社内の Platform を作る取り組み

OPEN CLOUD INNOVATION FESTA 2016

Kodai Sakabe @koudaiii Wantedly,Inc

ABOUT ME

— Software Engineer —

坂部 広大 (KOUDAI SAKABE)

2010/4- TIS

2015/8- Wantedly

@koudaiii

Site: http://koudaiii.com

WANTEDLY

ー シゴトでココロオドルひとをふやすー


「はたらく」を面白く
Wantedlyは、運命のチームや仕事に出会えたり
人脈をひろげ、情報収集に使えるビジネスSNSです

WANTEDLY/TOOLS

WANTEDLY / FEED
世界に自分の声を発信しよう

WANTEDLY/JOURNAL
はたらくを面白くするメディア

Launch New!!

話を聞きに行きたい

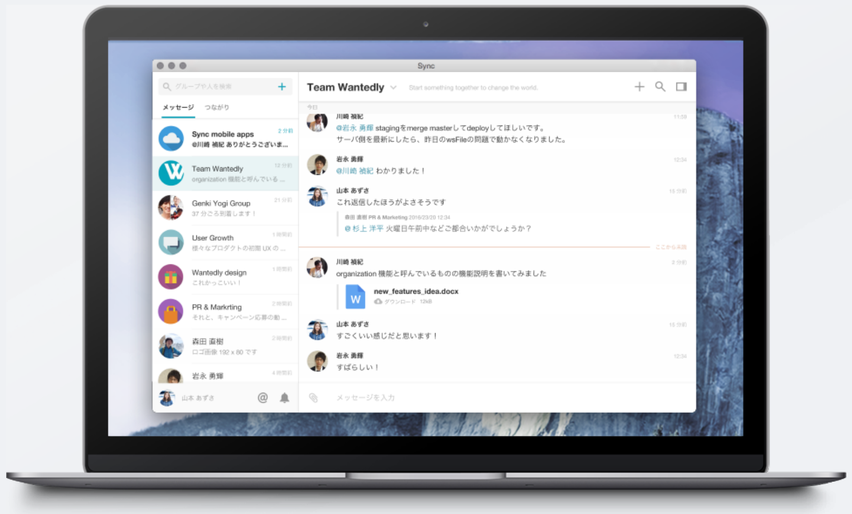
SYNC by WANTEDLY

SYNC by WANTEDLY

@メンションで宛先指定

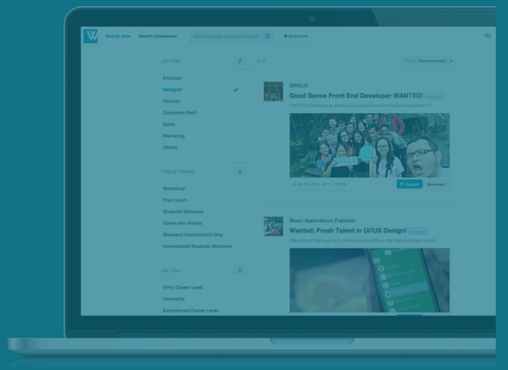
ファイル共有・一覧

メッセージ全文検索



Wantedly Open API

はたらくを面白く」をテーマにこれまでサービス開発を展開してきた
Wantedlyがその価値をWantedly.comドメイン以外にも広げていきます。
世の中の一人でも多くの人が、はたらくを面白く」感じられる世の中になりますように。

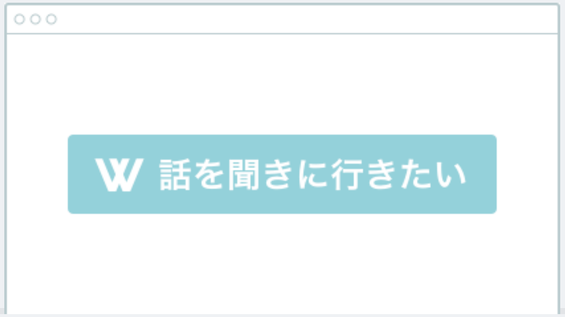


できること

会社フィードボックス

Wantedly上にある「会社フィード」を、ブログやコーポレートサイトなど、好きなページに簡単に埋め込むことができます。手軽な採用ブランディングや情報発信に利用できます。

詳しく



話を聞きに行くボタン

自社サイト等の募集要項に、「話を聞きに行きたい」ボタンを設置し、Wantedly上の「遊びにいく体験」を自社サイトでも展開できます。

詳しく

フォーム自動入力ボタン

採用管理システム提供企業様等への特別提供APIです。候補者が募集にエントリする際に、Wantedly上のプロフィール情報を活用してエントリが可能になります。

詳しく

WANTEDLY/JOURNAL

はたらくを面白くするメディア

シゴトでココロオドルってなんだろう？

Wanteldyは、シゴトでココロオドル人をふやしたいと考えています。
では、”シゴトでココロオドル”とはどういう状態なのでしょう？

お客さんに「ありがとう」と言われたとき
目標の売上を達成したとき
チームメンバーの笑顔を見たとき…

それは、人によって違うでしょうし、
時代や国によってもまったく異なるでしょう。

今回リリースする「WANTEDLY JOURNAL」では、
Wantedlyを通して転職した方や、普段あまり見たことがない企業への
インタビューを通して、”シゴトでココロオドル”とは何なのかを探求します。

シゴトを探している人も、そうでない人にとっても、
「WANTEDLY JOURNAL」が働き方のヒントになれば幸いです。

WANTEDLY / FEED

世界に自分の声を発信しよう

CASE by Wantedly

Feed | Projects

Add Project | Search ...

All Projects | Featured

DON'T TOUCH!

Bascule

Bascule Inc. Team

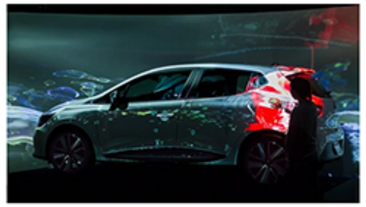
9 Likes 367 Views

AppleWatch PSD MockUp

3D , CGデザイン , AppleWatch , 無料ダウンロ..

Super Crowds Inc. Team

12 Likes 199 Views



Touch! NEW RENAULT DESIGN

Event , Installation , Movie , Projection-Mappin...

1→10design, Inc. Team

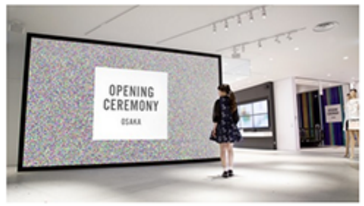
7 Likes 167 Views

Bridgestone RUN & STOP

Branding , Event , Installation , Car , Game , Sig...

1→10design, Inc. Team

5 Likes 88 Views



OPENING CEREMONY OSAKA

Branding , Space , デジタルストア , Shop/Mall

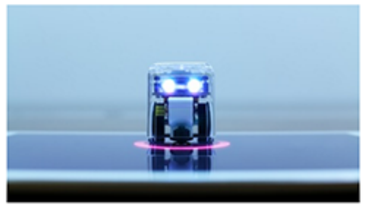
teamLab Team

5 Likes 157 Views

417 EDIFICE / SLOBE IENA

vision track Team

4 Likes 58 Views



TABO

Bascule

Bascule Inc. Team

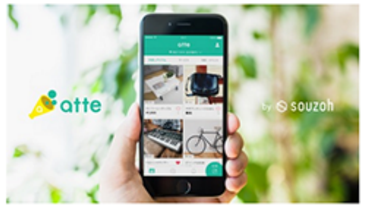
3 Likes 69 Views

TOYOTA FCV PLUS

Interactive , PARTY , Motor-Show , Hidrogen-V...

PARTY Team

2 Likes 57 Views



TRENDING TAGS

Branding, Event, Movie, Promotion, Web, Mobile, iOS, UI/UX, Android, UI

社内の Platform を作る取り組み

WHY

大胆に美しく、スマートに

「すごい」が自然と出てくる場を支える

HOW

技術的に優位性のあるインフラ

さくっと 1000 台建てれる・新しい物がすぐに出せる

WHAT

社内の Platform を作る

エンジニアが困っていることをなくす。ココロオドルことをやる。

Wantedly Way

User First / Code Wins Arguments / Simple is Not Easy

User First

- 「もっと速い馬車がほしい」というユーザの声を聞き続けるのではなく、「速く移動したい」というユーザの本質的な欲求に応える行為
- 「言葉 < 行動」
    - ユーザーテストで3人間違えるようだったりまごついたりしたら改善して、もう一回試してもらって違う人にも試してもらう

Code Wins Arguments

- チームで1時間 MTG するなら code を書いて検証しよう
- エンジニアの場合
  - 限られた情報で決断をして、前に進んでいるか
  - 仮説をあれこれ考えるより、まずプロトタイプを作って早く学習しているか
- 営業の場合
  - 自分で良い機能が浮かんだからエンジニアに頼むのではなく
  - 企画書を書いて会社を回って、5社程度を確約して、これなら行けると思ったらエンジニアに頼むことが出来るか

グロース基盤となるツール、仕組みを作る（プチじゃないハッカソン） #25522

Edit New issue

Open **south37** opened this issue 16 days ago · 15 comments

south37 commented 16 days ago • edited

WHY

Wantedly でも、グロースを加速させるために基盤を用意したい
cf. アイディア集 #25476

WHAT

グロース基盤となるツールや仕組みを作る。2人一組の「ペア」で、1週間で作る。
解決したい課題感やアイディアは #25476 にまとめたが、「取り組む内容」は「ペア」で決める。

日程

- 今日（8/31 水）- 来週（9/6 火）までの1週間
  - 時間は短いので、「1週間で出来るもの」という観点も入れて「取り組む内容」を決める
- 最終日（9/6 火）にはデモをする
- 一ヶ月後（10/5）に、どのくらい使われててどのくらい効果が出たかをチェックする

Pipeline
やるかも

Projects
None yet

Labels
Growth

Milestone
No milestone

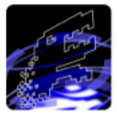
Estimate
No estimate yet

Assignees
No one—assign yourself

BigQuery フロント用のインスタンスを立てたい #1304

Edit New issue

Open **Altech** opened this issue 15 days ago · 7 comments



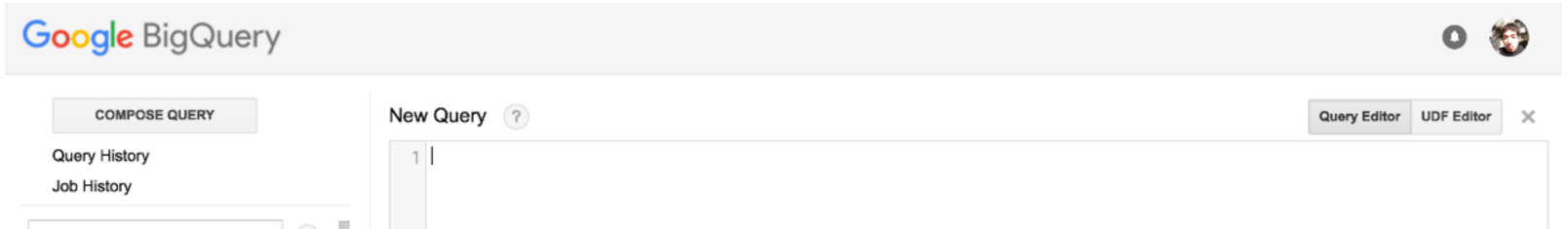
Altech commented 15 days ago • edited

WHY

BigQuery のデフォルトの WebUI は割と最小構成の作りになっていて、チームで使うのが難しい。

- 基本的にジョブとかクエリが全部アカウントに縦割りで紐付いてて見えない
  - なので、ジョブとかクエリをイシューに貼ることができない（！）
- 今のWebUIでは、UDFを各クエリ間＆各ユーザー間で使い回すことができない

一方、APIとライブラリ自体はかなりきちんと作られているので、基本的にこれベースで自前でフロントを用意したい。



Pipeline
やるかも

Projects
None yet

Labels
None yet

Milestone
No milestone

Estimate
No estimate yet

4日間で結果を出す「グロース基盤開発ハッカソン」

- https://www.wantedly.com/companies/wantedly/post_articles/35813
- 2人1組 ４チーム
- Rails / Node / React と好きな言語で新しく repository を作って始めたり
- AWS DynamoDB / ElastiCache

Simple is Not Easy

- 作る側が使う側の事を徹底的に考え抜いているか
- 足すよりも引いてフォーカスをしているか
- 効果が高い施策(効果が低いとその掛けた工数分でもっと有効な改善はなかった？)
- 仮説をたてて、その仮説のコアとなる部分だけにフォーカスできているか

etc.

- Demo Day => 金曜日に自分で作ったもの/施策を全社員の前で発表する場(10分のLT)
- 負債返済日 => デザイナーも含めてエンジニア全員で丸1日技術的な負債に取り組む日
- CMW => Can / Must / Want お互いのコミュニケーション
- 1 on 1 => リーダーとのコミュニケーション
- Expectation => 期待値設定
- biz_question => ビジネス側からの質問をエンジニアが持ち回りで回答する

Infrastructure

社内の Platform を作る

2014年08月09日より Heroku から AWS へ

Xaas と AWS の両方を使う

- 利用できるものはどんどん利用する
  - スピード感
  - コスト感
  - 環境の選択(build pack だとやりにくい)
  - ネットワーク
- Heroku や Redis や Firebase と AWS 組み合わせ

Bottleneck

エンジニアの増加、サービスの増加

WANTEDLY

SYNC by WANTEDLY

WANTEDLY/JOURNAL
はたらくを面白くするメディア

WANTEDLY / FEED
世界に自分の声を発信しよう

Developer

タスク依頼

インフラチーム

DATADOG

heroku

amazon web services

redis

logentries™

dnsimple

SendGrid

Wantedly Infrastructure Way

- Code Wins Arguments
  - 変化に強いインフラを
- User First
  - Developer がいつの間にか使っているものを
- Simple is Not Easy
  - 場当たり的な対応ではなく、根本解決を

コード化と自動化

・インフラを操作する Tool / 社内で便利なCLI&GUI / 新しい技術基盤

WANTEDLY

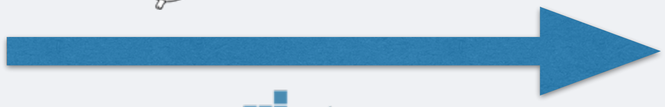
SYNC by WANTEDLY

WANTEDLY/JOURNAL
はたらくを面白くするメディア

WANTEDLY / FEED
世界に自分の声を発信しよう

API and Tool

Developer



インフラチーム

DATADOG

heroku

amazon web services

redis

logentries™

dnsimple

これまでの取り組み

- 便利ツール
- sap / deploy-server
- 二段階ビルドによる deploy 高速化 / blue-green deployment
- Terraform
- CoreOS
- cell
- replace

便利ツール

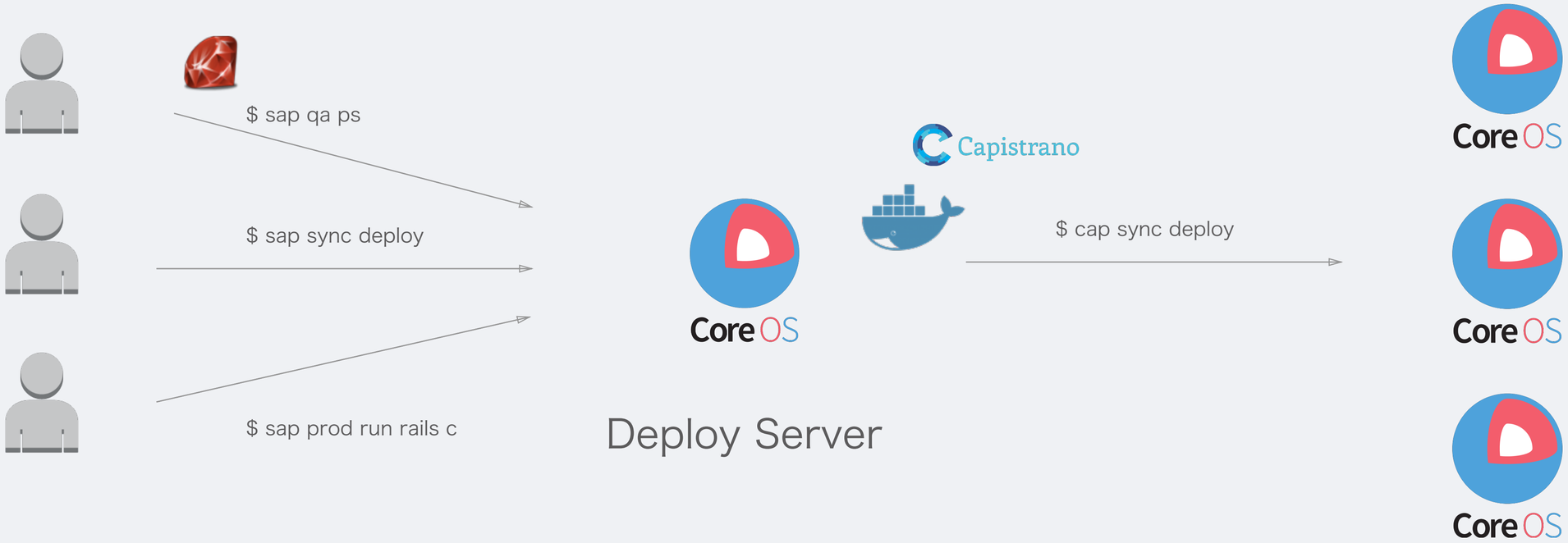
```
$ s3url –help #期限付きURLを発行
Usage of s3url:
   s3url https://s3-region.amazonaws.com/BUCKET/KEY [-d DURATION]
   s3url s3://BUCKET/KEY [-d DURATION]
   s3url -b BUCKET -k KEY [-d DURATION]

$ ec2c –help # vagrant で試したものを instance で建てる / tag を付けれる / ip を確認できる


Available commands are:
   launch           Launch new EC2 instance
    list             List EC2 instances
    tag              Tag EC2 instances
    terminate        Terminate the specified EC2 instance
    untag            Tag EC2 instances
```

sap / Deploy Server

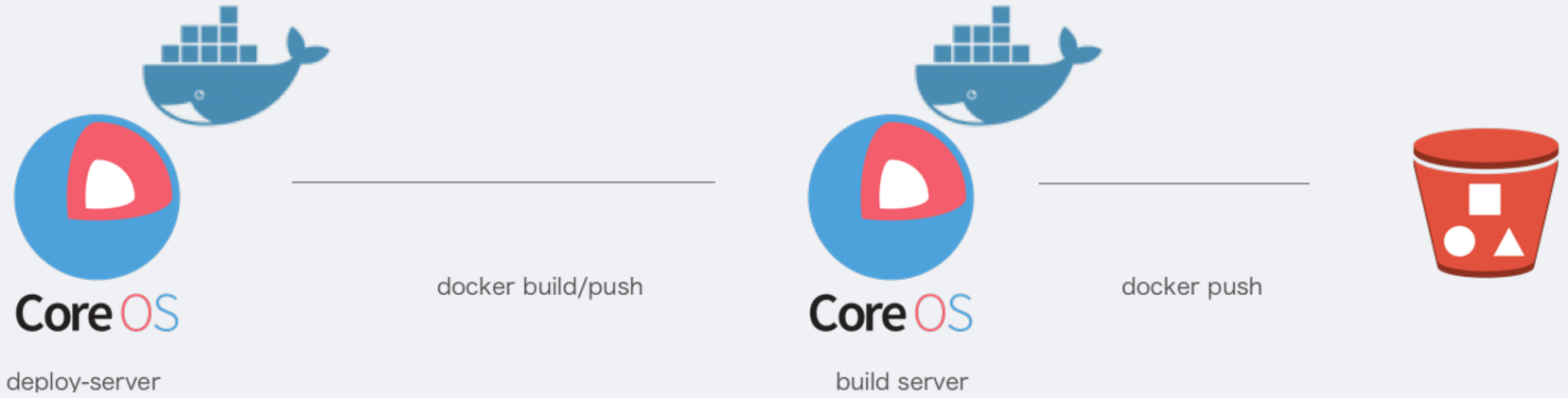
- 必要な様々な鍵や環境変数を個人で設定しなくてよい
- Deploy 方法が変わるようなインフラの構成変更時にも、各エンジニアは手元のアップデートをする必要がない
- Deploy 時の各個人の作業がコンテナのログとして残る



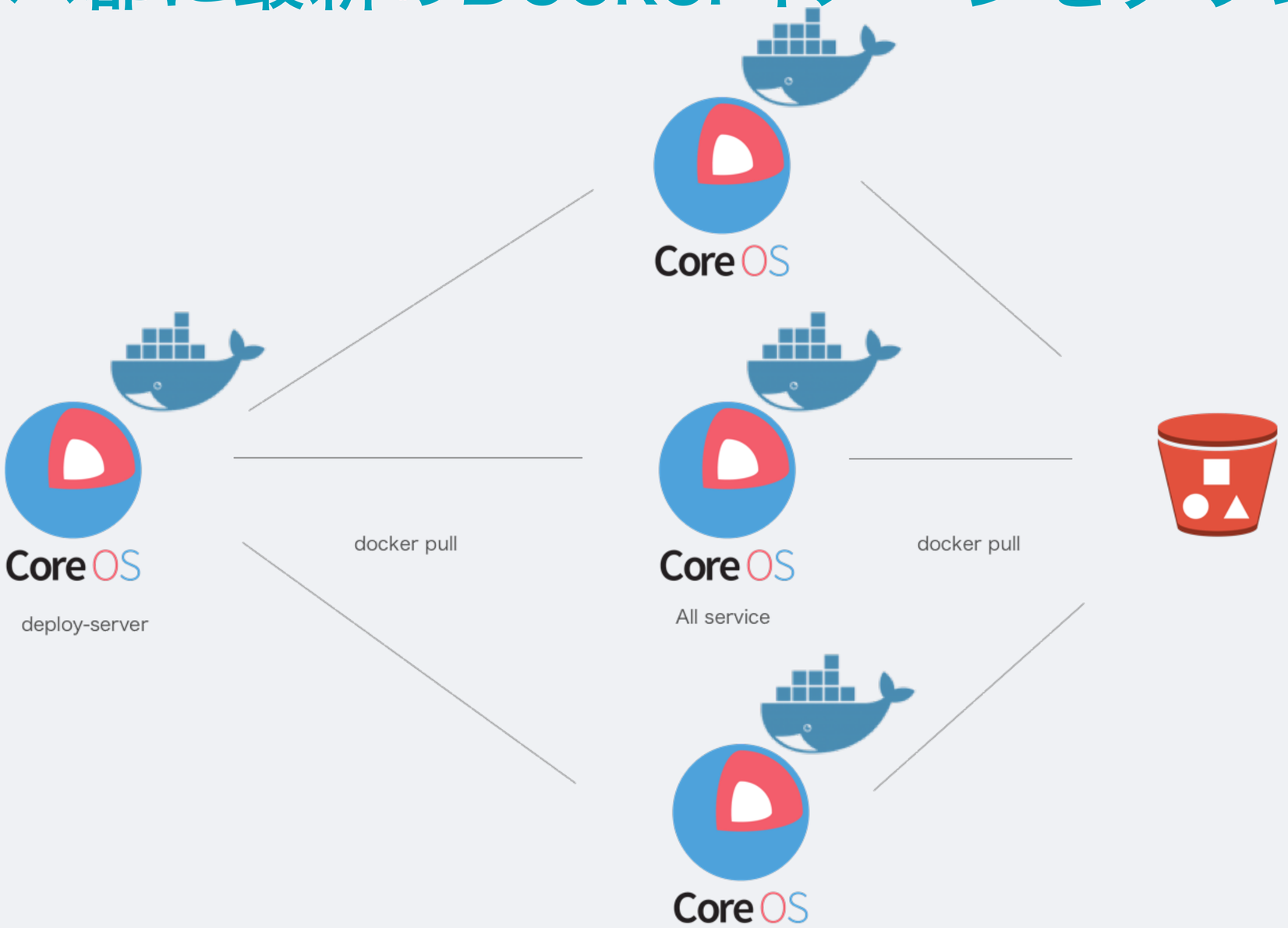
http://dev.classmethod.jp/devops/dev-ops-wantedly/

二段階ビルドによる Deploy 高速化

- ビルドサーバでDockerイメージをビルド
- プライベートDockerレジストリ(S3)へアップロード

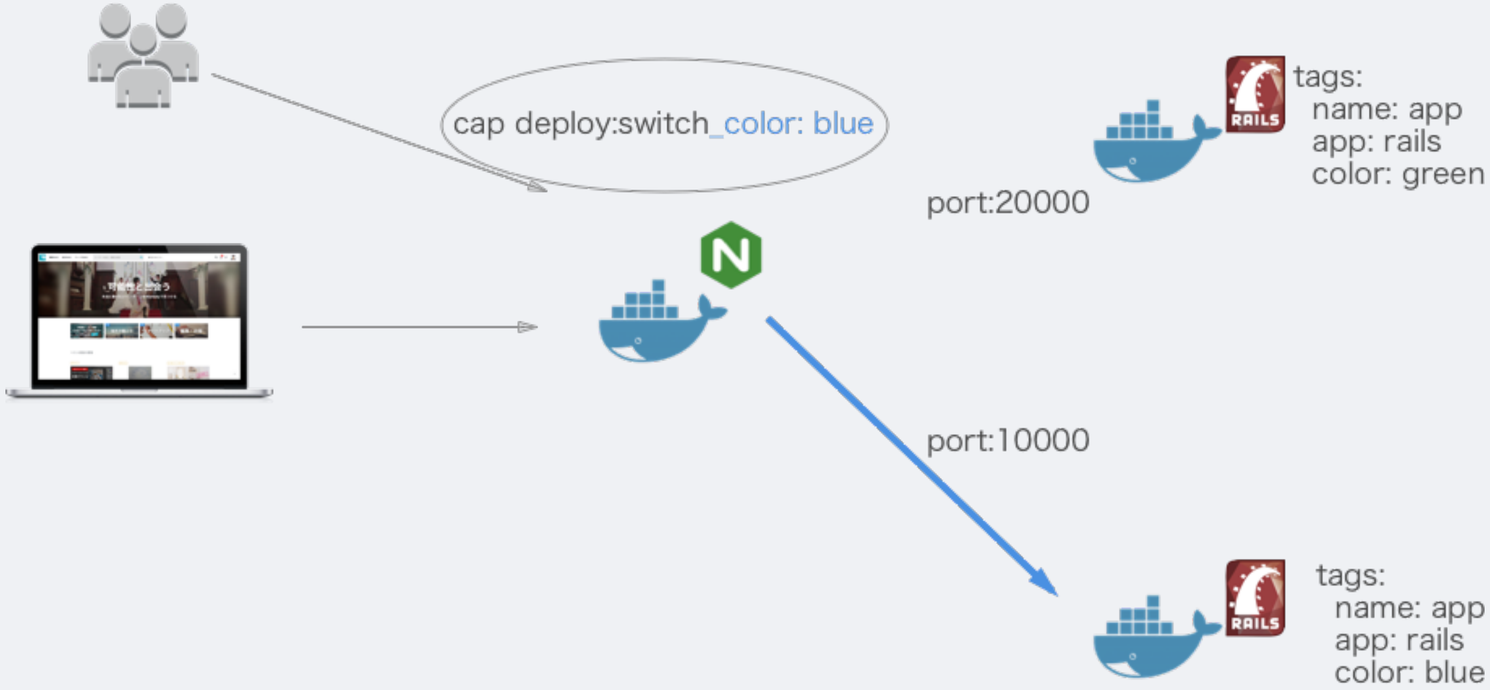


本番サーバ郡に最新のDockerイメージをダウンロード



blue-green deployment

- standbyで古いコンテナを削除し新しいコンテナを起動
- /healthcheck が成功したら switch color でLoad Balancerを切替



Deploy Flow

1.ビルドサーバで指定したブランチを GitHub から取得

2.ビルドサーバで予め作成されたイメージを元にDockerイメージを再ビルド

3.プライベートDockerレジストリ(S3)へアップロード

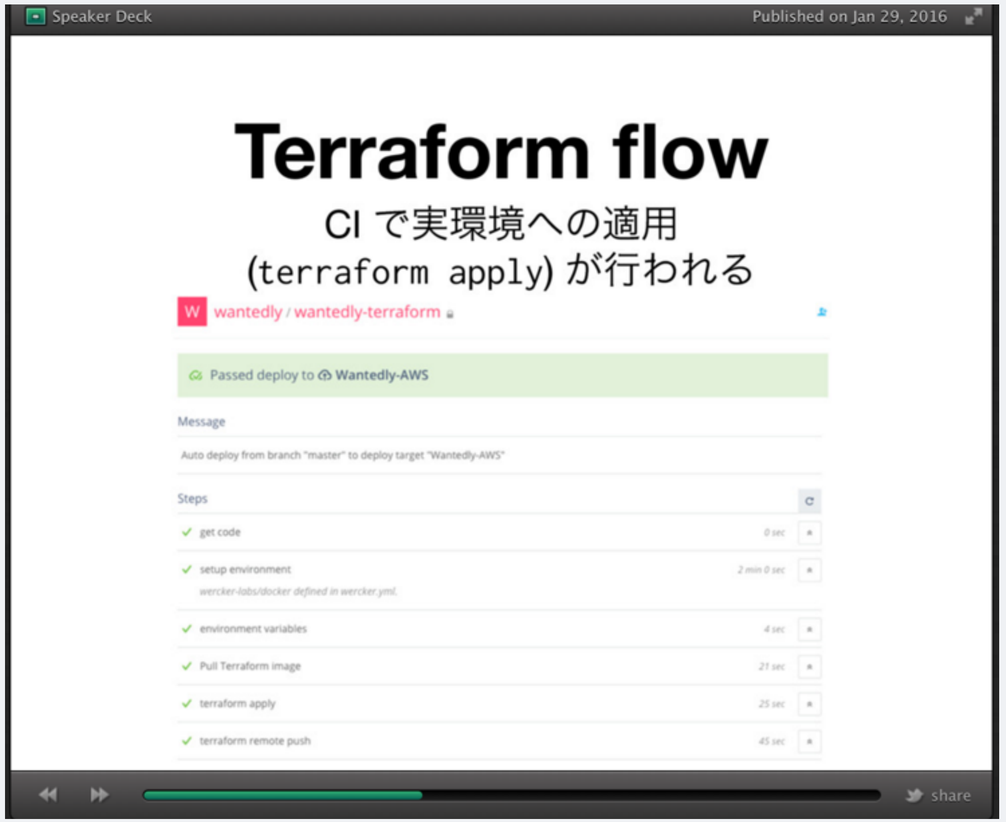
4.本番サーバ郡に最新のDockerイメージをダウンロード

5.本番サーバ群で新しいコンテナを起動し、healthcheck を通す

6.古いものと切替

Terraform(Pull Request でインフラの申請)

- AWS(S3/RDS/ELB etc) / dnsimple / datadog を操作



https://speakerdeck.com/dtan4/terraform-at-wantedly-tech-circle-number-12

CoreOS

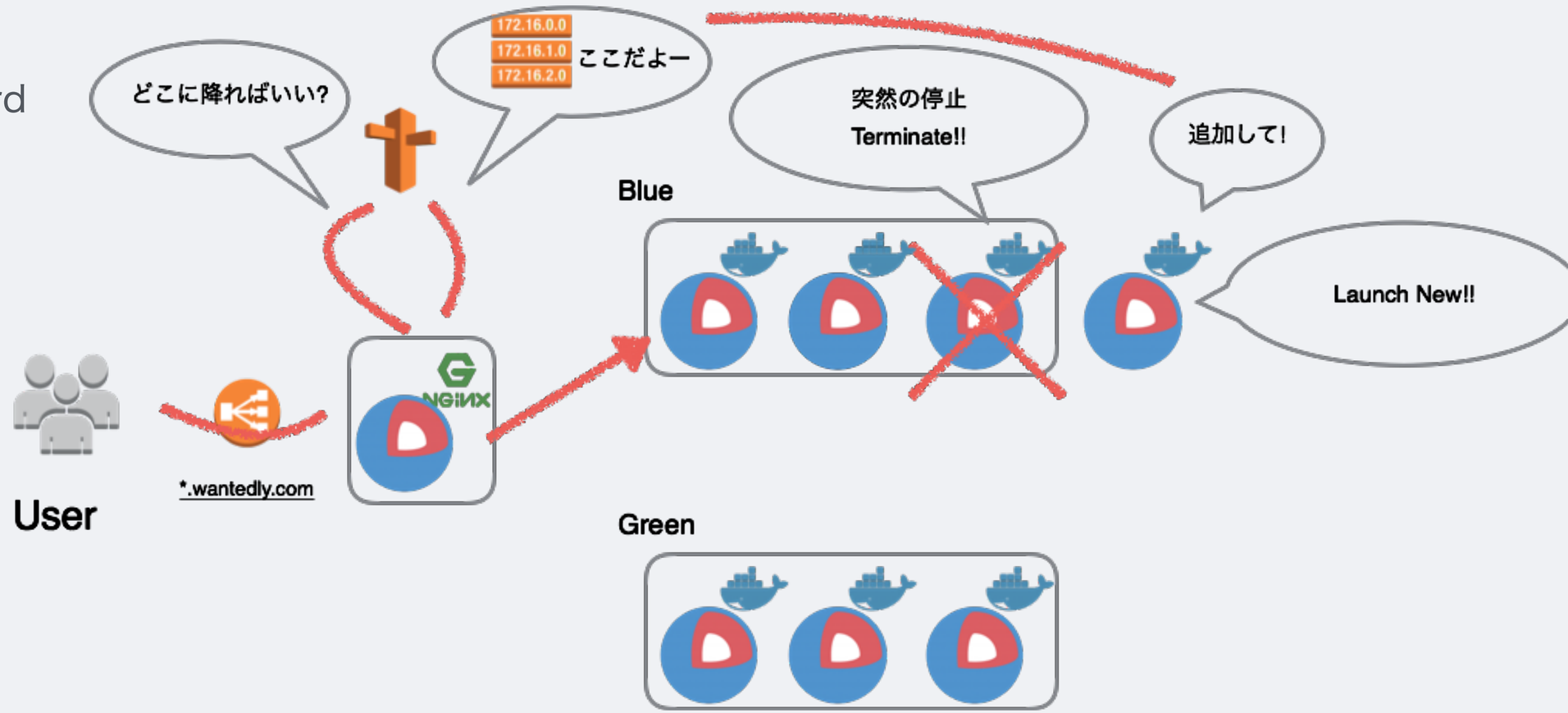
- Ubuntu を利用していた時、Docker や Registry の version up が大変
- CoreOS は docker が入っている Official ami がある
  - blue-green による OS version up
  - Systemd
  - Host 上で作業が出来ない
- すべてのサーバーを CoreOS に変えた

AWS の TAG と AutoScaleGroup

- すべての instance の特定は、 AWS の TAG で定義
  - { app:wantedly, role:web, deployment:bule-green }
- 何台動かしたいか？ => ASGにインスタンス数を定義
- 必要なインフラツールは Launch Config で定義
  - instance が立ち上がると当時に TAG を確認し、必要な service を systemd から立ち上げる

cell

- AWS TAG を元にサービスを特定し systemd に登録
- 自動で Service-in する
- Add/Delete DNS Record



replace command

- OpenSSL や ImageMagick 等の脆弱性は年に何回か発見される
- Instanceをすべて入れ替えるのは工数が掛かる
- 同じような作業はコードにして、自動化させる
- サービスごとにコマンド一つですべてのインスタンスを入れ替える
- 例: sap sync replace

Clustering 複数の instance から大きさリソースを実現

Kubernetes

WHY

- サービスの分割や新規サービスを出す場合、インフラチームのやることが多い
- インフラの安定化により複雑化
- 同じような作業も多い

Capistrano

CoreOS

Internal DNS

Load Balancer

Auto Scale

Web

Application

Monitoring

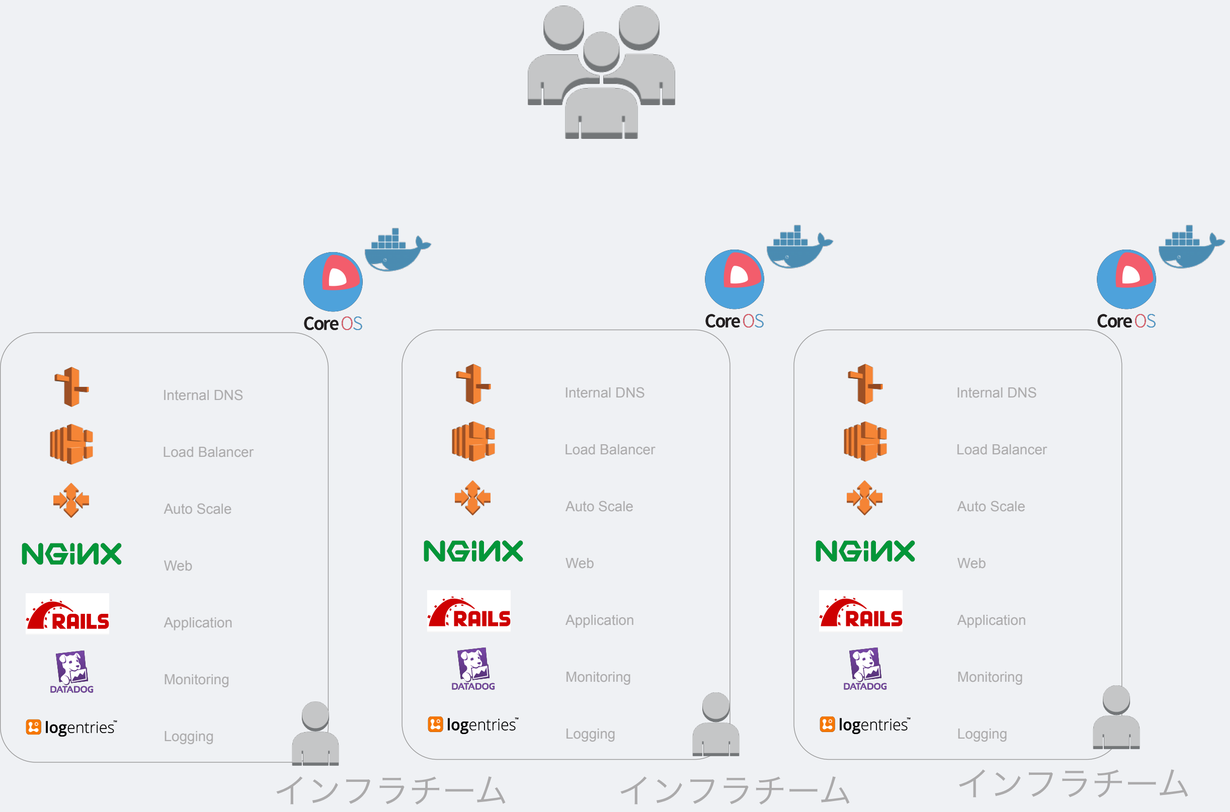
Logging

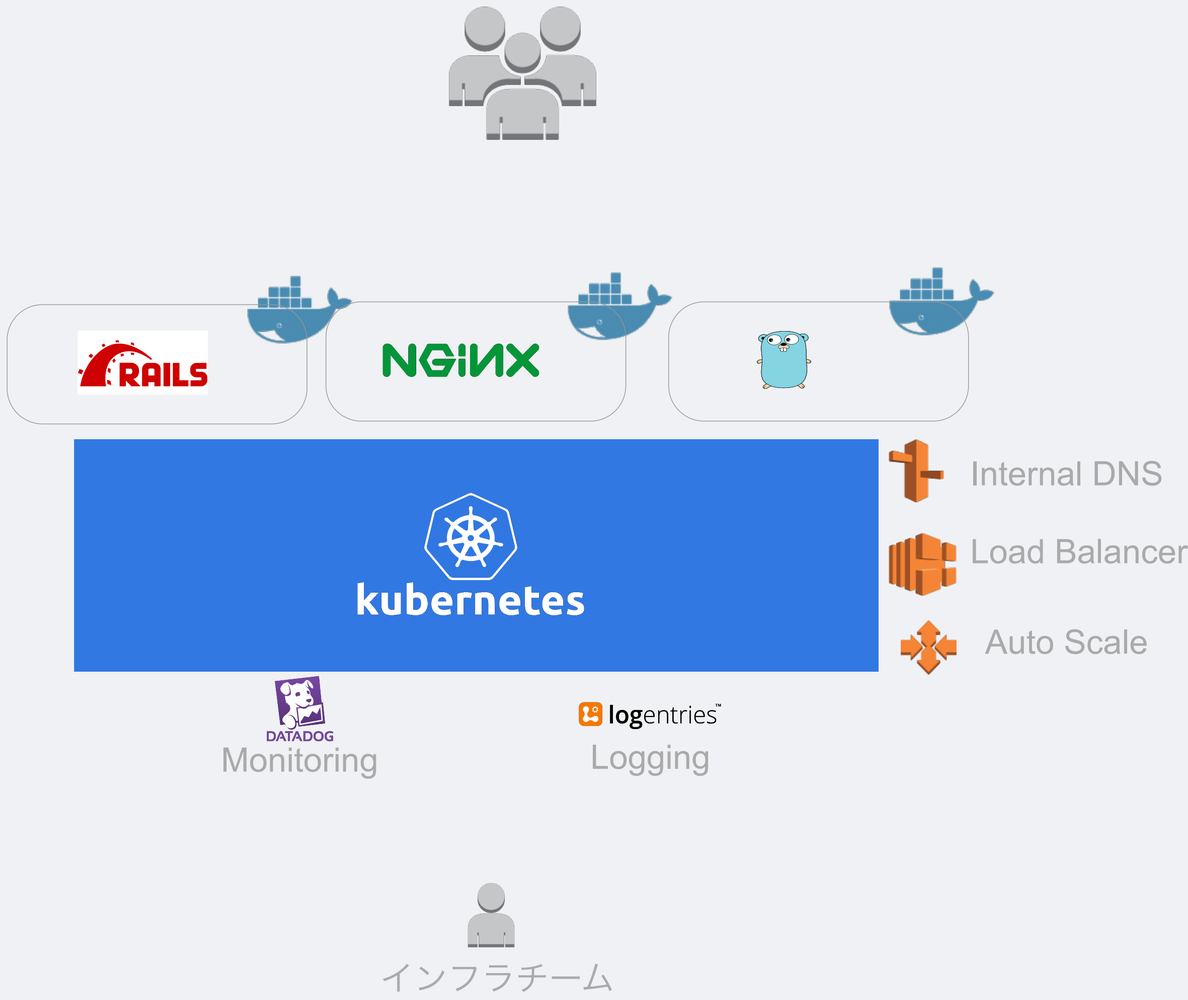
クラスタリング

- Application を動かせるインフラをすぐ作ってくれと言われた時に作れる
  - 1回限りのすぐ終わるタスクでも同じインターフェースで出せる
- エラーがあったとしても、別のところで走らせ直せる
- 一つの大きなリソースとして動くため、今まで行ってきていたMonitoring用にコンテナを設定して、次はASG用にマシンを立てて、次は...みたいのようなインフラ作業ことがない

「何が何処で動いているかを意識していなければならない世界」から「何が動いているかだけ意識していれば良い世界」

クラスタリング

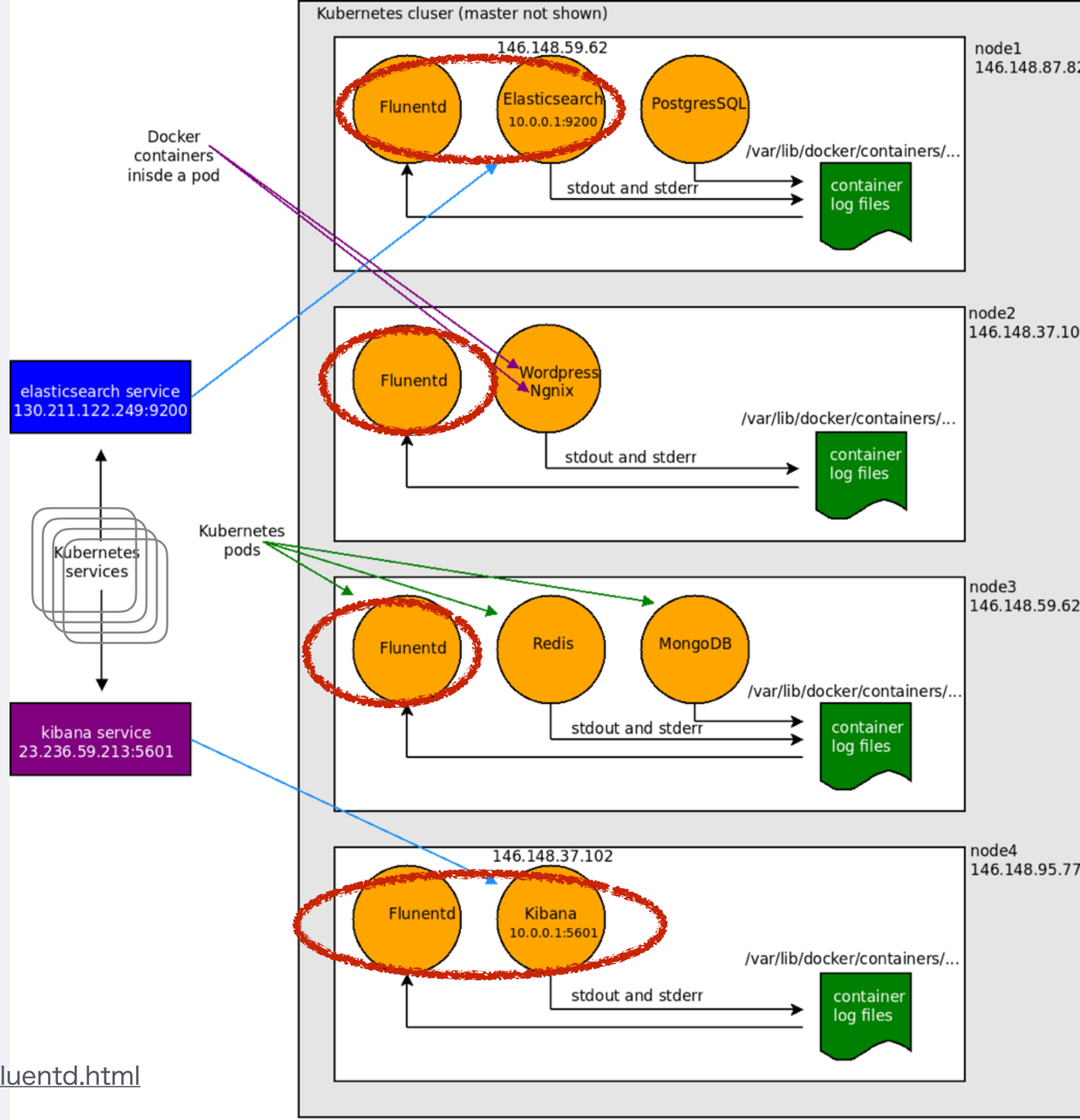




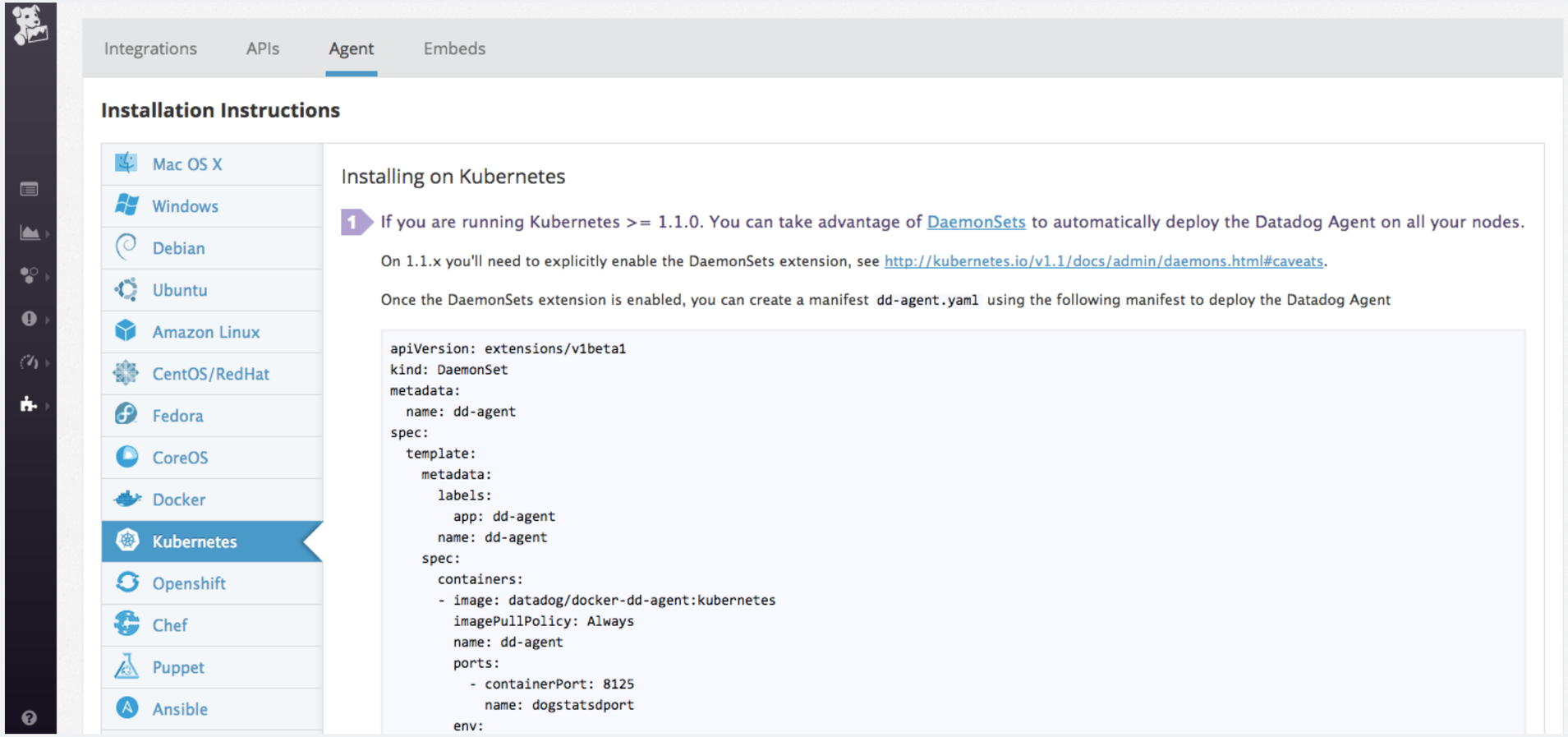
「何が何処で動いているかを意識していなければならない世界」から「何が動いているかだけ意識していれば良い世界」

Kubernetes

- Node = instance
- 複数の instance を一つリソース
- LoggingやMonitoringのような各Nodeに配置したいものもAPIを通じて実現
- Developerは必要なSpecを定義し、APIを呼び出す



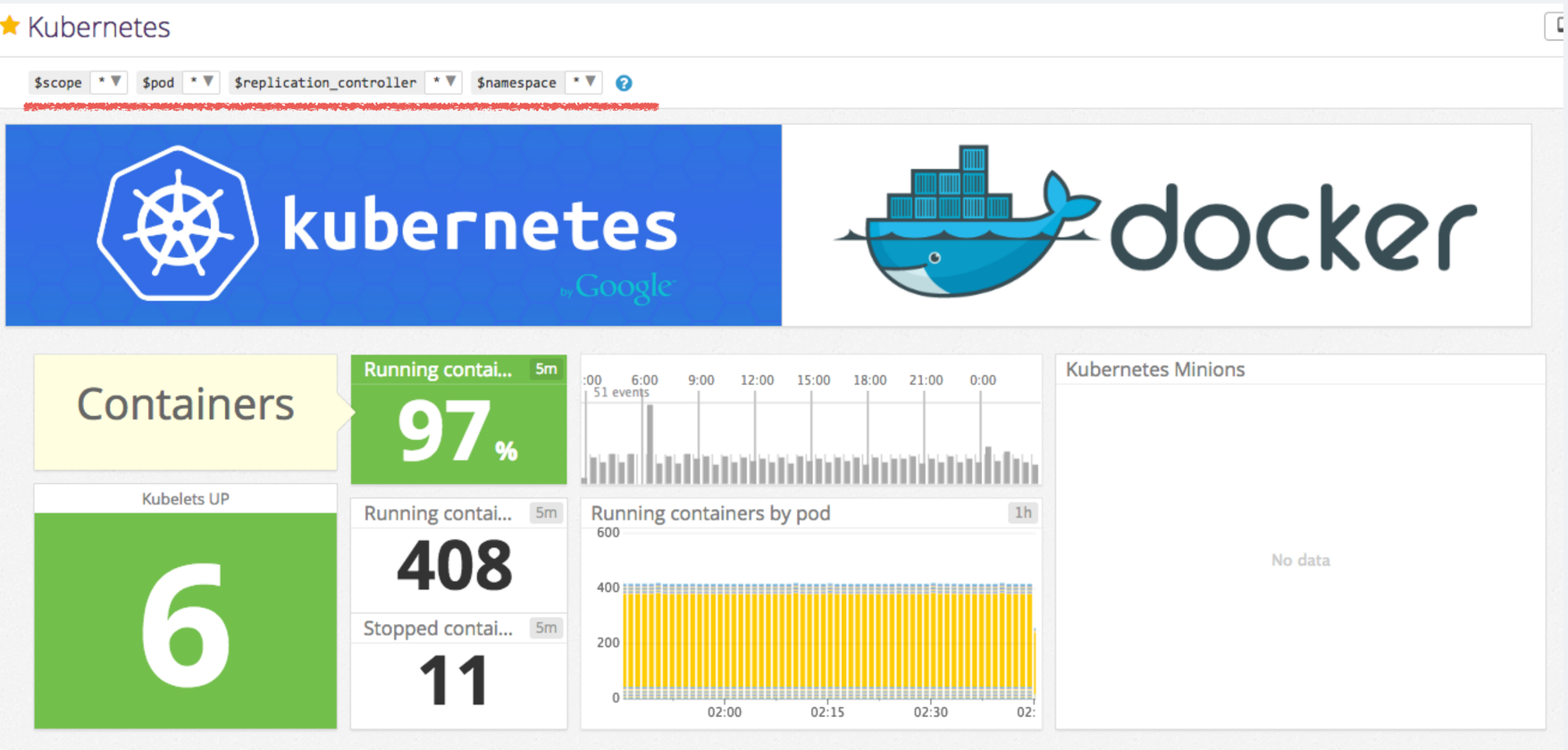
- http://blog.raintown.org/2014/11/logging-kubernetes-pods-using-fluentd.html

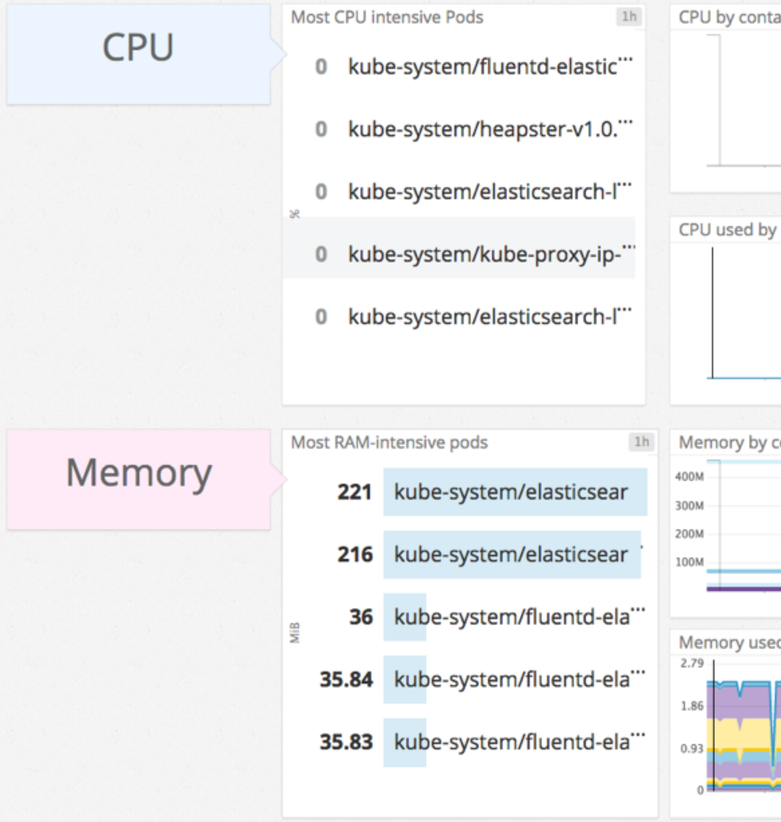


Monitoring by Datadog

DaemonSet(各 Node に対して配置する)

Cluster に対して一度manifest file を送ればそのClusterすべてのMonitoringが出来る





Dashboard

$scope $pod $rc/rc $namespace で絞り込み/Alert設定

Kubernetes 速習会

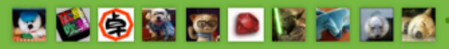
kubernetes 102

投稿を編集

koudaiiiが2016/08/18にKobitoから投稿(2016/08/23に編集) • 編集履歴(10) • 問題がある投稿を報告する

37 ストック | 0 コメント | 700 Views

ストック済み



この速習会の目標

Developer がさくっと kubernetes を理解して、実際にリリース出来る

- kubectl が叩ける用になる
- Kubernetes の manifest ファイルの書き方
- Kubernetes への Release / Deploy 方法
- Kubernetes 上にあるApplication の maintenance
- Kubernetes に上げたApplicationのMonitoring

koudaiii
344 Contribution

人気の投稿

- Wantedlyを支えるモニタリング
- Varnish入門と仕組み
- チェックリストはなぜ軽視されるのか！？

Kubernetes 速習会

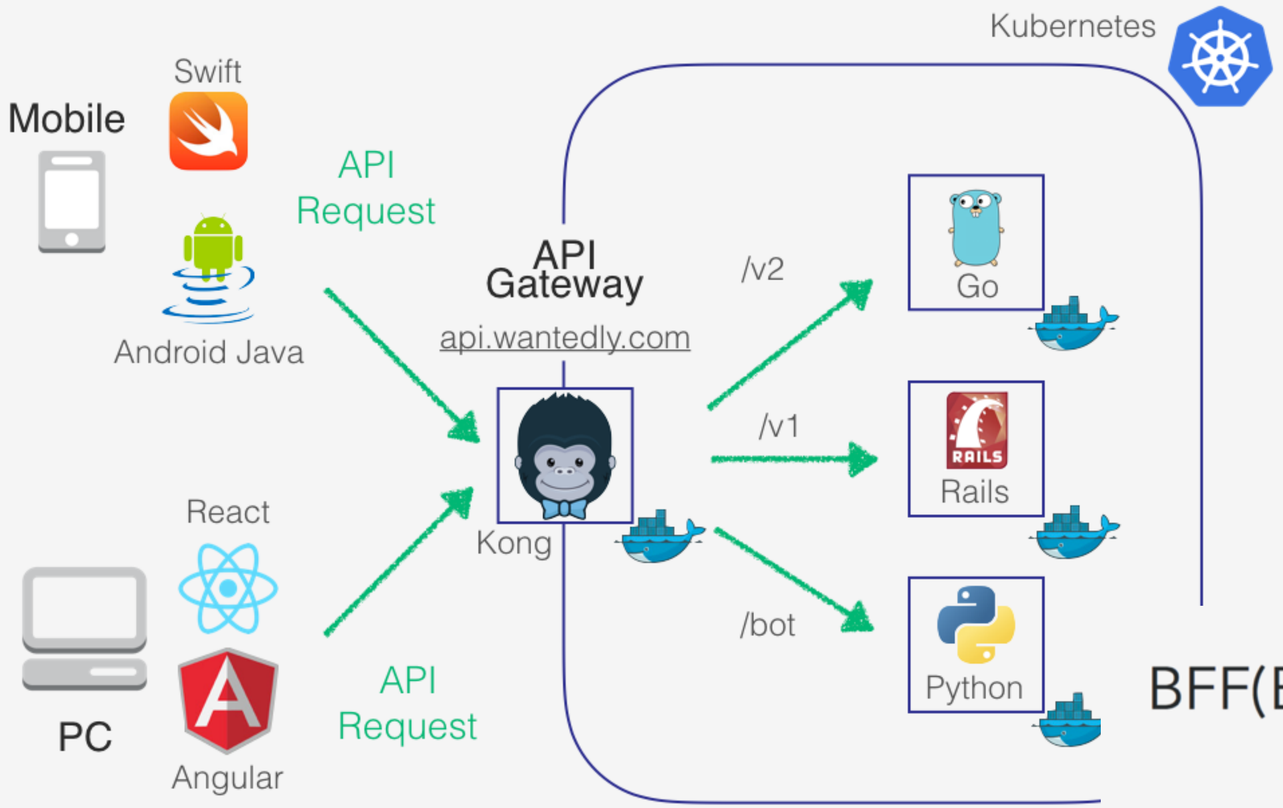
Developer がさくっと kubernetes を理解して、実際にリリースできる

PROJECT

— API and kube CLI—

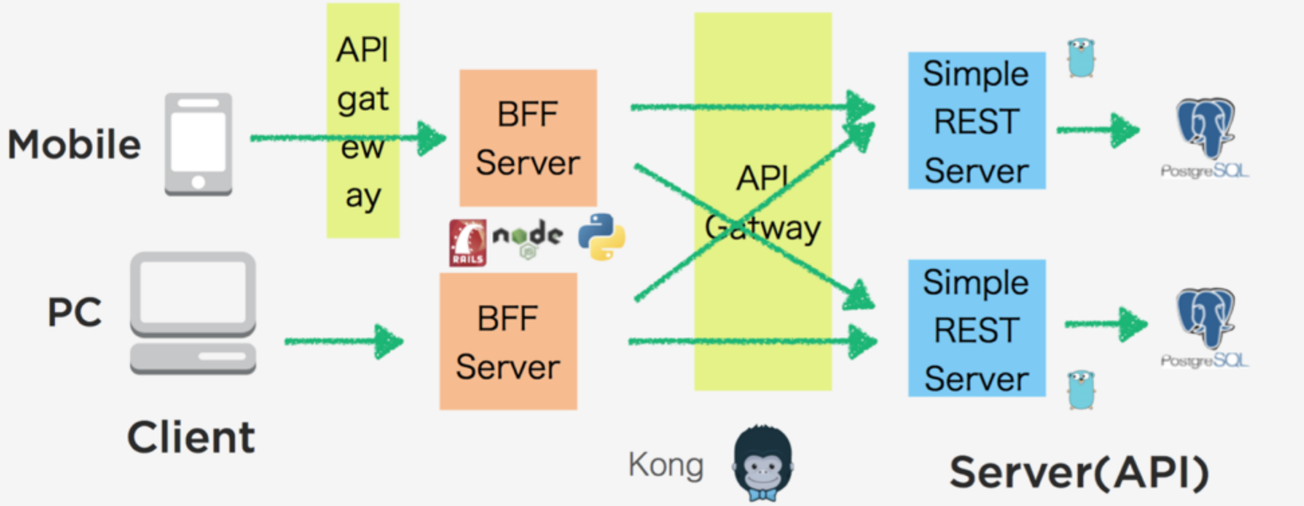
Microservices完成形

WANTEDLY



BFF(Backends for Frontends)

WANTEDLY



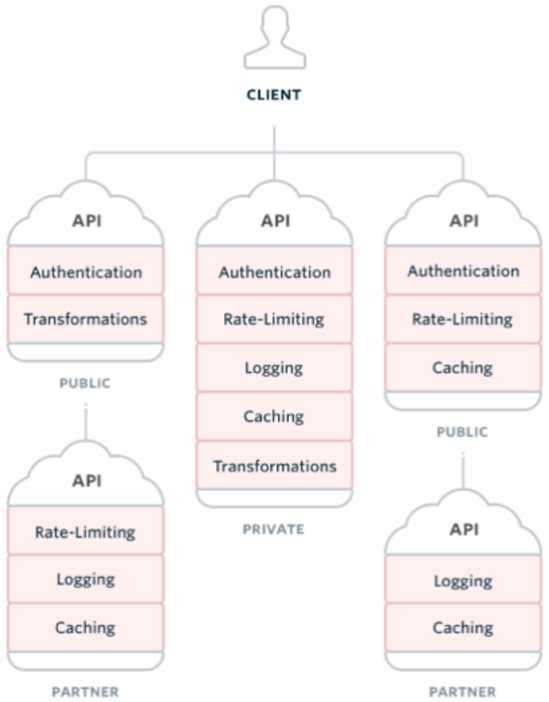
https://speakerdeck.com/awakia/yoriyoiapiwozuo-rutameni

Kong Architecture

- Kong Clustering
- plugin
  - OAuth2
  - ACL
  - IP Restriction
  - Rate Limit
  - Request Size limit
  - Response late limit

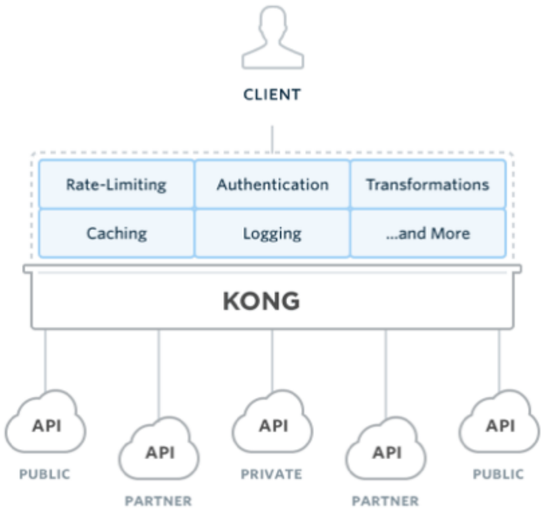
Orchestrate Common Functionality

Legacy Architecture



- ✖ Common functionality is duplicated across multiple services
- ✖ Systems tend to be monolithic and hard to maintain
- ✖ Difficult to expand without impacting other services
- ✖ Productivity is inefficient because of system constraints

Kong Architecture



- ✔ Kong centralizes and unifies functionality into one place
- ✔ Build efficient distributed architectures ready to scale
- ✔ Expand functionality from one place with a simple command
- ✔ Your team is focused on the product, Kong does the REST

API Generator

- wantedly/apig
  - Golang RESTful API Server Generator
- wantedly/pq2gorm
  - Generate gorm model structs from PostgreSQL database schema

README.md

apig: Golang RESTful API Server Generator

build passing

apig is an RESTful API server generator.

- Input: Model definitions based on gorm annotated struct
- Output: RESTful JSON API server using gin including tests and documents

Contents

README.md

pq2gorm - Generate gorm model structs from PostgreSQL database schema

build passing

pq2gorm is a generator of gorm model structs from a PostgresSQL database.

- Input: Connection URI of a PostgresSQL database.
- Output: Model definitions based on gorm annotated struct.

How to build and install

Prepare Go 1.6 or higher. Go 1.5 is acceptable, but `GO15VENDOREXPERIMENT=1` must be set.

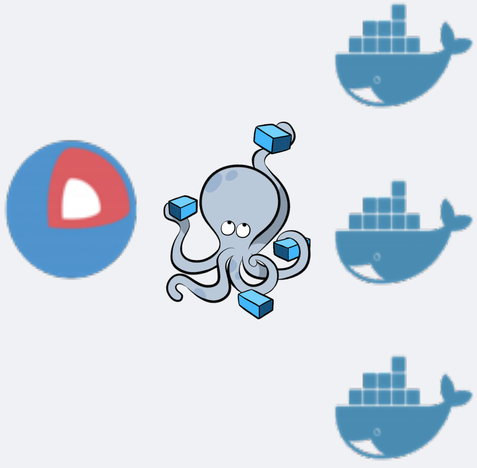
After installing required version of Go, you can build and install `pq2gorm` by

```
$ go get -d -u github.com/wantedly/pq2gorm
$ cd $GOPATH/src/github.com/wantedly/pq2gorm
```

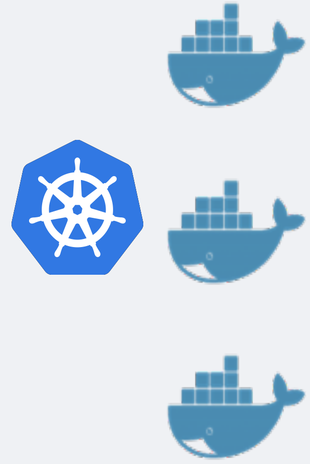
Kubernetes for Wantedly

Developer

kube Server



Kubernetes



kube Commands:

- kube deploy
- kube migration
- kube diff
- kube rollback
- kube enable dotenv
- kube enable private-repogitory quay
- kube generate
- kube tags
- kube (kubectl subcommands)

Container Tools:

- kubectl
- k8sec
- slack notification(kubenotification)
- deploy (kubeloy)
- scheduler (?????)
- ps/status (kubeps)

NEXT

ー 大胆に美しく、スマートにー