『このファン』における マネージドサービスドリブンな アーキテクチャへの挑戦

Nobuta Satoshi

Sumzap, Inc. / SRE Team

aws SUMMIT ONLINE

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

アジェンダ

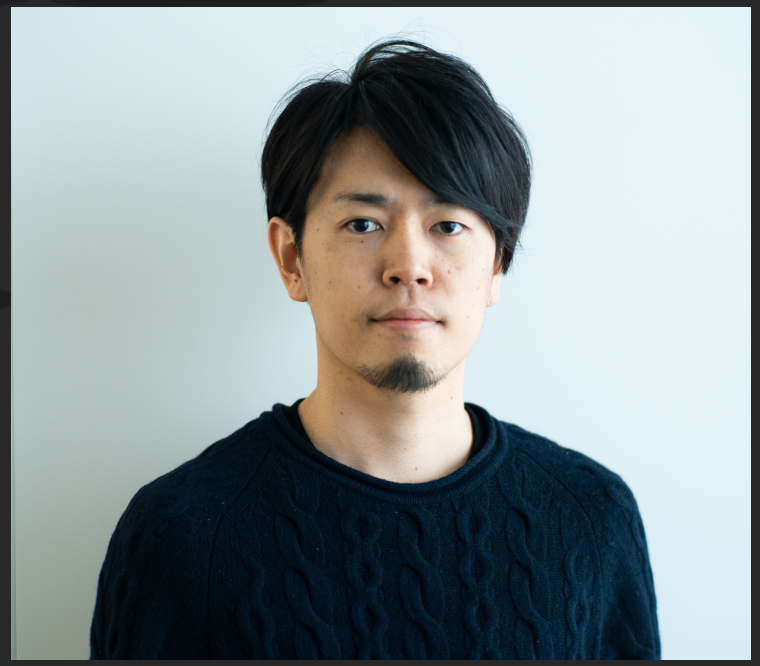
- 自己紹介/会社概要
- アーキテクチャの紹介とマネージドサービスの活用
- アーキテクチャの性能検証と事例共有
- マネージドサービス導入の利点と注意点
- まとめ

Copyright © Sumzap, Inc. All Rights Reserved.

自己紹介/会社概要

Copyright © Sumzap, Inc. All Rights Reserved.

自己紹介



株式会社サムザップ
SRE チーム
エンジニア
信田 悟至

2014年4月 サイバーエージェントに入社し、同年にサムザップに出向

運用中のプロダクトや新規開発チームのサーバサイドエンジニアを経て、2018年に SRE チームに参加

現在は 「この素晴らしい世界に祝福を！ファンタスティックデイズ」 のシステム設計、構築、運用を担当

Copyright © Sumzap, Inc. All Rights Reserved.

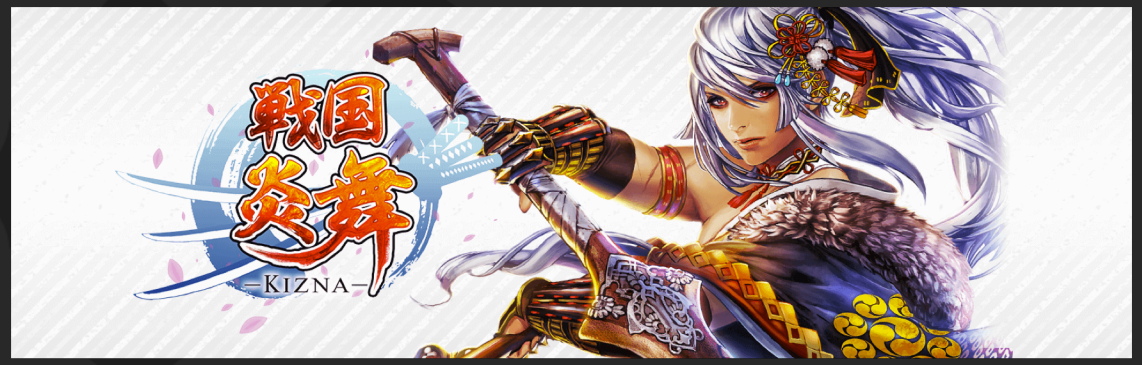
会社概要

- 株式会社サムザップ
  - 2009年5月 CyberAgent 子会社として設立

- 事業内容
  - スマートフォン向けゲームアプリの企画、運営、配信

Copyright © Sumzap, Inc. All Rights Reserved.

提供サービス



戦国炎舞 – KIZNA –

2013年4月配信開始
戦国時代を舞台にしたカードバトルゲーム
1日3回 20人対20人のリアルタイムバトル 「合戦」 が行われる

©2019 暁なつめ・三嶋くろね／KADOKAWA／映画このすば製作委員会　©Sumzap, Inc.

この素晴らしい世界に祝福を！ ファンタスティックデイズ

2020年2月配信開始
ラノベやアニメで大人気の 『このすば』 初のスマホゲーム
ゲームオリジナルのキャラクターやストーリーも登場

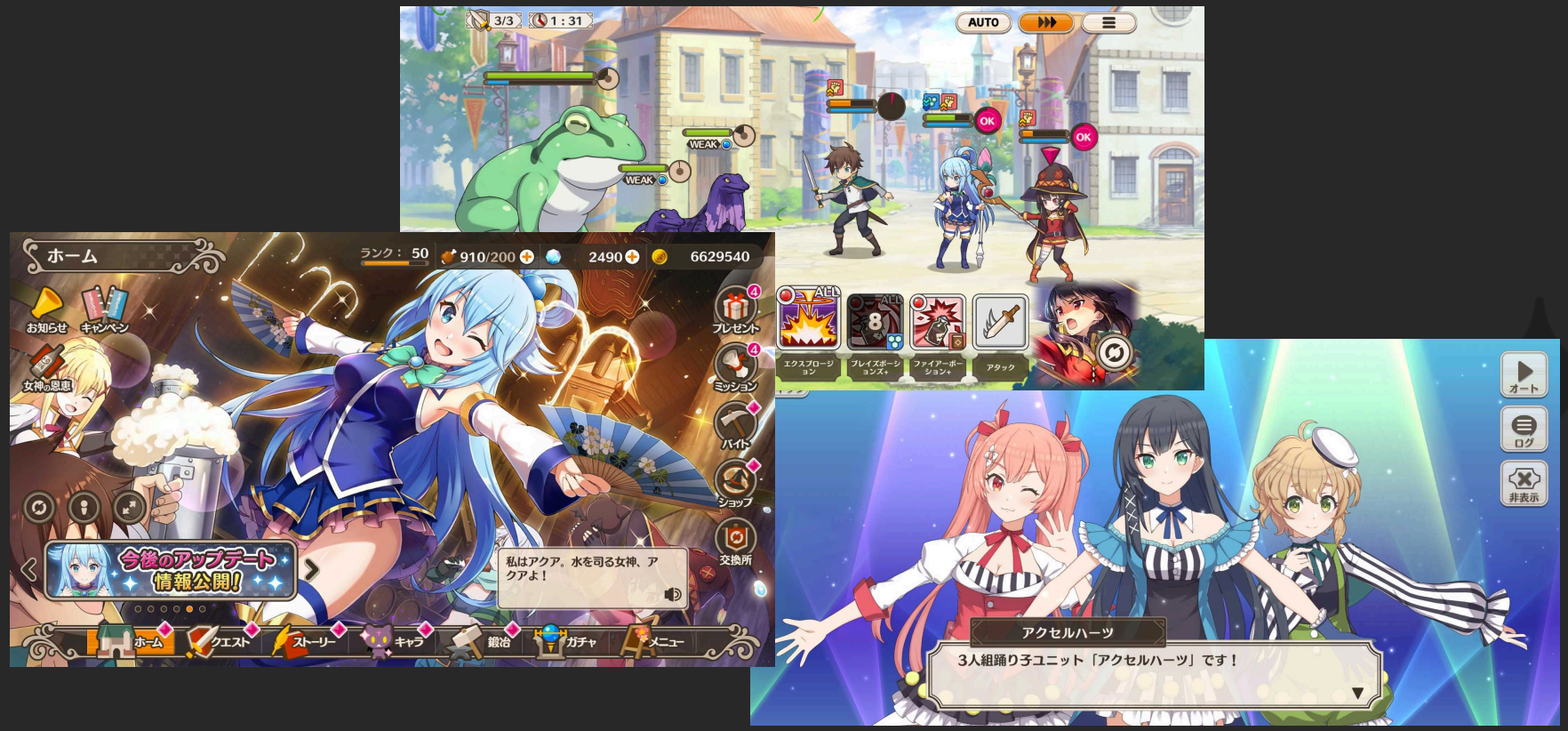
Copyright © Sumzap, Inc. All Rights Reserved.

『この素晴らしい世界に祝福を！ファンタスティックデイズ』

『この素晴らしい世界に祝福を！』初のスマホゲーム（通称『このファン』）

個性的なキャラクターが繰り広げるストーリーをフルボイスで楽しめる

オリジナルキャラクターや迫力のあるバトルアニメーションも魅力的



©2019 暁なつめ・三嶋くろね／KADOKAWA／映画このすば製作委員会　©Sumzap, Inc.

Copyright © Sumzap, Inc. All Rights Reserved.

アーキテクチャの紹介と
マネージドサービスの活用

Copyright © Sumzap, Inc. All Rights Reserved.

アーキテクチャ設計のコンセプト

- コンテナを使用したイミュータブルインフラストラクチャを構築
- 高可用性を目指すためにマルチ AZ のシステムを組む
- 管理コストを削減できるようにマネージドサービスに頼る

Copyright © Sumzap, Inc. All Rights Reserved.

アーキテクチャ設計のコンセプト

- コンテナを使用したイミュータブルインフラストラクチャを構築
- 高可用性を目指すためにマルチ AZ のシステムを組む
- 管理コストを削減できるようにマネージドサービスに頼る

Copyright © Sumzap, Inc. All Rights Reserved.

主に AWS のサービスを組み合わせて設計

Q. なぜ AWS を選んだのか？

A. MySQL と互換性があり、多機能かつ可用性の高い DB マネージドサービスである Aurora が存在することが決め手
自動のフェイルオーバーやバックアップが可能

Copyright © Sumzap, Inc. All Rights Reserved.

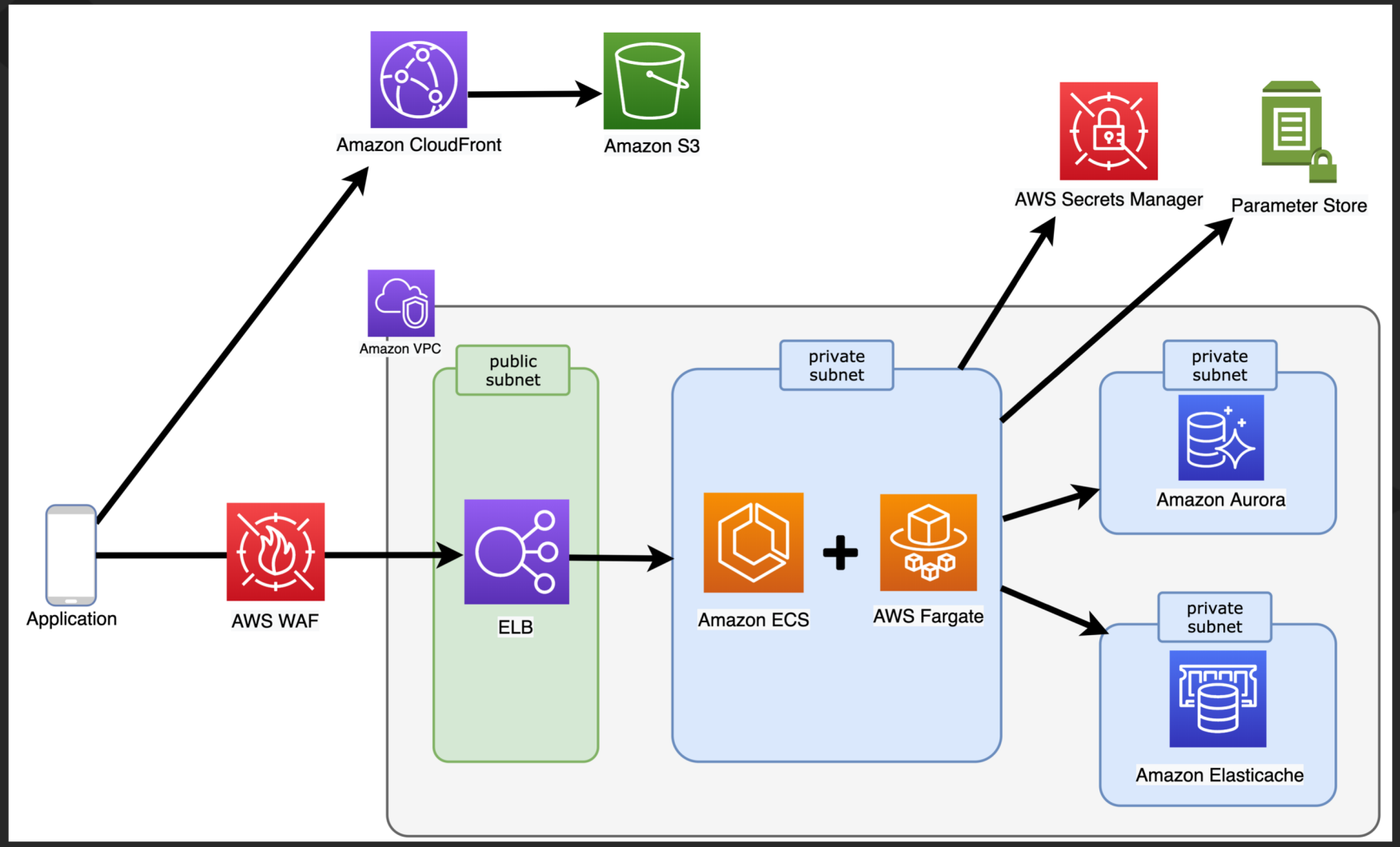
全体構成

- AWS のアカウントは本番、ステージング、検証用の 3 つのアカウントを使い分け
  - 本番ではリソースや費用管理を独立させたい
  - それぞれのアカウントは VPC Peering により一部相互通信を可能とさせている

- 構成管理は HashiCorp 社の Terraform を使用

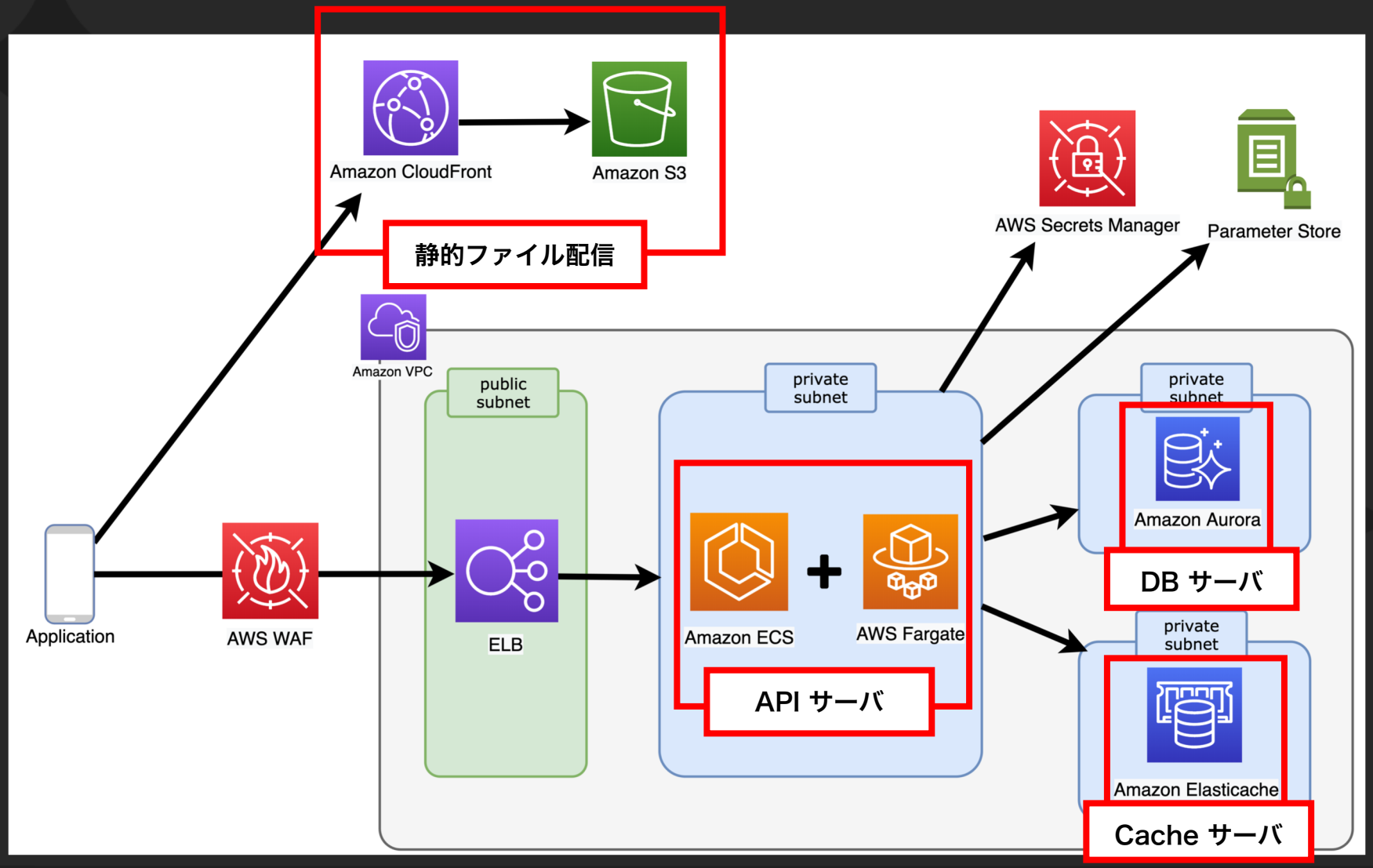
Copyright © Sumzap, Inc. All Rights Reserved.

基本構成



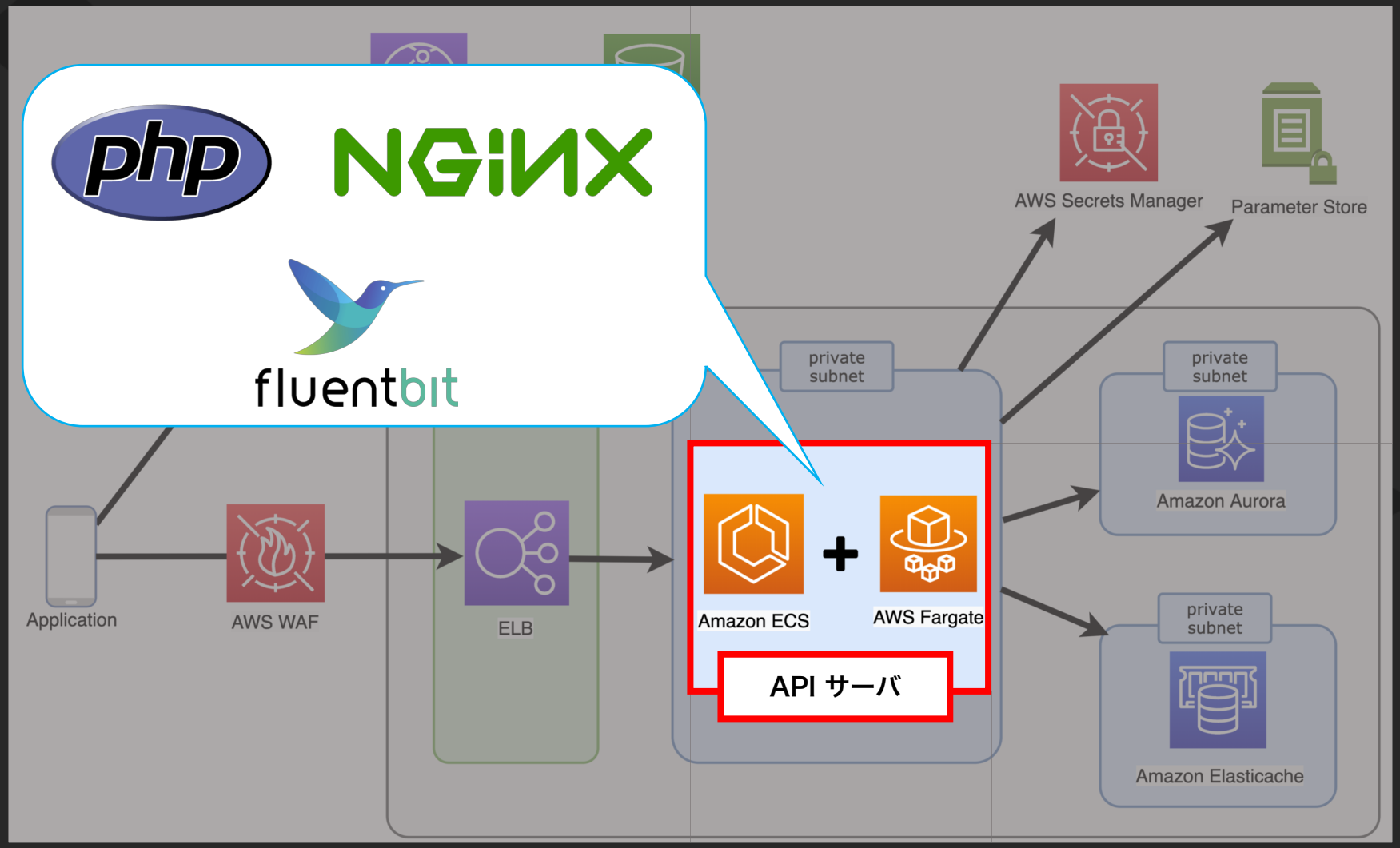
Copyright © Sumzap, Inc. All Rights Reserved.

基本構成



Copyright © Sumzap, Inc. All Rights Reserved.

基本構成



Copyright © Sumzap, Inc. All Rights Reserved.

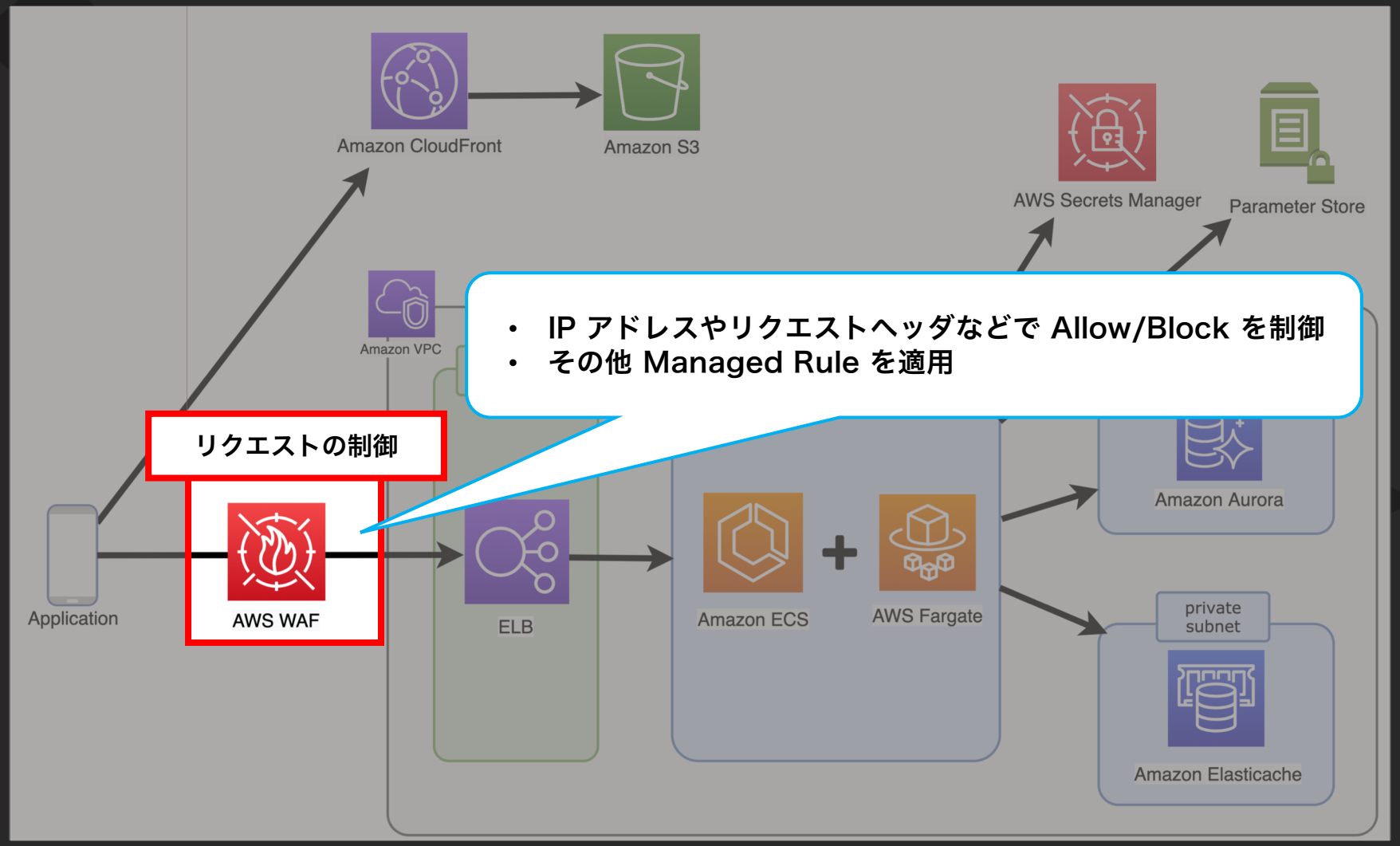
基本構成

【 ECS on Fargate を選んだ理由 】

- 運用上のインスタンス管理コストをできるだけ削減したかった
  - スケールの対象を ECS のタスクのみにすることで、運用上の変更管理箇所を最小限に
  - ゲーム系サービスは特にリクエスト量が大きく変動する可能性が高いので、スケールの観点を簡略化できることは大きい
- 2019年1月に Fargate の料金が大幅値下げした
  - on Amazon EC2 に比べてまだ 2 ~ 5 割ほど高くなるが、それでもマネージドに頼ることで
    運用上の管理コストを減らせる方が大きいと判断
- Amazon EKS は、アーキテクチャ設計当時は Fargate での稼働が
  なかったことと、Kubernetes の学習コストを考慮して、選択肢から除外

Copyright © Sumzap, Inc. All Rights Reserved.

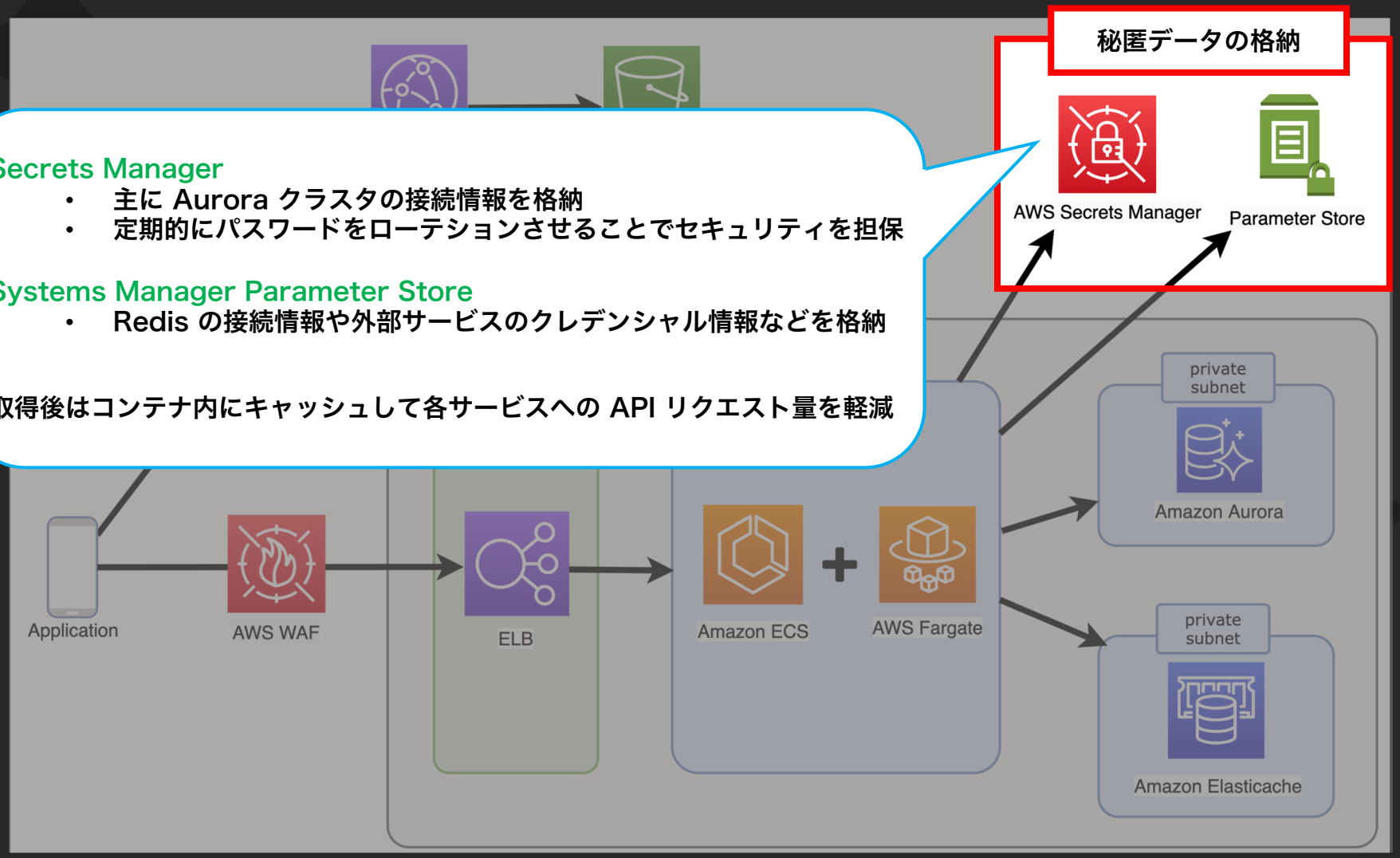
基本構成



- IP アドレスやリクエストヘッダなどで Allow/Block を制御
- その他 Managed Rule を適用

Copyright © Sumzap, Inc. All Rights Reserved.

基本構成



Secrets Manager

- 主に Aurora クラスタの接続情報を格納
- 定期的にパスワードをローテションさせることでセキュリティを担保

Systems Manager Parameter Store

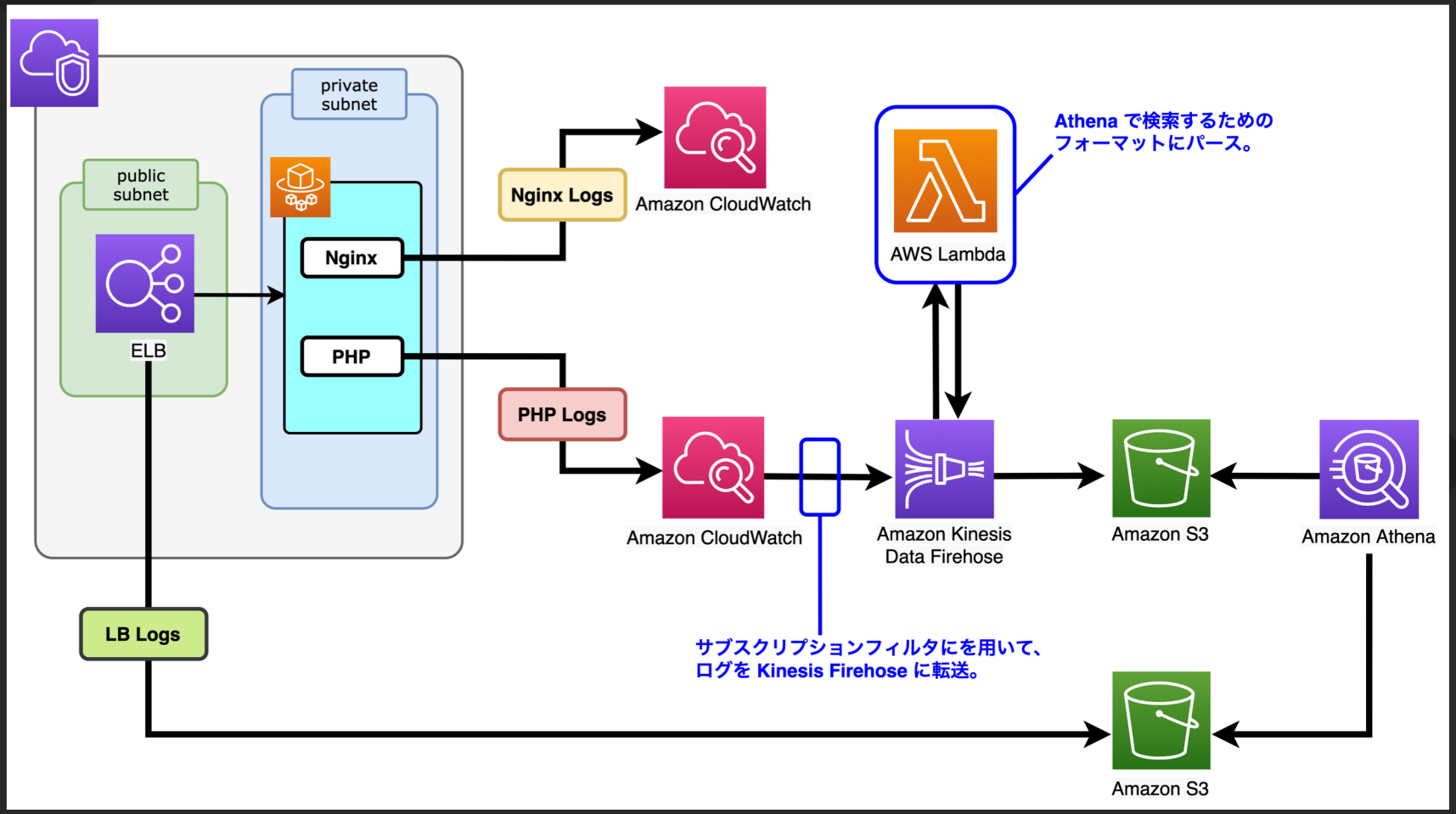
- Redis の接続情報や外部サービスのクレデンシャル情報などを格納

取得後はコンテナ内にキャッシュして各サービスへの API リクエスト量を軽減

Copyright © Sumzap, Inc. All Rights Reserved.

ロギング構成

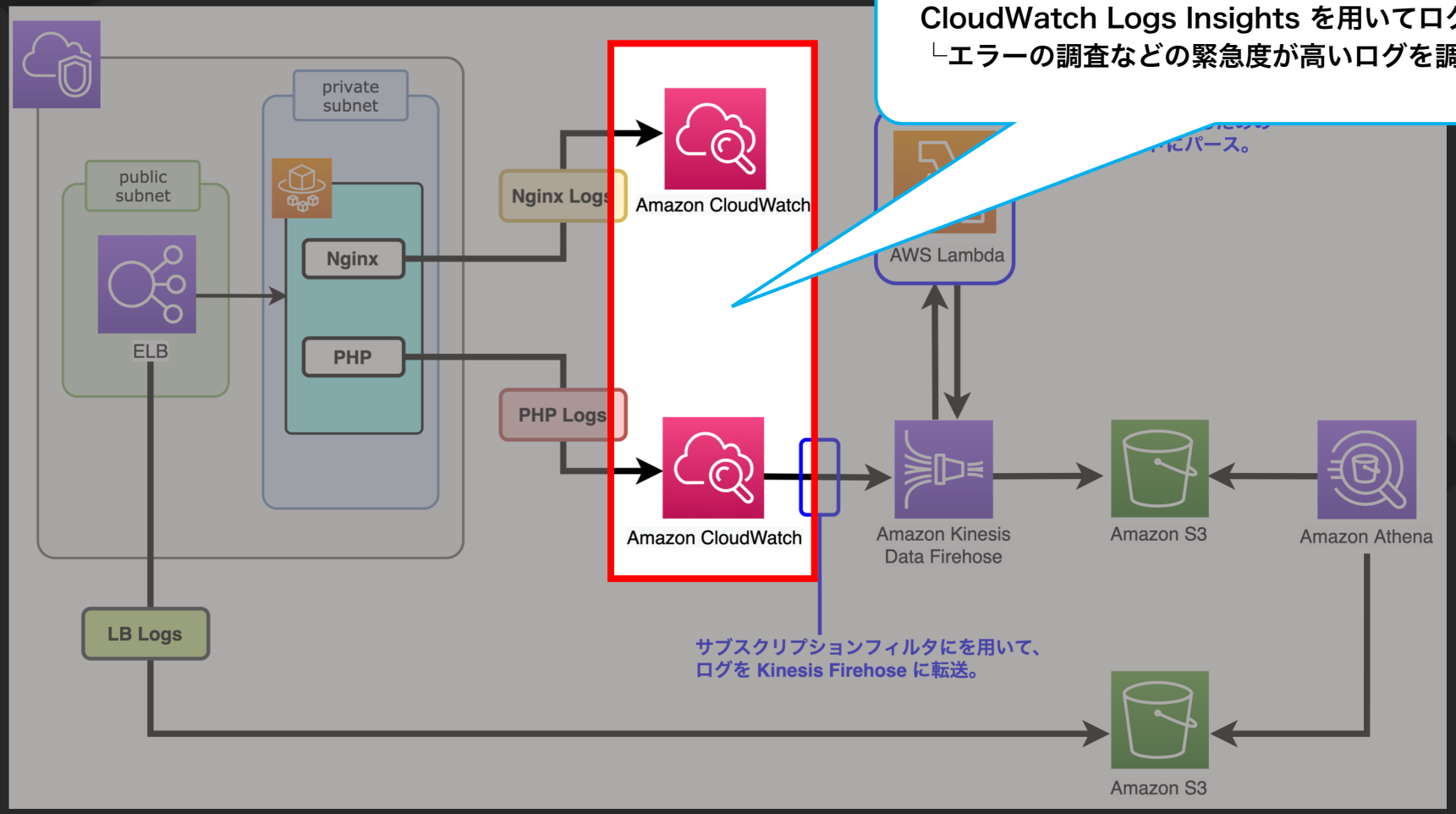
【 リリース時 】



Copyright © Sumzap, Inc. All Rights Reserved.

ロギング構成

【 リリース時 】

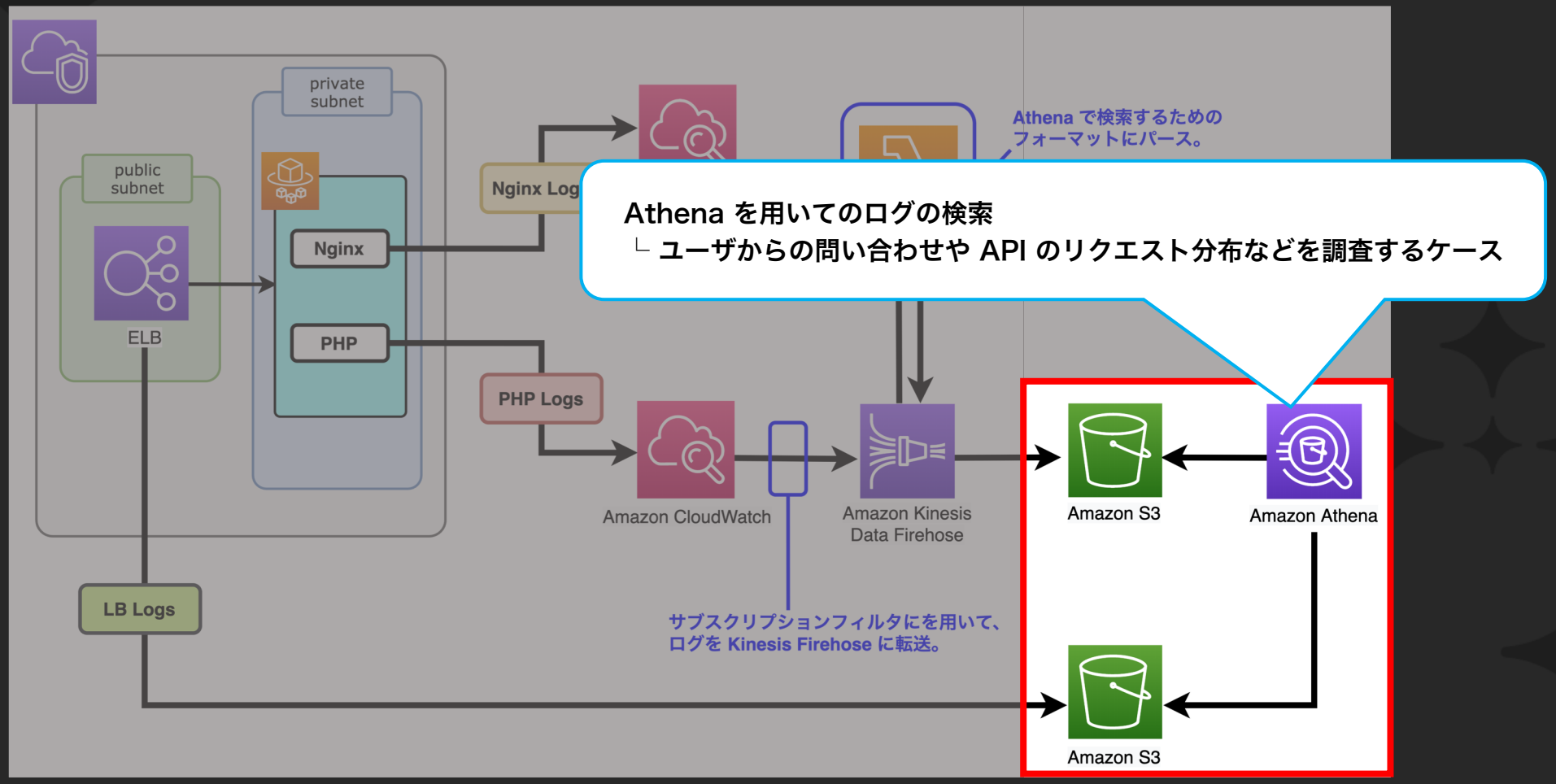


CloudWatch Logs Insights を用いてログの検索
└エラーの調査などの緊急度が高いログを調査するケース

Copyright © Sumzap, Inc. All Rights Reserved.

ロギング構成

【 リリース時 】



Copyright © Sumzap, Inc. All Rights Reserved.

ロギング構成

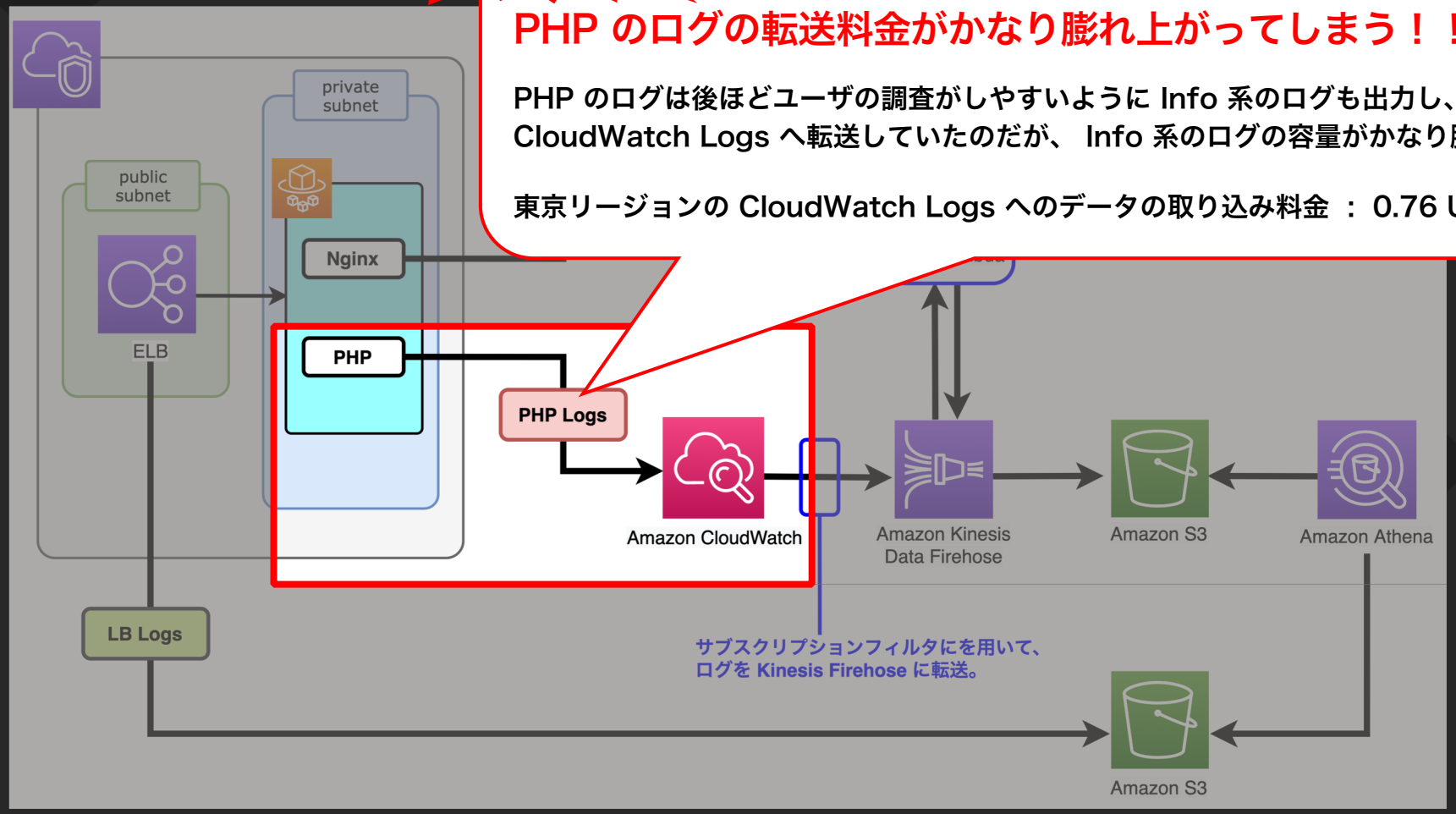
【 リリース時 】

大問題

PHP のログの転送料金がかなり膨れ上がってしまう！！！

PHP のログは後ほどユーザの調査がしやすいように Info 系のログも出力し、
CloudWatch Logs へ転送していたのだが、 Info 系のログの容量がかなり膨大！

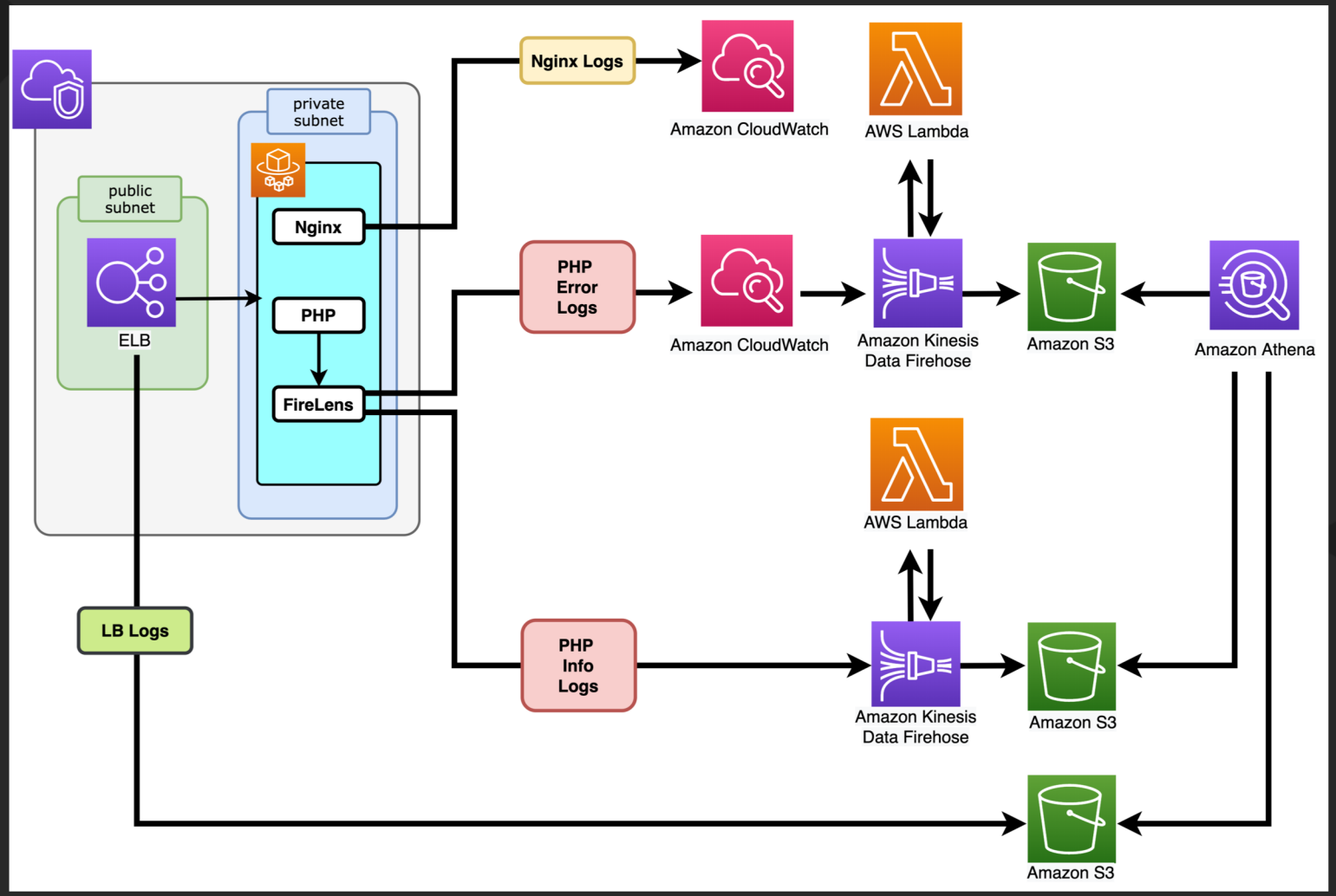
東京リージョンの CloudWatch Logs へのデータの取り込み料金 ： 0.76 USD/GB



Copyright © Sumzap, Inc. All Rights Reserved.

ロギング構成

【 改修後 】

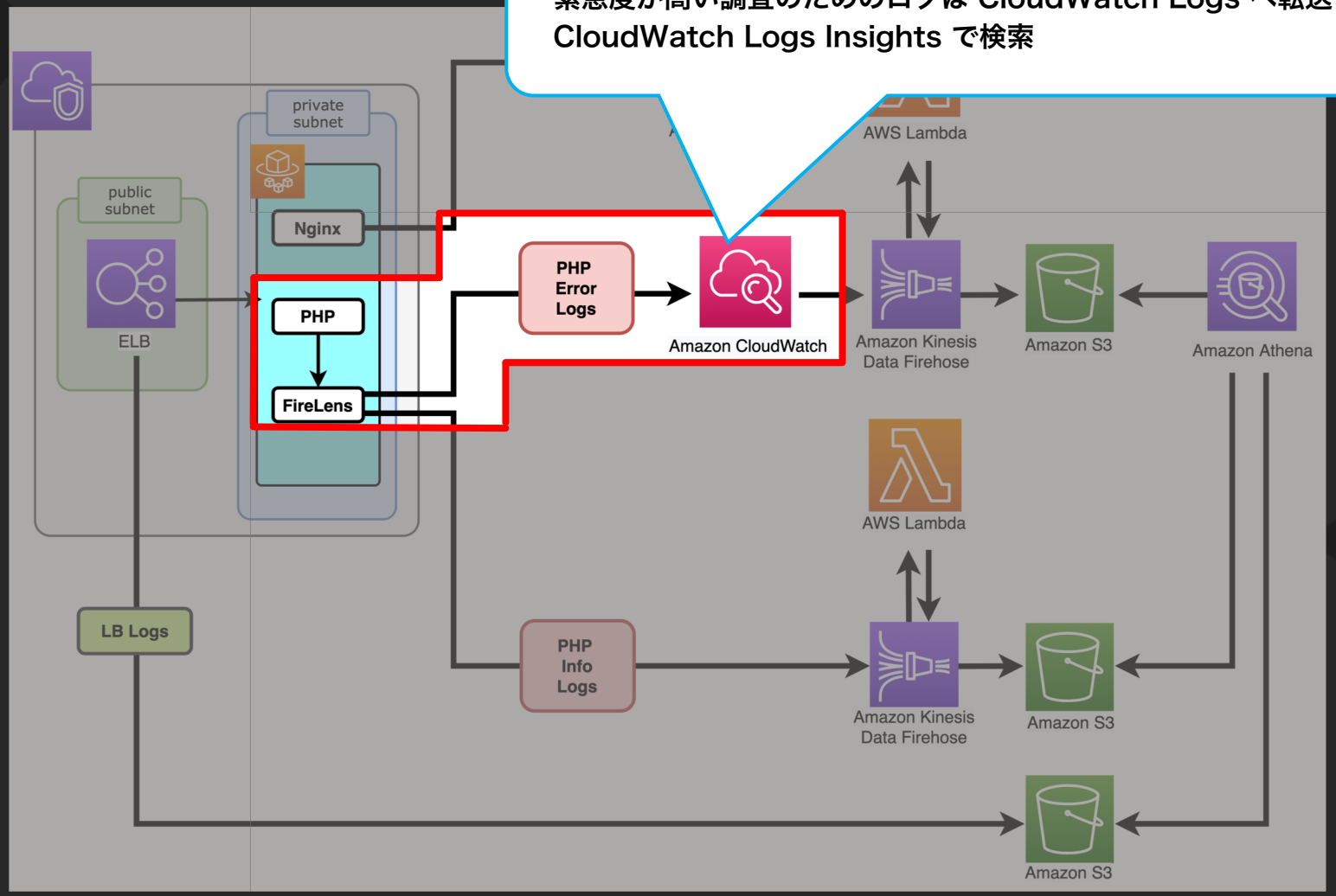


Copyright © Sumzap, Inc. All Rights Reserved.

ロギング構成

【 改修後 】

緊急度が高い調査のためのログは CloudWatch Logs へ転送し、
CloudWatch Logs Insights で検索



Copyright © Sumzap, Inc. All Rights Reserved.

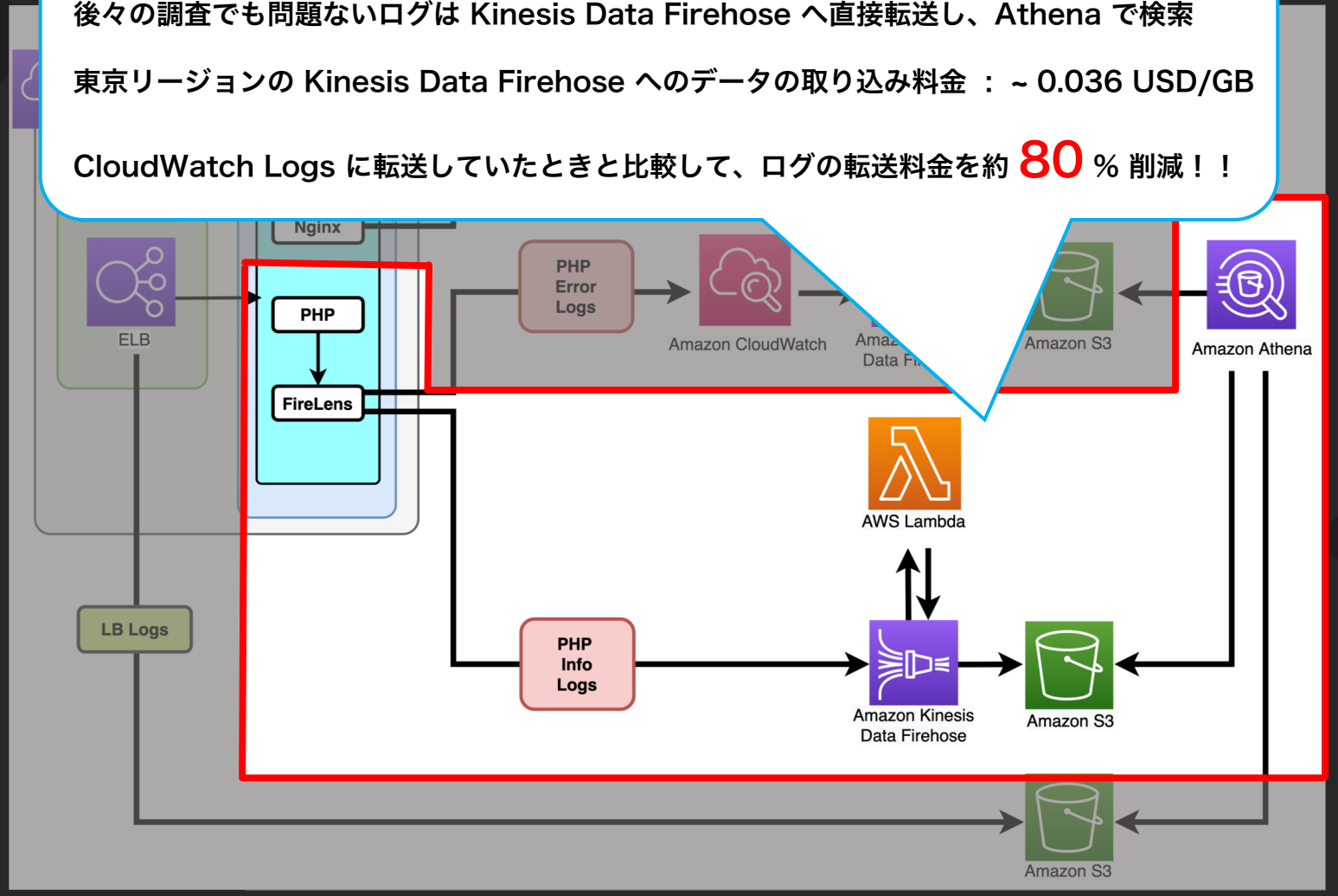
ロギング構成

【 改修後 】

後々の調査でも問題ないログは Kinesis Data Firehose へ直接転送し、Athena で検索

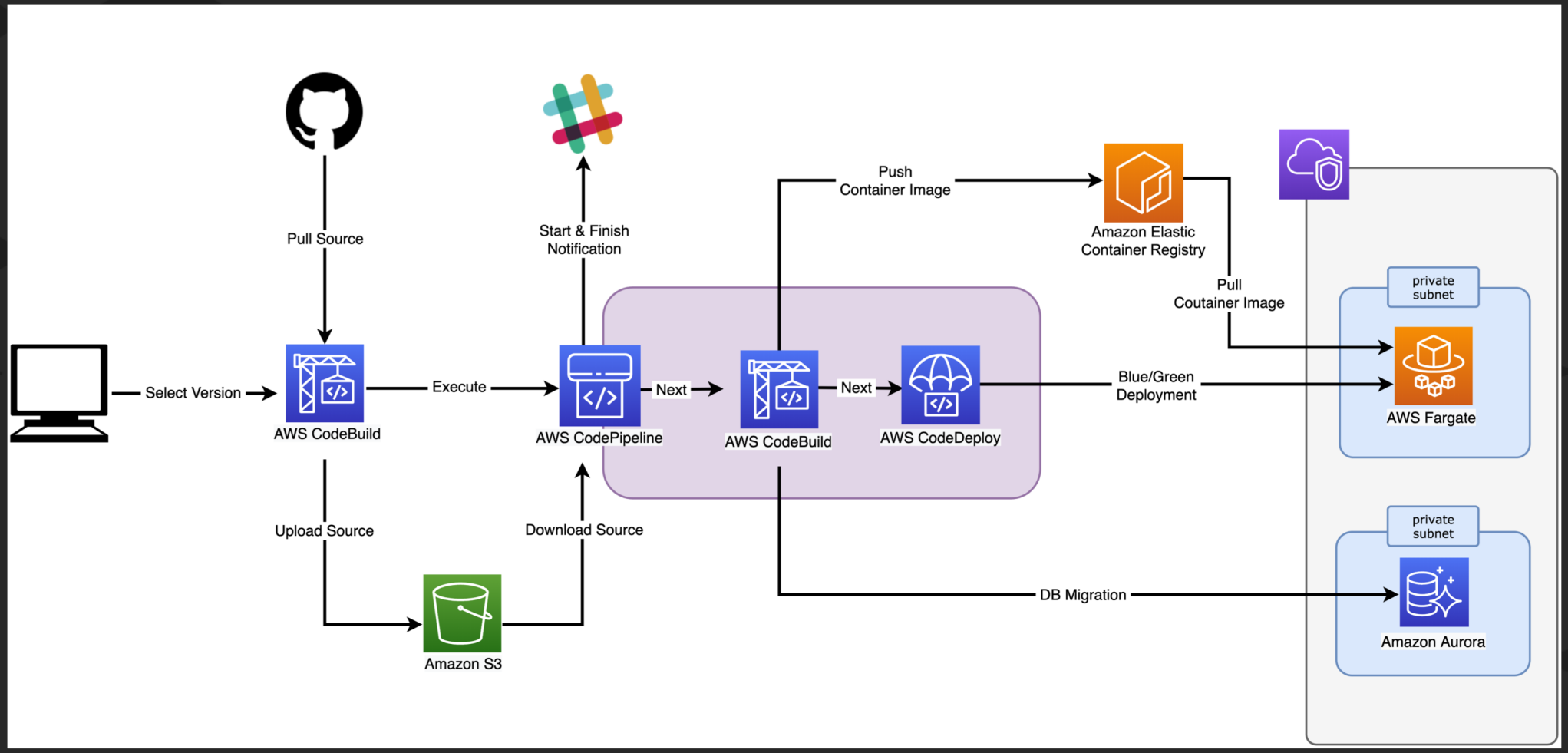
東京リージョンの Kinesis Data Firehose へのデータの取り込み料金 ： ~ 0.036 USD/GB

CloudWatch Logs に転送していたときと比較して、ログの転送料金を約 80 % 削減！！



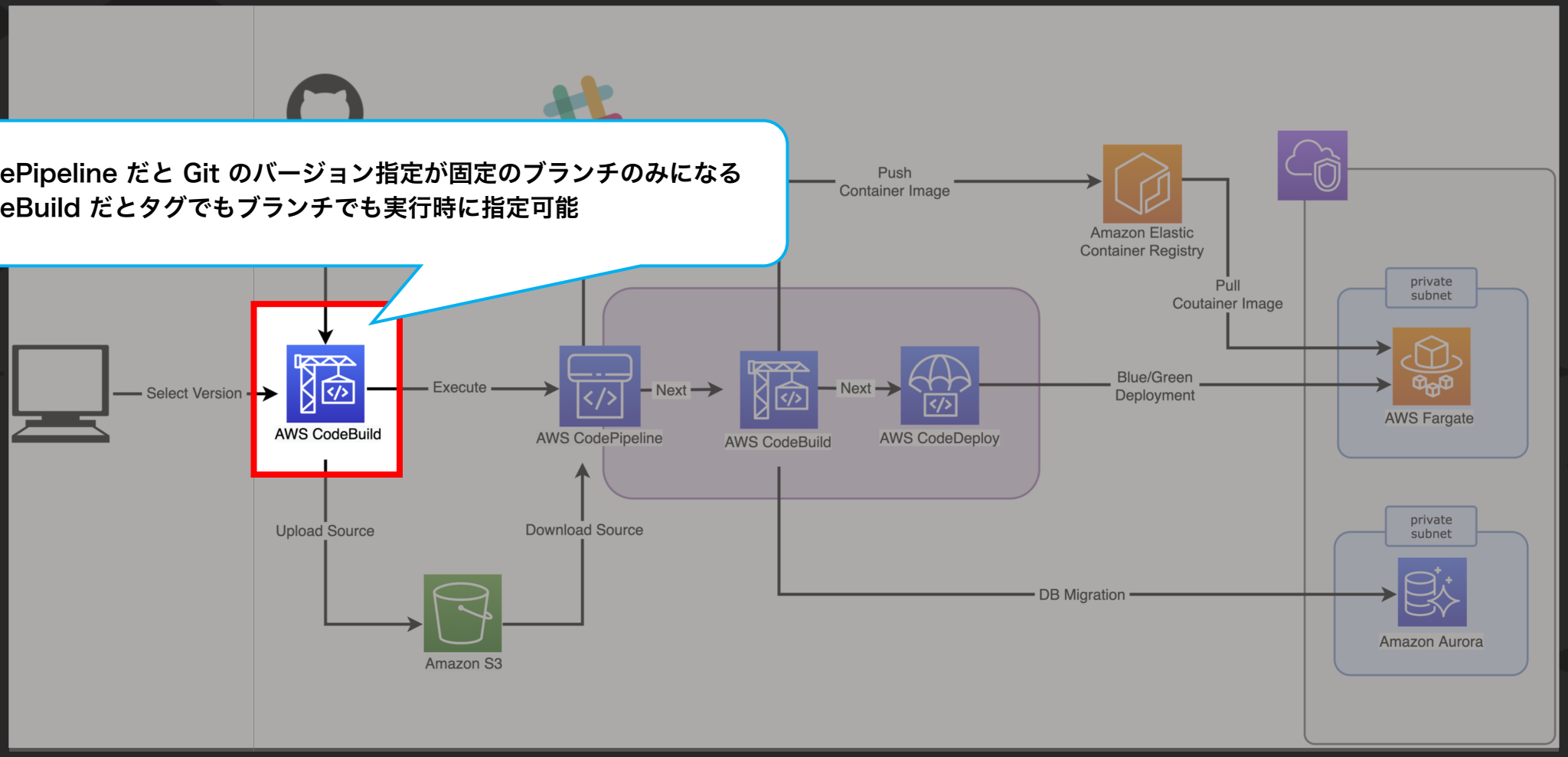
Copyright © Sumzap, Inc. All Rights Reserved.

デプロイ構成



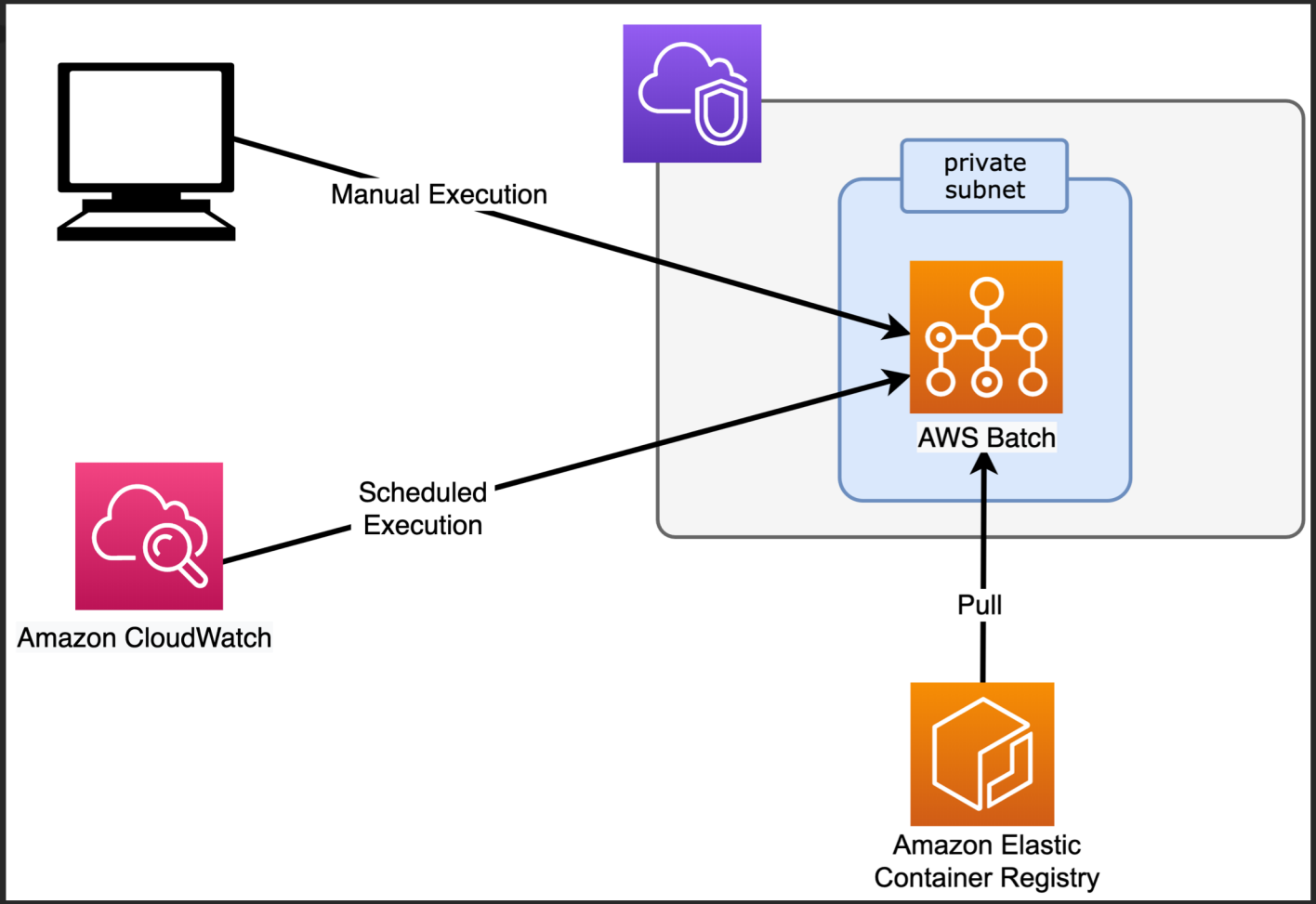
Copyright © Sumzap, Inc. All Rights Reserved.

デプロイ構成



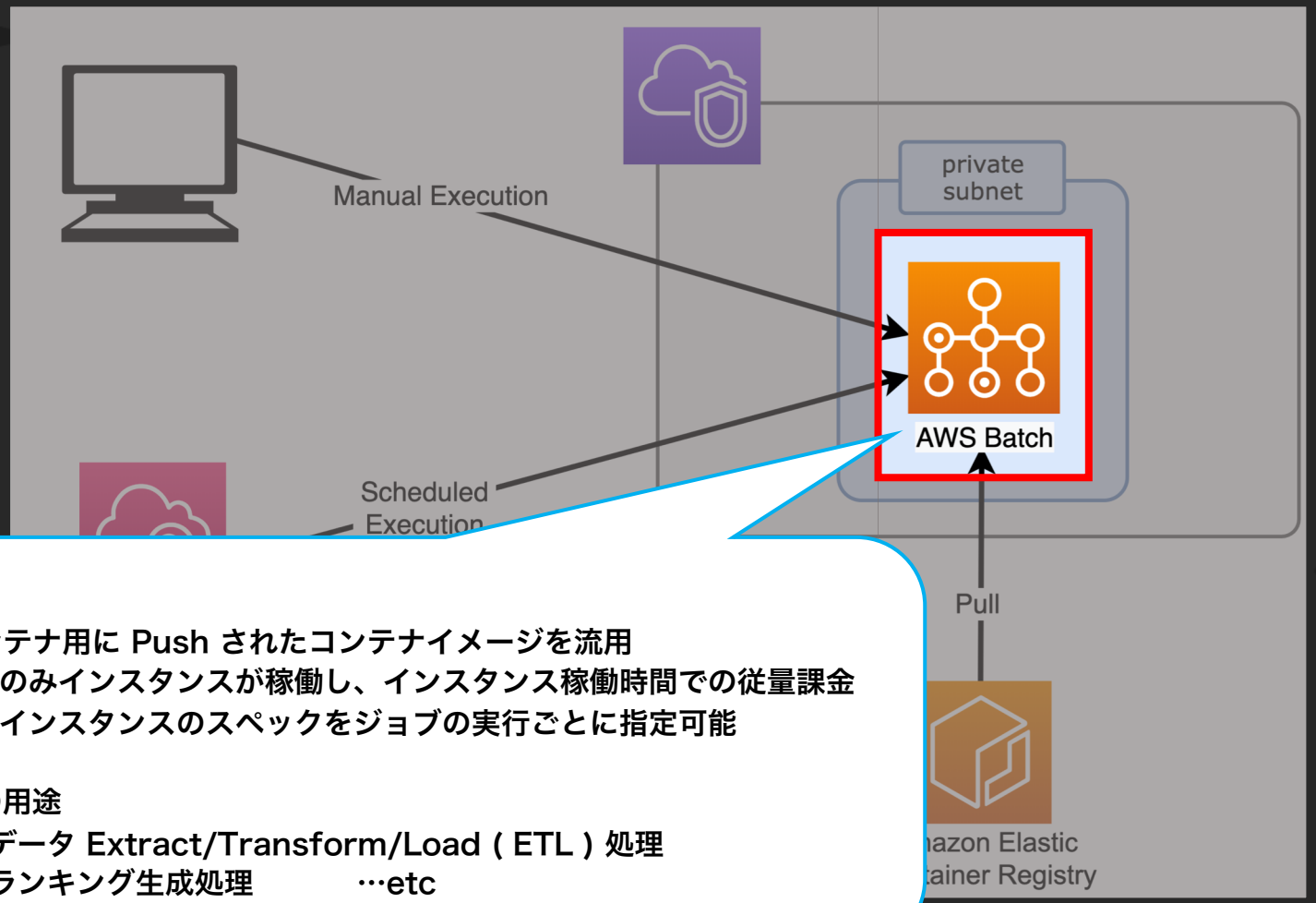
Copyright © Sumzap, Inc. All Rights Reserved.

バッチ構成



Copyright © Sumzap, Inc. All Rights Reserved.

バッチ構成



特徴

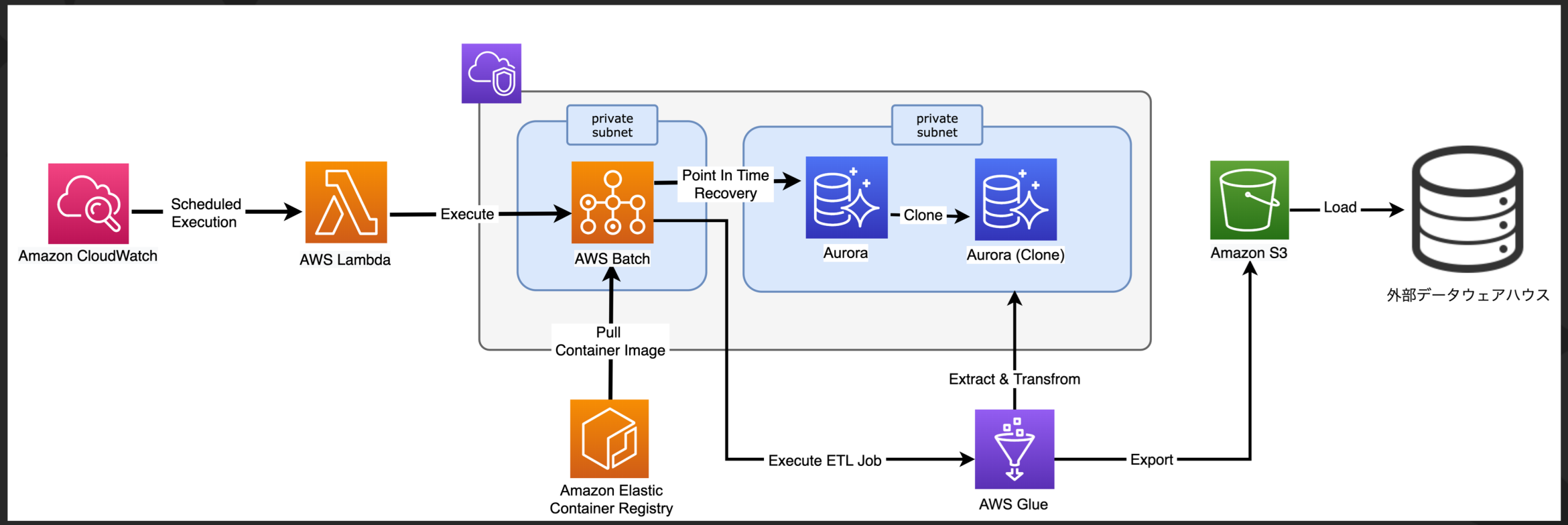
- API コンテナ用に Push されたコンテナイメージを流用
- 実行時にのみインスタンスが稼働し、インスタンス稼働時間での従量課金
- 稼働するインスタンスのスペックをジョブの実行ごとに指定可能

主なバッチの用途

- 分析用データ Extract/Transform/Load (ETL) 処理
- ユーザランキング生成処理　　…etc

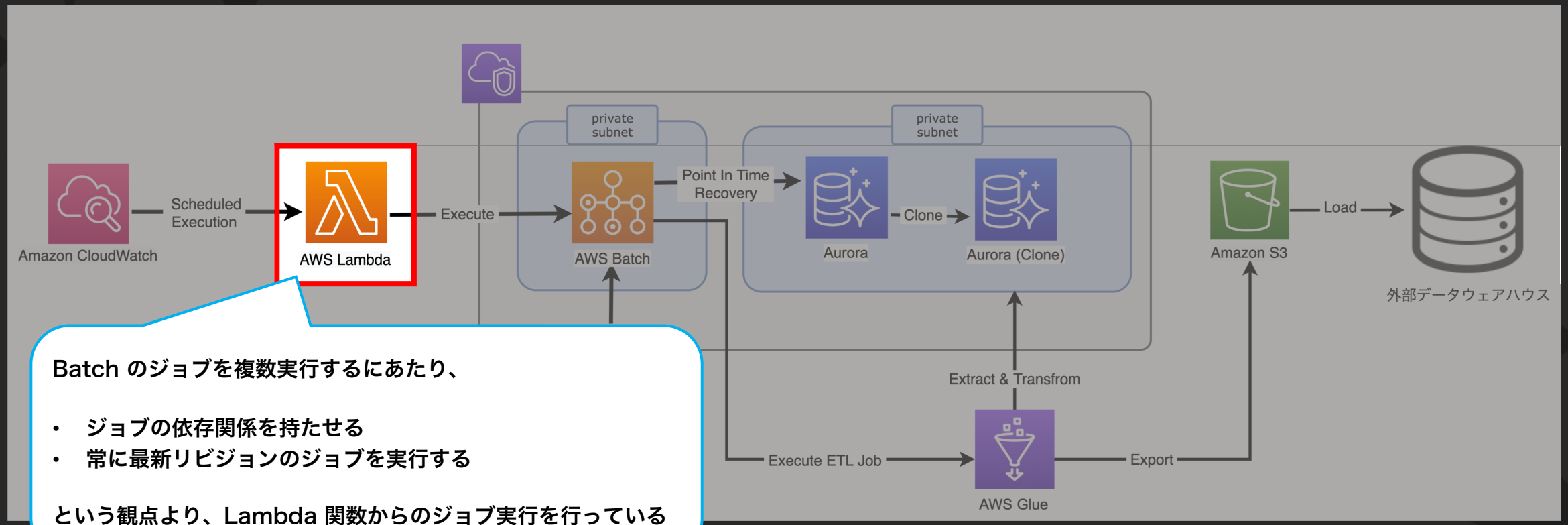
Copyright © Sumzap, Inc. All Rights Reserved.

分析用データ ETL 処理構成



Copyright © Sumzap, Inc. All Rights Reserved.

分析用データ ETL 処理構成



Batch のジョブを複数実行するにあたり、

- ジョブの依存関係を持たせる
- 常に最新リビジョンのジョブを実行する

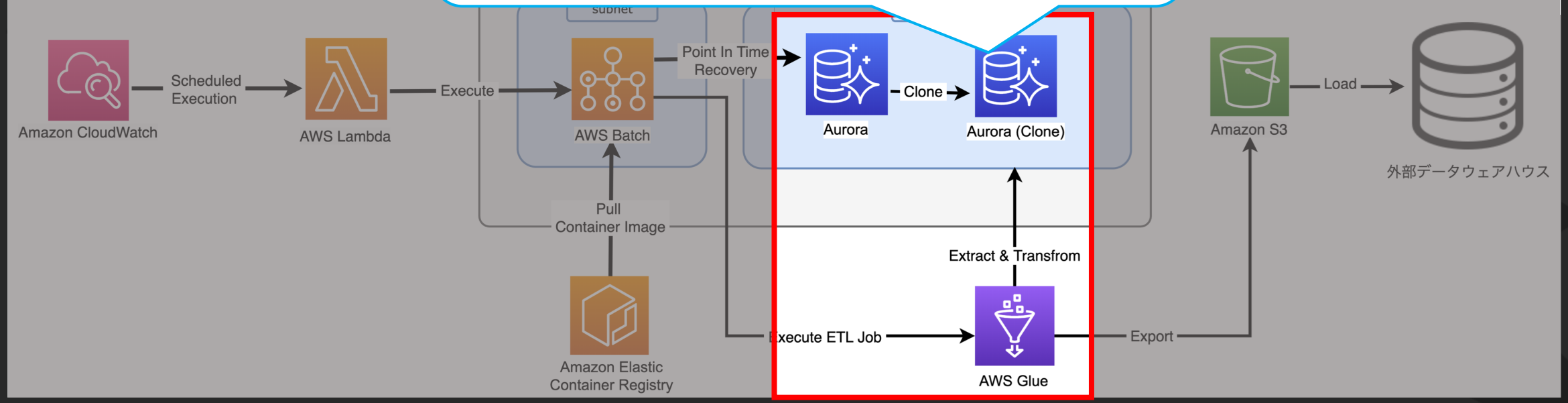
という観点より、Lambda 関数からのジョブ実行を行っている

Copyright © Sumzap, Inc. All Rights Reserved.

分析用データ ETL 処理構成

- 本番稼働のクラスタに影響を与えない
- 仮にジョブが失敗したとしてもデータの同一性の担保が可能

という観点より、 Aurora クローンを作成して ETL 処理を実行している



Copyright © Sumzap, Inc. All Rights Reserved.

その他

【 DNS 】
- Route53

【 SSL 証明書 】
- AWS Certificate Manager

【 監視・アラート通知 】
- CloudWatch Alarm
- Amazon SNS
- AWS Chatbot

【 ユーザ問い合わせ用 API 】
- API Gateway
- Step Functions
- Lambda

【 その他分析系情報送信 】
- SQS
- Lambda
- DynamoDB

Copyright © Sumzap, Inc. All Rights Reserved.

アーキテクチャの
性能検証と事例共有

Copyright © Sumzap, Inc. All Rights Reserved.

性能検証（負荷試験編）

【 準備 】

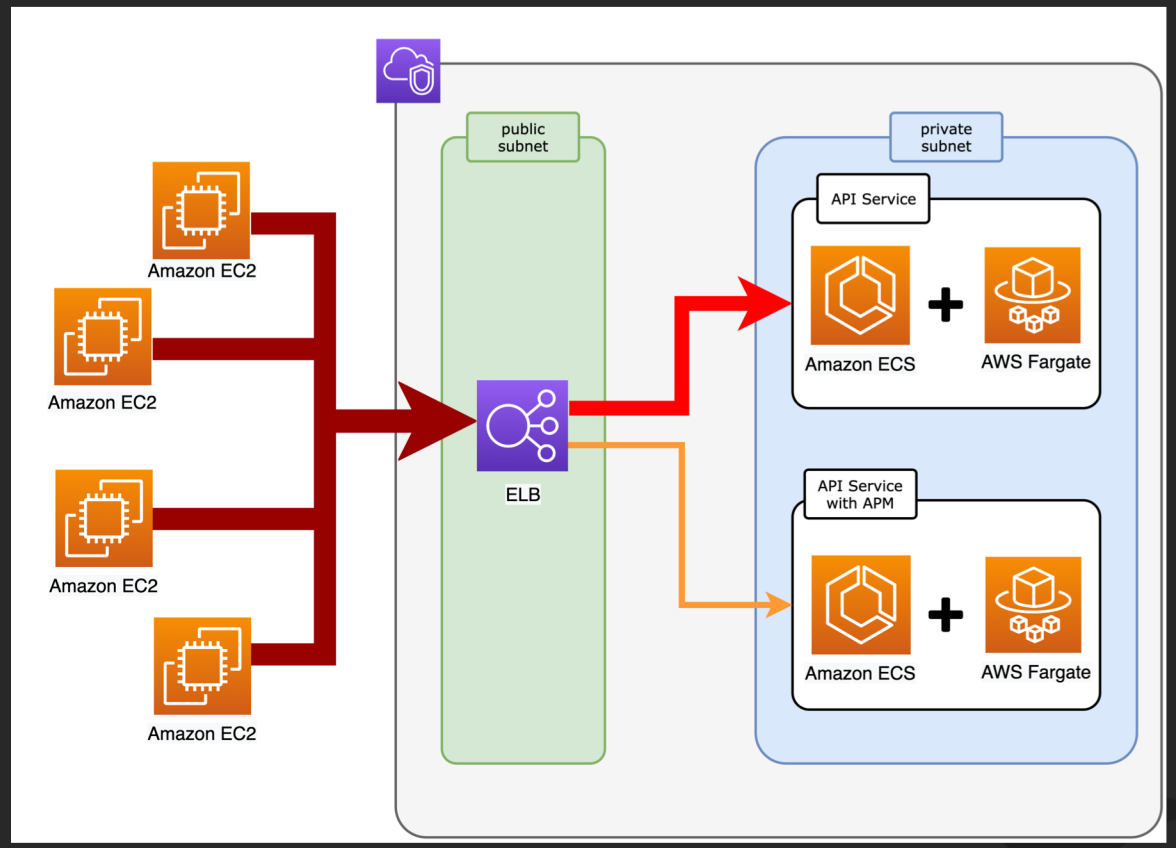
- 負荷試験におけるゴールの明確化
- 負荷試験用の環境は、本番で想定するスペックや台数と同等のものを準備
- 試験シナリオの作成
- 攻撃サーバは、負荷試験環境へのリクエストを確実に public な通信になるように、負荷試験環境とは別の VPC に構築
- 負荷試験におけるインフラ費用の見積もりと報告
- 負荷をかけているときのログやメトリクスが確認しやすいようにダッシュボード化

Copyright © Sumzap, Inc. All Rights Reserved.

性能検証（負荷試験編）

【 準備 】

- Application Performance Management (APM) には New Relic を使用
- ECS サービスを別途作成し、ALB でリクエストを一部振り分けるように調整
  → APM のコスト面を考慮



Copyright © Sumzap, Inc. All Rights Reserved.

性能検証（負荷試験編）

【 躓いたところと対処法の例① 】

Fargate で立ち上がる ECS タスクが起動ごとにパフォーマンスのばらつきがあった

- コンテナが稼働するためのホストマシンのスペックが同一サイズのタスクでも起動ごとに違う可能性がある
- 1 タスクあたりのおおよそのリクエスト許容限度を把握し、想定負荷に対して余裕を持ってタスクを用意する

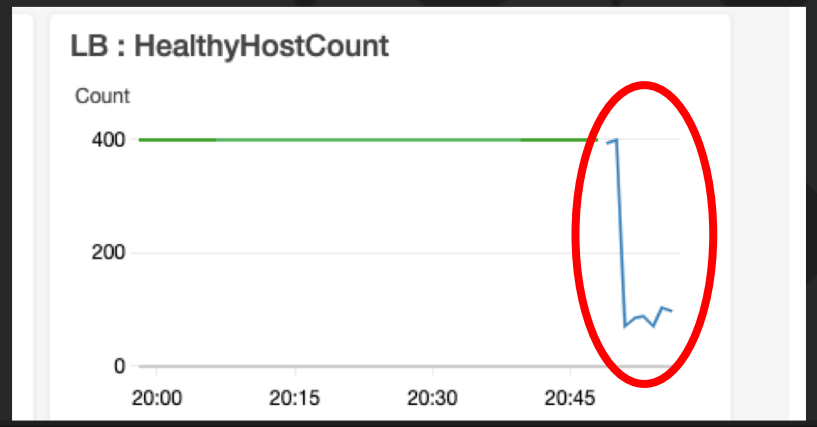
Copyright © Sumzap, Inc. All Rights Reserved.

性能検証（負荷試験編）

【 躓いたところと対処法の例② 】

ECS の数タスクが過負荷により落ち始めると、なし崩し的に他のタスクも落ちていく

- PHP コンテナへの瞬間的な高負荷を緩和させるために、 Nginx の設定に流量制限を設けることで対処
- 流量制限で引っかかった際の HTTP ステータスコードを ALB のヘルスチェックの正常コードに登録しておく必要あり



Copyright © Sumzap, Inc. All Rights Reserved.

性能検証（負荷試験編）

【 リリース直後の稼働リソース 】

- ECS on Fargate
  - vCPU : 4
  - Memory : 8GB
- Aurora
  - db.r5.4xlarge * 18 インスタンス（ 9 クラスタ）
- Elasticache for Redis（クラスタモード Off ）
  - cache.r5.xlarge * 6ノード（ 3 クラスタ）

Copyright © Sumzap, Inc. All Rights Reserved.

性能検証（障害試験編）

【 準備 】

障害試験を開始するための
前提条件をリストアップ

No	内容
1	アプリケーションからAPIにリクエストを送り、意図するレスポンスが返ってくる
2	障害試験を行うための隔離された環境が用意されている
3	サーバソースのデプロイがいつでもできる
4	管理画面のデプロイがいつでもできる
5	webページのデプロイがいつでもできる
6	LBのターゲットがHEALTH状態である
7	DBクラスタのフェイルオーバを実行することができる
8	サーバ稼働におけるログやメトリクスがいつでも閲覧できる
9	監視系統が全て稼働してる
10	アラート系統が全て稼働している
11	アプリケーションのビルドをいつでも行うことができる
12	アセットバンドルをいつでも適切なものに更新することができる
13	コードフリーズの期日が決まっている
14	サーバにSSHログインして調査することができる
15	試験で発覚したバグを修正するためのフローや期間が確立されている
16	障害試験のためのアプリがインストールされた端末が用意できている

Copyright © Sumzap, Inc. All Rights Reserved.

性能検証（障害試験編）

【 準備 】

実際に障害を起こすリソースと起こし方、及び起こした際の想定挙動をリストアップ

例.

- 対象リソース
  - Aurora クラスタ
- 障害の起こし方
  - 一定数のリクエストを送信している間にフェイルオーバーの実行
- 障害時の想定挙動
  - reader インスタンスが昇格するまではエラーが発生するが、昇格後は正常挙動

Copyright © Sumzap, Inc. All Rights Reserved.

性能検証（障害試験編）

【 準備 】

障害をこちらで意図的に起こせないサービスは、擬似的に障害が起こっている状態を再現する

例.

- 対象リソース
  - Amazon S3
- 障害の起こし方
  - 対象バケットにアクセスできないようにバケットポリシーで Block の設定を入れる
- 障害時の想定挙動
  - 対象バケットへの操作を必要とするリソース上でエラーが発生する

Copyright © Sumzap, Inc. All Rights Reserved.

性能検証（障害試験編）

【 躓いたところと対処法の例 】

Blue/Green Deployment 実行直後のリクエストでエラーが多く発生する

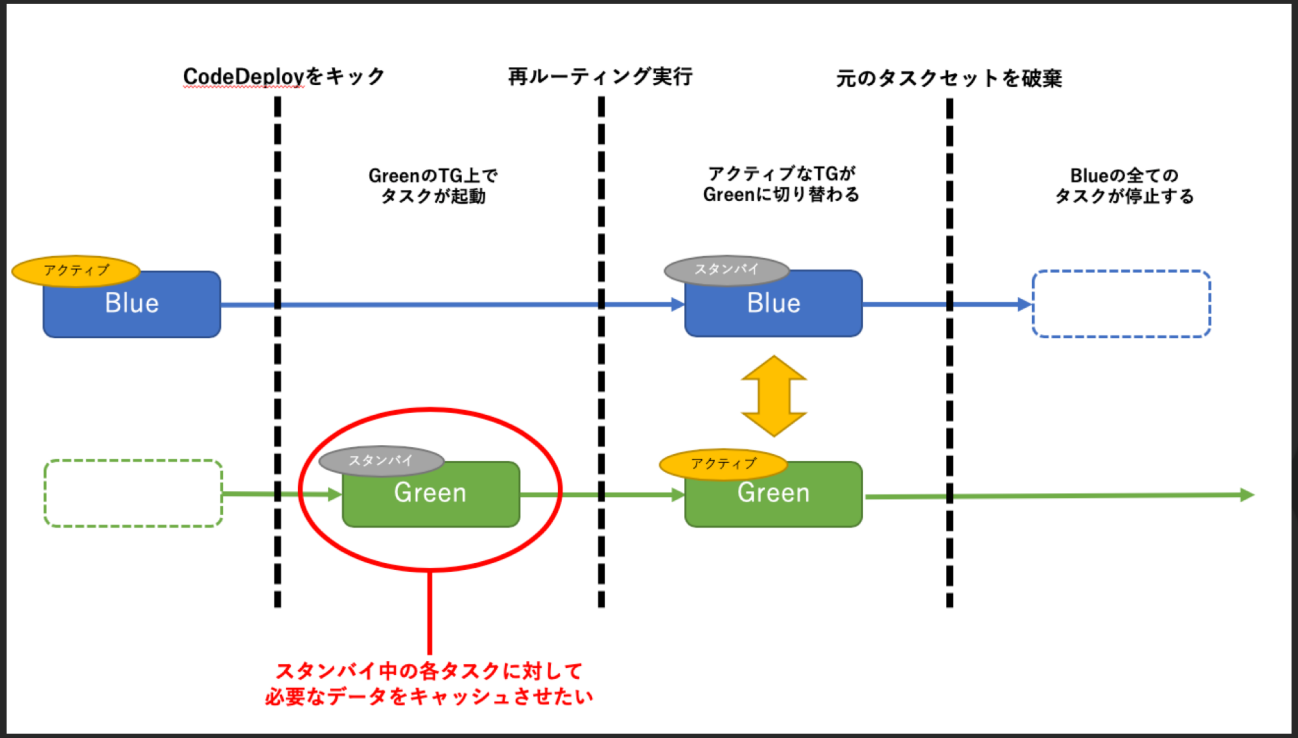
原因は、起動した ECS タスクが Parameter Store への各種データ取得を一斉に行うことによるAPI スループット上限の到達

Systems Manager Parameter Store のスループット上限はサービスクォータでの制限緩和不可

Copyright © Sumzap, Inc. All Rights Reserved.

性能検証（障害試験編）

【 躓いたところと対処法の例 】



ALB のヘルスチェックで使用している API の処理内でデータ取得を行うことで、
Blue/Green Deployment の再ルーティング前にデータ保持をさせるように改修

Copyright © Sumzap, Inc. All Rights Reserved.

マネージドサービス導入の利点と注意点

Copyright © Sumzap, Inc. All Rights Reserved.

マネージドサービスの利点

- 運用上の管理作業をかなり削減することができる
  - 特に Fargate、 CodeBuild、CodeDeploy、AWS Batch などの導入で今までのインスタンス運用から大きく効率化を図ることができた
- サーバエンジニアとインフラエンジニア間の共通認識の領域が増えた
  - マネージドサービスを使用した開発だと、両者共に各サービスの仕様を知っていることが前提となり、従来に比べて情報の共有がしやすくなった

Copyright © Sumzap, Inc. All Rights Reserved.

マネージドサービスの注意点

- 使用しているマネージドサービスに障害が発生した場合の自サービスへの影響範囲は必ず把握しておく
  - 事前に障害試験にて各リソースに対しての擬似的な障害シミュレーションの実施とそれに対する自サービスの影響をレポートとして残しておく
- マネージドサービスごとの障害検知を素早くキャッチアップできるようにしておく
  - 『このファン』では Personal Health Dashboard のイベントを Slack に通知している
- サポートにすぐに問い合わせができる体制を整えておく
  - 弊社はエンタープライズサポートプランにより、担当者との技術相談や緊急時の密な連携を行っている

Copyright © Sumzap, Inc. All Rights Reserved.

まとめ

Copyright © Sumzap, Inc. All Rights Reserved.

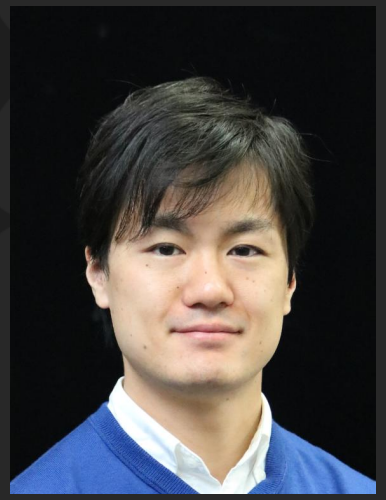
まとめ

- 『このファン』での AWS のマネージドサービスの活用事例の紹介
  - Fargate、 AWS Batch、 CodeDeploy など、インスタンスを意識しないアーキテクチャの構築
- アーキテクチャの性能検証
  - 負荷試験や障害試験の実施と、検証時の事例を一部紹介
- マネージドサービスにおける利点と注意点
  - 運用における管理は大きく低減できた
  - サービスごとに障害の検知と障害時の影響度を必ず把握する

Copyright © Sumzap, Inc. All Rights Reserved.

謝辞

リリース前の技術サポートや運用中の緊急時などで大変お世話になっております。



Taro Hirose

Ransui Iso

Kazunori Tsujimoto

Fumihide Nario

Yuji Tomita

Copyright © Sumzap, Inc. All Rights Reserved.

Thank you!

aws SUMMIT ONLINE

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.