

DMM GAMESが AWSに移行した理由

株式会社DMM.comラボ
田所健太



自己紹介

田所 健太

株式会社DMM.comラボ

ゲーム開発本部 第3システム部

(オンラインゲームの開発と運用をしている部署)

PHP+Javascriptを嗜んでいるWebエンジニア



アジェンダ

- **DMM GAMES**について
- 従来のゲームのサーバ環境について
- **AWS**への移行理由
- 将来の展望



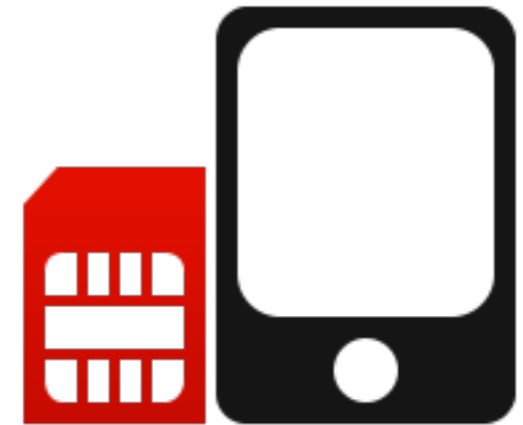
DMMのサービスの一部



動画配信



通販



モバイル



電子書籍



レンタル



ゲーム



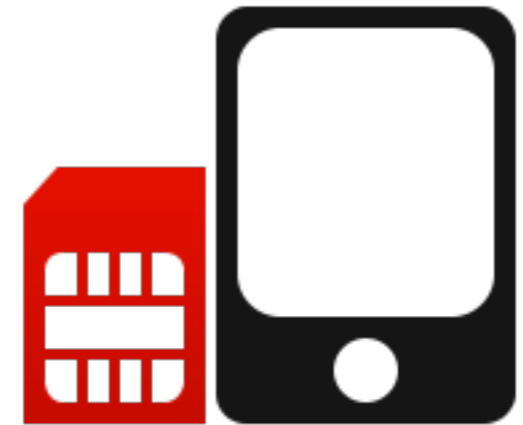
DMMのサービスの一部



動画配信



通販



モバイル



電子書籍



レンタル



ゲーム



DMM GAMES

- ・ **オンラインゲームの開発・運用・配信をしている**
- ・ **“艦これ”などの有名タイトルも配信中です**
- ・ **開発・運用チームに所属しています**



アジェンダ

- **DMM GAMES**について
- **従来のゲームのサーバ環境**について
- **AWSへの移行理由**
- **将来の展望**



DMMのサービスの一部



動画配信



通販



モバイル



電子書籍



レンタル



ゲーム



DMMのサービスの一部



DMMは社内に
大規模なインフラ環境を持つ



電子書籍



レンタル



ゲーム



従来の流れ

- 1. 想定アクセス数を算出する**
- 2. 負荷テストから本番に必要なサーバー数を算出する**
- 3. インフラチームにサーバー構築の依頼する**
- 4. 用意されたサーバーでサービス開始**
- 5. インフラチームと協力して負荷監視**
- 6. サーバーの台数を増減させていく**



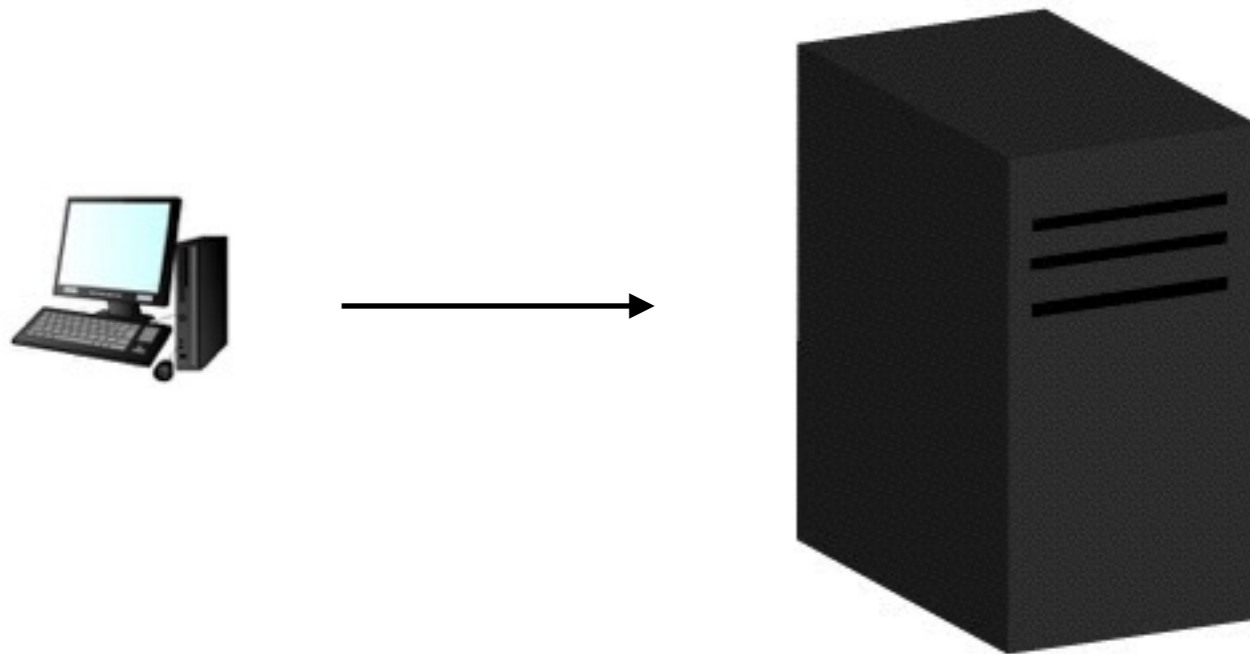
アジェンダ

- **DMM GAMES**について
- 従来のゲームのサーバ環境について
- **AWSへの移行理由**
- 将来の展望

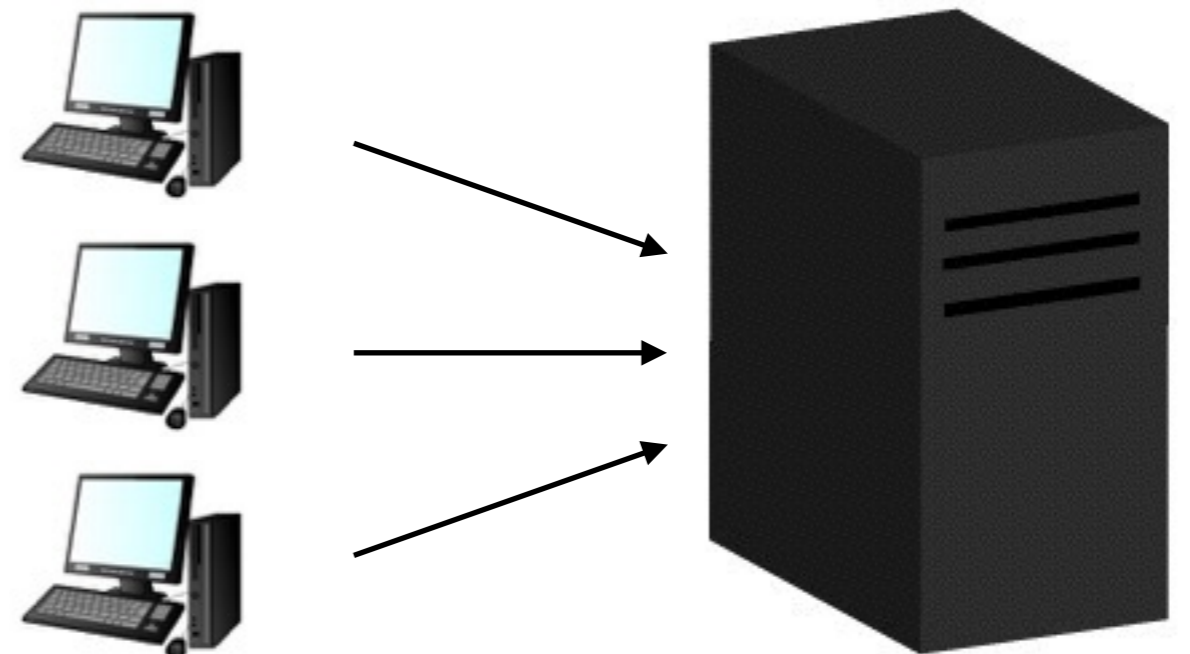


失敗談その1

予想



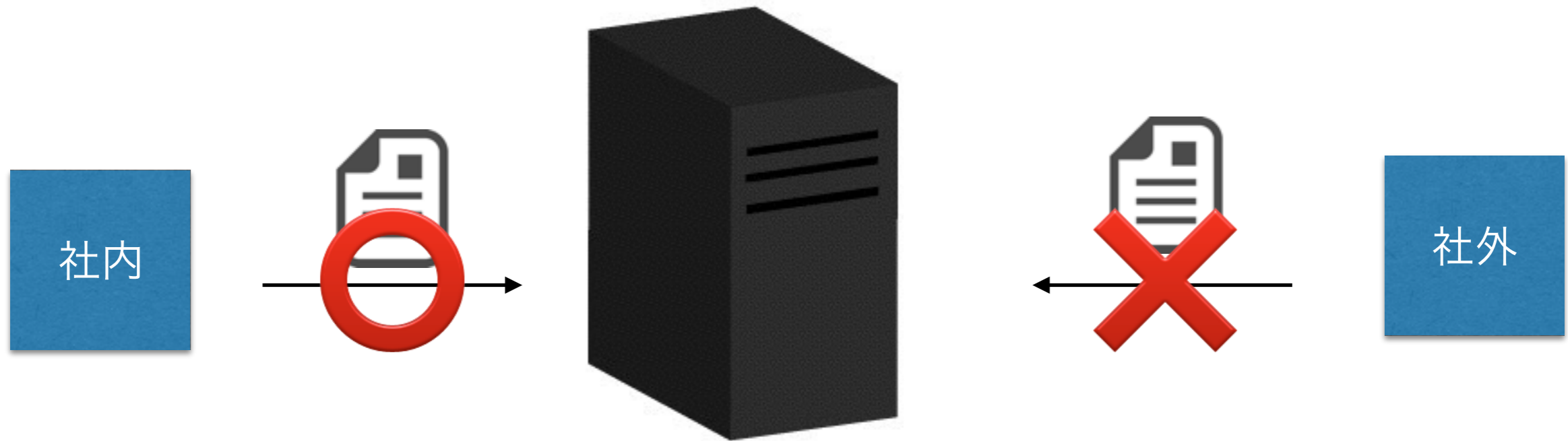
実際



- ・ サービス初日にサーバが落ちてしまった



失敗談その2

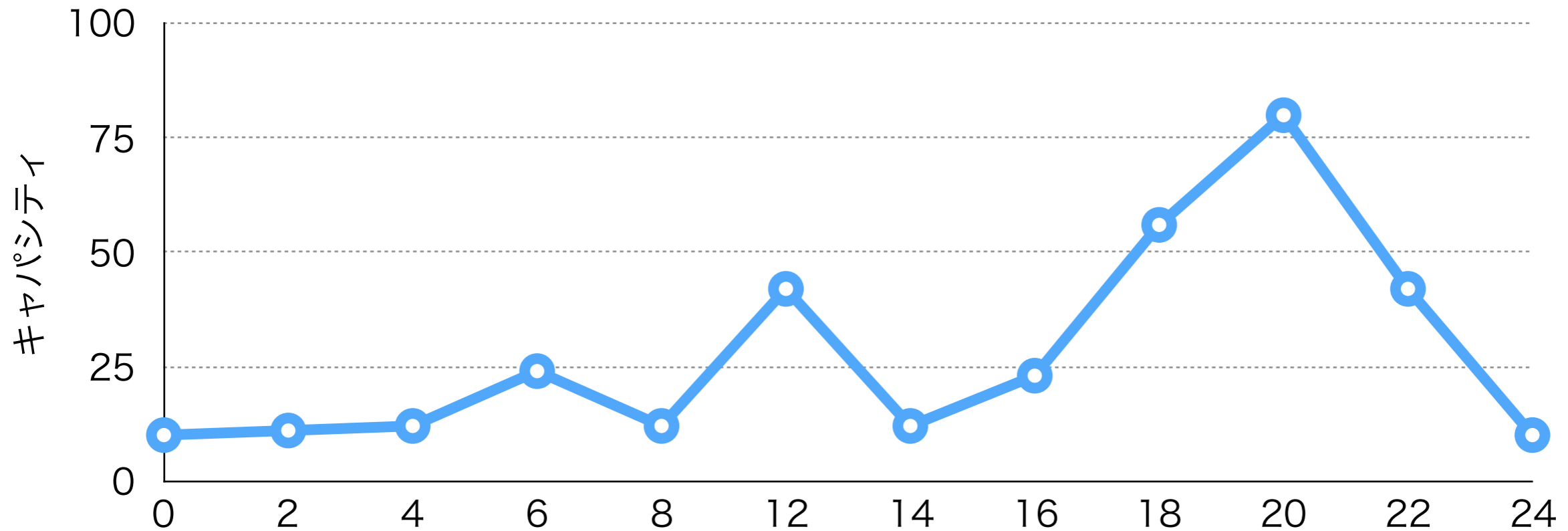


- **運用中のゲームを協力会社に委託をしようとした**
- **途中で社内インフラにアクセス出来ないことが発覚**
- **委託を断念し、社内で運用を継続していくことに**



失敗談その3

サーバの負荷状況



- ・ **ピークタイム時の負荷に合わせてサーバを増やすとピークタイム時以外はサーバが余る状態に**



不満が溜まりつつあった



クラウド
にしたい



説得を開始

- ・ **インフラにかかる時間が短縮できそうですよ**
- ・ **初日にサーバが落ちることを防げそうですよ**
- ・ **人気が発発してもすぐにサーバ増やせそうですよ**



反対意見

- ・ **オンプレミスのほうがサーバ代は安くなる**

→ **想定外の事態での損失を考えるとクラウドのが安い**



クラウドの利用が決定



AWS
にししたい



説得を開始

- **事例が多いので、何かあっても調べられますよ**
- **本社(東京)の転職者に経験者がいるので、何かあっても聞けますよ**
- **APIが豊富で色々な作業を自動化できそうですよ**



AWSの利用が決定

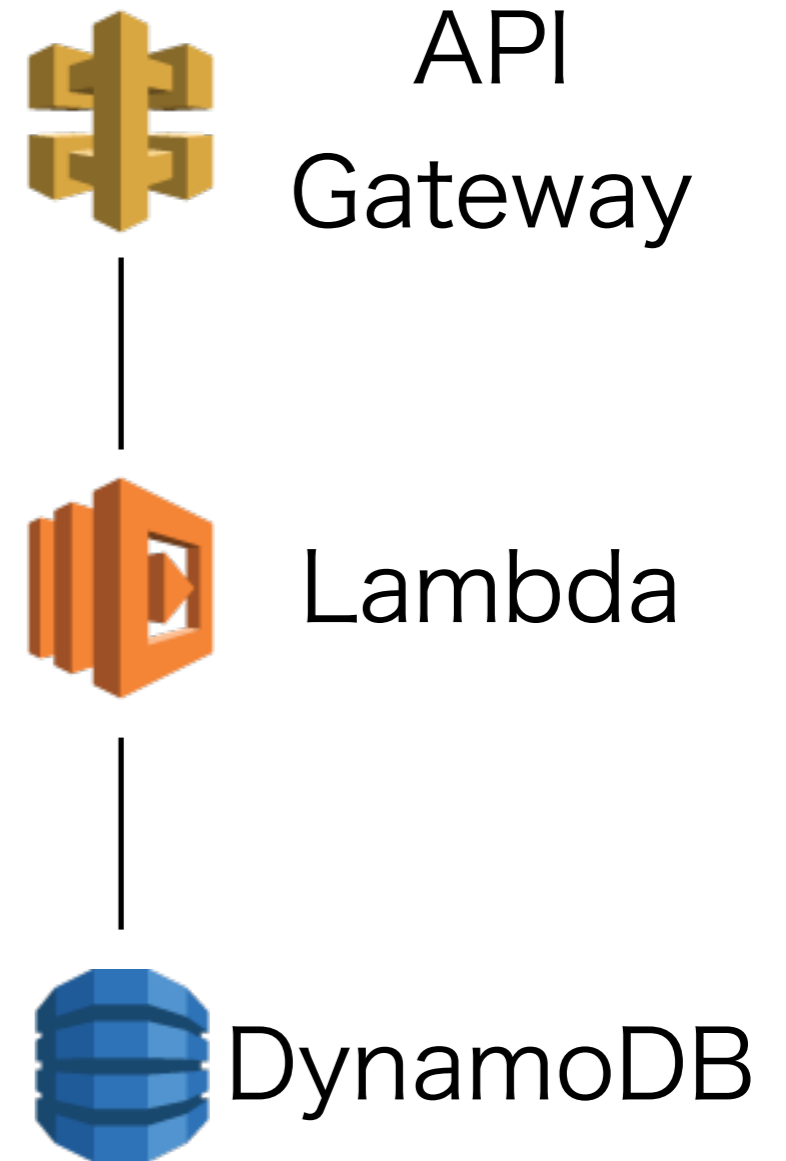
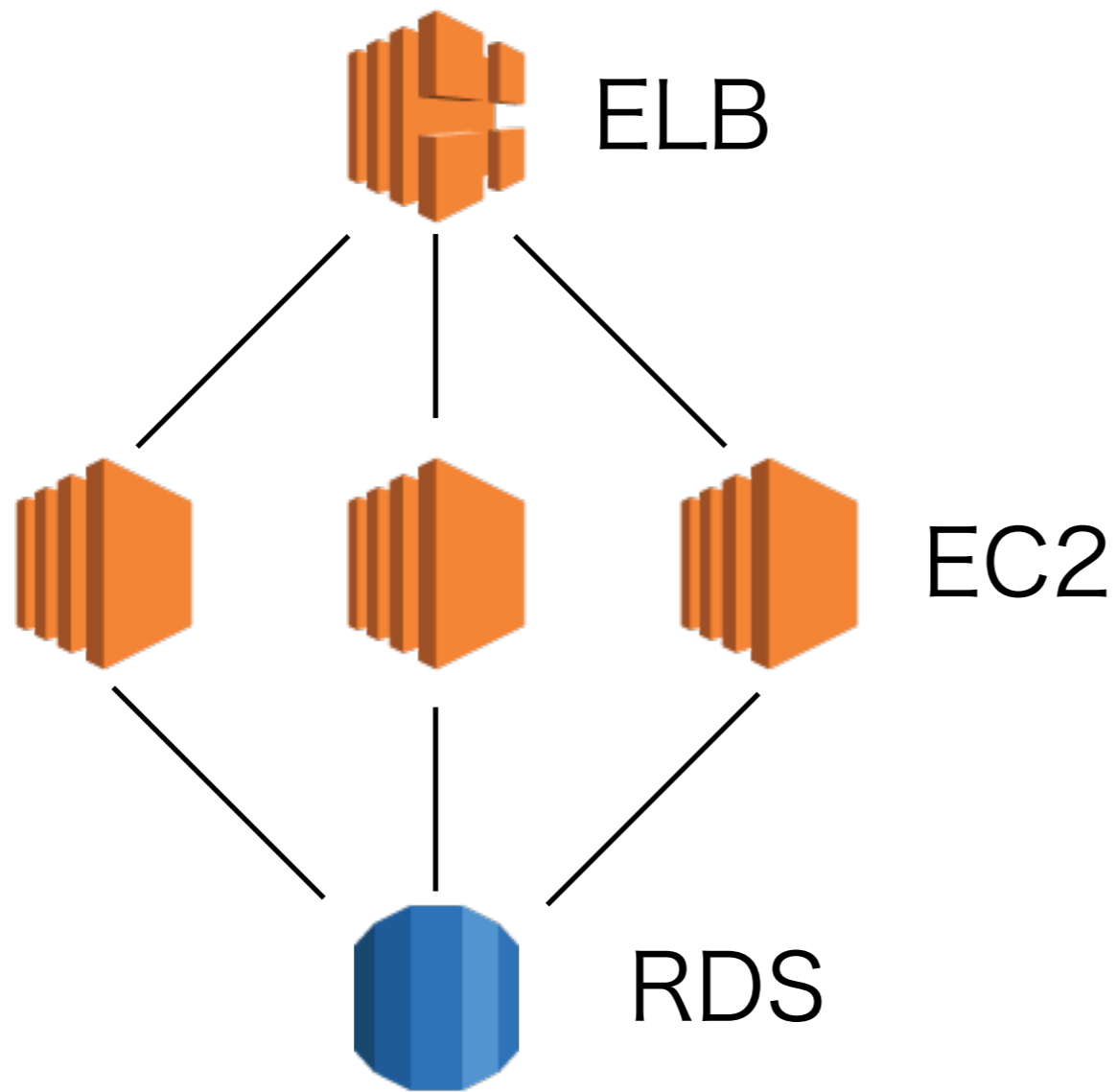


アジェンダ

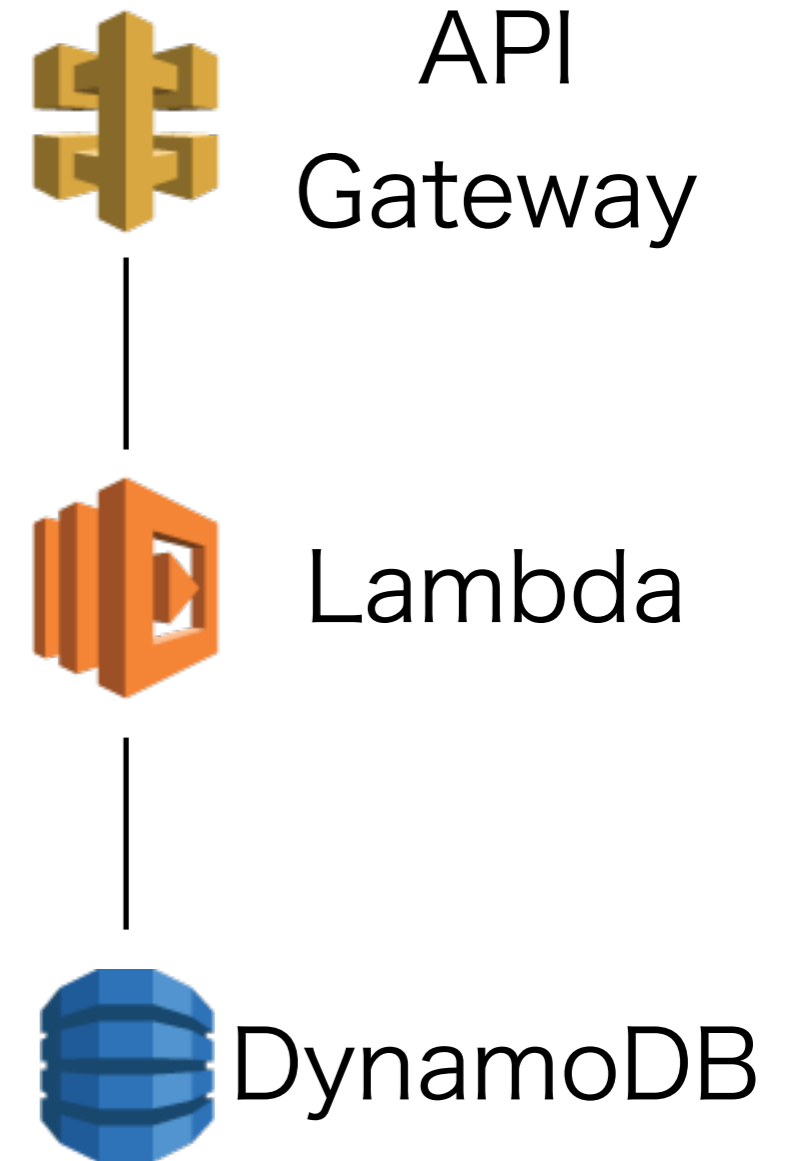
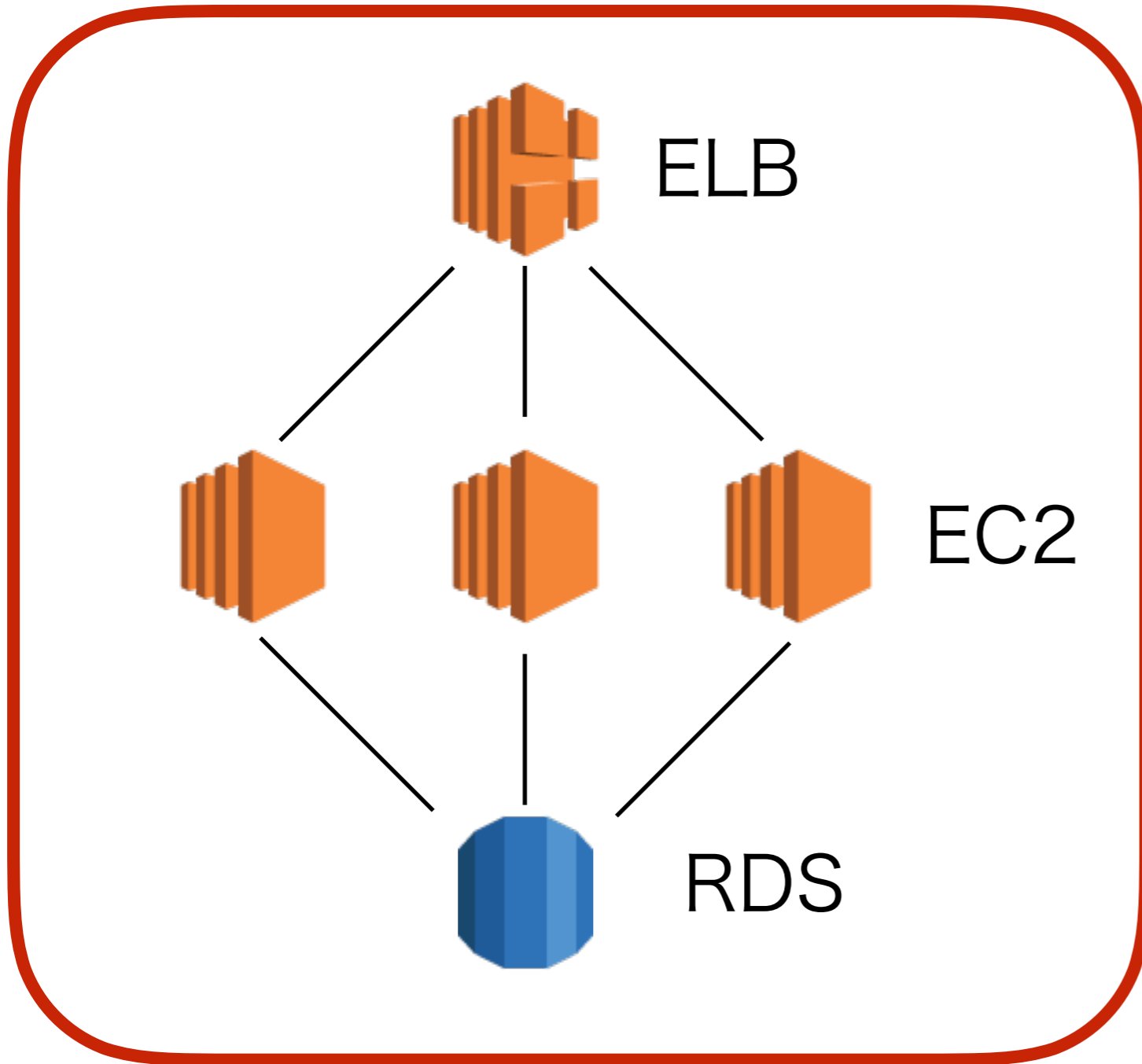
- **DMM GAMES**について
- **従来のゲームのサーバ環境**について
- **AWSへの移行理由**
- **将来の展望**



構成に悩む



ELB+EC2+RDSに決定



決定理由

- ・ **オンプレミスの時の構成がベース**
- ・ **DynamoDB用のテーブル設計をする時間が取れない(NoSQLのテーブル設計をしたことがない)**



利用する予定のサービス



ELB:ロードバランサ



EC2:一般的なサーバで動的コンテンツを配信する



RDS(Auroa):データベースサーバでデータの保存



利用する予定のサービス



Route53:DNS



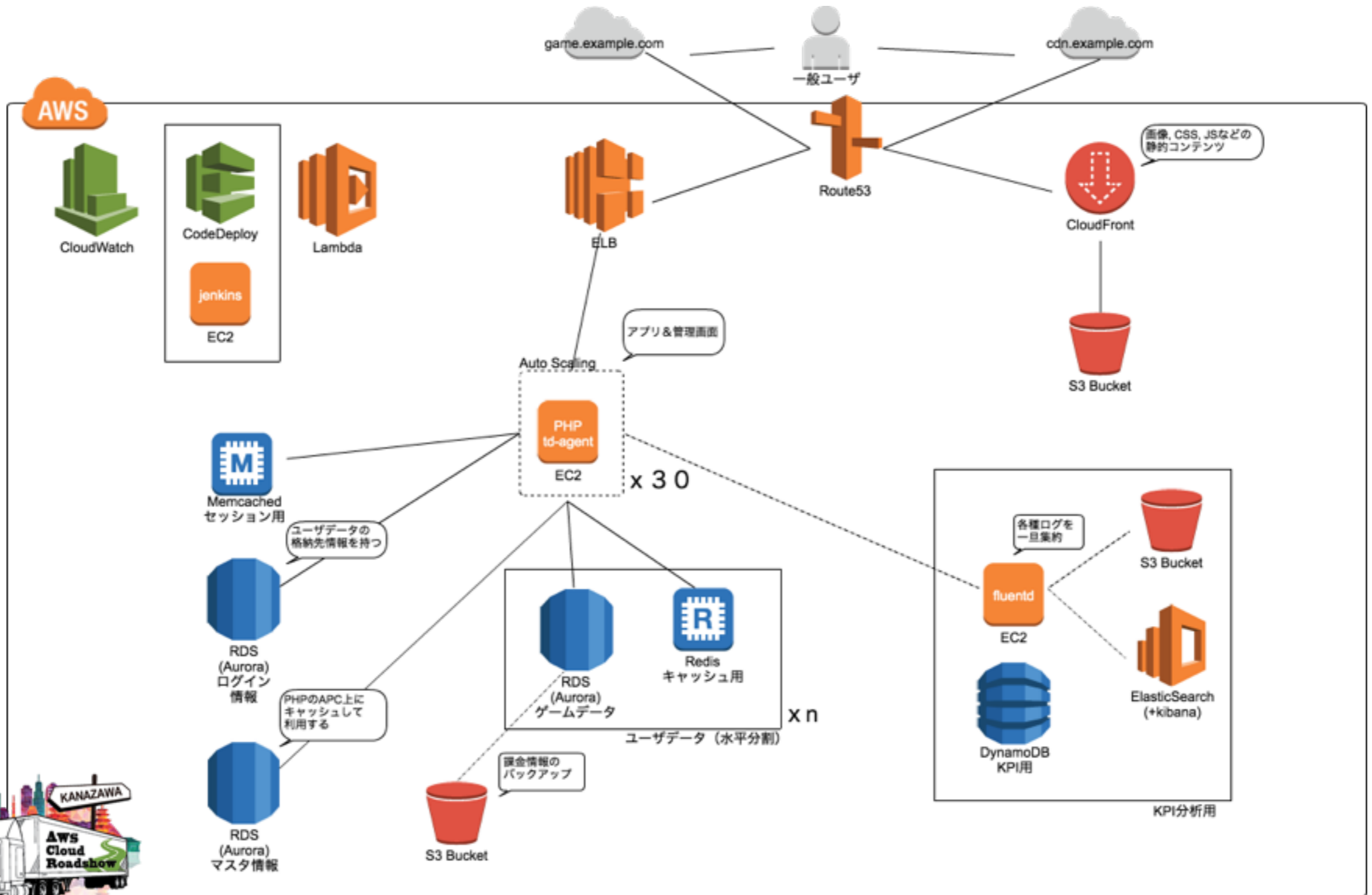
CloudFront:CDNで静的コンテンツを配信する



S3 :ストレージサービス



サービス構成予定図



使い始めてみて

- **とりあえず、試してみるが可能になった**
- **AWS CLIが凄く便利！**



試してみたの例

- **使ったことないサービスで見積もりが取れない**
- **AWS上で構築して、実際に試して見る**
- **すぐにAWSのサーバを落とせば、殆どお金がかからない**
- **今までだと、サーバの用意が必要ですぐには試せなかった**



AWS CLI

- **AWSの全部の機能を叩けるぐらいに揃ってる**
(ターミナルで全部出来るのでshellで楽できる)
- **Lambdaでイベント監視して、自動化もできそう**
(まだ試せていない)



AWS CLIで出来ることの例

- **出勤時間と退勤時間に合わせて、EC2インスタンスを落としたい（立ち上げておくとお金がかかる）**
- **cronにaws cliのコマンドを設定すればOK!**



まとめ

AWSに移行することにより、

時間の節約

サービスの稼働安定

が達成できそう

