

Spring Boot / Kotlinでの大規模クラスター運用

ルート42株式会社 由利 誠

自己紹介

由利 誠(ゆり まこと) / withpop

Docurainの開発全般、広報その他。
開発は初期から関わってきた

[Docurainエンジニアブログ](#)

個人ブログ: [anopara](#)

Twitter: [@anoparanominal](#)

このスライドの入手方法→

「Docurain Labo」でWeb検索



帳票開発って面倒くさいよね？

Docurainとは

The logo for Docurain, featuring the word "Docurain" in a white, sans-serif font centered within a solid blue rectangular background.

1. Excelで帳票テンプレートを作って
2. JSONでデータを流し込んで
3. PDFやExcelで出力できる
4. FaaS

番号 1234 A

令和1年12月3日

石川運輸支局長

自動車検査証

自動車登録番号又は車両番号		登録年月日/交付年月日	初度登録年月	自動車の種別	用途	自家用・事業用の別	車体の形状			
金沢 599 あ 12-34		平成 22年6月2日	平成 20年4月	普通	乗用	自家用	ステーションワゴン [003]			
車名				乗車定員	最大積載量		車両重量	車両総重量		
スバル [133]				5 人	- kg		1300 kg	1575 kg		
車台番号				長さ	幅	高さ	前前軸重	前後軸重	後前軸重	後後軸重
HN11S-1234567				441 cm	174 cm	147 cm	760 kg	- kg	- kg	540 kg
型式		原動機の形式		総排気量又は定格出力	燃料の種類		型式指定番号	類別区分番号		
DBA-GH3		EL15		kg 1.49 L	ガソリン		12321	0001		
所有者氏名又は名称		山田 太郎								
所有者の住所		石川県金沢市鞍月1丁目1番地 [100]								
使用者の氏名又は名称		***								
使用者の住所		***								
使用の本拠の位置		***								
有効期間の満了する日		平成 30年12月31日								
備考		[石川] , 継続検査 自動車重量増税 ¥24,600 平成27年度エネルギー消費効率 (JC08モード燃費値) 算定未了 平成22年度燃費基準20%向上達成車 [走行距離計表示値] 5,200 km (平成25年 5月13日) 平成10年騒音規制車, 近接排気騒音規制値 96 dB マフラー加速騒音規制適用車 以下余白								

高い表現力

こんなのとか

こんなのも

株式会社 FooBar様 取材記事公開スケジュール

2019年4月1日～

件名	担当者	開始日	終了日	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	1				
取材	高崎	2019/04/01	2019/04/08	■																																		
記事本文作成	高崎	2019/04/09	2019/04/15									■																										
記事レビュー	岡田	2019/04/15	2019/04/20															■																				
撮影写真レタッチ	山下	2019/04/09	2019/04/20									■																										
記事公開準備	篠田	2019/04/21	2019/04/22																					■														
記事公開	篠田	2019/04/23	2019/04/23																																			

こんなのも 超簡単に 作れます

川西帝都電鉄 時刻表 Kawatsutsu Time Table											
川鉄金山・山部国際空港方面 For Kawatsutsu Nagoya, Central Japan International Airport											
平日(月~金) Weekdays											
5	05	09	12	25	47						
6	05	09	12	15	20	30	35	40			
7	05	09	12	15	20	30	35	40			
8	01	09	12	15	20	25	30	40	50		
9	01	09	12	15	17	20	25	30	40	50	
10	01	07	12	15	20	25	30	35	40	50	
11	01	07	12	15	20	25	30	35	40	50	
12	01	07	12	15	20	25	30	35	40	50	
13	01	07	12	15	20	25	30	35	40	45	50
14	01	07	12	15	20	25	30	35	40	50	55
15	01	07	12	15	17	20	25	35	30	40	50 55
16	01	07	12	15	20	25	30	35	40	50	55
17	01	07	12	15	20	25	30	35	40	50	55
18	01	07	12	15	20	25	30	35	40	50	55
19	01	07	12	15	20	30	40	50	55		
20	07	09	12	15	20	30	40	55			
21	01	09	25	35	58						
22	01	09	25	35	47	58					

備考

ア: 沼の底駅停車, イ: 北沼駅停車, ウ: 上苗駅停車
 エ: 子苗駅停車, オ: 寺沢橋駅停車, カ: 月宮駅停車

時刻表の見かた

00

沼の底 ← 行先

発車時刻

赤字: ニュー特急自ライナー
 赤字: 快速特急(一部有料車)
 白字: 準特急
 黒字: 普通

●運行情報は川鉄ホームページでもご覧いただけます。
<https://kawatsutsu.co.xx.vv/timetable>



標準停車駅案内



性能

Docurain Labo ブログに詳細が載ってます

- 5534ページのPDFを **9秒** で出力
[6360ページのPDFを爆速で出力して経産省を援助する - Docurain Labo](#)
- 1時間あたり **約4億ページのPDF** を出力可能(青天井)
[日本の全帳票を出力するのは俺達だ！！～1時間あたり4億ページを出力し続けるシステム - Docurain Labo](#)

今日お話しすること

- なぜSpring Boot / Kotlinなのか？
- AWSでのインフラ運用
- 運用でのノウハウと苦労話

**なぜDocurainはSpring Bootと
Kotlinで動いているのか？**

今フルスクラッチで作るなら**Scala**と**Akka HTTP**やるな!
完全ストリーミング処理でOut Of Memoryの発生しないシステム!
薄くて高速なRESTful HTTPサーバ!バックエンドはAkkaで
Actorベースの高い耐障害性と回復力を誇る分散処理...



Docurain

上司

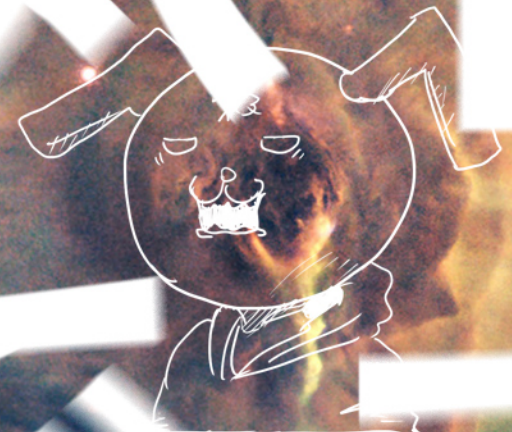


駄目です

Docurain

上司

歌人自序



わたし

現場で求められるのは



わたし

Docurain

上司

「精巧な機械式時計」ではなくて



Docurain

上司



わたし

「AK-47」です

Docurain

上司

精巧な
機械式時計

ではなく

AK-47



おとし

よい子のAK-47講座



実戦の苛酷な使用環境や、戦時下の劣悪な生産施設での生産可能性を考慮し、部品の公差が大きく取られ、卓越した信頼性と耐久性、および高い生産性を実現した。

[AK-47 - Wikipedia](#)

<https://www.quora.com/In-total-how-many-parts-does-an-AK-47-have>



基本設計から半世紀以上を経た今日においても、本銃とその派生型は、砂漠やジャングル、極地などあらゆる世界の地帯における軍隊や武装勢力の兵士にとって**最も信頼される基本装備**になり、『世界で最も多く使われた軍用銃』としてギネス世界記録に登録されている。

[AK-47 - Wikipedia](#)

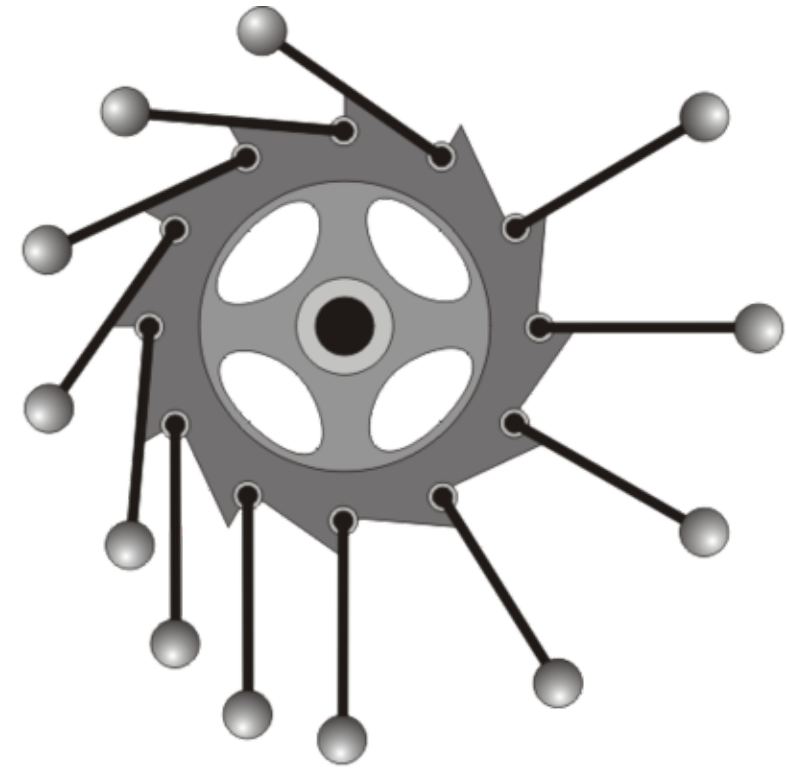
※武器が嫌いな人は何か他の信頼性の高い工業製品を想像してください。マキタとかハズキルーペみたいな。ただの例で他意は無いです。(画像: [Sputnik](#))

つまり何？

何があっても動いて欲しい

- AWSの大規模障害があっても
- 大災害が起こっても
- 大規模なメンテナンスが必要でも
- ITエンジニアのみが罹患する謎の伝染病により開発チームの大半が倒れても

戦うエンジニアのための信頼できる武器(マキタのハズキルーペ)でありたい



避けたいこと



<http://www.masahicom.com/blog/index.cgi/information/20150803pitagorasouchis p2015.htm>

設計方針

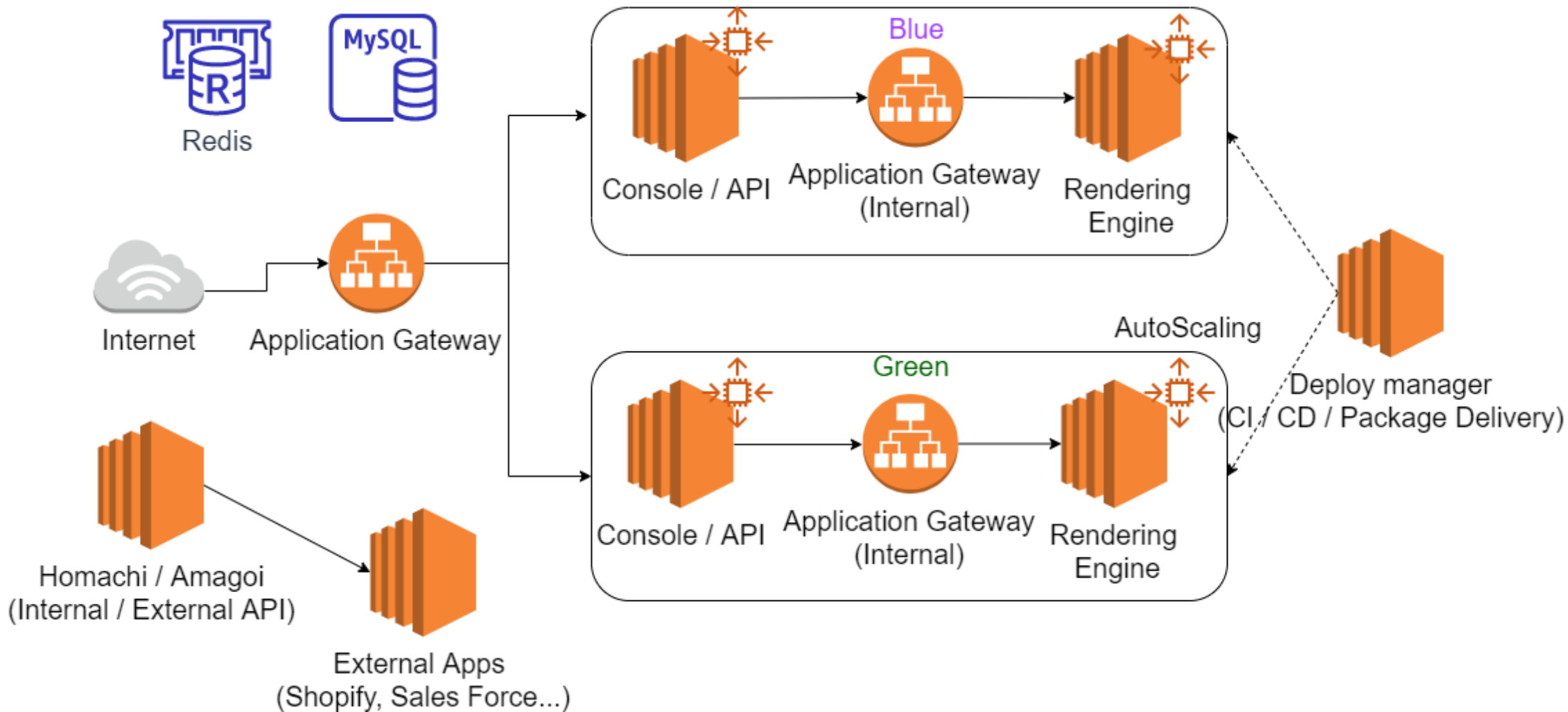
- むやみに複雑にしない (ピタゴラ装置の回避)
- 長きに渡って動く設計
- 間違いなく動く (動くことが検証できる)設計
- 信頼性、可用性、安定性が第一
- どちらかというと枯れた技術を好む
- 最後はレベルを上げて物理で殴る(予算をつけて金で解決、AWSへのお布施)
- サービスメッシュ？k8s？知らない子ですね…

採用した技術

- Kotlin
- Spring Boot
- Spring Security
- JDBC Template
- Hicari CP
- Spring Data Redis
- Vue.js

どれも標準スタック

システム概観



平常時: ~50インスタンス / 高負荷時: 100~ / 4億pph: 500インスタンス

Blue-Greenデプロイメント

- CloudFormation、ECSは使っていない
- ALBにアタッチするターゲットグループを切り替えている
- Blue / Greenどちらのリソースも常に保持している
- AWSコマンドを叩いている
 - (最悪コマンドラインから同じことが可能。保守性の高さ)
- 誰でも理解できる
- Jenkins, CircleCIなどは一部にしか使用していない

サービスイン後の苦勞 1

処理に40秒かかっている帳票が
あるんですが
高速化できないでしょうか



調査してみます。
ちなみに目標値は
どのくらいをご希望ですか？



40秒を最低20秒、
できれば10秒以下に



最低で2倍
できれば4倍
の速度向上





(© 牛次郎/ビッグ錠 『スーパーくいしん坊』)



特に性能で「これを実現して欲しい」と言われたら達成させたいと思うエンジニアがチーム内には多い

サービスイン後の苦勞 1～解決編

	ZSTD Lv.2	ZSTD Lv.7	ZSTD Lv.15
ALB越し	18.31	8.85	9.57
NLB越し	13.53	9.80	7.76
ダイレクト接続	11.54	7.88	8.51
※圧縮・伸張時間込みのトータルの処理時間[秒]			

- 調べたところApplication Load Balancer(ALB)の通過に極端に時間がかかっている
- 内部ネットワークでは1秒以内で転送完了するようなデータ量でもALBを介すると10秒以上時間がかかったりする(ばらつきが大きい)
- L7なのでHTTPメッセージを読んでもらうため？仕方ないか…。
- Network Load Balancerではその傾向が小さいが、やはり遅延は発生し大きいデータ以上の時に顕著

サービスイン後の苦勞 1～解決編

- 結果、L7スイッチのメリットも鑑みてリクエストをZSTD圧縮してALBを使用
- ZSTDの圧縮率、圧縮・伸張時間は **神**
- その他、数多の性能改善も含めて**最終的に40秒→6~8秒まで縮んだ**

サービスイン後の苦勞 1 ～教訓

- ALBでは内部ネットワークが早かろうとでかいデータは圧縮しろ
- 10MBくらいを超えると顕著
- **ZSTDがすごい**
[Zstandard GitHub: facebook/zstd](https://github.com/facebook/zstd)

サービスイン後の苦勞 2

大きなサイズのリクエストを
投げるとエラーになって
しまいます



申し訳ありません。
調査します。
データを頂くことは可能ですか？



はい、エラーになった
データを送ります。



わたし



お客さま


```
withpop@ubuntu:dr$  
withpop@ubuntu:dr$  
withpop@ubuntu:dr$ ls -l  
合計 906464  
-rw-r--r-- 1 withpop withpop 928211208 11月 27 11:13 entity.json  
withpop@ubuntu:dr$  
withpop@ubuntu:dr$ ls -lh  
合計 886M  
-rw-r--r-- 1 withpop withpop 886M 11月 27 11:13 entity.json  
withpop@ubuntu:dr$ █
```

1GB級の JSON







特に性能で「これを実現して欲しい」と言われたら達成させたいと思うエンジニアがチーム内には多い(大事なので2回)

サービスイン後の苦勞 2 ～解決編

- **これが現場で要求されてる性能だ...**
- 各アプリケーション中での徹底した性能改善を実施
- **並列Zramによってスワップ時の性能劣化を抑制**
- ほぼ1GiBのJSONを2分で処理(従来の帳票出力ツールでは20分)

zram…ラズパイみたい…。まともな処理速度が出るのか？

→意外となんとかなる

Zram設定時のポイント

- CPUコア数と同じZramを作成(並列処理される)
- 通常のスワップデバイスよりも優先度を高く
- 圧縮アルゴリズムはlz4 (将来的にはzstdも使えるらしい)
- OOM Killer発動を防ぐためにJavaプロセスの優先度を下げる

まとめ

- Docurainは戦うエンジニアのための武人の蛮用に耐える武器 (AK-47 or マキタ製ハズキルーペみたいな)
- Springスタックは安定してる。情報も多い。誰でも使える優しいフレームワーク
- すごい（と思う）技術を使うことで満足してしまっていないか？
- ALBではネットワーク速度が早くてもリクエストは圧縮しろ
- OOMになりそうだったらZramもお試してください
- **できらあ！！**



ありがとうございました