LANSCOPE

大規模なデータを Amazon Kinesis Data Firehose の 動的パーティショニングと Amazon Athena を利用して、 ニアリアルタイムに処理する事例の紹介

エムオーテックス株式会社
開発本部 サービス開発１部 小沼 祐貴 辻 翼

自己紹介

小沼 祐貴(おぬま ゆうき)

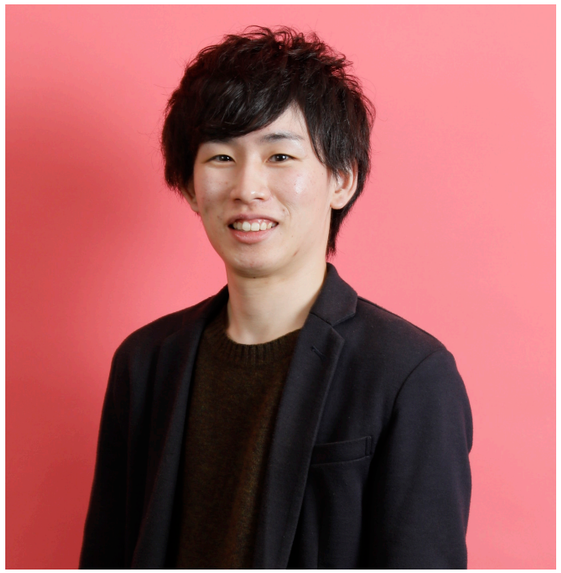
現在　　：エムオーテックス株式会社 2015年入社
　　LANSCOPE エンドポイントマネージャー クラウド版 アプリ開発チーム

経歴　　：LANSCOPE エンドポイントマネージャー オンプレミス版　開発

AWS歴：約4年

自己紹介

辻 翼(つじ つばさ)



現在　　：エムオーテックス株式会社 2017年入社
　　LANSCOPE エンドポイントマネージャー クラウド版 アプリ開発チーム

経歴　　：LANSCOPE エンドポイントマネージャー オンプレミス版　開発

AWS歴：約4年

会社紹介

私たちは企業の**資産**である「**情報**」を守り

IT環境における**安全・生産性を追求**することで

社会の進歩発展に貢献します。

MOTEX

大阪に**本社**を置く、**ソフトウェアメーカー**です！

プロダクト紹介

LANSCOPE
Endpoint Manager

ウイルス対策も情報漏洩対策もこの1本で

IT資産管理なのに、iOS・Androidにも対応
MDMなのに、Windowsの操作ログも管理できる

iOS | Android | Windows | MacOS

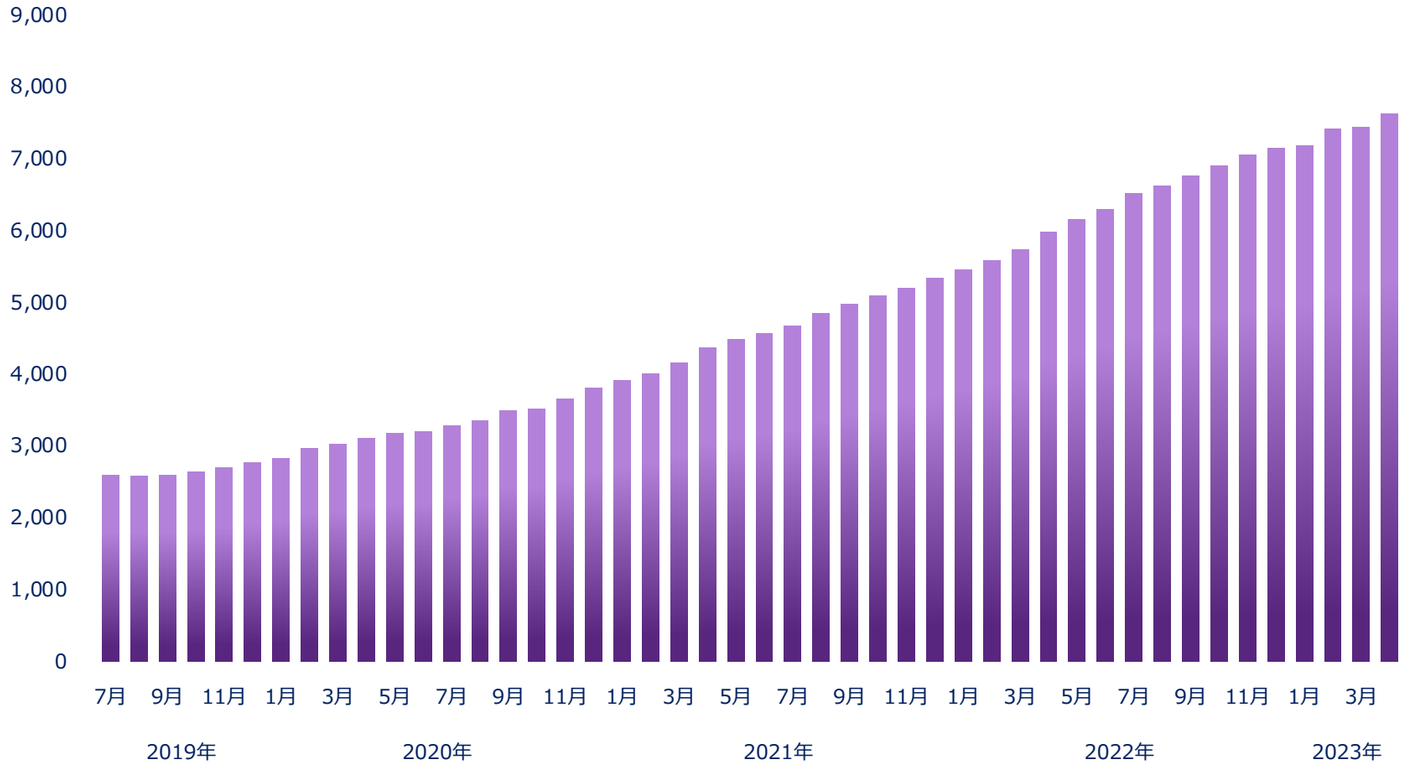
IT資産管理 | 内部不正対策 | 外部脅威対策

LANSCOPE エンドポイントマネージャー クラウド版の実績

導入社数8,000社突破！導入後も90%以上の方が使い続けている LANSCOPE シリーズ

導入後も90%以上の方に使い続けていただける「製品力」と「サポート力」が強み

＜　LANSCOPE エンドポイントマネジャー クラウド版　導入社数実績 推移　＞



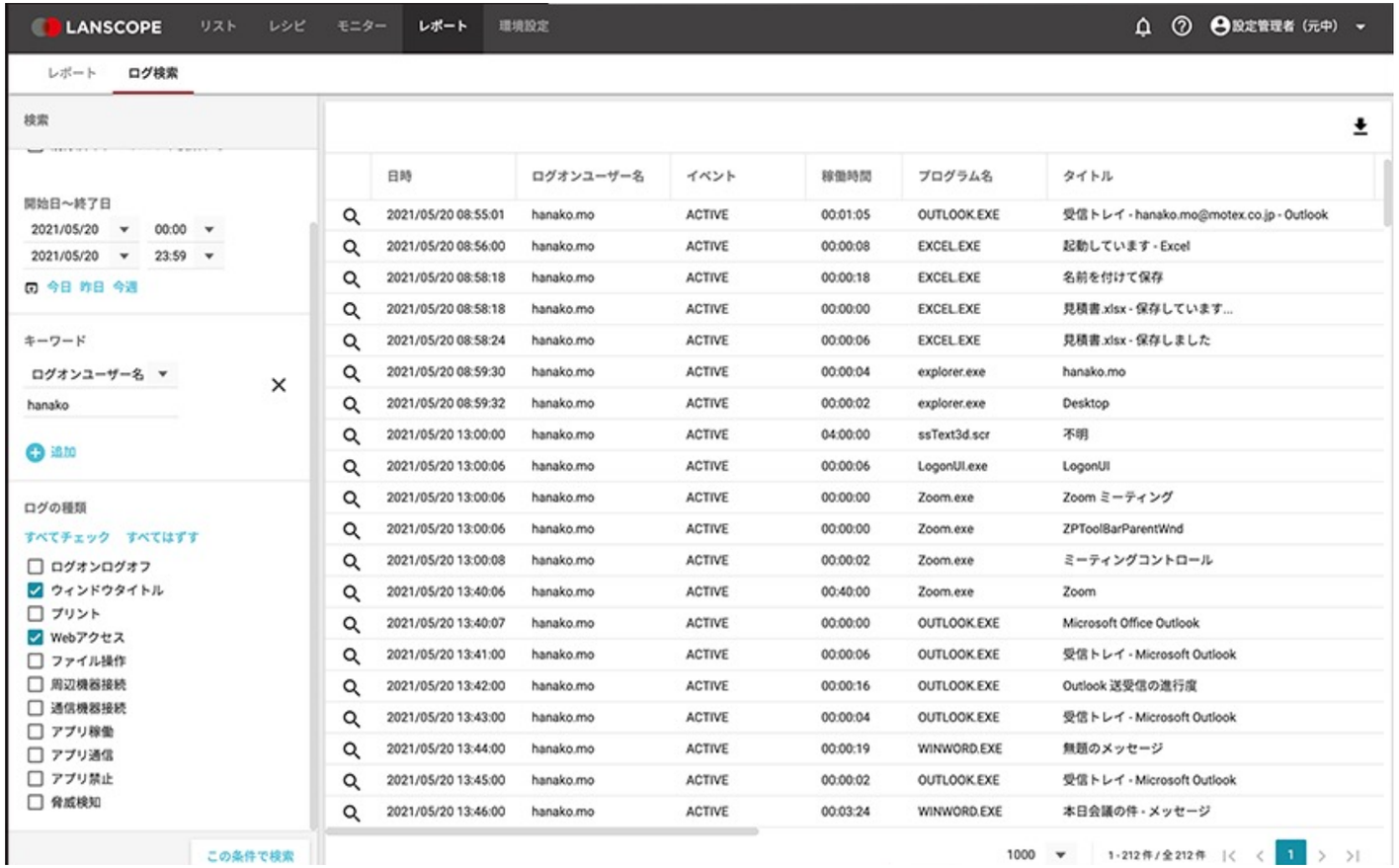
※2023年4月20日現在

事例紹介「ログ受信基盤改善」

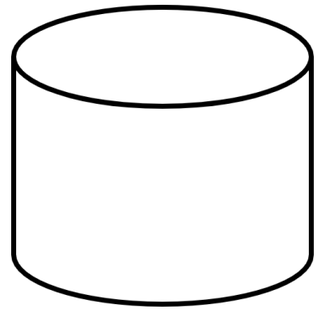
- **プロジェクト概要**
- アーキテクチャ Ver.1 (Amazon EMR)
- アーキテクチャ Ver.2 (Amazon EMR + Amazon Kinesis Data Firehose)
- アーキテクチャ Ver.3 (Amazon Kinesis Data Firehose 1本化)
- 今の課題と今後に向けて

プロジェクト概要

- 組織が管理する PC 上の操作をログとして取得・永続化し、有事の際に証跡が追える機能
  (アプリケーションの起動ログや Web の閲覧ログなど)
- 取得したログは、期間 / キーワード / ログの種類 を使用して検索することができる



検索



数億ログ / 日

プロジェクト概要

- PC 操作のログは、ニアリアルタイムに DB に格納する

  (通信帯域の圧迫を避けるため、バッファリング)

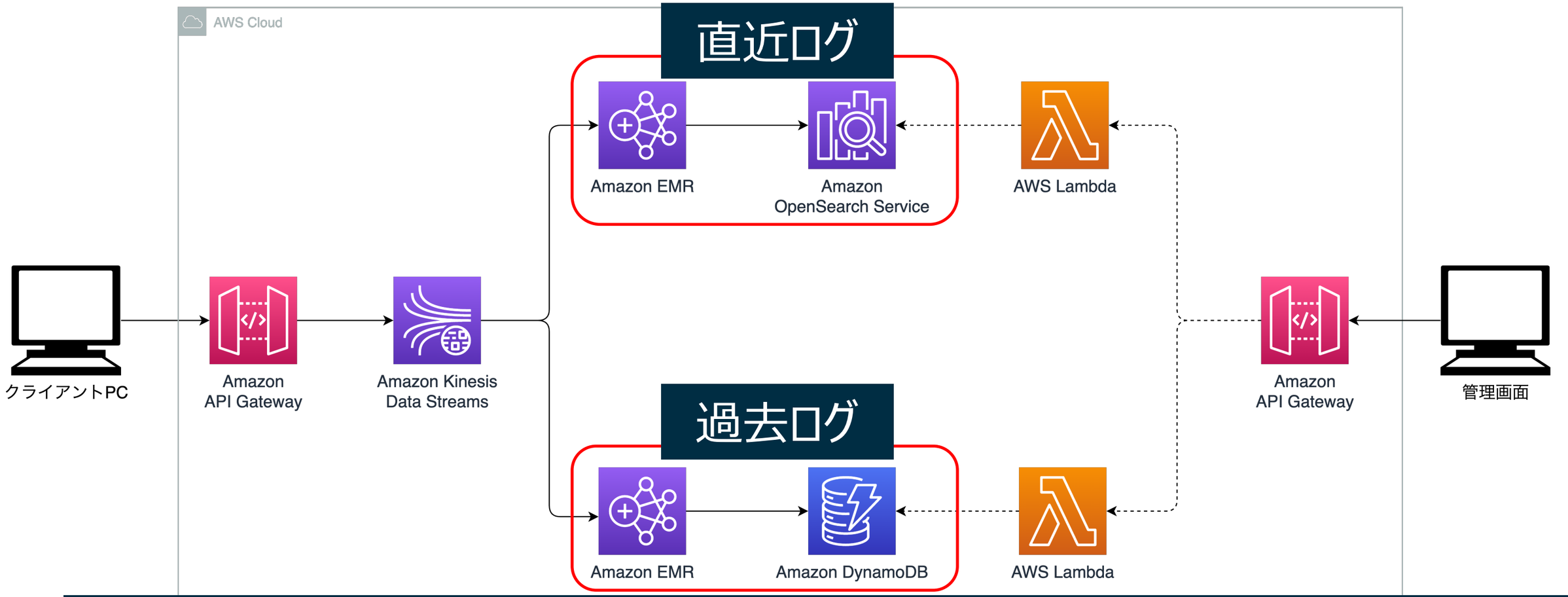

```
{
  "tenant_id": "AAAA",
  "log": [
    { "log1": "xxx" },
    { "log2": "yyy" },
    …
  ]
}
```

```
{
  "tenant_id": "AAAA",
  "log1": "xxx"
},
{
  "tenant_id": "AAAA",
  "log2": "yyy"
}
```

事例紹介「ログ受信基盤改善」

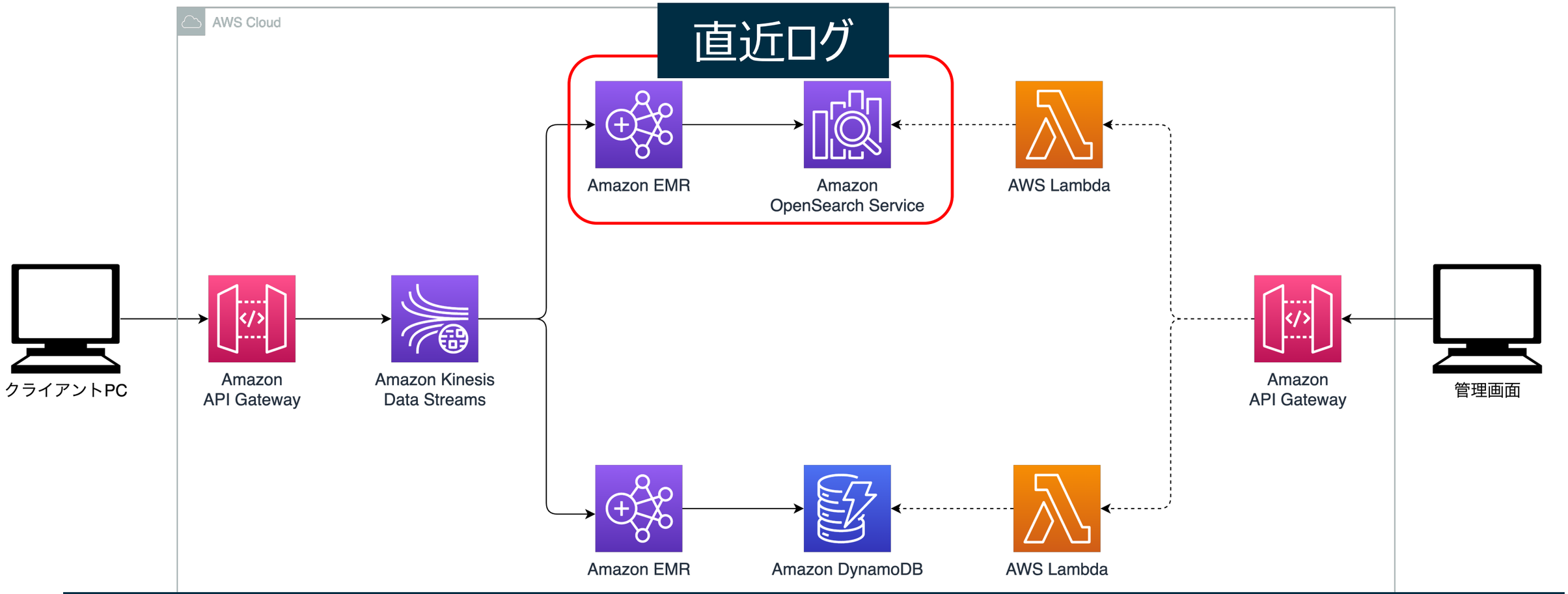
- プロジェクト概要
- **アーキテクチャ Ver.1 (Amazon EMR)**
- アーキテクチャ Ver.2 (Amazon EMR + Amazon Kinesis Data Firehose)
- アーキテクチャ Ver.3 (Amazon Kinesis Data Firehose 1本化)
- 今の課題と今後に向けて

アーキテクチャ Ver.1 (Amazon EMR)



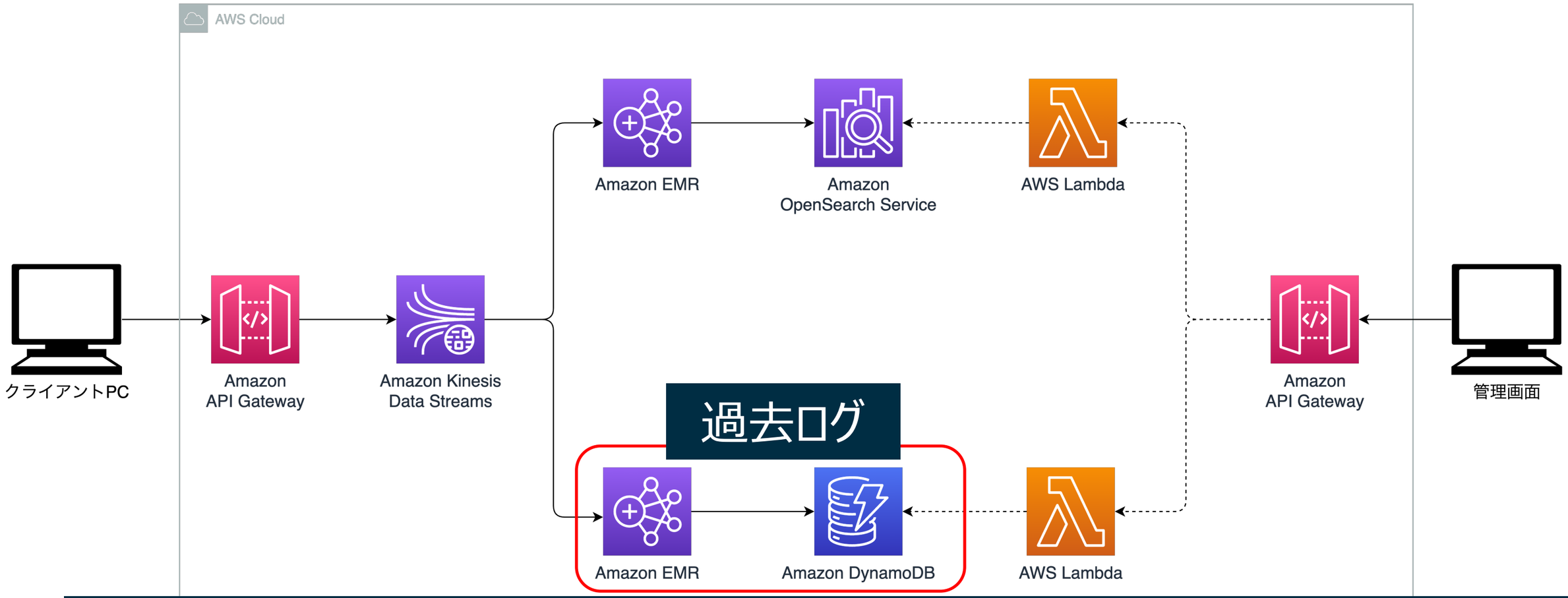
昨日今日のログ (直近ログ) と、
一昨日以前のログ (過去ログ) のストレージを分けて保存

アーキテクチャ Ver.1 (Amazon EMR)



直近ログは Amazon OpenSearch Service を
利用して高速に検索できるように設計

アーキテクチャ Ver.1 (Amazon EMR)



過去ログは多少検索に時間がかかることを許容し、
コストパフォーマンスを優先して設計

アーキテクチャ Ver.1 (Amazon EMR)

- アーキテクチャ候補（ストレージと検索）
  - Amazon Simple Storage Service (Amazon S3) + Amazon Athena の検索
    - 2018年当時、Amazon Athena の「同時実行数: 20」のハードリミットが検索の要件を満たせず不採用
  - Amazon TimeStream の使用
    - 2018年当時、東京リージョンに実装がされておらず、不採用
  - Amazon Aurora の使用
    - 数千社の5年分のログを保持する要件を満たせないと判断して不採用

アーキテクチャ Ver.1 (Amazon EMR) Badポイント

過去ログ機構に課題が発生

- クライアント PC あたりの実際のログ数が、想定していたよりも多かった
  - Amazon EMR のスケールアウトで対応する予定だったが、Amazon DynamoDB 側のスロットルが原因で、リアルタイム性を維持できなくなった

スロットルが発生した原因

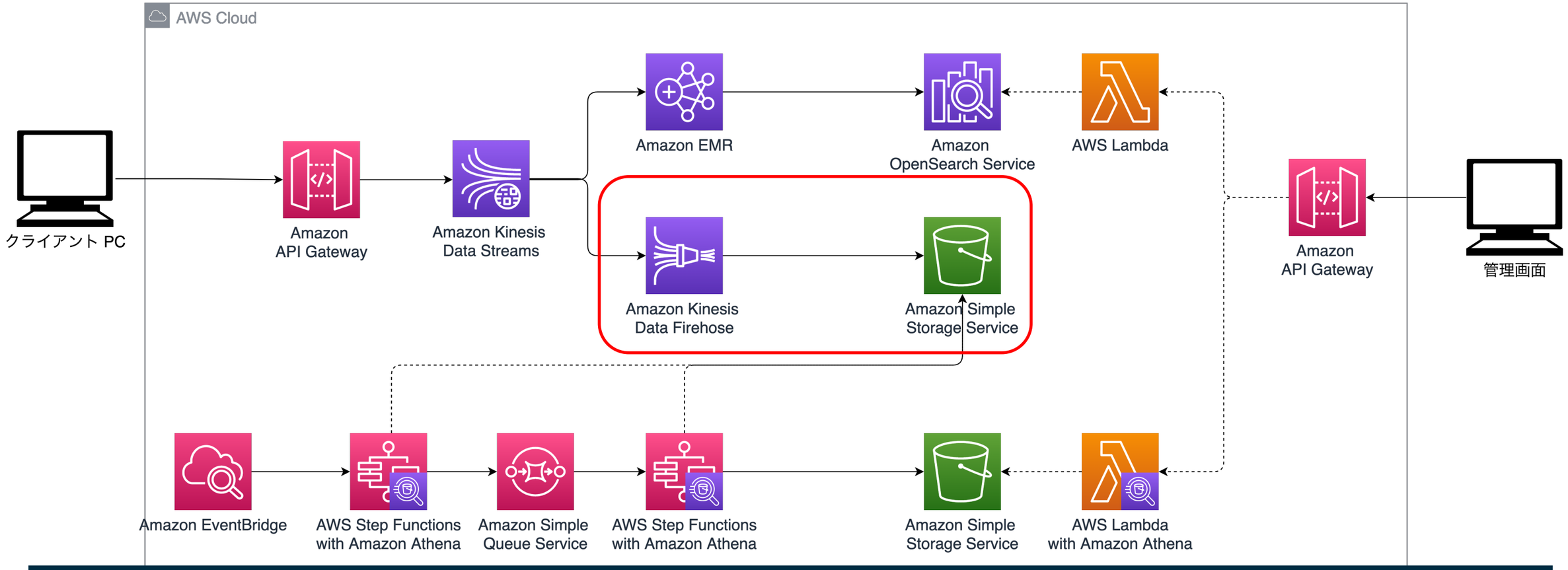
- Amazon DynamoDB のパーティションキーの設計が適切ではなかった
  - Amazon DynamoDB のパーティションは PartitionKey (企業ID+デバイスID), SortKey (秒までの日時) の組み合わせで決まると想定していたが、実際のパーティションは PartitionKey のみ(※1)で分かれるので、特定デバイスが大量ログを上げるだけでかんたんにスロットルしてしまう構成になっていた

※1) https://docs.aws.amazon.com/ja_jp/amazondynamodb/latest/developerguide/HowItWorks.Partitions.html#HowItWorks.Partitions.CompositeKey

事例紹介「ログ受信基盤改善」

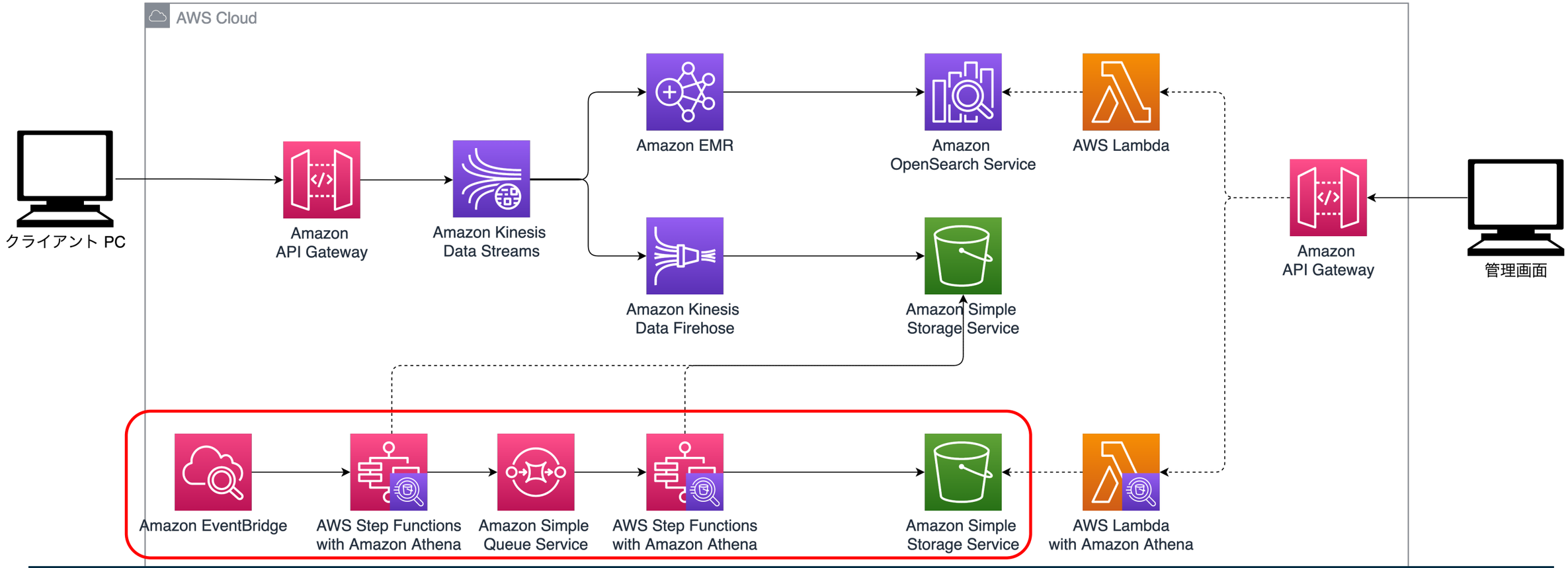
- プロジェクト概要
- アーキテクチャ Ver.1 (Amazon EMR)
- **アーキテクチャ Ver.2 (Amazon EMR + Amazon Kinesis Data Firehose)**
- アーキテクチャ Ver.3 (Amazon Kinesis Data Firehose 1本化)
- 今の課題と今後に向けて

アーキテクチャ Ver.2 (Amazon EMR + Amazon Kinesis Data Firehose)



Amazon Kinesis Data Firehose と Amazon S3 を利用して
Amazon DynamoDB のスロットル問題を解消

アーキテクチャ Ver.2 (Amazon EMR + Amazon Kinesis Data Firehose)



Amazon EMR で行っていたフォーマット処理の代替機構を新たに用意

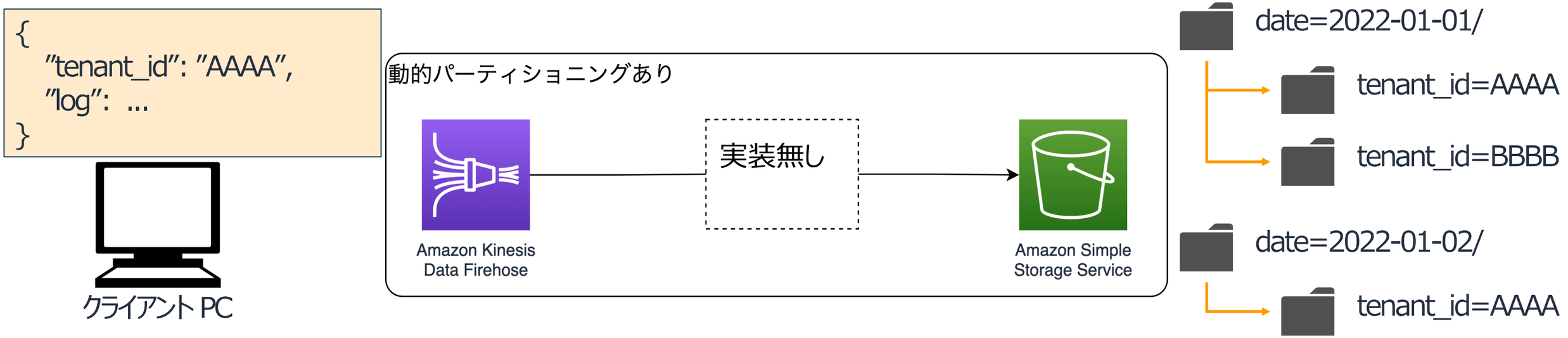
アーキテクチャ Ver.2 (Amazon EMR + Amazon Kinesis Data Firehose)

Good ポイント

- AWSのバージョンアップにより、Amazon Athena の「同時実行数: 20」がソフトリミットに緩和

  ※ 2023年4月 現在、DML(SELECT) クエリのデフォルトリミットは150

- 動的パーティショニングにより、AWS Lambda などの実装無しでテナントごとのプレフィックス作成を実現
- 動的パーティショニングを用いてバケット内のプレフィックスを日付とテナントで分けることにより、テナントごとに並列化したフォーマット処理を実現



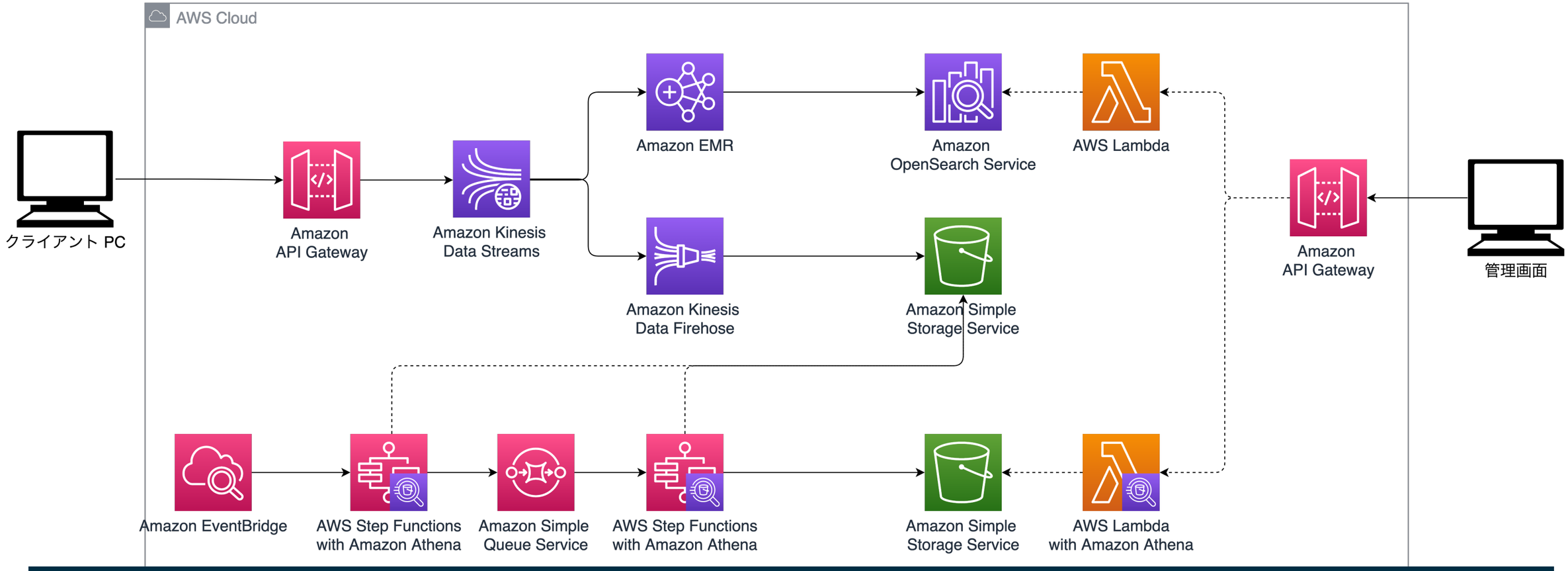
アーキテクチャ Ver.2 (Amazon EMR + Amazon Kinesis Data Firehose)

Good ポイント

- 管理するサービスは増えたが、AWS コストを大きく減らすことができた (+ $895 / 月, - $7,000 / 月)

サービス名	アーキテクチャ Ver.1	アーキテクチャ Ver.2	アーキテクチャ Ver.3
Amazon DynamoDB	$6,000 / 月	$0 / 月	-
Amazon EMR	$2,000 / 月	$1,000 / 月	-
Amazon S3	-	$700 / 月	-
Amazon Kinesis Data Firehose	-	$150 / 月	-
AWS Step Functions	-	$30 / 月	-
Amazon Athena	-	$15 / 月	-
Amazon OpenSearch Serivce	$4,700 / 月	$4,700 / 月	-

アーキテクチャ Ver.2 (Amazon EMR + Amazon Kinesis Data Firehose)



運用課題を解決したことにより、しばらく平和な日々が続いた

アーキテクチャ Ver.2 (Amazon EMR + Amazon Kinesis Data Firehose)

新たな課題発生

- が、それも束の間、新しく「直近ログの検索でタイムアウトが発生する」課題が発生
- ログの検索機能の特徴でもある、「前後を含む部分一致検索」が特定条件下で Amazon OpenSearch Service が苦手とすることが判明

ファイルパス「C:¥¥aaa¥bbb¥ccc¥…(略)¥社外秘.docs」 (最大255文字)
に対して
{"regexp": {"file_path": ".*社外秘.*"}}
で検索するとタイムアウトが発生する

長い文字列に対して前後の部分一致は苦手（時間がかかる）

アーキテクチャ Ver.2 (Amazon EMR + Amazon Kinesis Data Firehose)

- 同じ条件下で過去ログの検索（ Amazon Athena を使用）がタイムアウトしないことを確認
- 「高速で検索をしたい」要件で Amazon OpenSearch Service を選んでいたが、
  Amazon Athena の検索速度が許容範囲
- Amazon EMR + Amazon OpenSearch Service のコストと
  Amazon S3 + Amazon Athena のコストを比較した際に、圧倒的に安価

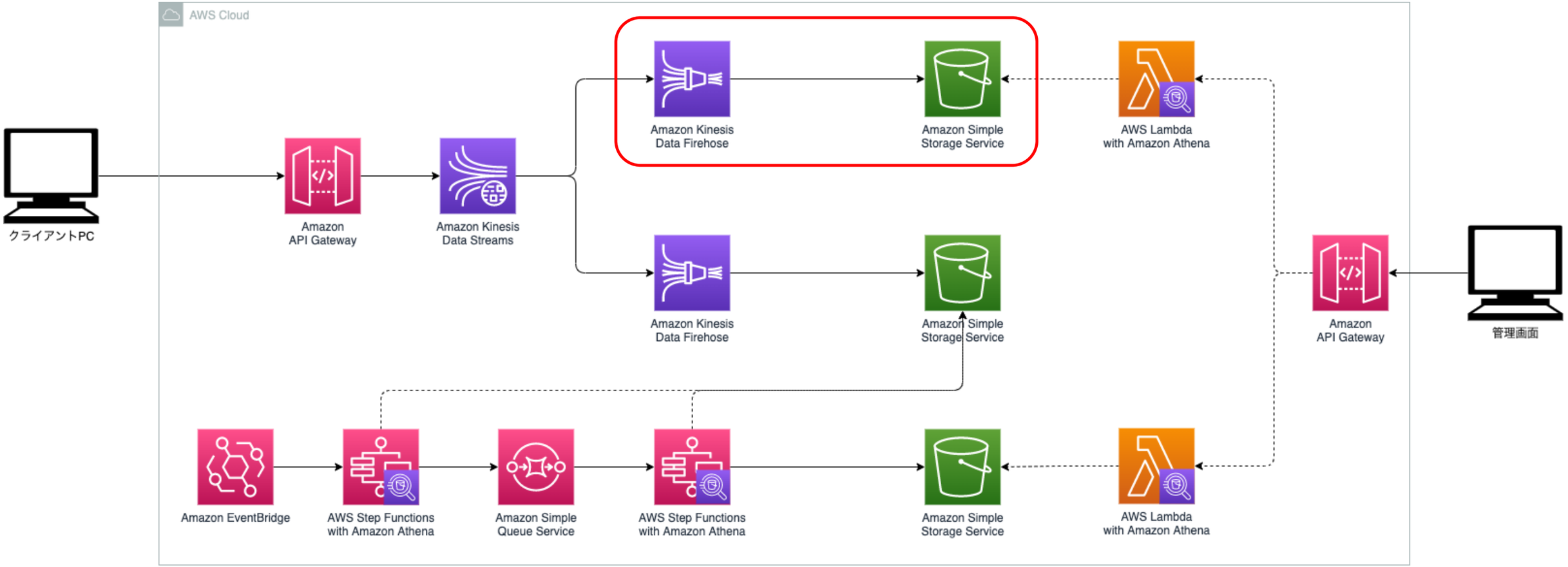
ファイルパス「C:¥¥aaa¥bbb¥ccc¥…(略)¥社外秘.docs」（最大255文字）
に対して
WHERE file_path LIKE %%社外秘%%
で検索するとタイムアウトが発生しない

タイムアウトが発生しない ＋ 検索速度が許容範囲 ＋ コストを抑えられる
＝> リプレイスを検討

事例紹介「ログ受信基盤改善」

- プロジェクト概要
- アーキテクチャ Ver.1 (Amazon EMR)
- アーキテクチャ Ver.2 (Amazon EMR + Amazon Kinesis Data Firehose)
- **アーキテクチャ Ver.3 (Amazon Kinesis Data Firehose 1本化)**
- 今の課題と今後に向けて

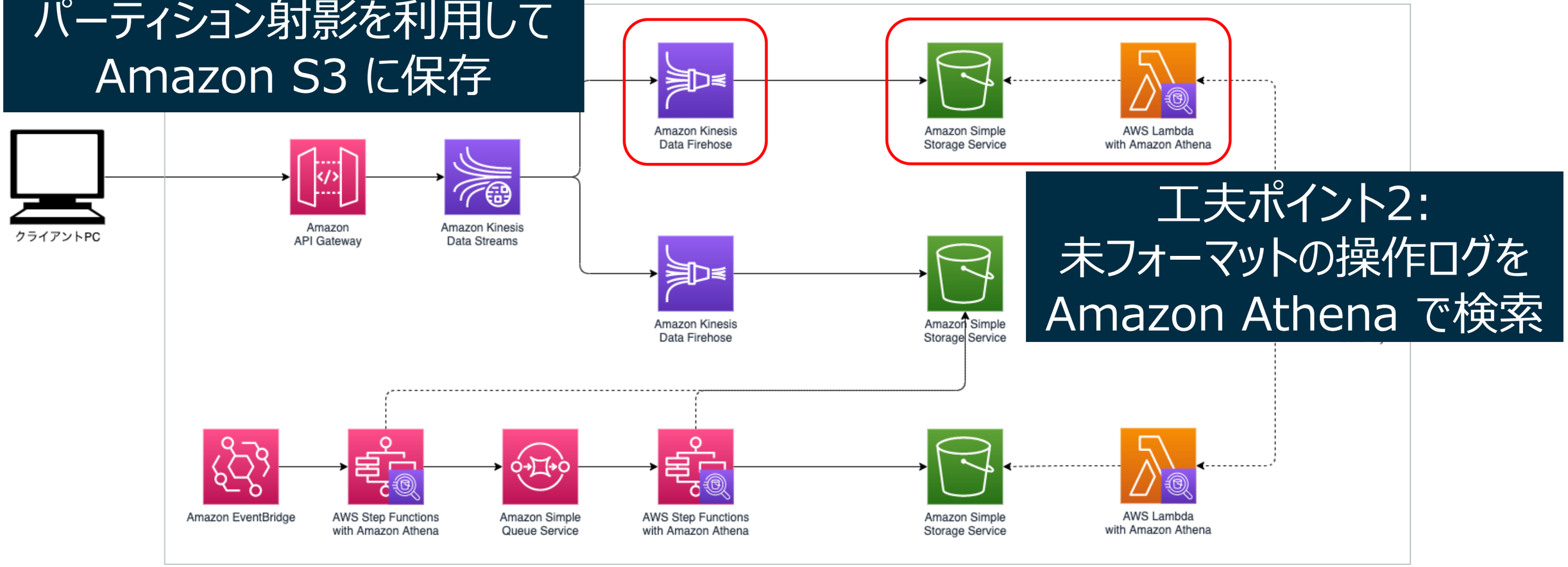
アーキテクチャ Ver.3 (Amazon Kinesis Data Firehose 1本化)



Amazon Kinesis Data Firehose と Amazon S3 を利用して
Amazon OpenSearch Service の課題を解消

アーキテクチャ Ver.3 (Amazon Kinesis Data Firehose 1本化) ~工夫点~

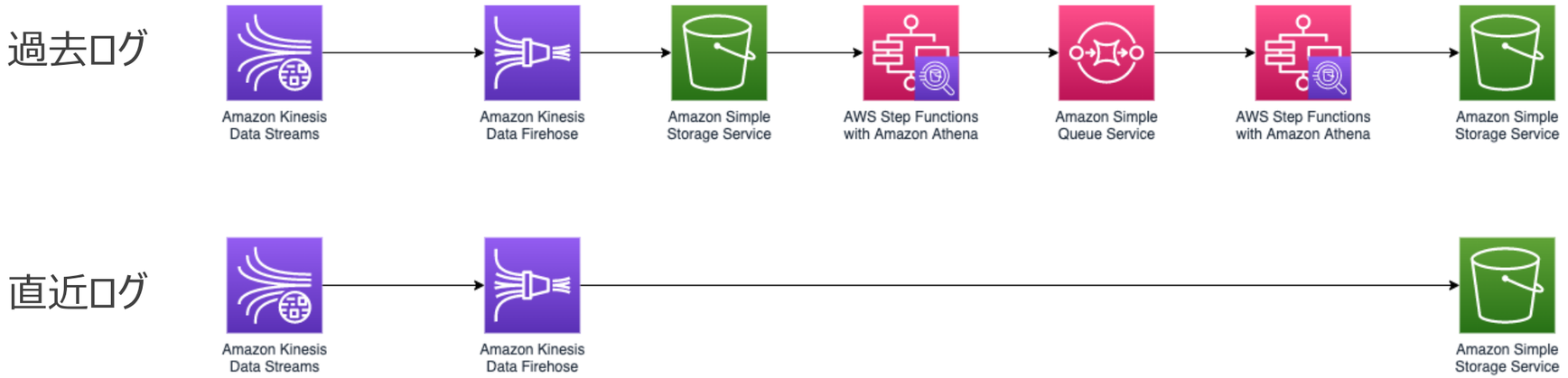
工夫ポイント1:
パーティション射影を利用して
Amazon S3 に保存



工夫ポイント2:
未フォーマットの操作ログを
Amazon Athena で検索

工夫ポイント１　パーティション射影を利用して Amazon S3 に保存

- 過去ログの定期処理では、操作ログを検索しやすいようにするために以下の処理を行っている
  - CREATE TABLE AS SELECT (CTAS) クエリ
  - MSCK REPAIR TABLE
- しかし、直近ログはリアルタイムに検索できる必要があるため、定期処理は実施できない
- そこで、パーティション射影を使用し、上記2処理を省略した



工夫ポイント２　未フォーマットのログを Amazon Athena で検索

- 検索時にフォーマットと同じ処理をすることで、定期処理を待たずに検索を実現
- ただし、クエリのコストが過去ログより多くかかってしまうので、
  過去ログとバケットを分けて検索期間を短くすることで、直近ログの性能を維持

Amazon Athena

直近ログの検索
（クエリ上でフォーマット）

Amazon Simple
Storage Service

```
{
  "tenant_id": "AAAA",
  "log": [
    { "log1": "xxx" },
    { "log2": "yyy" },
    …
  ]
}
```

過去ログの検索

Amazon Simple
Storage Service

```
{
  "tenant_id": "AAAA",
  "log1": "xxx"
},
{
  "tenant_id": "AAAA",
  "log2": "yyy"
}
```

改善後の効果

- Amazon OpenSearch Service のリプレイスもあり、AWS コストを大きく減らすことができた

  + $1,145 / 月, - $12,700 / 月

サービス名	アーキテクチャ Ver.1	アーキテクチャ Ver.2	アーキテクチャ Ver.3
Amazon DynamoDB	$6,000 / 月	$0 / 月	$0 / 月
Amazon EMR	$2,000 / 月	$1,000 / 月	$0 / 月
Amazon S3	-	$700 / 月	$800 / 月
Amazon Kinesis Data Firehose	-	$150 / 月	$300 / 月
AWS Step Functions	-	$30 / 月	$30 / 月
Amazon Athena	-	$15 / 月	$15 / 月
Amazon OpenSearch Serivce	$4,700 / 月	$4,700 / 月	$0 / 月

改善後の効果

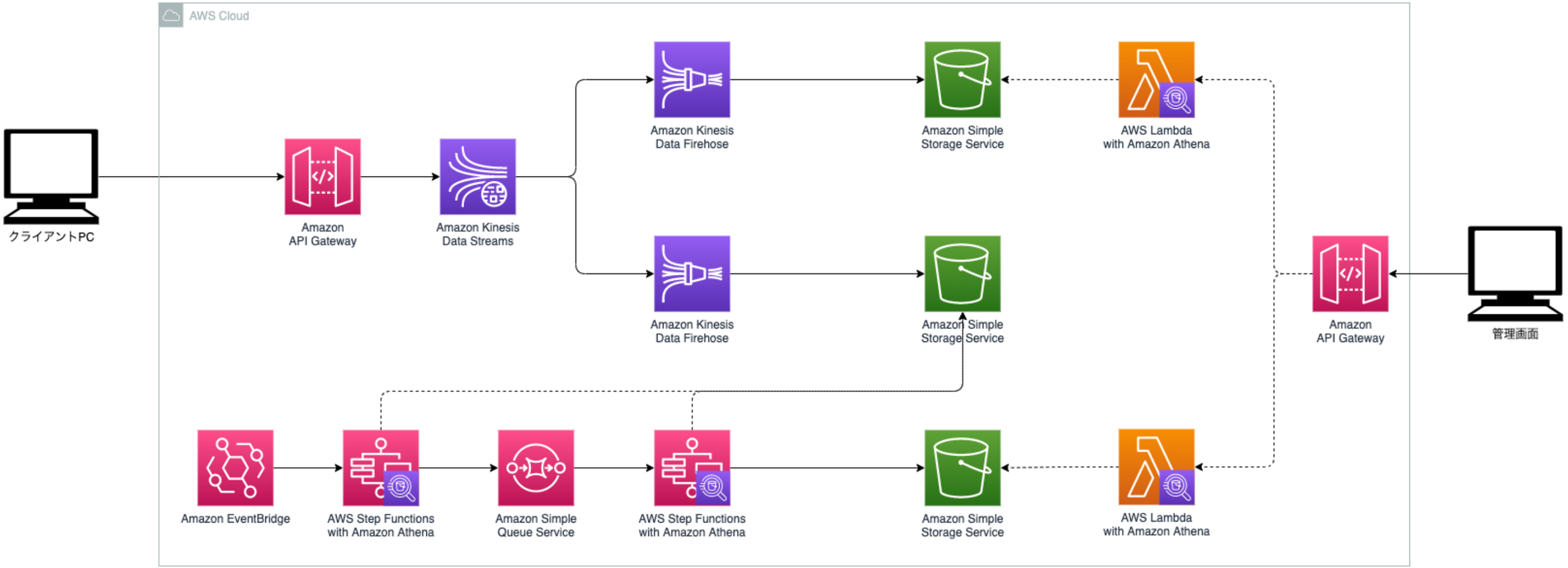
- Amazon OpenSearch Service のリプレイスもあり、AWS コストを大きく減らすことができた

  + $1,145 / 月, - $12,700 / 月

サービス名	アーキテクチャ Ver.1	アーキテクチャ Ver.2	アーキテクチャ Ver.3
Amazon DynamoDB	$6,000 / 月	$0 / 月	$0 / 月
Amazon EMR	$2,000 / 月	$1,000 / 月	$0 / 月
Amazon S3	-	$700 / 月	$800 / 月
Amazon Kinesis Data Firehose	-	$150 / 月	$300 / 月
AWS Step Functions	-	$30 / 月	$30 / 月
A			
A			

アーキテクチャ Ver.1 と比べて、–$10,000 / 月を実現

アーキテクチャ Ver.3 (Amazon Kinesis Data Firehose 1本化)



これですべての課題を解決・・・したはずだった

アーキテクチャ Ver.3 (Amazon Kinesis Data Firehose 1本化)

Amazon Kinesis Data Firehose のクォータ

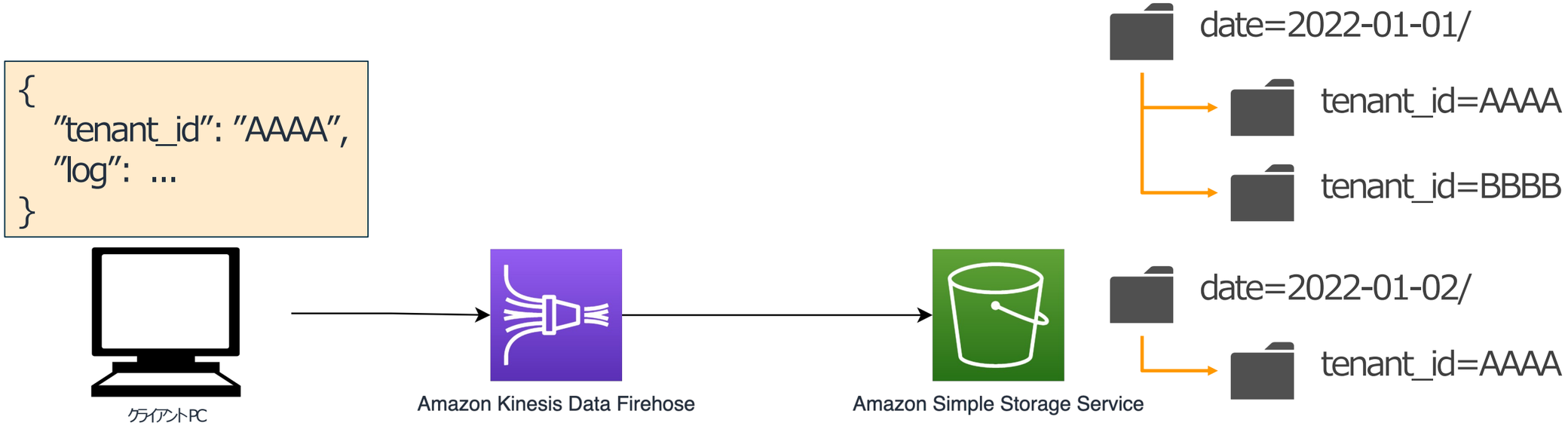
PDF | RSS

Amazon Kinesis Data Firehose には、次のクォータがあります。

- 配信ストリームで動的パーティショニングが有効な場合、その配信ストリームに対して作成できるアクティブパーティションは 500 個に制限されます。アクティブパーティション数は、配信バッファ内のアクティブパーティションの総数です。たとえば、動的パーティショニングクエリが 1 秒あたり 3 つのパーティションを構築し、60 秒ごとに配信をトリガーするバッファのヒント設定がある場合、平均して 180 個のアクティブパーティションが作成されます。データがパーティションで配信されると、そのパーティションはアクティブではなくなります。Amazon Kinesis Data Firehose の制限フォームを使用して、特定の配信ストリームごとに最大 5000 個のアクティブパーティションまで、このクォータの引き上げをリクエストできます。より多くのパーティションが必要な場合は、より多くのデリバリーストリームを作成し、アクティブなパーティションをそれら全体に分散できます。

動的パーティショニングに新たなハードリミットが設定された

アーキテクチャ Ver.3 (Amazon Kinesis Data Firehose 1本化)



テナントごとにパーティションを作成しているため、
ハードリミットとなってしまったのは致命的

アーキテクチャ Ver.3 (Amazon Kinesis Data Firehose 1本化)

- 今はまだハードリミットまでに余裕がある状況
- しかし、製品の成長見込みから、今期中には何かしらの対応を入れることを検討中
  - クォータに引っかかった際にリトライ機構を用意する　など

リトライありきのアーキテクチャにはしたくないので、
機能のリクエストをさせていただいております。
ぜひご検討お願いします！

まとめ

まとめ

- AWS のバージョンアップにより、当初取れなかった Amazon S3 -> Amazon Athena での検索を実現することができた
- さらに、動的パーティショニングや、パーティション射影の実装が、ログ受信基盤のフルマネージド化の追い風となった
- その結果、利用コストも運用コストも同時に抑えることができた
- AWS SA さんのサポートを活用することで、知見の無い新機能・新サービスも安心して使用することができた

AWS SA さんのサポートを最大限に活用しましょう

採用してます！

- カジュアル面談やってます！

MOTEX

【中途採用】自社サービスの開発エンジニア（1部・2部）

当社が提供するモバイルデバイス・PCの資産管理やセキュリティ対策ができる
法人向けサービスの設計・開発を担当頂きます。既存機能の強化に加え、積極的な新機能開発を進め、
サーバレスアーキテクチャを採用するなど、新しい技術にも挑戦し続けています。

【サービス開発1部】
・「LANSCOPEクラウド版」の機能開発、改善、保守、運用
・同サービスの機能開発における要件定義、設計、実装、テスト、デプロイメント
＜扱う技術＞（一部）
[言語]
・サーバー　Scala / Java / Python
・フロント　Typescript +&nbs… [more]

職種

技術職 ＞自社製品の開発エンジニア

勤務地

大阪

MOTEX