

2026.5.21 Oracle Developer Day 2026

**運用担当の敷居を下げよう！**

**- GenAI×OCIで作るアラート自動要約の  
アーキテクチャ**

**NTT西日本株式会社 山塚 友貴**

NTT西日本株式会社 ビジネス営業本部

**山塚 友貴** [yuuki.yamatsuka.ut@west.ntt.co.jp](mailto:yuuki.yamatsuka.ut@west.ntt.co.jp)

## 【略歴】

~2024.3 大学卒業(経営学部)

2024.4~現在 BS EP営業部 社会基盤営業部門 デジタルガバメント担当  
(名古屋)に配属、現在3年目

## 【業務内容】

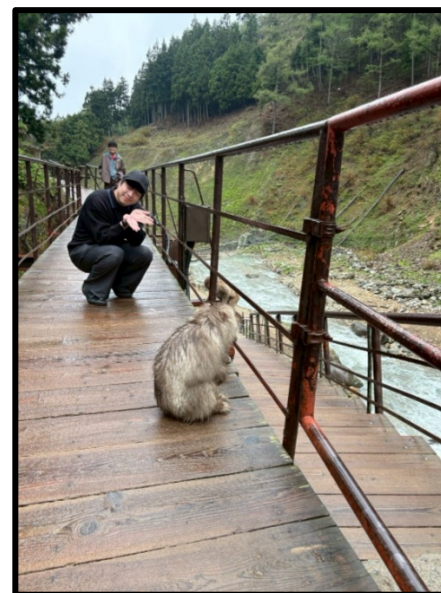
Oracle Cloudを使ったガバメントクラウド上の基幹システムのインフラ基盤に関する(特に非機能周り)設計や構築、保守を担っている。

## 【資格】

OCI Professional資格5種、AWS全冠(新資格除く)、AZ-305  
国内旅行業務取扱管理者 etc...

## 【趣味】

旅行(国内も海外も、一人も複数人も好きです！)



## 1. 私が直面した課題

- ・アラート通知がJSON形式で読みにくい問題
- ・生成AIで解決したいが、壁があった点

## 2. 2026年1月のアップデートで何が変わったか

- ・プライベートエンドポイント×共有モデルへの対応

## 3. 採用した構成とポイント

- ・全体像(東京⇔大阪のクロスリージョン構成)
- ・通知までの流れ(Functions→Generative AI→通知)

## 4. 結果と今後の課題

- ・Before / After
- ・ハルシネーション対策、プロンプトチューニング等

## 5. まとめ

## <以下実際に届くEventからのJSON形式の通知例>

```
{
  "eventType" : "com.oraclecloud.objectstorage.createobject",
  "cloudEventsVersion" : "0.1",
  "eventTypeVersion" : "2.0",
  "source" : "ObjectStorage",
  "eventTime" : "2026-05-14T07:12:11Z",
  "contentType" : "application/json",
  "data" : {
    "compartmentId" : "ocid1.compartment.oc1..xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
    "compartmentName" : "test-compartment",
    "resourceName" : "test/oci.text",
    "resourceId" : "/n/abcdefgh/b/oci-test-bucket/o/test/oci.text",
    "availabilityDomain" : "NRT-AD-1",
    "additionalDetails" : {
      "bucketName" : "oci-test-bucket",
      "versionId" : "12345678-abcd-12ab-cd34-abcdefghi",
      "archivalState" : "Available",
      "namespace" : "abcdefgh",
      "bucketId" : "ocid1.bucket.oc1.ap-tokyo-1..xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
      "eTag" : "aaaaaaaa-bbbb-cccc-dddd-eeeeeeee"
    }
  },
  "eventID" : "asdfghjk-1234-5678-qazw-qertyuio",
  "extensions" : {
    "compartmentId" : "ocid1.compartment.oc1..xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
  }
}
```

・(特に普段見慣れていない人にとっては)  
見ただけで瞬時にどこで何が起こったのかが  
わからない。

・見たところでどうしていいかわからない場合  
もある。(初動対応の動きが難しい)

・本当に必要な部分がどこかわかりにくい。

可読性を高めることでより効率的に、またいい意味で運用を担当することのハードルを下げたい。  
→Generative AIを活用した運用の効率化ができないか

今回の課題を考え始めた2025年秋ごろの時点では以下のような問題があった。

## ① Generative AIが日本では大阪リージョンにしかない。

→メインにしている東京リージョンではそもそもサービス自体がリリースされていないため、東京リージョンにある各種アラートからプライベート空間内において自然体で接続できない。

## ② インターネットの経由はセキュリティ上避けたい

→大阪のGenerative AIにインターネット経由であればアクセスが可能だったが、セキュリティの都合上選択できず。

## ③ 専有ホスティングモデルはコストが高い

→アラート要約レベルで使うには専有ホストは高級すぎる。(100通知当たり共有モデルは150円/月程度なのに対し、専有ホストでは30万円/月以上かかる見込み)

# OCIにアップデートが入り...

2026年1月末、Generative AIに関する機能のアップデートが！

(参考：[OCI生成AIでのオンデマンド・モードでのプライベート・エンドポイントの使用](#))

## ① Generative AIが日本では大阪リージョンにしかない。

→メインにしている東京リージョンではそもそもサービス自体がリリースされていないため、東京リージョンにある各種アラートからプライベート空間内において自然体で接続できない。

## ② インターネットの経由はセキュリティ上避けたい

★2026年1月下旬、プライベートエンドポイントをGenerative AIの共有モデルに対しても利用できるように！

### 従来はできなかったこと

プライベート空間内で完結する東京リージョンからGenerative AIの共有モデルの利用

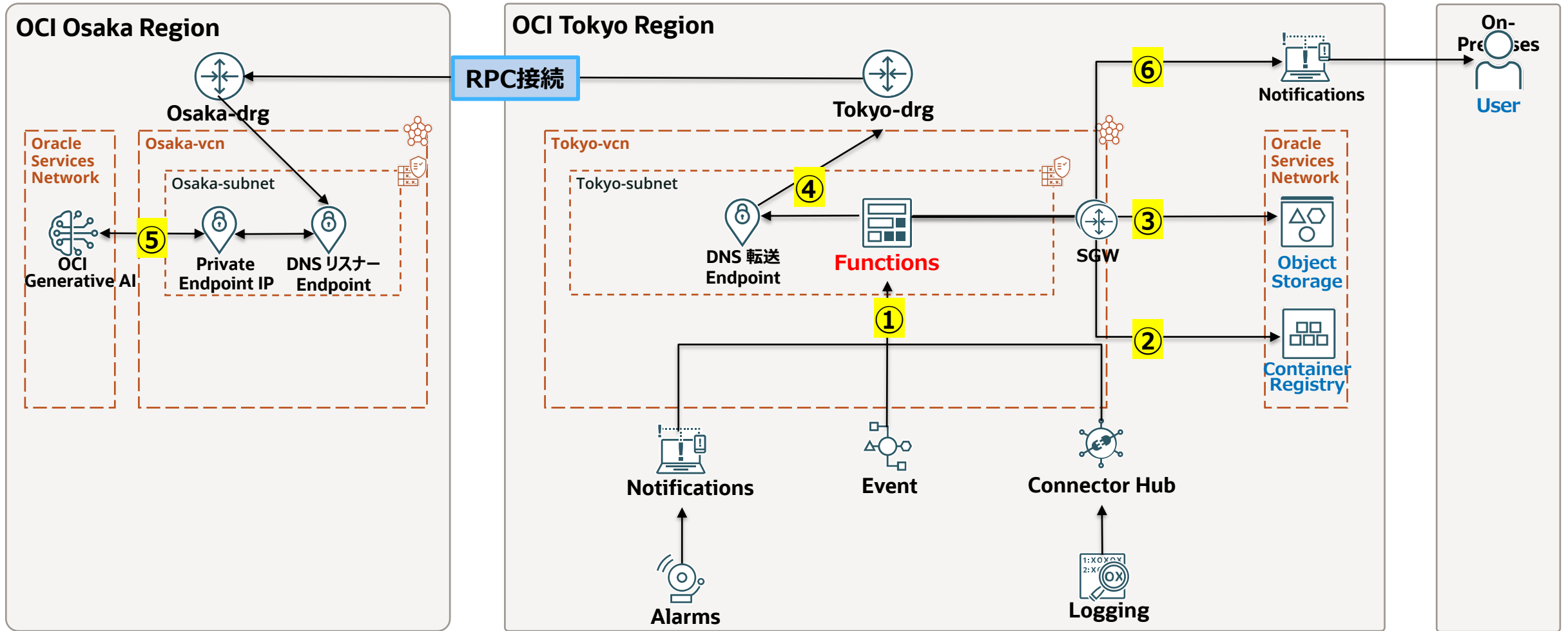


### できるようになったこと

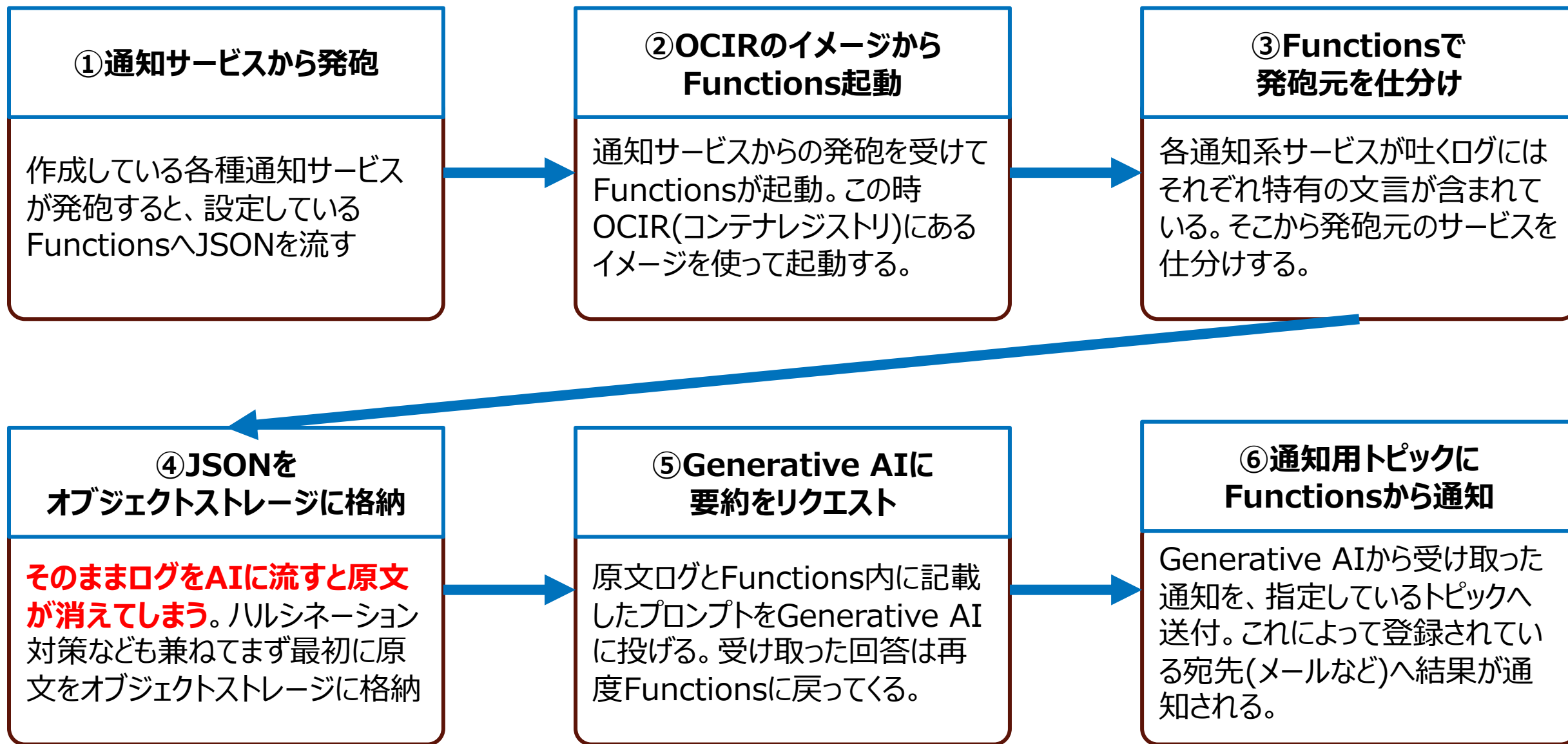
プライベートエンドポイントを設置することで、東京からもプライベート空間で接続が可能に

# 実際の構成

Generative AIのプライベートエンドポイントを活用し、閉域空間でアラートの発火から通知が届く構成を設計。



# 通知までの流れとポイント



# 結果 : Before / After

実際にアラートを発砲させ手元に届く形式としては以下のように変わりました。

## Before

```
{
  "eventType": "com.oraclecloud.objectstorage.createobject",
  "cloudEventsVersion": "0.1",
  "eventTypeVersion": "2.0",
  "source": "ObjectStorage",
  "eventTime": "2026-05-14T07:12:11Z",
  "contentType": "application/json",
  "data": {
    "compartmentId": "ocid1.compartment.oc1..xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
    "compartmentName": "test-compartment",
    "resourceName": "test/oci.text",
    "resourceId": "/n/abcdefgh/b/oci-test-bucket/o/test/oci.text",
    "availabilityDomain": "NRT-AD-1",
    "additionalDetails": {
      "bucketName": "oci-test-bucket",
      "versionId": "12345678-abcd-12ab-cd34-abcdeghi",
      "archivalState": "Available",
      "namespace": "abcdefgh",
      "bucketId": "ocid1.bucket.oc1.ap-tokyo-1..xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
      "etag": "aaaaaaaa-bbbb-cccc-dddd-eeeeeeee"
    }
  },
  "eventID": "asdfghjk-1234-5678-qazw-qertyuio",
  "extensions": {
    "compartmentId": "ocid1.compartment.oc1..xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
  }
}
```

## After

※出力されている項目はプロンプトで指定したものの。

### # [概要] ← 発砲したアラートの概要

「oci-test-bucket」バケットにて、オブジェクトの作成イベントが発生しました。

### # [詳細分析] ← JSONログから読み取れるリソース情報などのまとめ

Object Storageにて、以下のオブジェクトが作成されました:

- > リソース名: "test/oci.text"
- > バケット名: "oci-test-bucket"
- > 可用性ドメイン: "NRT-AD-1"

イベントタイプは「com.oraclecloud.objectstorage.createobject」であり、オブジェクトが新規にアップロードまたは作成されたことを示しています。アーカイブ状態は「Available」となっており、即座にアクセス可能な状態です。

このイベントは通常の運用操作の一環である可能性が高いですが、意図しないファイルのアップロードや、不正アクセスによるオブジェクト作成の可能性も考慮する必要があります。

### # [推奨されるアクション] ← 今回の内容を受けて行った方が良い推奨アクション

1. 該当のオブジェクト「test/oci.text」が意図したアップロードかどうかを確認してください。
2. 心当たりがない場合は、OCIコンソールの監査ログから、誰がいつこのオブジェクトを作成したかを調査してください。
3. 必要に応じて、バケットのIAMポリシーや事前認証リクエストの設定を見直し、意図しないアクセスがないかを確認してください。

### # [ログ参照]

パス: 20260514/071211\_Event\_a1b2c3d4.json ← ログ原文の保管先

### ① 生成AIによる情報の信頼性を、より効率的に担保するためにはどうしていくか

- ・「推奨アクション」が的外れな可能性もあるため、人による確認は現状発生している。
- ・効率化を優先して正確性が悪くなる、ということは避ける必要がある。

### ② プロンプトのチューニング

- ・運用をしていくうえでほしい回答や、はたまた unnecessary な情報を精査しプロンプトを改修していく必要がある。

### ③ 対応状況の記録と分析

- ・要約を送るだけでなく、発生したアラートの集計、レポートの生成なども行いたい。
- ・実際に対応したことがあるエラーなどは、より正確な対処方法などを出力させるようなアーキテクチャの設計。

## 【課題】

OCIのアラート通知がJSON形式で読みにくい

→Generative AIを使って対応したいが、リージョン制約とセキュリティの壁があった。

## 【解決策】

Generative AIプライベートエンドポイント + DRGのリージョン間接続を用いて、「クロスリージョン×プライベート」の構成を実現

## 【学び】

- ①リソースが使いたいリージョンにない場合の設計パターンを考える難しさ。
- ②新機能のキャッチアップが今ある課題の解決につながることもある。→ **アンテナを張ることの重要性。**
- ③やはり生成AIは便利ではあるものの、「利便性」と「信頼性」のバランスをどう調整していくか、継続して考えていく必要がある。

★より細かい内容については**NTT西日本エンジニアブログにて記事が公開されておりますので、ぜひこちらもお覧ください！**

前編：<https://engineers.ntt-west.co.jp/entry/oci-genai-architect>

後編：<https://engineers.ntt-west.co.jp/entry/oci-genai-functions>

(「**NTT西日本 ブログ oci**」でも記事がヒットします！)