

# HINASTORY TECH BOOK

# **Hinastory Tech Book**

**hinastory 著**

**2020-01-05 版    Cats Cats Cats 発行**

# はじめに

本書は、hinastory が運用している技術ブログ「cats cats cats」<sup>\*1</sup>の記事を元に書籍したものです<sup>\*2</sup>。振り返ればこの一年近くで 38 本もの記事を書いていました。少なくない量だとは思いますが一体「本」にしたらどのくらいの分量になるのか興味があって作成したものです。

目次を見て頂ければ分かると思いますがなんと 300 ページ強にもなっています。しかもこれは画像をすべて除いているので実際にはさらに 2 倍近くの分量になるのではないかと推測しています。私自身はそんなに書いたつもりはなかったのですが以外とありました。

せっくなのでもう少し校正と加筆をして、きちんとした本に仕上げてみたいと思います。もしかしたらこっそり販売するかもしれません（笑）。

## 免責事項

本書に記載された内容は、情報の提供のみを目的としています。したがって、本書を用いた開発、製作、運用は、必ずご自身の責任と判断によって行ってください。これらの情報による開発、製作、運用の結果について、著者はいかなる責任も負いません。

---

<sup>\*1</sup> <https://hinastory.github.io/cats-cats-cats/>

<sup>\*2</sup> 正確には書籍化途中のものです。まだまだ多くの不備があります・・・

# 目次

はじめに	2
免責事項	2
<b>第 1 章   Hexo でブログに再入門</b>	<b>19</b>
1.1   Hexo とは	19
1.2   静的サイトジェネレータ	19
1.3   静的サイトジェネレータの長所	20
1.4   静的サイトジェネレータの短所	20
1.5   なぜ Hexo を選んだのか？	21
1.6   まとめ	22
1.7   補足	22
<b>第 2 章   JapanContainerDays v18.12 に初参加</b>	<b>23</b>
2.1   会場の様子	23
2.2   キーノート	23
2.3   参加セッション一覧	25
2.3.1   24 日参加セッション	25
2.3.2   25 日参加セッション	26
2.4   食事	27
2.5   戦利品	27
2.6   感想とまとめ	27
<b>第 3 章   気が付いたら本棚がキーボード棚に変わっていた話</b>	<b>29</b>
3.1   自作キーボード以前	29
3.2   自作キーボードに興味を持つ	30
3.3   発狂する	30
3.4   自作キーボードの門を叩く	31

3.5	キーボード棚の作り方 . . . . .	32
3.6	デコキーのすすめ . . . . .	32
3.7	最後に . . . . .	33
<b>第 4 章</b>	<b>サイレントでホットスワップな ErgoDash mini を制作 (前編)</b>	<b>34</b>
4.1	はじめに . . . . .	34
4.2	ErgoDash mini とは . . . . .	34
4.3	制作工程 . . . . .	35
4.3.1	潤滑剤の塗布 . . . . .	35
4.3.2	MogeMicro 対策 . . . . .	36
4.3.3	ダイオードの取り付け . . . . .	36
4.3.4	TRRS ジャックとリセットスイッチの取り付け . . . . .	37
4.3.5	Undergrow LED の取り付け . . . . .	38
4.3.6	後編へと続く . . . . .	39
<b>第 5 章</b>	<b>サイレントでホットスワップな ErgoDash mini を制作 (後編)</b>	<b>40</b>
5.1	制作工程 . . . . .	40
5.1.1	Pro Micro 用のコンスルーの取り付け . . . . .	40
5.1.2	ホットスワップ用ベリリウム取り付け . . . . .	41
5.1.3	スタビライザー、キースイッチ、Pro Micro の取り付け . . . . .	42
5.1.4	Pro Micro に QMK Firmware の書き込み . . . . .	42
5.1.5	動作確認 . . . . .	43
5.1.6	ケースの組み立て . . . . .	43
5.1.7	キーキャップの取り付け ～O-ring による静音化～ . . . . .	43
5.1.8	完成! . . . . .	44
5.2	トラブルシューティング . . . . .	44
5.3	ホットスワップ化の長所と短所 . . . . .	45
5.4	まとめ . . . . .	46
<b>第 6 章</b>	<b>ようこそキーマップ沼へ</b>	<b>47</b>
6.1	そうだ、キーマップをいじろう! . . . . .	47
6.2	QMK Firmware 辛い問題 . . . . .	47
6.3	僕が考えた最強のキーマップ . . . . .	48
<b>第 7 章</b>	<b>GitHub のプライベートリポジトリに移行した話</b>	<b>50</b>
7.1	これまで . . . . .	50

---

7.2	なんでそんな運用にしていたのか？	50
7.3	現在	51
7.3.1	移行方法	51
7.3.2	注意点	51
7.4	まとめ	52
<b>第 8 章</b>	<b>Kubernetes Meetup Tokyo #15 - KubeCon 2018 Recap に初参加</b>	<b>53</b>
8.1	Kubernetes Meetup Tokyo #15 - KubeCon 2018 Recap とは？	53
8.2	会場の様子	54
8.3	発表内容	54
8.3.1	KubeCon + CloudNativeCon North America 2018 Overview - @ladicle	54
8.3.2	Lightning Talk: Introduction to GitOps Deployment to Kubernetes - @sakajunquality	55
8.3.3	Running VM Workloads Side by Side with Container Workloads - @amsy810	55
8.3.4	Keynote: Developing Kubernetes Services at Airbnb Scale - @jyoshise	56
8.3.5	Fly Your Containerized Environments by Joint Work of Harbor and Dragonfly - @capsmalt	56
8.3.6	Securing Kubernetes With Admission Controllers - @tkusumi	57
8.4	食事	57
8.5	感想とまとめ	57
<b>第 9 章</b>	<b>hexo-oembed を公開しました</b>	<b>59</b>
9.1	hexo-oembed について	59
9.2	インストール	60
9.3	使い方	60
9.4	デモ	60
9.4.1	YouTube	60
9.4.2	Twitter	60
9.4.3	Speaker Deck	61
9.5	設定	61
9.5.1	className	61
9.5.2	endpoints	61

9.5.3	embedlyKey . . . . .	61
9.5.4	設定サンプル . . . . .	62
9.6	リポジトリ . . . . .	62
<b>第 10 章</b>	<b>Hexo の oEmbed プラグインを npm に公開した話 (前編)</b>	<b>63</b>
10.1	前提 . . . . .	63
10.2	なぜ Hexo のタグを新しく作ろうと思ったか . . . . .	64
10.3	Hexo のタグの作り方 . . . . .	64
10.4	Speaker Deck のタグを作ろうとして惨敗 . . . . .	65
10.5	oEmbed とは . . . . .	66
10.5.1	hexo-tag-oembed 現る . . . . .	68
10.5.2	真打ち oEmbed Discovery . . . . .	68
10.5.3	図解 oEmbed . . . . .	69
10.5.4	後編へと続く . . . . .	69
<b>第 11 章</b>	<b>Hexo の oEmbed プラグインを npm に公開した話 (後編)</b>	<b>70</b>
11.1	Hexo タグの制作 . . . . .	70
11.1.1	完成形をイメージする . . . . .	70
11.1.2	モデルを探す . . . . .	71
11.1.3	一番簡単なタグを作ってみる . . . . .	72
11.1.4	プラグイン化してみる . . . . .	72
11.1.5	ソースコードを Git 管理してみる . . . . .	73
11.1.6	oEmbed を利用したタグを作ってみる . . . . .	74
11.1.7	答え合わせ . . . . .	75
11.2	Hexo タグの公開 . . . . .	75
11.2.1	GitHub でリポジトリを作成する . . . . .	75
11.2.2	README.md を書く . . . . .	76
11.2.3	package.json を書く . . . . .	77
11.2.4	GitHub に公開する . . . . .	78
11.2.5	npm に公開する . . . . .	78
11.2.6	README にバッジをつける . . . . .	79
11.2.7	Hexo 本家のプラグイン一覧に載せてもらう . . . . .	79
11.2.8	ブログやツイッター等で告知する . . . . .	80
11.3	hexo-oembed の今後 . . . . .	80
11.4	まとめ . . . . .	81

<b>第 12 章</b>	<b>Developers Summit 2019 に参加した感想など</b>	<b>82</b>
12.1	1 日目 参加セッション . . . . .	82
12.1.1	Cloud Native 時代における Docker / Kubernetes による開発 . . . . .	82
12.1.2	いまなぜ Google Cloud Platform を学ぶのか！？ ～GCP を支える Google テクノロジーと愛を語る～ . . . . .	83
12.1.3	【祝】k8s デビュー！ エンタープライズ巨大アプリをマイクロサービス・コンテナ化。段階的移行 (中) の全記録を追う。 . . . .	86
12.1.4	「IT エンジニアに読んでほしい！ 技術書・ビジネス書大賞 2019」プレゼン大会 . . . . .	86
12.2	2 日目 参加セッション . . . . .	87
12.2.1	機械学習システムのアーキテクチャ アラカルト ～ BrainPad における実例を交えて～ . . . . .	87
12.2.2	エンジニアの皆さんに贈る最速キャリア戦略 . . . . .	87
12.2.3	これをまだ知らない Web エンジニアへ贈る - 私が愛する Elixir/Erlang の楽しさと辛さ - . . . . .	88
12.2.4	サーバーレスで最高に楽しめるアプリ開発 . . . . .	89
12.2.5	無意味なアラートからの脱却 ～ Datadog を使ってモダンなモニタリングを始めよう ～ . . . . .	89
12.2.6	Web アプリのチューニングバトル「(社内) ISUCON」の魅力と楽しさ . . . . .	90
12.3	場外バトル . . . . .	90
12.4	まとめ . . . . .	90
<b>第 13 章</b>	<b>Scala3 に入るかもしれない Contextual Abstractions を味見してみた (更新・追記あり)</b>	<b>92</b>
13.1	TL;DR . . . . .	92
13.2	Scala3 とは . . . . .	93
13.3	Dotty とは . . . . .	93
13.4	Contextual Abstractions とは . . . . .	93
13.5	味見の方法 . . . . .	95
13.6	味見の結果 . . . . .	95
13.6.1	基本的な例 . . . . .	95
13.6.2	高度な例 . . . . .	96
13.7	Contextual Abstractions のその他の機能 . . . . .	98
13.8	味見してみた感想 . . . . .	99



13.9	追記・更新内容 . . . . .	99
13.9.1	2019 年 3 月 10 日の更新内容 . . . . .	99
13.9.2	2019 年 6 月 22 日の更新内容 . . . . .	100
13.9.3	2019 年 9 月 15 日の更新内容 . . . . .	100
13.9.4	2019 年 9 月 28 日の更新内容 . . . . .	101
<b>第 14 章</b>	<b>hexo-tag-details を公開しました</b>	<b>102</b>
14.1	hexo-tag-details について . . . . .	102
14.2	インストール . . . . .	102
14.3	使い方 . . . . .	103
14.4	設定 . . . . .	104
14.4.1	className . . . . .	104
14.4.2	open . . . . .	104
14.5	リポジトリ . . . . .	104
14.6	最後に . . . . .	104
<b>第 15 章</b>	<b>全プログラマに捧ぐ！ 図解「ノートブック」</b>	<b>105</b>
15.1	はじめに . . . . .	105
15.2	ノートブックとは . . . . .	105
15.3	ノートブックと実行結果 . . . . .	106
15.4	ノートブックとセル . . . . .	107
15.5	ノートブックとカーネル . . . . .	108
15.6	ノートブックとファイル . . . . .	108
15.7	ノートブックの変換 . . . . .	109
15.8	ノートブックの共有 . . . . .	109
15.9	ノートブックと JupyterLab . . . . .	109
15.10	ノートブックを始めてみる . . . . .	109
15.11	ノートブックの用途 . . . . .	110
15.12	まとめ . . . . .	110
<b>第 16 章</b>	<b>技術書典 6 に一般参加した話</b>	<b>112</b>
16.1	日時と場所 . . . . .	112
16.2	開場前 . . . . .	112
16.3	いざ戦場へ . . . . .	113
16.4	戦利品 . . . . .	113
16.5	感想 . . . . .	114

---

<b>第 17 章</b>	<b>RubyKaigi 2019 in 福岡に参加してきた話</b>	<b>115</b>
17.1	会場について . . . . .	115
17.1.1	会場内の様子 . . . . .	116
17.2	セッション内容 . . . . .	116
17.2.1	The Year of Concurrency (Keynote) @yukihiro_matz . . . . .	116
17.2.2	福岡県知事 . . . . .	118
17.2.3	A Type-level Ruby Interpreter for Testing and Understanding @mametter . . . . .	118
17.2.4	Pattern matching - New feature in Ruby 2.7 @k_tsj . . . . .	118
17.2.5	その他のセッション . . . . .	119
17.3	食事 . . . . .	123
17.3.1	朝食 . . . . .	123
17.3.2	昼食 . . . . .	124
17.3.3	間食 . . . . .	124
17.3.4	Official Party in 川端商店街 . . . . .	124
17.4	令和時代の RubyKaigi . . . . .	124
17.5	まとめ . . . . .	124
<b>第 18 章</b>	<b>全プログラマに捧ぐ！ 図解「パターンマッチ」</b>	<b>125</b>
18.1	はじめに . . . . .	125
18.2	パターンマッチとは . . . . .	126
18.3	パターンマッチとパターン . . . . .	127
18.4	パターンマッチの構造 . . . . .	128
18.5	パターンマッチと照合 . . . . .	128
18.6	パターンマッチと分解 . . . . .	129
18.7	パターンマッチと変換 . . . . .	129
18.8	パターンマッチと合成 . . . . .	129
18.9	パターンマッチとパターンの重なり . . . . .	130
18.10	パターンマッチと広がる世界 . . . . .	131
18.11	まとめ . . . . .	131
<b>第 19 章</b>	<b>「エンジニアの心を整える技術」が予想以上に「技術書」だった件</b>	<b>133</b>
19.1	過去の閉ざされし封印 . . . . .	133
19.2	再び邂逅する . . . . .	134
19.3	3 章までは IT 業界「あるある」 . . . . .	134

19.4	マインドフルネス . . . . .	134
19.5	アドラー心理学 . . . . .	135
19.6	人は本当に変わることができるのか? . . . . .	136
19.7	「エンジニアの心を整える技術」を読み始める前の事前準備 . . . . .	137
19.8	まとめ . . . . .	137
<b>第 20 章</b>	<b>ScalaMatsuri 2019 に参加してきました</b>	<b>139</b>
20.1	ScalaMatsuri とは . . . . .	139
20.2	会場について . . . . .	139
20.2.1	行動規範 . . . . .	140
20.3	カンファレンス DAY . . . . .	140
20.3.1	Scala 3 って、私にはどんな影響があるの? @joang38 (Joan Goyeau) . . . . .	141
20.3.2	再帰:スキーム, 代数,finally tagless, データ型. 統合されたビジョ ン @alexknvl (Alexander Kononov) . . . . .	142
20.3.3	プロジェクトで引き回す型を Eff にするメリット @wing_007 (ma2k8) . . . . .	143
20.3.4	こんなに違う! <-Scala と Kotlin-> @kaelaela (前川裕一) . . .	143
20.3.5	DOT 計算をやさしく説明する @kmizu (水島宏太) . . . . .	143
20.3.6	Making Logic Monad @halcat0x15a (ねこはる) . . . . .	144
20.3.7	sbt コア・コンセプト @eed3si9n (Eugene Yokota) . . . . .	144
20.3.8	継続と DI @gakuzzzz (中村学 (がくぞ)) . . . . .	145
20.4	アンカンファレンス DAY . . . . .	145
20.5	聞きたかったけど聞けなかったセッションたち . . . . .	147
20.6	食事 . . . . .	148
20.7	カフェスペース . . . . .	148
20.8	まとめ . . . . .	148
20.9	戦利品 . . . . .	148
<b>第 21 章</b>	<b>Go Modules とマルチモジュール構成で Go Home する方法</b>	<b>149</b>
21.1	TL;DR . . . . .	149
21.2	Go Modules とは . . . . .	149
21.3	Go Modules で管理を始める . . . . .	150
21.3.1	まずは基本のおはようの挨拶から . . . . .	150
21.3.2	おはようの挨拶をパッケージにしてみる . . . . .	151

---

21.3.3	Go Home しようとして失敗する . . . . .	152
21.3.4	replace のおかげで Go Home に成功する . . . . .	154
21.4	なぜ、マルチモジュール化しなかったのか? . . . . .	154
21.5	まとめ . . . . .	155
21.6	参考文献 . . . . .	155
<b>第 22 章</b>	<b>あの頃の自分に読ませたい「怖くない Qiita」を Qiita に書きました</b>	<b>156</b>
<b>第 23 章</b>	<b>Scala プログラマが圏論を学ぶためのオススメ文献 - 3 選</b>	<b>157</b>
23.1	はじめに . . . . .	157
23.2	プログラマが圏論を学ぶべき理由 . . . . .	157
23.3	オンラインで読めるオススメ文献 - 3 選 . . . . .	158
23.3.1	Category Theory for Programmers Scala Edition . . . . .	158
23.3.2	Scala with Cats . . . . .	165
23.3.3	猫番 . . . . .	170
23.4	プログラマが圏論で学んでおいたほうがよい概念 . . . . .	171
23.5	まとめ . . . . .	172
23.6	もっと圏論を学びたい人向けのオンラインで読めるオススメ資料 . . . . .	172
<b>第 24 章</b>	<b>gRPC UI を使ってみんなが触れる gRPC の遊び場を作りました</b>	<b>174</b>
24.1	TL;DR . . . . .	174
24.2	はじめに . . . . .	175
24.3	gRPC とは . . . . .	175
24.4	gRPC UI について . . . . .	176
24.5	gRPC UI で遊ぼう! ～ gRPC UI playground ～ . . . . .	176
24.6	grpcui-playground の作り方 . . . . .	177
24.6.1	Dockerfile を書く . . . . .	177
24.6.2	GitHub で Dockerfile を公開する . . . . .	178
24.6.3	Docker Hub でイメージをビルド & 公開する . . . . .	178
24.6.4	Arukas で Docker コンテナを公開する . . . . .	179
24.7	まとめ . . . . .	179
24.8	参考文献 . . . . .	180
<b>第 25 章</b>	<b>夢と希望とアイデアに満ち溢れたモノ作りの祭典、Maker Faire Tokyo 2019 に行ってきました</b>	<b>181</b>
25.1	日時・場所 . . . . .	181

25.2	Maker Faire とは . . . . .	182
25.3	作品紹介 . . . . .	182
25.4	Make:本の紹介 . . . . .	183
25.5	まとめ . . . . .	184
25.6	参考文献 . . . . .	184
<b>第 26 章</b>	<b>暗記が苦手な人間が「AWS 認定ソリューションアーキテクト - アソシエイト」に 5 日間で合格した話</b>	<b>185</b>
26.1	はじめに . . . . .	185
26.2	「AWS 認定」とは . . . . .	186
26.2.1	「AWS 認定」の全体像 . . . . .	186
26.2.2	「AWS 認定ソリューションアーキテクト - アソシエイト」の認定試験について . . . . .	187
26.3	スペック . . . . .	188
26.4	きっかけと動機と衝動 . . . . .	189
26.5	試験の申し込み . . . . .	189
26.6	勉強方法 . . . . .	190
26.6.1	一日目 (月曜日) . . . . .	190
26.6.2	二日目 (火曜日) . . . . .	191
26.6.3	三日目 (水曜日) . . . . .	192
26.6.4	四日目 (木曜日) . . . . .	192
26.6.5	五日目 (金曜日) . . . . .	193
26.6.6	試験当日 (土曜日) . . . . .	194
26.7	認定に費やした勉強時間と費用 . . . . .	194
26.8	試験結果 . . . . .	195
26.9	まとめ . . . . .	195
26.10	おまけ：手を動かして学んだほうが良い AWS サービス . . . . .	196
26.11	参考文献 . . . . .	196
<b>第 27 章</b>	<b>Pulumi でプログラマのための「Infrastructure as Code」を実践する</b>	<b>197</b>
27.1	はじめに . . . . .	197
27.2	実践 Infrastructure as Code . . . . .	198
27.2.1	Pulumi の導入 . . . . .	198
27.2.2	pulumi ログイン . . . . .	198
27.2.3	プロジェクトとスタックの作成 . . . . .	198

---

27.2.4	index.ts の作成 . . . . .	199
27.2.5	utils.ts の作成 . . . . .	199
27.2.6	インフラのデプロイ . . . . .	201
27.2.7	デプロイ結果の確認 . . . . .	201
27.2.8	リソースの後片付け . . . . .	201
27.2.9	コードの解説 . . . . .	202
27.3	Pulumi のいいところ . . . . .	203
27.4	Pulumi の仕組み . . . . .	205
27.5	まとめ . . . . .	205
27.6	参考文献 . . . . .	207
<b>第 28 章</b>	<b>長年育て続けた秘蔵の一括ファイル変換ライブラリを晒してみる</b>	<b>208</b>
28.1	はじめに . . . . .	208
28.2	インストール . . . . .	208
28.3	使い方 . . . . .	209
28.3.1	コンバータのフック . . . . .	210
28.3.2	コンバータの変数 . . . . .	212
28.3.3	デフォルトのメタコンバータ . . . . .	212
28.3.4	メタコンバータを作ってみる . . . . .	214
28.4	リポジトリ . . . . .	215
28.5	fileconv の失敗談 . . . . .	215
28.6	最後に . . . . .	216
<b>第 29 章</b>	<b>Scala 3、Python のようにインデントベースの構文で書けるようにな るってよ！</b>	<b>217</b>
29.1	TL;DR . . . . .	217
29.2	まずはコードを御覧ください . . . . .	218
29.3	コードの解説 . . . . .	220
29.3.1	定義 . . . . .	221
29.3.2	if 式 . . . . .	221
29.3.3	for 式 . . . . .	222
29.3.4	match 式 . . . . .	222
29.3.5	ラムダ式 . . . . .	223
29.4	エンドマーカ . . . . .	223
29.4.1	インデントマーカ： . . . . .	223

29.5	設定と書換え . . . . .	224
29.5.1	@main 関数 . . . . .	224
29.5.2	今すぐ試してみたい! . . . . .	225
29.6	インデントベース構文の状況 . . . . .	225
29.7	最後に . . . . .	226
29.8	参考文献 . . . . .	226
29.9	更新内容 . . . . .	227
29.9.1	2019 年 9 月 28 日の更新内容 . . . . .	227
29.9.2	2019 年 11 月 16 日の更新内容 . . . . .	227
<b>第 30 章</b>	<b>ベン図を使って、インタプリタとコンパイラの違いを説明してみる</b>	<b>228</b>
30.1	前提知識 . . . . .	228
30.2	ベン図による説明 . . . . .	229
30.3	コンパイラ . . . . .	229
30.4	インタプリタ . . . . .	229
30.5	視点、視座、視野による見え方の違い . . . . .	230
30.6	まとめ . . . . .	231
30.7	参考文献 . . . . .	232
<b>第 31 章</b>	<b>★記事 30 本突破★ なぜ自分は技術ブログを書き続けるのか？</b>	<b>233</b>
31.1	技術ブログを書く目的 . . . . .	233
31.1.1	「自己紹介」としてのブログ . . . . .	234
31.1.2	「自己研鑽」としてのブログ . . . . .	235
31.1.3	「社会貢献」としてのブログ . . . . .	235
31.1.4	「技術縛り」にした理由 . . . . .	236
31.2	実はブログを書くのは趣味じゃない . . . . .	237
31.3	技術ブログを書く技術 . . . . .	237
31.4	PV 数と人気記事の紹介 . . . . .	239
31.5	まとめ . . . . .	239
<b>第 32 章</b>	<b>Minikube と TensorFlow 2.0 で GPU を使って CIFAR-10</b>	<b>241</b>
32.1	対象読者 . . . . .	241
32.2	マシンの準備 . . . . .	241
32.3	OS のインストール . . . . .	242
32.4	NVIDIA ドライバのインストール . . . . .	242
32.5	Docker Engine for Community のインストール . . . . .	242

32.6	NVIDIA Container Toolkit のインストール . . . . .	244
32.7	kubectl のインストール . . . . .	245
32.8	Minikube のインストール . . . . .	245
32.9	Docker イメージの作成 . . . . .	245
32.10	デプロイメントの作成 . . . . .	247
32.11	サービスの作成 . . . . .	248
32.12	JupyterLab でノートブックを作成 . . . . .	249
32.13	TensorFlow 2.0 with Keras で画像分類 (CIFAR-10) . . . . .	249
32.14	実行結果 . . . . .	252
32.15	まとめ . . . . .	253
32.16	参考文献 . . . . .	254
32.17	おまけ . . . . .	255
<b>第 33 章</b>	<b>Serverless Days Tokyo 2019 に参加してきました</b>	<b>256</b>
33.1	ServerlessDays Tokyo とは . . . . .	256
33.2	会場について . . . . .	256
33.3	午前のセッション . . . . .	257
33.3.1	10x Serverless Product Development for a Startup with Microsoft Azure @ Yutaka Tachibana(EBILAB) . . . . .	257
33.3.2	Keynote @ Keisuke Nishitani (AWS) . . . . .	258
33.3.3	グローバル展開のコネクティッドカーを支える大規模サーバーレスシステム事例 @ Yuya Urayama (TOYOTA), Takanori Suzuki (Acroquest Technology) and Eiichiro Uchiumi (AWS) . . . . .	258
33.3.4	昼食の様子 . . . . .	259
33.4	午後のセッション . . . . .	259
33.4.1	All You Need Is JavaScript @ Jensen Hussey / Cloudflare . . . . .	259
33.4.2	Zero Scale Abstraction in Knative Serving @ Tsubasa Nagasawa (CyberAgent) . . . . .	260
33.4.3	空調設備向け IoT システムにおけるクラウドランニングコスト @ 野原 健太 / ダイキン工業株式会社 . . . . .	260
33.4.4	ISP がサーバレスに手を出した @ 伊藤良哉 & 松田丈司 (NTT コミュニケーションズ) . . . . .	261
33.4.5	ドーナツのデプロイの様子 . . . . .	261
33.4.6	AWS Lake Formation で実現、マイクロサービスのサーバーレスな分散トレーシング @ 江藤武司 & 岩井良和 (Sony Corporation) . . . . .	262



33.4.7	Don't think Serverless Security, think Application Security @ Ido Neeman (Nuweba) . . . . .	262
33.4.8	Azure でサーバーレス、Infrastructure as Code どうしてます か? @ Kazumi Ohira . . . . .	262
33.4.9	The hidden cost and technical debt of running huge Serverless service on production @ James Nguyen / MaaS Global . . . . .	263
33.5	スポンサーセッション & Lightning Talk . . . . .	263
33.5.1	Oracle も Serverless サービスやっています @ Sugiyama Suguru (スポンサーセッション) . . . . .	263
33.5.2	「サーバーレス」な同人誌の紹介 @ めもおきば (nekoruri) (Light- ning Talk) . . . . .	264
33.5.3	目つぶり検証作成期 @ Kana Kitagawa (Lightning Talk) . . . . .	264
33.5.4	SERVERLELESS なエンジニアのための SERVERLESS なオン ラインサロンを VUE/NUXT/COGNITO/STRIPE で作った話 @藤本竜之介 (Lightning Talk) . . . . .	265
33.6	懇親会の様子 . . . . .	265
33.7	まとめ . . . . .	265
33.8	おまけ . . . . .	266
33.9	参考文献 . . . . .	266
<b>第 34 章</b>	<b>AI・機械学習をビジネスに応用したい人向けの資格「G 検定 (JDLA Deep Learning for GENERAL 2019 # 3)」を取得しました</b>	<b>267</b>
34.1	はじめに . . . . .	267
34.2	G 検定とは . . . . .	268
34.3	出題範囲 . . . . .	269
34.3.1	学習のシラバス . . . . .	269
34.3.2	試験の例題 . . . . .	270
34.4	G 検定対策 . . . . .	270
34.5	試験に費やした勉強時間と費用 . . . . .	271
34.6	受験した感想 . . . . .	271
34.7	まとめ . . . . .	272
<b>第 35 章</b>	<b>天下一キーボードわいわい会 Vol.3 に参加してきました。(フォトギャラ リーもあるよ)</b>	<b>273</b>
35.1	日時・場所 . . . . .	273

35.2	天下一キーボードわいわい会とは . . . . .	274
35.3	フォトギャラリー . . . . .	275
35.4	購入品 . . . . .	277
35.5	感想とか . . . . .	277
<b>第 36 章</b>	<b>Google 好きの開発者が集結！ GDG DevFest Tokyo 2019 に参加して きました</b>	<b>278</b>
36.1	日時・場所 . . . . .	278
36.2	GDG DevFest とは . . . . .	279
36.3	セッションとホール . . . . .	279
36.4	参加セッション (午前) . . . . .	279
36.5	ランチ . . . . .	280
36.5.1	GKE を本番運用する時に最低限考えるべきこととその方法 (ラ ンチセッション) @ 富永裕貴 (クラウドエース) . . . . .	280
36.6	参加セッション (午後) . . . . .	281
36.6.1	TensorFlow の使い方 ～ TF2.x とエコシステム ～ @ 太田 満久 (ブレインパッド) . . . . .	281
36.6.2	Go の 10 年の道のりとその変遷 @ 山口 能迪 (Google) . . . . .	282
36.6.3	令和元年 Google 機械学習技術総決算 @ 足立 昌彦 (カブク) . . . . .	282
36.6.4	CloudNative 時代における GKE/Kubernetes ではじめる開発 @ 青山 真也 (サイバーエージェント) . . . . .	283
36.6.5	BigQuery の今年の多すぎるアップデートを振り返ってみる @ なかむら さとる . . . . .	283
36.7	まとめ . . . . .	283
36.8	おまけ . . . . .	284
<b>第 37 章</b>	<b>祝 Ruby2.7 リリース クリスマスなので Ruby の 22 年に渡る コミットの歴史を可視化してみた</b>	<b>285</b>
37.1	はじめに . . . . .	285
37.2	コミットログを収集する . . . . .	286
37.2.1	Ruby のリポジトリを取得する . . . . .	286
37.2.2	Ruby のリポジトリを覗いてみる . . . . .	286
37.2.3	作者別にコミット数をカウントしてみる . . . . .	288
37.2.4	年単位でコミット数をカウントしてみる . . . . .	289
37.3	データの前処理のための基盤を整える . . . . .	289

37.3.1	環境構築方法を選択する . . . . .	290
37.3.2	Dockerfile の準備 . . . . .	290
37.3.3	JupyterLab の起動画面 . . . . .	291
37.4	JupyterLab と Ruby でデータの前処理を行ってみる . . . . .	291
37.4.1	Ruby コミットの名寄せを行ってみる . . . . .	292
37.4.2	最終的な集計テーブルを作成する . . . . .	295
37.5	Flourish で可視化してみる . . . . .	298
37.6	完成した可視化 . . . . .	298
37.7	まとめ . . . . .	298
<b>第 38 章</b>	<b>【続】 Ruby の 22 年に渡るコミットの歴史を可視化してみた (ファイル編)</b>	<b>300</b>
38.1	はじめに . . . . .	300
38.2	ファイル一覧を作成する . . . . .	300
38.3	ファイル単位のコミット回数のテーブルを作成する . . . . .	301
38.4	Flourish で可視化してみる . . . . .	306
38.5	完成した可視化 . . . . .	306
38.6	まとめ . . . . .	307
<b>著者紹介</b>		<b>308</b>

## 第 1 章

# Hexo でブログに再入門

Advent Calendar 用に久々にブログを立てて見ました。以前にパブリックに書いたのは多分 10 年以上前のような気がします。レンタルサーバー借りてドメイン取って Ruby で動く何かを使って立てた気がするけど、よく思い出せません・・・

まあ、過去のことは置いておいてせっかく立てたので、何か役立ちそうなことをメモしていきたいと思います。

### 1.1 Hexo とは

とりあえず今回このブログは、Hexo 使って立ててみました<sup>\*1</sup>。Hexo はブログを書くためのフレームワークですが、WordPress のような従来型のブログサービスとの一番の違いは、静的にブログを生成するか、動的に生成するかです。

### 1.2 静的サイトジェネレータ

Hexo は静的サイトジェネレータというツールに分類されていて、あらかじめ用意したブログのテキストや画像等から、ブログの表示に必要な HTML,CSS,JavaScript などの静的ファイルを生成してくれます。Hexo がやってくれるのはここまでで、あとは Hexo が生成したファイルをお好みの Web サーバに公開すればブログが表示されるという仕組みです。WordPress はこれとは違い、Web にアクセスが来るごとに Web ページをサーバサイドのプログラムが動的に生成してブラウザに返しています。

---

<sup>\*1</sup> Hexo は「ヘキソ」と発音しています。正しいかどうかは分かりません。

### 1.3 静的サイトジェネレータの長所

どちらにも一長一短があって、静的サイトジェネレータの良いところはローカルで確認してから、チェックが完了してから Web 公開できる点です。このローカルで確認できるというのはすごい強みで、画像のアップロードも不要だし、テキストや CSS の修正も手元で直してすぐ確認できるというのが非常に心地いいです。これが動的サイトだと一度サーバにアップロードして確認と言う手間が一つ入るため、色々と面倒臭くなります。

また、静的サイトジェネレータの場合、サイトの生成に必要なファイルは必然的に手元で管理することになるので、Git などのバージョン管理システムと相性が良いです。これは非常に安心感があって、こまめにコミットしておけば様々な理由による『ブログの消失』を未然に防ぐことができます。動的サイトの場合も手元にファイル等の形で残しておけば、バージョン管理することができますが、ちょっとした修正をサーバにアップロードして、バージョン管理用ファイルに反映し忘れるという事故が起き易いです。

セキュリティ的に安心というのも魅力の一つですね。サーバサイドの処理がないので攻撃パターンが限られますし、特に Web サーバにホスティングサービスを使っている場合、自分ではほぼ何もセキュリティについて考えることは不会です。少ない懸念点の一つとしてあげるならば JavaScript を使っている場合です。まあ最近は大抵知らずのうちに使われていると思いますが、特に怪しいライブラリを使っている場合に勝手にマイニングされたり、踏み台にされる場合もあるので、気をつけた方が良いでしょう\*2。

### 1.4 静的サイトジェネレータの短所

このように Hexo のような静的サイトジェネレータ型のブログツールには利点がたくさんあるのですが、もちろん欠点もあります。一番目立つのは、非エンジニアにとって利用するのにハードルが高いということです。例えば、Hexo のトップページには「Simple」と謳いながら、以下の導入時のコマンドが載っています。

```
$ npm install hexo-cli -g
$ hexo init blog
$ cd blog
$ npm install
$ hexo server
```

---

\*2 つい先日 npm パッケージの「flatmap-stream」に仮想通貨を盗み出すコードが混入していて話題になりました。近年 JavaScript のセキュリティインシデントは増加傾向にあるので楽観視はできません。

これを見て「簡単そうだな、やってみようかな」と思える人は世の中に 1% もいないと思います。もちろん現役の Web フロントエンドエンジニアなら、これを見てお湯を注いだカップラーメンができるより早くブログサイトを構築できるでしょう\*<sup>3</sup>。つまり、現在のところ静的サイトジェネレータは Git が分かるフロントエンドエンジニアにとって、非常に使い易いツールだけど、それ以外の人は、まあ覚悟して取り組んでくださいとしか言いようがないですね。。もちろん将来的にはシンプル詐欺などではなく本当に非エンジニアでも無理なく使えるツールが出てくるかもしれないので、この点は静的サイトジェネレータ固有の短所というよりもまだ成熟途中の技術の特徴ということになります。

あと当然ですが、静的サイトジェネレータ単体ではサーバサイドという概念がないので RDB と連携して、大量データの検索とかテーブルの結合とかそういった処理も基本できません。カウンタやトラックバックやコメント機能などのブログにおなじみの機能もデフォルトではありません。このような場合は Web サービスと連携するのが常套手段ですが\*<sup>4</sup>、複雑になりそうなら最初から動的サイトや SPA や SSR 等の構成\*<sup>5</sup>を検討した方が良いでしょう。逆に考えると静的サイトジェネレータの一番の使い道はこのような処理が基本的に不要な場合のプライベートなブログで、まさに Hexo はこのサイトにうってつけなのでしょう。

## 1.5 なぜ Hexo を選んだのか？

まさしく、前掲のコマンドを見て簡単そうに思えたからです (笑)。つまりそれをコピペして実行するだけで試せる環境と知識をすでに有していたからということになります。あとは Hexo というより静的ジェネレータの利点の方が大きいですね。最初は VPS やパブリッククラウド上に、以前のように動的サイトを構築しようと考えていましたが、構築コストやセキュリティのことを考えるとだんだん気が重くなってきました。ブログサービスも検討しましたが、ソースは全て Git 管理したかったのと手元で表示確認がすぐにできないので却下しました。つまり Git 管理と親和性がある手元ですぐにサイトの表示が確認できて、セキュリティの心配をあまり心配しなくてもいい記事投稿が簡単にできるブログツールを探していたということになります。その中で一番最初に目に付いたのが Hexo だったというだけです。記事が Markdown でかけるのもいいですね。なのでもっといいツールが見つければ移行するかもしれませんが、しばらくは Hexo と仲良くやって行こうと思っています。目下の悩みはテーマをどうするかです。デフォルトの landscape も拡張しやすくて悪くはなかったのですが、せっかくこれだけのテーマが公開されているので

\*<sup>3</sup> テーマを弄りはじめるとエンジニアでも頭を抱えそうですが・・・

\*<sup>4</sup> 例えばコメント機能は Disqus と連携すれば実現できます。

\*<sup>5</sup> SPA = Single Page Application, SSR = Server Side Rendering

色々と試したいと思っています。<sup>\*6</sup>。

### 1.6 まとめ

まとめです。Hexo の箇所は静的サイトジェネレータに置き換えても大体合ってます。3 行を目指しましたが、なかなか難しいですね。

- Hexo は静的サイトジェネレータ型のブログツール
- Hexo はローカルで編集して、すぐに実際の表示が確認ができるので、心地よい
- Hexo はバージョン管理と相性が良く、こまめにコミットで安心できる
- Hexo はセキュリティ的に動的サイトに比べて安心
- Hexo は非エンジニアに基本向かない
- Hexo ではカウンタやコメント機能等、サーバサイドの処理が必要なものは基本使えない

### 1.7 補足

前述の通りで Hexo を最初に選んだのは私にとって簡単そうに見えたからです。けどそう見えたのは私の好みや経験という色眼鏡があったからなので、申し訳ないですがこのブログを読んでくれた方に当てはまるかどうかは分かりません。もちろん世の中には Hexo 以外にも山のように静的サイトジェネレータがあるので、余力のある人にはぜひ StaticGen を参考に自分にあった運命の静的サイトジェネレータを探し出して欲しいです。。。<sup>\*7</sup>

---

<sup>\*6</sup> 現在このブログで利用しているテーマはフッターを参照してください。テーマは今後変わる予定があるのであえて本文には記しません。

<sup>\*7</sup> 願わくばこの記事を読んだ人が静的サイトジェネレータの旅に出て、ブログを書いてくれてそれが、私の目に止まりますように・・・

## 第 2 章

# JapanContainerDays v18.12 に 初参加

12/4(火) に JapanContainerDays v18.12 に初参加しました。このイベントは春に 1 回目を開催して今回で 2 回目の開催だそうです\*<sup>1</sup>、会場には 800 人以上の人が詰め掛けたようで非常に盛況でした。

### 2.1 会場の様子

会場は御茶ノ水のソラシティカンファレンスセンターでした。ここには以前にデブサミ夏や大江戸 Ruby 会議 06 に参加した時に来たことがあったので特に迷うことなく到着しました\*<sup>2</sup>。9:40 開始だったのですが念の為 30 分早く会場入りしたのでキーノートの会場はまだ半分近く空いていました。

### 2.2 キーノート

さすがにキーノート開始直前には満席近くになっていました。キーノートは会社やコミュニティの代表が順番に登壇する形式で 2 時間行われました。その中でなんといっても目玉は CNCF の COO である Chris Aniszczyk さんの発表だと思います。

CNCF の歩みや今後の展望が語られていました。気になったのは IoT や Edge にも Kubernetes が進出するという話題でしょうか。KubeEdge には注目していきたいと思います。あとはどんどん Serverless や Nodeless に進んでいく方向性が強調されていまし

---

\*<sup>1</sup> JapanContainerDays は 2 回目にして、今回で最終回だそうです。次回からは CloudNative Days に変わるそうですが Cloud Native Days Tokyo との関係性はよくわかりません・・・

\*<sup>2</sup> 駅から徒歩 1 分なので、そもそも迷うほうが難しい気がします・・・



た。virtual-kubelet の紹介もあり今後が非常に楽しみです。

あと、クラウドネイティブの定義も紹介されていました。今までクラウドネイティブとは何ぞやと思っていたので少しはもやもやが晴れたような気がします。

クラウドネイティブテクノロジーは、パブリック、プライベート、およびハイブリッドクラウドなどの最新のダイナミックな環境でスケーラブルなアプリケーションを構築および実行するための組織の能力を強化します。コンテナ、サービスメッシュ、マイクロサービス、不変インフラストラクチャ、および宣言型 API は、このアプローチの例です。

これらの技術は、復元力があり、管理しやすく、観測可能な疎結合システムを可能にします。堅牢な自動化と組み合わせることで、エンジニアは頻繁に、そして予想通りに影響の少ない変更を最小限の労力で行うことができます。

クラウドネイティブコンピューティング基盤は、ベンダーに依存しないオープンソースのプロジェクトのエコシステムを促進し維持することによって、このパラダイムの採用を推進しようとしています。最先端のパターンを民主化し、これらのイノベーションを誰もが利用できるようにします。

あとは CNCF の歩き方や全体像も紹介されていました。

Trail Map はともかく Landscape は圧巻ですね・・・

まあその中でも CNCF が直接ホスティングしているのはごく僅かです。これを見ると全部で 30 程ありますが、現時点では成熟度が最高の Graduated になっているのは Kubernetes、Prometheus、Envoy のわずか 3 つです\*3。これからクラウドネイティブが浸透していく中でこれらのプロジェクトがどの様になっていくのか注視していきたいと思っています。

キーノートには他にも面白い発表がいくつもあったのですが、印象に残ったのはメルカリの発表でした。なぜコンテナオーケストレーションに Kubernetes なのかという問いに対して拡張性とエコシステムを挙げていましたが、これと同じ答えはこの後に続くセッションでも繰り返し聞くことになりました。そしてこの答えが OSS で勝ち抜くための到達点なんだという気がしました。シンプルで直交的で整合性のとれた拡張性で素早くエコシステムを構築すること。言葉にすると短いですがこの状況をわずか数年で作り上げて決着までほぼつけてしまうとは、改めて Kubernetes の成し遂げたことの凄さを実感しました。

---

\*3 というか Prometheus は CNCF だったとは初めて知りました・・・

## 2.3 参加セッション一覧

とりあえず参加セッションを列挙してみます。メモったキーワードを記載していますが、あやふやな部分も多く含むためさりと流していただけると幸いです。

### 2.3.1 24 日参加セッション

- keynote
  - CNCF
  - Microservices on Kubernetes at Mercari (Mercari)
  - Kubernetes による機械学習基盤への挑戦 (Preferred Networks)
  - LINE エンジニアを支える CaaS 基盤の今とこれから (LINE)
  - Cloud Native の未来と IBM の取組み (IBM)
  - クラウドネイティブで作る、新しいクルマの世界 (デンソー)
  - ZOZOTOWN システムリプレイスにおける Kubernetes 活用 (ZOZO テクノロジーズ)
- コンテナネットワーキング (CNI) 最前線
  - コンテナネットワーキングは闇が深い、動きが早い、スケーラビリティに注意
- Kubernetes ネットワーキングのすべて
  - Services, Ingress, NetworkPolicy
- 40 topic of Kubernetes in 40 minutes
  - service でなく ingress を使え、latest タグは使うな、DockerHub のイメージは信用するな、バージョンアップは容易にするな、ボリュームは使うな、全機能を使う必要はない、ingress は GA ではない
- Ansible、Terraform、Packer で作る Self-Hosted Kubernetes
- Kubernetes から始めるクラウドネイティブエンジニアへの道 ～ Kubernetes トレーニングと、CKA/CKAD 資格取得に向けて ～
  - 構築ツールを使わずにやってみた (kubernetes the hard way)、k8s は Cloud Native においてはスターティングポイント
- 2019 年はコンテナよりもクラウドネイティブ!? Knative のすべて
  - コンテナ辛い！そこで knative, カナリアリリースが楽、riff もあるよ
- After Party & Booth Crawl !!

### 2.3.2 25 日参加セッション

- 2020 年のコンテナはどうなる!? コンテナプラットフォームのこれまでとこれから
  - マイクロサービスは組織論に行き着くので難しい
  - k8s は拡張性の高い OS ようなもの
  - K8s は開発や POC まではうまくいくが本番運用に課題あり
  - istio がここまでくると思っていなかった。サービスメッシュはエンタープライズにこそ必要
  - cloudfoudry,istio,mesoss,open shift,rancher,rio, core os,kubeflow, knative
- マイクロサービスの高可用性：Ingress, サービスメッシュの世界におけるロードバランシング
  - nginx 社は実在する
  - plus もあるよ
  - 複雑になった Web を簡素化したい
  - fabric model ではすべての app にサイドカーとして nginx が入る
- Docker セキュリティ：今すぐ役に立つテクニックから、次世代技術まで
  - 鍵をイメージに入れるな、--ssh,--seacret を使え。使えない場合はスカッシュやマルチステージというワークアラウンドを使え
  - イメージスキャナ - clair, aqua microscanner
  - 誤検知もあるので信用しすぎるな
  - docker ソケットに気をつけろ。docker グループに入れるのは危険
  - コンテナ is namespace + capabilities + cgroups
  - SELinux 無効化するときは「石川さん、Walsh さんごめんなさい」と唱えてから実行
- Kubernetes と暮らす Rancher な生活
  - rancher は k8s のオーケストレーション層ではなくその上のストラテジー層
  - 2.0 で k8s 一本になり、Pure Go になり、マネージドも OK になりカタログが Helm になった
  - rio はおもしろそうだがまだかなり柔らかい。k8s を意識しない時代に・・・
- MeetUp を活性化せよ！ 最強のリアルタイム QA 投稿アプリを Cloud\_Native につくってみた
  - Qicoo
  - 自演が可能です
  - eks,ec2,elastic cache, rds,cloud front, route 53,cert manager, elb, travis ci,

kustomize, hepioak, spinnaker,kanyeta,prometheousOperator, Halyard

- Cloud Native Developers JP
- Cloudcraft – Draw AWS diagrams
- コンテナ時代の OpenStack
- Kubernetes Meetup Tokyo 2 年間の振り返りと未来
  - 参加者の規模数がすごい
  - 登壇者限定で濃い話をする会がある
  - 登壇してください

自分が参加したセッション以外は以下にまとめがあるのであとで読みたいと思います。

## 2.4 食事

昼食はランチセッション中に配布されたサンドイッチやまい泉ヒレかつサンドを食しました。

お菓子やドリンクも無料で楽しむ事ができました\*4。ContainerDays のロゴが入ったマカロンが可愛かったです。

After Party ではビールやワインや焼酎をの呑みながらワイガヤしてました。急遽決まった LT 大会では個性あふれる様々な会社が Hiring トークをしていました。

## 2.5 戦利品

勢い余って電子書籍 3 冊も買ってしまいました。面白かったのは NGINX で配られた USB ケーブルです。正直そのタコ足のような発想はありませんでした (笑)

## 2.6 感想とまとめ

参加したことで CloudNative がどこまできているか肌で実感できた気がしました。自宅では minikube から GPU を触れる環境を構築していますが、いまいち有効利用できていない気がしていたので、このイベントを機会に手を動かしながらいろいろと CloudNative に近づけて見たいと思います。

---

\*4 当日は本当に無料だと喜んだのですが、冷静に考えると参加費を払っているので無料でもなんでもなかったです・・・しかし、チケットは最後まで Early Bird 価格のまま安売りされていたけど主催者の想定した人数が集まらなかったから値下げしたのか、それとも早めに予定金額を調達できたから値下げしたのか真相がわからない

- Kubernetes の一人勝ち
- 今後は Kubernetes のエコシステムの何処を占めるかの戦い
  - サービスメッシュ (Istio) がかなり熱い
  - Helm もデファクトになりそう
  - knative, virtual-kube は注視したい
  - 機械学習のインフラとしても面白そう
- Kubernetes は本番運用に課題あり
  - knative, OpenShift, Rancher 等さまざまなアプローチで解決しようとしている
  - 最終的には NoOps やサーバーレスや FaaS な未来に行き着くのだろう・・・  
(時期未定)
- 来年は CloudNative Days