

レガシーバッヂからの 脱却+a

2018/09/28 BigData JAWS #14

フリーランス ソフトウェアエンジニア 田村彰一

自己紹介

田村 彰一

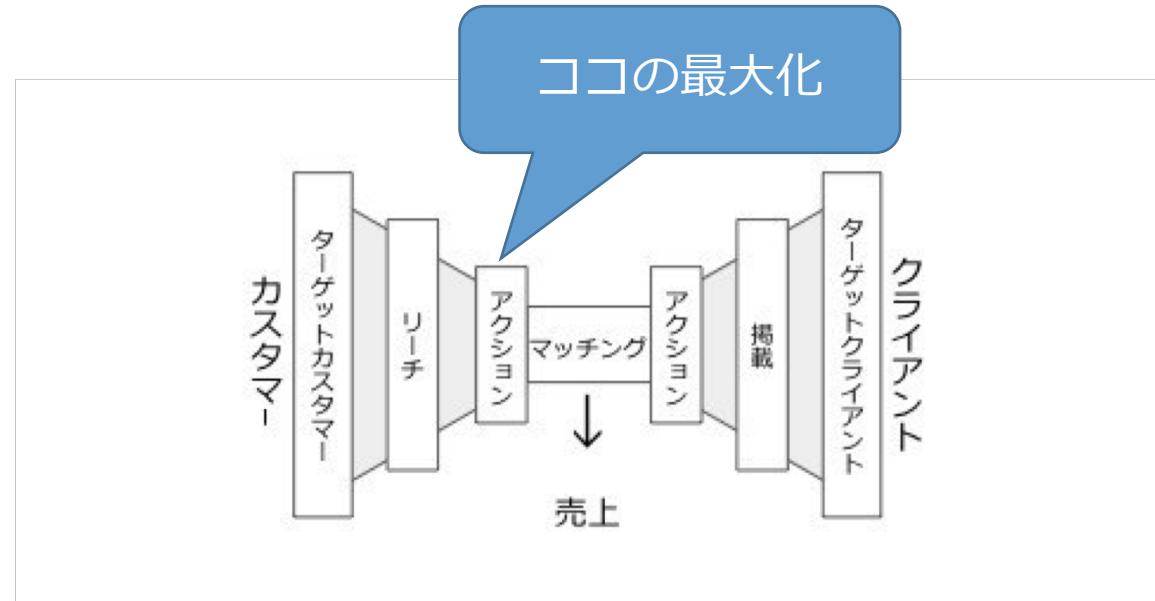
某社の人材領域のレコメンドサービス基盤を担当しつつ、機械学習モデルの開発もやっております。

Agenda

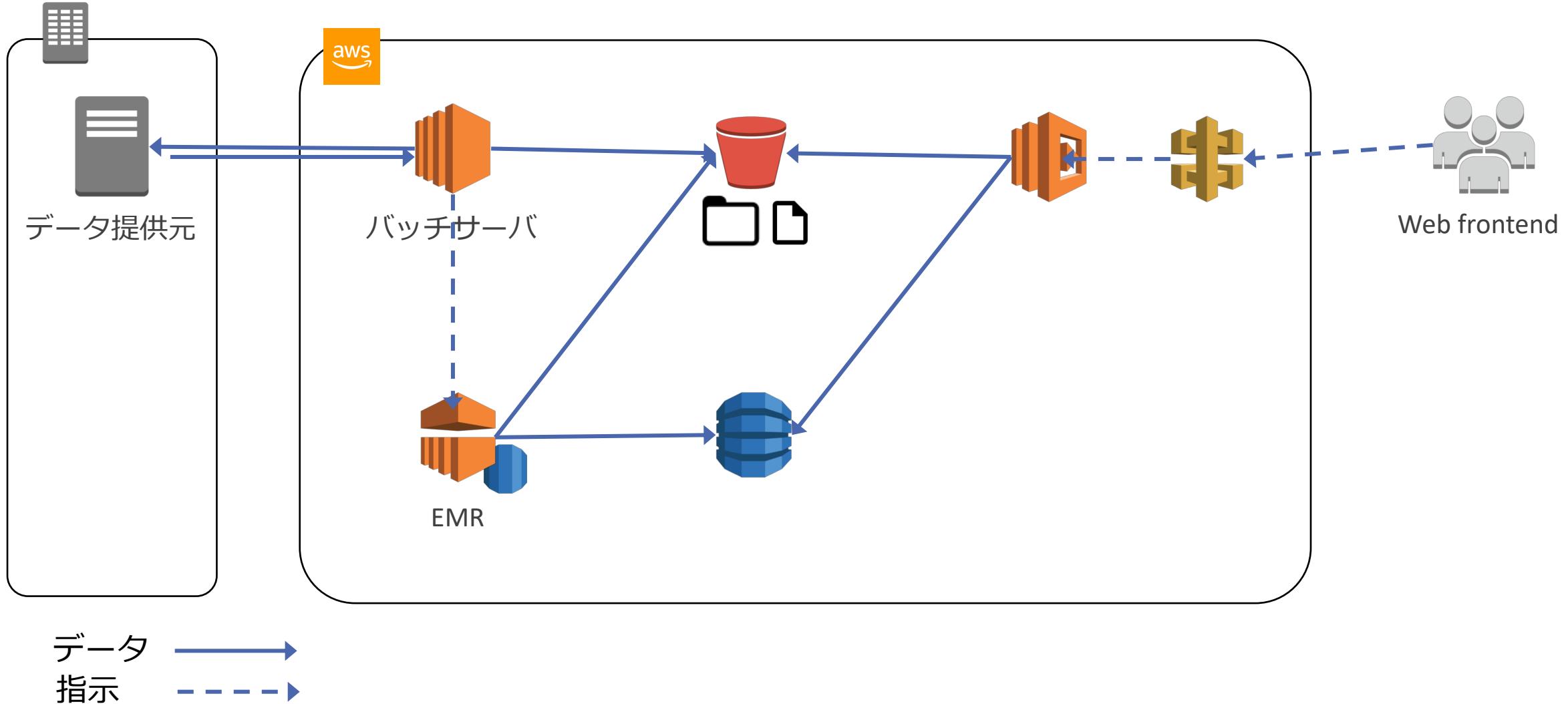
- 対象サービス概要
- 現状
- 理想像
- 取り組み状況
- 他

対象サービス概要

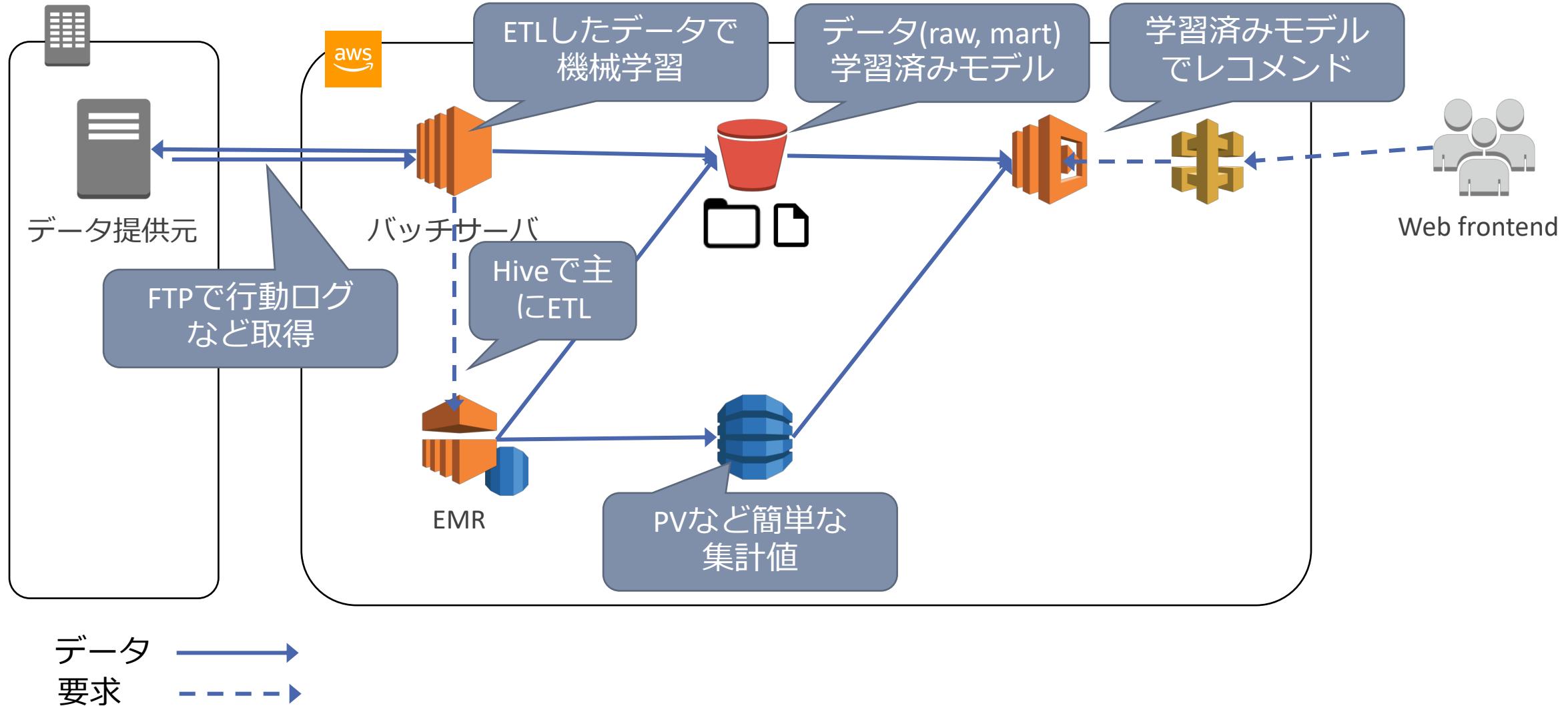
- 一言でまとめると、顧客属性や行動からレコメンデーションしてコンバージョン向上を目的としたサービス。



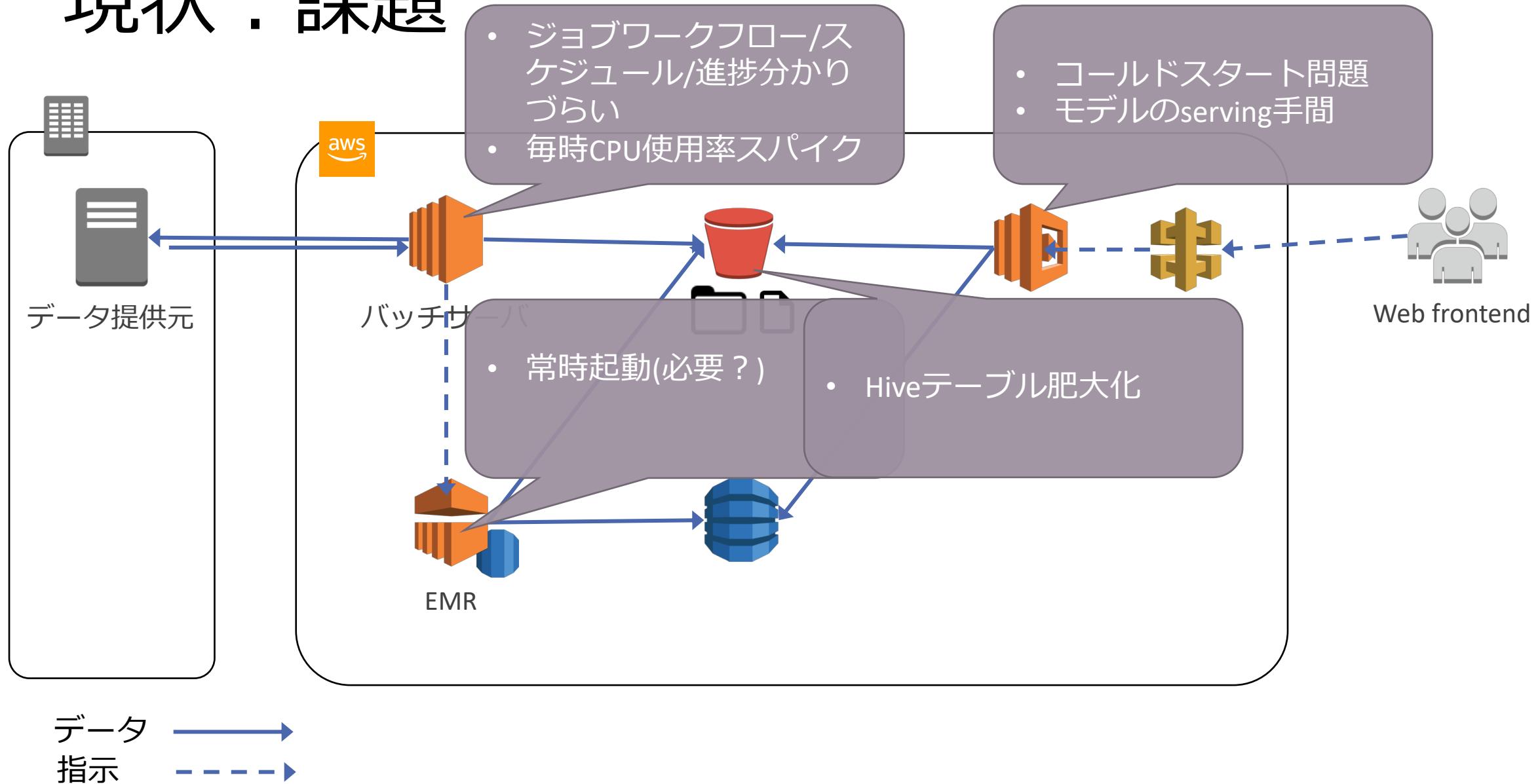
現状：構成



現状：構成



現状：課題



課題：バッチ

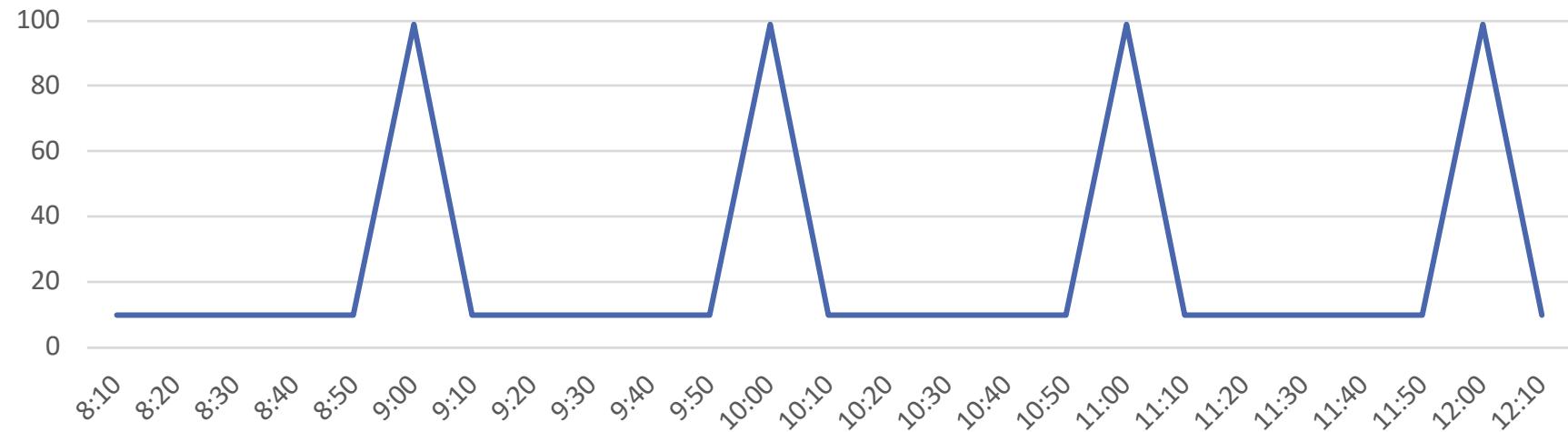
- ジョブワークフローが分からない
 - 秘伝の親シェルで全てを貯う
 - ワークフローは子シェル呼び出し。コードを見ればワークフロー分かるといえば分かるが。
 - 進捗は秘伝の親シェルから送るメールで分かる。
- ジョブスケジュールが分からない
 - 100行以上のcron
 - Excelでスケジュール俯瞰！
 - 大半が毎分起動だったり。。。データ提供元のファイル存在チェック。
- ジョブ進捗
 - 親シェルから開始・終了メールのみ。子シェルの状況はログ見て。。。

課題：バッチ

- ジョブワークフローが分からない
 - 秘伝の親シェルで全てを貰う
 - ワークフローの仕組みが分からぬかるといえば
 - 分かりぬかるといふ
 - 進むべき方向性が分からぬかるといふ
- ジョブエンジン導入して親シェルからの卒業！
 - 100行のスクリプトを書く。
 - ジョブスケジューラも導入！
 - Example : cron
 - 大量のジョブを並列実行する。
- ジョブ追跡
 - 親シェルから開始・終了メールのみ。子シェルの状況はログ見て。。。

課題：バッチ

- 毎時CPU使用率スパイク
 - こんな感じ。。。ワークロードの異なるバッチがごった煮。



TODO : Grafanaのグラフにする

課題：バッチ

- 毎時CPU使用率スパイク

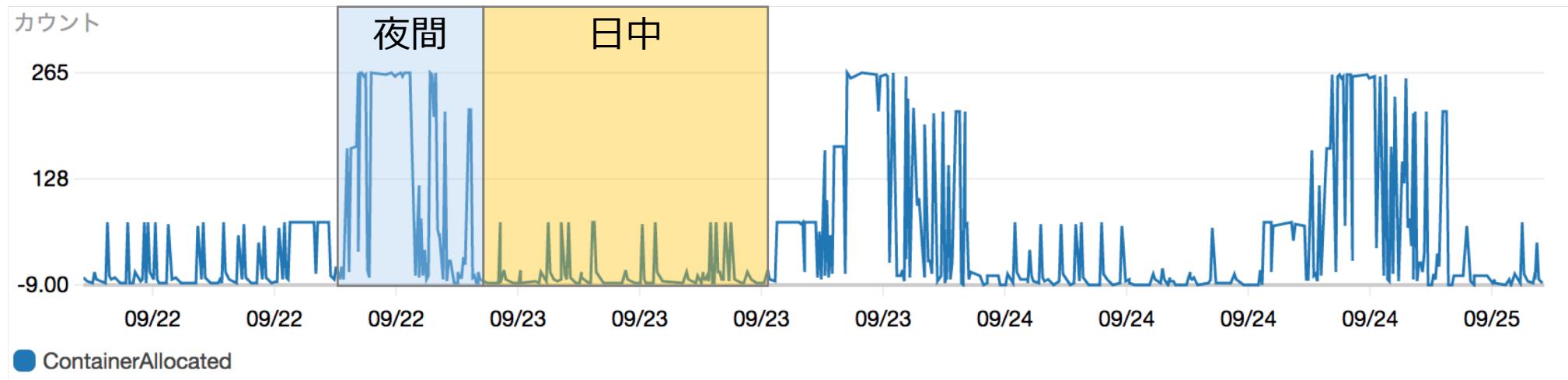
- こんな感じ。 ワークロードの異なるバッチが混ざった者、



TODO : Grafanaのグラフにする

課題：EMR

- ・常時起動？
 - ・ContainerAllocatedはこんな感じ。。。。



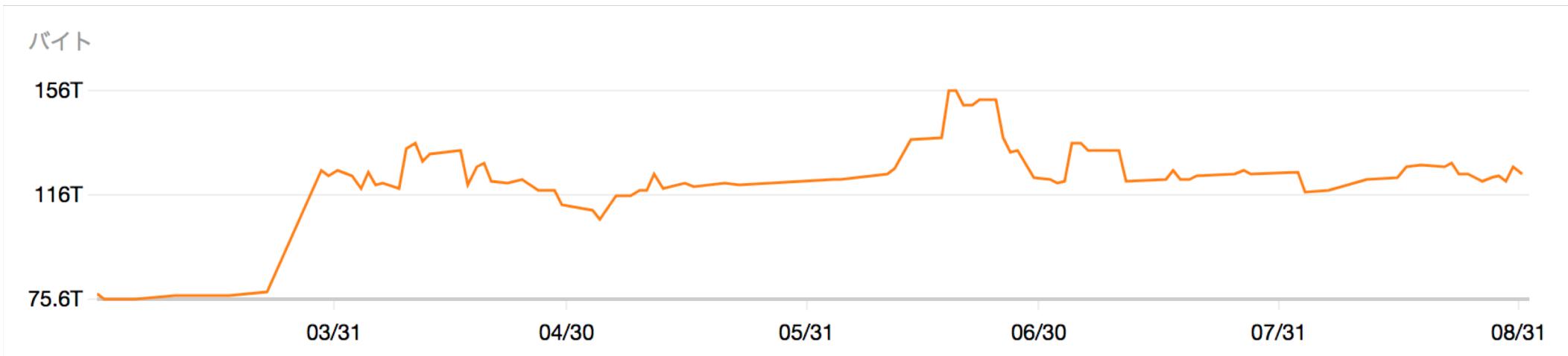
課題：EMR

- 常時起動？
 - ContainerAllocatedはこんな感じ。。。



課題：EMR

- S3容量
 - ほぼほぼTSVで非圧縮。
 - 容量と関係ないが、テーブル統計情報がおかしい。。。



課題：EMR

- S3容量
 - ほぼほぼTSVで非圧縮。
 - 容量と



課題：API

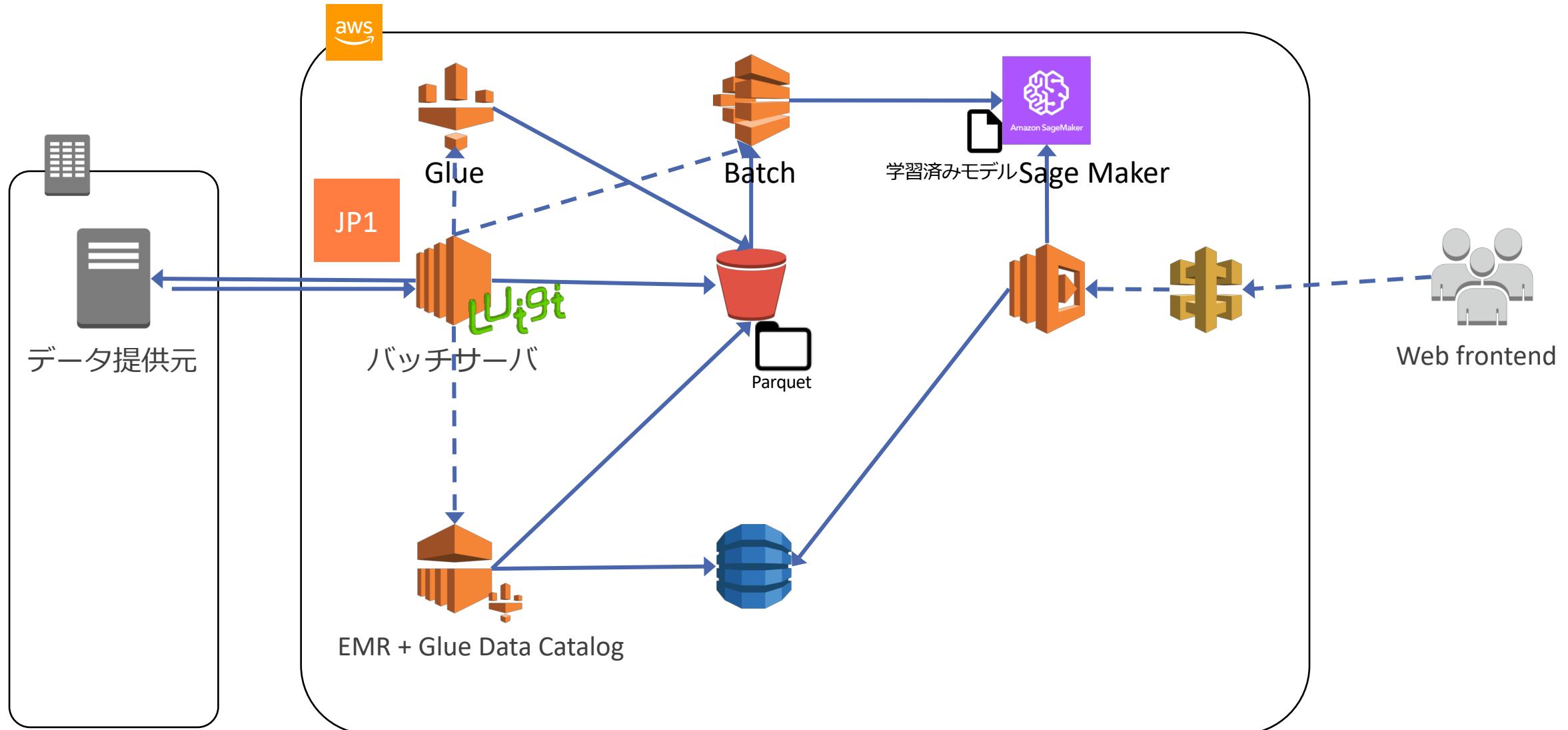
- Lambdaコードスタート
 - 機械学習系のPythonパッケージが大きい。
 - S3から取ってくる学習済みモデルも大きい。
- 学習済みモデルのAPI化の手間

課題：API

- Lambdaコールドスタート
 - 機械学習系のPythonパッケージが大きい。
 - S3から読み込む。
- 学習済みモデルをSage Makerで実行する。

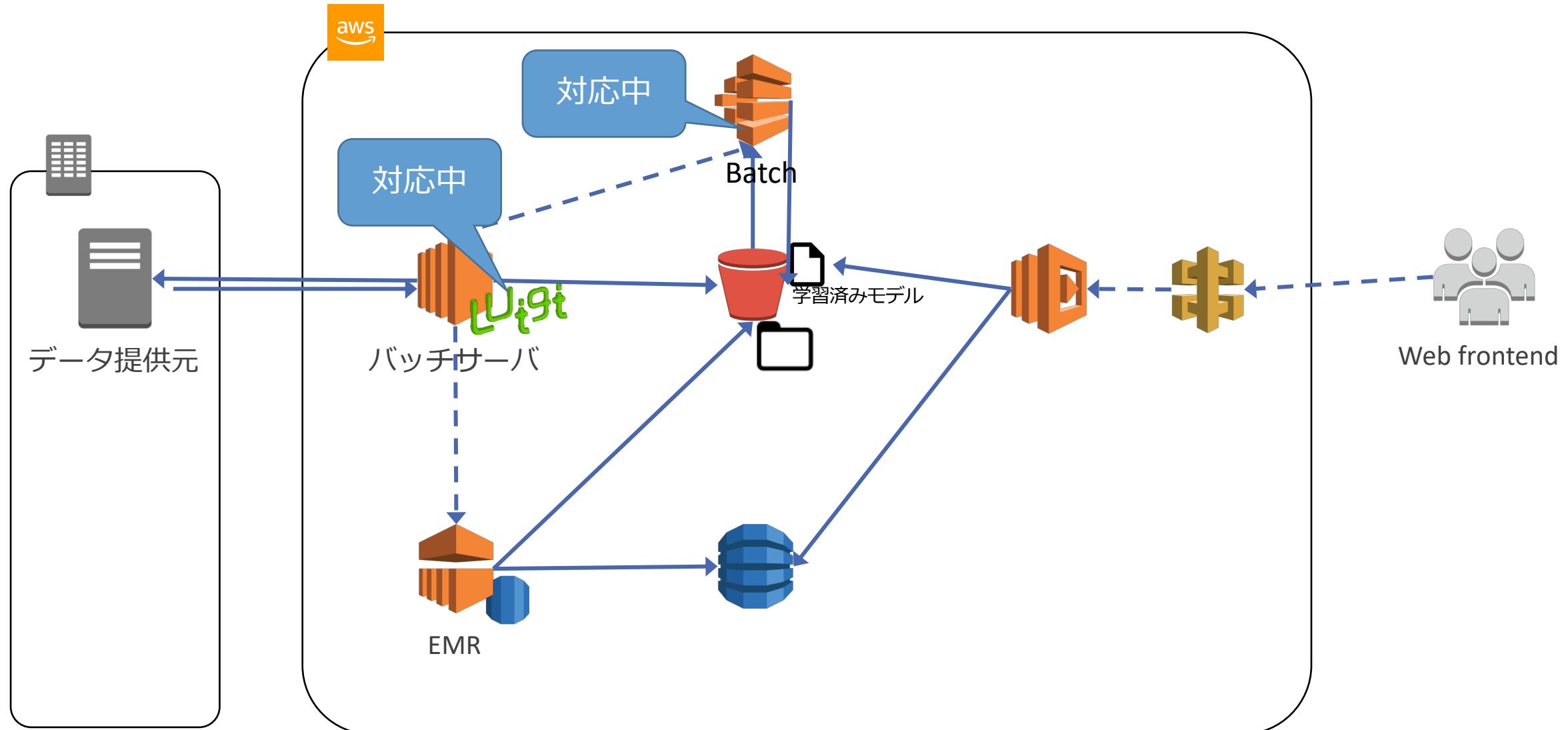
Sage Maker の推論エンドポイント！

理想像



複雑化したような気もするが、ひとまずはこれで行ってみる。

ここまででは来た...

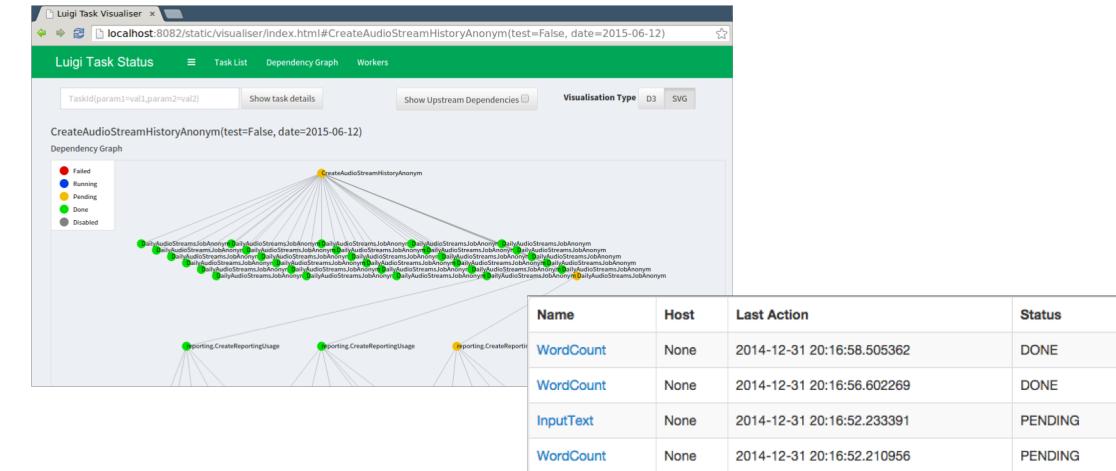


道半ば過ぎて、お伝えできることが少なすぎる。。。。

Tips的なネタも織り交ぜさせてください m(_)_m

取り組み1

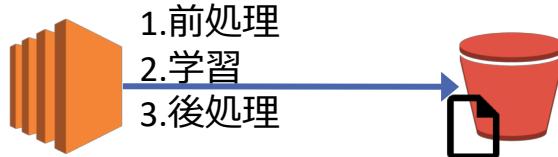
- ・ワークフロー管理、ジョブ進捗可視化
 - ・秘伝の親シェル卒業間近！
 - ・まだ local scheduler なので親シェルから親pythonに変わった程度だが、リランがだいぶシンプルになっただけでも嬉しい。
 - ・エラーになったタスクのOutputファイル消してジョブ実行。でOK。
 - ・central scheduler にすればこうなる。
タスクレベルの進捗や履歴の可視化。
- ・ジョブスケジューラ
 - ・複雑なスケジュールでもないし、
安定のcronのままでも良いかも。。。？
 - ・大半を占める毎分ジョブ(ファイル配置チェック)は別方式検討中。



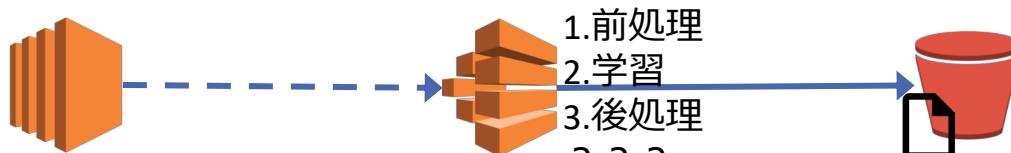
取り組み2

- CPU使用率スパイク
 - AWS Batch へ、重いタスクから移行中。

- Before

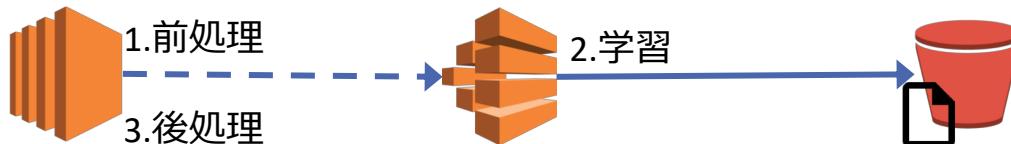


- After



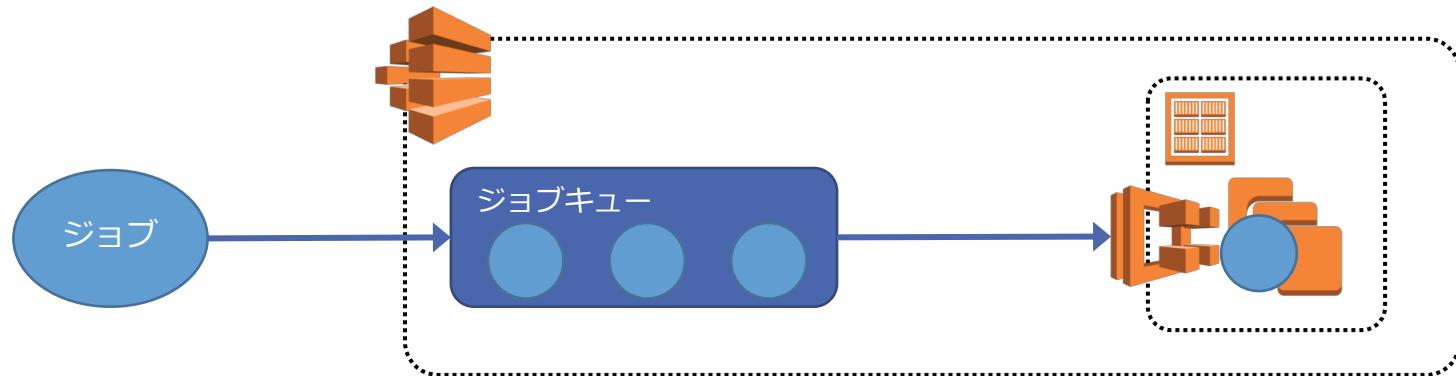
- 勢い余って、バッチサーバにとって軽い処理(待つだけのHiveクエリ)まで移行してしまった模様。高コスト!

- 本当は



AWS Batch : 概要

- ジョブに必要なリソースを指定してジョブ(Dockerイメージリポジトリパス+コマンド)を投げると、良い感じのリソース(EC2)を用意してジョブを実行してくれる。



- タイムアウト、リトライも指定できる。
- 料金は、EC2等の消費リソース分のみ。

AWS Batch : 概要

- CloudWatch Events のソース、ターゲットの両方に指定できる

The screenshot shows the 'Rules' section of the AWS CloudWatch Events console. A rule named 'batch_fail' is selected. The 'Overview' tab is active, showing the ARN: arn:aws:events:ap-northeast-1:...:rule/batch_fail. The 'Event pattern' section displays a JSON-based event pattern:

```
{"source": ["aws.batch"], "detail-type": ["Batch Job State Change"], "detail": {"status": ["FAILED"]}}
```

The 'Status' is listed as 'Active'. The 'Description' field contains the text '失敗したバッチについての通知'. The 'Metrics' section is labeled 'モニタリング Show metrics for the rule'. The 'Targets' section is titled 'ターゲット' and lists one target: an SNS topic named 'batch_error'.

簡単なバッチ処理基盤なら、Batchで良さそう。

AWS Batch : Tips

- ログ
 - 標準出力はジョブ毎に CloudWatch Logs のロググループ /aws/batch/job/ へ。
 - ログストリーム名はこれで分かる。

```
$ aws batch describe-jobs --jobs 2b980841-5f29-4cd7-8152-f4a880094af6 | jq '.["jobs"] | .[] | .container.logStreamName'  
"test/default/5d67bdb9-0773-4c8f-b8f5-ace12b6ff3ec"
```
- 監視
 - ジョブの異常はバッチ内から送るメールがあるのでそのまま。。。
 - ひとまずジョブの滞留を監視。
 - Lambdaでキューごとの滞留ジョブ(statusがfailed,succeeded以外)をカウントして、CloudWatch Metricsとして登録。

AWS Batch : Tips

- ・サービス制限。
 - ・ドキュメントでは制限緩和リクエストできなさそうに見えるが、サポート窓口からは「コンピューティング環境数」「ジョブキュー数」についてはできそうだった。

AWS ドキュメント » AWS Batch » ユーザーガイド » AWS Batch サービスの制限

AWS Batch サービスの制限

次の表は、AWS Batch のサービス制限を示していて、これは変更できません。

リソース	制限
ジョブキューの最大数	20
コンピューティング環境の最大数	18
ジョブキューあたりのコンピューティング環境の最大数	3
ジョブの依存関係の最大数	20
ジョブ定義の最大サイズ (RegisterJobDefinition API オペレーションの場合)	24 KiB
ジョブの最大ペイロードサイズ (SubmitJob API オペレーションの場合)	30 KiB
配列ジョブの最大配列サイズ	10,000
SUBMITTED 状態のジョブの最大数	100 万回

内容* アカウントおよび請求サポート
 サービス制限の緩和
 技術サポート
ベーシックサポートプランではご利用いただけません

制限タイプ* Batch

リクエスト1

リージョン* アジアパシフィック（東京）

制限* 選択または検索

コンピューティング環境の数
ジョブキューの数

他のリクエストの追加

AWS Batch : 残念な点

- ジョブ起動までに時間がかかる。
 - ジョブを投げると諸々の環境準備が行われるので、致し方ない。
 1. ジョブ投下
 2. 要求リソースに合わせた Auto Scaling Group が作成される
 3. オートスケール発動、EC2起動
 4. ECSクラスタが作成され、EC2が紐づけられる
 5. ジョブ(Dockerイメージ)がECSに投げられる
 - サポートへ「遅いす」と連絡。改善リクエスト上げてくれる、との回答はいただけた。
 - 最小vCPUを設定しておけば、環境は維持されるが、それが必要ということはBatch以外の最適解がありそう。

最小 vCPU

1

これを常に 0 に設定しておくことで、実行する作業がない場合にインスタンスが時間を浪費することを避けられます。これを 0 より上に設定すると、その数の vCPU を常に維持する必要があります。

AWS Batch : 残念な点

- ・コンピューティング環境で指定したインスタンスタイプを超えるリソースでジョブを投げると、ジョブ滞留。
 - ・以下のように指定したコンピューティング環境を使うキューに、vCPU=16 のジョブを投げても反応なし。



- ・トラブルシューティングにも書いてはあるが、そもそも受け付けられないジョブに対しての警告はほしい！
 - ・ https://docs.aws.amazon.com/ja_jp/batch/latest/userguide/troubleshooting.html#job_stuck_in_runnable

リソースが不十分である

コンピューティングリソースが割り当てられるリソースを超えた CPU またはメモリーリソースがジョブ定義に指定されていると、ジョブは配置されません。たとえば、ジョブが 4 GiB のメモリーを指定し、コンピューティングリソースがそれ以下の場合、このジョブをこのコンピューティングリソースに配置することはできません。この場合、ジョブ定義に指定するメモリを減らすか、あるいは環境のコンピューティングリソースを追加する必要があります。

取り組み3

- EMRクラスタ常時起動
 - 日中帯は停止して、Glueへ。
 - 日中処理はHiveテーブルからファイルへETLしているだけ。Hiveテーブルはパーティショニング済みなので、Glueでも効率よく処理できそう。

<input type="checkbox"/> 名前 ↑三	最終更新日時 ↑三	サイズ ↑三	ストレージクラス ↑三
<input type="checkbox"/> p_ymd=20160701	--	--	--
<input type="checkbox"/> p_ymd=20160702	--	--	--
<input type="checkbox"/> p_ymd=20160703	--	--	--
<input type="checkbox"/> p_ymd=20160704	--	--	--
<input type="checkbox"/> p_ymd=20160705	--	--	--
<input type="checkbox"/> p_ymd=20160706	--	--	--
<input type="checkbox"/> p_ymd=20160707	--	--	--

未着手！これから進めて行きます！

取り組み4

- Hive メタストア
 - Glue Data Catalog へ。
 - と思ったのですが、以下の制約もあり、
リスク + 改修コスト > RDS 分のコスト削減 で メリットないので中止。
 - https://docs.aws.amazon.com/ja_jp/emr/latest/ReleaseGuide/emr-hive-metastore-glue.html
 - Hive シェルを使用した補助 JAR の追加はサポートされません。この問題を回避するには、hive-site 設定分類を使用して hive.aux.jars.path プロパティを設定します。これにより、Hive のクラスパスに補助 JAR が追加されます。
 - 述語式でユーザー定義関数 (UDF) を使用することはお勧めしません。Hive がクエリの実行を最適化しようとするため、クエリが失敗する場合があります。
Hive シェルから UDF の補助 JAR を追加して、述語式でも使っている。
 - Hive でのコストベースの最適化はサポートされていません。値 hive.cbo.enable から true への変更はサポートされていません。
**CBO は有効化しているが、統計情報全然取っていない。。。
けど CBO 無効と言われると怖い。**

EMR、Glue周りも道半ば。。。。

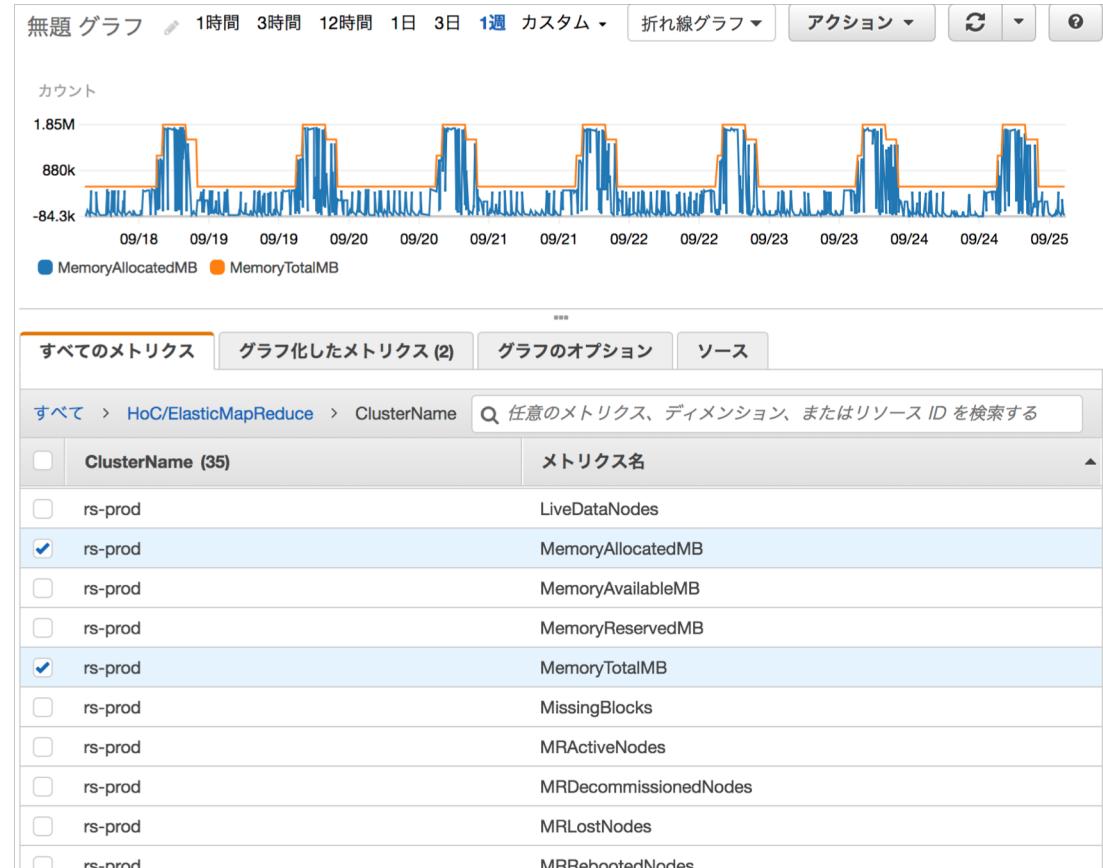
EMR : Tips

- メトリクス

- クラスタは定期的に再起動しているが、各種メトリクスは再起動前後(別クラスタ)を結合した形で見たい。
 - クラスタの設定変更等のした場合、前後の変動など捉えやすくしたい。

Lambdaで定期的にカスタムメトリクスへEMRメトリクスを保存。

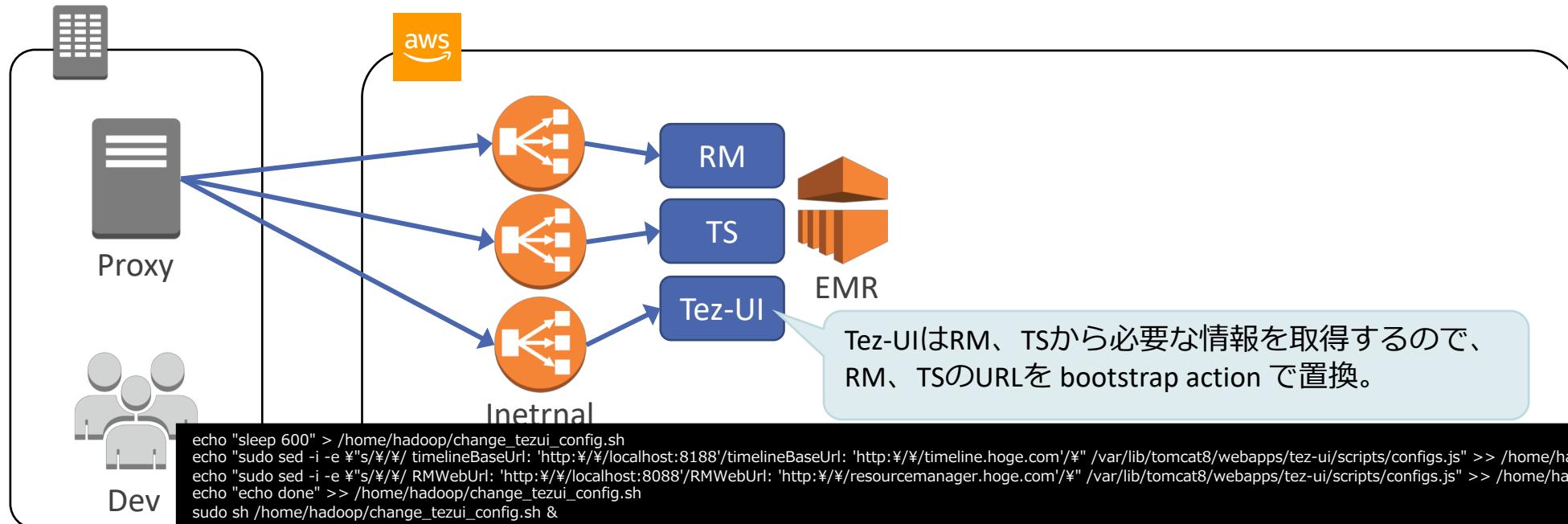
- クラスタ名毎に集約。全ノードの平均値 / 合計を保持。
- シンプルだが、傾向把握などにだいぶ役立っている。



EMR : Tips

- WEB UI
 - 社内NWからAWSへの接続は80/443のみ許可、という基本ルールがあり、マスターノード上のリソーススマネージャ、Tez-UIを見れない。

Lambdaで定期的にUIエンドポイントとELBを紐づけ。ELBへは社内リバプロンジ由で接続。これで見える。



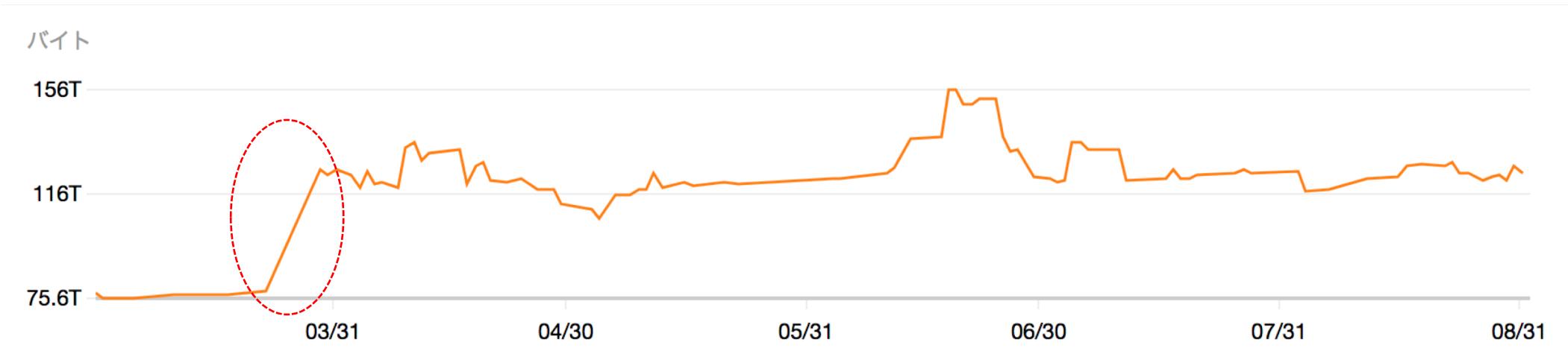
EMR : トラブル

- コアノードUnHealthy
 - コアノードにr3シリーズ、EC2ローカルストレージ+EBSをアタッチして利用。容量は以下の通りでアンバランス。
 - /mnt : 145GB
 - /mnt1 : 768GB
 - /mntが先に枯渇してUnHealthy発生。

ローカルストレージの無いr4シリーズに大きめのEBSアタッチすることで
UnHealthy解消。

EMR : トラブル

- S3容量爆増
 - Hive実行エンジンをMRからTezへ変更したら。



- 回避策はない。
- でも、今データはほぼ非圧縮のテキスト形式だから、parquet変換するだけでも結構減らせるし、処理効率も上がるはず。

EMR : トラブル

- 一部パーティション消失
 - Drop Table→Create Table→Insert overwriteで何故か一部パーティションが消える。
 - メタデータは存在し、実データだけ消える = 不整合でジョブエラー終了。

名前	バージョン ID	最終更新日時	サイズ	ストレージクラス
000050_0	4bqOZqQpchf0_nqYqiwwFlauf5XGq...		--	--
<input type="checkbox"/> 2018/09/10 13:49:18 (削除マーカー)	Dpa.p8WNcAhoptCPV3kiF4cM5CT0...		33.5 KB	スタンダード
<input type="checkbox"/> 2018/09/10 13:47:18	HUOzVHAx4sYNleJcwfnaskmFmHEi...		--	--
<input type="checkbox"/> 2018/09/10 13:45:40 (削除マーカー)	uq.xT4XRHFb0T.ETmvYUEqXr_DBL...		33.5 KB	スタンダード
<input type="checkbox"/> 2018/09/10 13:04:12				

EMR5.17.0より古いバージョンで

emrfs-site > fs.s3.consistent.fastList = true (デフォルトtrue)

の場合に稀に発生する事象。EMR5.17.0で修正されたバグ。

falseにすれば回避できることだが、性能影響はあります。

来週検証します。

fs.s3.consistent.fastList	true	このプロパティは、 true に設定されると、複数スレッドを使用してディレクトリを一覧表示します（必要な場合）。このプロパティを使用するには、整合性を有効にする必要があります。
---------------------------	------	---

取り組み5

- Lambdaコールドスタート、モデル提供の手間



未着手！これから進めて行きます！

レガシー脱却は道半ば。
引き続き進めて行きます。

ご清聴、ありがとうございました。