

cloudコンテナ基盤(GKE/Kubernetes)を用いた実運用システムの開発の現実

DataCenter とソフトウェア開発ワークショップ - 2016-02-18

PhalanXware 加藤 真透

ファランクスウェア

加藤 真透

OCRを用いたシステム開発実装
ISDB-T 視聴アプリ開発実装
同 視聴STB開発実装
ソーシャルゲーム開発実装運用
ECサイト開発実装運用



アウトライン

- ・ 開発に用いた基盤システムについて
 - ・ Kubernetes、GCE、GKE
- ・ 開発に用いた方法、そのツールについて
 - ・ Ruby、Ansible、Docker、Google Cloud SDK
 - ・ それらの最新版追従の方法
- ・ 他、観測した開発手法について

開発に用いた基盤システムについて

Kubernetesとは

- ・ 現在 GitHub 上で開発が行われている OSS
- ・ Docker Container の オーケストレーションシステム
- ・ クラスタを構成するノードを管理し、コンテナ実行をユーザ定義要件(CPU、メモリなど)に従ってスケジューリングしてくれる
- ・ label や pod と呼ばれる論理的な単位で作成、管理、公開などができる



kubernetes by Google

Manage a cluster of Linux containers as a single system to accelerate Dev and simplify Ops.

[View on GitHub](#)

[Try Kubernetes](#)

GKEとは

Google Cloud Platform による
Kubernetes の提供
(Amazon によるものなどがある)



CONTAINER ENGINE

Google Cloud Platform で Docker コンテナを実行し、Kubernetes で管理

[CONTAINER ENGINE ドキュメントを見る](#)

[コンソールへ移動](#)

コンテナ管理を自動化

Google Container Engine は、Docker コンテナの実行を支える強力なクラスターマネージャおよびオーケストレーションシステムとして機能します。ユーザーが定義する要件（CPU やメモリなど）に基づいてコンテナをクラスターにスケジューリングし、自動的に管理します。オープンソースの **Kubernetes** システム上に構築されており、ユーザーはオンプレミス、ハイブリッド、パブリッククラウドインフラストラクチャを柔軟に利用できます。



クラスターを数分でセットアップ

わずか数分で仮想マシンのマネージドコンテナクラスターをセットアップし、デプロイの準備を整えることができます。クラスターではロギングやヘルスチェックといった機能が提供され、アプリケーションを簡単に管理できます。

宣言的管理

予約する CPU/メモリ量、レプリカの数、キープアライブポリシーといったコンテナの要件は、シンプルな JSON 形式の設定ファイルで宣言します。Container Engine は宣言に沿ってコンテナのスケジューリングを行い、要件が満たされるようにアプリケーションを能動的に管理します。



GKEのメリット

- ・ Kubernetes の開発を主導している Google が提供しているだけあって、追従がはやい。
- ・ 実質Kubernetes リファレンス実装
- ・ Kubernetesクラスタをほぼ 1 コマンドで構築でき、GCPの提供する他サービスを簡単に利用することができる。(一部の機能はGCPを用いて実装されている)



CONTAINER ENGINE

Google Cloud Platform で Docker コンテナを実行し、Kubernetes で管理

[CONTAINER ENGINE ドキュメントを見る](#)

[コンソールへ移動](#)

コンテナ管理を自動化

Google Container Engine は、Docker コンテナの実行を支える強力なクラスタ マネージャおよびオーケストレーション システムとして機能します。ユーザーが定義する要件 (CPU やメモリなど) に基づいてコンテナをクラスタにスケジューリングし、自動的に管理します。オープンソースの **Kubernetes** システム上に構築されており、ユーザーはオンプレミス、ハイブリッド、パブリック クラウド インフラストラクチャを柔軟に利用できます。



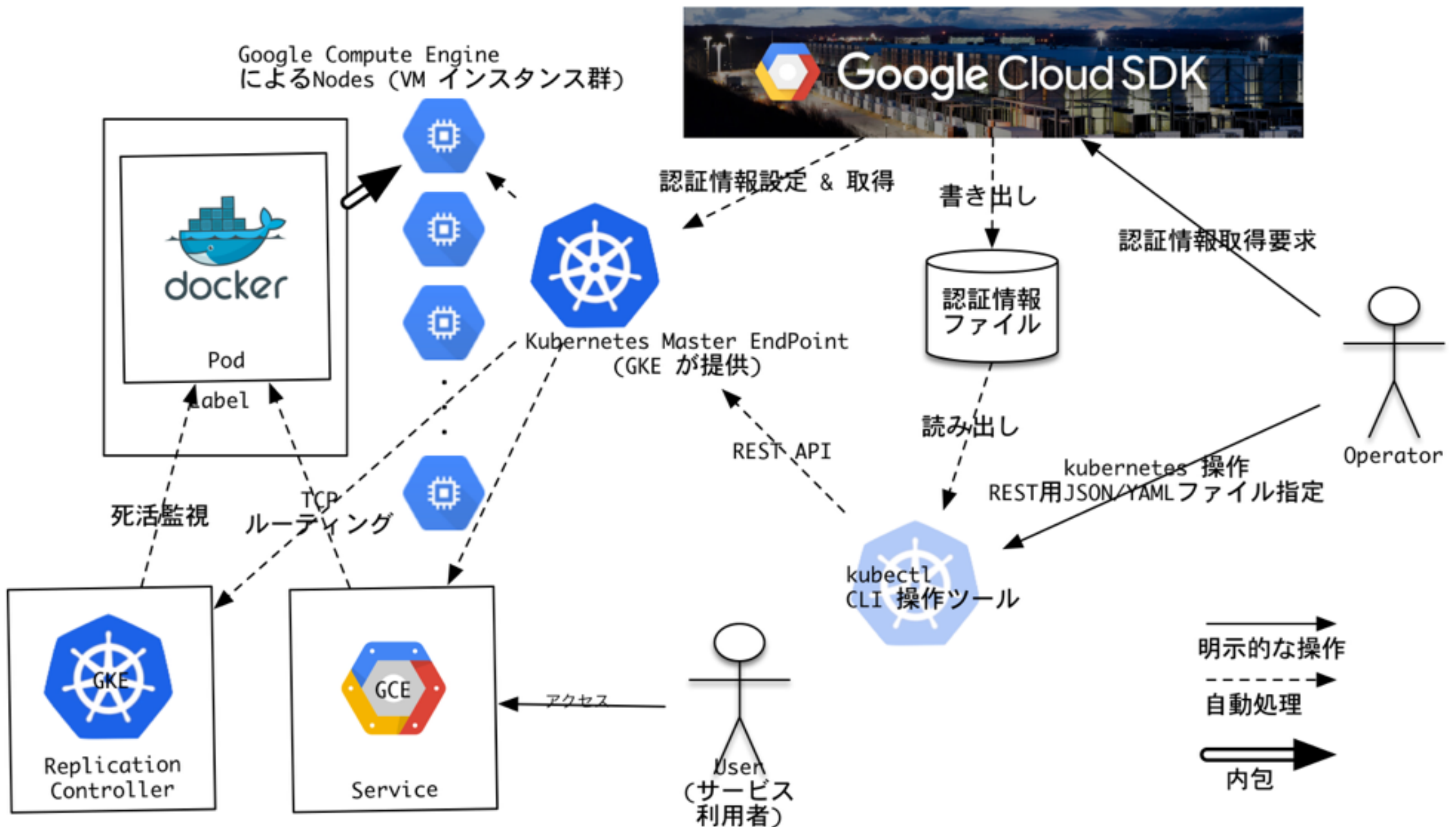
クラスタを数分でセットアップ

わずか数分で仮想マシンのマネージド コンテナ クラスタをセットアップし、デプロイの準備を整えることができます。クラスタでは **ロギング** や **ヘルス チェック** といった機能が提供され、アプリケーションを簡単に管理できます。

宣言的管理

予約する CPU/メモリ量、レプリカの数、キープアライブ ポリシーといったコンテナの要件は、シンプルな JSON 形式の設定ファイルで宣言します。Container Engine は宣言に沿ってコンテナのスケジューリングを行い、要件が満たされるようにアプリケーションを能動的に管理します。



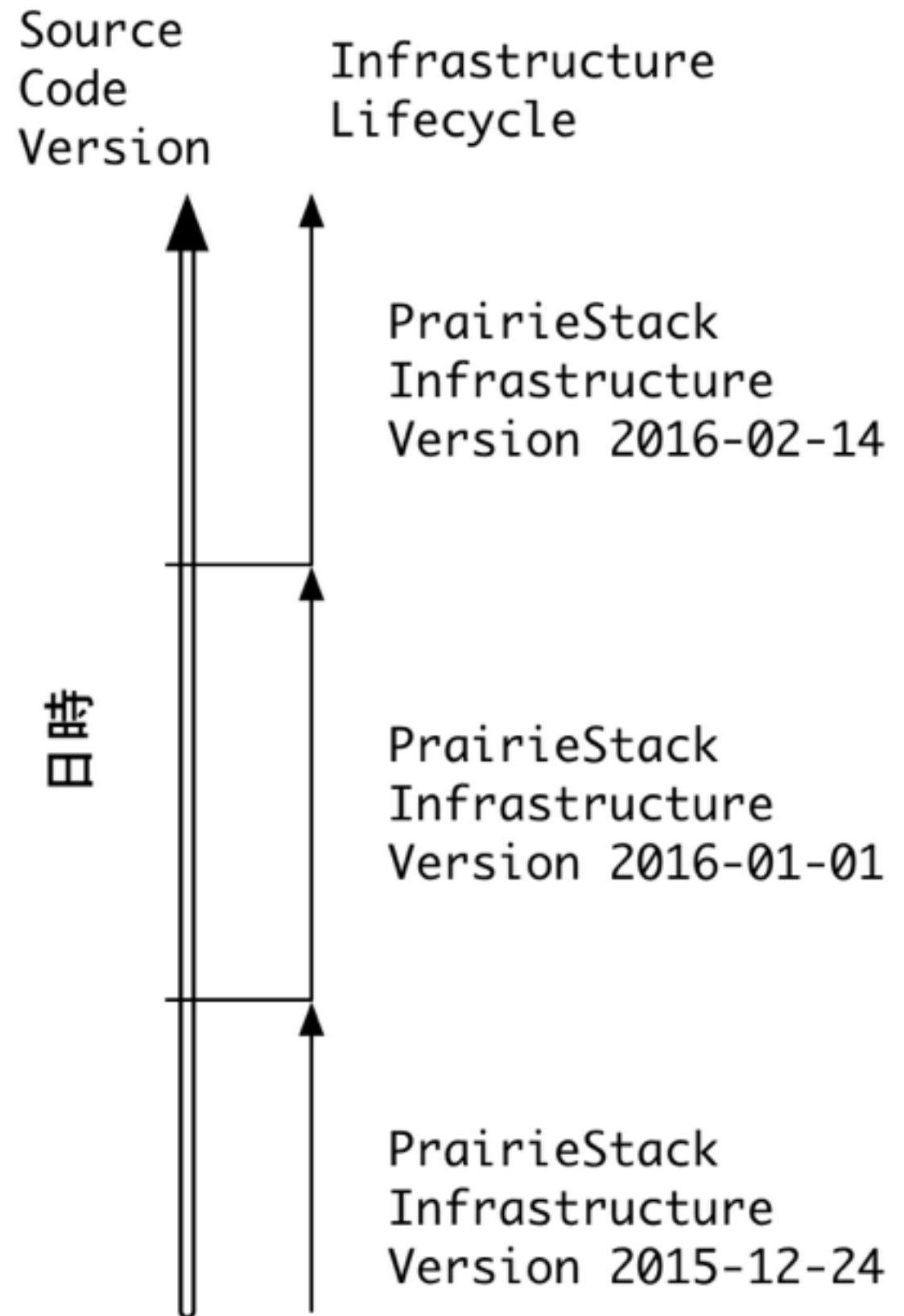


GKEとKubernetes 関係図

開発に用いた方法、そのツールについて

Immutable Infrastructure

- ・ クラウドの利点を活かして、作って壊す、「使い捨て」で開発がよい。
- ・ 新しくなるものをクリーン(旧バージョンなソフトウェアが入っていない)な状態から構築する
- ・ 各インスタンスの状態を考えなくて良い
- ・ 開発者ごとの環境を作りやすい



Infrastructure as Code

- Ruby
 - Rake task
 - GEM(Bundler)
- Ansible
- Docker
- Google Cloud SDK
 - Kubernetes

```
Dockerfile<psc-wee-base>
1 FROM ubuntu:latest
2
3 ENV DEBIAN_FRONTEND noninteractive
4 RUN echo Asia/Tokyo > /etc/timezone
5 RUN dpkg-reconfigure --frontend noninteractive tzdata
6
7 RUN apt-get update -y
8
9 # for Ruby 2.2
10 RUN apt-get -y install software-properties-common
11 RUN add-apt-repository -y ppa:brightbox/ruby-ng
12 RUN apt-get -y update
13
14 RUN apt-get -y install build-essential make libssl-dev ruby2.2 \
15     ruby2.2-dev && apt-get clean
```


苦しんだこと - 1

構築スクリプト実行環境を構築・統一する

- ・ 課題

- ・ Infrastructure構築をコード化しても、そのコードを実行させることのできる環境を構築するのに苦労する

- ・ 解決策

- ・ staging(踏み台) インスタンスを用意した

- ・ 注意点

- ・ staging インスタンスからさらに他サービスを利用するには特別なパーミッションが必要 (googleでもAWSでも)

苦しんだこと - 2

構築する際、別のインフラに依存がある

- ・ 課題

- ・ 依存を解決するのにインターネット上の別インフラに依存していることが多い
- ・ 例えばGemfile.lockなどで使用するパッケージのバージョンも含めて指定できるとしても、それが入手できないことがある(CDNに障害がでたこともあった)

- ・ 解決策

- ・ 出来る限り、内部パッケージングするところもコード化し、依存を減らす、依存が解決できたときの状態をパッケージングする

苦しんだこと - 3

Dockerのキャッシュ機能の利点と課題

- ・ 課題

- ・ Docker Image のビルドは、なるべくキャッシュが使われてとても早い
- ・ しかし、基本イメージ(= Linux)に加えて要求するパッケージを取得したい場合、キャッシュされているイメージに埋め込まれている、そのパッケージ取得先が変わってしまい、キャッシュが使えないことがある

- ・ 解決方法

- ・ 速度は犠牲にして毎度no-cacheを使う
- ・ 毎度ビルドせず、動いたものの自体を用いる

苦しんだこと - 4

月曜日デモがしにくい

- ・ 課題

- ・ 金曜日確認して、壊して、翌月曜日作れない
- ・ α 更新リリースが日本時間からずれて9時間後

- ・ 解決策

- ・ 影響先のリリーススケジュールやバージョン固定機能を確認、あれば用いる
- ・ CIを用いて定期的にテスト・確認し追従する（後述）

GKE、Kubernetes、Docker を α バージョン から追従する

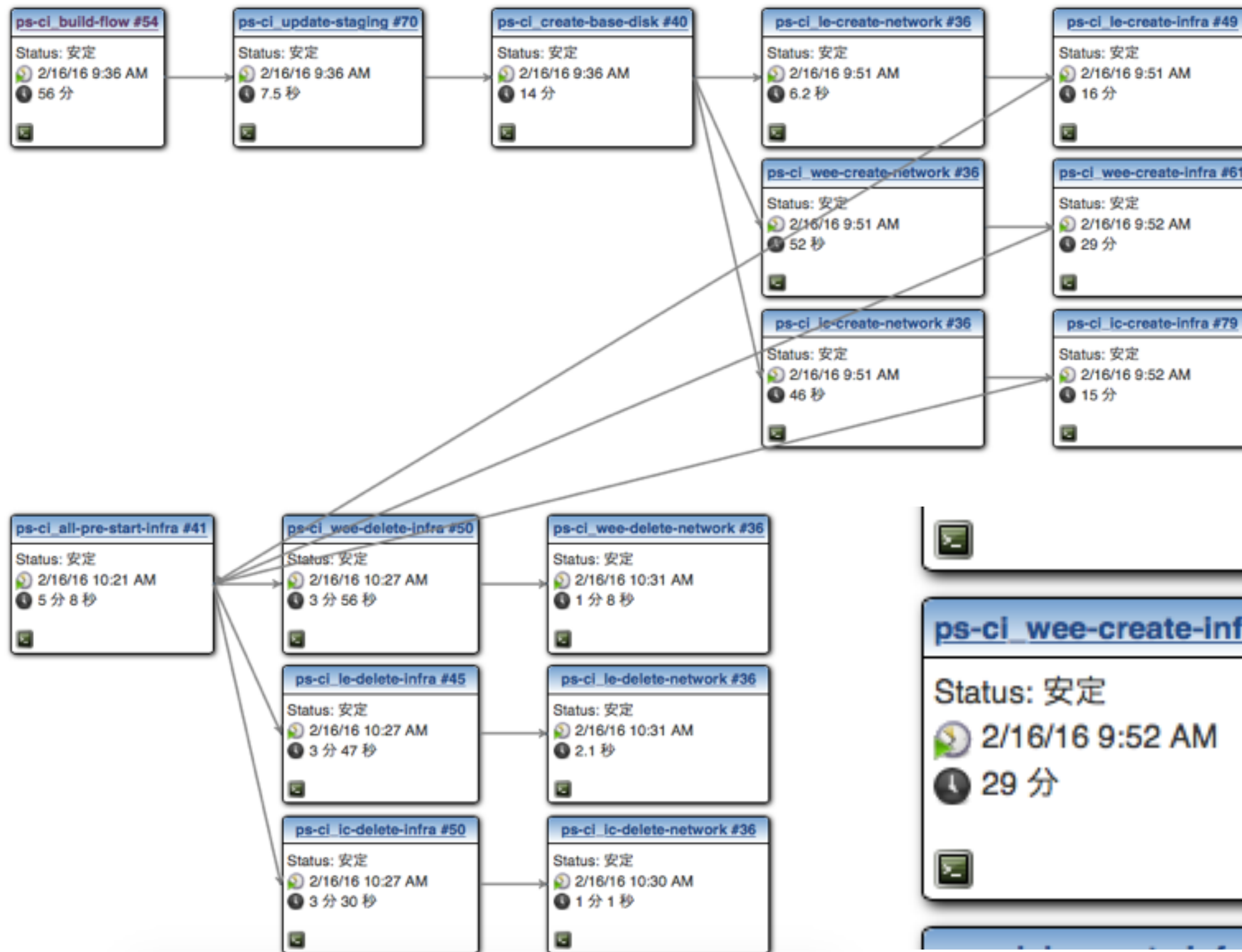
だいたいの α 版は「リリースノート以外（ほとんどの場合、リリースノートにすら乗らずに）パラメータが変更されることが多い
我々が使用している間に、望む機能がどんどん追加されていくため、積極的に追従したい。

追従するには？

- ・ 公開されている議論を読む
- ・ 公開されているissueを読む
- ・ 同じissueで苦しんでいる人がたくさんいると優先度が高くなる
- ・ 優先的に実装されそうなものが見えてくる

CIによるテスト

- ・ 構成されるツールがいつ最新版がリリースされるかわからない。
- ・ 定期的に、これまで動いていたパラメータで構築を行い、ログを保存し、以下を確認する。
 - ・ ログ中にエラー、警告メッセージが確認されるか
 - ・ REST API Server に投げる値
 - ・ REST API Server から得られた値
 - ・ 構築されたものに必要なものがあるか

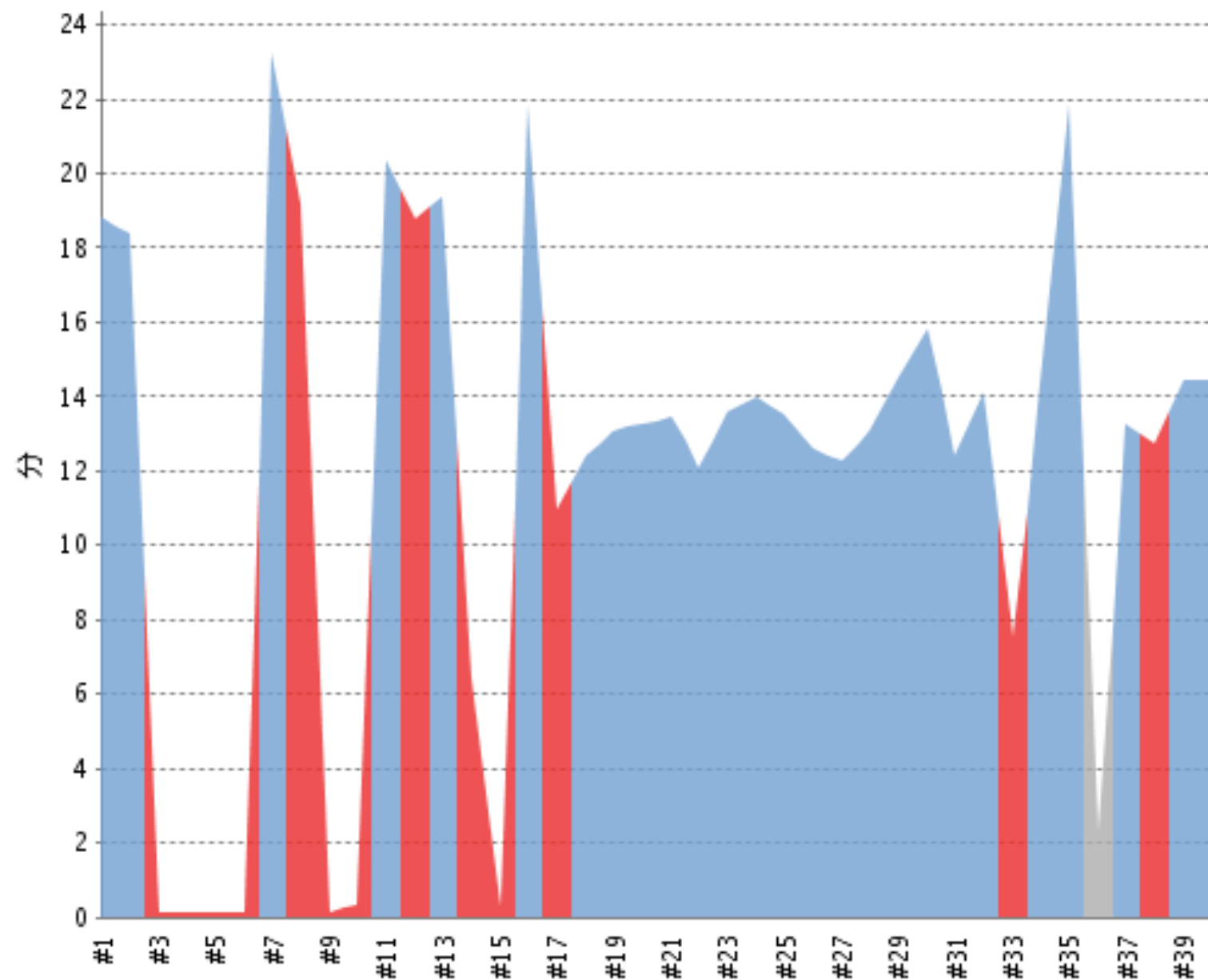


Jenkins CI Build Flow

DSLでテストする順序を書ける
失敗したらログは残すがインスタンス
は捨てる

ビルド時間の推移

ビルド	↑ 所要時間
#40	14 分
#39	14 分
#38	12 分
#37	13 分
#36	2 分 15 秒
#35	21 分
#34	14 分
#33	7 分 29 秒
#32	14 分
#31	12 分
#30	15 分
#29	14 分
#28	13 分
#27	12 分
#26	12 分
#25	13 分
#24	13 分
#23	13 分
#22	12 分



Jenkins CI Build 履歴

必要な時間、失敗の頻度など
確認できる

他観測した手法について

観測した手法 - 1

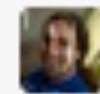
GitHub -

PullRequest 上

で議論する

狙いや設計などをブランチを切ってテキストファイルを書いて議論
GitHubでは行ごとにツッコミを入れることができるので進めやすいようだ。

```
574 + during beta may frequently turn up
575 + - Upgradeability: the object schema
    software release; when
576 + this happens, an upgrade path will
    will be automatically
```



lavalamp added a note on Oct 15, 2015

s/documentedr/documented/

Add a line note

```
577 + converted to the new version; in c
    necessary; a manual
578 + upgrade may require downtime for a
    may require
579 + manual conversion of objects to th
    necessary, the
580 + project will provide documentation
    conversion
581 + tips](../api.md))
582 + - Cluster Reliability: since the fea
    via a flag should not
583 + create new bugs in unrelated featu
    have minor bugs
```


観測した手法 - 2

GitHub 上で Bot が 書式に沿え と突っ 込んでくる

恨まれやすいことは bot にやら
せると、恨みが積もりにくい??

GordonTheTurtle commented 4 hours ago

If you are reporting a new issue, make sure that we do not have an existing issue for this problem. Please ensure this by searching the issue list for this repository. If there is an existing issue, please add a comment to the existing issue instead.

If you suspect your issue is a bug, please edit your issue description to include the following INFORMATION shown below. If you fail to provide this information, we will close your issue and will close it. We will, however, reopen it if you later provide the information.

For more information about reporting issues, see [CONTRIBUTING.md](#).

You *don't* have to include this information if this is a feature request.

(This is an automated, informational response)

BUG REPORT INFORMATION

Use the commands below to provide key information from your environment:

```
docker version :
```

```
docker info :
```

Provide additional environment details (AWS, VirtualBox, physical hardware, etc.):

List the steps to reproduce the issue:

- 1.

観測した手法 - 3

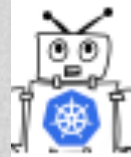
GitHubにいるBOT がテストしたりライセ ンス同意を求めてくる

恨まれやすいことは bot にやら
せると、恨みが積もりにくい??



janetkuo commented 12 minutes ago

@k8s-bot unit test this issue: #IGNORE (test didn't start)



k8s-bot commented 12 minutes ago

GCE e2e test build/test **passed** for commit [b341df4](#) .

- [Build Log](#)
- [Test Artifacts](#)
- [Internal Jenkins Results](#)



Some checks haven't completed yet

1 failing, 2 pending, and 2 successful checks



✗ Jenkins unit/Integration — No test results found.



● Submit Queue — Github CI tests are not green.



● continuous-integration/travis-ci/pr — The Travis CI b



✓ Jenkins GCE e2e — 235 tests run, 105 skipped, 0 failed



✓ cla/google — All necessary [CLAs](#) are signed



This branch has no conflicts with the base branch

Only those with [write access](#) to this repository can merge p

まとめ

- ・ 開発に用いた基盤システムGKEなどについて説明した
- ・ 開発に用いた方法、そのツールについて説明し、利点と課題について説明した
- ・ 他、観測した開発手法と予想できた利点について、紹介した