

AWS Summit Tokyo 2018 Day 3

gumi Inc. が目指す…

Less DevOps, More Code

～運用を削減して, もっとコードを書こう!!～

2018年6月1日

gumi Inc. CTO 幾田 雅仁



Copyright (C) gumi Inc. 2018 All Rights Reserved

会社紹介



Copyright (C) gumi Inc. 2018 All Rights Reserved

会社紹介













海外展開




設立	2007年6月13日
国内子会社	3社
海外子会社	5社

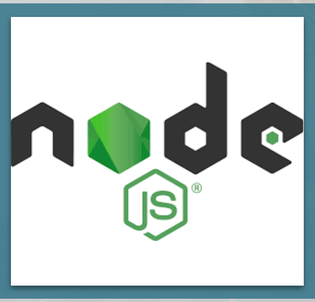
※2018年6月現在

Copyright (C) gumi Inc. 2018 All Rights Reserved

gumi

サーバサイド技術

API

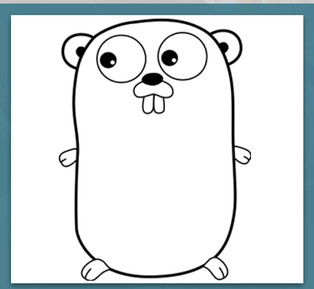





共通サービス

(認証, 課金, ログ収集, ツール群)





インフラ



Copyright (C) gumi Inc. 2018 All Rights Reserved

会社紹介

gumi

サーバサイド技術

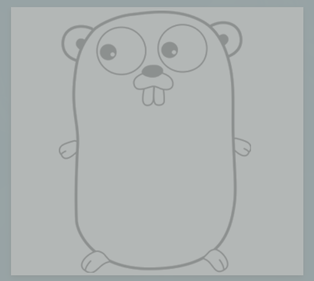
API






共通基盤
(認証, 課金, ログ収集, ツール群)





インフラ



Copyright (C) gumi Inc. 2018 All Rights Reserved

gumi TECH Drinkup

技術情報を共有する飲み会

第一回は，参加者を募集中

https://gumitech.connpass.com/event/84157/

Copyright (C) gumi Inc. 2018 All Rights Reserved

自己紹介



Copyright (C) gumi Inc. 2018 All Rights Reserved

自己紹介



幾田(いくた) 雅仁(まさひと)

gumi Inc. CTO, 兼ボードゲームクラブ部長

Japan Elixir Association 理事

ID: cooldaemon

1997年4月	パソコン通信 (サーバ, クライアント, プロトコル設計)
2002年6月	ポータルサイト (Webエンジニア!!)
2007年6月	クレジットカード (決済の端末やサーバ開発)
2012年6月	ソーシャルゲーム (イマココ)

Copyright (C) gumi Inc. 2018 All Rights Reserved

今日の話し



Copyright (C) gumi Inc. 2018 All Rights Reserved

今日の話し

1200以上の
監視対象

20以上の
AWSアカウント

70以上の
環境

VS.

5人の
インフラエンジニア

Copyright (C) gumi Inc. 2018 All Rights Reserved

今日の話し

1200 以上の
監視対象

20 以上の
AWS アカウント

70 以上の
環境

VS.

5 人の
インフラエンジニア

講義者ノート

これが, いまの gumi の状況.

Copyright (C) gumi Inc. 2018 All Rights Reserved

今日の話し

1200以上の
監視対象

20以上の
AWSアカウント

70以上の
環境

急な差し込み対応

日々の運用

高品質

VS. 5人の
インフラエンジニア

リビルド歓迎

新環境5営業日内!!

Copyright (C) gumi Inc. 2018 All Rights Reserved

今日の話し

1200 以上の
監視対象

20 以上の
AWS アカウント

70 以上の
環境

急な差し込み対応

日々の運用

高品質

VS.

5 人の
インフラエンジニア

リビルド歓迎

新環境5営業日内!!

講義者ノート

日々の日常業務を行いながら, 急な再仕込みに対応しつつ, 新環境をミーティング含めて, 5営業日以内に準備する. しかも, よく検証されており, 障害に強く, 作業に抜け漏れがない. 要件が漏れていも, 再構築を大歓迎している. 仮に, VPC の作り直しが連鎖して, 全て作り直しになっても大歓迎.

Copyright (C) gumi Inc. 2018 All Rights Reserved

今日の話し

急な差し込み対応

日々の運用

高品質

5人の
インフラエンジニア

リビルド歓迎

新環境5営業日内!!

Copyright (C) gumi Inc. 2018 All Rights Reserved

今日の話し

急な差し込み対応

日々の運用

高品質

5人の
インフラエンジニア

リビルド歓迎

新環境5営業日内!!



Copyright (C) gumi Inc. 2018 All Rights Reserved

今日の話し

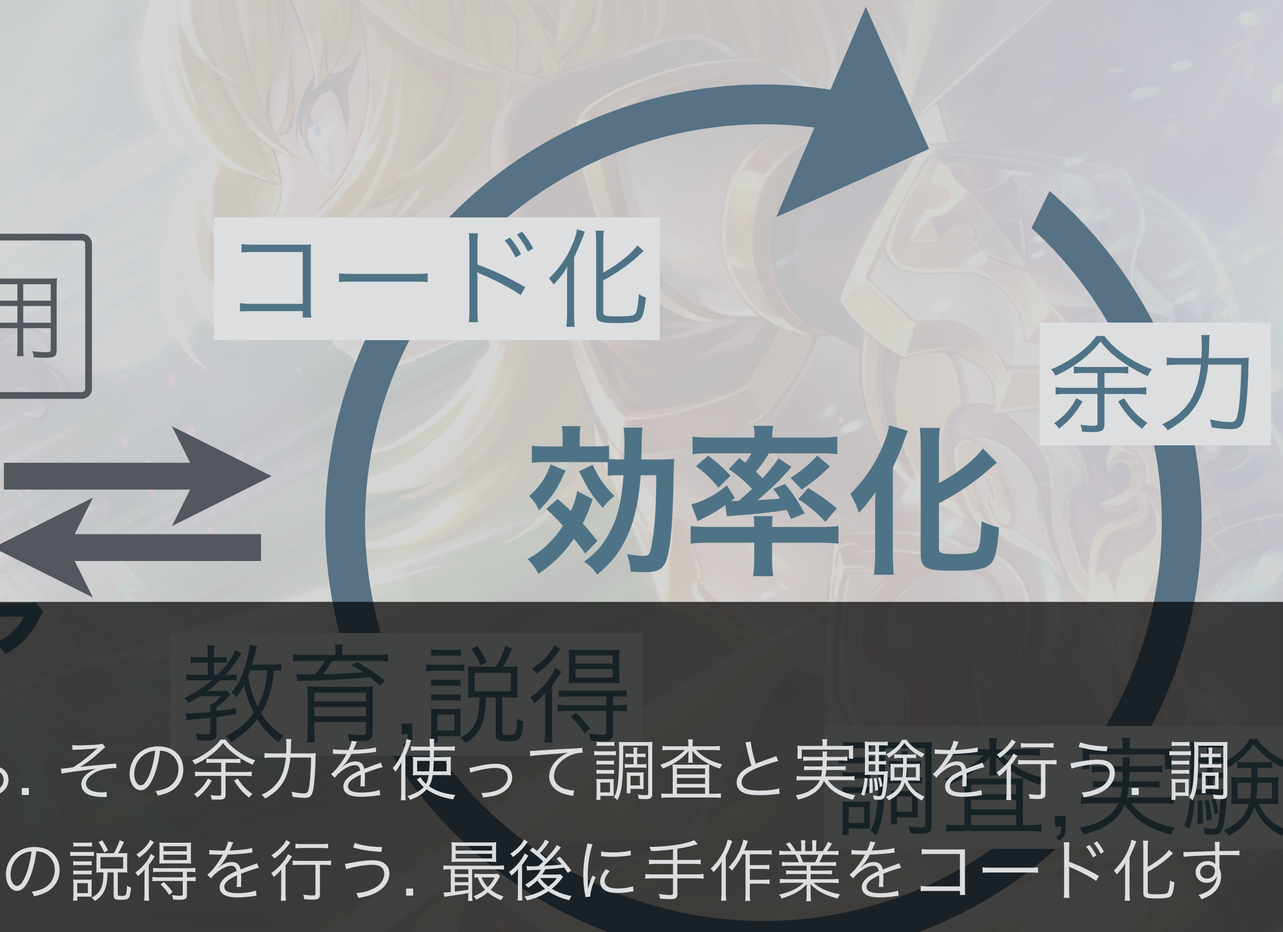
急な差し込み対応

高品質

日々の運用

5人の

インフラエンジニア



新環境5営業日内!!

講義者ノート

どう実現しているのか? まず余力を作る. その余力を使って調査と実験を行う. 調査結果を元に情報共有や教育, 決裁権者の説得を行う. 最後に手作業をコード化する. そうすると, 再び余力が生まれる. これで効率化のサイクルが回る. だから実現できる.

Copyright (C) gumi Inc. 2018 All Rights Reserved

今日の話し

そろそろ，みなさんも…

インフラで疲弊するの止めませんか？

Copyright (C) gumi Inc. 2018 All Rights Reserved

今日の話し？？？

今の話し…

胡散臭い!!!

と思った人は**挙手!**をお願いします



Copyright (C) gumi Inc. 2018 All Rights Reserved

今日の話し？？？

gumi

胡散臭い!!!

臭いところに蓋，見落とし，無関心

本当の現場は問題だらけ

Copyright (C) gumi Inc. 2018 All Rights Reserved

今日の話し？？？

本当の現場は問題だらけ

操作ミスでDB 消しちゃった

私たちも失敗してきた…

設定ミスでMQ 止まっちゃった

Copyright (C) gumi Inc. 2018 All Rights Reserved

今日の話し？？？

本当の**現場**は**問題だらけ**

私たちもコード化は試行錯誤･･･

Copyright (C) gumi Inc. 2018 All Rights Reserved

今日の話し？？？

本当の現場は問題だらけ

私たちもコード化は試行錯誤･･･

講義者ノート

まだ, 全ての環境をコード化できていない. 未だに旧環境をメンテナンスしながら, 少しずつ改善を続けている. 外部のインフラ業者へ手作業を依頼しているゲームタイトルもある.

Copyright (C) gumi Inc. 2018 All Rights Reserved

今日の話し？？？

本当の**現場**は**問題だらけ**

私たちもコード化は試行錯誤…

制約や条件に竦む

▽

残る手作業

▽

運用コスト増大

Copyright (C) gumi Inc. 2018 All Rights Reserved

今日の話し？？？

本当の現場は問題だらけ

私たちもコード化は試行錯誤･･･

制約や条件に竦む

▽

残る手作業

▽

運用コスト増大

講義者ノート

コード化自体も難しい. コードが書けない運用者にコード化をお願いすると, 何をコード化して良いのか判断できない. 加えて, 制約条件が多く見えているため, 面倒になってコード化を諦めてしまう. 結果, 業務拡大と共に仕事は増え続けるため, 運用コストは増加し続ける.

Copyright (C) gumi Inc. 2018 All Rights Reserved

今日の話し？？？

本当の**現場**は**問題だらけ**

私たちもコード化は試行錯誤…

制約や条件に竦む

▽

残る手作業

果敢にコード化

▽

複雑怪奇な仕組み

▽

運用コスト増大

Copyright (C) gumi Inc. 2018 All Rights Reserved

今日の話し？？？

gumi

本当の現場は問題だらけ

私たちもコード化は試行錯誤…

制約や条件に竦む

果敢にコード化

残る手作業

複雑怪奇な仕組み

運用コスト増大

講義者ノート

逆に，運用を知らない開発者にコード化をお願いすると，ほど良い要件定義をできず，無駄なコード化に走る．多大な工数と引き換えに，メンテナンス不能な複雑な仕組みができあがる．

Copyright (C) gumi Inc. 2018 All Rights Reserved

今日の話し？？？

本当の**現場**は**問題だらけ**

私たちもコード化は試行錯誤…

制約や条件に竦む

果敢にコード化

残る手作業

複雑怪奇な仕組み

運用コスト増大

講義者ノート

蛇足だが, 数値は結果であり, 理念や行動ではない. CTO が 1200 監視ポイントを 5 人で運用しろと言いた出したら, 非常に危険な兆候.

Copyright (C) gumi Inc. 2018 All Rights Reserved

今日の話し？？？

本当の**現場**は**問題だらけ**

私たちもコーディングは試行錯誤…

では

何を伝えに

きたの?

制約条件を読む

▽

残る手作業

▽

運用コスト増大

▽

複雑怪奇な仕組み

▽

果敢にコード化

Copyright (C) gumi Inc. 2018 All Rights Reserved

今日の話し

事例紹介

スタートアップ ▷ 一部上場企業

Copyright (C) gumi Inc. 2018 All Rights Reserved

今日の話し

事例紹介

スタートアップ ▷ 一部上場企業

講義者ノート

スタートアップから上場企業へ成長するまでに味わった インフラ運用の成長痛の話しをする. この成長痛は今も変わらずに感じている. この話しから, 「そんな状況から改善を続けてきたのか？」「ならウチでもできる！」と思っていただくのが今日の作戦.

Copyright (C) gumi Inc. 2018 All Rights Reserved

今日の話し

gumi

事例紹介

スタートアップ	▷	一部上場企業
一人	▷	チーム,組織
手作業	▷	コード化

Copyright (C) gumi Inc. 2018 All Rights Reserved

今日の話し

事例紹介

講義者ノート

今日の話しは, 一人のインフラ担当者が手作業を行っていた話しから始める. 話しの最後にはチーム/組織となり, 手作業がコード化される. よって本講演の対象者の間口は広い. 既に高度な自動化が完了している方は対象外. 全ての現場が同じ道を辿るわけではないが, みなさんの職場と gumi の似ているところを探してほしい.

Copyright (C) gumi Inc. 2018 All Rights Reserved

今日の話し

持ち帰ってほしいこと

○×メソッドで大成功　□△ツールで大改善

△

事例紹介

スタートアップ ▷ 一部上場企業

一人 ▷ チーム,組織

手作業 ▷ コード化

Copyright (C) gumi Inc. 2018 All Rights Reserved

今日の話し

持ち帰ってほしいこと

~~○×メソッドで大成功 □△ツールで大改善~~

△

事例紹介

スタートアップ ▷ 一部上場企業

一人 ▷ チーム,組織

手作業 ▷ コード化

Copyright (C) gumi Inc. 2018 All Rights Reserved

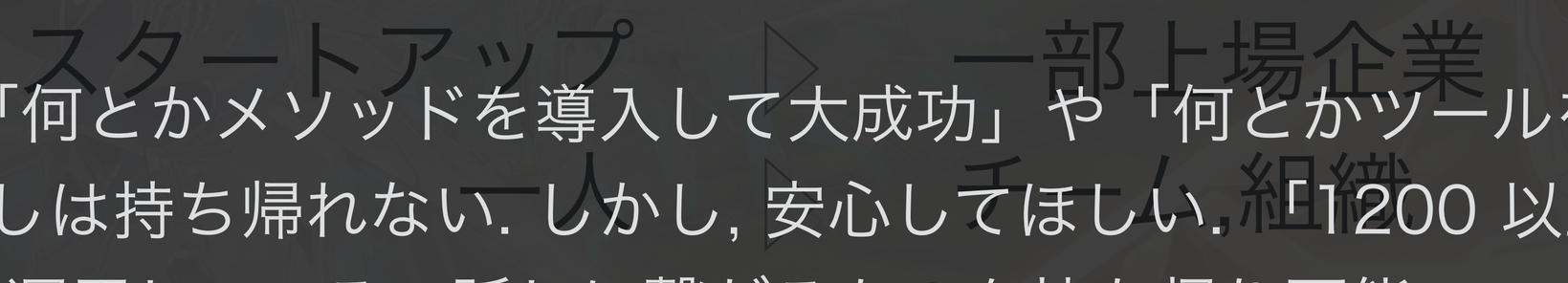
今日の話し

持ち帰ってほしいこと

~~○×メソッドで大成功 □△ツールで大改善~~

△

事例紹介



講義者ノート

残念ながら，「何とかメソッドを導入して大成功」や「何とかツールを入れて大改善」という話しは持ち帰れない．しかし，安心してほしい．「1200 以上の監視ポイントを 5 人で運用している」話しに繋がるものを持ち帰り可能．

Copyright (C) gumi Inc. 2018 All Rights Reserved

今日の話し

持ち帰ってほしいこと

実験的であること　失敗から学ぶこと

First to Try, First to Fail, First to Recover!

△

事例紹介

スタートアップ ▷ 一部上場企業

一人 ▷ チーム,組織

手作業 ▷ コード化

Copyright (C) gumi Inc. 2018 All Rights Reserved

今日の話し

持ち帰ってほしいこと

実験的であること　失敗から学ぶこと

First to Try, First to Fail, First to Recover!

△

事例紹介

スタートアップ　▷　一部上場企業

人　▷　チーム／組織

手作業　▷　コード化

講義者ノート

インフラ運用の効率化は, 失敗と改善の繰り返しが延々と続く. 実験を面倒がり, 失敗を恐れていると, いつまでも手作業地獄から抜け出せない.

Copyright (C) gumi Inc. 2018 All Rights Reserved

目次

悪戦苦闘した
改善の歩み (17分想定)
は8年かけても終わらない

▽

改善を促進する
コード化 (17分想定)
の進め方は試行錯誤の連続

▽

まとめ (30秒想定)

Copyright (C) gumi Inc. 2018 All Rights Reserved

目次

悪戦苦闘した
改善の歩み (17分想定)
は8年かけても終わらない

▽

改善を促進する
コード化 (17分想定)
の進め方は試行錯誤の連続

▽

まとめ (30秒想定)

講義者ノート

今日の目次. gumi の改善の歩みを聞いて, アレならウチにもできる！と思ってほしい. 組織的なエモい話しですまぬ. その後, 開発やツールの話しをする. みなさんが使っているツールと比較してほしい. 最後に, 簡単にまとめを話す.

Copyright (C) gumi Inc. 2018 All Rights Reserved

目次

悪戦苦闘した
改善の歩み
は8年かけても終わらない

▽

改善を促進する
コード化
の進め方は試行錯誤の連続

▽

まとめ

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み



2010-2011年　2012-2013年　2014-2016年　2017-2018年

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み





Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2010-2011年)

ヒーロー登場 試行

初代CTOによる鶴の一声



コスト管理が容易

コード化が可能

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2010-2011年)

ヒーロー登場 試行

初代CTOによる鶴の一声



コスト管理が容易

コード化が可能

AWS 専任者 任命



専門家に権限と責任を集中

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2010-2011年)

ヒーロー登場 試行

初代CTOによる鶴の一声



AWS 専任者 任命



講義者ノート

一人にインフラの権限と責任を集中させ, 一定した品質を保つ. AWS の操作は AWS に精通した専門家(ヒーロー)が責任を持って行うため, 事故などおきない！

コスト管理が容易

専門家に権限と責任を集中

コード化が可能

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2010-2011年)

ヒーロー登場 課題

AWS 専任者 任命



専門家に権限と責任を集中

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2010-2011年)

ヒーロー登場 課題

AWS 専任者 任命



専門家に権限と責任を集中

ドキュメント不足

知識の偏り

業務量増加

▽

タスク消化不良

トラックナンバー1

聖域化

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2010-2011年)

ヒーロー登場 課題

AWS 専任者 任命



ドキュメント不足

知識の偏り

業務量増加

▽

タスク消化不良

講義者ノート

一人で処理できる業務に限界がある. 業務量が増えると, すぐに業務消化不良となる. また, 業務が一人に閉じているため, 引き継ぎを意識したドキュメントが存在せず, 知識が一人に偏っている. この一人に何かがあると, 業務が止まる状態.

Copyright (C) gumi Inc. 2018 All Rights Reserved

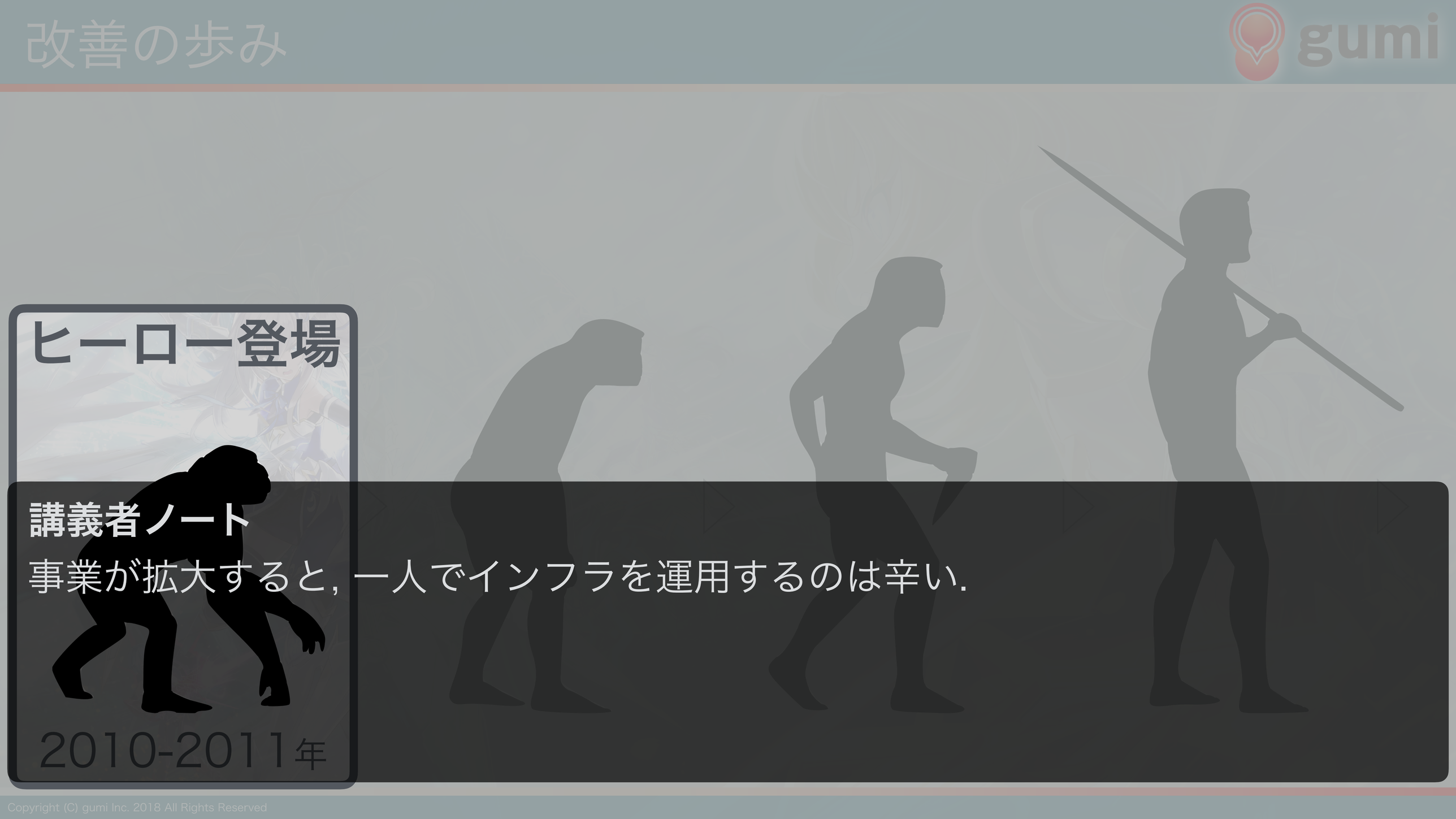
改善の歩み





Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み



講義者ノート

事業が拡大すると, 一人でインフラを運用するのは辛い.

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み

ヒーロー登場



2010-2011年

チーム結成



2012-2013年





Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2012-2013年)

チーム結成 試行

エンジニア首脳陣
による**合議**



Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2012-2013年)

チーム結成 試行

エンジニア首脳陣
による合議



運用者



Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2012-2013年)

チーム結成 試行

エンジニア首脳陣
による合議



運用チーム



Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2012-2013年)

チーム結成 試行

エンジニア首脳陣
による合議



運用チーム



講義者ノート

当時の運用担当者の下に運用チームを結成した. 社内外から AWS に詳しそうな運用者を採用する方針.

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2012-2013年)

チーム結成 試行

エンジニア首脳陣
による合議



タスク管理
ドキュメント整備

運用チーム



Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2012-2013年)

チーム結成 試行

エンジニア首脳陣
による合議



運用チーム



タスク管理
ドキュメント整備

講義者ノート

一人ではなく, チームで仕事をするため, タスク管理ツールと, ドキュメント管理ツールを導入.

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2012-2013年)

チーム結成 試行

共通開発チーム



エンジニア首脳陣
による合議



タスク管理
ドキュメント整備

運用チーム



Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2012-2013年)

チーム結成 試行

共通開発チーム

エンジニア首脳陣
による合議

運用チーム



△タスク管理
ドキュメント整備

講義者ノート

開発チームと運用チームの隙間を埋めたり，サーバ負荷問題を解決する共通サービス開発のチームを結成. 2012年当時の議事録に，運用チームと共通サービス開発の両チームは，将来的に業務の垣根が消え，緩やかに混ざっていくと記載さている.

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2012-2013年)

チーム結成 試行

共通開発チーム



エンジニア首脳陣
による**合議**



運用チーム



タスク管理
ドキュメント整備

講義者ノート

このチームの結成により, 各ゲームが共通で利用するサービスやライブラリ, フレームワークの開発や, 便利な外部サービスやツールの導入が積極手に行われるようになった. 障害や負荷に強いサーバ設計なども, このチームの功績.

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2012-2013年)

チーム結成 課題

運用チーム



業務量増加 + チーム問題

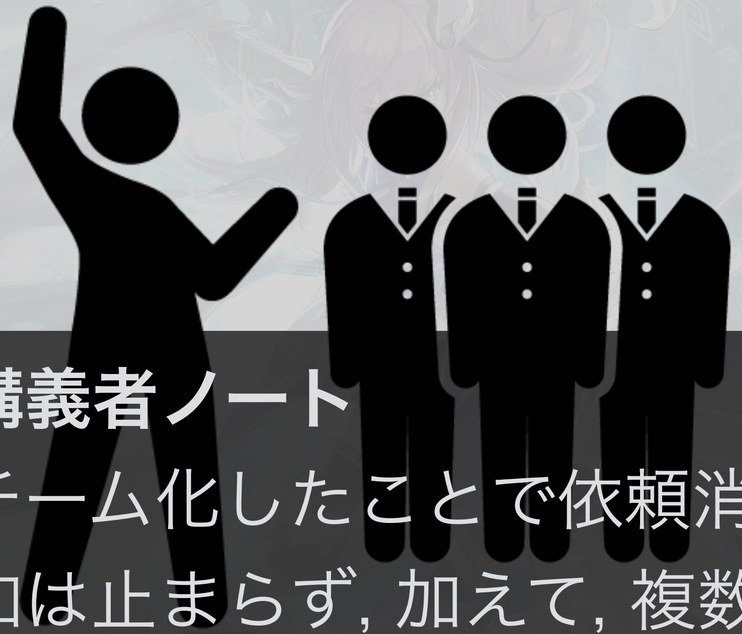
Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2012-2013年)

チーム結成 課題

業務量増加 + チーム問題

運用チーム



講義者ノート

チーム化したことで依頼消化の並列度が上った. しかし, 事業拡大に伴う業務量増加は止まらず, 加えて, 複数の人間が共に働くと必ず問題が発生する.

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2012-2013年)

チーム結成 課題

運用チーム



業務量増加 + チーム問題

▽

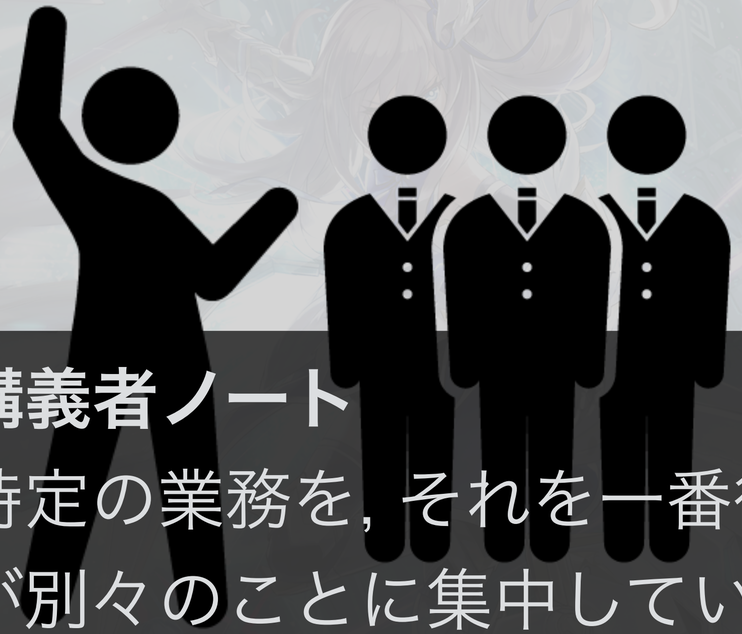
属人化(カウボーイ)

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2012-2013年)

チーム結成 課題

運用チーム



業務量増加 + チーム問題

▽

属人化(カウボーイ)

講義者ノート

特定の業務を, それを一番得意とする人に渡し続けるため, チーム内で一人ひとりが別々のことに集中していた. 結果, 一人が倒れると, 業務に支障がでる.

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2012-2013年)

チーム結成 課題

運用チーム



業務量増加 + チーム問題

▽

属人化(カウボーイ)

サイロ化

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2012-2013年)

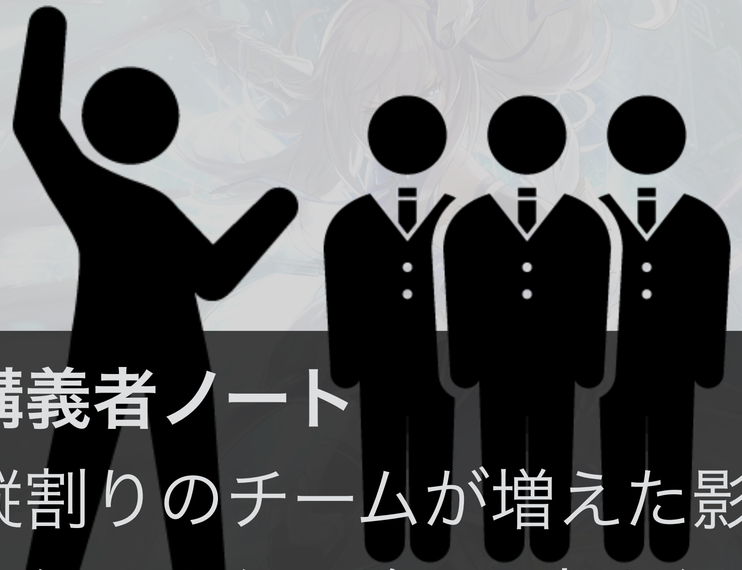
チーム結成 課題

業務量増加 + チーム問題

▽

属人化(カウボーイ)

運用チーム



▽

サイロ化

講義者ノート

縦割りのチームが増えた影響で, コミュニケーションロスが増えた. 何か一つの取り組みを行う際, 同意を得るコストが高い状態.

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2012-2013年)

チーム結成 課題

運用チーム



業務量増加 + チーム問題

▽

属人化(カウボーイ)

構成ドリフト多発

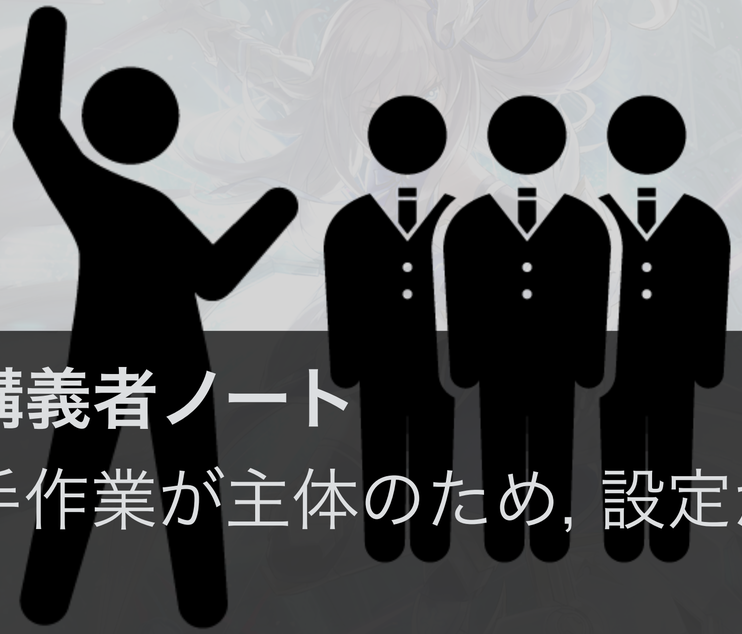
スノーフレーク量産
サイロ化

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2012-2013年)

チーム結成 課題

運用チーム



業務量増加 + チーム問題

▽

属人化(カウボーイ)

構成ドリフト多発

スノーフレーク量産

サイロ化

講義者ノート

手作業が主体のため, 設定が降り積もった謎サーバが量産された.

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2012-2013年)

チーム結成 課題

運用チーム



業務量増加 + チーム問題

▽

属人化(カウボーイ)

構成ドリフト多発

スノーフレーク量産

サイロ化

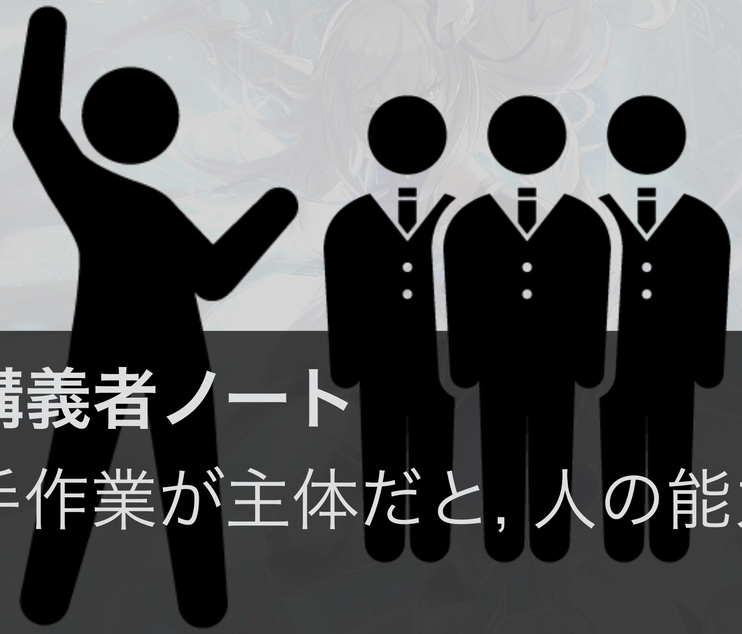
個人能力差

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2012-2013年)

チーム結成 課題

運用チーム



業務量増加 + チーム問題

▽

属人化(カウボーイ)　個人能力差

構成ドリフト多発

スノーフレーク量産

サイロ化

講義者ノート

手作業が主体だと, 人の能力差も出やすい.

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2012-2013年)

チーム結成 課題

運用チーム



業務量増加 + チーム問題

▽

属人化(カウボーイ)　個人能力差

構成ドリフト多発　手戻り　作業ミス

スノーフレーク量産　遅延　設定漏れ

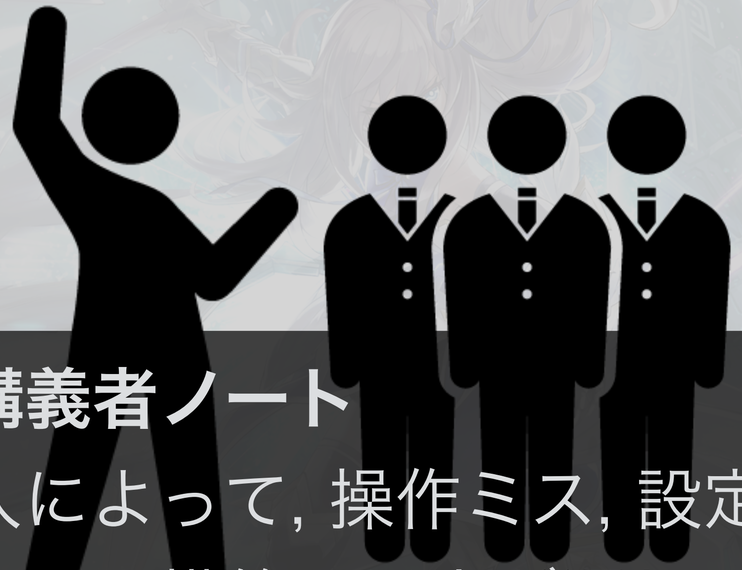
サイロ化

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2012-2013年)

チーム結成 課題

運用チーム



業務量増加 + チーム問題

▽

属人化(カウボーイ)　個人能力差

構成ドリフト多発　手戻り　作業ミス

スノーフレーク　量産　遅延　設定漏れ

サイロ化

講義者ノート

人によって, 操作ミス, 設定漏れが多発する. そうなると手戻りが発生するため, インフラ構築の予定が遅れていく.

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2012-2013年)

チーム結成 課題

運用チーム



業務量増加 + チーム問題

▽

属人化(カウボーイ)　個人能力差

構成ドリフト多発　手戻り　作業ミス

スノーフレーク量産　遅延　設定漏れ

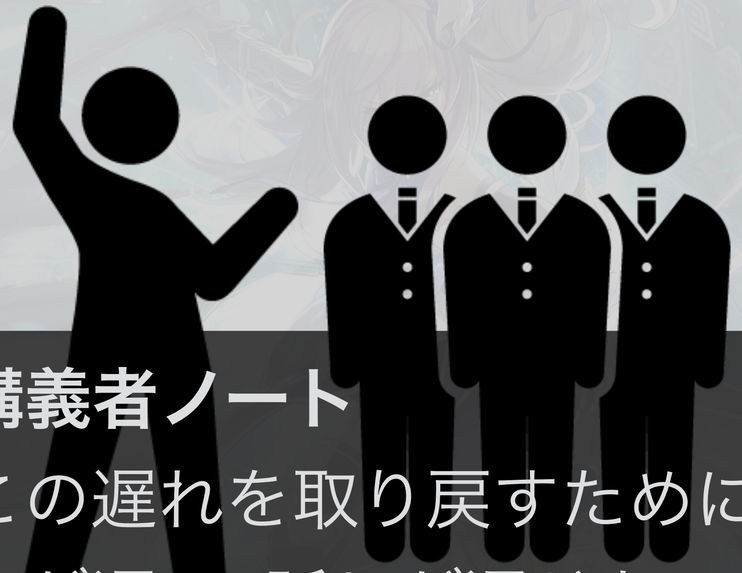
サイロ化　残業でカバー

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2012-2013年)

チーム結成 課題

運用チーム



業務量増加 + チーム問題

▽

属人化(カウボーイ)　個人能力差

構成ドリフト多発　手戻り　作業ミス

スノーフレーク量産　遅延　設定漏れ

サイロ化　残業でカバー

講義者ノート

この遅れを取り戻すために, 残業でカバーし始める. 仕事が多い上に, 周囲から「仕事が遅い. 話しが通じない. ミスが多い.」と怒られる. 本当に辛い職場となった.

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み

ヒーロー登場



2010-2011年

チーム結成



2012-2013年





Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み

ヒーロー登場



2010-2011年

講義者ノート

業務が増えたからといって, 単純に人を増やしたり, 今まで現場のヒーローだった者をチームリーダーに任命しても, 新たな問題が発生する.

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み

ヒーロー登場



2010-2011年

チーム結成



2012-2013年

管理!統制!



2014-2016年



Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み



講義者ノート

ここにきて, やっとシルエットの人物が道具を手に持つ.

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2014-2016年)

管理!統制! 試行

運用管理者
による**合議**



属人化を解消
サイロ化を解消
業務を効率化

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2014-2016年)

管理!統制! 試行



属人化を解消

サイロ化を解消

業務を効率化

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2014-2016年)

管理!統制! 試行



属人化を解消

サイロ化を解消

業務を効率化

講義者ノート

様々な案があり, 実施された. まず, インフラに関する相談と依頼がプライベートチャットや口頭で行われており, チーム全体が知らないタスクを, 一人のメンバーが抱え込んで苦しんでいた. そこで…

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2014-2016年)

管理!統制! 試行



属人化を解消

非公開相談を非推奨

サイロ化を解消

業務を効率化

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2014-2016年)

管理!統制! 試行



属人化を解消

非公開相談を非推奨

講義者ノート

プライベートチャットで相談を受けた際, オープンチャットで相談するよう, 相談者を促すようにした. 加えて…

サイロ化を解消

業務を効率化

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2014-2016年)

管理!統制! 試行



属人化を解消

非公開相談を非推奨

チケット化の徹底

サイロ化を解消

業務を効率化

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2014-2016年)

管理!統制! 試行



属人化を解消

非公開相談を非推奨

チケット化の徹底

講義者ノート

相談から依頼が発生した場合, 必ずチケット化するよう徹底した. また…

サイロ化を解消

業務を効率化

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2014-2016年)

管理!統制! 試行



属人化を解消

非公開相談を非推奨

チケット化の徹底

管理者の承認徹底

サイロ化を解消

業務を効率化

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2014-2016年)

管理!統制! 試行



属人化を解消

非公開相談を非推奨

チケット化の徹底

管理者の承認徹底

講義者ノート

依頼(チケット)に対して, 管理者による確認と承認を徹底した. さらに, プライベートチャット禁止, チケット化徹底などにより, 相談や依頼がし難い…という意見が出ることを予測し…

サイロ化を解消

業務を効率化

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2014-2016年)

管理!統制! 試行



属人化を解消

非公開相談を非推奨

チケット化の徹底

管理者の承認徹底

案件毎に担当制

サイロ化を解消

業務を効率化

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2014-2016年)

管理!統制! 試行



属人化を解消

非公開相談を非推奨

チケット化の徹底

管理者の承認徹底

案件毎に担当制

サイロ化を解消

業務を効率化

講義者ノート

運用チームの中で, ゲームタイトル毎の専任窓口となる担当者を任命した. 担当者は, 各ゲームタイトル毎の背景(社内の立ち位置, リードエンジニアの性格など, 様々な事柄)を理解し, ゲームタイトルの運営に寄り添い…

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2014-2016年)

管理!統制! 試行



属人化を解消

非公開相談を非推奨

チケット化の徹底

管理者の承認徹底

案件毎に担当制

サイロ化を解消

業務の背景を探る

業務を効率化

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2014-2016年)

管理!統制! 試行



属人化を解消

非公開相談を非推奨

チケット化の徹底

管理者の承認徹底

サイロ化を解消

案件毎に担当制

業務の背景を探る

業務を効率化

講義者ノート

依頼の裏に隠れた背景を運用チームに持ち帰る. 例えば,「6/1 までに WEB サーバを 30 台増やして」という依頼があったとする. 何故ですか？と尋ねると「イベントがあります」と回答がある. その場合, アクセス数が増加した場合の本当のボトルネックを探る必要があるため...

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2014-2016年)

管理!統制! 試行



属人化を解消

非公開相談を非推奨

チケット化の徹底

管理者の承認徹底

案件毎に担当制

サイロ化を解消

業務の背景を探る

業務を効率化

講義者ノート

「負荷試験環境を用意する」という依頼チケットが新たに追加となる. また, 負荷試験の後, 本当のボトルネックが DB であれば, WEB サーバ増設が DB サーバの一時的な増強に変更となる. また, 背景を探ると…

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2014-2016年)

管理!統制! 試行



属人化を解消

非公開相談を非推奨

チケット化の徹底

管理者の承認徹底

案件毎に担当制

サイロ化を解消

業務の背景を探る

業務を効率化

講義者ノート

依頼の優先順位が明確になる. その結果, 優先順位の低い依頼は, 依頼元への説明の時間を設け, 期日を延ばしてもらうなどの調整が可能となった. 次に, 運用チームは, 全員が運用オペレーターであったため…

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2014-2016年)

管理!統制! 試行



属人化を解消

非公開相談を非推奨

チケット化の徹底

管理者の承認徹底

ジョブローテーション

案件毎に担当制

業務の背景を探る

サイロ化を解消

業務を効率化

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2014-2016年)

管理!統制! 試行



属人化を解消

非公開相談を非推奨

チケット化の徹底

管理者の承認徹底

ジョブローテーション

案件毎に担当制

業務の背景を探る

サイロ化を解消

業務を効率化

講義者ノート

現場を経験した開発者をチームへ招き入れ, Ansible 導入などの挑戦を行った. 開発チームから人を招き入れたため, サイロ化の解消にも役立った. また…

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2014-2016年)

管理!統制! 試行



属人化を解消

非公開相談を非推奨

チケット化の徹底

管理者の承認徹底

ジョブローテーション

AWS窓口に開発者追加

案件毎に担当制

業務の背景を探る

サイロ化を解消

業務を効率化

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2014-2016年)

管理!統制! 試行



属人化を解消

非公開相談を非推奨

チケット化の徹底

管理者の承認徹底

ジョブローテーション

AWS窓口と開発者追加

案件毎に担当制

業務の背景を探る

サイロ化を解消

業務を効率化

講義者ノート

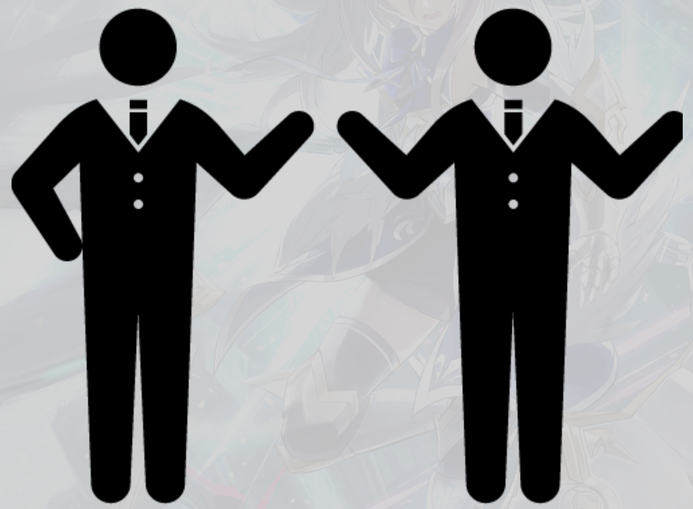
開発と運用, 二つの側面から AWS と向き合うようになり, AWS の新規サービスを早い段階で選定し, 業務に取り込めるようになった.

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2014-2016年)

管理!統制! 課題

運用管理者



管理者の責任増加

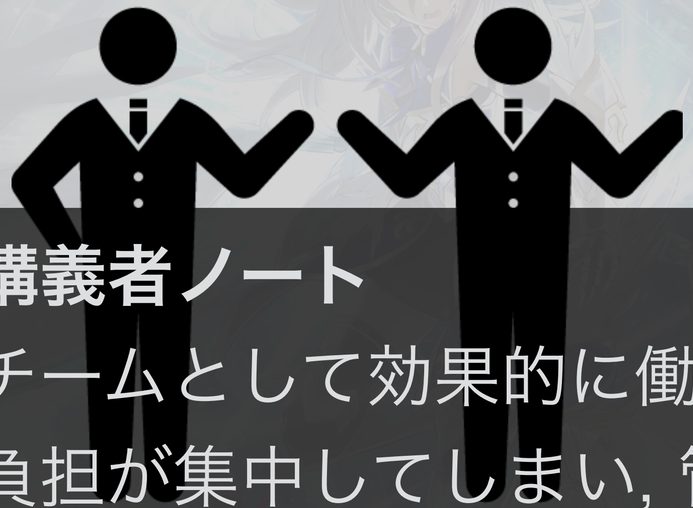
Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2014-2016年)

管理!統制! 課題

管理者の責任増加

運用管理者



講義者ノート

チームとして効果的に働けるように, 様々な施策を行ったが, 運用チームの管理者に負担が集中してしまい, 管理者が業務のボトルネックとなった. 例えば, 手作業の抜け漏れが発生するたび, それを防ごうと…

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2014-2016年)

管理!統制! 課題

運用管理者



管理者の責任増加

▽

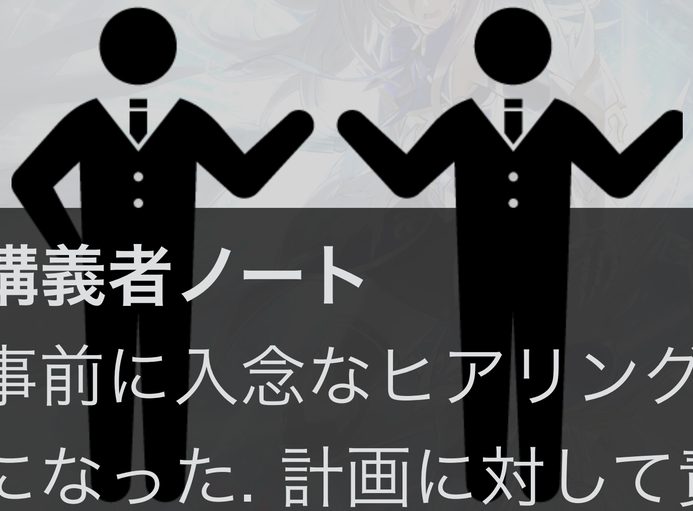
長期の計画フェーズ

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2014-2016年)

管理!統制! 課題

運用管理者



管理者の責任増加

▽

長期の計画フェーズ

講義者ノート

事前に入念なヒアリングを行い, 手堅い計画を立てることに大きな工数を割くようになった. 計画に対して責任を持つのは管理者であるため, 管理者の負担が増えていった.

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2014-2016年)

管理!統制! 課題

運用管理者



管理者の責任増加

▽

長期の計画フェーズ

変化に弱い

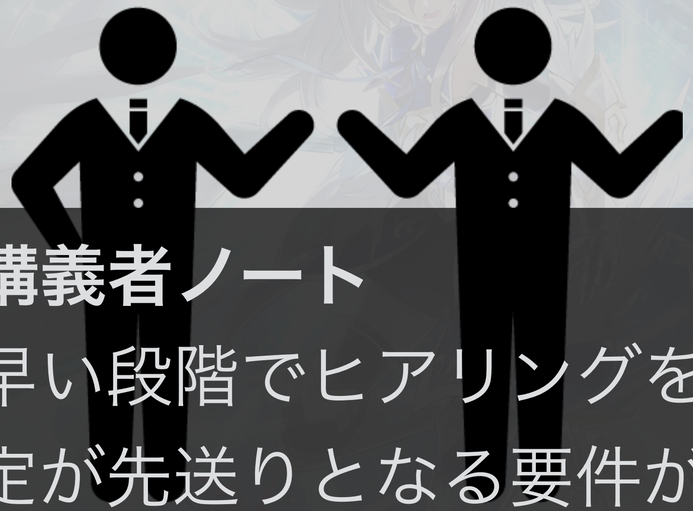
遅延

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2014-2016年)

管理!統制! 課題

運用管理者



管理者の責任増加

▽

長期の計画フェーズ

変化に弱い

遅延

講義者ノート

早い段階でヒアリングをしても, プロジェクト初期段階では不確定要素が多く, 確定が先送りとなる要件が多くあった. また, 確定している要件であっても, 状況の変化と共に変更となることもある. 結果, 手戻りが発生する.

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2014-2016年)

管理!統制! 課題

運用管理者



管理者の責任増加

▽

管理者だけ挑戦

長期の計画フェーズ

変化に弱い

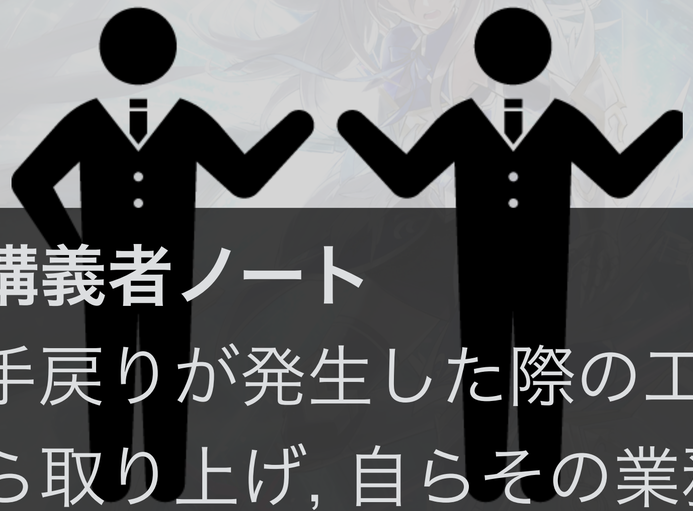
遅延

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2014-2016年)

管理!統制! 課題

運用管理者



管理者の責任増加

▽

管理者だけ挑戦

長期の計画フェーズ

変化に弱い

遅延

講義者ノート

手戻りが発生した際の工数を下げるため, 管理者が難易度の高い業務をスタッフから取り上げ, 自らその業務を行い始める. そうすると, 徐々にスタッフは管理者が残した業務の整理のみが仕事となっていき…

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2014-2016年)

管理!統制! 課題

運用管理者



管理者の責任増加

▽

管理者だけ挑戦

長期の計画フェーズ

メンバー権限不足

変化に弱い

何でもエスカレーション

遅延

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2014-2016年)

管理!統制! 課題

運用管理者



管理者の責任増加

▽

管理者だけ挑戦　　長期の計画フェーズ

ムダ　　権限不足　　変化に弱い

何でもエスカレーション　　遅延

講義者ノート

徐々にスタッフから責任と権限が失われ, 何でも管理者に相談しないと分かりませんという状態となった. また…

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2014-2016年)

管理!統制! 課題

運用管理者



管理者の責任増加

▽

管理者だけ挑戦

長期の計画フェーズ

メンバー権限不足

変化に弱い

中途半端なコード化

何でもエスカレーション

遅延

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2014-2016年)

管理!統制! 課題

運用管理者



管理者の責任増加

▽

管理者だけ挑戦

長期の計画フェーズ

講義者ノート

ジョブローテーションにより取り込んだ開発経験者は運用に忙殺され, コード化が中途半端なところで止まっていた. 一時的にヘルプで開発経験者を新たに送り込んでも, 十分な運用権限が与えられず, 力を発揮できなかった. コード化が進められない状態であるため…

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2014-2016年)

管理!統制! 課題

運用管理者



管理者の責任増加

▽

管理者だけ挑戦

長期の計画フェーズ

構成ドリフト多発

メンバー権限不足

変化に弱い

中途半端なコード化

何でもエスカレーション

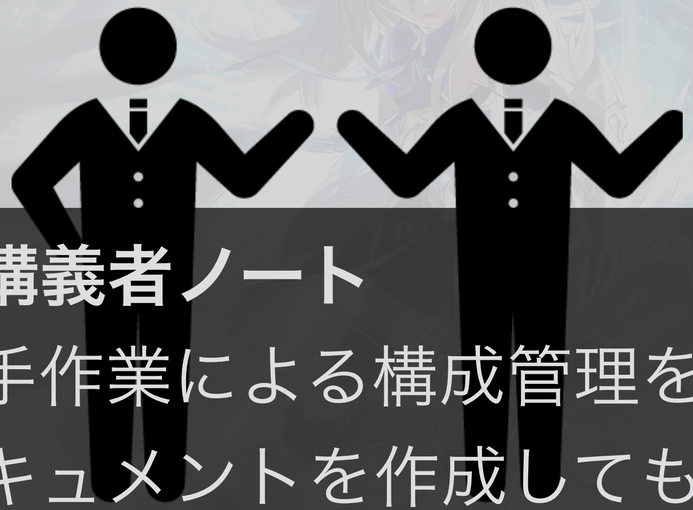
遅延

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2014-2016年)

管理!統制! 課題

運用管理者



管理者の責任増加

▽

管理者だけ挑戦

長期の計画フェーズ

構成ドリフト多発

変化に弱い

何でもエスカレーション

遅延

講義者ノート

手作業による構成管理を続けることとなり, ズノーフレークの量産は止まらない. ドキュメントを作成してもメンテナンス時間を確保できないため, 時間と共に実環境の構成と乖離していく.

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み

ヒーロー登場



2010-2011年

チーム結成



2012-2013年

管理!統制!



2014-2016年



Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み

ヒーロー登場

チーム結成



2010-2011年

2012-2013年

2014-2016年

講義者ノート

管理者による統制を強めたが, 管理者の負担増加により新たな問題が発生した.

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み

ヒーロー登場



2010-2011年

チーム結成



2012-2013年

管理!統制!



2014-2016年

自己組織化



2017-2018年

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2017-2018年)

自己組織化 試行

運用者+開発者
による**合議**



Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2017-2018年)

自己組織化 試行

運用者+開発者
による合議



他社事例？



Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2017-2018年)

自己組織化 試行

運用者+開発者
による**合議**



他社事例？

運用者を増やす



Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2017-2018年)

自己組織化 試行

運用者+開発者
による合議



他社事例？

運用者を増やす



並列度は上っているが
管理者の負担増

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2017-2018年)

自己組織化 試行

運用者+開発者
による合議



他社事例？



Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2017-2018年)

自己組織化 試行

運用者+開発者
による合議



他社事例？







Copyright (C) gumi Inc. 2018 All Rights Reserved

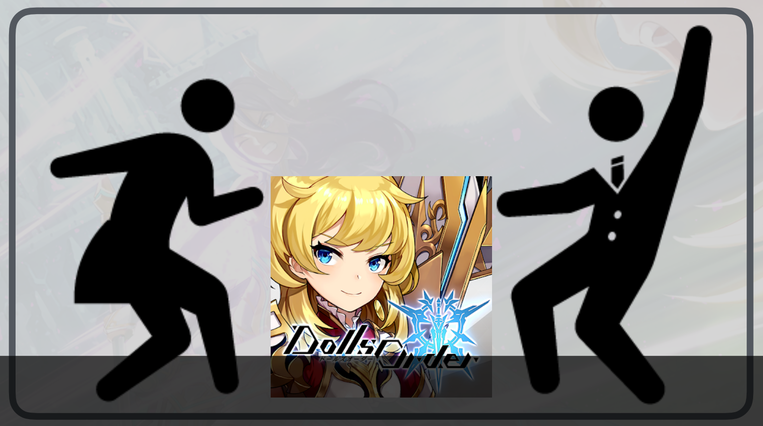
改善の歩み (2017-2018年)

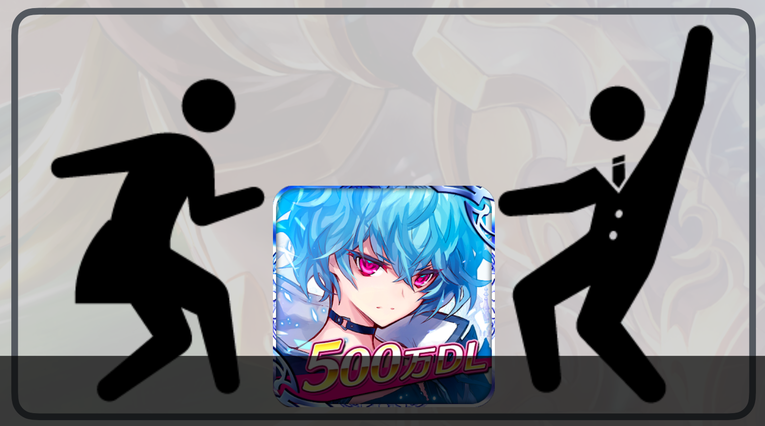
自己組織化 試行

運用者+開発者 による合議



他社事例？







講義者ノート

gumi では, 複数の職種が一つのチームに所属し, ゲーム開発を行っている. インフラ運用者は例外的に, ゲーム開発チームに所属していない. そこで…

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2017-2018年)

自己組織化 試行

運用者+開発者
による**合議**



他社事例？

開発チームが運用?





Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2017-2018年)

自己組織化 試行

運用者+開発者
による**合議**



他社事例？

開発チームが運用?





講義者ノート

他の職種と同じように, インフラ運用者も所属をゲーム開発チームにすると効率が良くなる. これで業務改善できた他社事例もある. よし, コレで行こう.

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2017-2018年)

自己組織化 試行

運用者+開発者
による合議



他社事例？

開発チームが運用?



品質の担保が
非常に大変

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2017-2018年)

自己組織化 試行

運用者+開発者 による合議



他社事例？

開発チームが運用?

品質の担保が非常に大変

講義者ノート

チームごとにインフラ運用スキルの高低差が出るため, インフラの品質が担保できない.

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2017-2018年)

自己組織化 試行

運用者+開発者
による**合議**



要件

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2017-2018年)

自己組織化 試行

運用者+開発者
による合議



要件

品質と効率の両立

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2017-2018年)

自己組織化 試行

運用者+開発者
による合議



要件

品質と**効率**の両立

という無茶振り…

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2017-2018年)

gumi

自己組織化 試行

運用者+開発者
による**合議**



要件を満たすため…

品質

インフラに責任を持つ**チーム**を**維持**

効率

AWS 活用でもっと**コード化**

その他

チケット化, 担当制, オープンな会話,
背景を探る, ジョブローテーション

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2017-2018年)

自己組織化 試行

運用者+開発者
による合議



要件を満たすため…

品質
インシデント責任者のクオリティを**維持**

効率
AWS利用料でもっとコストを…

その他
チケット化, 担当制, オープンな会話,
背景を探る, ジョブローテーション

現状維持で

精一杯

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2017-2018年)

自己組織化 試行

運用者+開発者
による**合議**



社内外から数名**招聘**する方針決定

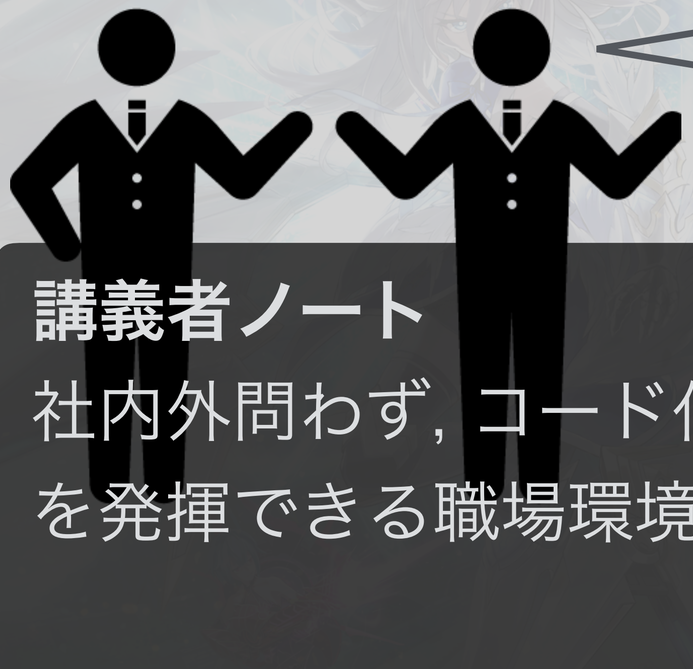


Copyright (C) gumi Inc. 2018 All Rights Reserved

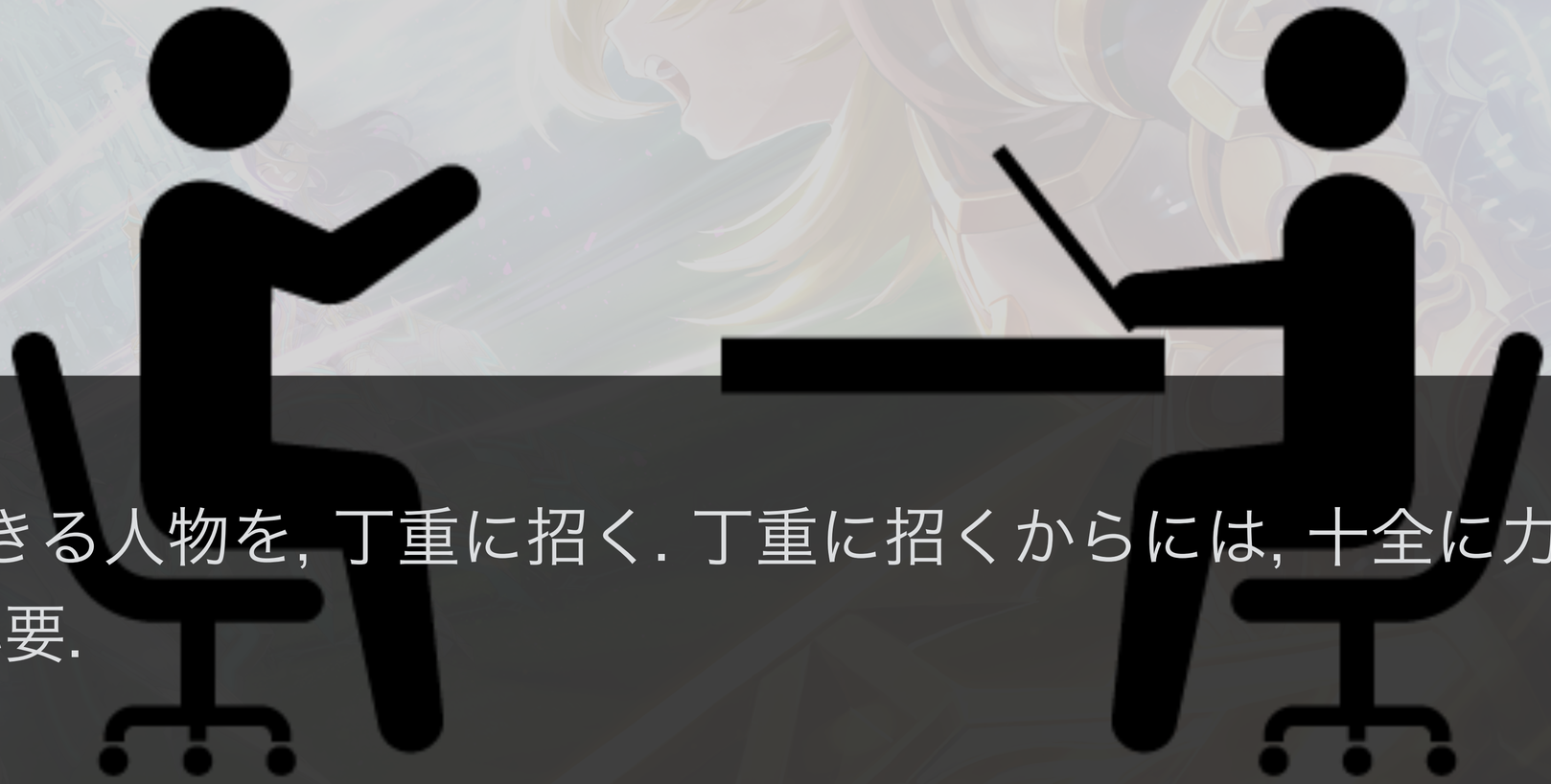
改善の歩み (2017-2018年)

自己組織化 試行

運用者+開発者
による**合議**



社内外から数名**招聘**する方針決定



講義者ノート

社内外問わず, コード化できる人物を, 丁重に招く. 丁重に招くからには, 十全に力を発揮できる職場環境が必要.

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2017-2018年)

自己組織化 試行

運用者+開発者
による**合議**



社内外から数名**招聘**する方針決定

インフラに関する**権限**と**責任**を渡す

学習と**挑戦**ができる**環境**を用意する

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2017-2018年)

自己組織化 試行

運用者+開発者
による**合議**



社内外から数名**招聘**する方針決定

インフラに関する**権限**と**責任**を渡す

コード化に長けている, 手作業が嫌い,
人助けが大好き,
自分や他人が楽になることに全力

学習と**挑戦**ができる**環境**を用意する

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2017-2018年)

自己組織化 試行

運用者+開発者
による**合議**



社内外から数名**招聘**する方針決定

インフラに関する**権限**と**責任**を渡す

コード化に長けている, 手作業が嫌い,
人助けが大好き,
自分や他人が楽になることに全力

学習と**挑戦**ができる**環境**を用意する

講義者ノート

権限を渡すため, 高い能力が必要. このような切羽詰まった状態で探せるのかという不安はあったが, すぐに見つからなくとも, 手作業を伴うインフラ業務をアウトソーシングし, 時間稼ぎをしようと考えていた.

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2017-2018年)

自己組織化 試行

運用者+開発者
による**合議**



社内外から数名**招聘**する方針決定

インフラに関する**権限**と**責任**を渡す

コード化に長けている, 手作業が嫌い,
人助けが大好き,
自分や他人が楽になることに全力

学習と**挑戦**ができる**環境**を用意する

AWSトレーニング, 関連書籍,
実験アカウント, 座席配置,
AWSマグネット etc…

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2017-2018年)

自己組織化 試行

運用者+開発者
による**合議**



社内外から数名**招聘**する方針決定

インフラに関する**権限**と**責任**を渡す

コード化に長けている, 手作業が嫌い,
人助けが大好き,
自分や他人が楽になることに全力

学習と**挑戦**ができる**環境**を用意する

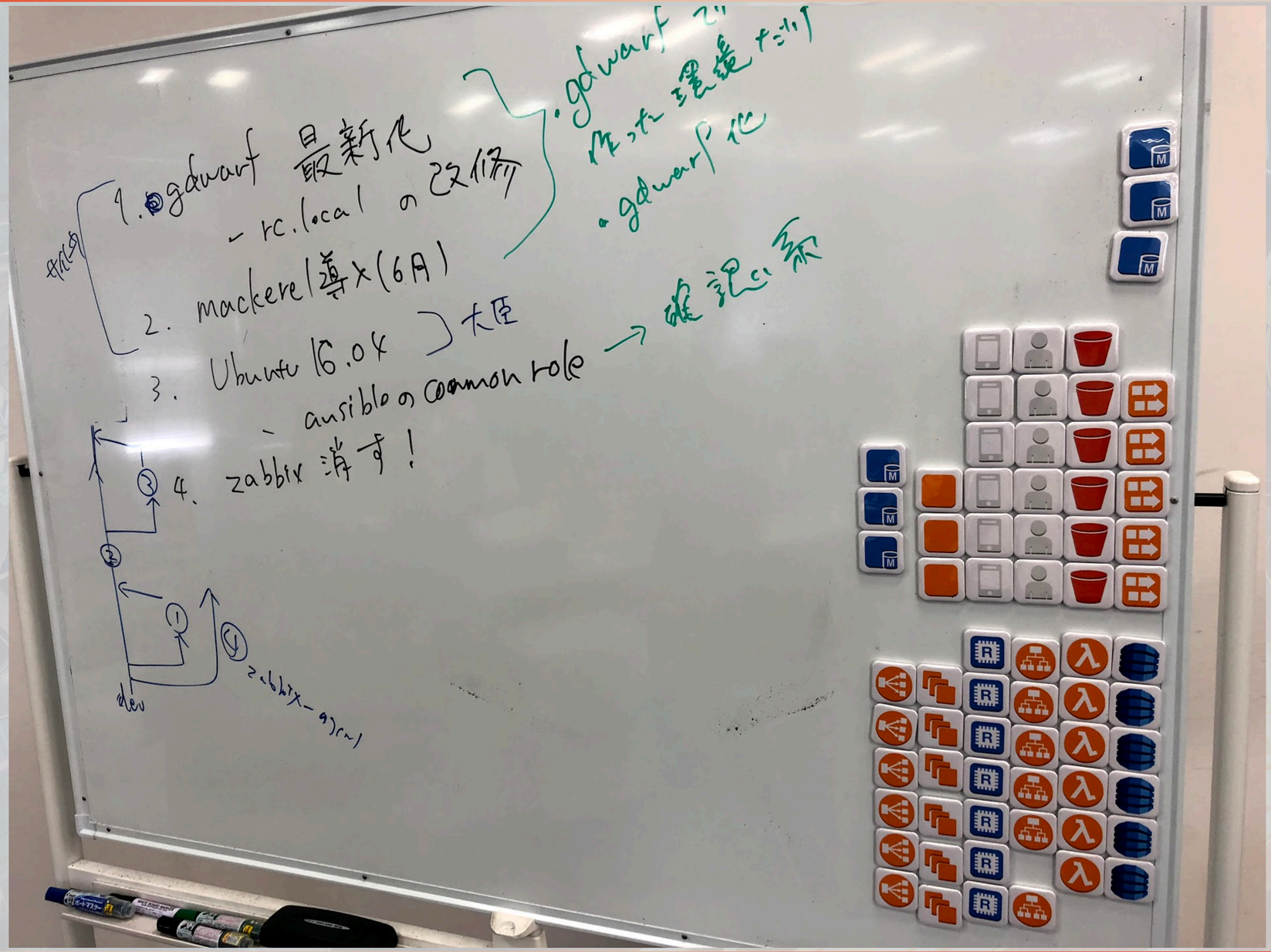
AWSトレーニング, 関連書籍,
実験アカウント, 座席配置,
AWSマグネット etc…

講義者ノート

AWS トレーニング受講や Certificate 取得, 関連書籍の購入, AWS 検証アカウントなどの予算確保を行った. また…

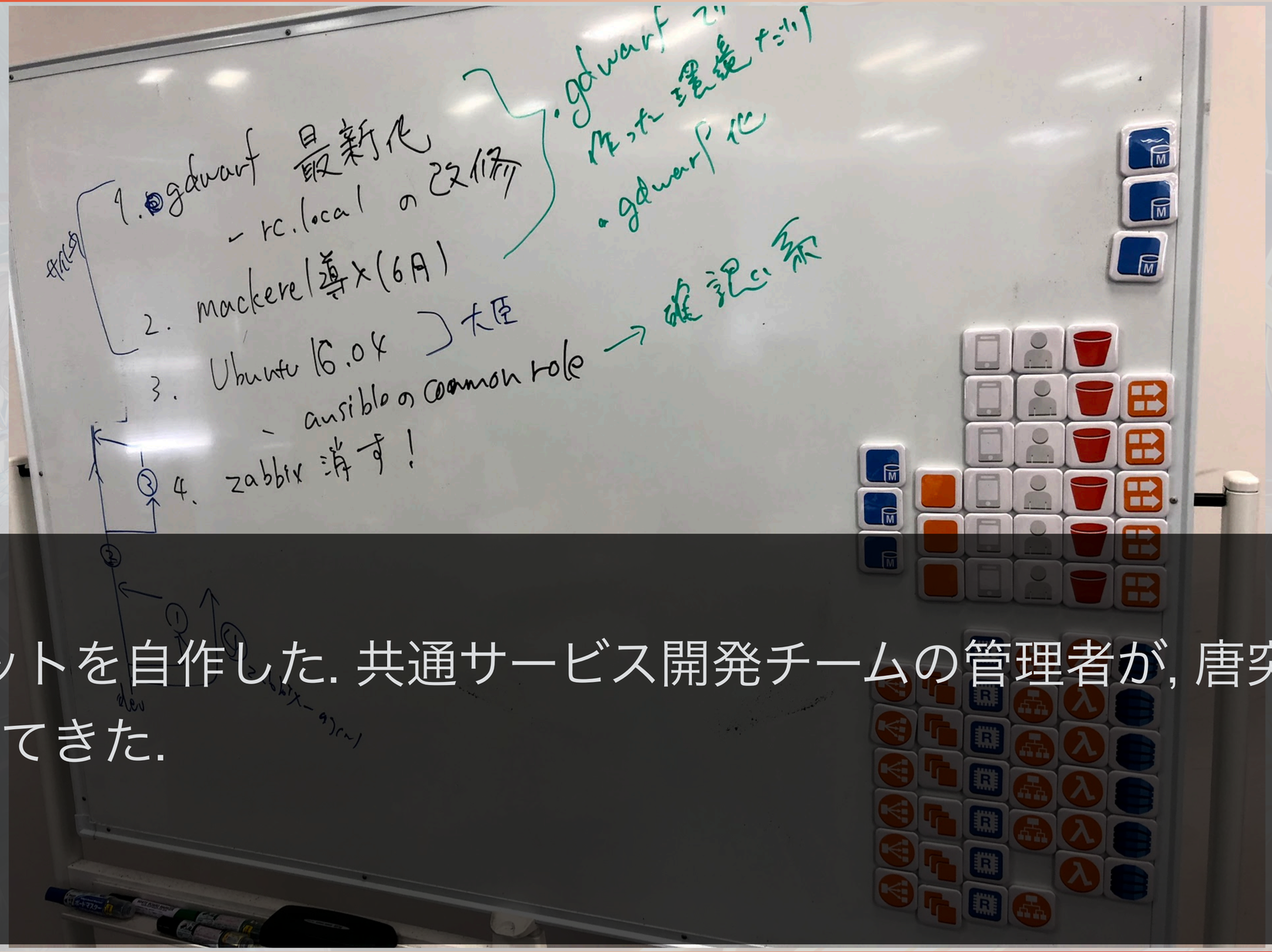
Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2017-2018年)



Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2017-2018年)



講義者ノート

AWS マグネットを自作した. 共通サービス開発チームの管理者が, 唐突に自腹で作って会社に持ってきた.

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2017-2018年)



Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2017-2018年)



講義者ノート

島配置+ひな壇を廃止して, 谷配置に変更. ミーティングスペースとお菓子スペースを近づけるなど工夫を行った. 結果…

Copyright (C) gumi Inc. 2018 All Rights Reserved



7-8ヶ月ほどで二名の

招聘に成功

※一年後にさらに一名

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2017-2018年)

自己組織化 試行

新生**インフラチーム**が**考えて行った**こと

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2017-2018年)

自己組織化 試行

新生**インフラチーム**が**考えて行った**こと

一般的な開発とリリースのフローを導入

Pull Request とレビュー, リリースブランチの運用

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2017-2018年)

自己組織化 試行

新生インフラチームが考えて行ったこと

一般的な開発とリリースのフローを導入

Pull Request とレビュー, リリースブランチの運用

講義者ノート

開発現場だと当たり前のことだが, Pull Request 運用やリリースブランチの管理を行っている. 開発したツールだけではなく, CloudFormation のテンプレートや Ansible Playbook などの YAML ファイルもチーム内でレビューし合う.

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2017-2018年)

自己組織化 試行

新生**インフラチーム**が**考えて行った**こと

一般的な開発とリリースのフローを導入

Pull Request とレビュー, リリースブランチの運用

開発チームへの段階的な権限移譲

AutoScaling 操作などを開発チームが行う

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2017-2018年)

自己組織化 試行

新生**インフラチーム**が**考えて行った**こと

一般的な開発とリリースのフローを導入

Pull Request とレビュー, リリースブランチの運用

開発チームへの段階的な権限移譲

AutoScaling 操作などを開発チームが行う

講義者ノート

全ての権限をインフラチームだけで囲い込むのではなく, 部分的に権限を開発チームへ移譲し, 業務の効率化を計る.

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2017-2018年)

自己組織化 試行

新生インフラチームが考えて行ったこと

一般的な開発とリリースのフローを導入

Pull Request とレビュー, リリースブランチの運用

開発チームへの段階的な権限移譲

AutoScaling 操作などを開発チームが行う

Management Console と SSH は基本禁止

構成ドリフトを防ぐ

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2017-2018年)

自己組織化 試行

新生インフラチームが考えて行ったこと

一般的な開発とリリースのフローを導入

Pull Request とレビュー, リリースブランチの運用

開発チームへの段階的な権限移譲

AutoScaling 操作などを開発チームが行う

Management Console と SSH は基本禁止

構成ドリフトを防ぐ

講義者ノート

手作業の温床を撲滅する.

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2017-2018年)

自己組織化 試行

新生インフラチームが考えて行ったこと

一般的な開発とリリースのフローを導入

Pull Request とレビュー, リリースブランチの運用

開発チームへの段階的な権限移譲

AutoScaling 操作などを開発チームが行う

Management Console と SSH は基本禁止

構成ドリフトを防ぐ

ツールを適材適所で使い分ける

次の章で紹介

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2017-2018年)

自己組織化 試行

新生**インフラチーム**が**無意識で行った**こと

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2017-2018年)

自己組織化 試行

新生**インフラチーム**が**無意識で行った**こと

クイックミーティングを座席で**頻繁**に**開催**

情報共有, 疑問解消, 意思決定が早い

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2017-2018年)

自己組織化 試行

新生**インフラチーム**が**無意識で行った**こと

クイックミーティングを座席で**頻繁**に**開催**

情報共有, 疑問解消, 意思決定が早い

想定以上に管理者から権限移譲されていた

管理者の負担軽減となる

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2017-2018年)

自己組織化 試行

新生**インフラチーム**が**無意識で行った**こと

クイックミーティングを座席で**頻繁**に**開催**

情報共有, 疑問解消, 意思決定が早い

想定以上に管理者から権限移譲されていた

管理者の負担軽減となる

開発チームから**高い信頼**を得ていた

権限と責任を持って対話している

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2017-2018年)

自己組織化 試行

開発チームから高い信頼を得ていた

～開発現場の Technical Director からインタビュー～

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2017-2018年)

自己組織化 試行

開発チームから高い信頼を得ていた

～開発現場の Technical Director からインタビュー～

現場の意見を聞いてもらえる



以前は, 運用チームに何か相談しても, 運用の偉い人から,
「管理上, 無理です」と却下されていた.
今は, 意見が通りやすく, 現場判断で物事が進む印象がある.

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2017-2018年)

自己組織化 試行

開発チームから高い信頼を得ていた

～開発現場の Technical Director からインタビュー～

現場の意見を聞いてもらえる



以前は, 運用チームに何か相談しても, 運用の偉い人から,
「管理上, 無理です」と却下されていた.
今は, 意見が通りやすく, 現場判断で物事が進む印象がある.

信頼されて任せてもらえる



インフラは怖いと思っており, 権限を譲渡された初めのころ,
少し気持ちに抵抗があった. 実際に運用してみると, 業務効率
が格段に良くなって驚いた. もっと, いろいろ任せてほしい.

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2017-2018年)

自己組織化 試行

開発チームから高い信頼を得ていた

～開発現場の Technical Director からインタビュー～

現場の意見を聞いてもらえる



以前は, 運用チームに何か相談しても, 運用の偉い人から,
「管理上, 無理です」と却下されていた.
今は, 意見が通りやすく, 現場判断で物事が進む印象がある.



インフラは怖いと思っており, 権限を譲渡された初めのころ,
リーダー気持ちで抵抗があったが, 実際に運用してみると, 業務効率
が格段に良くなって驚いた. もっと, いろいろ任せてほしい.

講義者ノート

「キャパシティプランニングの必要があり, インフラチームへ頻繁に相談へ伺っている. これを通じて, 自分たちにインフラ知識が必要だと感じ始めた」とのこと. 彼らも AWS トレーニングなどを受ける予定.

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2017-2018年)

自己組織化 試行



Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2017-2018年)

自己組織化 試行

急な差し込み対応

高品質

日々の運用

5人の
インフラエンジニア

リビルド歓迎

新環境5営業日内!!



Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2017-2018年)

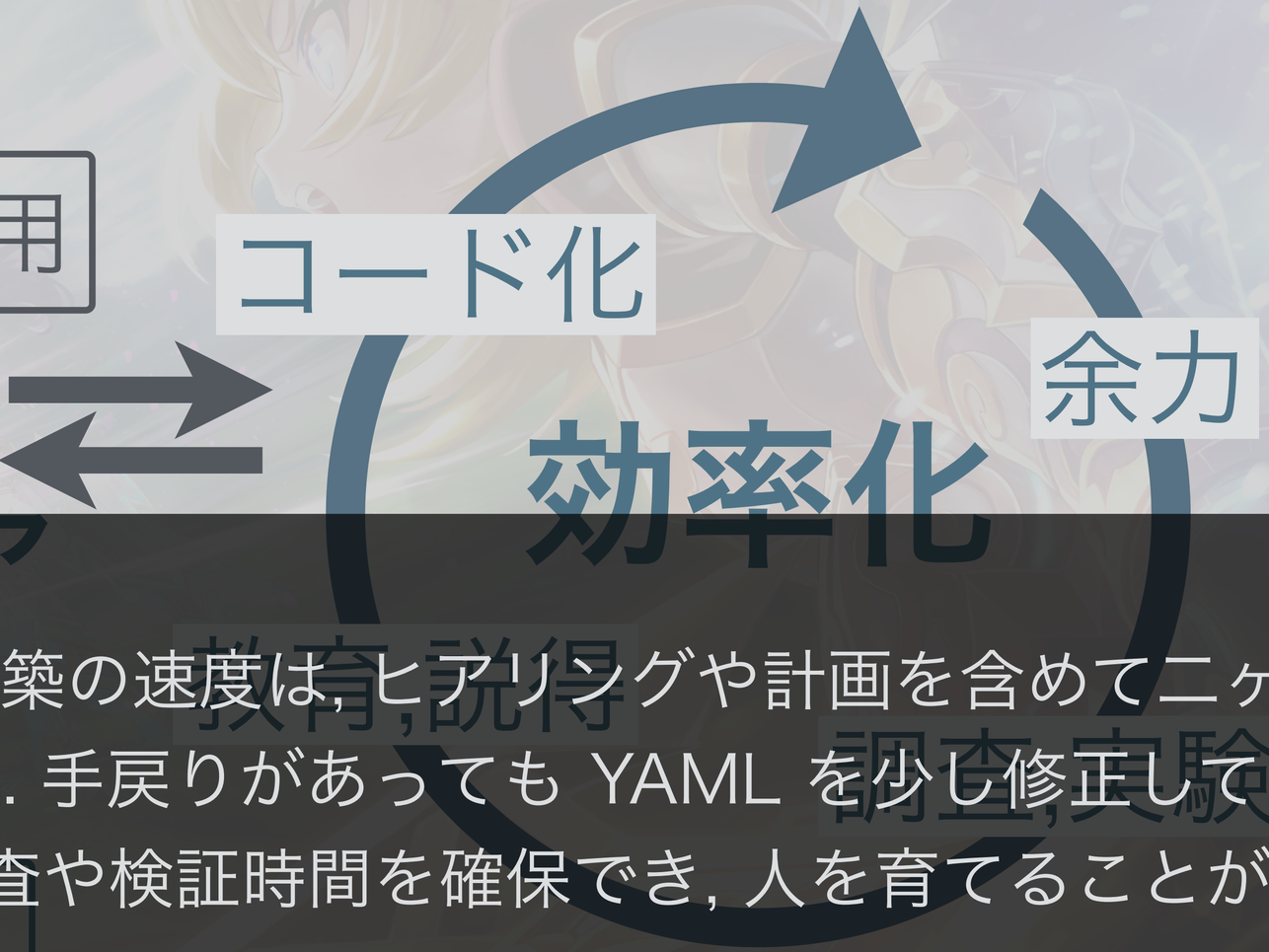
自己組織化 試行

急な差し込み対応

高品質

5人の インフラエンジニア

日々の運用



講義者ノート

冒頭の胡散臭い図へ戻る. 新規の環境構築の速度は, ヒアリングや計画を含めて二ヶ月掛かっていたが, 五日間に短縮された. 手戻りがあっても YAML を少し修正してリビルドするだけ. 残業時間が減り, 調査や検証時間を確保でき, 人を育てることが可能となった. しかし…

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2017-2018年)

自己組織化 課題



Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2017-2018年)

自己組織化 課題



旧環境との戦い

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2017-2018年)

自己組織化 課題



旧環境との戦い

開発力に優れる者が犠牲となり
墓守(辛い手作業) or リプレイス

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2017-2018年)

自己組織化 課題



旧環境との戦い

開発力に優れる者が犠牲となり
墓守(辛い手作業) or リプレイス

コード化は
コンテキストスイッチ発生で
効率が落ちるため墓守役が偏る

早く負債を返済したい…

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2017-2018年)

自己組織化 課題 (予測)



Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2017-2018年)

自己組織化 課題 (予測)



コード化の罠
～まだ未体験～

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2017-2018年)

自己組織化 課題 (予測)



コード化の罠
～まだ未体験～

あとから入社すると
知識0で作業可能

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2017-2018年)

自己組織化 課題 (予測)



コード化の罠
～まだ未体験～

あとから入社すると
知識0で作業可能

多大なコストを支払って
巨大で複雑な仕組み

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2017-2018年)

自己組織化 課題 (予測)



コード化の罠

～まだ未体験～

あとから入社すると
知識0で作業可能

多大なコストを支払って
巨大で複雑な仕組み

新たな**聖域**が**誕生？**

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2017-2018年)

自己組織化 課題 (予測)



コード化の罠
～まだ未体験～

あとから入社すると
知識0で作業可能

多大なコストを支払って
巨大で複雑な仕組み

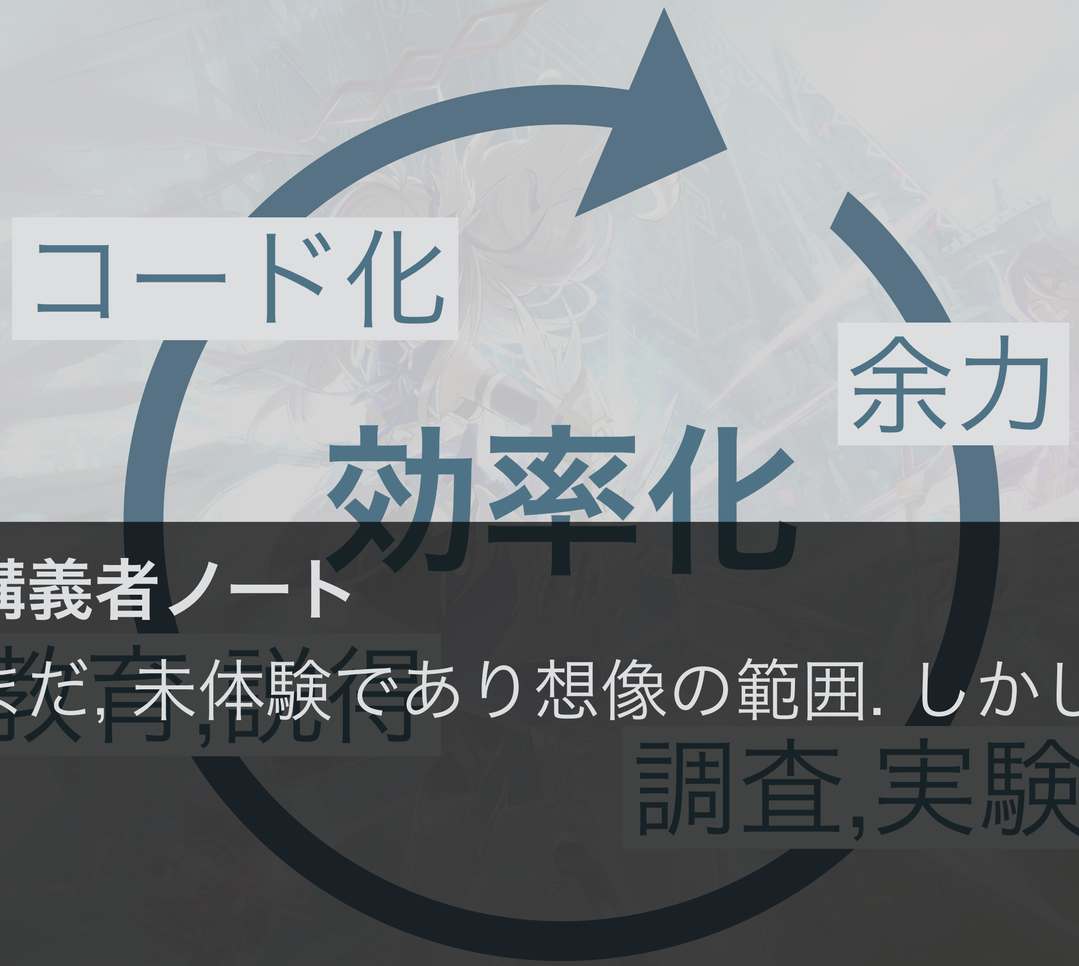
新たな**聖域**が**誕生？**

改善は続く!!!

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み (2017-2018年)

自己組織化 課題 (予測)



コード化の罠
～まだ未体験～

あとから入社すると
知識0で作業可能

多大なコストを支払って
巨大で複雑な仕組み
新たな**聖域**が**誕生？**

講義者ノート

まだ,未体験であり想像の範囲. しかし, 問題が発生するならば, 改善すれば良い.

改善は続く!!!

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み

ヒーロー登場



2010-2011年

チーム結成



2012-2013年

管理!統制!



2014-2016年

自己組織化



2017-2018年

Copyright (C) gumi Inc. 2018 All Rights Reserved

改善の歩み

ヒーロー登場



2010-2011年

チーム結成



2012-2013年

管理!統制!



2014-2016年

自己組織化



2017-2018年

Copyright (C) gumi Inc. 2018 All Rights Reserved

目次

悪戦苦闘した
改善の歩み
は8年かけても終わらない

▽

改善を促進する
コード化
の進め方は試行錯誤の連続

▽

まとめ

Copyright (C) gumi Inc. 2018 All Rights Reserved

目次

悪戦苦闘した
改善の歩み
は8年かけても終わらない

▽

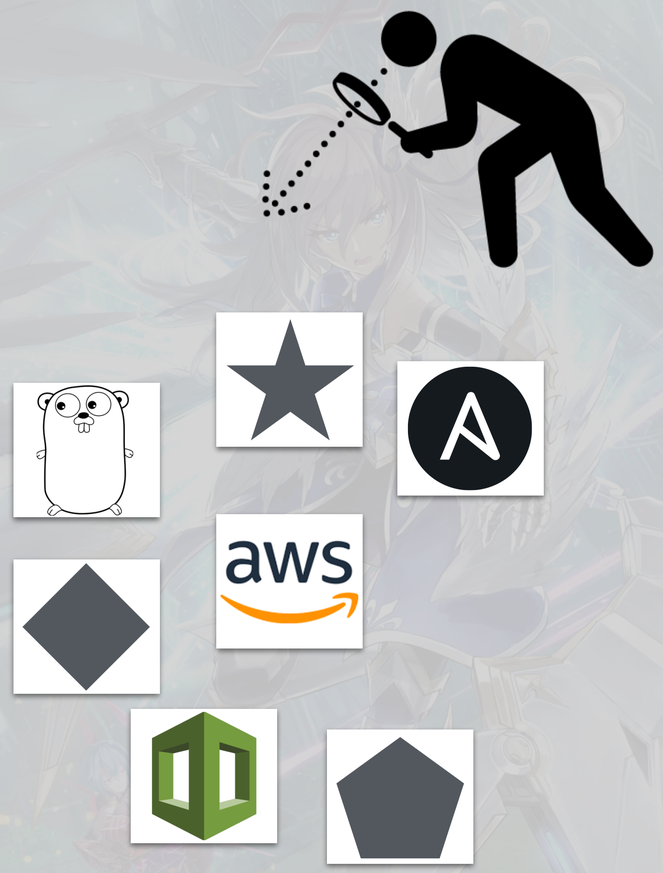
改善を促進する
コード化
の進め方は試行錯誤の連続

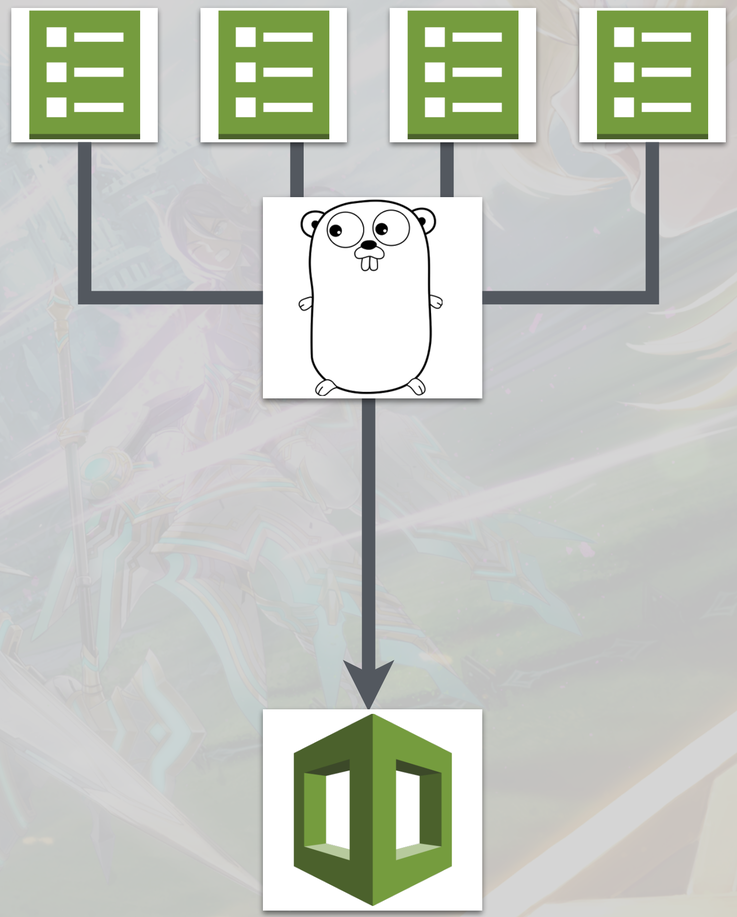
▽

まとめ

Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化







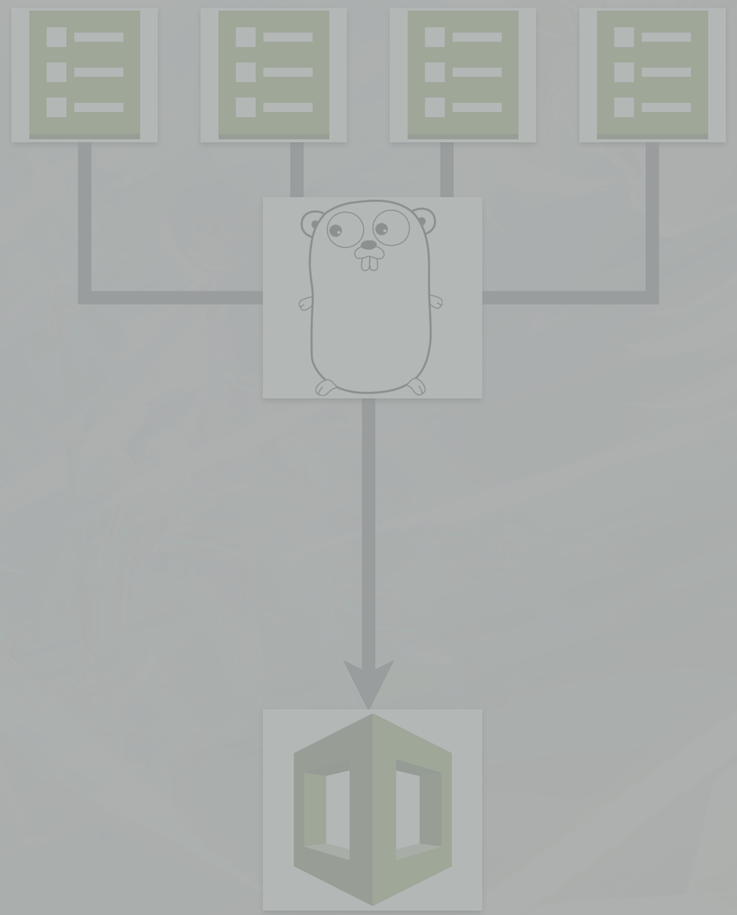
Copyright (C) gumi Inc. 2018 All Rights Reserved

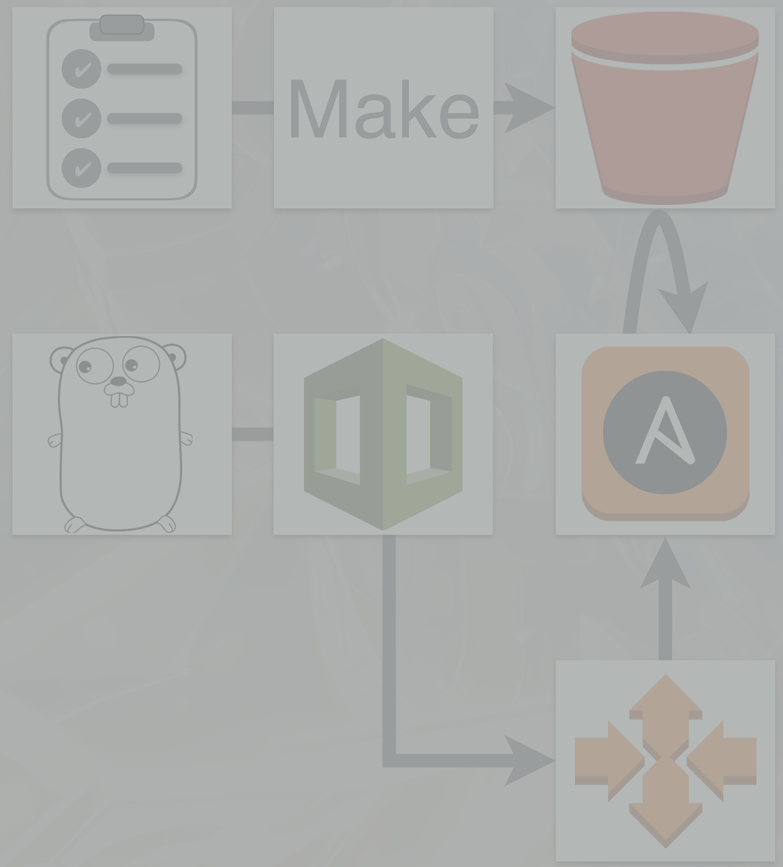
コード化

ツールを選択









Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツールを**選択**する際に**大切**なこと



Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツールを**選択**する際に**大切**なこと

実験的であること

Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツールを**選択**する際に**大切**なこと

実験的であること

学習意欲をもって改善し続ける姿勢と環境が大切
素早く試す / 小さく本番に適用する

Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツールを**選択**する際に**大切**なこと

実験的であること

学習意欲をもって改善し続ける姿勢と環境が大切
素早く試す / 小さく本番に適用する

講義者ノート

新しいものは, すぐに調べて, すぐに試す. 自分なりで良いから, 技術やツールに対する意見を持つこと. 業務で使っているものだけに長けていれば良い…という考えは捨てること. 古いものに固執しないこと. 同じものでもバージョンを上げること.

Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツールを**選択**する際に**大切**なこと

実験的であること

学習意欲をもって改善し続ける姿勢と環境が大切
素早く試す / 小さく本番に適用する

講義者ノート

調査, 検証して良さそうなら, 早めに製品環境に導入すること. 実際に現場で使ってみないと解らないことが多くある. ただし, 製品環境に導入する際, リクスを抑えるため, 小さく狭い範囲で導入し, 問題があれば, すぐに戻せるようにすること.

Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツールを**選択**する際に**大切**なこと

実験的であること

学習意欲をもって改善し続ける姿勢と環境が大切
素早く試す / 小さく本番に適用する

現場ごとの背景を理解すること

Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツールを**選択**する際に**大切**なこと

実験的であること

学習意欲をもって改善し続ける姿勢と環境が大切
素早く試す / 小さく本番に適用する

現場ごとの背景を理解すること

制約や条件が異なれば適するツールは変わる
思考停止で同じツールを使う前に, 背景を探る

Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツールを**選択**する際に**大切**なこと

実験的であること

学習意欲をもって改善し続ける姿勢と環境が大切
素早く試す / 小さく本番に適用する

現場ごとの背景を理解すること

制約や条件が異なれば適するツールは変わる
思考停止で同じツールを使う前に, 背景を探る

講義者ノート

「なぜ, あの現場では, あのツールを使っているのか？」という現場ごとの背景を理解すること. 各現場ごとに制約や条件が異なるため, 合うツールが異なる. 他社事例を聞く場合, 自社と他社の背景(制約と条件)を真っ先に比較すること.

Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツールを**選択**する際に**大切**なこと

実験的であること

学習意欲をもって改善し続ける姿勢と環境が大切
素早く試す / 小さく本番に適用する

現場ごとの背景を理解すること

制約や条件が異なれば適するツールは変わる
思考停止で同じツールを使う前に, 背景を探る

適材適所で使うこと

Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツールを**選択**する際に**大切**なこと

実験的であること

学習意欲をもって改善し続ける姿勢と環境が大切
素早く試す / 小さく本番に適用する

現場ごとの背景を理解すること

制約や条件が異なれば適するツールは変わる
思考停止で同じツールを使う前に, 背景を探る

適材適所で使うこと

ツールの本質を理解し, 正しく使う
組み合わせで状況に対応する

Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツールを**選択**する際に**大切**なこと

実験的であること

学習意欲をもって改善し続ける姿勢と環境が大切
素早く試す / 小さく本番に適用する

現場ごとの背景を理解すること

制約や条件が異なれば適するツールは変わる
思考停止で同じツールを使う前に, 背景を探る

適材適所で使うこと

ツールの本質を理解し, 正しく使う
組み合わせで状況に対応する

講義者ノート

あらゆる全ての言語やツール, サービスに向き不向きがある. 本質を捉え, 正しく使い, 正しく組み合わせること. すぐに「vs.」対立構造で語るのは考察不足.

Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツールを選択



Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツールを選択 aws

Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

gumi

ツールを選択 aws

とても良い!

Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

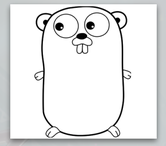
ツールを選択



Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

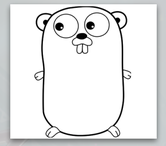
ツールを選択



Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツールを選択

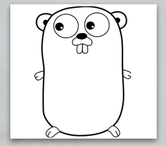


クロスコンパイル　シングルバイナリ
言語仕様がシンプル

Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツールを選択



配布が簡単　　引き継ぎが簡単

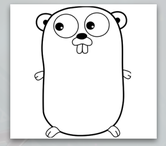
△

クロスコンパイル　シングルバイナリ
言語仕様がシンプル

Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツールを選択



配布が簡単　　引き継ぎが簡単

クロスコンパイル　シングルバイナリ

言語仕様がシンプル

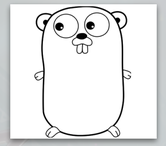
講義者ノート

依存関係を気にしながら配置する必要がない. 設置するとすぐ動く. 少ない学習コストで, すぐにコードの読み書きができる. インフラのコード化に適した言語.

Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツールを選択



運用コスト低下

△

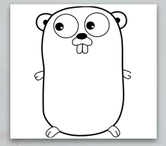
配布が簡単　引き継ぎが簡単

△

クロスコンパイル　シングルバイナリ
言語仕様がシンプル

Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツールを**選択** 

運用コスト低下

△

配布が簡単　引き継ぎが簡単

△

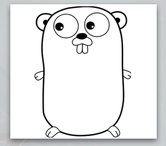
クロスコンパイル　シングルバイナリ
言語仕様がシンプル

手間を惜しまずコード化する　**好きな言語**で良い

Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツールを選択



運用コスト低下

△

配布が簡単　　引き継ぎが簡単

△

講義者ノート

各現場の背景に合わせた言語を採用すること. 選定基準は, 社内にその言語に精通しているエンジニアが多数在籍しているなどで良い. 世の中には, Go が非知的であるがゆえに, Go でコードを書くことが苦痛な人もいる. 一方で, 非知的な言語を採用することで恩恵を受ける現場もある.

Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツールを選択



Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツールを選択 

Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツールを選択 

中央サーバ不要　　工夫次第でSSH不要
構成がシンプル

Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツールを選択 

管理が簡単　　引き継ぎが簡単

△

中央サーバ不要　　工夫次第でSSH不要

構成がシンプル

Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

gumi

ツールを選択 

運用コスト低下

△

管理が簡単　　引き継ぎが簡単

△

中央サーバ不要　　工夫次第でSSH不要

構成がシンプル

Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

gumi

ツールを選択 

運用コスト低下

△

管理が簡単　　引き継ぎが簡単

△

中央サーバ不要　　工夫次第でSSH不要

構成がシンプル

手間を惜しまずコード化する　**好きなツール**で良い

Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツールを選択 

運用コスト低下

△

管理が簡単　　引き継ぎが簡単

△

中央サーバ不要　　証明書でSSL不要

構成がシンプル

手間を惜しまずコード化する　**好きなツール**で良い

講義者ノート

OpsWorks が合う現場もある. OpsWorks でも同様のことが実現できる. 後ほど, gumi での Ansible の使い方を紹介するが, それを参照すると gumi に Ansible が合う理由が分かる. 蛇足だが, gumi 海外拠点では SaltStack を採用している.

Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツールを選択



Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツールを選択 

Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツールを選択 

安心と信頼の AWS 純正

Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツールを選択 

問い合わせが簡単
新サービス対応が早い

△

安心と信頼の AWS 純正

Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツールを選択 

問い合わせが簡単
新サービス対応が早い

安心と信頼の AWS 純正

講義者ノート

何か問題が発生した際, クラウドサービス側の問題なのか？ツール側の問題なのか？などの切り分けをせず, すぐに AWS に問い合わせできる. 新サービスへの対応は, 同じ用途のツールの中で最も早い.

Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツールを選択 

運用コスト低下

△

問い合わせが簡単
新サービス対応が早い

△

安心と信頼の AWS 純正

Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツールを選択 

運用コスト低下

△

問い合わせが簡単
新サービス対応が早い

△

安心と信頼の AWS 純正

手間を惜しまずコード化する **好きなツール**で良い

Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツールを選択 

運用コスト低下

△

問い合わせが簡単
新サービス対応が早い

講義者ノート

AWS エンタープライズサポートを契約しているならば, CloudFormation を強く推奨する. 蛇足だが, Elastic Beanstalk という選択肢もある. Elastic Beanstalk によって敷かれたレールに乗れるならば, 良い選択肢. gumi ではレールから脱線したため, CloudFormation を採用した.

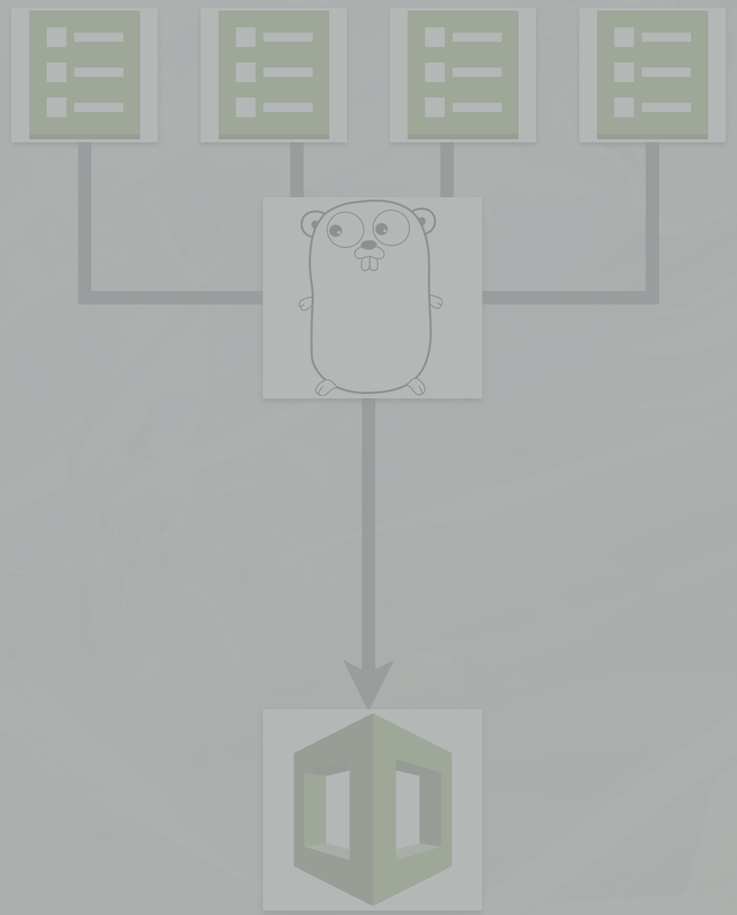
Copyright (C) gumi Inc. 2018 All Rights Reserved

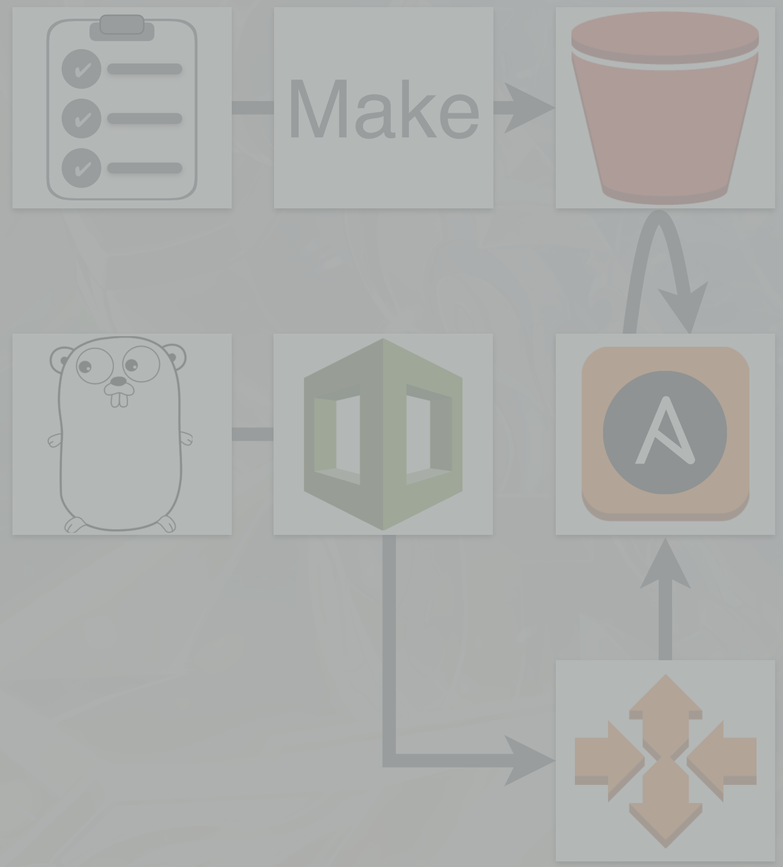
コード化

ツールを選択









Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツールを選択









講義者ノート

現場の背景を鑑みて, 運用コストが下がるようにツールを選択すること.

Copyright (C) gumi Inc. 2018 All Rights Reserved

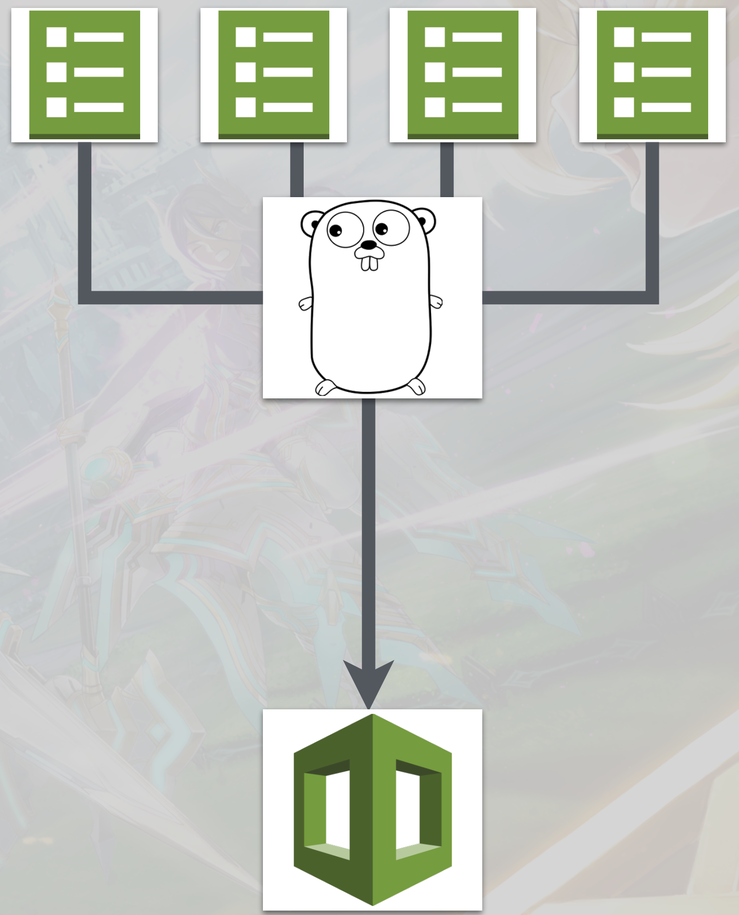
コード化

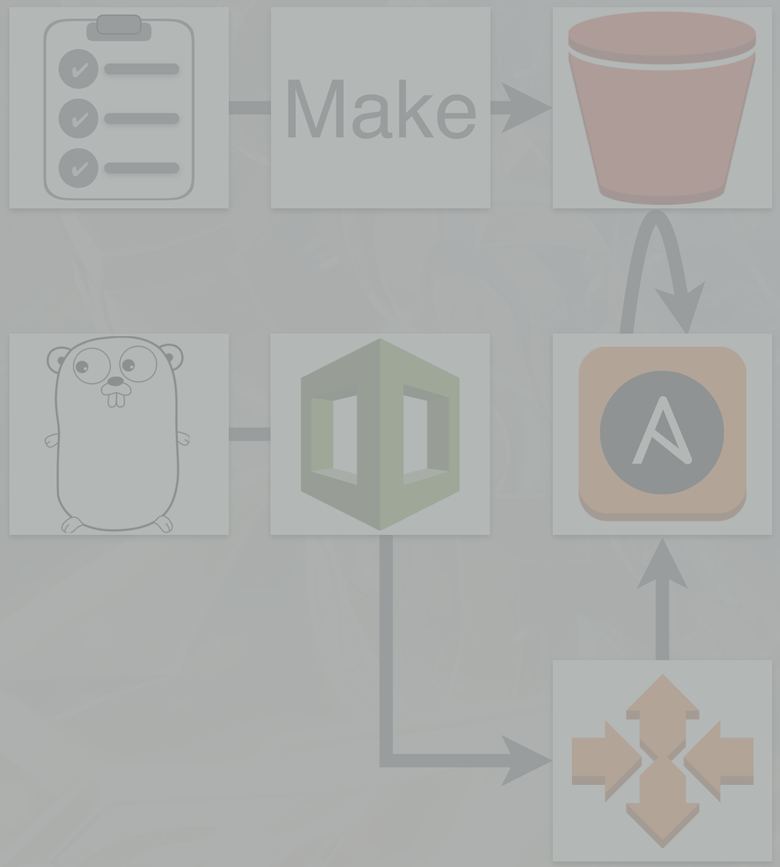
ツールを選択





コードで補う





Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

コードで補う

CloudFormationを素直に使う？

Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

コードで補う

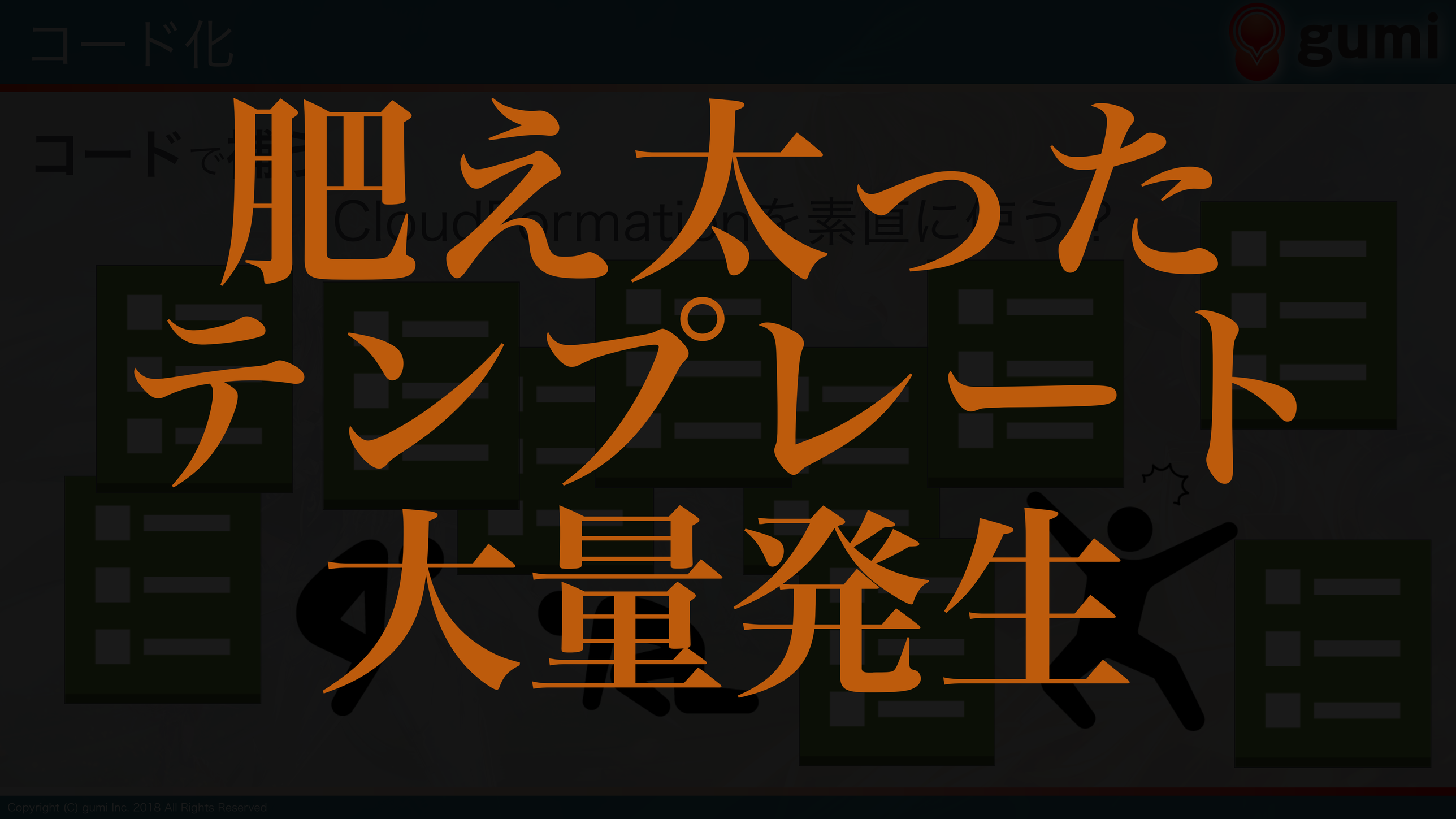
CloudFormationを素直に使う？



Copyright (C) gumi Inc. 2018 All Rights Reserved

gumi

肥え太ったテンプレート大量発生



Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

コードで補う

~~CloudFormationを素直に使う？~~

Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

コードで補う

~~CloudFormationを素直に使う？~~

講義者ノート

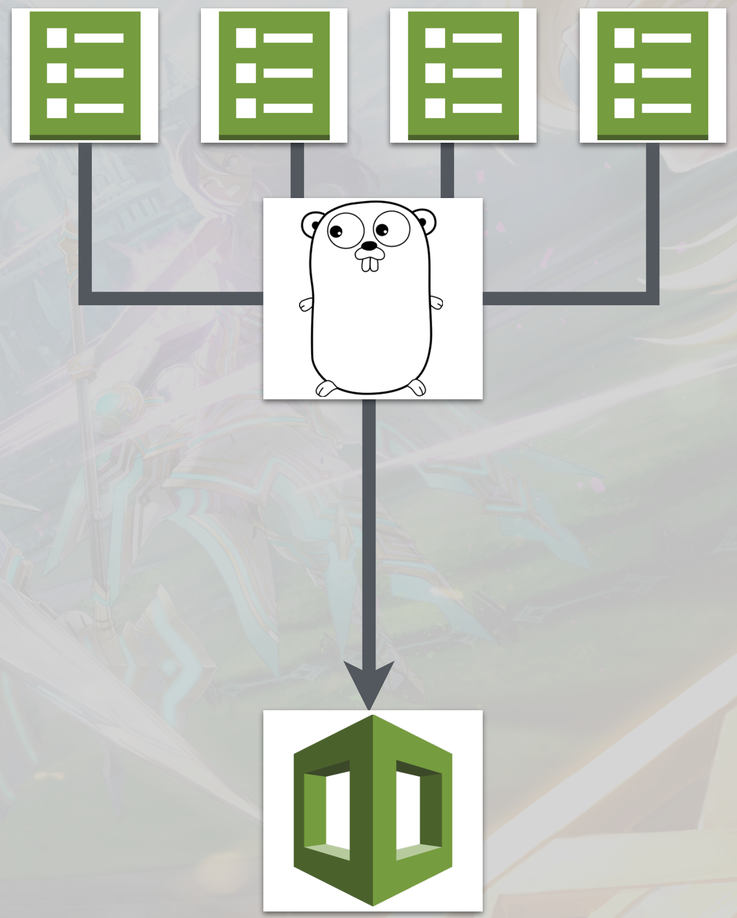
環境の数が多く，一つ一つのテンプレートファイルの行数も多くなる．これでは管理コストが上がってしまう．

Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツール(CloudFormation)を**コード**で**補う**

~~CloudFormationを素直に使う？~~

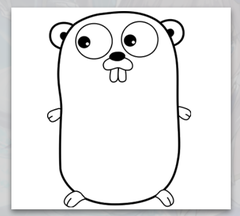


Copyright (C) gumi Inc. 2018 All Rights Reserved

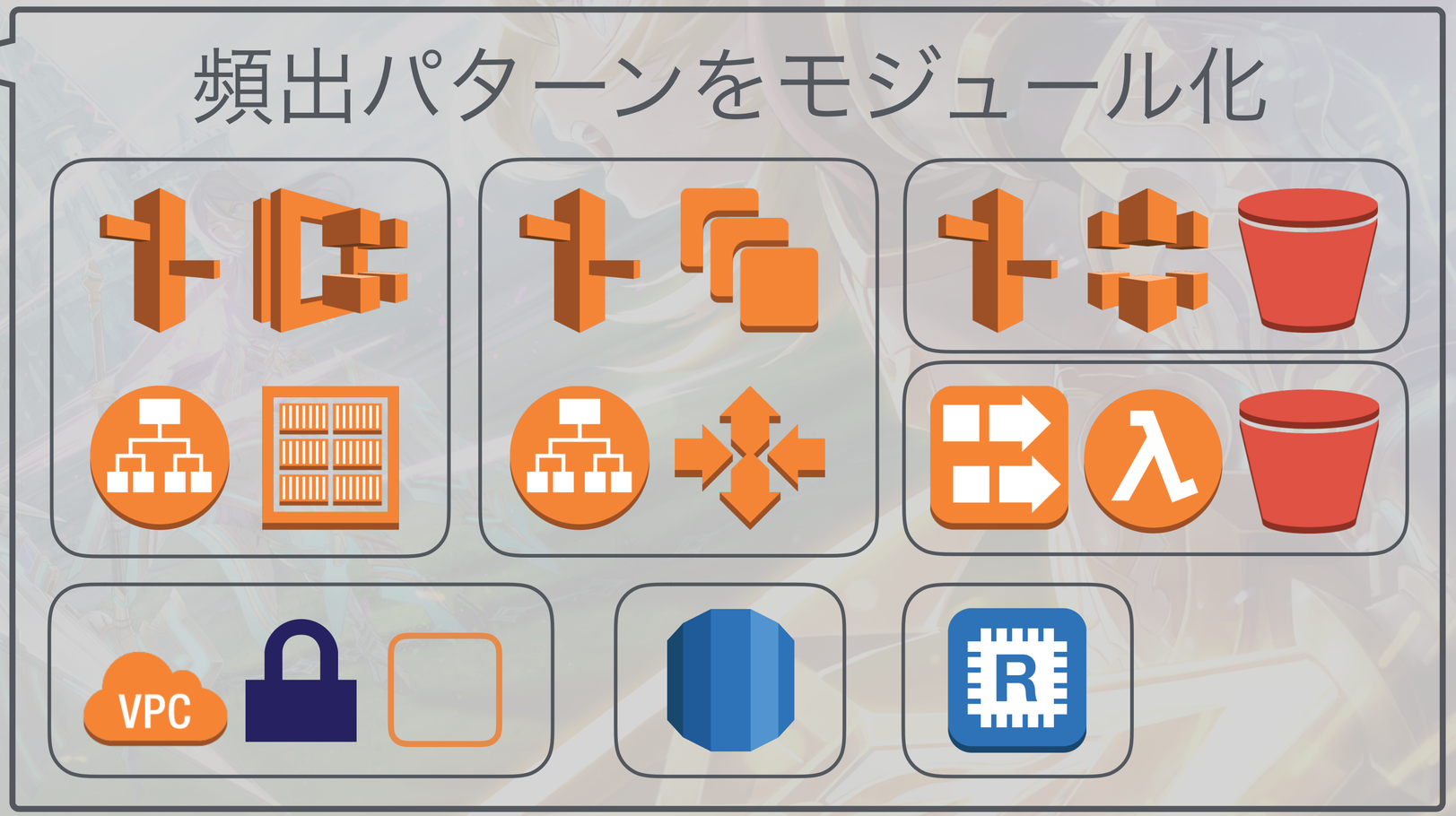
コード化

ツール(CloudFormation)を**コード**で**補う**







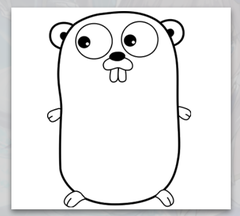


Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツール(CloudFormation)を**コード**で**補う**







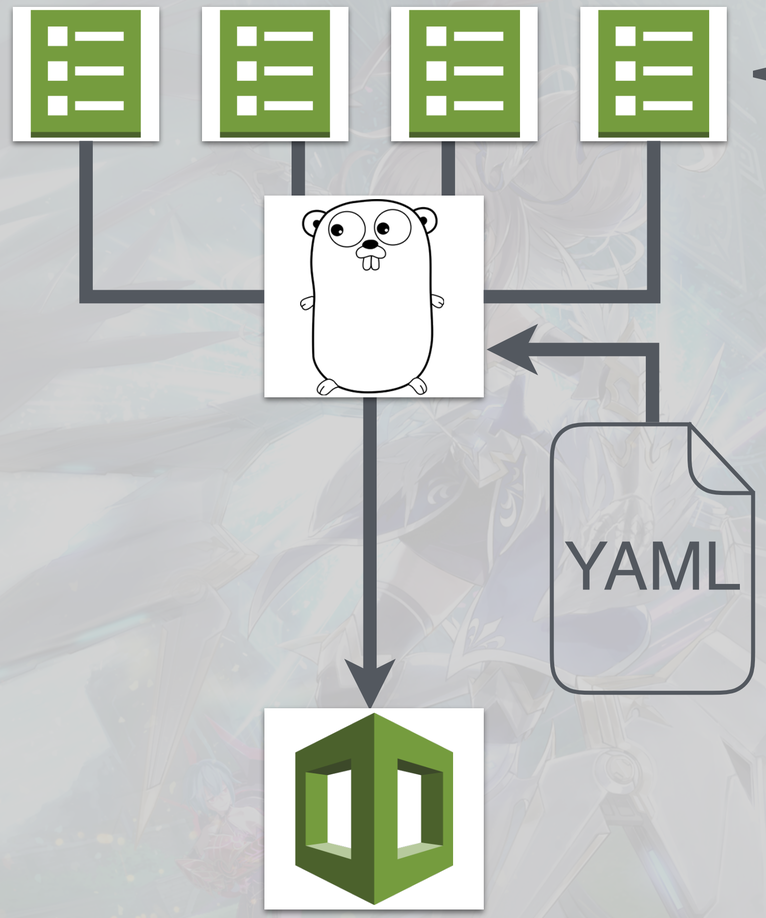
講義者ノート

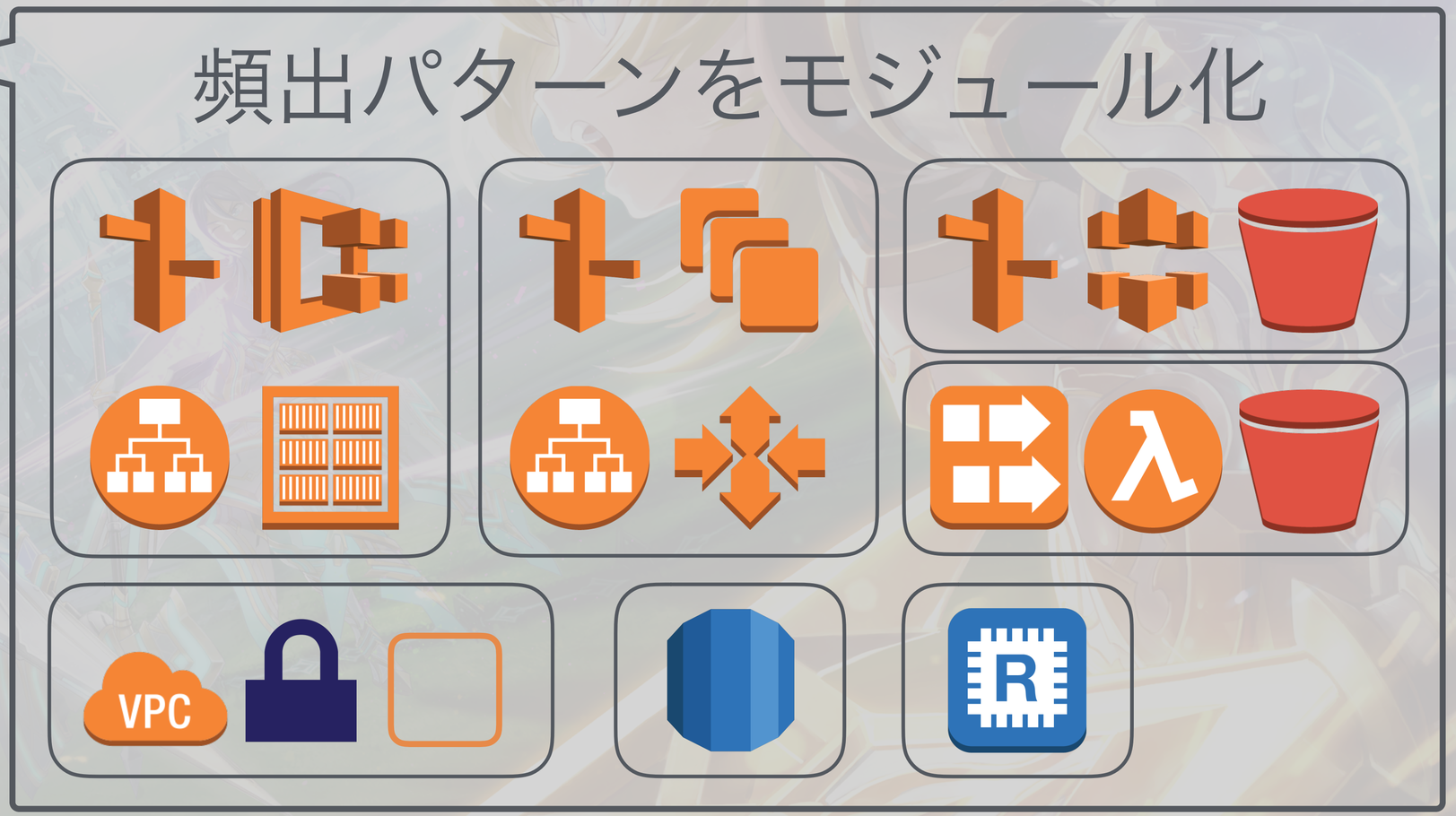
ゲーム案件でよくある頻出パターンごとに, CloudFormation テンプレートの部分パーツを作る. 分割の粒度や方針は, AWS サービスアイコンで何となく想像して欲しい.

Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツール(CloudFormation)を**コード**で**補う**





Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツール(CloudFormation)を**コード**で**補う**



講義者ノート

環境毎に小さな YAML を記述する. これを自社開発したツールに渡すと, ツールは事前に準備してある小さなテンプレートパーツを, YAML に従って順番に CloudFormation に適用していく.

Copyright (C) gumi Inc. 2018 All Rights Reserved

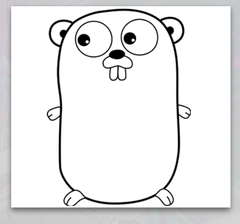
コード化

ツール(CloudFormation)を**コード**で**補う**

```
resources:
  - name: default
    type: network

  - name: player-db
    type: aurora

  - name: game-server
    type: application
```



Copyright (C) gumi Inc. 2018 All Rights Reserved

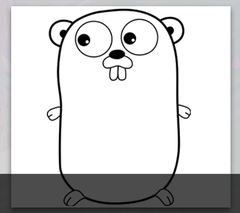
コード化

ツール(CloudFormation)を**コード**で**補う**

```
resources:
  - name: default
    type: network

  - name: player-db
    type: aurora

  - name: game-server
    type: application
```



講義者ノート

環境毎に用意された YAML は, このような記載となる. ネットワーク, DB サーバ, Web サーバの存在が読み取れる.

Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

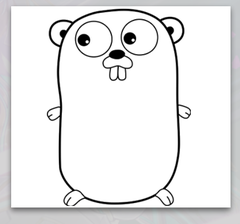
ツール(CloudFormation)を**コード**で**補う**

```
resources:
  - name: default
    type: network

  - name: player-db
    type: aurora

  - name: game-server
    type: application
```

create
feature1



Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

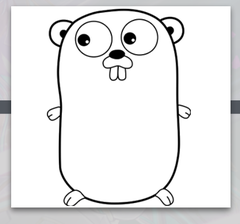
ツール(CloudFormation)を**コード**で**補う**

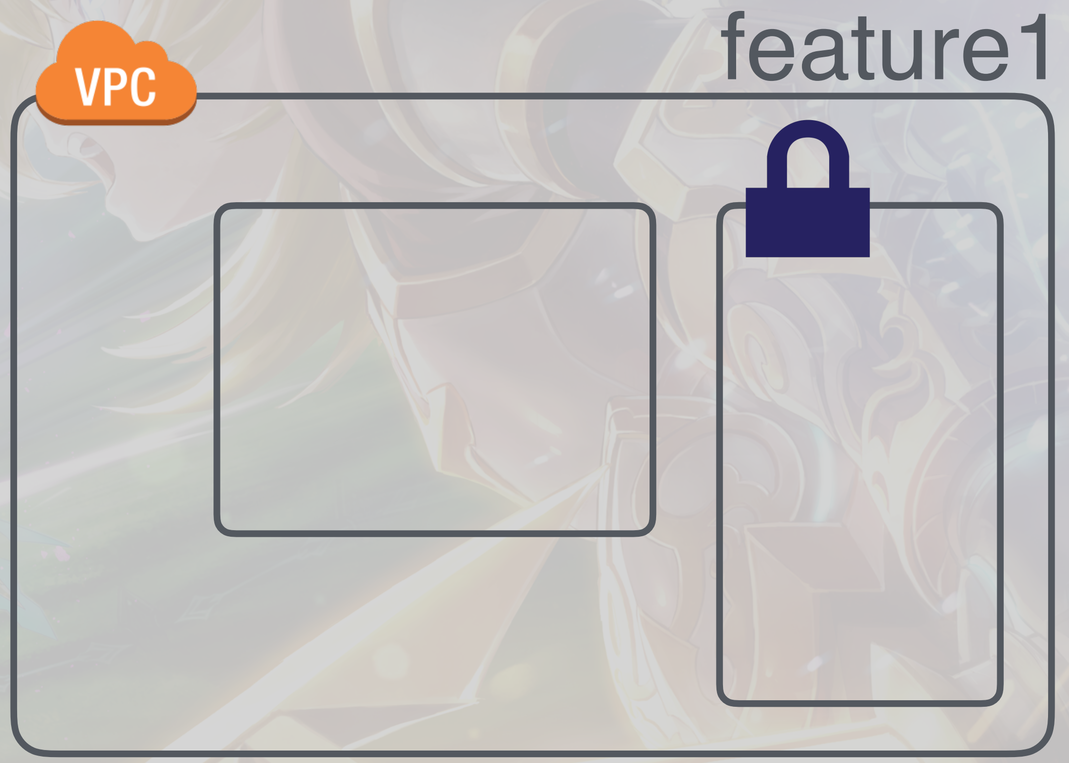
```
resources:
  - name: default
    type: network

  - name: player-db
    type: aurora

  - name: game-server
    type: application
```

create
feature1





Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

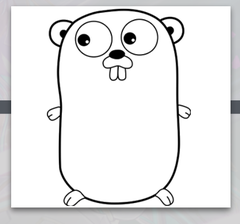
ツール(CloudFormation)を**コード**で**補う**

```
resources:
  - name: default
    type: network

  - name: player-db
    type: aurora

  - name: game-server
    type: application
```

create
feature1





Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

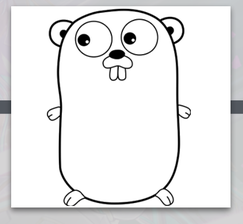
ツール(CloudFormation)を**コード**で**補う**

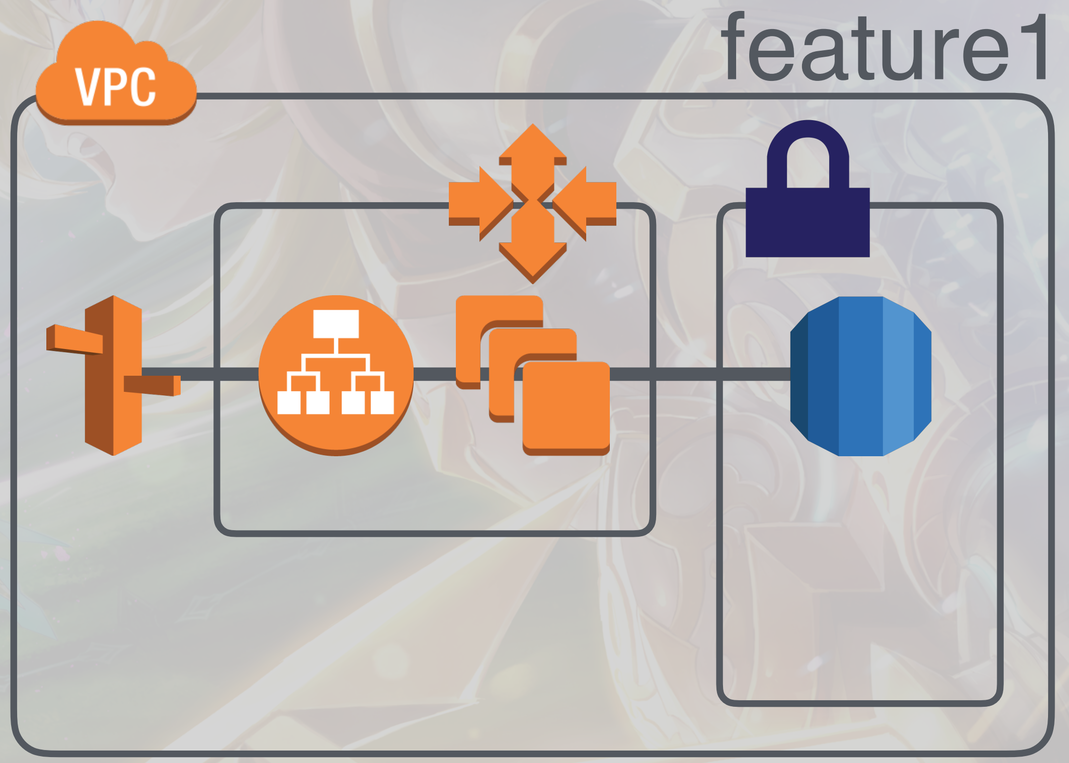
```
resources:
  - name: default
    type: network

  - name: player-db
    type: aurora

  - name: game-server
    type: application
```

create
feature1





Copyright (C) gumi Inc. 2018 All Rights Reserved

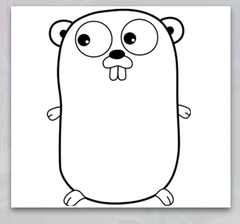
コード化

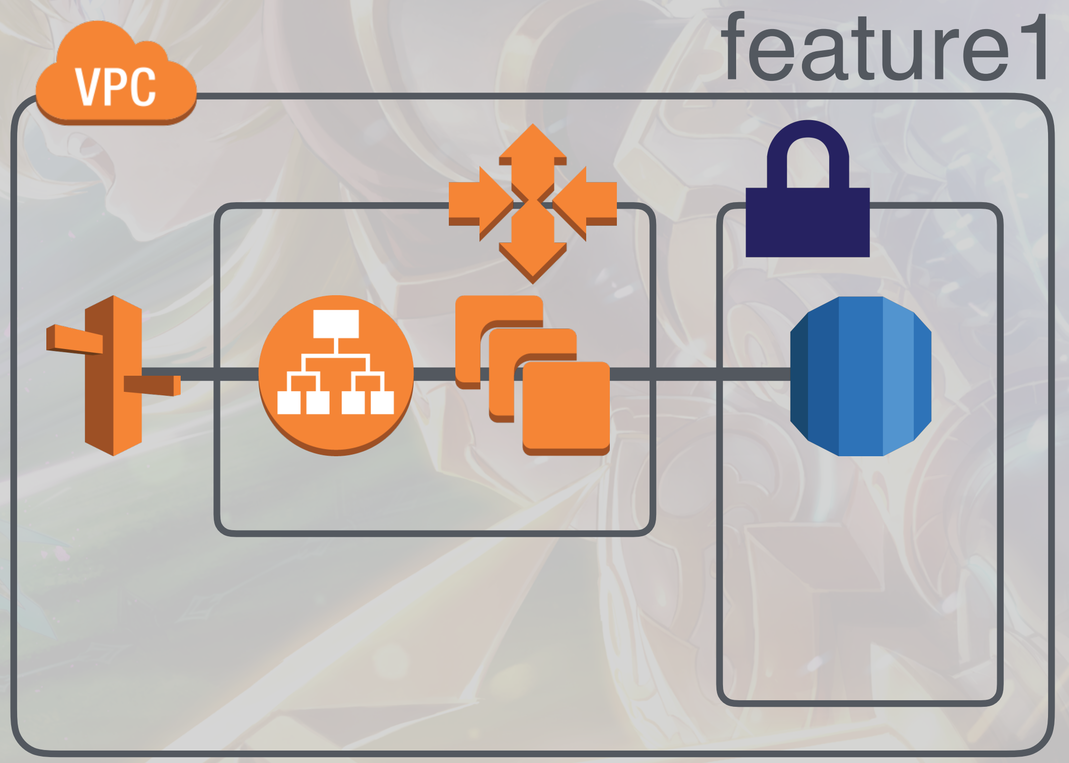
ツール(CloudFormation)を**コード**で**補う**

```
resources:
  - name: default
    type: network

  - name: player-db
    type: aurora

  - name: game-server
    type: application
```





Copyright (C) gumi Inc. 2018 All Rights Reserved

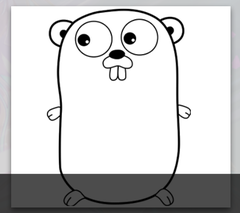
コード化

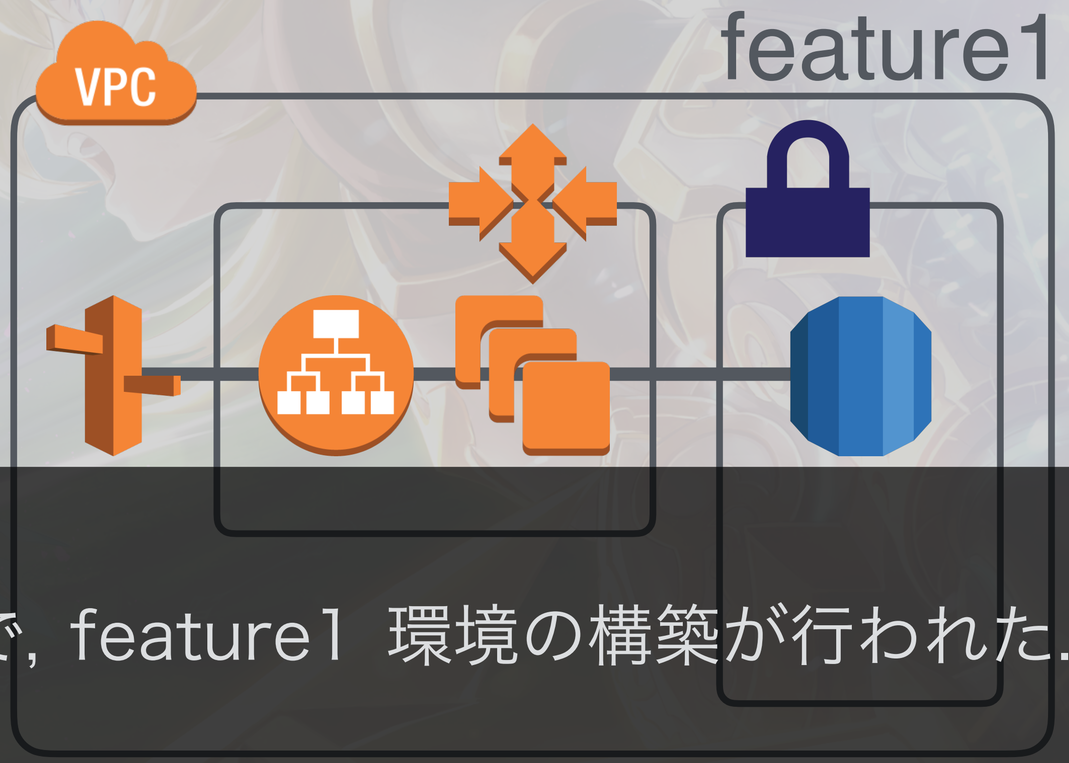
ツール(CloudFormation)を**コード**で**補う**

```
resources:
  - name: default
    type: network

  - name: player-db
    type: aurora

  - name: game-server
    type: application
```





講義者ノート

YAML に記載された内容を順に適用することで, feature1 環境の構築が行われた.

次に, この環境にキャッシュを追加する.

Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

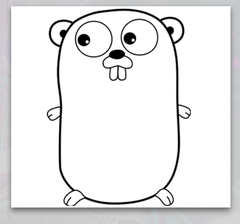
ツール(CloudFormation)を**コード**で**補う**

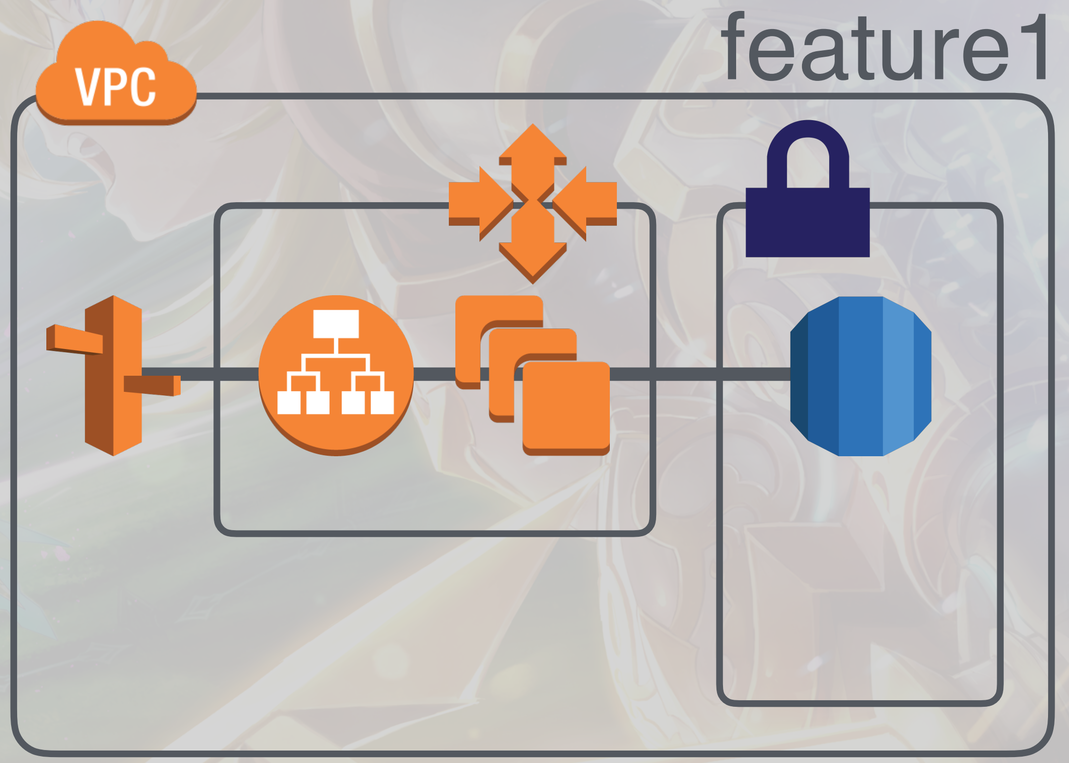
```
resources:
  - name: default
    type: network

  - name: player-db
    type: aurora

  - name: player-cache
    type: elasticache

  - name: game-server
    type: application
```





Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツール(CloudFormation)を**コード**で**補う**

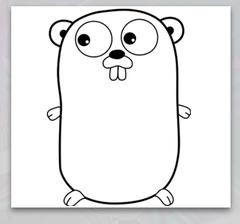
```
resources:
  - name: default
    type: network

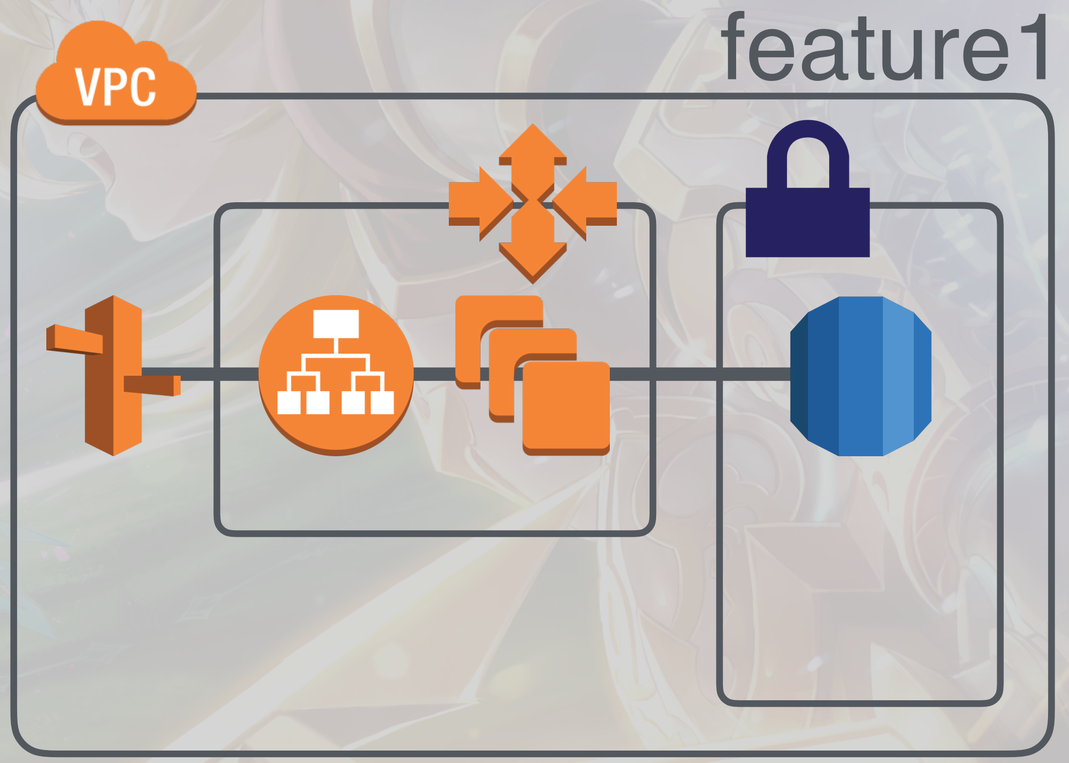
  - name: player-db
    type: aurora

  - name: player-cache
    type: elasticache

  - name: game-server
    type: application
```

update
feature1





Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツール(CloudFormation)を**コード**で**補う**

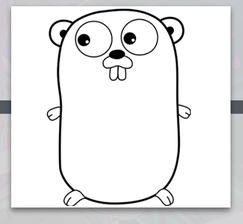
```
resources:
  - name: default
    type: network

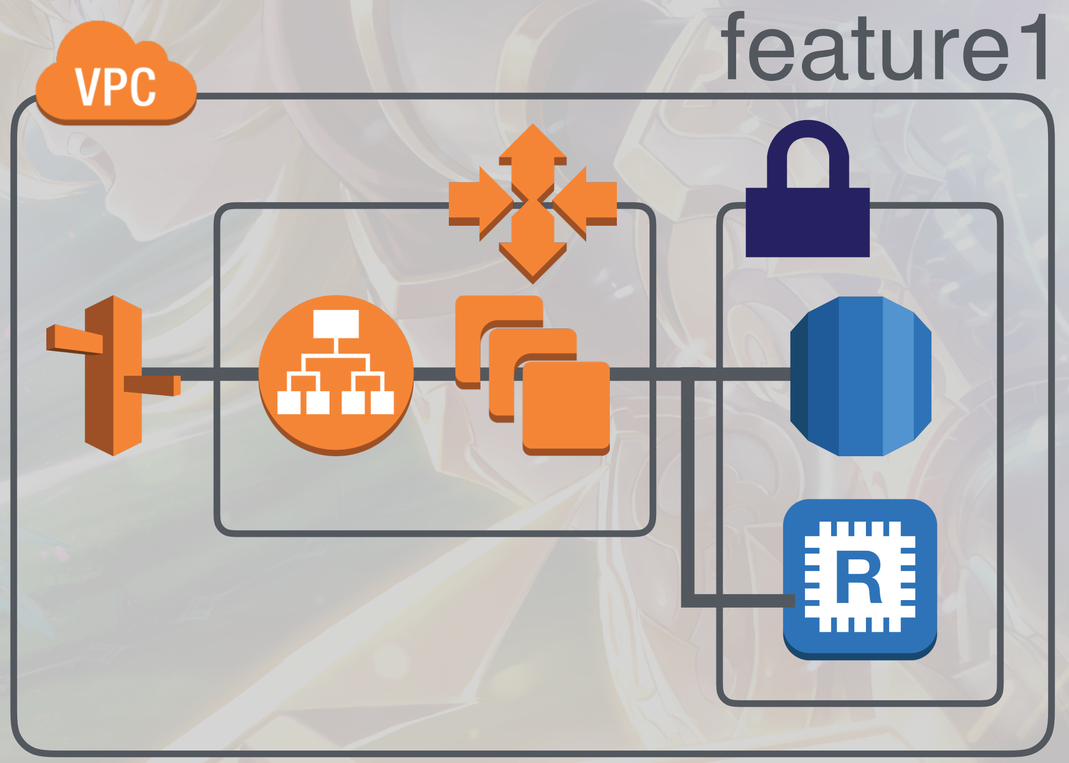
  - name: player-db
    type: aurora

  - name: player-cache
    type: elasticache

  - name: game-server
    type: application
```

update
feature1





Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

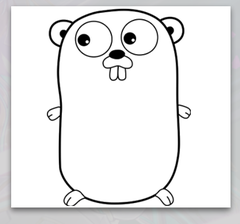
ツール(CloudFormation)を**コード**で**補う**

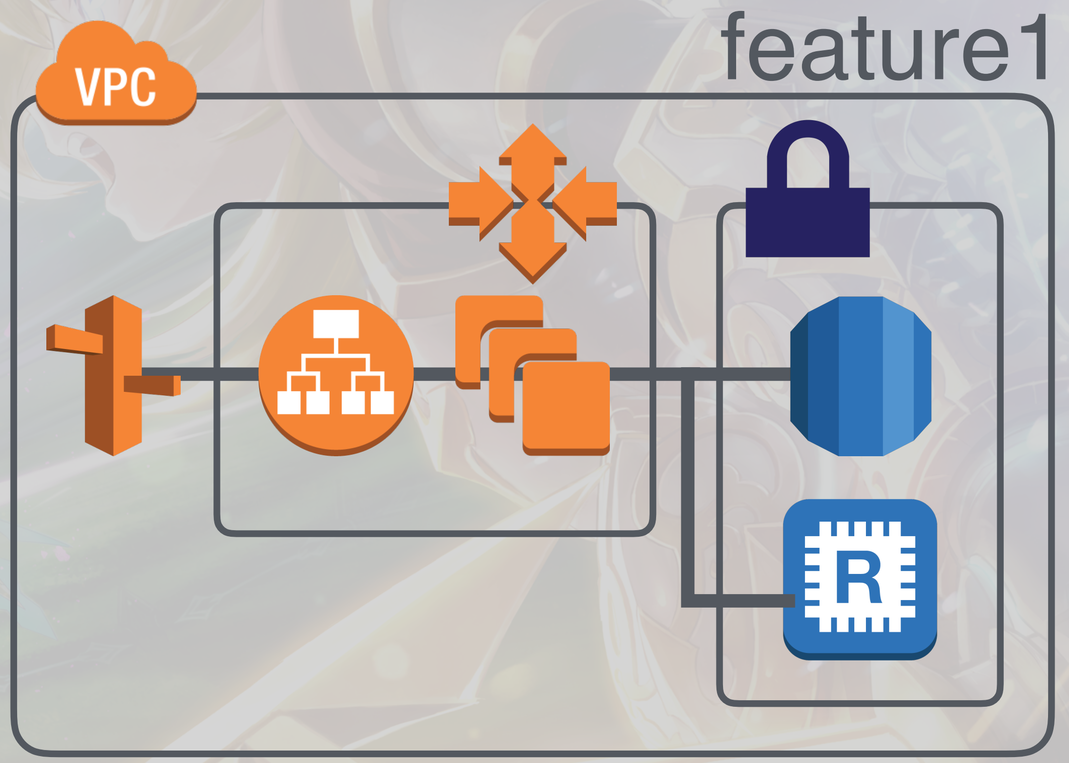
```
resources:
  - name: default
    type: network

  - name: player-db
    type: aurora

  - name: player-cache
    type: elasticache

  - name: game-server
    type: application
```





Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

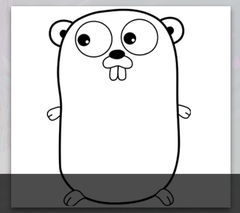
ツール(CloudFormation)を**コード**で**補う**

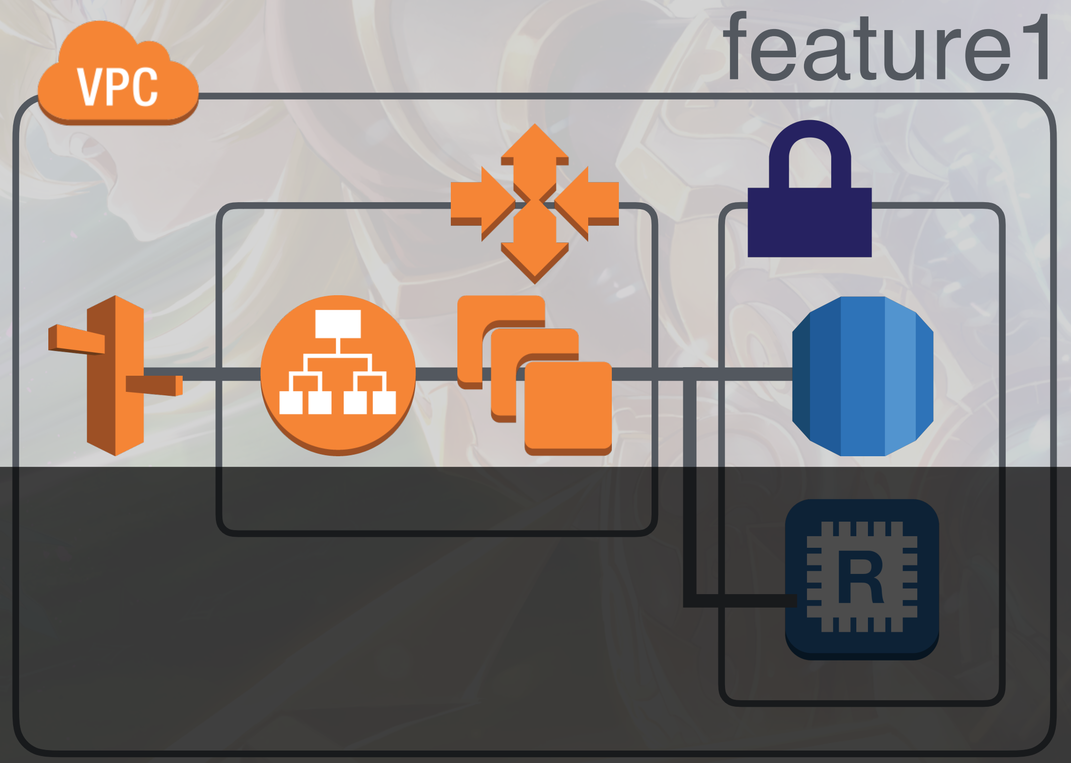
```
resources:
  - name: default
    type: network

  - name: player-db
    type: aurora

  - name: player-cache
    type: elasticache

  - name: game-server
    type: application
```





講義者ノート

次に, インスタンスサイズを上げる.

Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

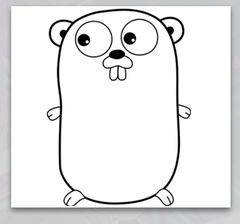
ツール(CloudFormation)を**コード**で**補う**

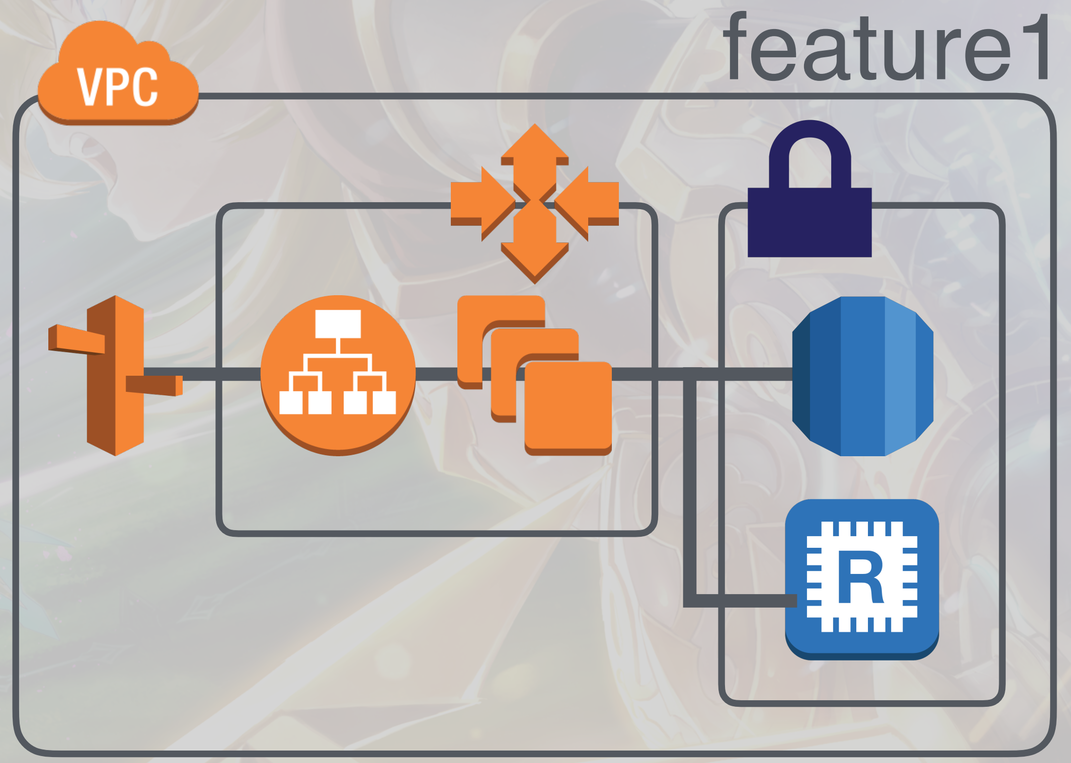
```
resources:
  - name: default
    type: network

  - name: player-db
    type: aurora

  - name: player-cache
    type: elasticache

  - name: game-server
    type: application
    properties:
      - name: InstanceType
        type: m4.16xlarge
```





Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツール(CloudFormation)を**コード**で**補う**

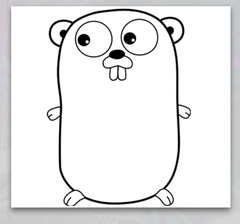
```
resources:
  - name: default
    type: network

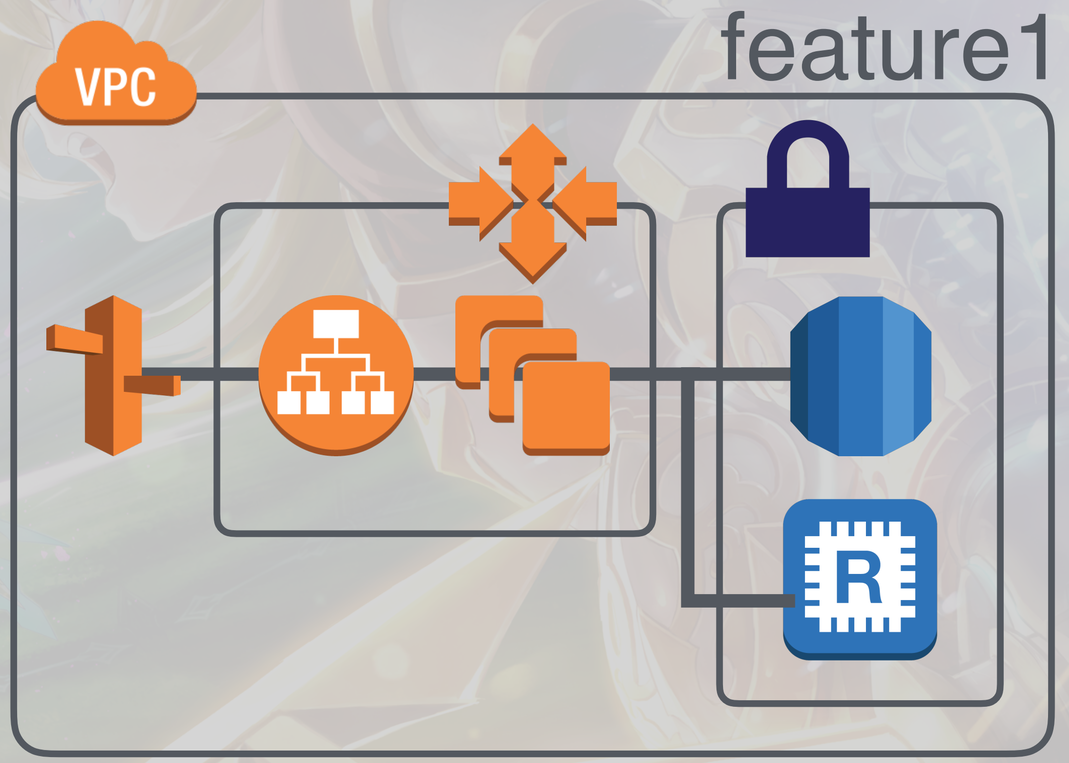
  - name: player-db
    type: aurora

  - name: player-cache
    type: elasticache

  - name: game-server
    type: application
    properties:
      - name: InstanceType
        type: m4.16xlarge
```

update
feature1





Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツール(CloudFormation)を**コード**で**補う**

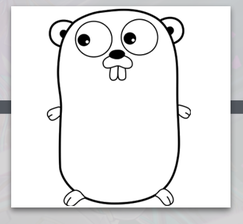
```
resources:
  - name: default
    type: network

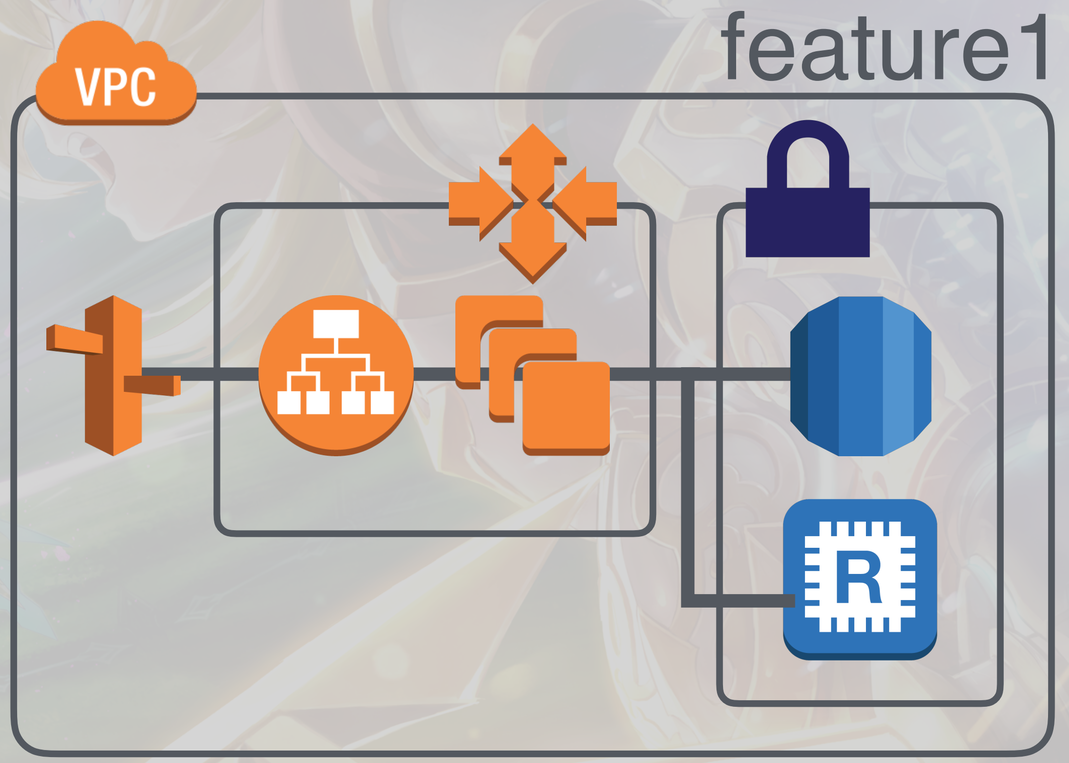
  - name: player-db
    type: aurora

  - name: player-cache
    type: elasticache

  - name: game-server
    type: application
    properties:
      - name: InstanceType
        type: m4.16xlarge
```

update feature1





Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツール(CloudFormation)を**コード**で**補う**

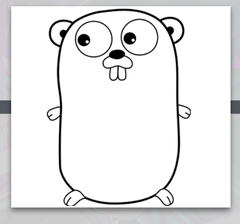
```
resources:
  - name: default
    type: network

  - name: player-db
    type: aurora

  - name: player-cache
    type: elasticache

  - name: game-server
    type: application
    properties:
      - name: InstanceType
        type: m4.16xlarge
```

update
feature1





Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

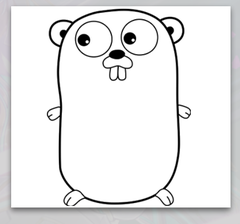
ツール(CloudFormation)を**コード**で**補う**

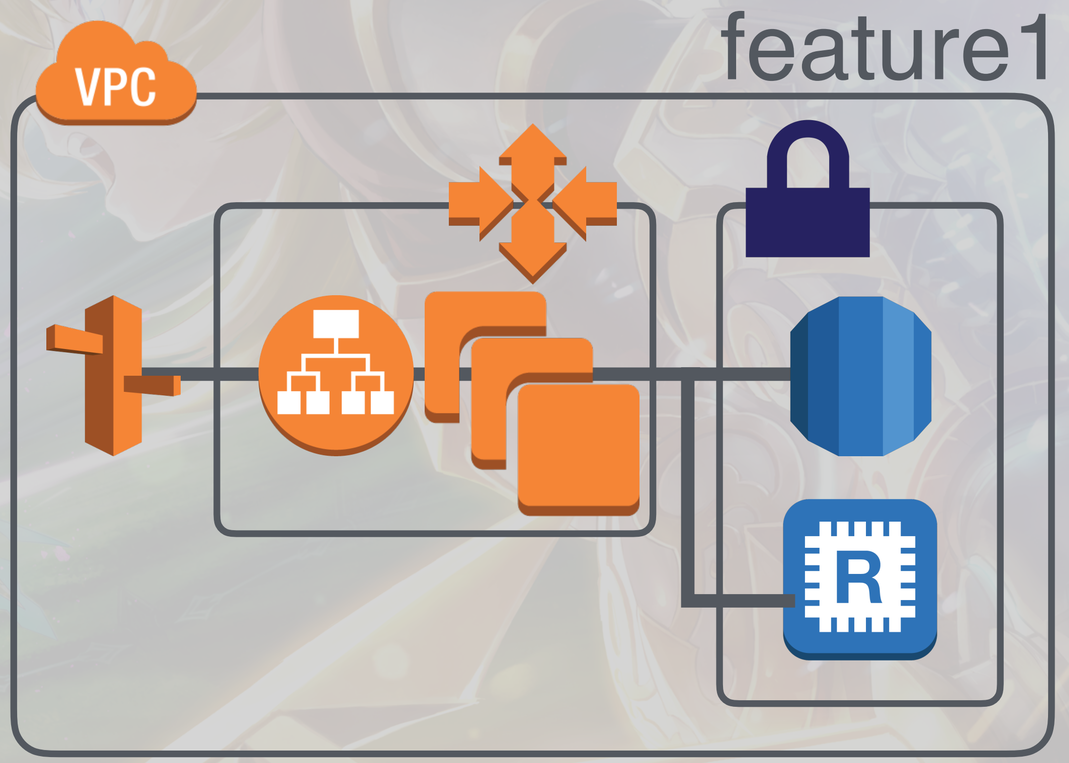
```
resources:
  - name: default
    type: network

  - name: player-db
    type: aurora

  - name: player-cache
    type: elasticache

  - name: game-server
    type: application
    properties:
      - name: InstanceType
        type: m4.16xlarge
```





Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

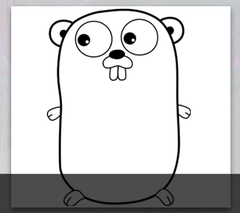
ツール(CloudFormation)を**コード**で**補う**

```
resources:
  - name: default
    type: network

  - name: player-db
    type: aurora

  - name: player-cache
    type: elasticache

  - name: game-server
    type: application
    properties:
      - name: InstanceType
        type: m4.16xlarge
```





講義者ノート

次に, インスタンスサイズを元に戻し, コンテナを追加する.

Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツール(CloudFormation)を**コード**で**補う**

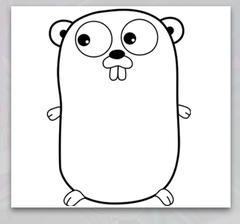
```
resources:
  - name: default
    type: network

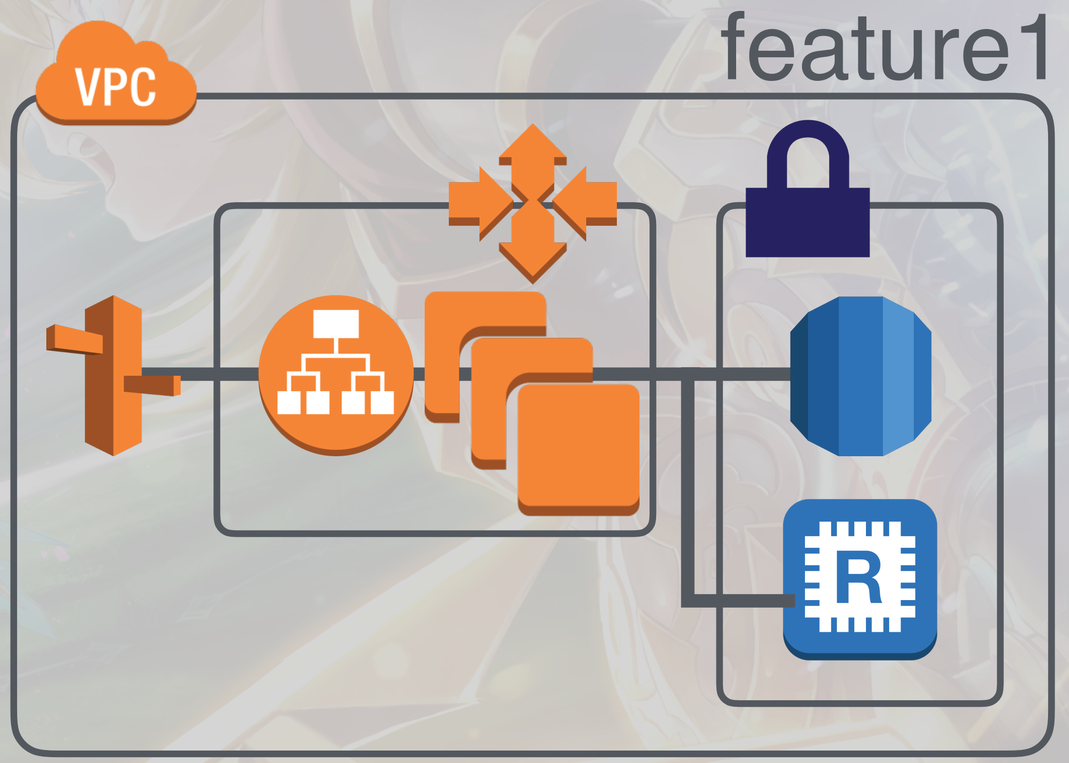
  - name: player-db
    type: aurora

  - name: player-cache
    type: elasticache

  - name: game-server
    type: application

  - name: game-container
    type: container
```





Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツール(CloudFormation)を**コード**で**補う**

```
resources:
  - name: default
    type: network

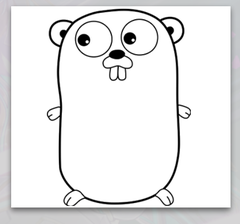
  - name: player-db
    type: aurora

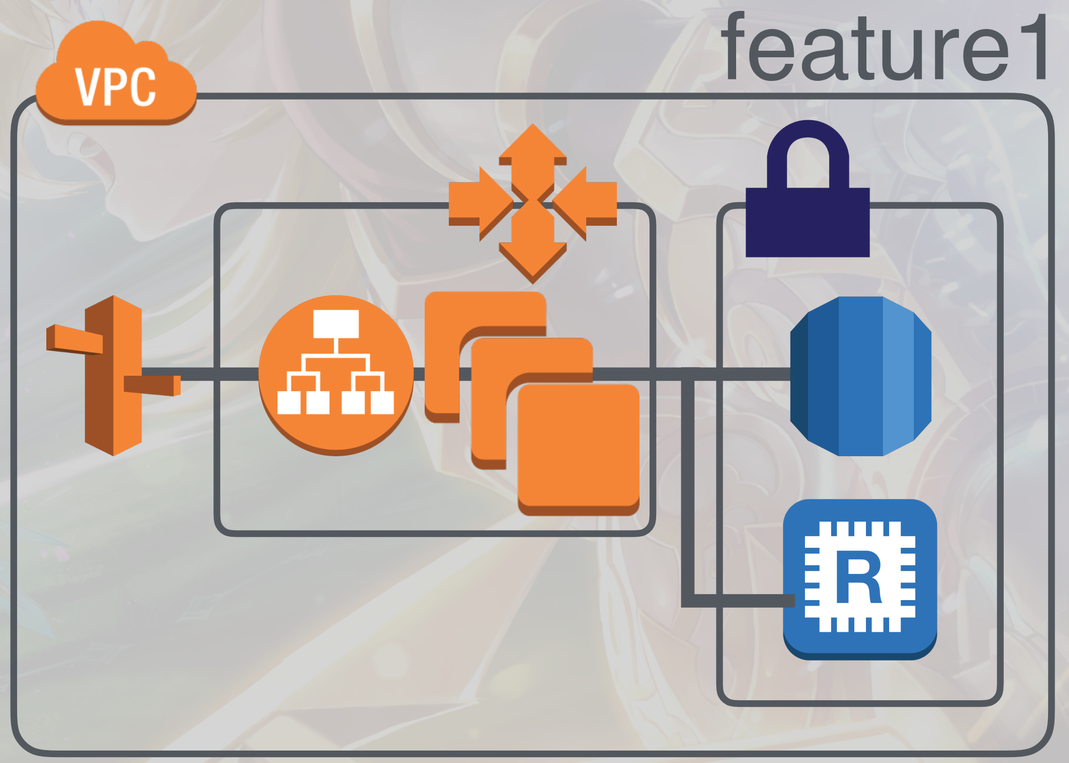
  - name: player-cache
    type: elasticache

  - name: game-server
    type: application

  - name: game-container
    type: container
```

update
feature1





Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツール(CloudFormation)を**コード**で**補う**

```
resources:
  - name: default
    type: network

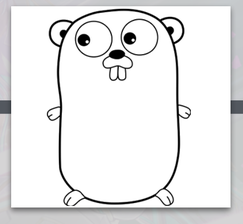
  - name: player-db
    type: aurora

  - name: player-cache
    type: elasticache

  - name: game-server
    type: application

  - name: game-container
    type: container
```

update
feature1





Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツール(CloudFormation)を**コード**で**補う**

```
resources:
  - name: default
    type: network

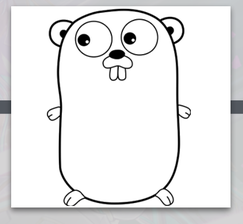
  - name: player-db
    type: aurora

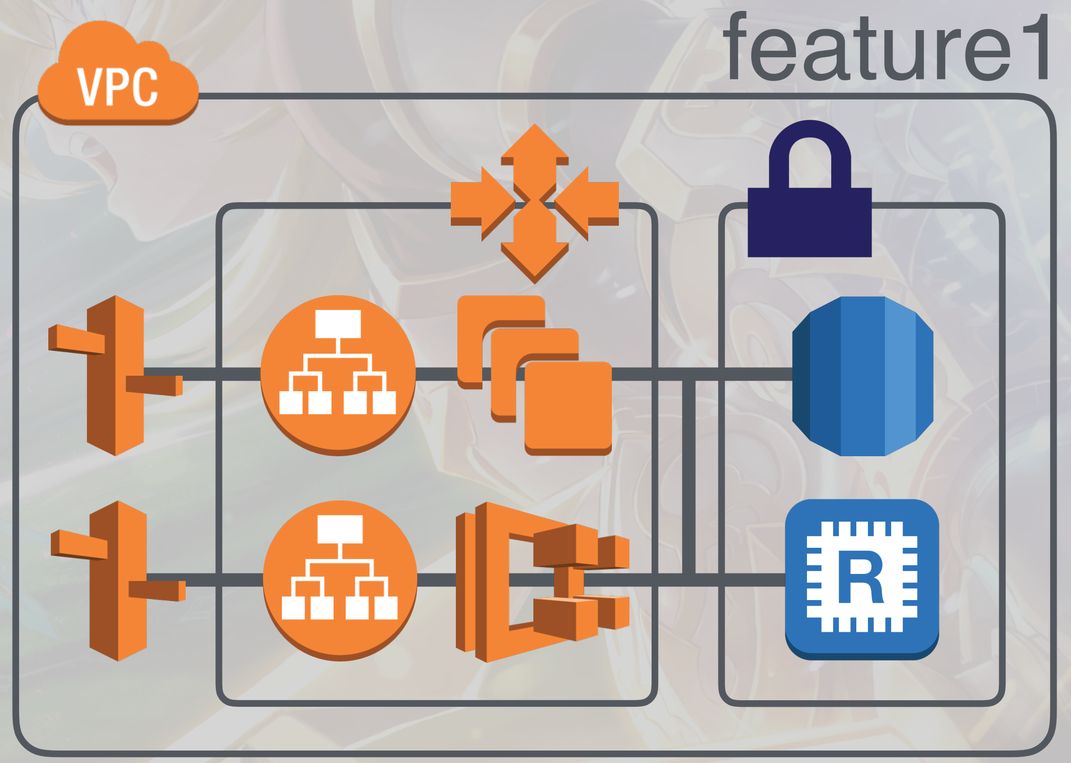
  - name: player-cache
    type: elasticache

  - name: game-server
    type: application

  - name: game-container
    type: container
```

update feature1





Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツール(CloudFormation)を**コード**で**補う**

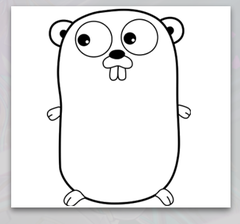
```
resources:
  - name: default
    type: network

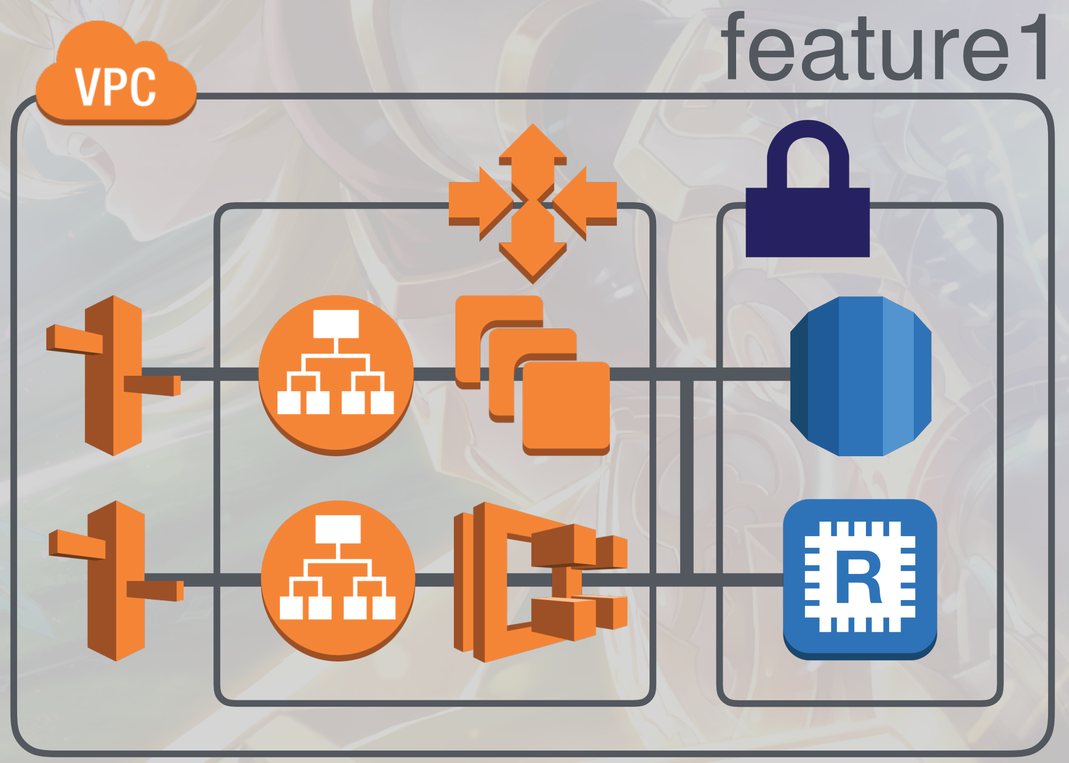
  - name: player-db
    type: aurora

  - name: player-cache
    type: elasticache

  - name: game-server
    type: application

  - name: game-container
    type: container
```





Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツール(CloudFormation)を**コード**で**補う**

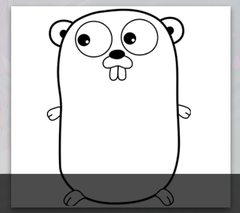
```
resources:
  - name: default
    type: network

  - name: player-db
    type: aurora

  - name: player-cache
    type: elasticache

  - name: game-server
    type: application

  - name: game-container
    type: container
```





講義者ノート

並行運用し, 動作確認が終わったのち, EC2 側を削除する.

Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツール(CloudFormation)を**コード**で**補う**

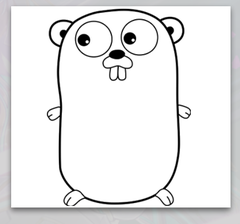
```
resources:
  - name: default
    type: network

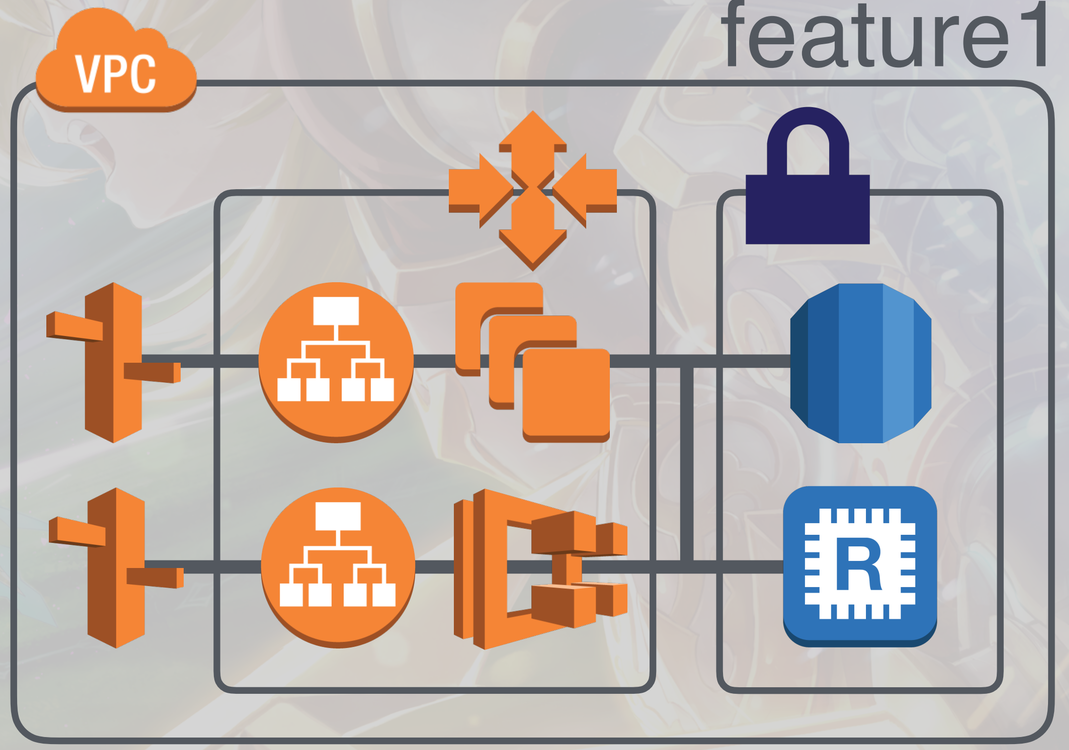
  - name: player-db
    type: aurora

  - name: player-cache
    type: elasticache



  - name: game-container
    type: container
```





Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツール(CloudFormation)を**コード**で**補う**

```
resources:
  - name: default
    type: network

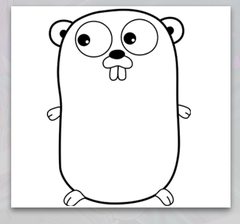
  - name: player-db
    type: aurora

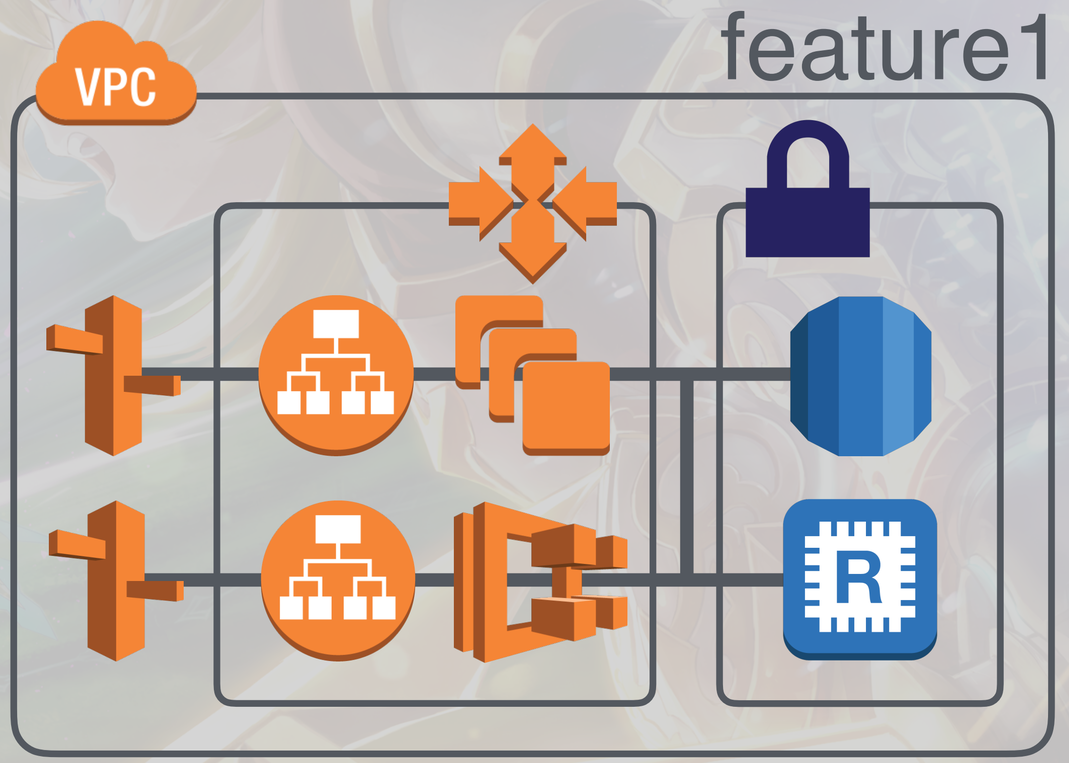
  - name: player-cache
    type: elasticache



  - name: game-container
    type: container
```

update
feature1





Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツール(CloudFormation)を**コード**で**補う**

```
resources:
  - name: default
    type: network

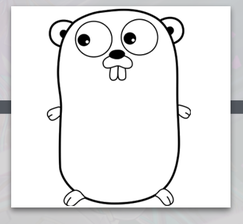
  - name: player-db
    type: aurora

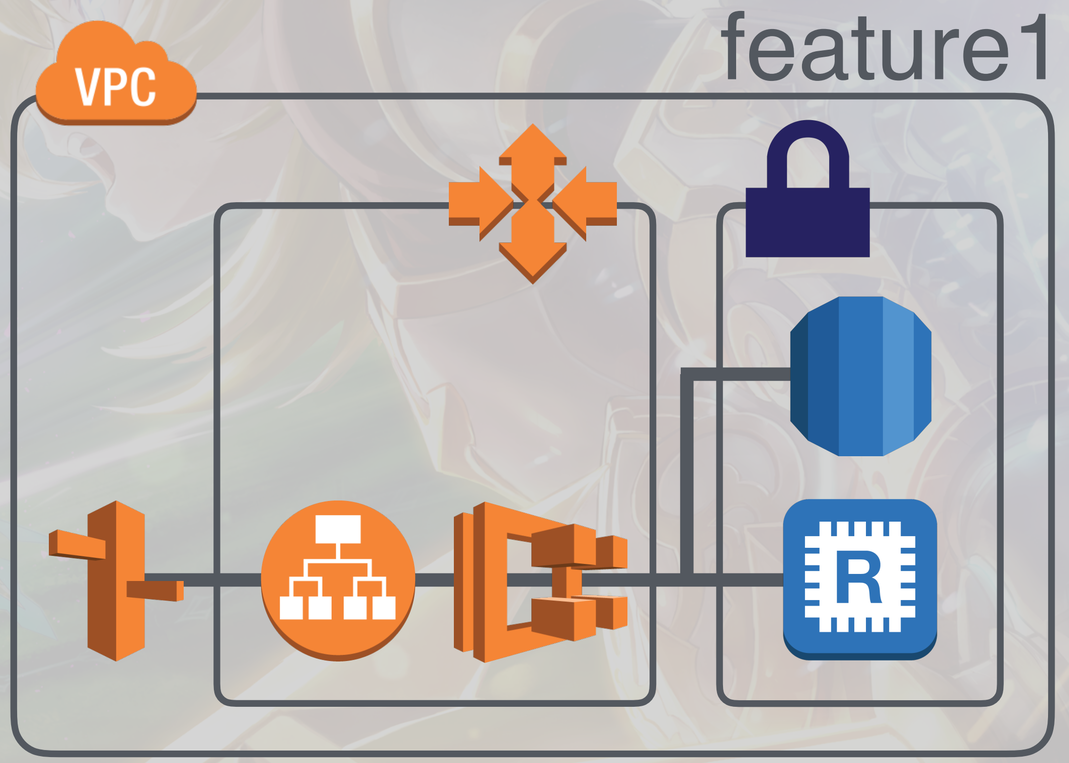
  - name: player-cache
    type: elasticache



  - name: game-container
    type: container
```

update
feature1





Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツール(CloudFormation)を**コード**で**補う**

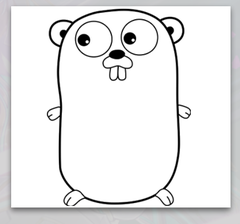
```
resources:
  - name: default
    type: network

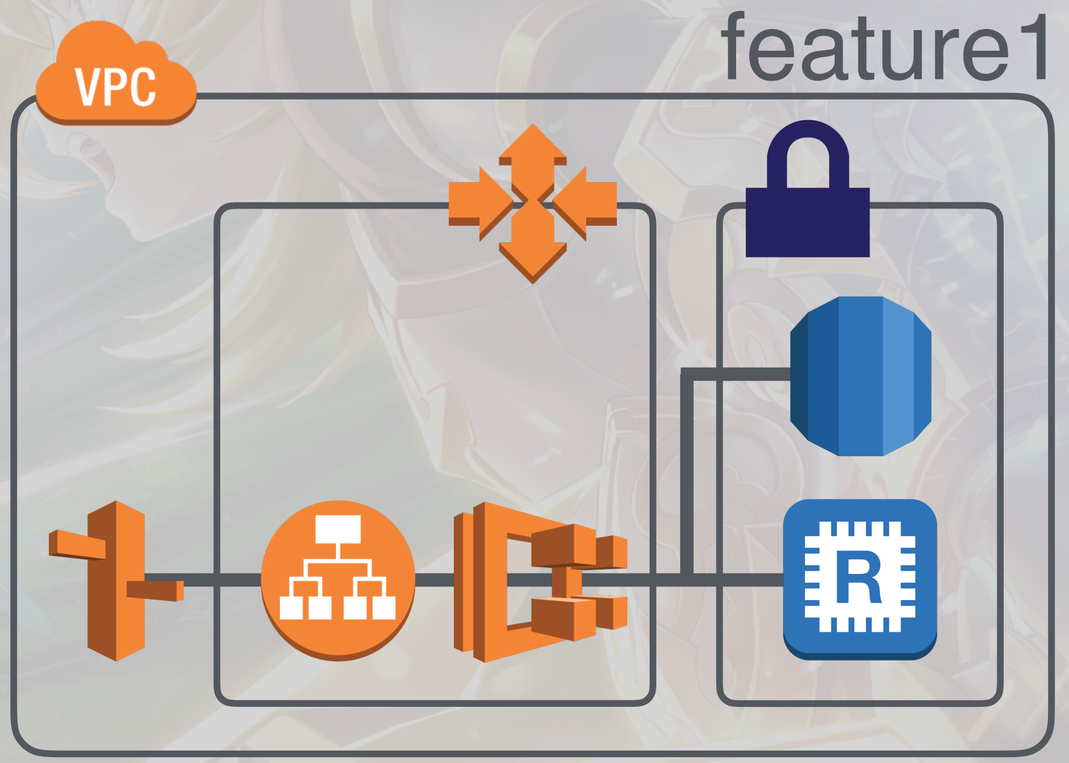
  - name: player-db
    type: aurora

  - name: player-cache
    type: elasticache



  - name: game-container
    type: container
```





Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

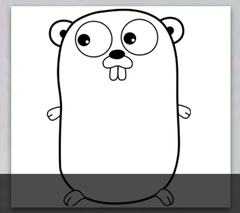
ツール(CloudFormation)を**コード**で**補う**

```
resources:
  - name: default
    type: network

  - name: player-db
    type: aurora

  - name: player-cache
    type: elasticache

  - name: game-container
    type: container
```





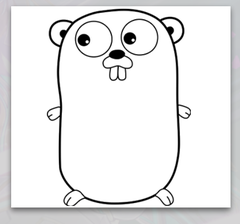
講義者ノート

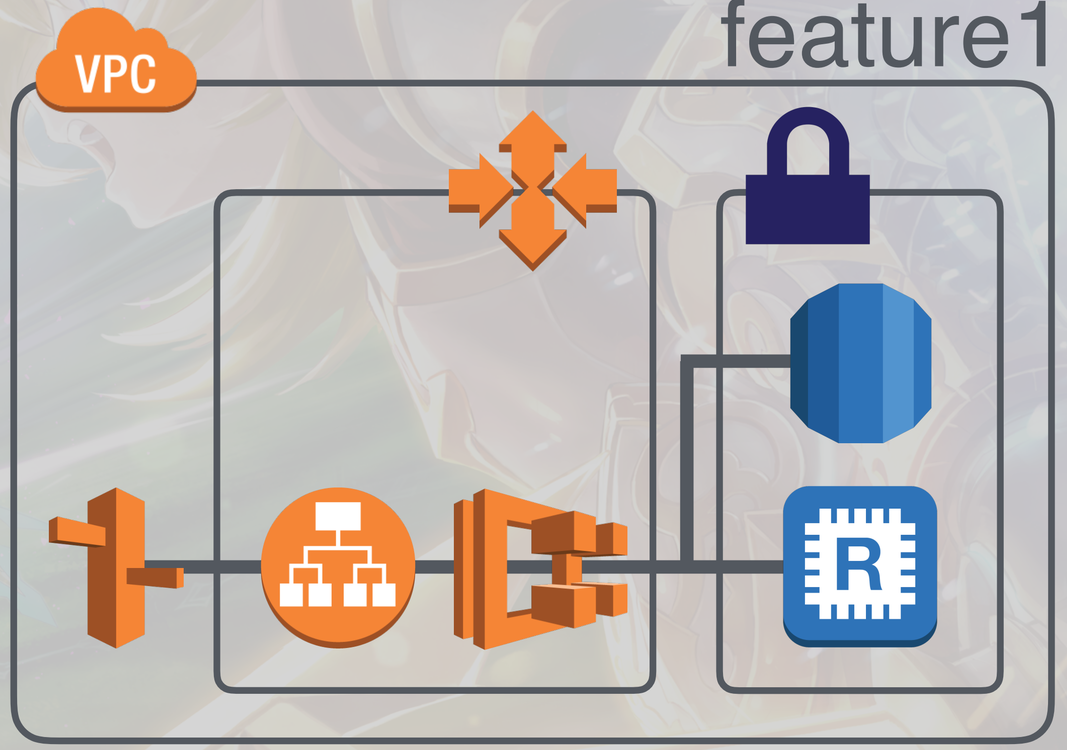
最後に, feature1 環境全てを削除する.

Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツール(CloudFormation)を**コード**で**補う**

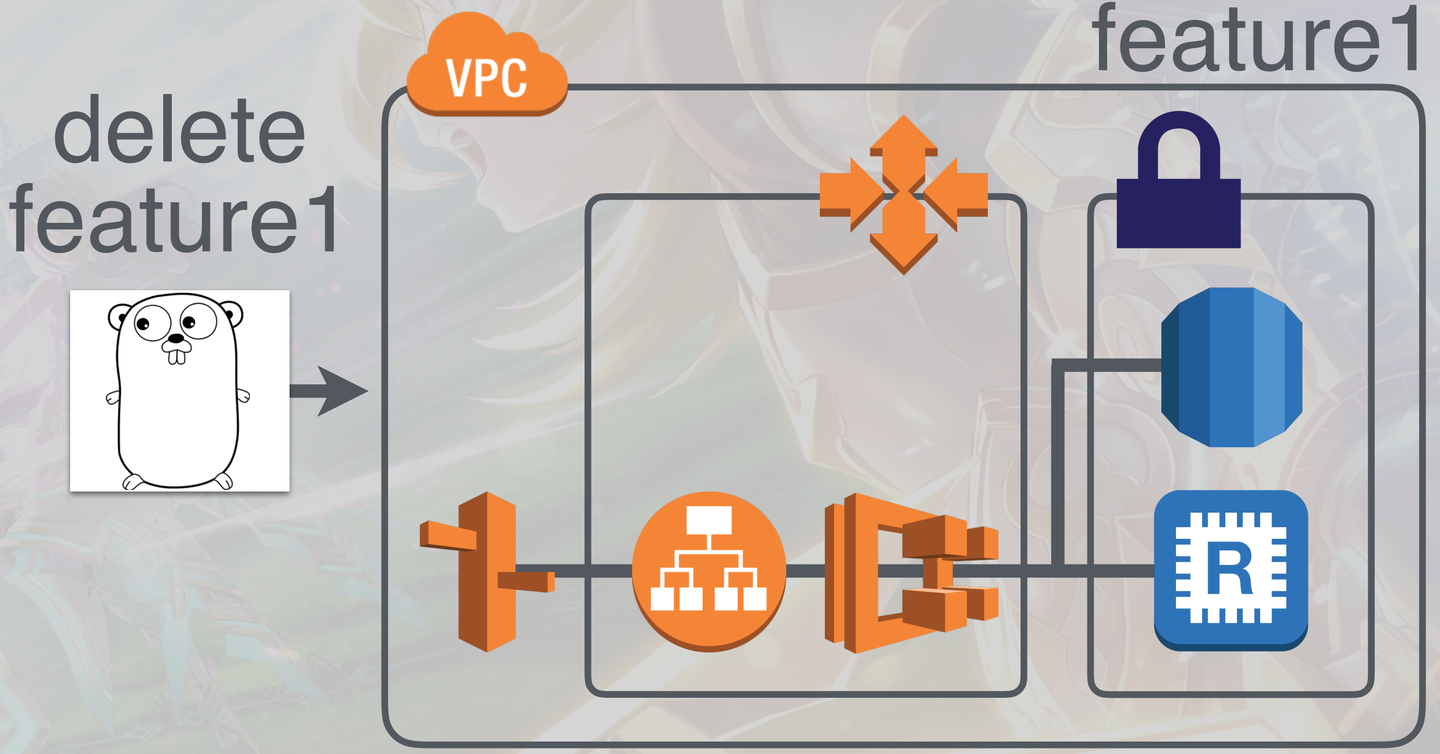




Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

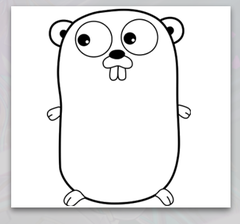
ツール(CloudFormation)を**コード**で**補う**



Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツール(CloudFormation)を**コード**で**補う**



Copyright (C) gumi Inc. 2018 All Rights Reserved

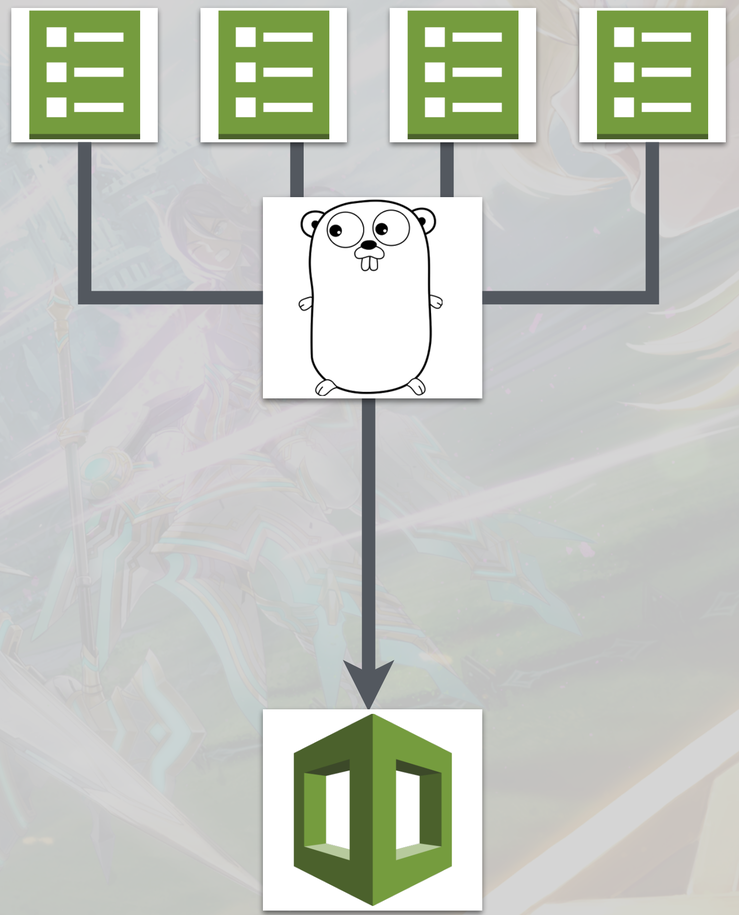
コード化

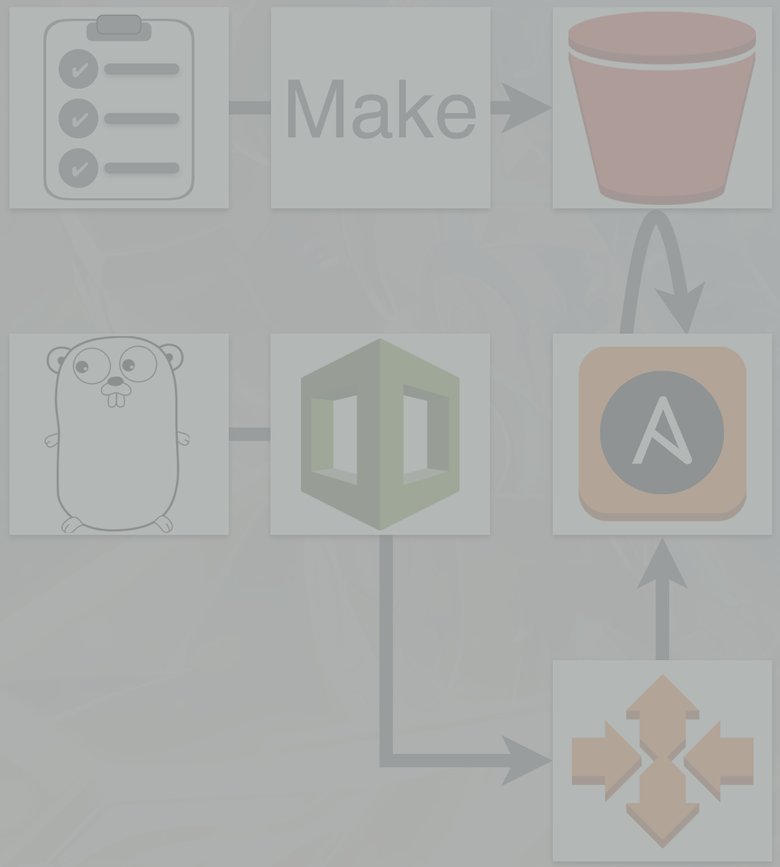
ツールを選択





コードで補う





Copyright (C) gumi Inc. 2018 All Rights Reserved

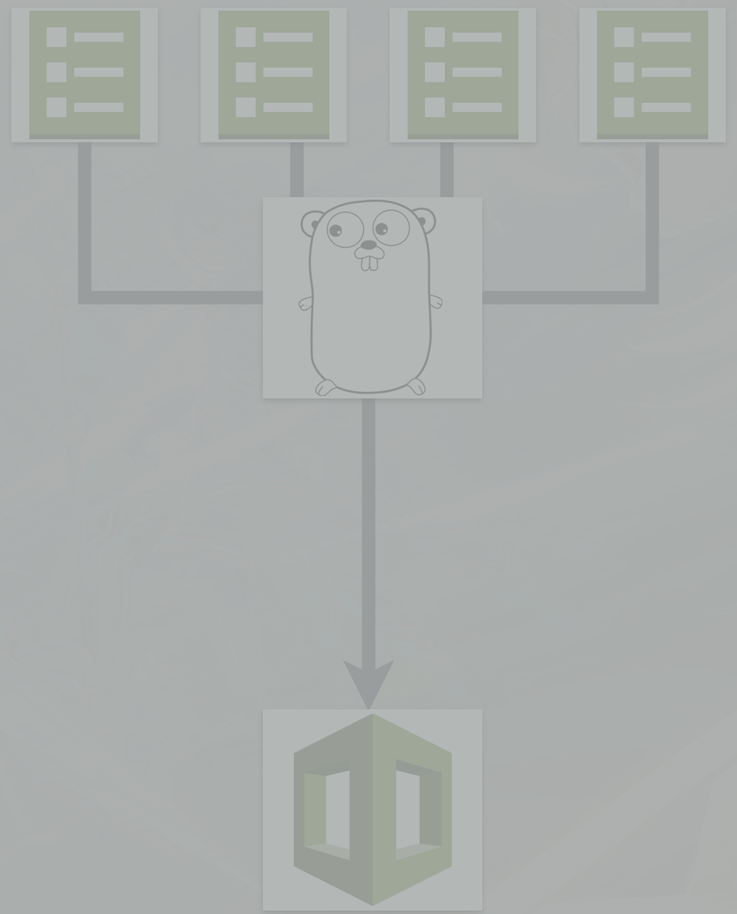
コード化

ツールを選択





コードで補う



コードで接着



Copyright (C) gumi Inc. 2018 All Rights Reserved

コードで接着

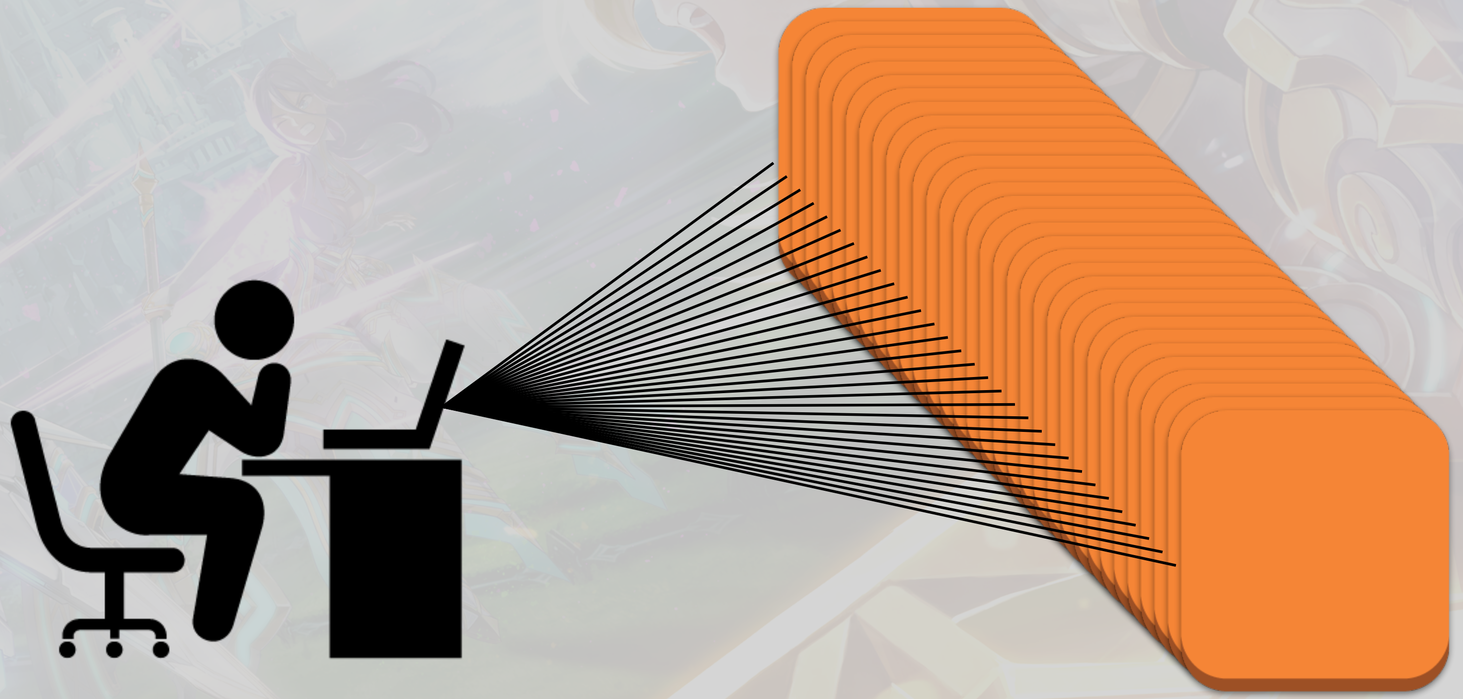
Ansibleを素直に使う？

Copyright (C) gumi Inc. 2018 All Rights Reserved

gumi

コードで接着

Ansibleを素直に使う？

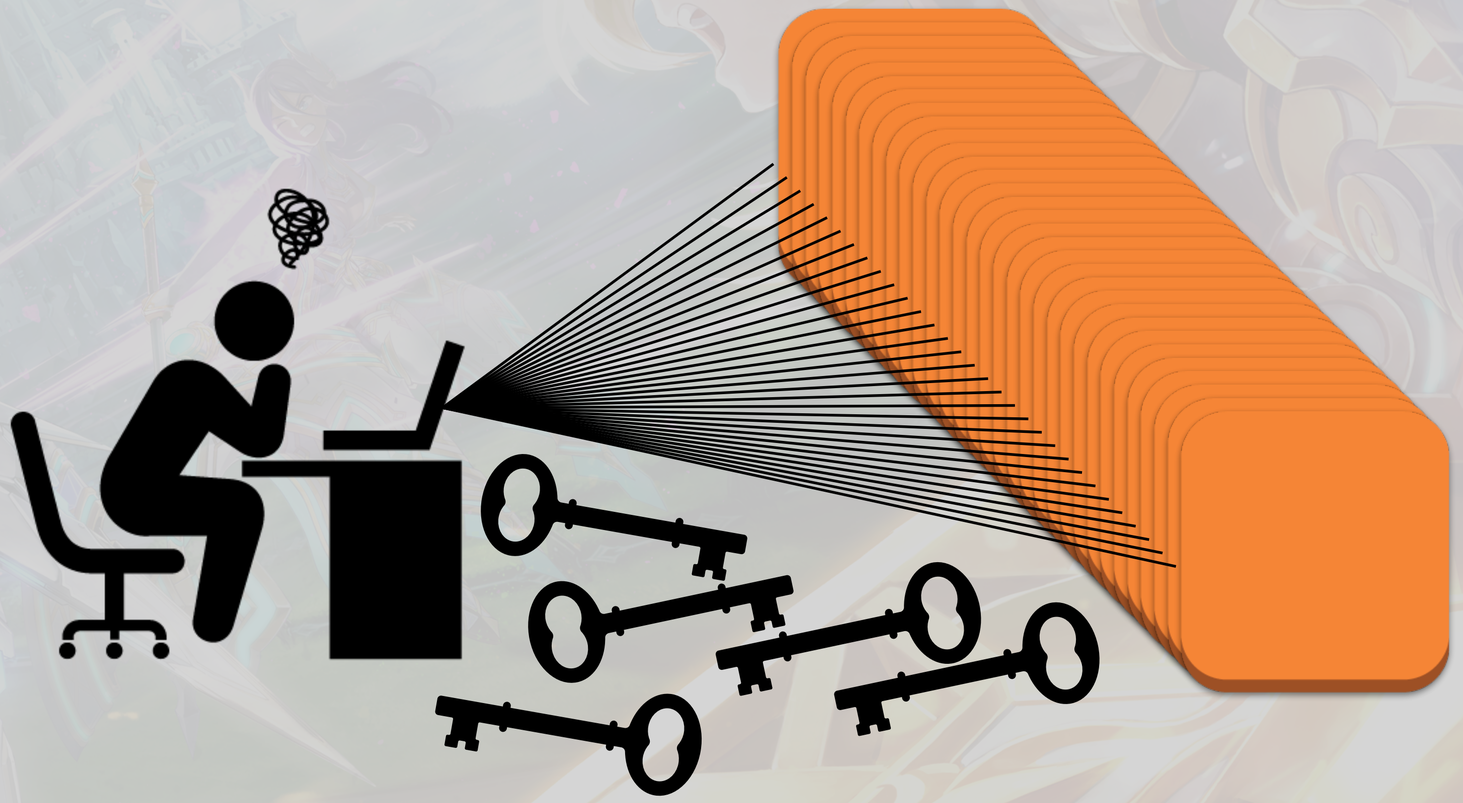


Copyright (C) gumi Inc. 2018 All Rights Reserved

gumi

コードで接着

Ansibleを素直に使う？



Copyright (C) gumi Inc. 2018 All Rights Reserved

コードで接着

Ansibleを素直に使う？



講義者ノート

SSH ログインが必要なため，鍵管理コストが高い

Copyright (C) gumi Inc. 2018 All Rights Reserved

コードで接着

Ansibleを素直に使う？

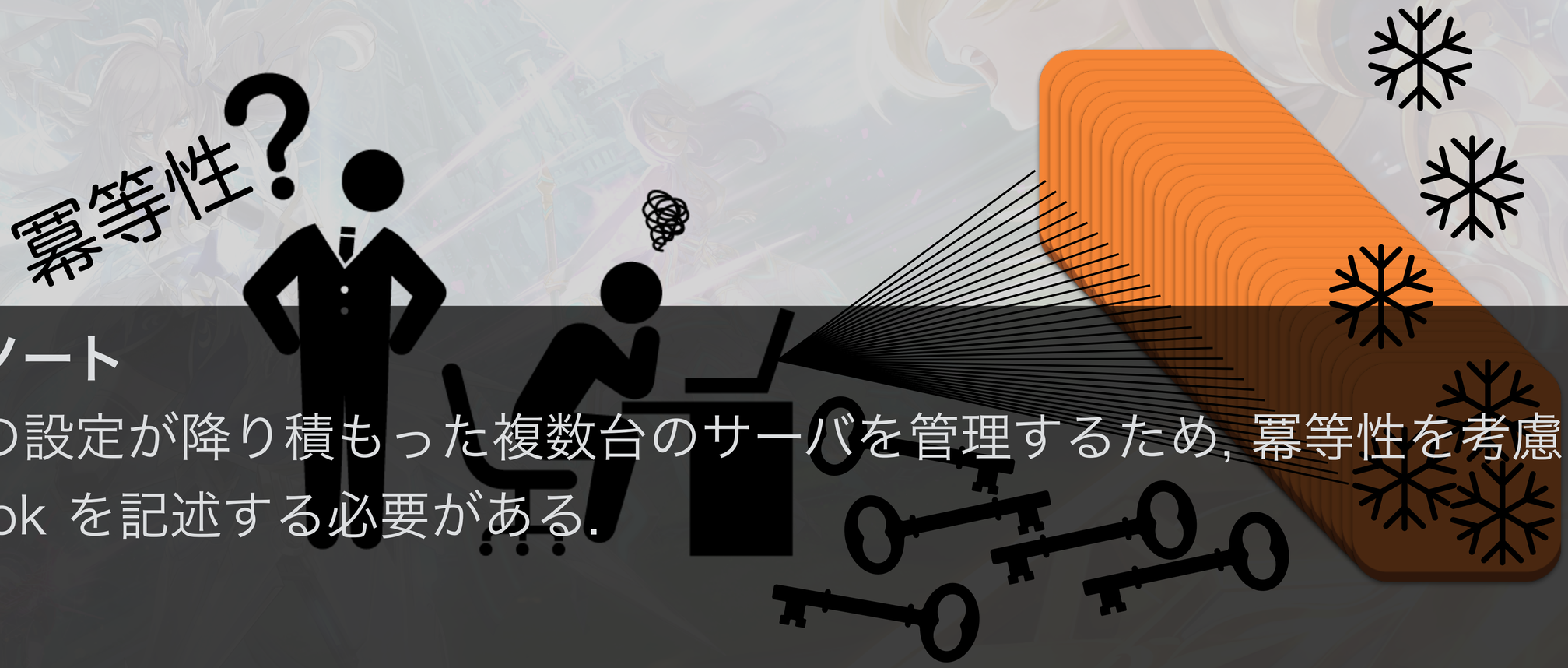


Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

コードで接着

Ansibleを素直に使う？



講義者ノート

運用中の設定が降り積もった複数台のサーバを管理するため, 冪等性を考慮して Playbook を記述する必要がある.

Copyright (C) gumi Inc. 2018 All Rights Reserved

コードで接着

Ansibleを素直に使う？



Copyright (C) gumi Inc. 2018 All Rights Reserved

コードで接着

Ansibleを素直に使う？



講義者ノート

何らかの事情により，稀に数台，Playbook の適用に失敗する．

Copyright (C) gumi Inc. 2018 All Rights Reserved

gumi

コードで接着

Ansibleを素直に使う？

オンプレ脳



Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

コードで接着

~~Ansibleを素直に使う？~~

Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツール(Ansible)とツール(CloudFormation)を**コード**で**接着**

~~Ansibleを素直に使う？~~



Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツール(Ansible)とツール(CloudFormation)を**コード**で**接着**

~~Ansibleを素直に使う？~~



運用中のEC2は一切変更しない

変更したくなったら
インスタンスごと捨てる

Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツール(Ansible)とツール(CloudFormation)を**コード**で**接着**

~~Ansibleを素直に使う？~~



運用中のEC2は一切変更しない

変更したくなったら
インスタンスごと捨てる

講義者ノート

AWS を使用しているため, 富豪的に EC2 インスタンスを作り捨て可能. 冪等性を考慮する必要はない. 運用に入った EC2 インスタンスの設定変更は絶対に行わない. SSH も禁止する.

Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

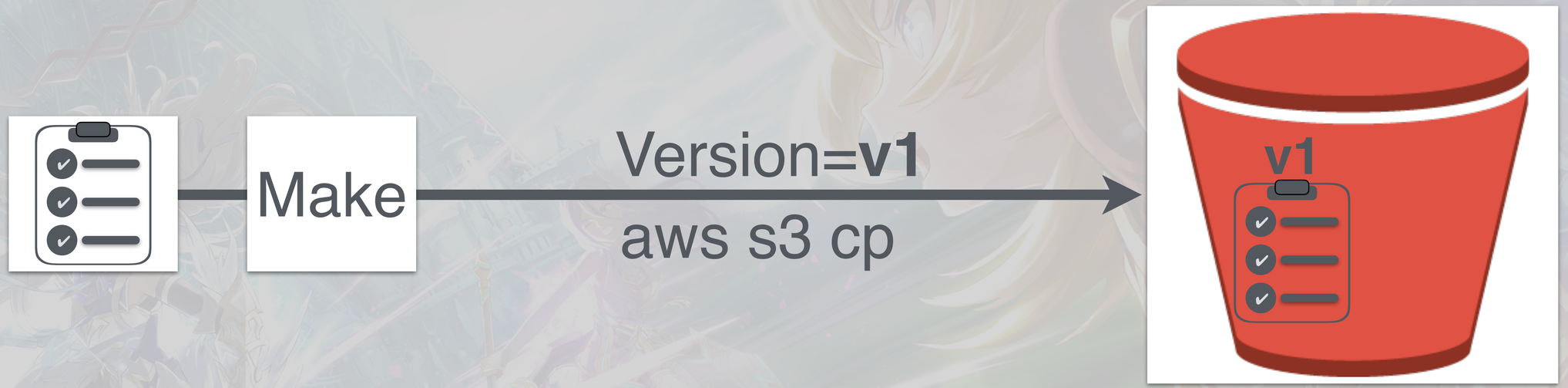
ツール(Ansible)とツール(CloudFormation)を**コード**で**接着**



Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツール(Ansible)とツール(CloudFormation)を**コード**で**接着**



講義者ノート

Playbook を S3 に設置する. 図にある通り, aws コマンドを実行するシンプルな Makefile を作成した.





Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツール(Ansible)とツール(CloudFormation)を**コード**で**接着**

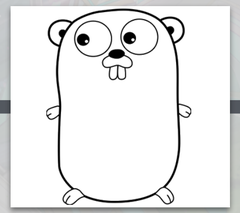


Make

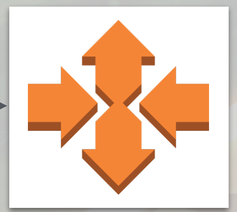



```
resources:
 - name: IMAGE_ID
   value: ami-0d74386b

 - name: PLAYBOOKVERSION
   value: v1
```







Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツール(Ansible)とツール(CloudFormation)を**コード**で**接着**



Make




```
resources:
  - name: IMAGE_ID
    value: ami-0d74386b
  - name: PLAYBOOKVERSION
    value: v1
```

講義者ノート

さきほど紹介したツールの YAML に AMI や Playbook のバージョンを指定し，CloudFormation に適用する.



Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツール(Ansible)とツール(CloudFormation)を**コード**で**接着**

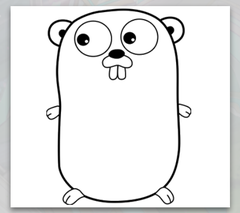


Make

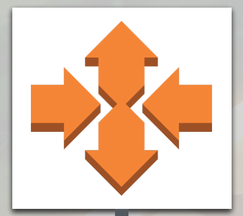



```
resources:
 - name: IMAGE_ID
   value: ami-0d74386b

 - name: PLAYBOOKVERSION
   value: v1
```









Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

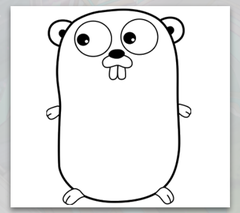
ツール(Ansible)とツール(CloudFormation)を**コード**で**接着**



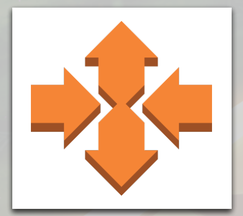
Make

```
resources:
 - name: IMAGE_ID
   value: ami-0d74386b

 - name: PLAYBOOKVERSION
   value: v1
```











Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツール(Ansible)とツール(CloudFormation)を**コード**で**接着**

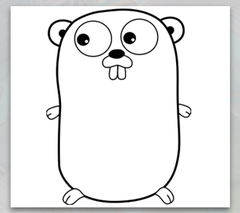


Make

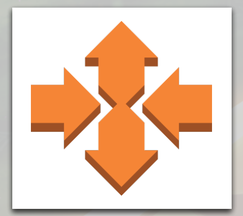



```
resources:
 - name: IMAGE_ID
   value: ami-0d74386b

 - name: PLAYBOOKVERSION
   value: v1
```









Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツール(Ansible)とツール(CloudFormation)を**コード**で**接着**



Make




```
resources:
  name: IMAGE_ID
    value: ami-0d74386b
  name: PLAYBOOKVERSION
    value: v1
```

講義者ノート

これで, Playbook v1 の適用を行った EC2 インスタンスが起動する. gumi では, AWS が配布している AMI に手を加えずに使用しており, 構成管理は全て Playbook で管理している. 起動時間が遅いなどの問題が発生するならば見直すが, 現状は, この管理方法で問題はない.

Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

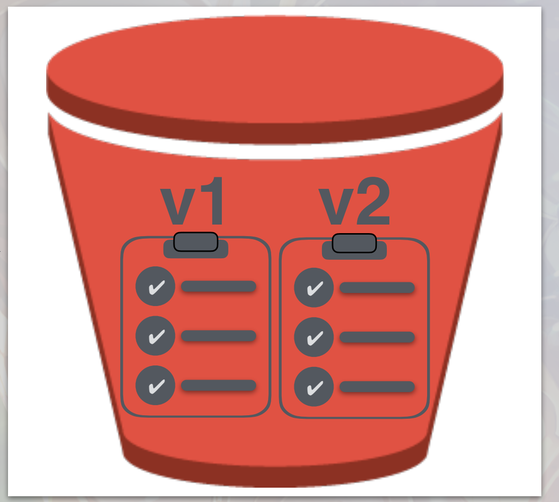
ツール(Ansible)とツール(CloudFormation)を**コード**で**接着**



Make

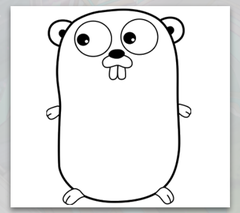
Version=**v2**

aws s3 cp

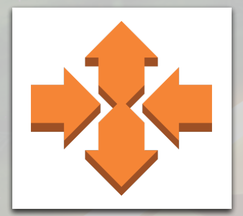



```
resources:
 - name: IMAGE_ID
   value: ami-0d74386b

 - name: PLAYBOOKVERSION
   value: v1
```





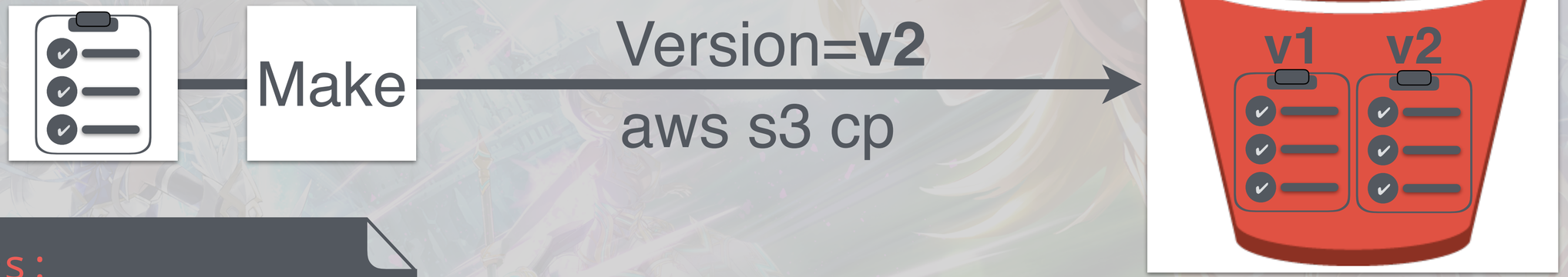




Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

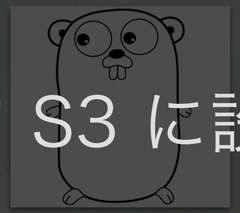
ツール(Ansible)とツール(CloudFormation)を**コード**で**接着**

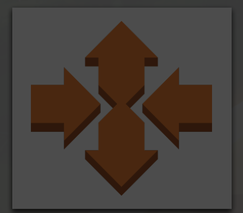



```
resources:
  - name: IMAGE_ID
    value: ami-0cd74386b
  - name: PLAYBOOKVERSION
    value: v1
```

講義者ノート

Playbook を変更し, v2 として S3 に設置する.





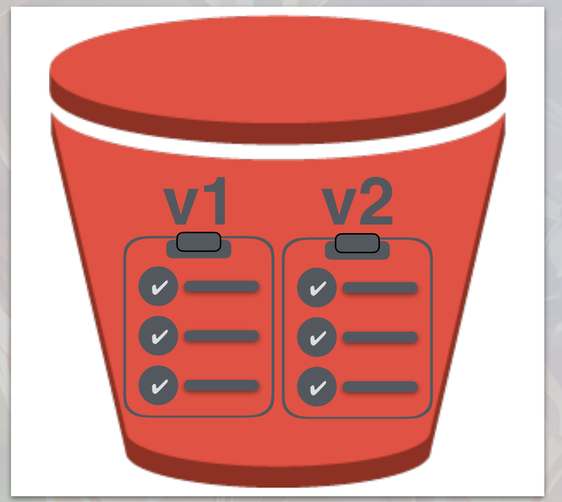

Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツール(Ansible)とツール(CloudFormation)を**コード**で**接着**

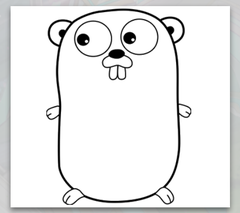


Make

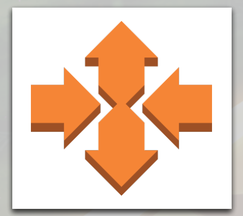



```
resources:
 - name: IMAGE_ID
   value: ami-0d74386b

 - name: PLAYBOOKVERSION
   value: v2
```









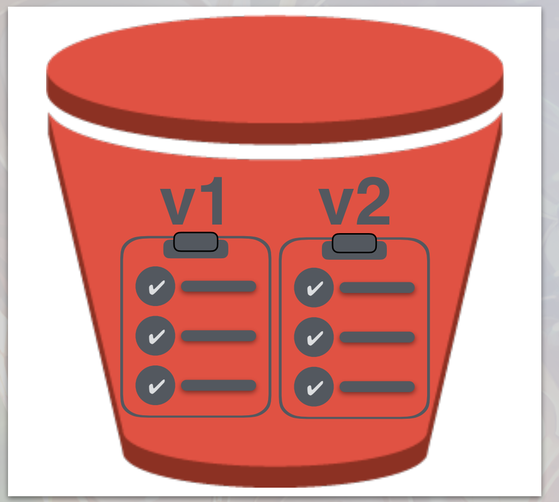
Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツール(Ansible)とツール(CloudFormation)を**コード**で**接着**

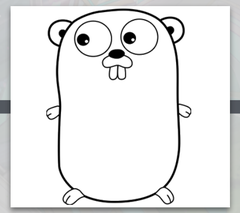


Make

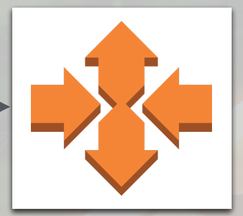



```
resources:
 - name: IMAGE_ID
   value: ami-0d74386b

 - name: PLAYBOOKVERSION
   value: v2
```









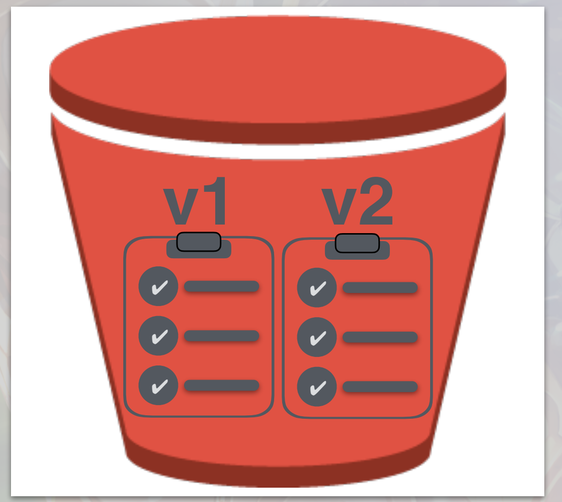
Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツール(Ansible)とツール(CloudFormation)を**コード**で**接着**



Make




```
resources:
  - name: IMAGE_ID
    value: ami-0d74386b
  - name: PLAYBOOKVERSION
    value: v2
```

講義者ノート

YAML の Playbook バージョンのみ変更したことで, AutoScaling の LaunchConfiguration の UserData のみが変更となる.





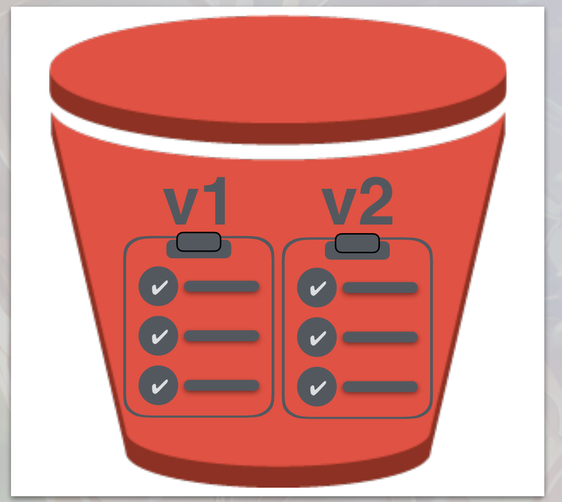
Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツール(Ansible)とツール(CloudFormation)を**コード**で**接着**

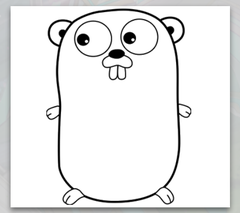


Make

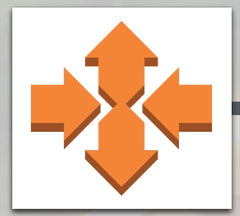



```
resources:
 - name: IMAGE_ID
   value: ami-0d74386b

 - name: PLAYBOOKVERSION
   value: v2
```











Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

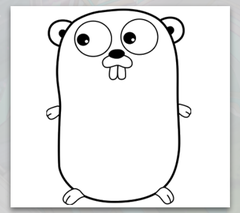
ツール(Ansible)とツール(CloudFormation)を**コード**で**接着**



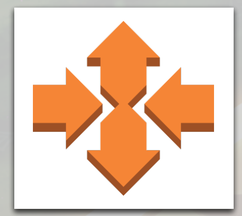
Make

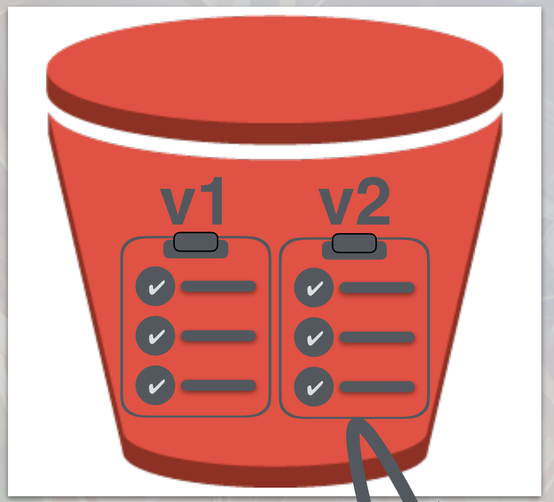
```
resources:
 - name: IMAGE_ID
   value: ami-0d74386b

 - name: PLAYBOOKVERSION
   value: v2
```













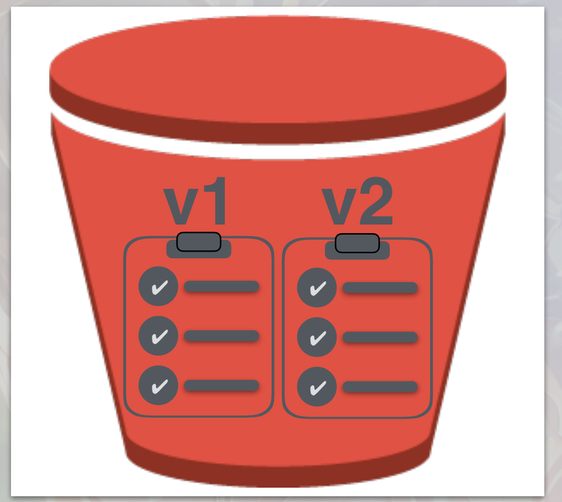
Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツール(Ansible)とツール(CloudFormation)を**コード**で**接着**

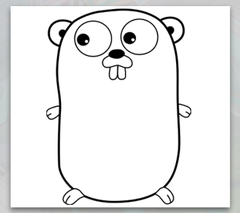


Make

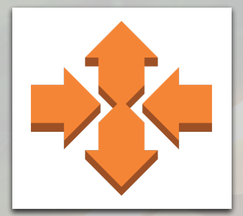



```
resources:
 - name: IMAGE_ID
   value: ami-0d74386b

 - name: PLAYBOOKVERSION
   value: v2
```











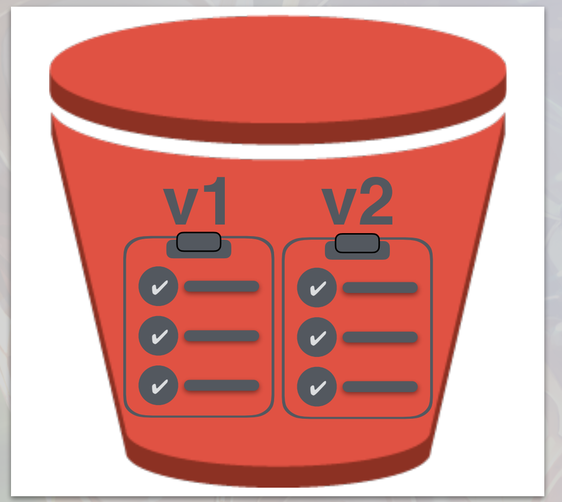
Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツール(Ansible)とツール(CloudFormation)を**コード**で**接着**



Make




```
resources:
  Name: IMAGE_ID
    value: ami-0d74386b
  Name: PLAYBOOKVERSION
    value: v2
```

講義者ノート

これで, Playbook v2 が適用された EC2 インスタンスが起動する. しばらく様子を見て問題がなければ, Playbook v1 が適用された EC2 インスタンスを停止する.











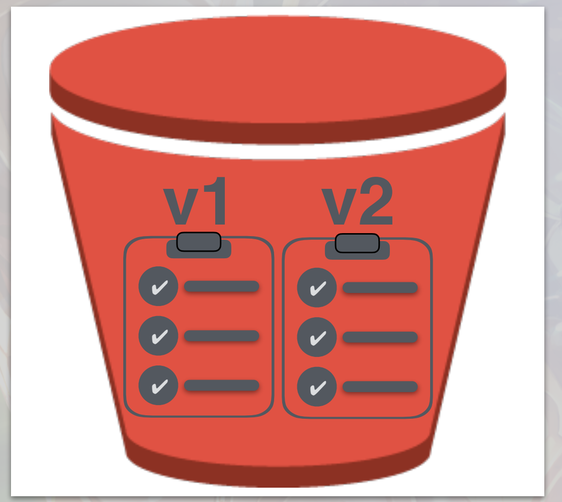
Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツール(Ansible)とツール(CloudFormation)を**コード**で**接着**

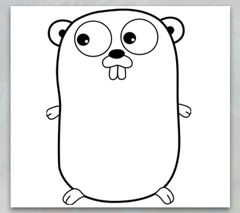


Make

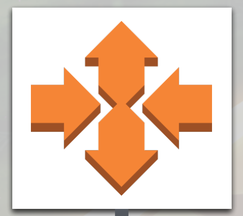



```
resources:
 - name: IMAGE_ID
   value: ami-0d74386b

 - name: PLAYBOOKVERSION
   value: v2
```











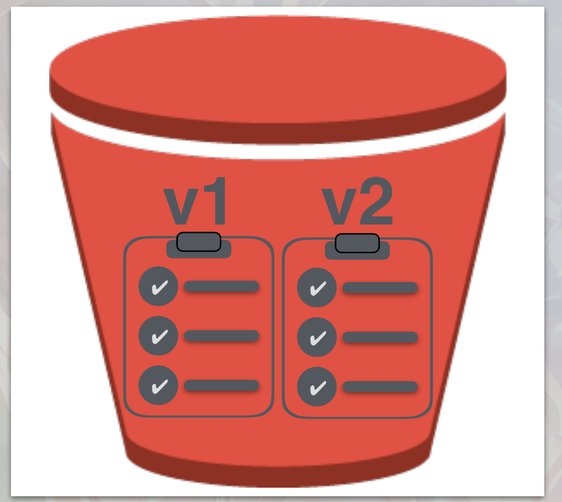
Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツール(Ansible)とツール(CloudFormation)を**コード**で**接着**



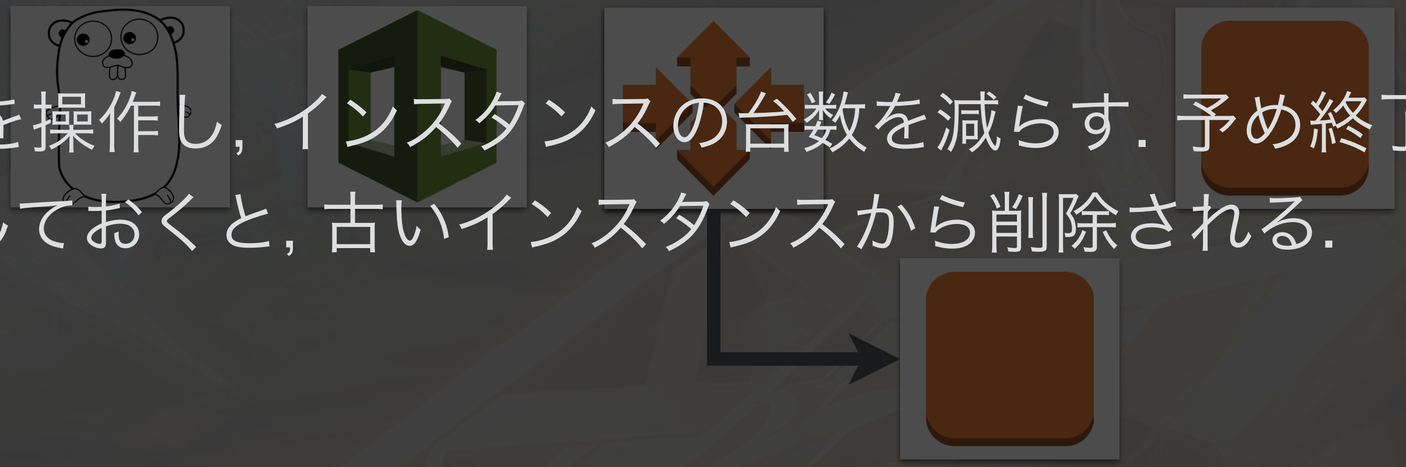
Make




```
resources:
  name: IMAGE_ID
  value: ami-0d74386b
  name: PLAYBOOKVERSION
  value: v2
```

講義者ノート

aws cli 経由で AutoScaling を操作し, インスタンスの台数を減らす. 予め終了ポリシーを OldestInstance にしておくと, 古いインスタンスから削除される.



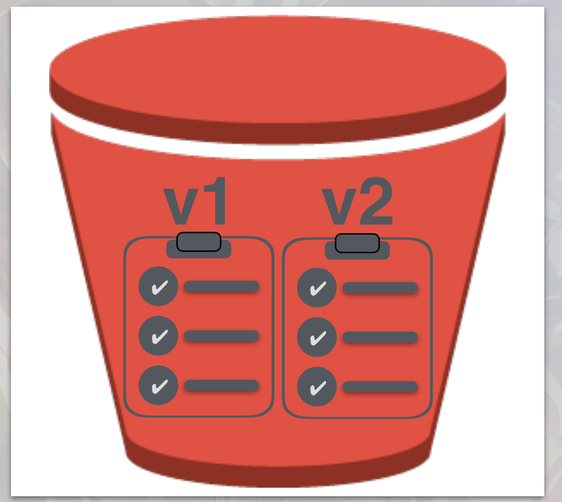
Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツール(Ansible)とツール(CloudFormation)を**コード**で**接着**

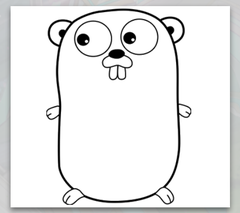


Make

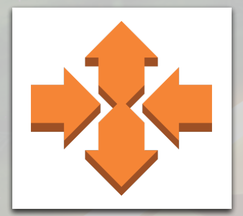



```
resources:
 - name: IMAGE_ID
   value: ami-0d74386b

 - name: PLAYBOOKVERSION
   value: v2
```









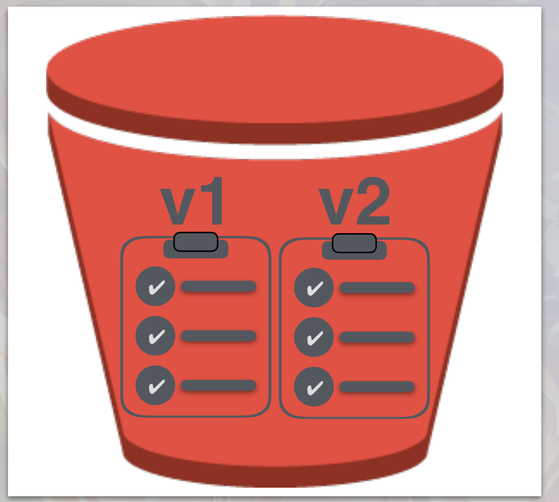
Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツール(Ansible)とツール(CloudFormation)を**コード**で**接着**



Make




```
resources:
  name: IMAGE_ID
  value: ami-0d74386b
  name: PLAYBOOKVERSION
  value: v2
```

講義者ノート

一連の流れの中で, aws cli などの複数のツールを使い分けている. 全てをまとめられないか？と考えるかもしれないが, 実際にまとめようとすると, 人の判断が必要なところをコード化する際に苦労する. ただし, 改善の余地はあるため, 今後の課題としている.

Copyright (C) gumi Inc. 2018 All Rights Reserved

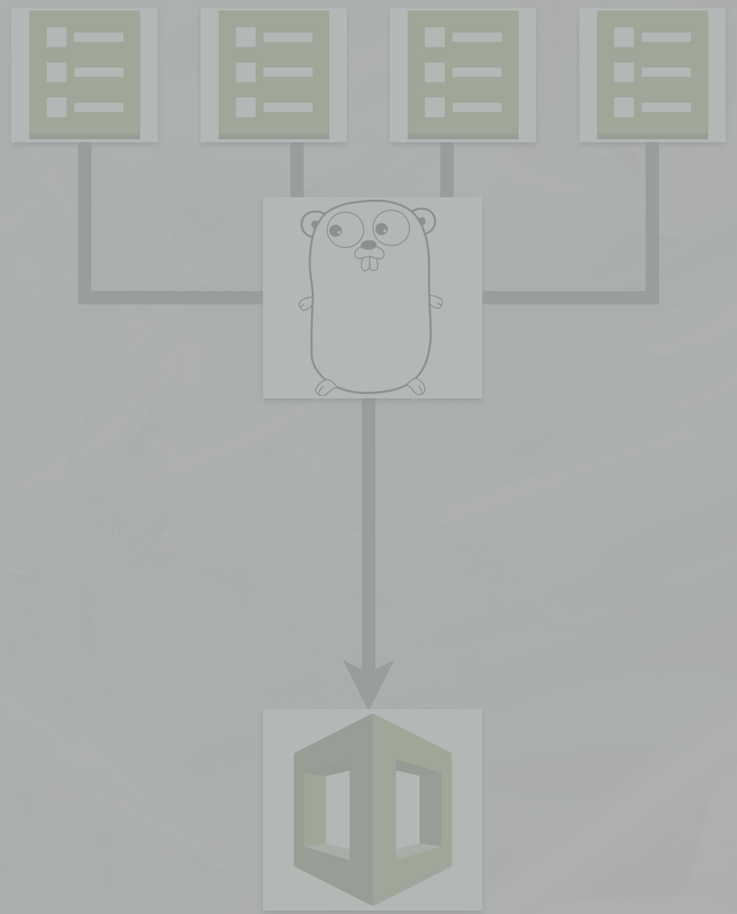
コード化

ツールを選択





コードで補う



コードで接着



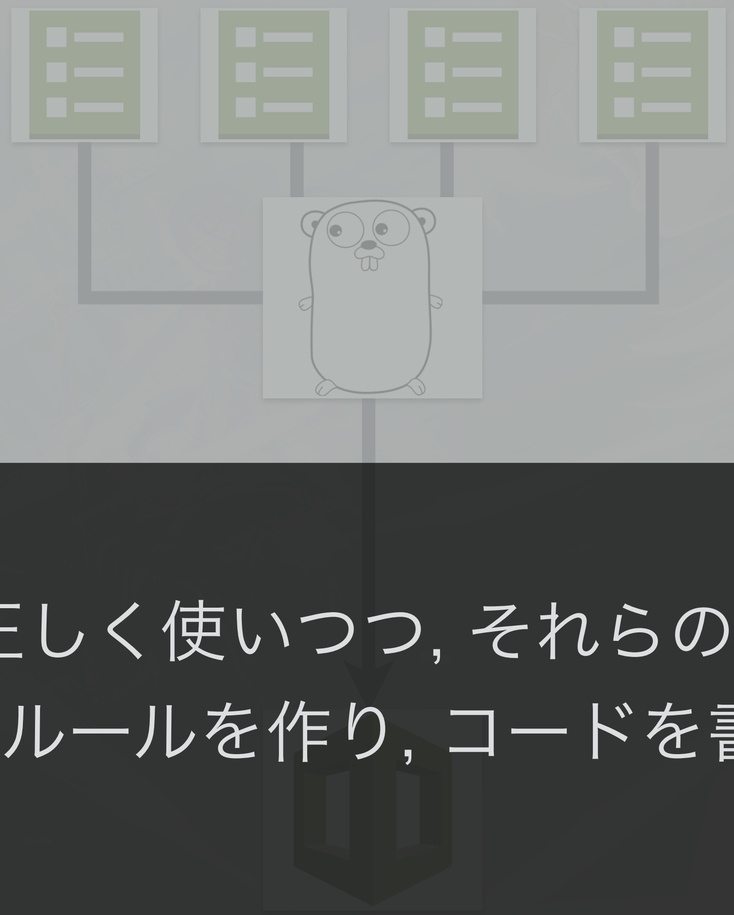
Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツールを選択



コードで補う



コードで接着



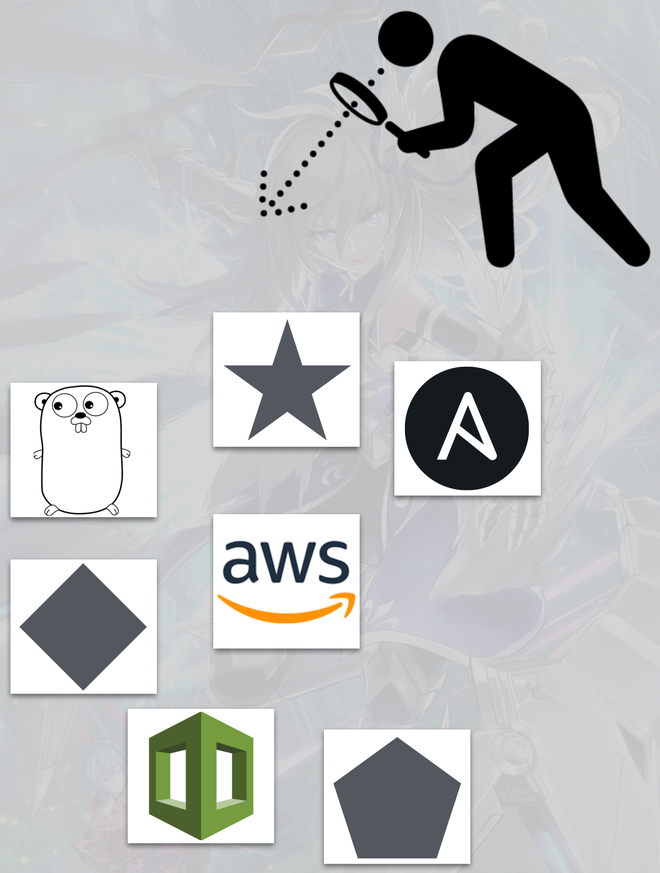
講義者ノート

既存ツールの本質を理解し, 正しく使いつつ, それらの隙間を埋めることに注力すること. 隙間を埋めるために, ルールを作り, コードを書くこと.

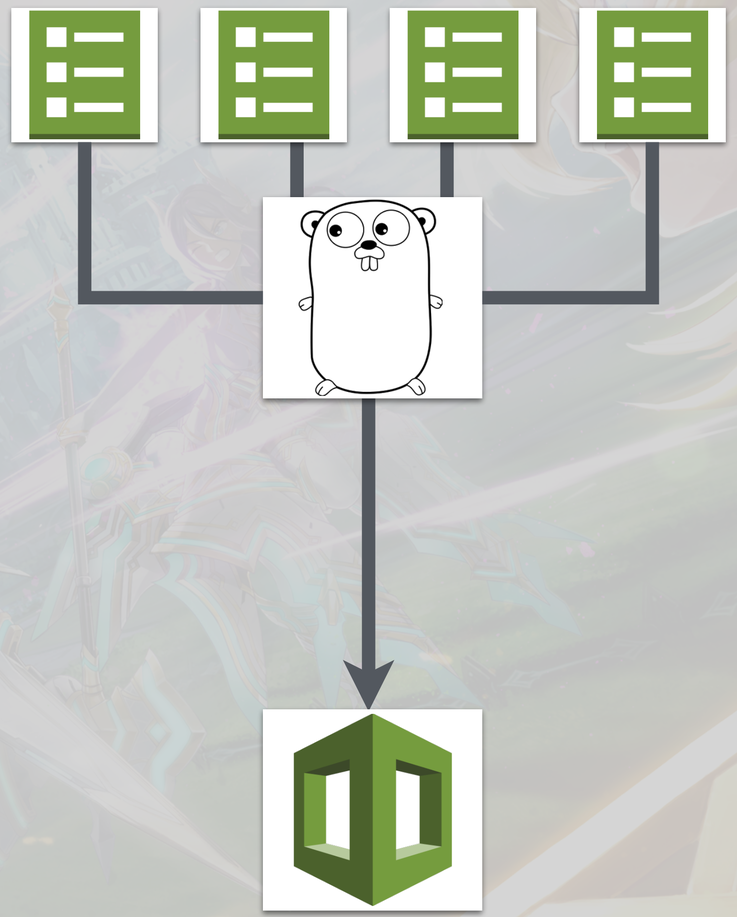
Copyright (C) gumi Inc. 2018 All Rights Reserved

コード化

ツールを選択



コードで補う



コードで接着

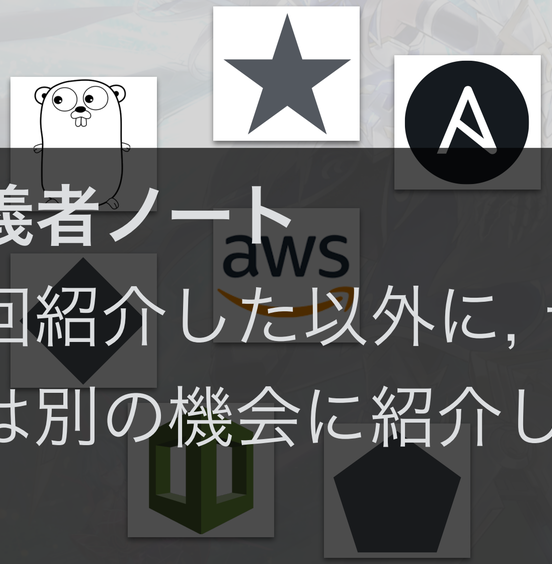


Copyright (C) gumi Inc. 2018 All Rights Reserved

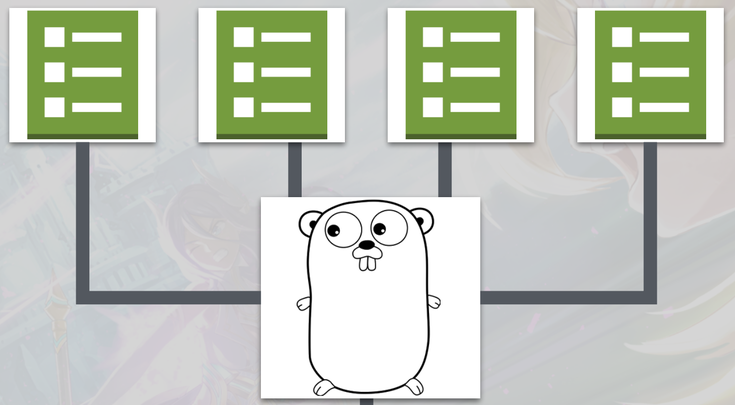
コード化

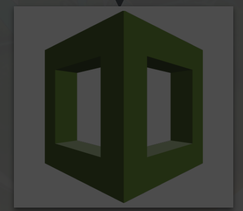
ツールを選択





コードで補う





コードで接着





講義者ノート

今回紹介した以外に, デプロイツールなどもインフラチームが開発している. それらは別の機会に紹介したい.

Copyright (C) gumi Inc. 2018 All Rights Reserved

目次

悪戦苦闘した
改善の歩み
は8年かけても終わらない

▽

改善を促進する
コード化
の進め方は試行錯誤の連続

▽

まとめ

Copyright (C) gumi Inc. 2018 All Rights Reserved

目次

悪戦苦闘した
改善の歩み
は8年かけても終わらない

▽

改善を促進する
コード化
の進め方は試行錯誤の連続

▽

まとめ

Copyright (C) gumi Inc. 2018 All Rights Reserved

まとめ



Copyright (C) gumi Inc. 2018 All Rights Reserved

まとめ

持ち帰ってほしいこと

実験的であること　失敗から学ぶこと

First to Try, First to Fail, First to Recover!

事例紹介

スタートアップ	▷	一部上場企業
一人	▷	チーム,組織
手作業	▷	コード化

Copyright (C) gumi Inc. 2018 All Rights Reserved

まとめ

持ち帰ってほしいこと

実験的であること　失敗から学ぶこと

First to Try, First to Fail, First to Recover!

Copyright (C) gumi Inc. 2018 All Rights Reserved

まとめ

持ち帰ってほしいこと

実験的であること　失敗から学ぶこと

First to Try, First to Fail, First to Recover!

講義者ノート

「実験して失敗して学ぶ」という行動は, ツールだけに限った話しではなく, 組織の体制やチーム活動など, あらゆるものを対象としている.

Copyright (C) gumi Inc. 2018 All Rights Reserved

まとめ

持ち帰ってほしいこと

実験的であること　失敗から学ぶこと

First to Try, First to Fail, First to Recover!

当事者の方々へ

Copyright (C) gumi Inc. 2018 All Rights Reserved

まとめ

持ち帰ってほしいこと

実験的であること　失敗から学ぶこと

First to Try, First to Fail, First to Recover!

当事者の方々へ

小さく試し, 小さく失敗

失敗が前提で良い

ただし謙虚に学ぶこと

かならず改善すること

Copyright (C) gumi Inc. 2018 All Rights Reserved

まとめ

持ち帰ってほしいこと

実験的であること　失敗から学ぶこと

First to Try, First to Fail, First to Recover!

当事者の方々へ

小さく試し, 小さく失敗

失敗が前提で良い

ただし謙虚に学ぶこと

かならず改善すること

利用者や外野の方々へ

Copyright (C) gumi Inc. 2018 All Rights Reserved

まとめ

持ち帰ってほしいこと

実験的であること　失敗から学ぶこと

First to Try, First to Fail, First to Recover!

当事者の方々へ

小さく試し, 小さく失敗

失敗が前提で良い

ただし謙虚に学ぶこと

かならず改善すること

利用者や外野の方々へ

減点方式から加点方式へ

前のめりの失敗には寛容に

Copyright (C) gumi Inc. 2018 All Rights Reserved

まとめ

持ち帰ってほしいこと

実験的であること　失敗から学ぶこと

First to Try, First to Fail, First to Recover!

当事者の方々へ

小さく試し，小さく失敗

失敗が前提で良い

ただし謙虚に学ぶこと

かならず改善すること

利用者や外野の方々へ

減点方式から加点方式へ

前のめりの失敗には寛容に

講義者ノート

減点方式で運用者を責めると，守りに入り，無駄な防御行動で多大な工数を浪費する．「前のめりの失敗」は，寛容に受け止めること．

Copyright (C) gumi Inc. 2018 All Rights Reserved

まとめ

持ち帰ってほしいこと

実験的であること　失敗から学ぶこと

First to Try, First to Fail, First to Recover!

小さく試し，小さく失敗

失敗が前提で良い

ただし謙虚に学ぶこと

かならず改善すること

減点方式から加点方式へ

前のめりの失敗には寛容に

いや，まて…

Less DevOps, More Code

どこへ行った?

Copyright (C) gumi Inc. 2018 All Rights Reserved

まとめ

Less DevOps, More Code

今日の話しは,
インフラエンジニアの視点からなら

Less Ops, More Code

Copyright (C) gumi Inc. 2018 All Rights Reserved

Less DevOps, More Code

インフラチームと共通開発チームから
ゲーム開発チームへのメッセージ

Copyright (C) gumi Inc. 2018 All Rights Reserved

まとめ

gumi

Less DevOps, More Code

インフラチームと共通開発チームから
ゲーム開発チームへのメッセージ

私達は, ゲームの面白さに直結しない Dev と Ops を引き受け, 皆さんにゲームの面白さに直結するコードを, もっとたくさん書いてもらえるよう, 継続的に業務改善し続けます.

Copyright (C) gumi Inc. 2018 All Rights Reserved

まとめ

Less DevOps, More Code

インフラチームと共通開発チームから
ゲーム開発チームへのメッセージ

私達は，ゲームの面白さに直結しない Dev と Ops を
引き受け，皆さんにゲームの面白さに直結するコードを
もっとたくさん書いてもらえるよう，継続的に業務改善
し続けます．

講義者ノート

このような姿勢がインフラ運用に必要だと gumi では考えている. みなさんの参考になることを願っている.

Copyright (C) gumi Inc. 2018 All Rights Reserved