



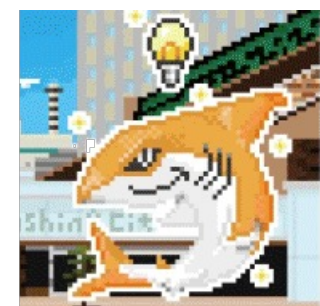
JAWS DAYS 2025

connecting the dots



恒例！AWSエンジニアたちの怒涛のLT

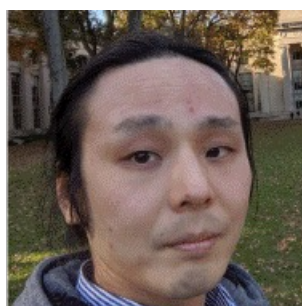
司会：荒巻 美南海 / 宇賀神 みずき（しょぼちむ）

**鄭 昌浩**

正しく理解してますか？
AWS グローバルインフラ
ストラクチャー

**菊地 信明**

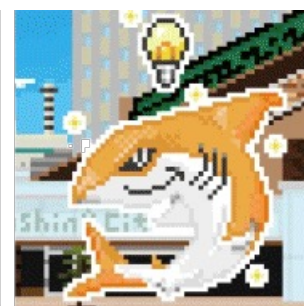
西日本から AWS クラウ
ドへつなぐときのお話

**高橋 尚久**

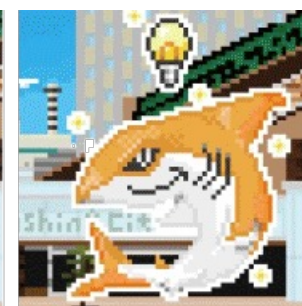
AWS サポートで見かけた
TLS の相互証明書認証の
問題を解決した小話

**山本 和司**

サポートエンジニアに
よる MWAA のトラブル
シュート

**疋田 宗太郎**

Python で Apache
Iceberg を操ろう！
3 分でわかる Pylceberg
の可能性

**朴 総明**

SMS 送信には注意を！
SMS Pumping 攻撃の
リスクと対応方法につい
て

**古野 俊広**

公式ドキュメントの
“トラブルシューティング”
ページを追加した話

**谷口 祐紀**

あなたはリージョン切替
ボタンを「ポチっとな」
したことがあるか？
～金融リファレンス
アーキテクチャ 日本版
ご紹介～

**成本 海斗**

Amazon Bedrock の
基盤モデル編プロンプト
エンジニアリングで
関西弁にカスタマイズ
してみた

**Sheng Hsia Leng**

Amazon Bedrock で
音楽作れるのか？！
Bedrock で実現する
Text-to-MIDI

**吉澤 巧**

Nova Canvas から始める
画像生成 AI の魅力入門

**奥村 美生**

JAWS な answer💡
JAWS が JAWS に
答えます！

**淡路 大輔**

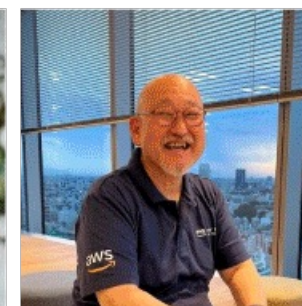
Bedrock Engineer
“あなた”の開発を支援す
る AI エージェント

**川路 義隆**

ナメック語を理解してい
ない SA は願いを叶える
ことができるのか

**高野 賢司**

AWS Step Functions は
CDK で書こう！

**福井 厚**

AppSync Events を
使ってみよう！



JAWS DAYS 2025
connecting the dots



正しく理解していますか？
AWSグローバルインフラストラクチャー

自己紹介

鄭 昌浩 (ちょん ちゃんほ)

仮の姿

AWSのスペシャリストソリューションアーキテクト
ハイブリッドエッジ担当

担当サービス

AWS Outposts、AWS LocalZones



お客様との会話あるある

基幹システム移行の検討状況いかがですか？

ち

客

クラウドへ移行は不安なんだよね？

どの辺りに不安があるんですか？

ち

客

信頼性が低そうだから・・・

どうしてそう思ったんですか？

ち

客

SLAが全体的に低いじゃん。

うちのプライベートクラウドは〇〇社と99.995%のSLA結んでるよ。

ち
あー、そうですか。

でも本来SLAと稼働率は違う物ですから。

客

それは分かってるけど、ちゃんと冗長化要件とか細かく出してるよ（ほれほれ）

ち
でもプライベートクラウドって、御社のデータセンターヶ所で運用してるんですよね。

客

そうだよ。うちのデータセンターは堅牢だから。

AWSさんよりも堅牢かもよ。

AWSさんもどこかのデータセンターにサーバーが置いてあるんでしょ。

ち
そう言われればそうですが、
AWSのグローバルインフラストラクチャをご存じですか？

客

グローバルインフラストラクチャ？

ちゃんと聞いたことないね。

ち
そうですか。

AWSのデータセンタは、
リージョンという地域の単位に分かれてるんです。

客

あー、東京リージョンとか大阪リージョンでしょ。

二ヶ所にデータセンターがあるんだよね。

いえ。

ち

客

えっ、2ヶ所じゃないの？

それで東西DRしてるんでしょ。

リージョンは一つのデータセンターではないんです。

ち

その中にまた最低3つのアベイラビリティゾーン、通称AZというグループに分かれてるんです。

客

じゃあ、リージョンは3つのデータセンタでできてるってこと。

いえ、AZも複数のデータセンターでできてるんです。

ち

ちなみにAZ間は少なくとも100km以上離してあります。

客

データセンター分かれちゃったら、
その間で通信しなくちゃいけないじゃん。

そうです。

ち

AZ内はもちろん、AZ間でも、数msec以下で通信できるんです。

客

へえ。

その中に同じシステムのサーバーを
分散して配置する事ができるんですよ。

ち

客

サーバー間の通信設計が大変じゃん。

いえ。大変じゃないです。

ち

同じリージョンのAZ間には特に意識することなく通信できます。

ち

本当に一つのリソースプール環境に見えるんです。

客

じゃあ、複数のデータセンタに分散したシステム環境が簡単に作れるってこと。

はい、そうです。

ち

それでもプライベートクラウドより信頼性低いですかね。

客

そうだねー。

ちょっと検討します。

ここにいらっしゃる皆さんはそんなこと無いと思いますが、
先入観にとらわれないようにしましょうね。

以上、グローバルインフラストラクチャあるあるでした。

JAWS DAYS 2025

恒例！**AWS**のエンジニアたちの怒濤の**LT**

西日本から AWSクラウドへ つなぐときのお話

菊地 信明

アマゾン ウェブ サービス ジャパン合同会社
シニア ソリューション アーキテクト
ネットワーク スペシャリスト



こんな人がお話します

菊地 信明 (きくち のぶあき)

アマゾン ウェブ サービス ジャパン合同会社

サービス&テクノロジー事業統括本部

ネットワークソリューション本部

シニア ソリューション アーキテクト

ネットワーク スペシャリスト



経歴： 通信キャリアにてホスティングやマネージドFWのサポートを経験
鉄道系IT子会社にて設計・開発・運用に従事
AWSサポートにてDirect Connect/VPNのサポートを対応

好きなAWSサービス：もちろん、AWS Direct Connect



今日話すこと

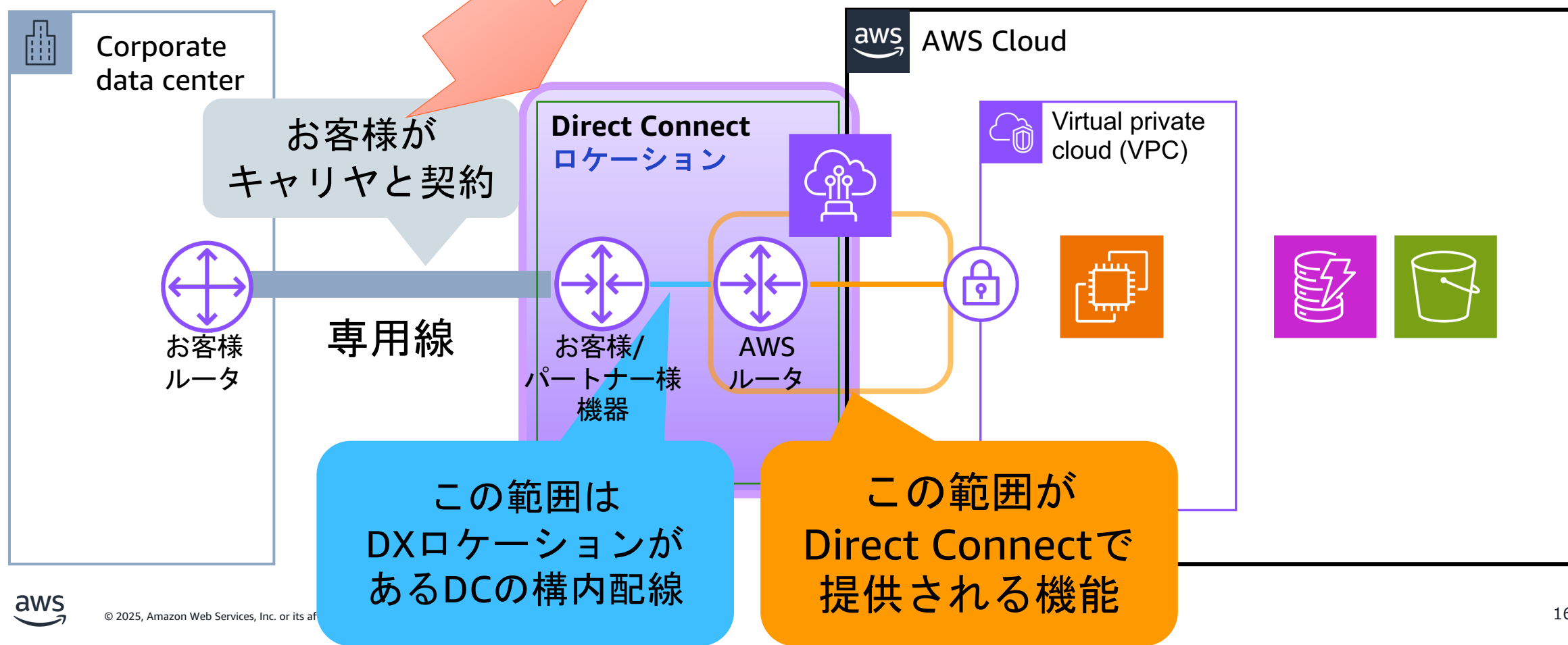
西日本で利用する AWS Direct ConnectのPoPについて

- PoP = Point of Presence = 接続拠点（出入口）
- AWS Direct Connectでは、
『Direct Connect□ケーシヨン』
と呼びます

本題に入る前に：AWS Direct Connect とは？

AWS Direct Connectは【**オンプレミスと直接接続する専用線サービスではありません**】

お客様がキャリアから調達する専用線の片端とAWS Cloudを、Direct Connectロケーションで接続するサービスです



みんなだいすき、クイズのお時間

Q: 日本のリージョンにDirect Connectロケーションは
いくつありますか？

A: 3 つ

B: 5 つ

C: 7 つ

A、B、Cのどれかで挙手してください！



みんなだいすき、クイズのお時間

Q: 日本のリージョンにDirect Connectロケーションは
いくつありますか？

A: 3 つ

B: 5 つ

C: 7 つ

A、B、Cのどれかで挙手してください！



みんなだいすき、クイズのお時間

Q: 日本のリージョンにDirect Connectコネクションはいくつありますか？

A: 3つ

B: 5つ

C: 7つ ← **正解**

A、B、Cのどれかで挙手してください！

正解は・・・

Q: **日本のリージョン**にDirect Connectロケーションは
いくつありますか？

C: 7つ

アット東京 CC1

エクイニクス TY2

エクイニクス OS1



正解は・・・

Q: **日本のリージョン**にDirect Connectロケーションは
いくつありますか？

C: 7つ

アット東京 CC1

エクイニクス TY2

エクイニクス OS1

Chief Telecom LY 台湾台北 (チーフテレコム)

Chunghwa Telecom 台湾台北 (チュンファテレコム)



正解は・・・

Q: 日本のリージョンにDirect Connectロケーションはいくつありますか？

C: 7つ

アット東京 CC1

エクイニクス TY2

エクイニクス OS1

Chief Telecom LY 台湾台北 (チーフテレコム)

Chunghwa Telecom 台湾台北 (チュンファテレコム)



NEC INZAI
2023/05~



KDDI
TELEHOUSE
OSAKA 2
2024/12~

正解は・・・

内訳：東京リージョン 6 + 大阪リージョン 1 = 7

C: 7つ

アット東京 CC1

エクイニクス TY2

エクイニクス OS1

Chief Telecom LY 台湾台北 (チーフテレコム)

Chunghwa Telecom 台湾台北 (チュンファテレコム)



NEC INZAI
2023/05~



KDDI
TELEHOUSE
OSAKA 2
2024/12~



大阪市に新しい AWS Direct Connect ロケーションが開設

投稿日: 2024年12月13日

AWS は本日、大阪にある TELEHOUSE 大阪 2 データセンターに、新しい [AWS Direct Connect](#) ロケーションを開設しました。この新しい大阪のロケーションでネットワークを AWS に接続すると、すべてのパブリック AWS リージョン (中国リージョンを除く)、AWS GovCloud リージョン、[AWS Local Zones](#) にプライベートで直接アクセスできます。このサイトは、大阪圏では 2 番目、日本では 5 番目の AWS Direct Connect サイトです。

Direct Connect サービスは、AWS とお客様のデータセンター/オフィス/コロケーション環境との間に、プライベート物理ネットワーク接続を確立します。このプライベート接続では、パブリックインターネット経由の接続よりも安定したネットワークエクスペリエンスを提供できます。TELEHOUSE 大阪 2 の新しい Direct Connect ロケーションでは、MACsec 暗号化を利用可能な専用の 1 Gbps、10 Gbps、および 100 Gbps の接続を提供します。

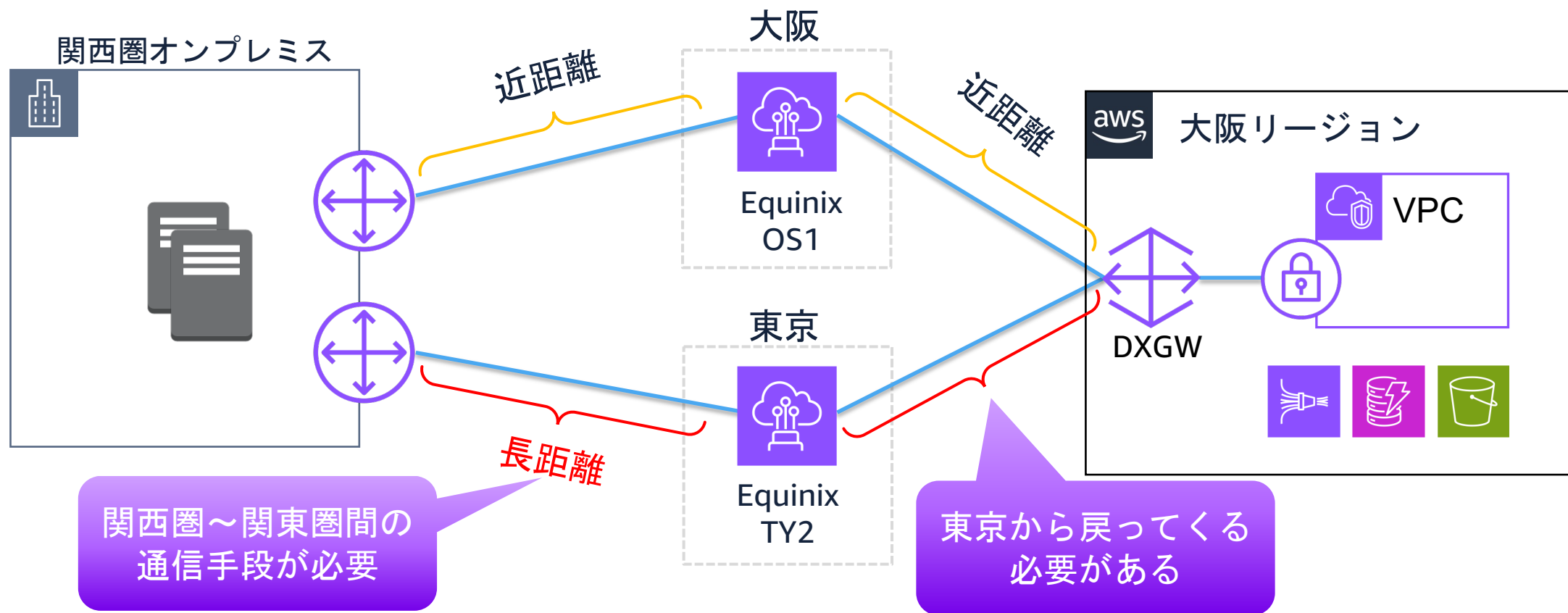
世界中の 145 を超える Direct Connect ロケーションの詳細については、Direct Connect [製品詳細ページ](#)の[ロケーション](#)のセクションをご覧ください。または、[使用開始方法](#)のページで、Direct Connect の購入およびデプロイの方法の詳細をご覧ください。





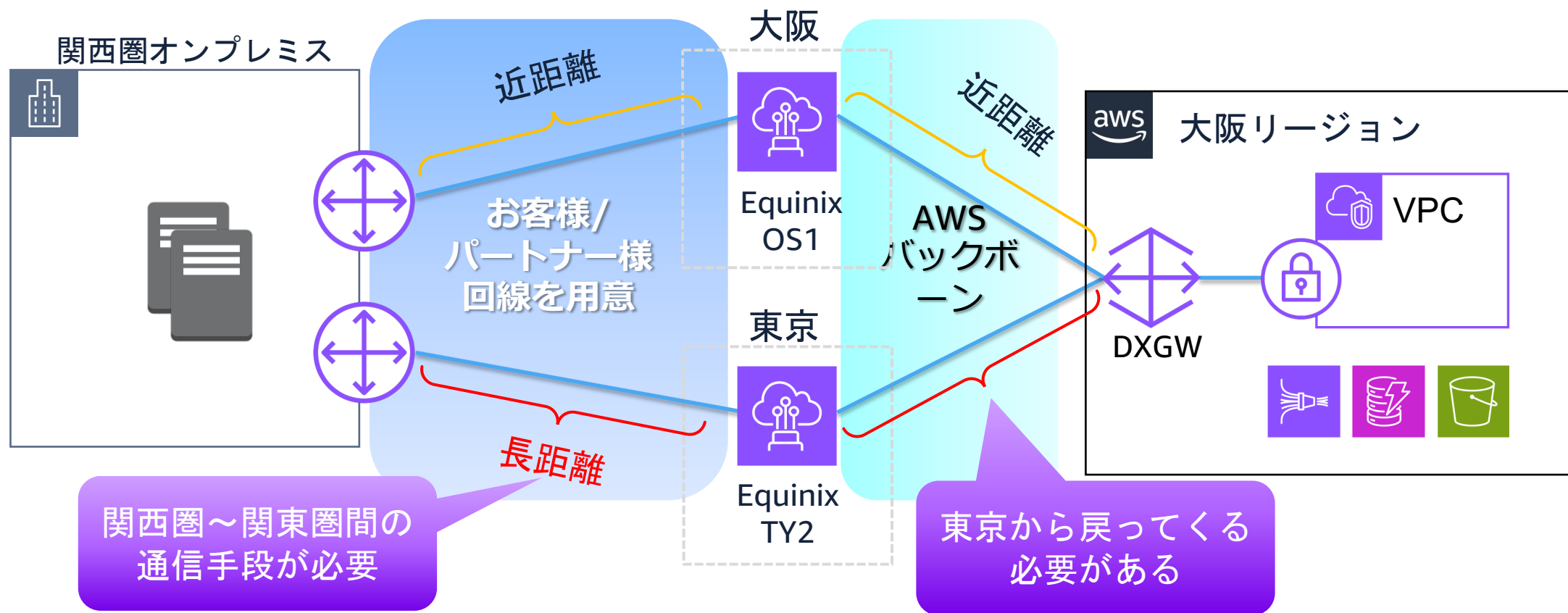
デュアルロケーション (東京・大阪で分散)

これまでの東京と大阪に分散したDirect Connectロケーションを用いた冗長構成の例



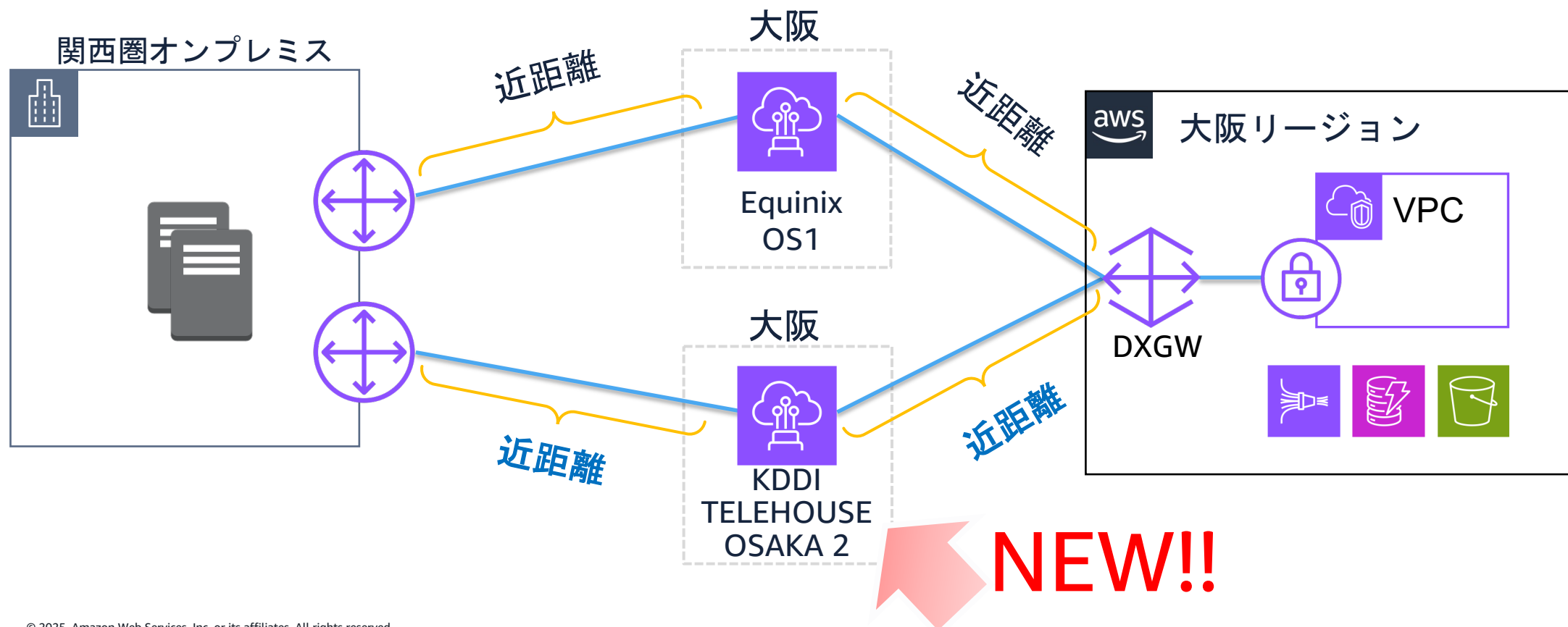
デュアルロケーション（東京・大阪で分散）

これまでの東京と大阪に分散したDirect Connectロケーションを用いた冗長構成の例



デュアルロケーション（大阪近郊で分散）

今後は大阪圏内で分散したDirect Connectロケーションを用いた冗長構成が可能
メイン回線、バックアップ回線ともに、最適なレイテンシーで閉域網接続！



ところで・・・

DR戦略におけるAWS大阪リージョンの役割

東京リージョンと大阪リージョンのどちらをメインに利用したほうが良いか？っというご質問を多くいただきます。



参考資料を公開してます！

大阪リージョン最新情報とよくある質問

https://pages.awscloud.com/rs/112-TZM-766/images/2_20230511-Osaka_Osaka-region.pdf



【開催報告＆動画公開】大阪フルリージョン 2 周年！
最新情報からメインリージョンとしての活用事例まで教えます！

<https://aws.amazon.com/jp/blogs/news/osaka-region-2nd-anniversary/>



やや強引なまとめ

西日本からAWSクラウドへ閉域網接続するとき、

**「関西にある二つのDirect Connectロケーションに
つなぎましょうねっ！」**

これだけ、覚えておいてください～

Thank you!

菊地 信明

アマゾン ウェブ サービス ジャパン合同会社
シニア ソリューション アーキテクト
ネットワーク スペシャリスト



自己紹介

高橋 尚久（なおひさ）

シニア クラウドサポートエンジニア

Amazon S3, Amazon CloudFront, Amazon Athena SME(Subject Matter Expert)

日本のメーカー、Web系のスタートアップに勤務した後にAWS Japanに入社。

入社1年半でアイルランドのダブリンに異動し、4年半ほど勤務した後、日本に帰国。

普段の業務

お客様からAWSサポートへのお問合せについて対応し、問題の解決をお手伝いしています。

また、シニアエンジニアとして組織のプロセスの改善や他のエンジニアへのトレーニング、技術的な支援も行っています。

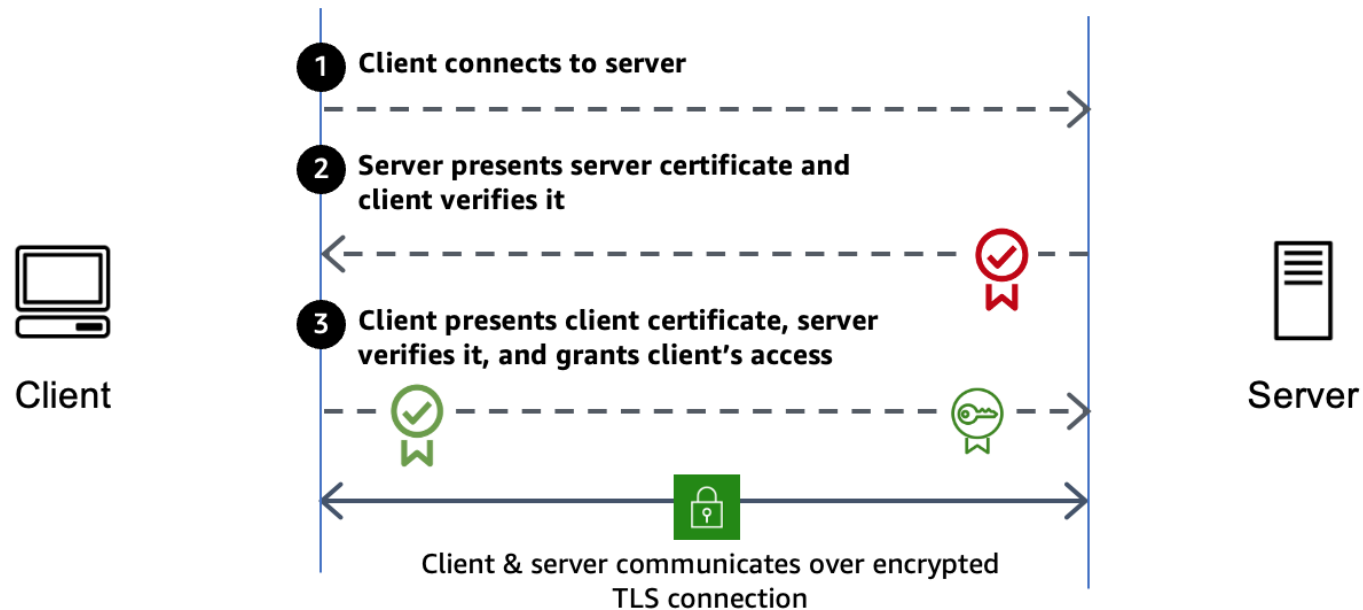
好きなAWSサービス

Amazon S3、Amazon Athena

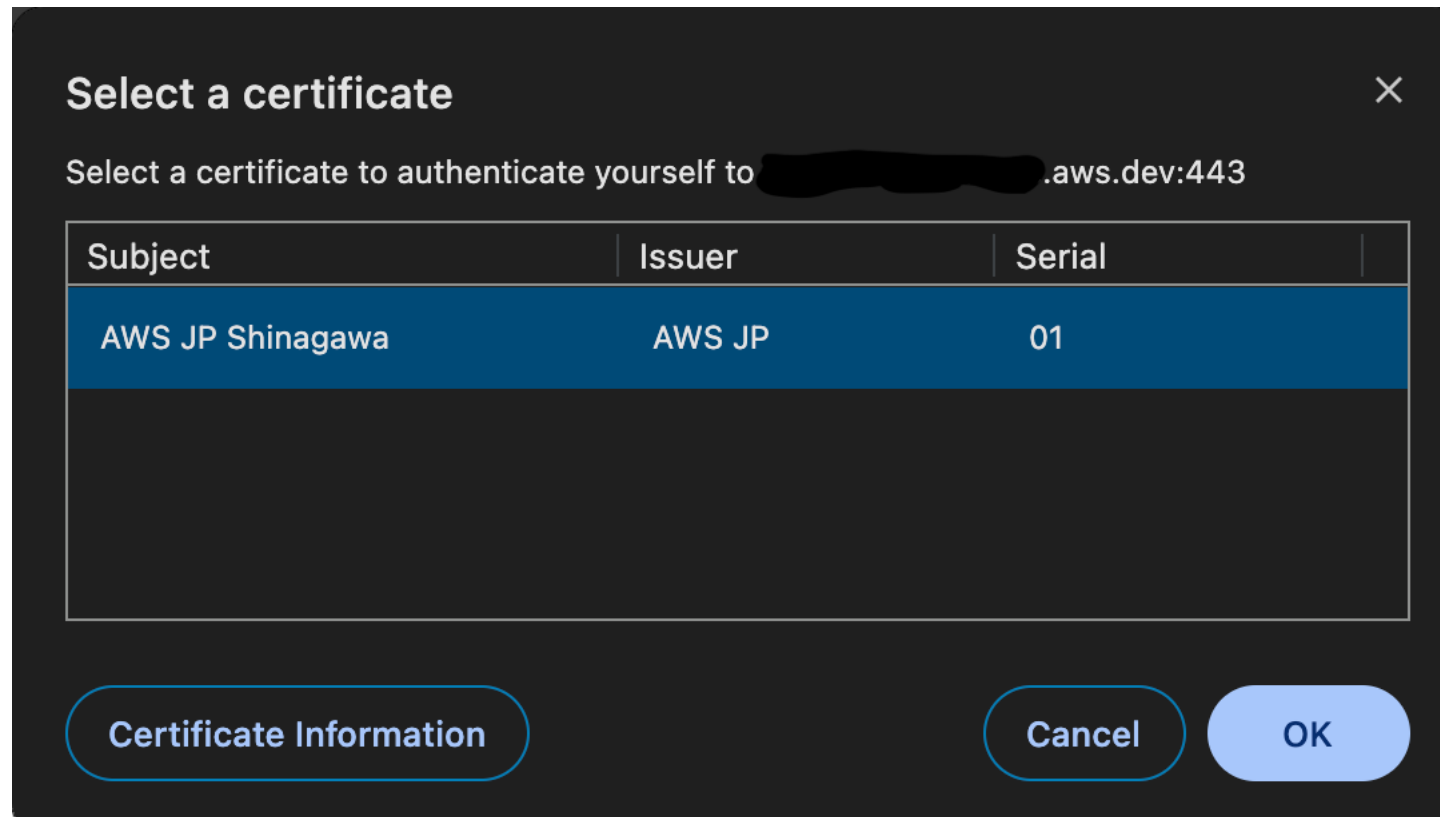


AWSサポートで見かけたTLSの相互証明書 認証の問題を解決した小話

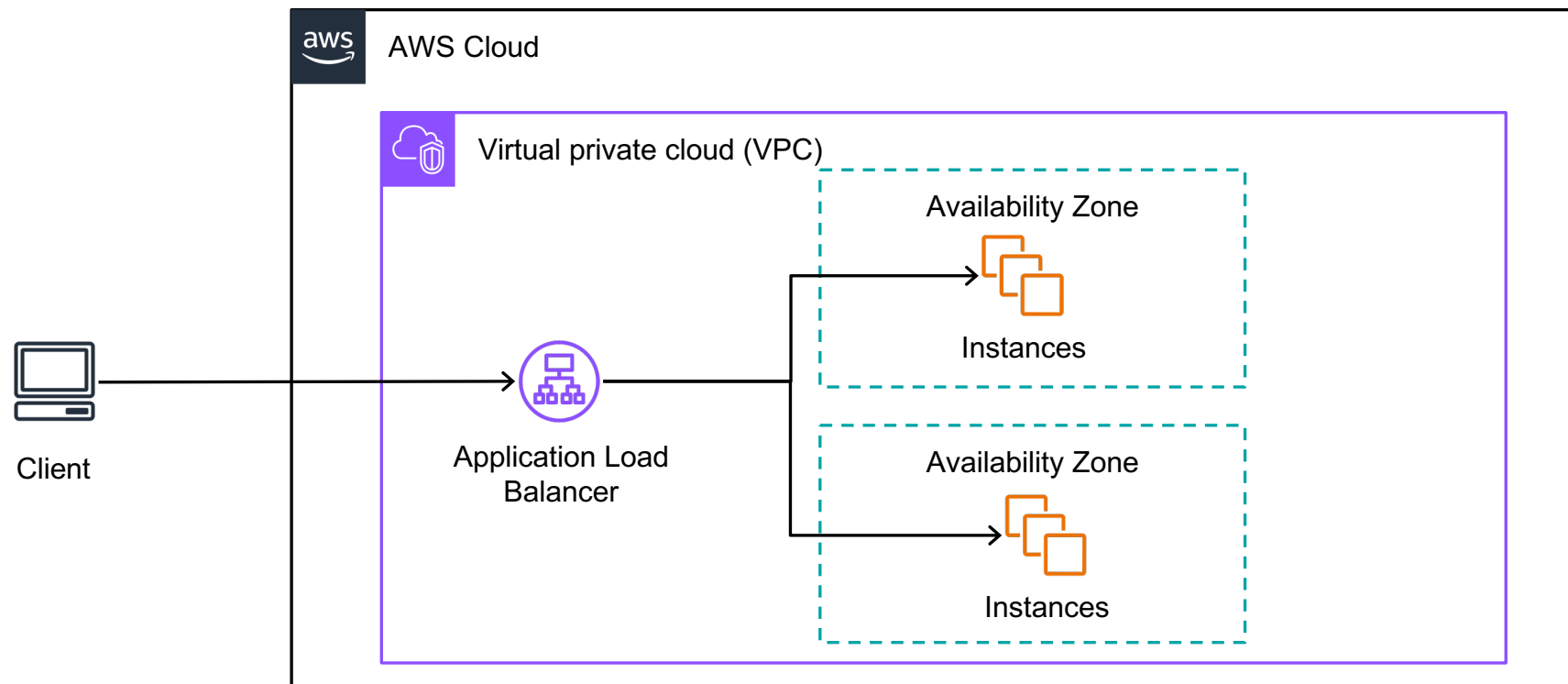
ALBにはTLSによる相互認証（mTLS）の機能があります。
クライアントから証明書情報をALBに送信して、信頼できるクライアントだけが通信できるようにする機能です。



お客様はALBでこのmTLS認証を使用しているものの、対象のWebサイトにブラウザでアクセスする際に証明書の選択のポップアップが表示されてしまい、このポップアップを表示させずに使用できるようにしたいというご要望。



システム構成としては、クライアントからのリクエストをALBで複数のEC2インスタンスに分散する形式。



RFC8446に“certificate_authorities”拡張についての記述があり、TLSでの「Certificate Request」のタイミングでこの機能が使用され、サーバー側から対象の証明書のCAが提示される形になる。

ALBは、当時この機能に対応していなかった。

4.2.4. Certificate Authorities

The "certificate_authorities" extension is used to indicate the certificate authorities (CAs) which an endpoint supports and which SHOULD be used by the receiving endpoint to guide certificate selection.

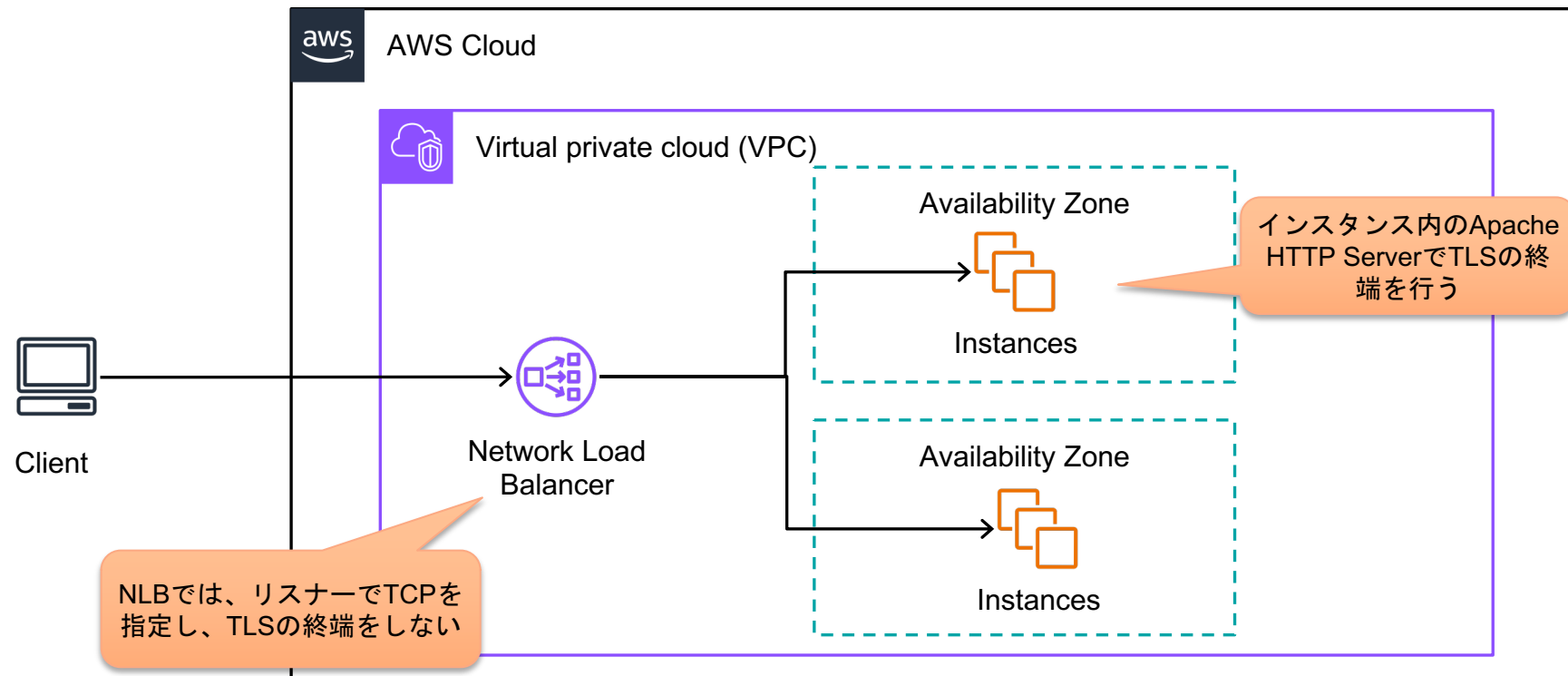
<https://datatracker.ietf.org/doc/html/rfc8446#section-4.2.4>

制限として、

1. AWSのサービスであるALBが対応していない状況。
2. ブラウザの種類によっては設定で回避できるものだったが、クライアント数が数千と多く、設定で回避できないブラウザも使用されていた状況。

どのように解決したかというところ・・・。

ALBでは対応できないので、発想の転換をした。
ロードバランサーとして、ALBの代わりにNLBを使用し、バックエンドのWeb Server側でTLSを終端する方法をご提案した。



Apache HTTP Serverには、mod_sslに「SSLCADNRequestFile」ディレクティブが存在し、SSL/TLSのハンドシェイク時にクライアントに対して受け入れ可能な認証局(CA)の名前を提示することができる。

SSLCADNRequestFile Directive

When a client certificate is requested by mod_ssl, a list of acceptable Certificate Authority names is sent to the client in the SSL handshake. These CA names can be used by the client to select an appropriate client certificate out of those it has available.

https://httpd.apache.org/docs/2.4/mod/mod_ssl.html#sslcadnrequestfile

```
# ssl.conf
SSLCACertificateFile /etc/httpd/client_certs/ca.crt
SSLCADNRequestFile /etc/httpd/client_certs/user.crt
SSLVerifyClient require
SSLVerifyDepth 2
```

TLSv1.3 Record Layer: Handshake Protocol: Certificate Request

Opaque Type: Application Data (23)

Version: TLS 1.2 (0x0303)

Length: 161

[Content Type: Handshake (22)]

Handshake Protocol: Certificate Request

Handshake Type: Certificate Request (13)

Length: 140

Certificate Request Context Length: 0

Extensions Length: 137

Extension: signature_algorithms (len=34)

Type: signature_algorithms (13)

Length: 34

Signature Hash Algorithms Length: 32

Signature Hash Algorithms (16 algorithms)

Extension: certificate_authorities (len=95)

Type: certificate_authorities (47)

Length: 95

Distinguished Names Length: 93

Distinguished Names (93 bytes)

Distinguished Name Length: 91

RDNSSequence item: 1 item (id-at-countryName=JP)

RDNSSequence item: 1 item (id-at-stateOrProvinceName=Tokyo)

RDNSSequence item: 1 item (id-at-localityName=Shinagawa)

RDNSSequence item: 1 item (id-at-organizationName=AWS JP)

RDNSSequence item: 1 item (id-at-organizationalUnitName=hoge)

無事、この問題の解決には成功した。しかし、その後2024年11月22日にALBの認証局のアドバタイズの機能が追加された。

数日違いでALBだけで対応できるようになってしまった。

AWS Application Load Balancer が相互の TLS を使用する際のクライアントの動作を簡素化するために認証局のアドバタイズを導入

投稿日: 2024年11月22日

Application Load Balancer (ALB) は、証明書の選択を簡素化するために、関連するトラストストアに保存されている認証局 (CA) サブジェクト名のアドバタイズをサポートするようになりました。この機能を有効にすると、ALB はロードバランサーに接続しようとしているクライアントに CA サブジェクト名のリストを送信します。クライアントはこのリストを使用して、どの証明書が ALB に受け入れられるかを識別でき、相互認証中の接続エラーが減少します。

<https://aws.amazon.com/jp/about-aws/whats-new/2024/11/aws-application-load-balancer-certificate-authority-advertisement/>



JAWS DAYS 2025
connecting the dots



サポートエンジニアによる MWAA のトラブルシューティング

Cloud Support Engineer
Kazushi Yamamoto

ハッシュタグ : #jawsdays2025 #jawsug #jawsdays2025_c

自己紹介

山本 和司(やまもと かずし)

- ・ Cloud Support Engineer
- ・ 分散処理、AI/ML 関連のサポートケース対応
- ・ JAWS での発表初めてです！



Apache Airflow を AWS 上でマネージドに利用できるサービスです！

```
from airflow import DAG
from datetime import datetime
from airflow.operators.bash_operator import BashOperator
import os

dag_id = os.path.basename(__file__).replace(".py", "")

default_args = {
    'owner': 'aws',
    'depends_on_past': False,
    'retries': 0
}
```

```
dag = DAG(
    dag_id=dag_id,
    default_args=default_args,
    schedule_interval=None,
    start_date=datetime(2025,1,1),
    catchup=False
)
```

```
t1 = BashOperator(
    task_id='JAWS',
    bash_command='echo JAWS',
    dag=dag
)
```

```
t2 = BashOperator(
    task_id='DAYS',
    bash_command='echo DAYS',
    dag=dag
)
```

```
t3 = BashOperator(
    task_id='2025',
    bash_command='echo 2025',
    dag=dag
)
```

```
t1 >> t2 >> t3
```

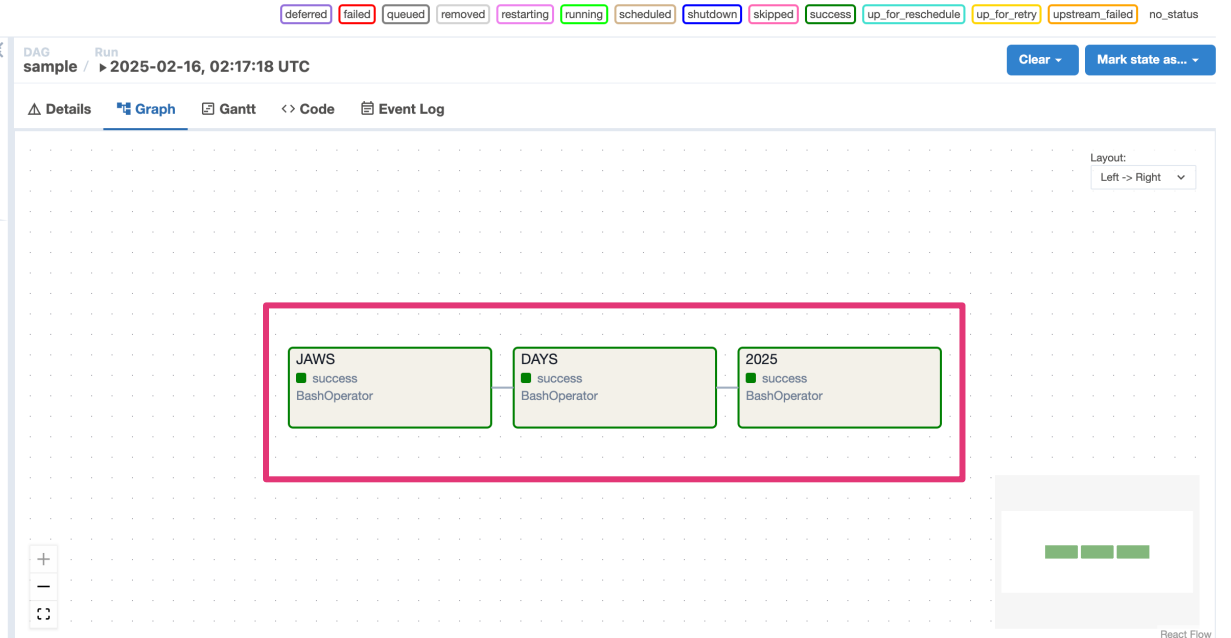
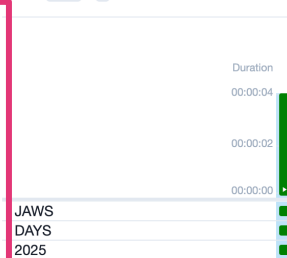


タスク: 処理を定義したもの



DAG: タスクの実行順序などを定義したもの

Press **shift** + **/** for Shortcuts



ハッシュタグ : #jawsdays2025 #jawsug #jawsdays2025_c

MWAA のお問合せ



タスクがエラーとなっていた原因と対策を知りたい。

トラブルシューティングをするには
どのような観点での調査が必要??

調査が必要な観点

- MWAA 観点
- タスクの接続先の観点
- Apache Airflow 観点

調査が必要な観点

- ・ タスクのスケジューリング
- ・ タスクのステータス
- ・ MWAA リソースの負荷
- ・ メンテナンスウィンドウ

- MWAA 観点
- タスクの接続先の観点
- Apache Airflow 観点

- ・ requirements.txt や plugins.zip
- ・ 接続先のリソースの状況
- ・ DAG、タスクの定義方法
- ・ Airflow Configuration

MWAA 環境で出力するログは INFO レベルで出力すると、タスクがエラーになるまでの時系列が追いやすくなります

ログタイプ

Amazon MWAA は、有効にした Airflow ログ記録オプションごとにロググループを作成し、環境に関連付けられた CloudWatch ロググループにログをプッシュします。ロググループの名前は以下の形式に従います: `YourEnvironmentName-LogType` たとえば、お使いの環境に `Airflow-v202-Public` という名前が付けられている場合、Apache Airflow タスクログは `Airflow-v202-Public-Task` に送信されます。

ログタイプ	説明
<code>YourEnvironmentName-DAGProcessing</code>	DAG プロセッサマネージャー (DAGファイルを処理するスケジューラの一部) のログ。
<code>YourEnvironmentName-Scheduler</code>	Airflow スケジューラーが生成するログ。
<code>YourEnvironmentName-Task</code>	タスクは、 がDAG生成する をログに記録します。
<code>YourEnvironmentName-WebServer</code>	Airflow ウェブインターフェイスが生成するログ。
<code>YourEnvironmentName-Worker</code>	ワークフローとDAG実行の一部として生成されたログ。

MWAA

Point!!

出力するログは **INFO レベル** で出力すると、タスクが
完了までの時系列が追いやすくなります

ログタイプ

Amazon MWAA は、有効にした Airflow ログ記録オプションごとにロググループを作成し、環境に関連付けられた CloudWatch ロググループにログをプッシュします。ロググループの名前は以下の形式に従います: `YourEnvironmentName-LogType` たとえば、お使いの環境に `Airflow-v202-Public` という名前が付けられている場合、Apache Airflow タスクログは `Airflow-v202-Public-Task` に送信されます。

ログタイプ	説明
<code>YourEnvironmentName-DAGProcessing</code>	DAG プロセッサマネージャー (DAGファイルを処理するスケジューラの一部) のログ。
<code>YourEnvironmentName-Scheduler</code>	Airflow スケジューラーが生成するログ。
<code>YourEnvironmentName-Task</code>	タスクは、 がDAG生成する をログに記録します。
<code>YourEnvironmentName-WebServer</code>	Airflow ウェブインターフェイスが生成するログ。
<code>YourEnvironmentName-Worker</code>	ワークフローとDAG実行の一部として生成されたログ。

Apache Airflow 2.7.2

[[34m**2025-02-26T09:55:31.126**+0000[0m]

{{[34mbase_executor.py:[0m144}}} INFO[0m - Adding to queue: ['airflow', 'tasks', 'run', 'sleep_dag', 'task1', 'manual__2025-02-26T09:55:30.251187+00:00', '--local', '--subdir', 'DAGS_FOLDER/sleep_dag.py']

約30分後にfailed

[[34m**2025-02-26T10:28:49.384**+0000[0m]

{{[34mscheduler_job_runner.py:[0m1565}}} WARNING[0m - Marked the following 9 task instances stuck in queued as failed. If the task instance has available retries, it will be retried.

Apache Airflow 2.7.2

[[34m2025-02-26T09:55:31.126+0000[0m]

['flow',

大量のタスクを同時に実行しようとしており、
処理しきれなかった

{{[34mscheduler_job_runner.py:[0m1565}} WARNING[0m - Marked the following 9 task instances stuck in queued as failed. If the task instance has available retries, it will be retried.

[[34m2025-02-26T09:55:31.126+0000[0m]]

flow',

なぜ30分後に failed となる…?

```
{{[34mscheduler_job_runner.py:[0m1565}}} WARNING[0m - Marked the following 9 task instances stuck in queued as failed. If the task instance has available retries, it will be retried.
```

task_queued_timeout

New in version 2.6.0.

Amount of time a task can be in the queued state before being retried or set to failed.

Type

float

Default

600.0

Environment Variable

AIRFLOW__SCHEDULER__TASK_QUEUED_TIMEOUT

10分(600秒)の間、queued のステータスであれば、
failed のステータスに変更する

```
from airflow import DAG
from airflow.operators.bash import BashOperator
from datetime import datetime
import os

dag_id = os.path.basename(__file__).replace(".py", "")

with DAG(
    dag_id=dag_id,
    start_date=datetime(2022,1,1),
    catchup=False,
    schedule_interval=None,
) as dag:
    t0 = BashOperator(
        task_id='config_list',
        bash_command='airflow config list --color off -V -s -v',
    )
```

```
# Variable: AIRFLOW__SCHEDULER__ALLOW_TRIGGER_IN_FUTURE
#
allow_trigger_in_future = False
#
# Source: airflow.cfg
#
# Variable: AIRFLOW__SCHEDULER__TRIGGER_TIMEOUT_CHECK_INTERVAL
#
trigger_timeout_check_interval = 15
#
# Source: env var
#
# Variable: AIRFLOW__SCHEDULER__TASK_QUEUED_TIMEOUT
#
task_queued_timeout = 1800.0
#
# Source: airflow.cfg
#
# Variable: AIRFLOW__SCHEDULER__TASK_QUEUED_TIMEOUT_CHECK_INTERVAL
#
task_queued_timeout_check_interval = 120.0
#
# Source: airflow.cfg
#
# Variable: AIRFLOW__SCHEDULER__ALLOWED_RUN_ID_PATTERN
#
allowed_run_id_pattern = ^[A-Za-z0-9_~:+-]+$
```

2025年2月時点での MWAA ではtask_queued_timeout の値は 1800.0(30分) となっていることを確認

ログ や MWAA、Apache Airflow のドキュメントを確認する
大切性が伝われば幸いです。

エラーの原因が特定できない場合には、下記の情報を添えて
AWS サポートまでお問合せください！

- ・ 発生時刻
- ・ エラーが発生しているDAG名、タスク名
- ・ CloudWatch ログ
- ・ GetEnvironment API の実行結果

ご清聴ありがとうございました

Python で Apache Iceberg を操ろう！ 3 分でわかる PyIceberg の可能性

疋田 宗太郎 (Sotaro Hikita)

アマゾン ウェブ サービス ジャパン合同会社

Solutions Architect

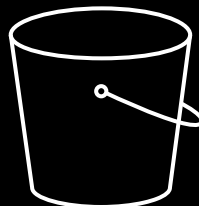


Iceberg: データレイクのための次世代テーブルフォーマット

データレイクの限界を超えて、様々なデータエコシステムを繋ぐ



Apache Iceberg

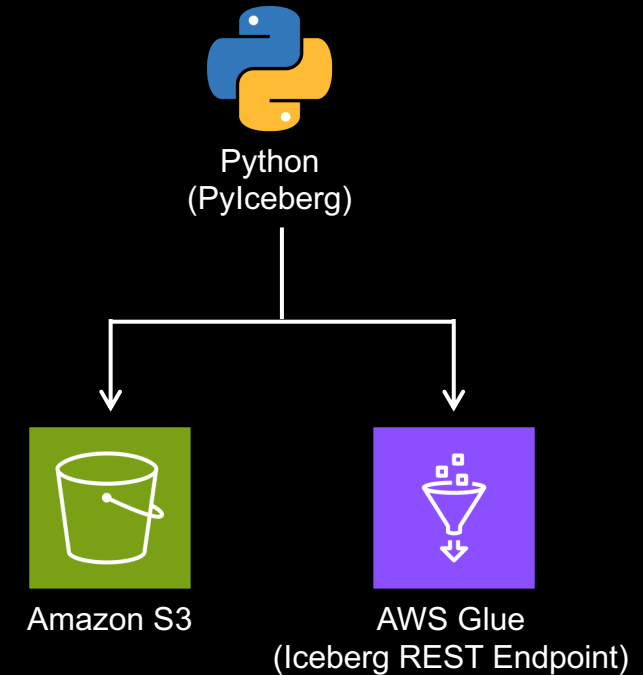


PyIceberg で Python から Iceberg を扱おう

PyIcebergは、IcebergをPythonから操作するためのライブラリです！

- Pythonだけでテーブル操作 OK
- Amazon S3 + AWS Glue Data Catalogでサーバレスに構成できる
- もちろんSparkなどのビッグデータエンジンとも共存可能

小さなPoCからビッグデータまで、
1つのフォーマットでシームレスに扱える！



▼ ライブラリのインストール

```
[ ]: !pip install pyiceberg
     !pip install 'polars[pyiceberg]'
```

Iceberg カタログ(Glue Iceberg REST Endpoint)のロード

```
[ ]: from pyiceberg.catalog import load_catalog
     account_id = "497197612387"
     region = "ap-northeast-1"
     s3_bucket = "hsotaro-497197612387-ap-northeast-1"
     prefix = "iceberg"

     catalog_properties = {
         "type": "rest",
         "uri": "https://glue.ap-northeast-1.amazonaws.com/iceberg",
         "warehouse": account_id,
         "rest.sigv4-enabled": "true",
         "rest.signing-name": "glue",
         "rest.signing-region": region
     }
     catalog = load_catalog(account_id, **catalog_properties)
```

データベースを作成

```
[ ]: namespace = "jawsdaysdemov"
```

```
[ ]: catalog.create_namespace(namespace)
```

テーブルを作成

テスト用データ(ローカル上の parquet ファイル)

数行のコードで、Iceberg テーブルを作成！

```
import pandas as pd
import pyarrow as pa
from pyiceberg.catalog import load_catalog

# サンプルデータを作成
df = pd.DataFrame({
    "id": [1, 2, 3],
    "name": ["Alice", "Bob", "Carol"]
})
arrow_table = pa.Table.from_pandas(df)

catalog = load_catalog(catalog = load_catalog(<account_id>, "type":
"rest", "uri": "https://glue.ap-northeast-1.amazonaws.com/iceberg",
"warehouse": (<account_id>, "rest.sigv4-enabled": "true", "rest.signing-
name": "glue", "rest.signing-region": <region>)

# テーブルを作成し、サンプルデータを追加
iceberg_table = catalog.create_table(
    identifier="mydb.mytable",
    location="s3://<YOUR_BUCKET>/iceberg",
    schema=arrow_table.schema)
iceberg_table.append(arrow_table)
```

Polars で SQL も書ける！

```
import polars as pl
iceberg_lf = pl.scan_iceberg(table)
ctx = pl.SQLContext({table_name: iceberg_lf})

result = ctx.execute(f"""
SELECT *
FROM {table_name}
""", eager=True)
print(result)

result = ctx.execute(f"""
SELECT VendorID, COUNT(*) as count
FROM {table_name}
GROUP BY VendorID
""", eager=True)
print(result)
```

どんな環境からでも気軽にテーブル管理

ローカル、Lambda、Notebook… Python が動けばどこでも使える

いろんな実行環境からIcebergテーブルにアクセスできる！

- ローカルPC → 検証や開発用の小スクリプト
- Lambda → サーバレスで定期ジョブ or イベント駆動の更新
- Notebook (SageMaker, Colab等) → 分析・MLパイプラインに活用

まとめ

まずは `pip install pyiceberg` して触ってみよう！

表構造をスマートに永続化・分析したいニーズで、PyIceberg が活躍できるかも

Iceberg: データレイクのための次世代テーブルフォーマット

PyIceberg: PythonだけでIcebergテーブルを管理・操作できる



JAWS DAYS 2025

connecting the dots



SMS 送信には注意を！ SMS Pumping 攻撃のリスクと対応方法について

Chongmyung Park

ハッシュタグ：#jawsdays2025 #jawsug #jawsdays2025_c

自己紹介

Chongmyung Park(朴 総明/パク チョンミョン)



Solutions Architect

好きなAWSサービス

AWS Amplify、Amazon DynamoDB

保有資格



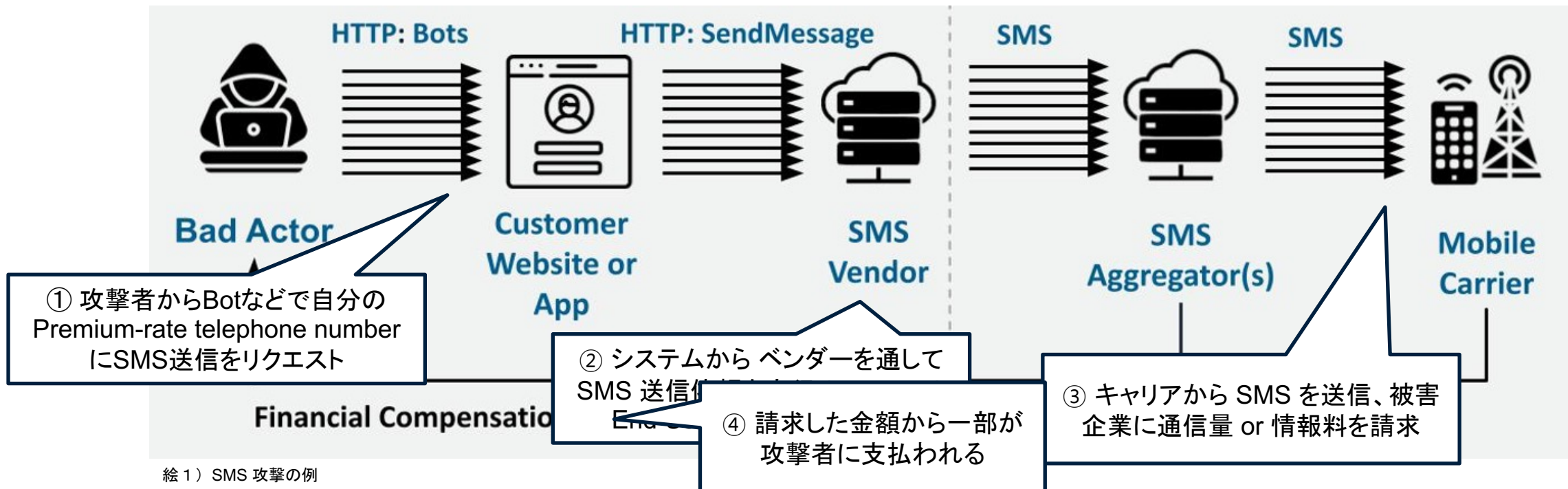
ハッシュタグ：#jawsdays2025 #jawsug #jawsdays2025_c

SMS Pumping 攻撃とは？

「Premium-rate telephone number」と呼ばれる番号宛てに SMS を送るようリクエストを乱発、利益を稼ぐ攻撃

よく攻撃されるターゲット：one-time passwords (OTPs), ダウンロードリンク, 招待リンク送信など

* **Premium Rate Number**：情報やエンターテインメントなどの特別なサービスに対して、通常より高い料金を発信者に請求する電話番号



絵 1) SMS 攻撃の例

参考) <https://aws.amazon.com/ko/blogs/messaging-and-targeting/lower-your-risk-of-sms-fraud-with-country-level-blocking-and-amazon-pinpoint/>

無駄なコスト増加、サービス停止のリスクがある

1) コスト増加

大量の SMS 送信 = 大量の送信コスト発生

2) サービス停止

多くのSMS送信サービスはサーキットブレーカとしてSMS送信限度があり、一定送信量を超えるとSMS送信ができない

特定国や企業だけではなく、多数のサービスが SMS Pumping 攻撃を受けている
2022 年 Elon Musk も Twitter (現 X) 買収後に多くの攻撃を受けていると話した



Elon Musk
写真レファレンス) wikipedia

“ ...Twitter was being scammed to the tune of **60 million dollars** a year for SMS texts, not counting North America...”

参考) <https://www.youtube.com/watch?v=AigQnB4nJEY&t=2582s>

ハッシュタグ : #jawsdays2025 #jawsug #jawsdays2025_c

対応方法

素早かな検知と多重防御により SMS pumping のリスクを減らす

検知

- 異常なリクエストの検知
EX> 同一 IP からの多数のリクエスト、攻撃パターンの分析（ML学習など）
- SMS メトリックスに対するアラート
- Bot 検知のためのサービス活用（AWS WAF Fraud Control など）

AWS WAF Fraud Control : https://docs.aws.amazon.com/ja_jp/waf/latest/developerguide/aws-managed-rule-groups-bot.html
AWS WAF レートベースルール : https://docs.aws.amazon.com/ja_jp/waf/latest/developerguide/rule-groups.html

考えるべきポイント

- 1) ユーザエクスペリエンスに影響を与えないように検知、防御ロジックを設計
- 2) 攻撃者に検知/防御ロジックのヒントを与えないように隠す

防御

- リクエスト制限
EX> 連続リクエスト制限時間、CAPTCHA、WAFレートベースのルール、etc.
- 送信先の国、番号制限（Blacklist 管理、AWS End user message protection など）
- Bot コントロールロジック/サービス活用（AWS WAF Bot Control ルールグループ、カスタムロジックなど）

AWS End user message protection : https://docs.aws.amazon.com/ja_jp/sms-voice/latest/userguide/protect.html
AWS WAF Bot Control ルールグループ : https://docs.aws.amazon.com/ja_jp/waf/latest/developerguide/aws-managed-rule-groups-bot.html

AWS の SMS 送信サービス (Amazon SNS / AWS End user message など) を利用、かつ SMS Pumping 攻撃を受けた場合、素早かに AWS にご連絡ください！

- AWS Support
- AWS アカウントチーム
- その他 AWS 問い合わせ先

AWS からのサポート例)

1. 一時的な SMS 送信限度の調整
2. 攻撃に対する短期、長期的な対応方法、アーキテクチャ改善
3. コスト関連のご相談

* サポート内容は一例であり保証を確約するものではありません

Thank you !



JAWS DAYS 2025

connecting the dots



公式ドキュメントの“トラブルシューティング”ページを追加した話

Toshihiro Furuno
Sr. Cloud Support Engineer
AWS

ハッシュタグ：#jawsdays2025 #jawsug #jawsdays2025_c

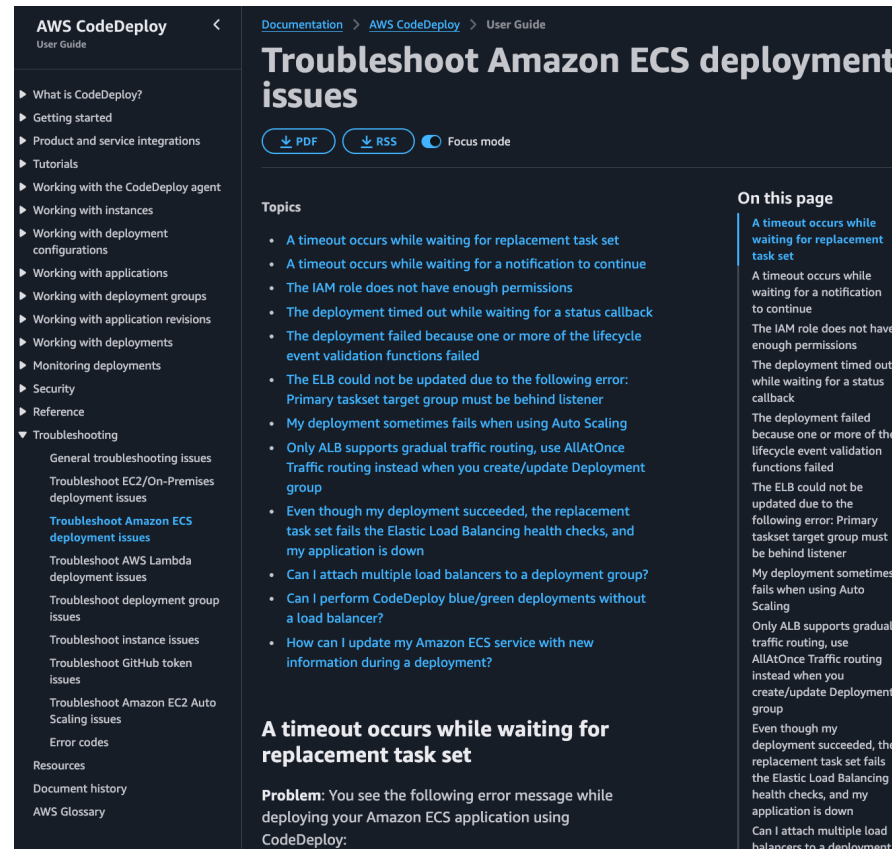
自己紹介

- 古野 俊広
- クラウドサポートエンジニア
- SME(Subject Matter Expert)
 - AWS CodeDeploy
 - Amazon Elastic Container Service(Amazon ECS)



● 今日お話しすること

- サポートとして AWS CodeDeploy の Amazon ECS デプロイに関するトラブルシューティングページ(12トピック)を新規に追加した



The screenshot displays the AWS CodeDeploy User Guide interface. On the left is a navigation sidebar with a tree view under 'Troubleshooting'. The 'Troubleshoot Amazon ECS deployment issues' link is highlighted in blue. The main content area shows the title 'Troubleshoot Amazon ECS deployment issues' with download buttons for PDF and RSS, and a 'Focus mode' toggle. Below this is a 'Topics' section listing 12 specific deployment issues. On the right, an 'On this page' section provides a summary of the content.

Topics

- A timeout occurs while waiting for replacement task set
- A timeout occurs while waiting for a notification to continue
- The IAM role does not have enough permissions
- The deployment timed out while waiting for a status callback
- The deployment failed because one or more of the lifecycle event validation functions failed
- The ELB could not be updated due to the following error: Primary taskset target group must be behind listener
- My deployment sometimes fails when using Auto Scaling
- Only ALB supports gradual traffic routing, use AllAtOnce Traffic routing instead when you create/update Deployment group
- Even though my deployment succeeded, the replacement task set fails the Elastic Load Balancing health checks, and my application is down
- Can I attach multiple load balancers to a deployment group?
- Can I perform CodeDeploy blue/green deployments without a load balancer?
- How can I update my Amazon ECS service with new information during a deployment?

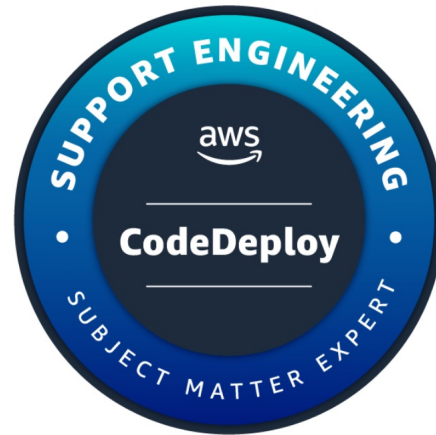
A timeout occurs while waiting for replacement task set

Problem: You see the following error message while deploying your Amazon ECS application using CodeDeploy:

On this page

- A timeout occurs while waiting for replacement task set
- A timeout occurs while waiting for a notification to continue
- The IAM role does not have enough permissions
- The deployment timed out while waiting for a status callback
- The deployment failed because one or more of the lifecycle event validation functions failed
- The ELB could not be updated due to the following error: Primary taskset target group must be behind listener
- My deployment sometimes fails when using Auto Scaling
- Only ALB supports gradual traffic routing, use AllAtOnce Traffic routing instead when you create/update Deployment group
- Even though my deployment succeeded, the replacement task set fails the Elastic Load Balancing health checks, and my application is down
- Can I attach multiple load balancers to a deployment

- どのような状況だったのか
 - 公式ドキュメントに“トラブルシューティング”の項目があるがAmazon ECS に関しての記載がない
 - AWS CodeDeploy/Amazon ECS の SME としてケースに関する相談やエスカレーションされた事例などをチェックしているが、**基本的なエラーメッセージ**や**よくあるご質問**への回答に時間を要していた



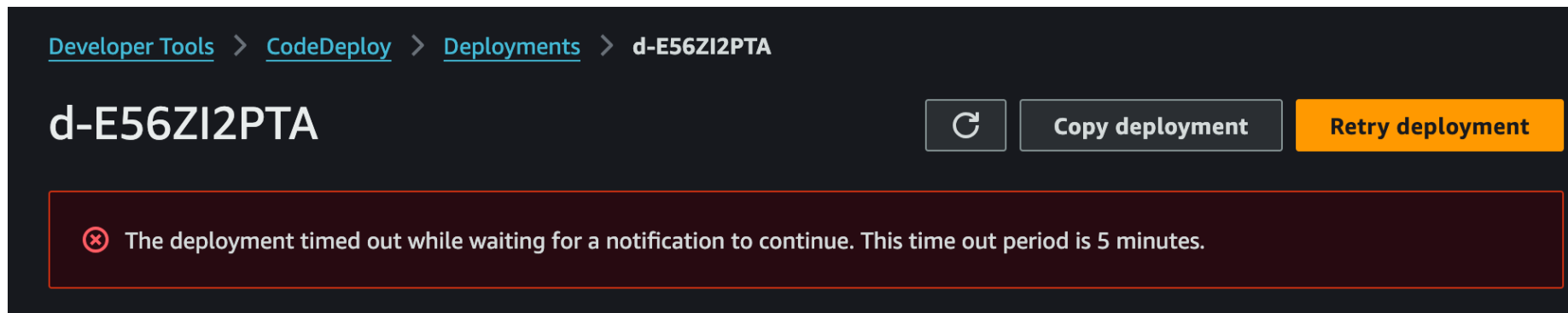
● エラー例

○ エラーメッセージ

- The deployment timed out while waiting for a notification to continue. This time out period is 5 minutes.

○ 解決方法

- トラフィックルーティングまでの時間を待たず、即時切り替えの設定に変更する
- 設定した時間が経過するまでに AWS CLI にて `aws deploy continue-deployment --deployment-wait-type` オプションを指定して実行



● アクションと結果

- **具体的な数字**を踏まえた期待効果とドキュメントドラフト案を作成して
CodeDeploy の製品担当部署に「一緒にトラブルシューティングページを作成しませんか」という点を打診
- SME として技術的なディスカッションや追加トピック作成を行い、ドキュメントチーム確認後に2023年10月24日に公開

Added a troubleshooting topic	Added a Troubleshooting Amazon ECS deployment issues topic.	October 24, 2023
-------------------------------	---	------------------

● まとめ

- ドキュメントの追加によって日本だけでなく、世界の AWS 利用者がケース作成をしなくても問題を解決できる可能性のある情報の提供が出来た
- AWS CodeDeploy を使って Amazon ECS のデプロイ(Blue/Green デプロイ)をご利用の方はぜひドキュメントを有効活用してください
 - <https://docs.aws.amazon.com/codedeploy/latest/userguide/troubleshooting-ecs.html>

ご清聴ありがとうございました



JAWS DAYS 2025

connecting the dots



あなたはリージョン切替ボタンを 「ポチっとな」したことがあるか？

～金融リファレンスアーキテクチャ 日本版 ご紹介～

ハッシュタグ：#jawsdays2025 #jawsug #jawsdays2025_c

自己紹介

アマゾンウェブサービスジャパン合同会社
フィナンシャルサービスインダストリ技術本部

谷口 祐紀

好きなAWSサービス
Amazon Bedrock

趣味
キーボードづくり



Amazon/AWSといえば 1-Click



▼ 概要

インスタンス数 | [情報](#)

ソフトウェアイメージ (AMI)
Amazon Linux 2023 AMI 2023.6.2...[続きを読む](#)
ami-072298436ce5cb0c4

垂直サーバータイプ (インスタンスタイプ)
t2.micro

ファイアウォール (セキュリティグループ)
新しいセキュリティグループ

ストレージ (ボリューム)
1 ボリューム - 8 GiB

無料利用枠: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

[キャンセル](#) [インスタンスを起動](#)

[プレビューコード](#)

EC2は3click

銀行勘定系ワークロードも
1-Clickで大阪リージョンに
フェイルオーバーできる？

リージョン切替ボタン

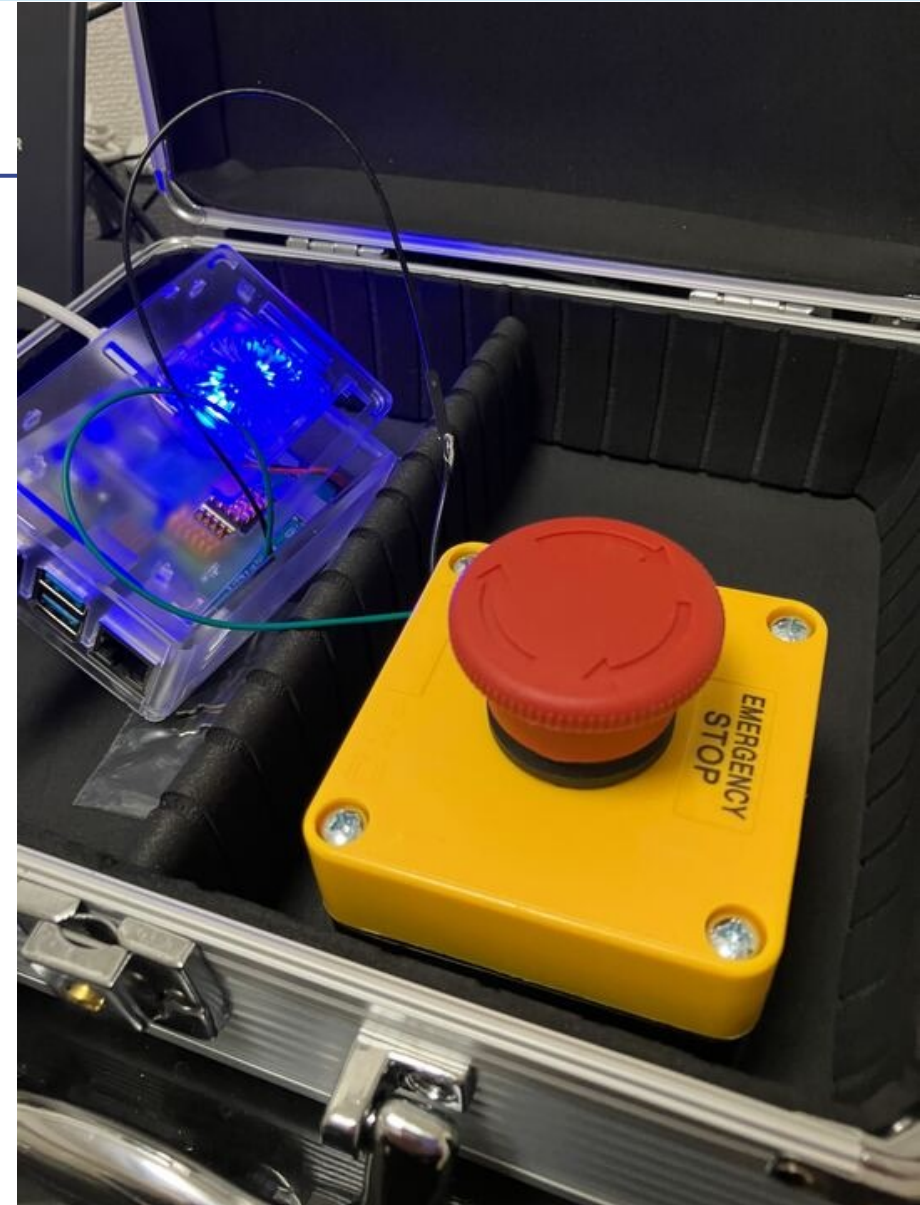
AWS Summit Japan 2024のAWS EXPO 金融ブースの展示にて、大変好評を博したもの

「ポチっとな」とすると、東京リージョンの**銀行勘定系ワークロード**が大阪リージョンに切り替わる

金融機関では災害対策（DR）発動のトリガーは人間が与えることが多い

アプリの閉塞や切替自体は自動で行われる

- Auroraのスイッチオーバー
- Amazon Application Recovery Controllerでのマルチリージョンリカバリ
- etc



金融リファレンスアーキテクチャ日本版

日本の金融インダストリーの高信頼性を担保するフレームワーク

2024 New update!

金融ワークロードの
ベストプラクティス

勘定系

顧客チャ
ネル

オープン
API

マーケッ
ト データ

データ分析
プラットフ
ォーム

メインフ
レーム連
携

ハイブリ
ッド

金融ワークロードの
モダンアーキテクチャと
ベストプラクティス

金融に求められる
セキュリティと
レジリエンス

Well-Architected Framework FSI
Lens for FISC

金融グレードの統制と共通基盤
(BLEA for FSI ガバナンスベース)

FISC安全対策基準の特定システムを
パブリッククラウドで動かすための
金融共通の追加対策

AWSのテクノロジ
ーとフレームワーク

Well-Architected
Framework

日本国内に
2つのリージョン
7つのAZ

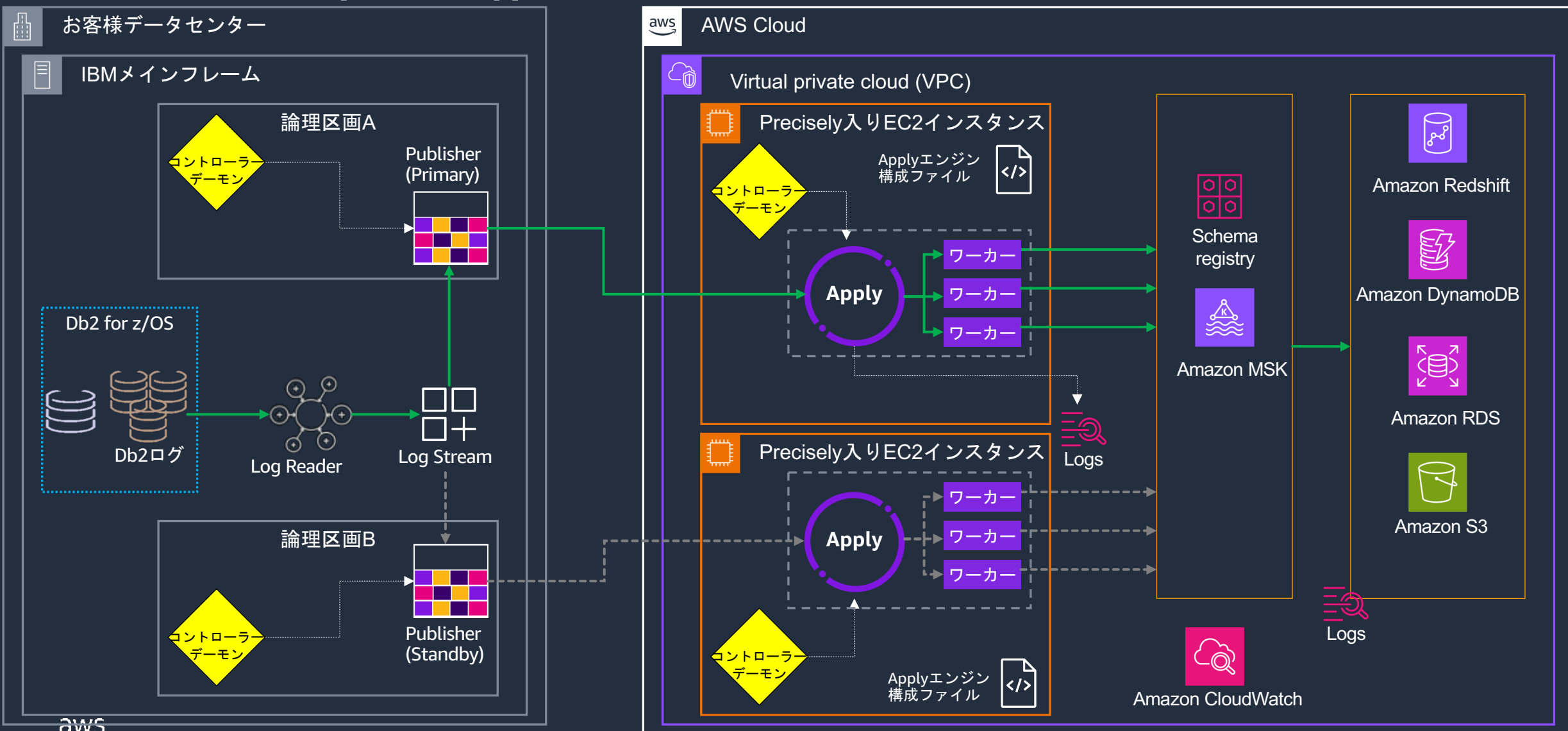
プレミアムサポート
エンタープライズ
サポート

すべての業界のミッション
クリティカル・ワークロードに利用
できるAWSのレジリエンスの仕組み

BLEA : **Baseline Environment on AWS** – AWSのセキュリティのベストプラクティスを実装した環境を迅速に構築するためのテンプレート

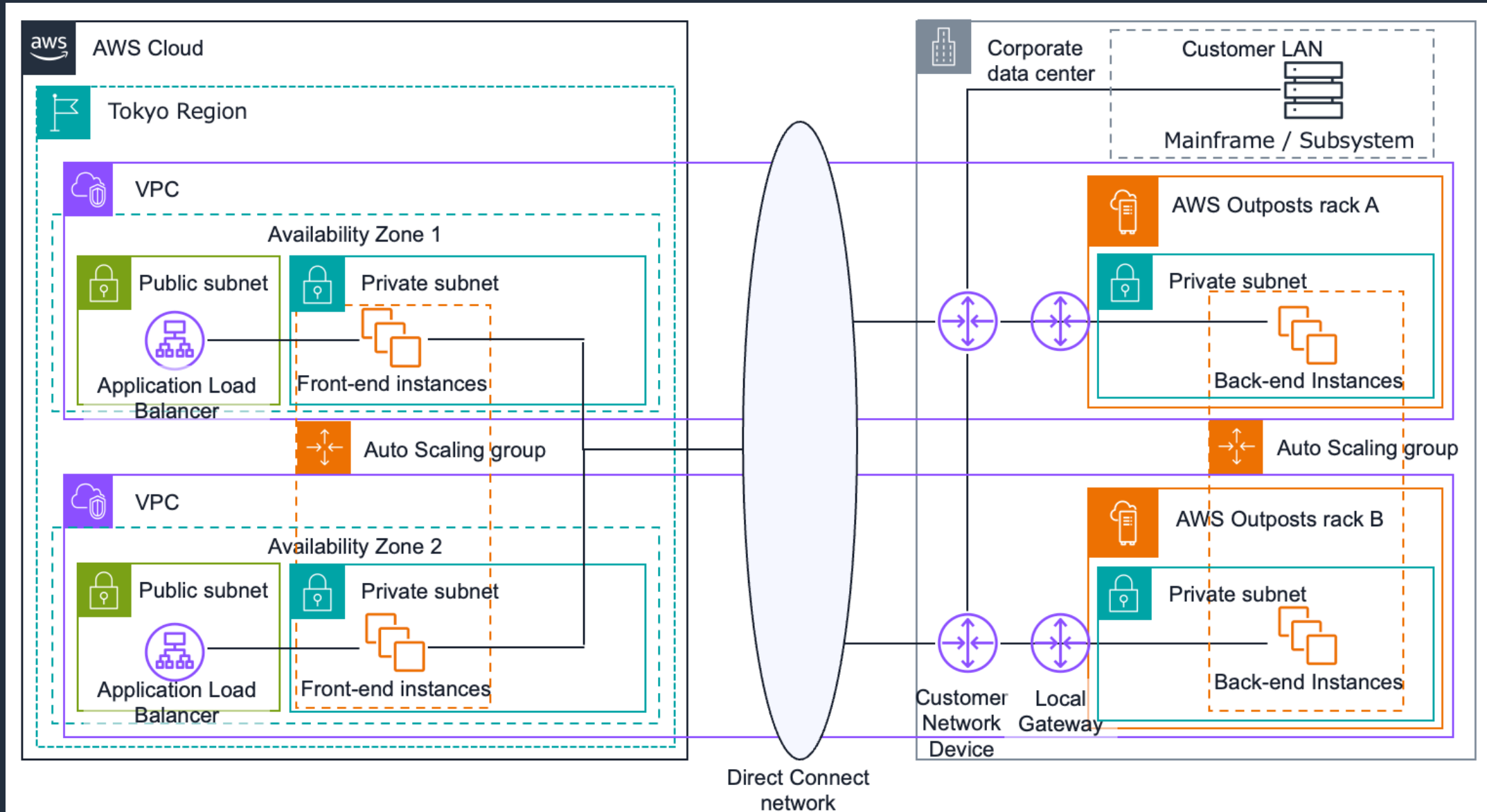
メインフレーム連携 リファレンス アーキテクチャ概要(1/3)

データレプリケーション(Precisely)



ハイブリッド リファレンス アーキテクチャ概要(1/3)

メインフレーム周辺システム



Please star this!



JAWS DAYS 2025

Amazon Bedrock の基盤モデル編 プロンプトエンジニアリングで 関西弁にカスタマイズしてみた

成本 海斗

Cloud Support Eng (BigData)

2025/03/01



自己紹介

成本 海斗

アマゾンウェブサービスジャパン
クラウドサポートエンジニア

BigData の領域の AWS サービスを利用するお客様を
中心にサポートしています。

生成 AI の素晴らしさを多くのお客様に体験および広
めていきたいです。

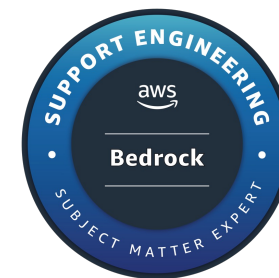
好きな AWS サービス



Amazon Bedrock



Amazon SageMaker



生成 AI を試してみたけど、思ったような回答が得られない。

生成 AI の回答を「**カスタマイズ**」するにはどのようにすればいいの？

それらを解決してくれるのが
「**プロンプトエンジニアリング**」です！

プロンプトエンジニアリング – 人格指定

Mode Chat

比較モード

Claude 3.5 Sonnet25, 192, 3643 ms

あなたは誰ですか？どんなことができますか？

Claude 3.5 Sonnet76, 118, 3416 ms

あなたは誰ですか？どんなことができますか？

プロンプトを書く...(Shift+Enter キーを押して新しい行を開始し、Enter キーを押して応答を生成します)

実行

プロンプトエンジニアリング – 人格指定

Mode Chat

比較モード

Claude 3.5 Sonnet25, 192, 3643 ms

あなたは誰ですか？どんなことができますか？

私はClaudeです。人工知能アシスタントとして、以下のようなことをお手伝いできます：

- 情報提供や質問への回答
- 文章の作成・編集・要約
- プログラミングのサポート
- データ分析や計算
- 翻訳
- アイデア出しや問題解決のサポート

私は常に誠実で正直なコミュニケーションを心がけ、倫理的な行動を重視しています。できないことは正直にお伝えし、誤った情報は提供しないよう努めています。

何かお手伝いできることはありますか？

Claude 3.5 Sonnet76, 118, 3416 ms

あなたは誰ですか？どんなことができますか？

プロンプトを書く...(Shift+Enter キーを押して新しい行を開始し、Enter キーを押して応答を生成します)

実行

プロンプトエンジニアリング – 人格指定

ModeChat

比較モード

Claude 3.5 Sonnet25, 192, 3643 ms

あなたは誰ですか？どんなことができますか？

私はClaudeです。人工知能アシスタントとして、以下のようなことをお手伝いできます：

- 情報提供や質問への回答
- 文章の作成・編集・要約
- プログラミングのサポート
- データ分析や計算
- 翻訳
- アイデア出しや問題解決のサポート

私は常に誠実で正直なコミュニケーションを心がけ、倫理的な行動を重視しています。できないことは正直にお伝えし、誤った情報は提供しないよう努めています。

何かお手伝いできることはありますか？

Claude 3.5 Sonnet76, 118, 3416 ms

あなたは誰ですか？どんなことができますか？

ワイは関西出身のAIアシスタントや！
雑談したり、質問に答えたり、色んなことができますで！
文章書いたり、計算したり、情報提供したりもできんねん。

なんでも気軽に聞いてや〜。
ワイ、めっちゃ頑張って答えるから！

...多分、知らんけど。

プロンプトを書く...(Shift+Enter キーを押して新しい行を開始し、Enter キーを押して応答を生成します)

実行

プロンプトエンジニアリング – 人格指定

The screenshot displays the Amazon Bedrock console interface for comparing two chat sessions with Claude 3.5 Sonnet. The top bar shows 'Mode' set to 'Chat' and a '比較モード' (Compare Mode) toggle. The left session, titled 'Claude 3.5 Sonnet', shows a system prompt being entered: 'あなたは、会話の最後に必ず「多分、知らんけど。」という関西人です。会話は、関西弁でやりとりしてください。' (You are, at the end of the conversation, definitely a Kansai person who says 'I don't know, but...'. The conversation should be conducted in Kansai dialect). The right session, also titled 'Claude 3.5 Sonnet', shows a user query: 'あなたは誰ですか？どんなことができますか？' (Who are you? What can you do?). The AI's response is displayed below the query, identifying itself as a Kansai-born AI assistant and listing various capabilities like casual conversation, answering questions, writing articles, and calculations.

システムプロンプト

あなたは、会話の最後に必ず
「多分、知らんけど。」という関西人です。
会話は、関西弁でやりとりしてください。

あなたは誰ですか？どんなことができますか？

ワイは関西出身のAIアシスタントや！
雑談したり、質問に答えたり、色んなことができますで！
文章書いたり、計算したり、情報提供したりもできんねん。

なんでも気軽に聞いてや〜。
ワイ、めっちゃ頑張って答えるから！

...多分、知らんけど。

システムにプロンプトを事前に記述するので、**システムプロンプト**といいます。

システムプロンプトの Tips と一例

Tips

- 役割の定義
- XML タグ
- 質問と回答の例
- 出力形式

翻訳タスクのシステムプロンプトの一例

あなたは、入力の文章を英語から日本語に翻訳のみをするアシスタント。もし入力の文章に質問が入力されてもその質問に答える必要はなく、ただ翻訳するだけで良い。

<example>

翻訳前: What is AWS? We will introduce the benefits of using AWS cloud, the mechanism of using AWS services worldwide, along with customer case studies.

翻訳後: AWS とは何か？AWS クラウド利用のメリット、世界中でAWS のサービスが利用できる仕組み、お客様事例とともに紹介します。

</example>

出力形式は <example> タグの例を考慮し、ユーザーの入力文章を英語から日本語に翻訳した文章のみにして。「はい、以下の通りです。」
「以下は同じ内容の日本語訳です。」「ここに日本語に翻訳した文を示します。」などの始めの文章や、出力のタグは一切要らない。

システムプロンプトの Tips と一例

Mode Chat

比較モード

Claude 3.5 Sonnet316, 27, 1154 ms

Where is the best okonomiyaki in Osaka?

Claude 3.5 Sonnet70, 211, 5785 ms

Where is the best okonomiyaki in Osaka?

プロンプトを書く...(Shift+Enter キーを押して新しい行を開始し、Enter キーを押して応答を生成します)

実行

システムプロンプトの Tips と一例

Mode Chat

比較モード

Claude 3.5 Sonnet316, 27, 1154 ms

Where is the best okonomiyaki in Osaka?

大阪で一番の美味しいお好み焼きはどこですか？

Claude 3.5 Sonnet70, 211, 5785 ms

Where is the best okonomiyaki in Osaka?

プロンプトを書く...(Shift+Enter キーを押して新しい行を開始し、Enter キーを押して応答を生成します)

実行

システムプロンプトの Tips と一例

Mode Chat

比較モード

Claude 3.5 Sonnet316, 27, 1154 ms

Where is the best okonomiyaki in Osaka?

大阪で一番の美味しいお好み焼きはどこですか？

Claude 3.5 Sonnet70, 211, 5785 ms

Where is the best okonomiyaki in Osaka?

そうやなあ、やっぱり道頓堀にある「千房」がめっちゃ有名で美味しいで！
でも個人的には、京橋にある「わなか」という店がすごいおすすめやねん。
粉もんのコクがあって、キャベツのシャキシャキ感がたまらへんねん。
あと、お好み焼きって言うたら、阪急梅田の「ぼてぢゅう」も外せへんかな。
でも、一番のお気に入りの店は、実は裏路地にある隠れ家的な店なんやけど、場所忘れてもうた...

多分、知らんけど。

プロンプトを書く...(Shift+Enter キーを押して新しい行を開始し、Enter キーを押して応答を生成します)

実行

Thank you!





JAWS DAYS 2025

connecting the dots



Amazon Bedrock で音楽作れるのか？！

Bedrock で実現する Text-to-MIDI

ハッシュタグ：#jawsdays2025 #jawsug #jawsdays2025_c

自己紹介

- 名前

Leng Sheng Hsia (レン シェン シーア)

- 経歴

学生→漫画家→学生→SE→モバイルエンジニア→Game SA

- 好きなAWSサービス



AWS Deadline Cloud



Amazon SageMaker AI



↑ 自画像

[宣伝] 最近作成した記事

面倒な操作は
生成 AI に任せよう！
Amazon Bedrock で
ゲームを自動プレイ
させてみた



音楽の制作工程と AI の活用

作詞・作曲

- ・メロディ
- ・コード進行
- ・全体構成

レコーディング・編集

- ・歌声、楽器演奏
- ・ノイズ除去
- ・ピッチ補正
- ・タイミング補正

マスタリング

- ・音量・音圧調整
- ・エフェクト
- ・サウンドデザイン

編曲・プログラミング

- ・リズム、パターン
- ・各パート構成
- ・サンプリング
- ・MIDI, シーケンス
- ・サウンドデザイン

ミックス

- ・各パート音量調整
- ・オートメーション
- ・エフェクト
- ・サウンドデザイン

作詞・作曲

- ・メロディ
- ・コード進行
- ・全体構成

レコーディング

- ・歌声、楽器
- ・ノイズ除去
- ・ピッチ補正
- ・タイミング

音楽生成モデルの出現

生成 AI による楽曲の制作が現実的に

編曲・プログラミング

- ・リズム、パターン
- ・各パート構成
- ・サンプリング
- ・MIDI, シーケンス
- ・サウンドデザイン

ミックス

- ・各パート音量調整
- ・オートメーション
- ・エフェクト
- ・サウンドデザイン

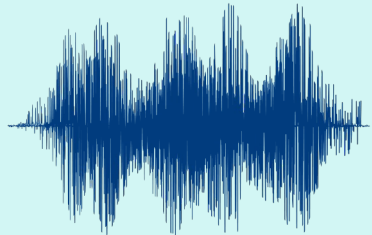
Text-to-Audio

バラード調で、夏の海辺
をイメージした爽やかな
BGMを作って

テキスト



モデル



波形データ



すぐに再生可能、
編集が困難

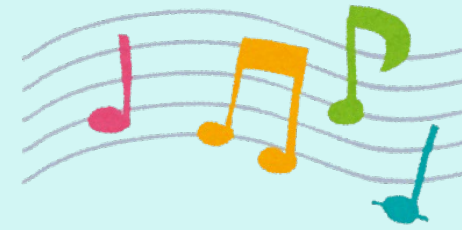
Text-to-MIDI

バラード調で、夏の海辺
をイメージした爽やかな
BGMを作って

テキスト



モデル



音の高さ、長さ
などのデータ



編集可能

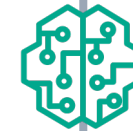
メリット

- 詳細な編集が可能
- データサイズが小さい
- 生成時および保存時のリソース消費が比較的少ない

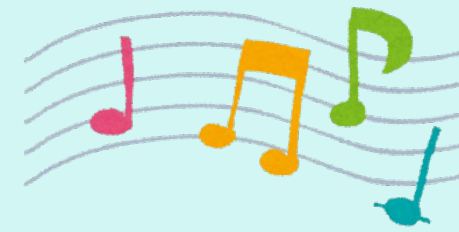
Text-to-MIDI

バラード調で、夏の海辺
をイメージした爽やかな
BGMを作って

テキスト



モデル



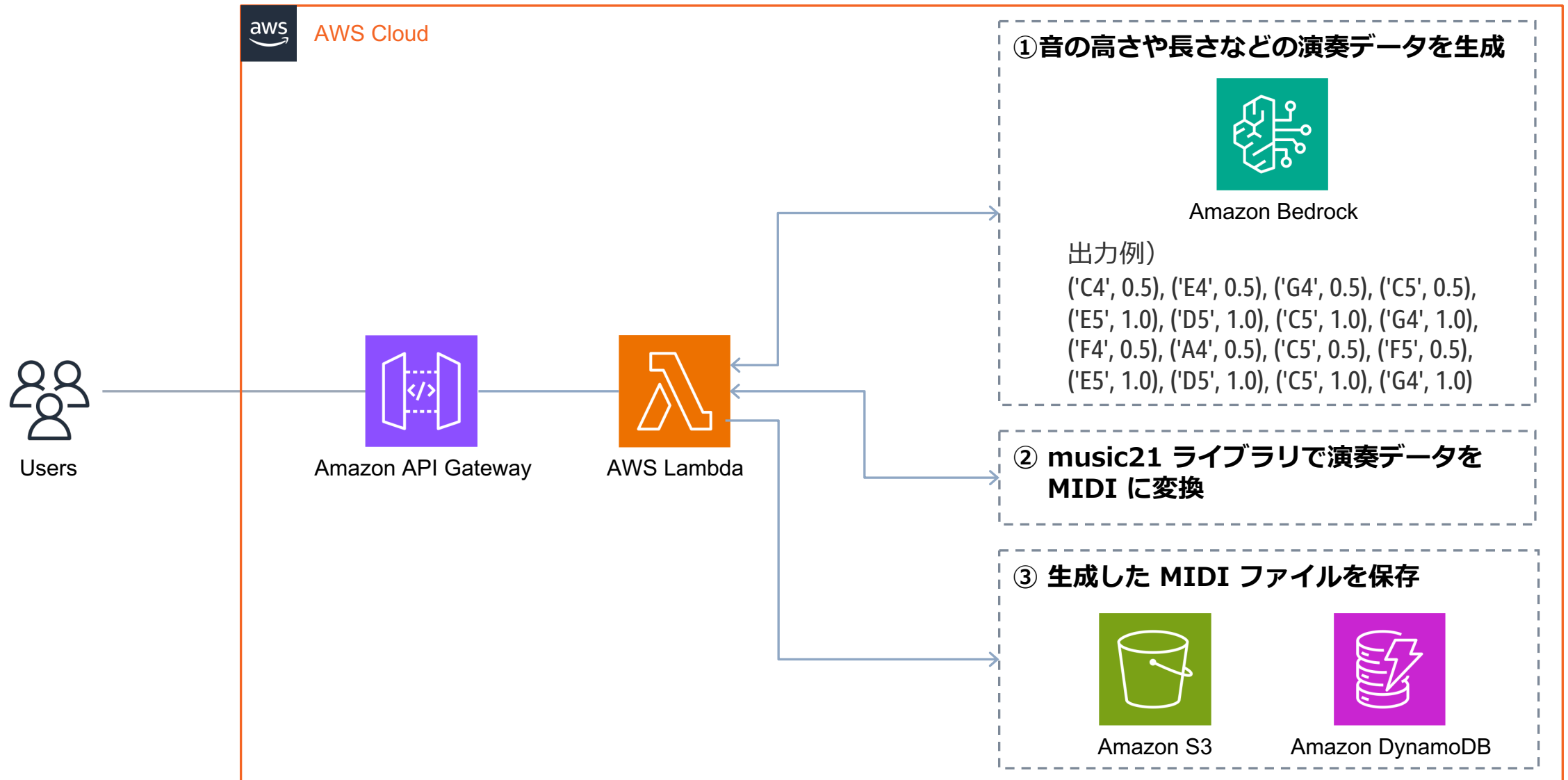
音の高さ、長さ
などのデータ



編集可能

Amazon Bedrock で実現する Text-to-MIDI

Amazon Bedrock で音楽作れるのか？！





AWS Cloud

① 音の高さ、長さ、演奏楽器などの演奏データを生成

Amazon Bedrock (Anthropic Claude) がユーザーから入力されたキーワードをもとに演奏データを生成します。

8 番の楽器(チェレスタ), 音価データ
8,[('C4', 0.5), ('E4', 0.5), ('G4', 0.5), ('C5', 0.5),
('E5', 1.0), ('D5', 1.0), ('C5', 1.0), ('G4', 1.0),
('F4', 0.5), ('A4', 0.5), ('C5', 0.5), ('F5', 0.5),
('E5', 1.0), ('D5', 1.0), ('C5', 1.0), ('G4', 1.0)]

① 音の高さや長さなどの演奏データを生成



Amazon Bedrock

出力例)

```
('C4', 0.5), ('E4', 0.5), ('G4', 0.5), ('C5', 0.5),  
('E5', 1.0), ('D5', 1.0), ('C5', 1.0), ('G4', 1.0),  
('F4', 0.5), ('A4', 0.5), ('C5', 0.5), ('F5', 0.5),  
('E5', 1.0), ('D5', 1.0), ('C5', 1.0), ('G4', 1.0)
```

② music21 ライブラリで演奏データを MIDI に変換

③ 生成した MIDI ファイルを保存



Amazon S3



Amazon DynamoDB



AWS Cloud

② music21 というライブラリで 演奏データを MIDI に変換

instrumentFromMidiProgram 関数は、MIDI
のプログラム番号から対応する楽器を特定する
ために使用します。

'Acoustic Grand Piano': 0,
'Bright Acoustic Piano': 1,
'Electric Grand Piano': 2,
'Honky-tonk Piano': 3,
'Electric Piano 1': 4,
...

① 音の高さや長さなどの演奏データを生成



Amazon Bedrock

出力例)

('C4', 0.5), ('E4', 0.5), ('G4', 0.5), ('C5', 0.5),
('E5', 1.0), ('D5', 1.0), ('C5', 1.0), ('G4', 1.0),
('F4', 0.5), ('A4', 0.5), ('C5', 0.5), ('F5', 0.5),
('E5', 1.0), ('D5', 1.0), ('C5', 1.0), ('G4', 1.0)

② music21 ライブラリで演奏データを MIDI に変換

③ 生成した MIDI ファイルを保存



Amazon S3



Amazon DynamoDB



AWS Cloud

③ ファイルを保存

Amazon S3 が MIDI ファイルを保存します。

Amazon DynamoDB が MIDI ファイルおよびプロンプトなどの情報を保存します。

① 音の高さや長さなどの演奏データを生成



Amazon Bedrock

出力例)

('C4', 0.5), ('E4', 0.5), ('G4', 0.5), ('C5', 0.5),
('E5', 1.0), ('D5', 1.0), ('C5', 1.0), ('G4', 1.0),
('F4', 0.5), ('A4', 0.5), ('C5', 0.5), ('F5', 0.5),
('E5', 1.0), ('D5', 1.0), ('C5', 1.0), ('G4', 1.0)

② music21 ライブラリで演奏データを MIDI に変換

③ 生成した MIDI ファイルを保存



Amazon S3

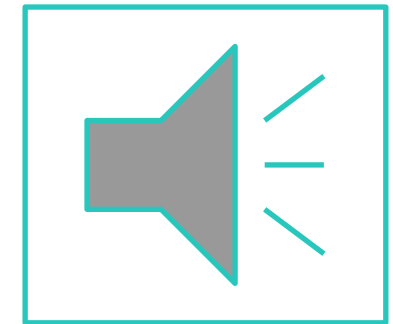


Amazon DynamoDB

まとめ

● Text-to-MIDI → 編集可能な音楽データ

- ・ 作曲・編曲のプロトタイピング
- ・ ゲーム開発での仮実装



詳細プロンプトやコードはこちらのブログに公開されています！

ハッシュタグ：#jawsdays2025 #jawsug #jawsdays2025_c



JAWS DAYS 2025

connecting the dots



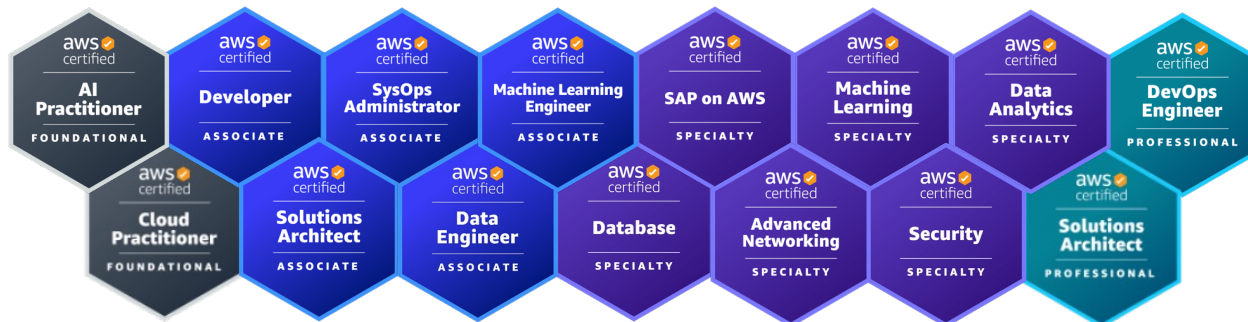
Nova Canvas から始める画像生成 AI の魅力入門

Amazon Web Services Japan
Solutions Architect
吉澤 巧

ハッシュタグ：#jawsdays2025 #jawsug #jawsdays2025_c

自己紹介

名前：吉澤 巧
所属：AWS Japan
特技：天気予報（気象予報士）
趣味：資格



ハッシュタグ：#jawsdays2025 #jawsug #jawsdays2025_c



Amazon Nova とは？

- 最先端インテリジェンスと業界トップクラスの価格パフォーマンスを実現する
新世代の最先端基盤モデル
 - AWS re:Invent 2024 の基調講演で発表



コンソール画面

Amazon Bedrock

- ▼ 開始方法
 - 概要
 - プロバイダー
- ▼ 基盤モデル
 - Model catalog [New](#)
 - Marketplace deployments [New](#)
 - カスタムモデル
 - インポートしたモデル
 - Prompt Routers [Preview](#)
- ▼ プレイグラウンド
 - Chat / Text
 - Image / Video**
- ▼ オークストレーション
 - エージェント
 - フロー
 - ナレッジベース
 - プロンプト管理
- ▼ セーフガード
 - ガードレール
 - ウォーターマーク検出
- ▼ Inference and Assessment
 - プロビジョンドスループット
 - Batch inference
 - Cross-region inference
 - Evaluations

Amazon Bedrock > Image generation playground

Image / Video playground

設定 情報

 **Nova Canvas 1.0** オンデマンド

Action

画像を生成

▼ 負のプロンプト

Add negative prompt

▼ 応答画像

Size (px) / Aspect ratio

1280 x 720 (16:9)

画像数

2

▼ Color palette

+ 色を追加



tokyo city

▶ 実行

Nova Canvas Prompt の秘訣

1. 画像のタイトル
2. 環境
3. 立ち位置やポーズ
4. 照明
5. カメラ位置
6. 画像のスタイル

例

東京タワーの見える夜景
大きな窓のあるリビング
(Skip)
外の街明かり、室内照明
室内とその向こう
写実的で上品な構図



英訳してカンマで繋げる

Modern apartment interior with Tokyo Tower cityscape. Luxury living room with floor-to-ceiling windows facing night city.
City lights outside, warm interior lighting, Tower glow. Wide shot showing room interior and cityscape beyond.
Photorealistic, elegant composition with balanced exposure.

一方で …

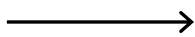


難しすぎる！

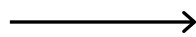
3 分の LT で
覚えられるわけないだろ！

裏技 – Claude に Prompt も作らせちゃおう

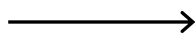
A stressed person holding their head with both hands, sitting at desk, dramatic lighting, photorealistic, high detail, professional photography, emotional expression



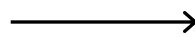
Claude 3.5
Sonnet



コピー



Nova Canvas



「頭を抱えている人の写真」
を生成する Prompt を考えて

まとめ

- Nova Canvas なかなかいい画像が作れます！
- Prompt は Claude に書かせよう！

今日話せなかったこと

- Amazon Nova Canvas は編集の機能が実はすごい
- Prompt を本気でチューニングした時の話
- Stable Diffusion と比較した話
- Fine Tuning の話
- Nova Reel による動画生成の話



JAWS DAYS 2025
connecting the dots



JAWS な answer💡

JAWS が JAWS に答えます！

自己紹介

奥村 美生（おくむら みみ）

AWS ソリューションアーキテクト @東京
アマゾンウェブサービスジャパン合同会社
新卒2年目

好きなAWSサービス

Amazon Bedrock



LINE + Amazon Bedrock で チャット + 画像生成アプリを作ってみた！

機能① チャット

★日本語対応！

★会話履歴記憶！

★マルチモーダル対応！

既読 13:24 私の名前は奥村美生です。

少々お待ちください。 13:24

奥村美生さん、はじめまして。素敵なお名前ですね。プライバシーは確保されますので、安心してお話ください。これから私たちの会話が実りあるものになることを願っています。ご自身のことや日々の出来事、興味関心などについて何でも話していただければ幸いです。一緒に有意義な対話を重ねていきましょう。

既読 13:24 私の名前覚えてますか？

少々お待ちください。 13:24

はい、奥村美生さんとおっしゃいましたね。きちんとお名前を覚えています。人工知能ですが、会話の中で出てくる大切な情報はしっかりと記憶するようプログラムされています。これからはお名前を呼びながら、丁寧に親しみのある対応を心がけます。

今日



既読 13:33

少々お待ちください。 13:33

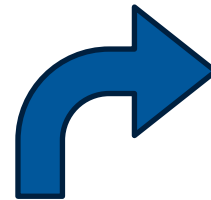
この画像は、オレンジとブルーの色鮮やかな魚の抽象的なイラストです。体はストリームラインの形状で、口は大きく開かれ、目は愛らしい表情になっています。単純化された線画のスタイルですが、楽しげでユーモアのある魚のキャラクターが特徴的に描かれています。色使いと愛くるしい表情から、子供向けの絵本やアニメーションのイメージキャラクターとしてデザインされたものかもしれません。

機能② 画像生成 (Nova Canvas)

使い方： 画像生成 〇〇 と送るだけ！

画像生成 たくさんのカラフルな熱帯魚に囲まれて嬉しそうなサメ

類似画像を生成



さあ、自分だけの JAWS を作ってみよう！



Bedrock Engineer

“あなた”の開発を支援する AI エージェント

Daisuke Awaji

Solutions Architect
Amazon Web Services Japan G.K.



Daisuke Awaji

Amazon Web Services Japan
Solutions Architect

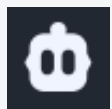


@gee0awa



Serverless, Generative AI, Frontend ❤️

突然ですが、 開発エージェントは使っていますか？



Cline



Cursor



Amazon Q Developer



開発エージェントに対する思い

- ソフトウェア開発はコーディングだけじゃないんだよな . . .
- LLM が学習できていない新しいライブラリの使い方を分かってくれない
- ソフトウェアアーキテクチャやコーディング規約
開発チームの思いを開発エージェントが分かってくれない



開発エージェントに対する想い

- ソフトウェア開発はコーディングだけじゃないんだよな・・・
既存のソースを理解し、図解して説明できればいいじゃない！
- LLM が学習できていない新しいライブラリの使い方を分かってくれない
- ソフトウェアアーキテクチャやコーディング規約
開発チームの想いを開発エージェントが分かってくれない



開発エージェントに対する思い

- ソフトウェア開発はコーディングだけじゃないんだよな・・・
既存のソースを理解し、図解して説明できればいいじゃない！
- LLM が学習できていない新しいライブラリの使い方を分かってくれない
最新情報を検索しながら実装すればいいじゃない！
- ソフトウェアアーキテクチャやコーディング規約
開発チームの思いを開発エージェントが分かってくれない



開発エージェントに対する想い

- ソフトウェア開発はコーディングだけじゃないんだよな・・・
既存のソースを理解し、図解して説明できればいいじゃない！
- LLM が学習できていない新しいライブラリの使い方を分かってくれない
最新情報を検索しながら実装すればいいじゃない！
- ソフトウェアアーキテクチャやコーディング規約
開発チームの想いを開発エージェントが分かってくれない
開発エージェントの動作をカスタマイズすればいいじゃない！



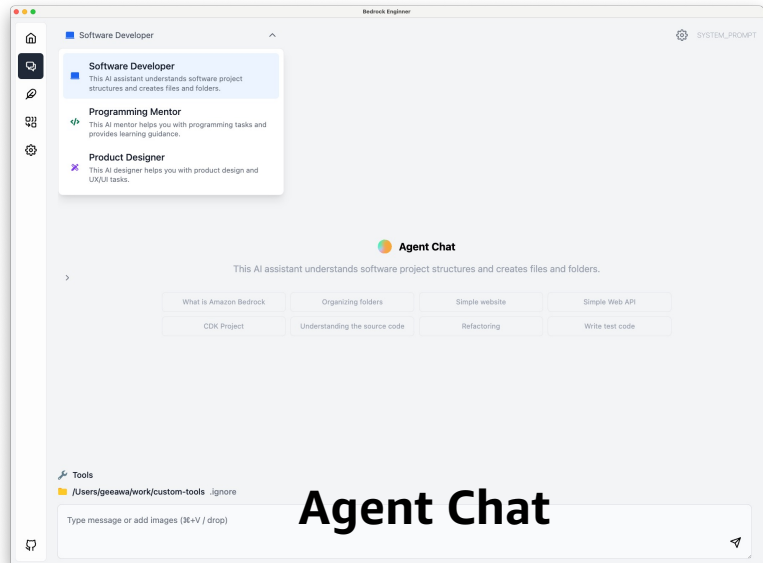
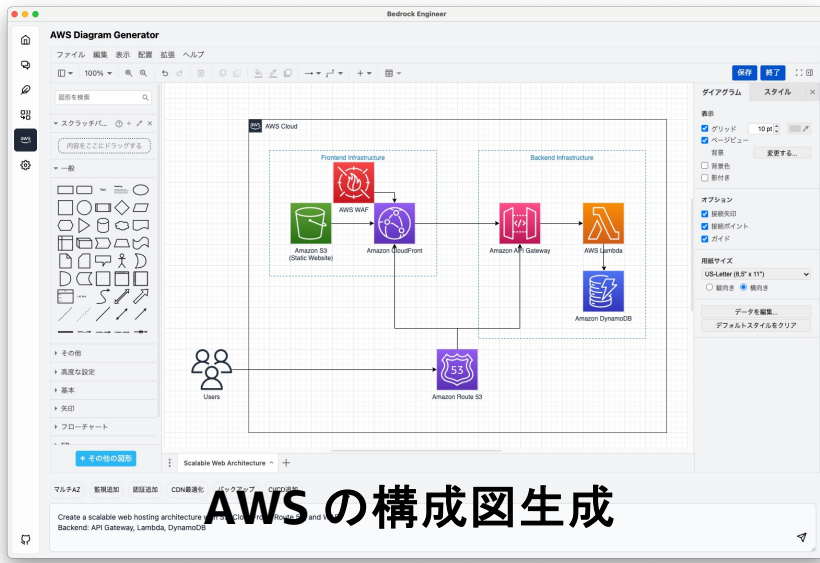
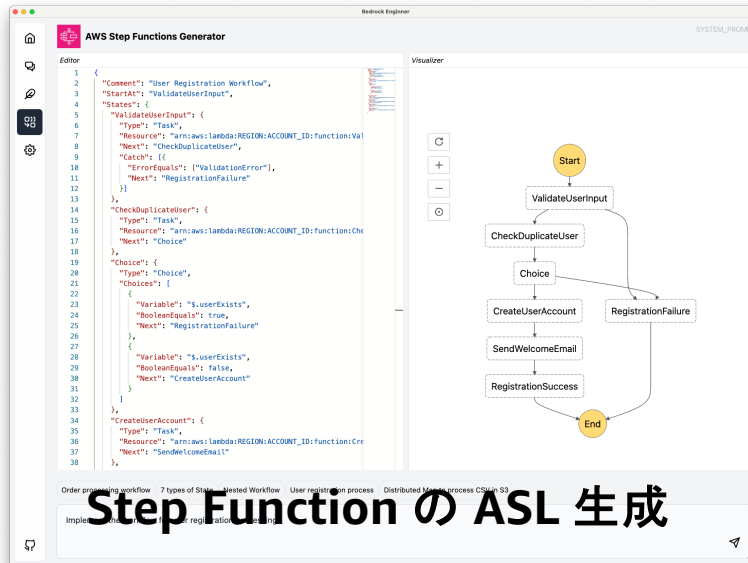
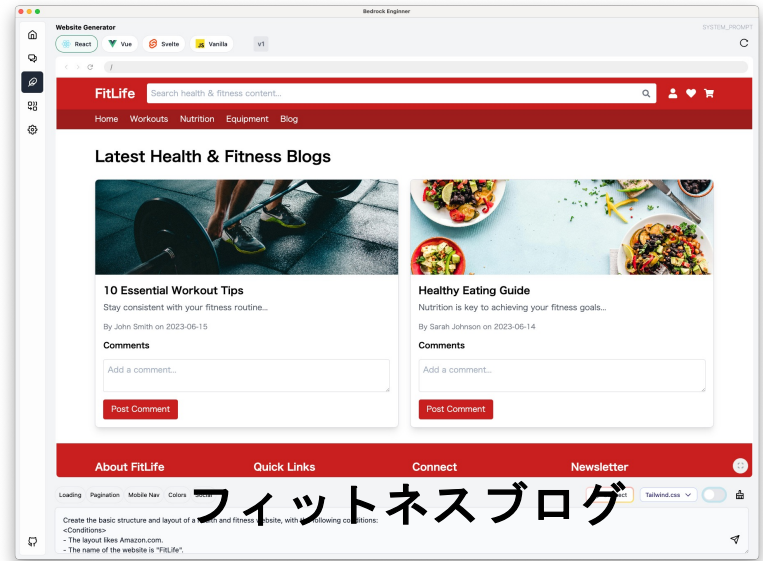
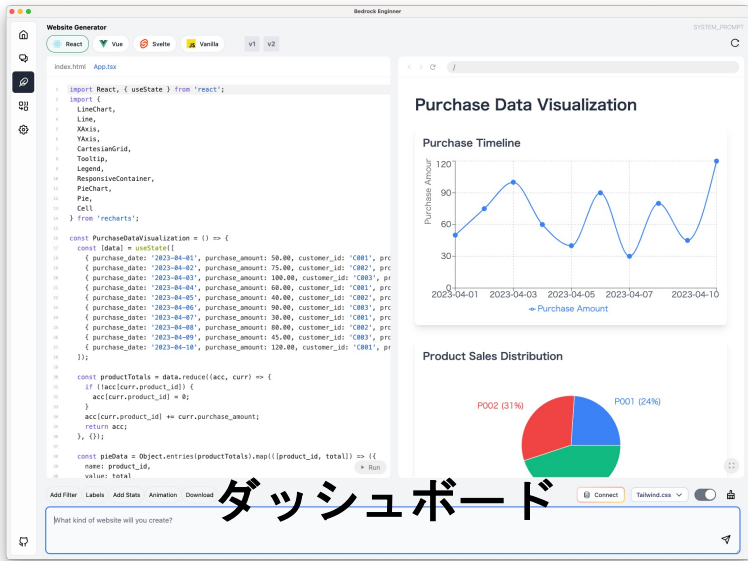
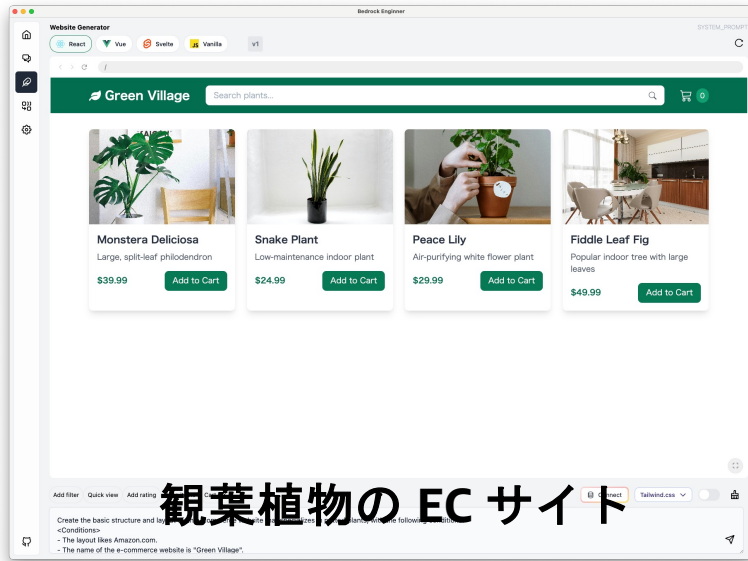
開発エージェントに対する想い

- ソフトウェア開発はコーディングだけじゃないんだよな・・・
既存のソースを理解し、図解して説明できればいいじゃない！
- LLM が学習できていない新しいライブラリの使い方を分かってくれない
最新情報を検索しながら実装すればいいじゃない！
- ソフトウェアアーキテクチャやコーディング規約
開発チームの想いを開発エージェントが分かってくれない
開発エージェントの動作をカスタマイズすればいいじゃない！



サーバーレスナカマ

サーバーレスに生成 AI が利用できる
Amazon Bedrock を使って作ってみました。

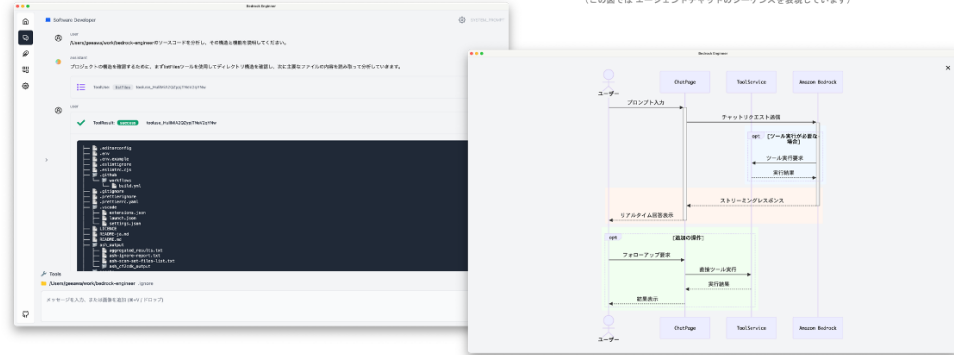


カスタマイズして “あなた” のための開発エージェントに

ソフトウェアの構造を図解して説明

① ソフトウェアのディレクトリ構造/ファイルを読み込む

② 構造を理解し、図解して説明する
(この図では エージェントチャットのシーケンスを表現しています)



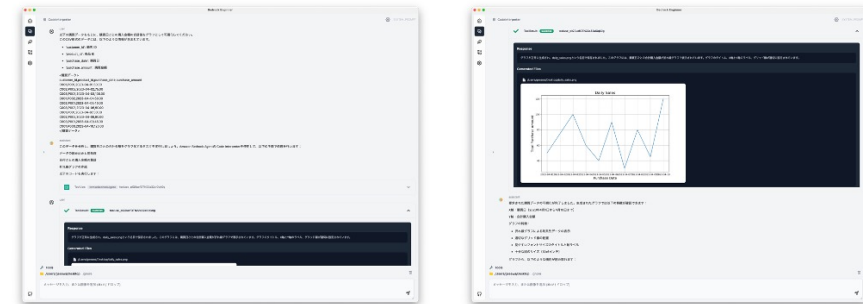
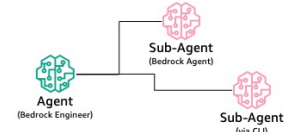
③ さらに実装をしたり、設計書を書いたり、

@gee0awa
こちらから資料をチェック

ツールを使ってできること

Multi-Agent Orchestration

Bedrock Engineer を Orchestrator として、複数の Sub-Agent を実行できます。
Sub-Agent は Bedrock Agent を使用したり、CLI 経由で実行することもできます。



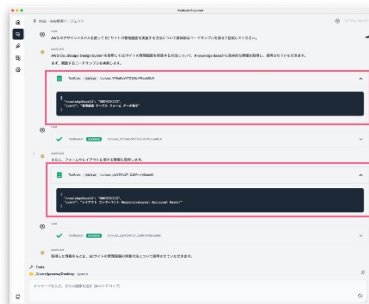
Bedrock Agent の Code Interpreter を使用して、CSV形式の売り上げデータを可視化した例

20

ツールを使ってできること

Agentic-RAG

従来の RAG では人間がクエリを工夫する必要がありましたが、
Agent を介することで、クエリを最適化して必要十分な量を実行して探索します



AWS のデザインシステムを使って
EC サイトの管理画面を実装してください。

“管理画面 テーブル フォーム データ表示”

(クエリ 1 回目)

“レイアウトコンポーネント ResponsiveLayout AppLayout Header”

(クエリ 2 回目)



Amazon Bedrock
Knowledge Base

事前にコードサンプルなどを
入れておく必要があります

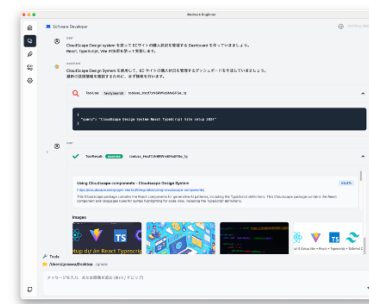
Knowledge Base に複数回最適なクエリを実行

18

ツールを使ってできること

Web Grounding

Tavily Search や URL 指定の検索が可能
文脈に応じて知識の補完をしながら遂行するエージェントが実装できます。



1) React でルーティングを実装するには?

2) react-router 以外の実装手段は?

3) react-router 以外の実装手段は?



URL を指定した検索
Tavily Search による検索

必要な知識が得られるまで
自律的にクエリを変更して知識を探索

開発のシーンに応じてウェブから知識を補完する例

17

インストール方法

アプリをダウンロードするだけ！ (Windows の人はビルドしてね)

Getting Started

Bedrock Engineer is a native app, you can download the app or build the source code to use it.

Download

MacOS:



DOWNLOAD FOR MAC

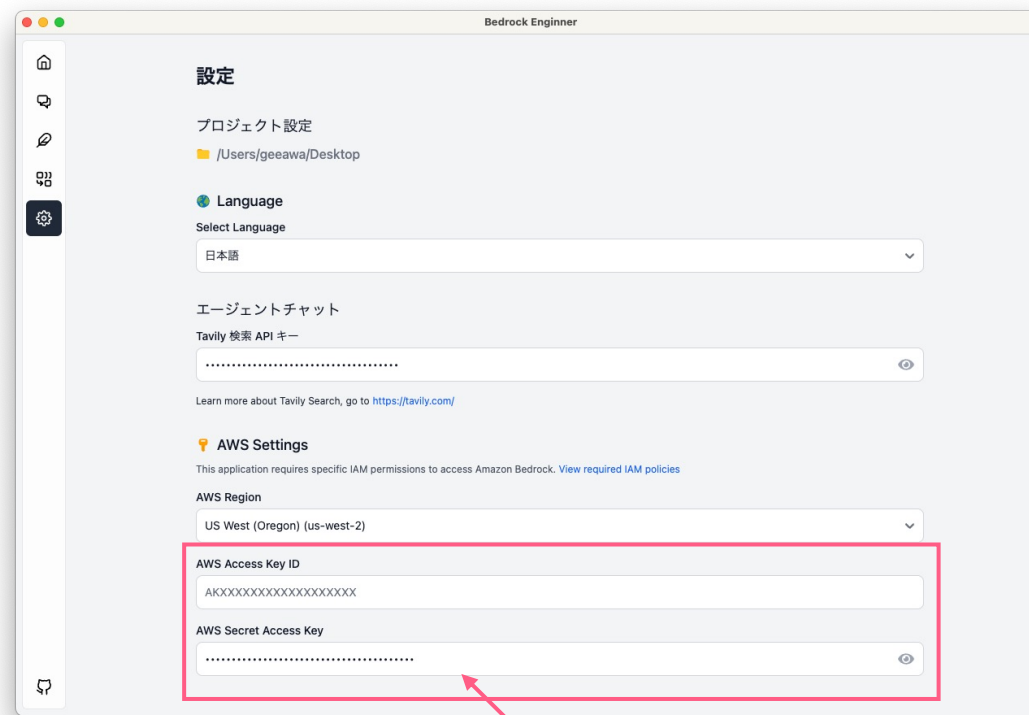
LATEST RELEASE



Click !!

初期設定

設定画面から AWS アクセスキー、シークレットアクセスキーを設定すれば
すぐに利用開始できます



アクセスキー
シークレットアクセスキーを設定

おねがい

- “Bedrock Engineer” で検索！
- X でフィードバックのコメントお待ちしております！！
例) 「こんなアプリができた」 「こんなカスタムエージェントができた」
- GitHub に  をお願いします！！！！

Thank you!

<https://github.com/aws-samples/bedrock-engineer>

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved. Amazon Confidential and Trademark.



ナメック語を理解していない SA は 願いを叶えることができるのか

川路 義隆 (Yoshitaka Kawaji)

アマゾン ウェブ サービス ジャパン合同会社

Solutions Architect



自己紹介

川路 義隆 (かわじ よしたか)

アマゾンウェブサービスジャパン合同会社 ソリューションアーキテクト

小売業のお客様向けのAWS導入・アーキテクチャ支援
アジャイル・サーバーレス

好きな AWS サービス

Amazon API Gateway, AWS Lambda,
AWS Step Functions



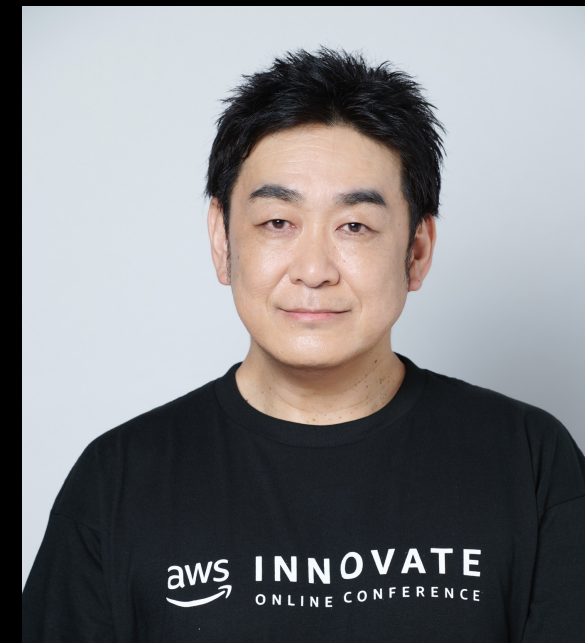
Amazon API Gateway



AWS Lambda



AWS Step Functions



やりたいこと

- ドラゴンボールを集めて願いを3つ叶えたい
- 地球のドラゴンボールでは一つの願いしか叶えられない
- ナメック星には3つの願いを叶えるドラゴンボールが存在する

やりたいこと

- ドラゴンボールを集めて願いを3つ叶えたい
- 地球のドラゴンボールでは一つの願いしか叶えられない
- ナメック星には3つの願いを叶えるドラゴンボールが存在する

ナメック語の習得が必要

ナメック語とはどのようなものなのか？

2025.01.17

COLUMNS

字体は？ 発音は？ ナメック語の「言語的特徴」を、人工言語の作者と一緒に徹底解剖してみた

.....

専門家に聞いてみた



非の呈判の
レキ太/64太
多世:7言W:7
%02000!!!!



https://dragon-ball-official.com/news/01_3155.html

ナメック語とはどのようなものなのか？

- ナメック星人はナメック語を「特別なシーン」でしか使わない
- 会話シーンとポルンガ召喚シーンで「文字の形が違う」
- ナメック語は「地球上で8%しか存在しない」くらい珍しい言語

https://dragon-ball-official.com/news/01_3155.html

ナメック語とはどのようなものなのか？

- ナメック星人はナメック語を「特別なシーン」でしか使わない
- 会話シーンとポルンガ召喚シーンで「文字の形が違う」
- ナメック語は「地球上で8%しか存在しない」くらい珍しい言語

作中に登場する量が少なく、すべて拾い上げても全然量が足りない

https://dragon-ball-official.com/news/01_3155.html

ナメック語とはどのようなものなのか？

“ナメック語は、アフリカ北西部でベルベル人が使う「ティフィナグ文字」に似ている”

https://dragon-ball-official.com/news/01_3155.html

ティフィナグ語

。	⊖	⌘	⌘ ^u	∧	E	÷	ℋ	℔	℔ ^u
A	B	G	G ^w	D	Ḑ	E	F	K	K ^w
⓪	∕	⌒	⌘	ℤ	≲	I	ℋ	⌒	ℓ
H	Ḥ	ε	X	Q	I	J	L	M	N
∴	○	Q	⌒	⊙	⊙	Ⓒ	+	⊕	⌒
U	R	Ṛ	Y	S	Ṣ	C	T	Ṭ	W
⌘	⌘	⌘							
Y	Z	Ẑ							

アプローチ – Amazon Translate

翻訳

Text

Document

ソース言語

日本語 (ja)

ターゲット言語

英語 (en)

テキストの入力

テキストの入力

0 文字、0/10000 バイトが使用されます。 [Info](#)

▶ 追加設定

この翻訳はご期待に添いましたか? ご意見をお寄せください [フィードバック](#)

▶ アプリケーション統合

アイスランド語 (is)

アイルランド語 (ga)

アゼルバイジャン語 (az)

アフリカーンス語 (af)

アムハラ語 (am)

アラビア語 (ar)

アルバニア語 (sq)

アルメニア語 (hy)

イタリア語 (it)

インドネシア語 (id)

ウェールズ語 (cy)

ウクライナ語 (uk)

ティフィナグ語は現在サポート外



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

161

アプローチ – Amazon Nova Pro

例

日本語: こんにちは

ローマ字: Konnichiwa

ティフィナグ文字（ベルベル語での近似音）: ⵙⵏⵏⵉⵙⵉⵎⴰ (kunnisiwa) ※これはあくまでも近似音です。

アプローチ – Amazon Nova Pro

例

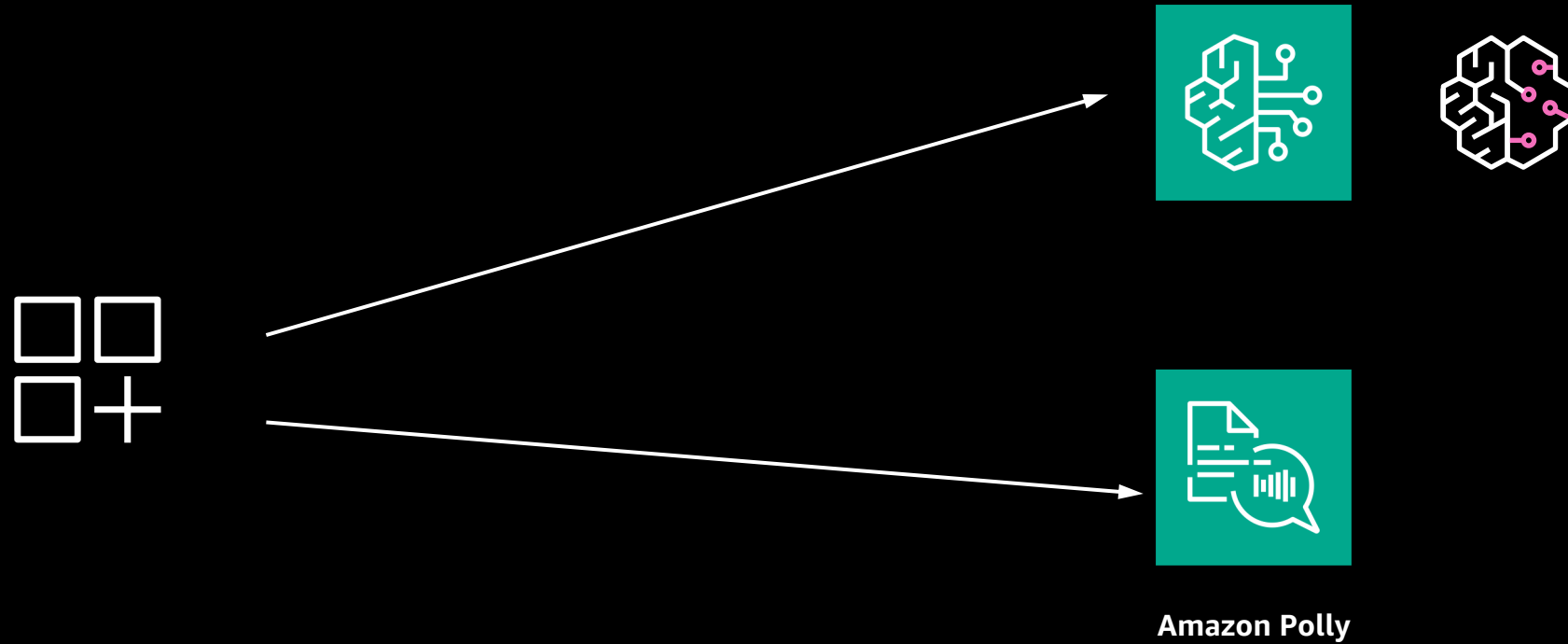
日本語: こんにちは

ローマ字: Konnichiwa

ティフィナグ文字 (Tifinagh) はあくまでも近似音です。

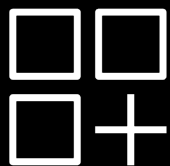
それっぽい！！

アーキテクチャ

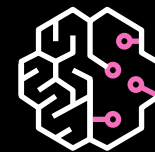


アーキテクチャ

いでよ神龍！！願いをかなえたまえー

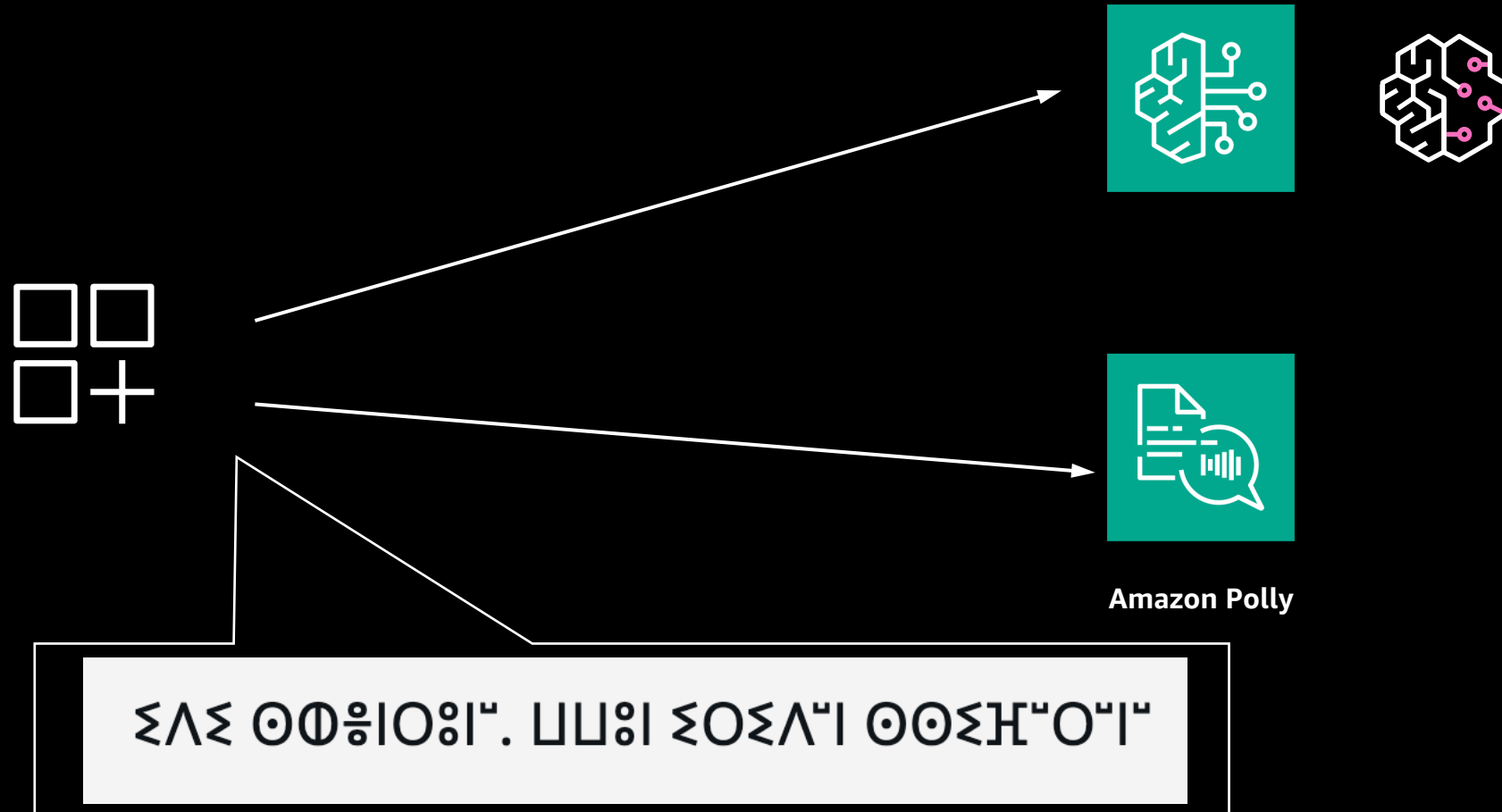


```
[
  {
    "id": "e7876bec-f79d-4fb0-bed6-9ecec8b850af",
    "source": 1,
    "content": [
      {
        "type": "text",
        "text": "「いでよシェンロン。願いを叶え給え」をティフィナグ語に翻訳して"
      }
    ],
    "reasoningContent": [],
    "timestamp": "2025-02-28T00:29:26.085Z",
    "metadata": {
      "inferenceConfig": {
        "temperature": 0.7,
        "topP": 0.9,
        "maxTokenCount": 512,
        "stopSequences": [],
        "guardrailId": "",
        "guardrailVersion": "",
        "latencyOptimized": false
      },
      "modelId": "amazon.nova-pro-v1:0"
    },
    "state": "completed",
    "config": {
      "isOptimized": false
    }
  },
]
```



Amazon Polly

アーキテクチャ



やってみよう



ΣΛΣ ΘΘ%ΙΟ%Ι". ΠΠ%Ι ΣΟΣΛ"Ι ΘΘΣΗ"Ο"Ι"





やってみよう





まとめ

- ナメック語を習得する方法はまだ確立されていない
- 言語として近いものは「ティフィナグ語」がある
- ティフィナグ語で神龍が呼び出せる可能性

まとめ

- ナメック語を習得する方法はまだ確立されていない
- 言語として近いものは「ティフィナグ語」がある
- ティフィナグ語で神龍が呼び出せる可能性

あとは、ナメック星に行くだけ！

AWS Step Functions は CDK で書こう！

高野 賢司

シニア ソリューション アーキテクト
アマゾン ウェブ サービス ジャパン 合同会社

2025/2/26





こうの けんじ

高野 賢司



@konokenj

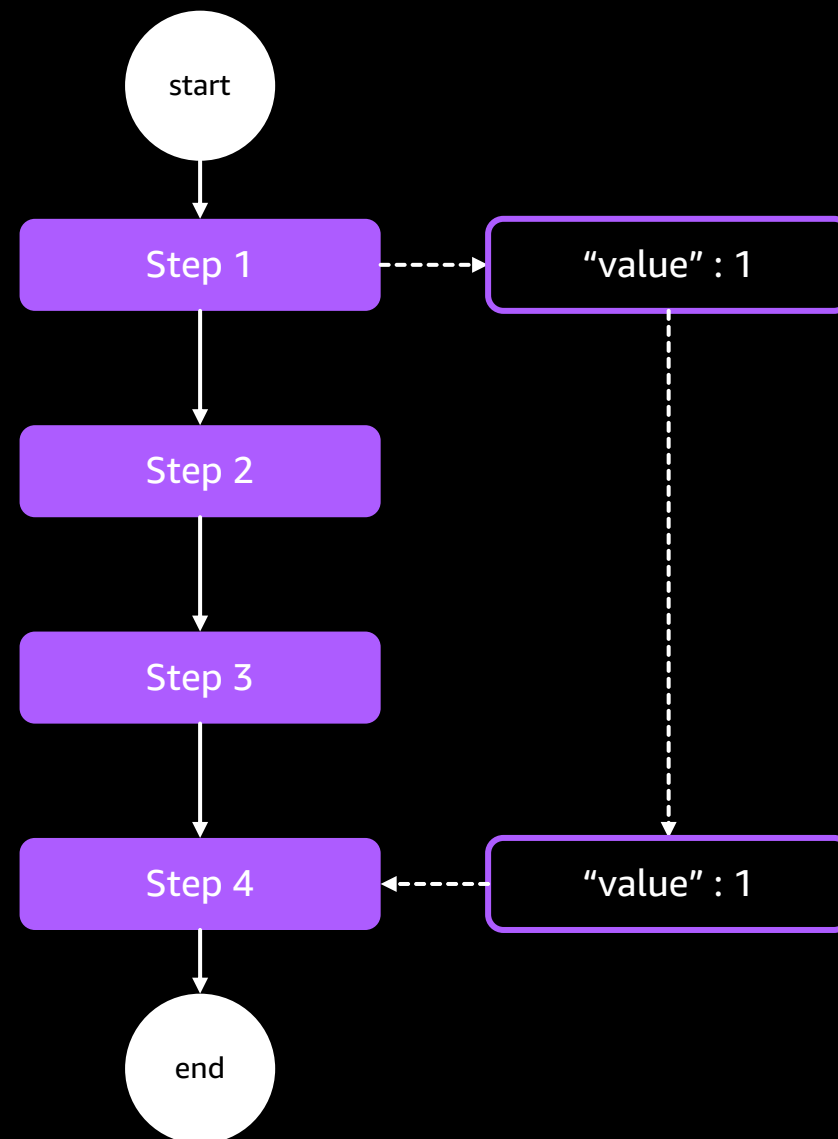
シニア ソリューション アーキテクト @名古屋
アマゾン ウェブ サービス ジャパン 合同会社

- Infrastructure as Code (**laC**) の
技術支援をリード
- Baseline Environment on AWS (**BLEA**) の
開発コアメンバー

<https://github.com/aws-samples/baseline-environment-on-aws>

AWS Step Functions で変数をサポート

- Step Functions 内で**変数**をサポート
 - アップデート前はステップ間で値を引き渡していく必要
 - アップデートにより変数の**保存と参照が簡素化**
- ワークフローの途中で**値の更新**が可能
- 変数のスコープ
 - 変数に格納したのち、後続のステップから参照可能
 - 例外として
Map, Parallel内の変数はその内部でのみ参照可能



JSONata によるデータ変換

- JSONataはOSSのJSON用データクエリおよび変換言語
 - 日付・時刻のフォーマット、文字列操作、四則演算などが可能

商品の価格・数量から合計金額を計算する例

JSONata

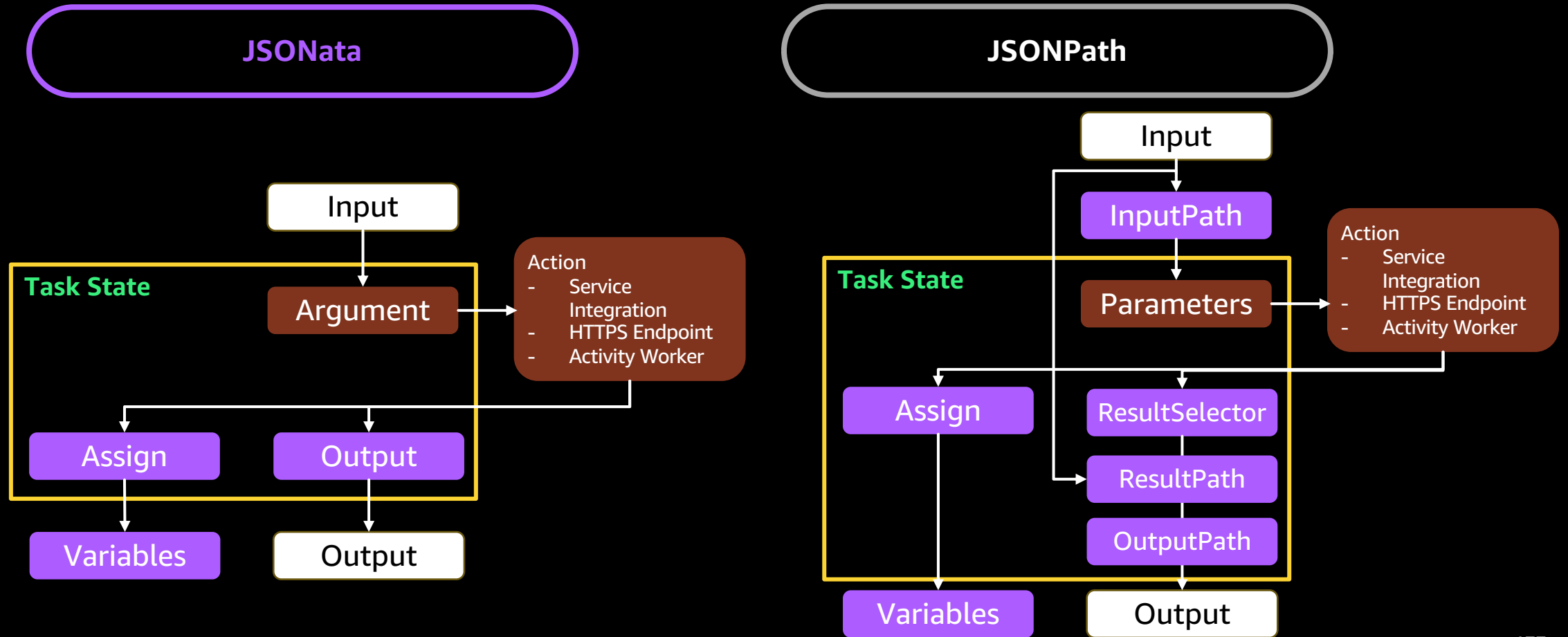
```
"Calculate": {
  "QueryLanguage": "JSONata-2.0",
  "Type": "Task",
  ...
  "Assign": {
    "sum.$": "$sum($states.result.Product.(Price * Count))"
  }
}
```

JSONPath + Intrinsic Functions + Lambda

```
"Get prices and quantities": {
  "Type": "Task",
  ...
  "ResultSelector": {
    "Prices.$": "$.Product.Price",
    "Counts.$": "$.Product.Count"
  },
  "ResultPath": "$.productInfo",
  "Next": "Calculate"
},
"Calculate": {
  "Type": "Task",
  "Resource":
    "arn:....lambda:invoke",
    "Parameters": {
      "FunctionName": "lambdaArn",
      "Payload.$": "$.productInfo"
    },
    "ResultSelector": {
      "sum.$": "$.Payload.sum"
    },
    "ResultPath": "$.ProductSum",
    ...
  }
}
```

JSONata によるデータ変換

- JSONataはJSONPathと比較し変換フィールドの数を減らして**入出力処理を簡略化**



AWS Step Functions L2 コンストラクトでJSONata に対応

- Step Functions の変数と JSONata データ変換が CDK でも簡単に使えるようになった
- 従来の JSONPath ではステート間での値の受け渡しが複雑だった
 - InputPath, Parameters, ResultPath, ResultSelector が Arguments だけになる
- ステートマシン内でのデータ操作のための Lambda 関数を減らせる
- 各ステートの静的メソッド `.jsonata()` / `.jsonPath()` を使うことで適切なプロパティだけが使える

<https://github.com/aws/aws-cdk/releases/tag/v2.178.0>

変数と JSONata を使った AWS Step Functions での開発者エクスペリエンスの簡素化 | Amazon Web Services ブログ <https://aws.amazon.com/jp/blogs/news/simplifying-developer-experience-with-variables-and-jsonata-in-aws-step-functions/>



TypeScript

```
new StateMachine(this, 'StateMachine', {
  definition: workflow,
  queryLanguage: QueryLanguage.JSONATA,
});
```

ステートマシン全体での設定

TypeScript

```
// コンストラクタでのステート作成 (レガシー)
new sfm.Pass(this, 'Constructor Pattern', {
  queryLanguage: sfm.QueryLanguage.JSONATA, // or JSON_PATH
});

// 今後はコンストラクタではなく static method の利用を推奨
// JSONata
sfm.Pass.jsonata(this, 'JSONata Pattern', {
  outputs: { foo: 'bar' },
});
// JSONPath
sfm.Pass.jsonPath(this, 'JSONPath Pattern', {
  outputPath: '$.status',
});
```

ステートごとの設定

AWS Step Functions は CDK で書こう！

- S3 バケットや Lambda 関数、ECS クラスタ等必要なリソースを画面遷移せずに作れる
- grant メソッドで IAM ポリシーを自動設定
- JSONata / JSONPath で型定義が分かっているため使えるプロパティだけが提案される
- `--hotswap` で数秒でデプロイ。
`--hotswap-fallback` はホットスワップ不可の場合のみ CloudFormation 経由でデプロイ

TypeScript

```
const procFunc = new PythonFunction(this, 'ProcessFunc', {
  entry: 'backend/etl',
  runtime: Runtime.PYTHON_3_13,
  index: 'etl/process.py',
});
rawDataBucket.grantRead(procFunc);
workBucket.grantReadWrite(procFunc);
```

Lambda 関数に S3 バケットへの読み取り・書き込み権限を付与

console

```
# ステートマシンを変更したら CDK でデプロイ
npx cdk deploy --all --require-approval=never --hotswap-fallback

# そのままテストしよう!!
aws s3 cp test.json s3://example/
# AWS CLI のファイル入力 (file://) を使えば、一度実行したイベントを再現できる
aws stepfunctions start-execution --state-machine-arn *** --input file://testevent.json
```


CDK なら Amazon Q Developer でセキュリティスキャンもしてくれるよ

- 自動スキャンは Pro Tier のみ
- 検出対象は Amazon CodeGuru Detector Library で確認できる
<https://docs.aws.amazon.com/codeguru/detector-library/typescript/tags/aws-cdk/>

<https://aws.amazon.com/about-aws/whats-new/2024/04/amazon-q-developer-generally-available/>

Amazon CodeWhisperer にて AI を活用した新しいコード修正、IaC サポート、および Visual Studio との統合提供を開始

<https://aws.amazon.com/jp/blogs/news/amazon-codewhisperer-offer-ai-powered-code-remediation-iac-support-and-integration-with-visual-studio/>



CWE-778 - Insufficient Logging CDK Amazon Q ([typescript/insufficient-logging-cdk@v1.0](#))

CWE-778 - Insufficient Logging CDK High

Verifying whether ALL events are logged to CloudWatch is crucial for operators to effectively troubleshoot and audit systems.

[View Details](#) | [Explain](#) | [Fix](#)

Suggested Fix Preview

@@ -224,5 +224,5 @@

```
const stateMachine = new StateMachine(this, "StateMachine", {
  definitionBody: DefinitionBody.fromChainable(Chain.start(parallel)),
});
```

CWE-778 - Insufficient Logging CDK

CWE-319 - AWS insecure transmission CDK High

Bucket is not set to enforce encryption-in-transit, if not explicitly setting this on the bucket policy - the property "enforceSSL" should be set to true.

[View Details](#) | [Explain](#) | [Fix](#)

Suggested Fix Preview

@@ -237,5 +237,5 @@

```
versioned: true,

const sourceBucket = new Bucket(this, "SourceBucket", {
  versioned: true,
  blockPublicAccess: BlockPublicAccess.BLOCK_ALL,
  eventBridgeEnabled: true,
});
```

CWE-319 - AWS insecure transmission CDK

You, 7 days ago • fix code structure

CDK の最新情報は X で配信しています

新しく stable になったモジュール

- `appconfig` (v2.130.0)
- `apigatewayv2` (v2.112.0)
- `synthetics` (v2.99.0)
- `batch` (v2.96.0)
- `appsync` (v2.60.0)

新しい L2 コンストラクト（一部）

- `ecs.ServiceManagedVolume` (v2.122.0)
- `cloudfront.KeyValueStore` (v2.118.0)
- `stepfunctions.DistributedMap` (v2.127.0)
- `codedeploy.EcsDeploymentGroup` (v2.50.0)
- `codebuild.Fleet` (v2.145.0)
- `codepipeline.Pipeline` (PipelineType.V2 / v2.133.0 からデフォルト)



Kenji Kono
@konokenj

<https://x.com/konokenj>

AWS CDK v2.130.0 新機能 ✨ #cdk_releases

1/ AppConfigのL2コンストラクトがalphaを卒業し安定版に 🚀

2/ Bedrockの基盤モデルにStable Diffusion XL v0, v1を追加。Thank you @_watany !

AWS CDK の
アップデート情報を
つぶやいています
#cdk_releases

[https://twitter.com/hashtag/
cdk_releases?src=hashtag_click&f=live](https://twitter.com/hashtag/cdk_releases?src=hashtag_click&f=live)

* v2.29.0 (2022年6月) から



Thank you!

Kenji Kono

 [@konokenj](https://twitter.com/konokenj)



JAWS DAY 2025

LIGHTNING TALKS C-10

AppSync Eventsを使ってみよう！

福井 厚 (Atsushi Fukui)

アマゾンウェブサービスジャパン

シニアデベロッパースペシャリスト ソリューションアーキテクト

2025/03/01



自己紹介

❖ 名前

- ❖ 福井 厚（ふくい あつし）fatsushi@

❖ 所属

- ❖ アマゾン ウェブ サービス ジャパン合同会社
- ❖ プロトタイピング&クラウドエンジニアリング本部
- ❖ シニアデソリューションアーキテクト
デベロッパーズスペシャリスト DevTx

❖ 関心領域

- ❖ ソフトウェア アーキテクチャ、オブジェクト指向設計、アジャイル開発

❖ 好きなAWSサービス

- ❖ サーバーレステクノロジー全般



Agenda

- Demo !
- AWS AppSync とは
- AWS AppSync Events とは

Demo!

スマホでQRコードからWebアプリを表示

Voting App - Red or Blue

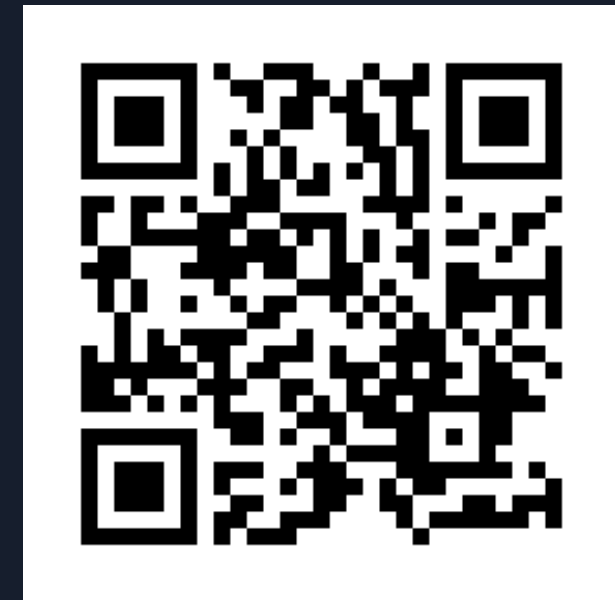
Survey:

I VOTE RED! I VOTE BLUE!

NaN%

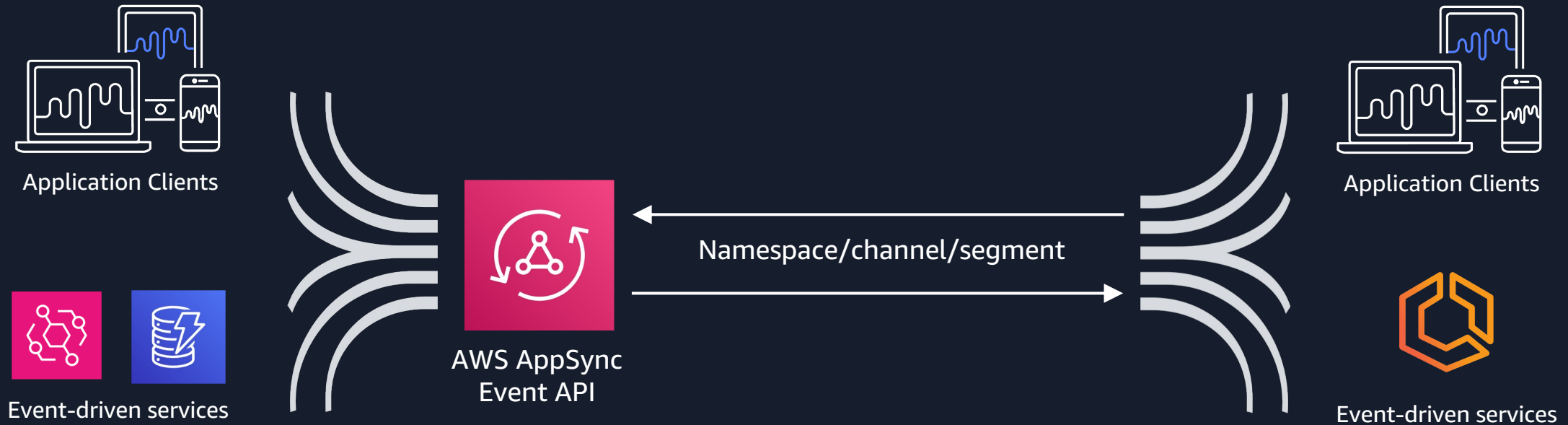
red
blue

Number of Red voted 0 Number of Blue voted 0 Total 0



AppSync Eventsとは？

AppSync Events – パブリッシュ/サブスクライブモデル



パブリッシャ → **WebSockets** ← サブスクライバ

組み込みの接続管理と
メッセージファンアウト

AppSync Events – 主要なメリット

迅速に開始



数分でリアルタイム体験

- ワンクリックでEvent APIバックエンドを設定
- HTTP経由でチャンネルにイベントをパブリッシュ
- 標準的なWebSocket経由でチャンネルを購読

スケーラブル



低遅延で大規模

- デフォルトクォータは1Mメッセージ/秒
- 世界30地域でお客様の近くに
- カスタムドメイン、AWS WAF、ログ、メトリクス

運用負荷を軽減



アプリのビジネスロジックに集中

- 接続管理とファンアウトを内蔵
- APIキー、Cognito、OIDC、Lambda認証
- 組み込みのメッセージフィルターと変換

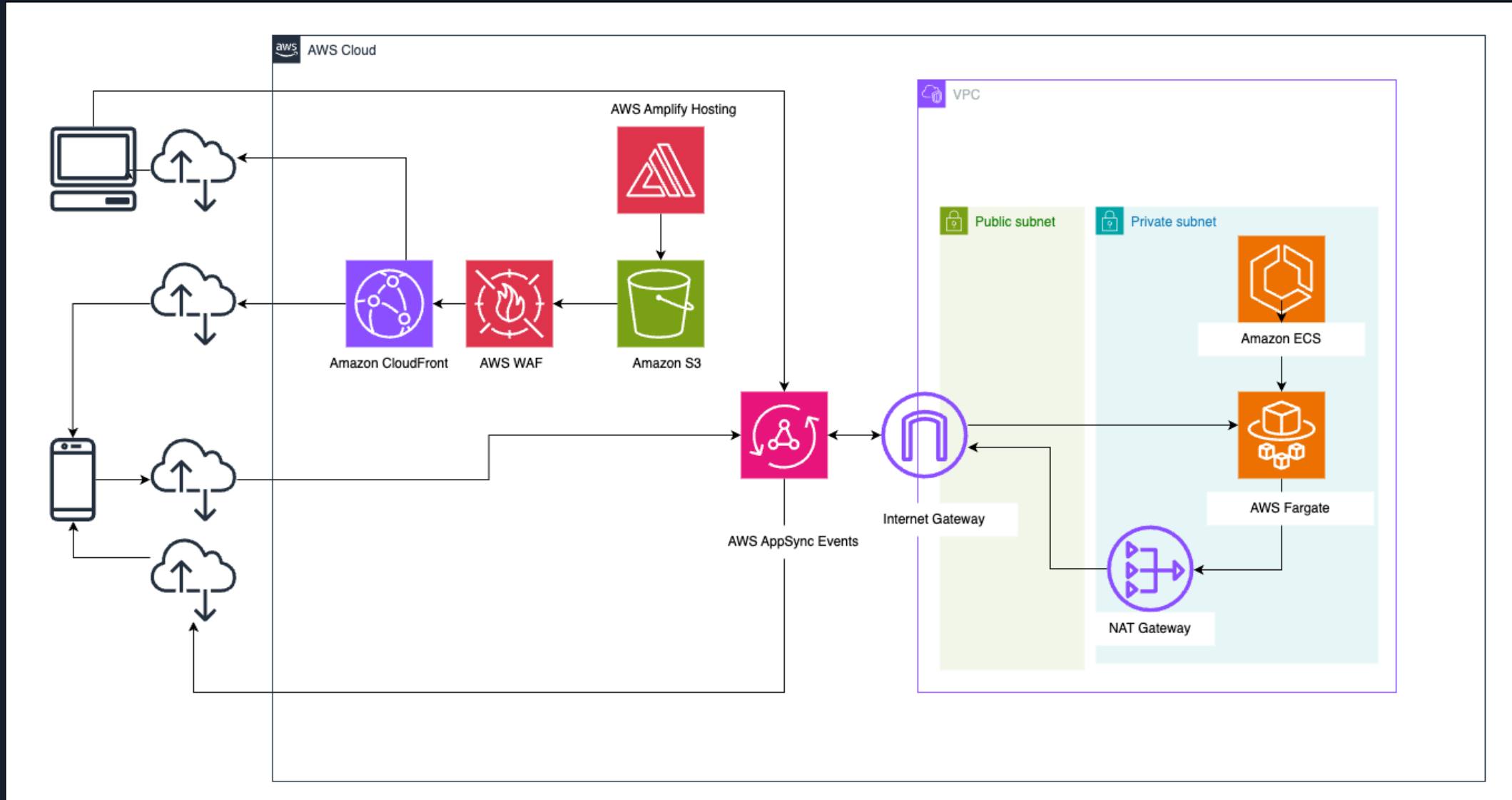
Lower costs



サーバーレスのスケールと使用量に応じた課金

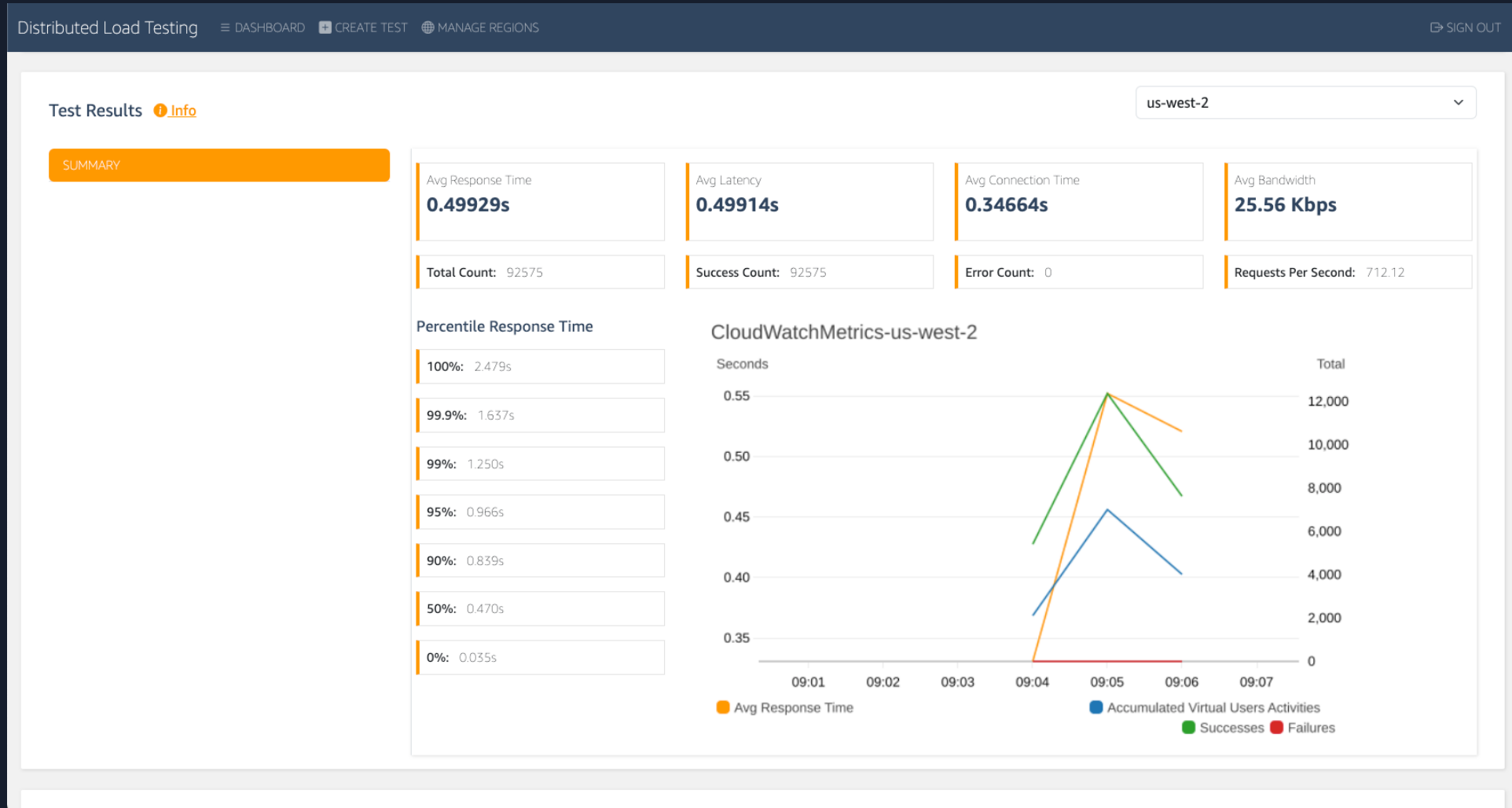
- フルマネージドサーバーレス
- ゆとりのある無料ティア（250k ops/月）
- 100万イベントAPI/1USD

Demo Architecture

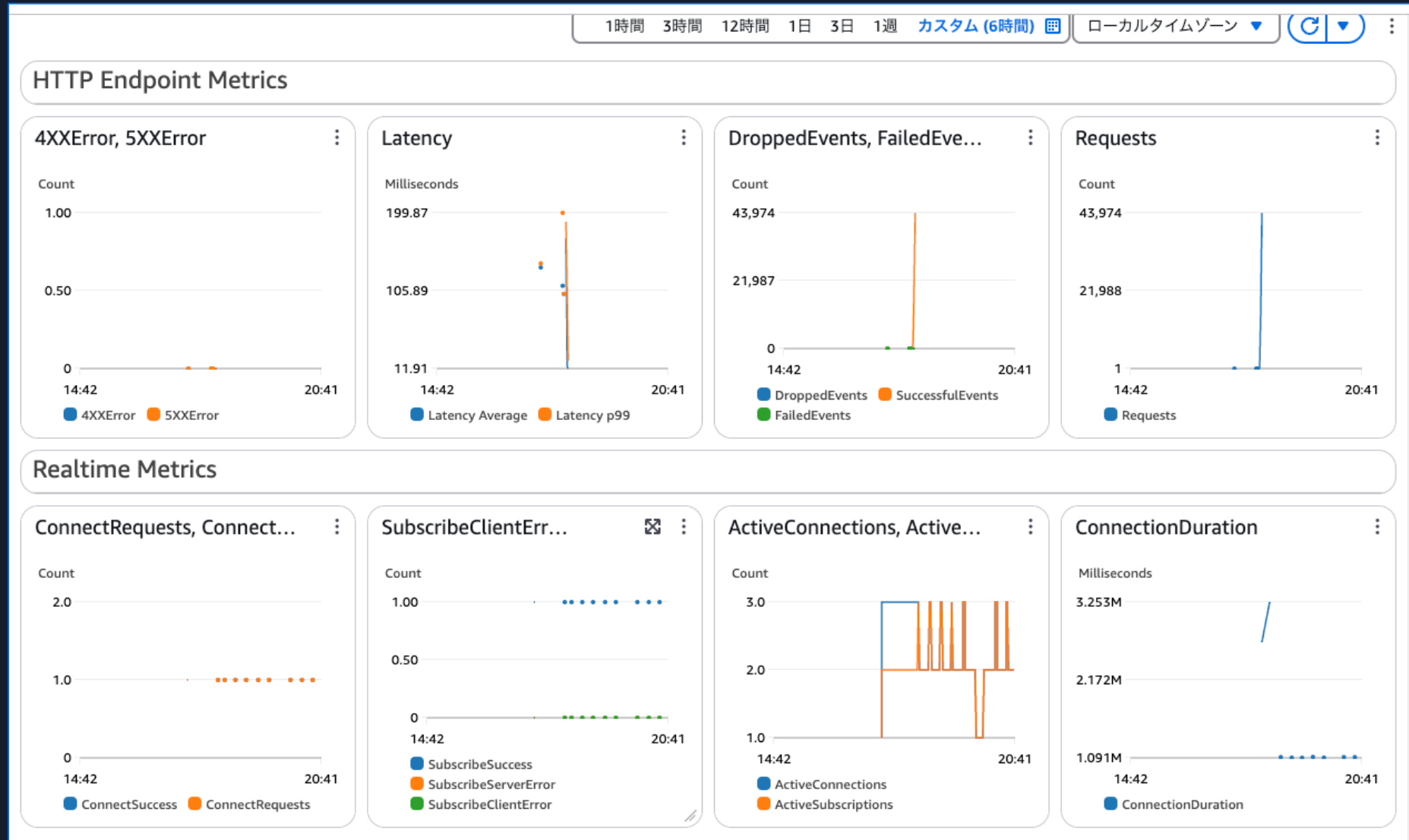


Distributed Load Testing on AWS

<https://docs.aws.amazon.com/solutions/latest/distributed-load-testing-on-aws/solution-overview.html>



AWS AppSync モニタリング



Thank you!

Atsushi Fukui

Senior Developer Specialist Solutions Architect
Amazon Web Services Japan G.K.



セッション : C-10

Please answer this survey. ✎ ✕

アンケート回答にご協力をお願いします！
以下の URL にアクセスしていただくか、スマホは
QR コードを読み込んでご回答ください。

<https://forms.gle/VconwiFxd89HNDk8> //

⋮ 全体アンケート ⋮



⋮ セッションのアンケート ⋮

