謝辞

謝辭

謝辭

謝辞

何度もレビューしてくださったsuper_unko創
設者の

ぐれさんさん、ありがとうございました

謝辞

大量のコマンドを作ってくださっていた

unkontributorsの皆さん

ありがとうございます

終わりの言葉

駆け足のLTでしたが
皆さんのシェル芸の
一助となれたら
幸いです

シェル芸ではないような...

終わりの言葉

exit 0

ご静聴ありがとうございました

