Amazon S3 セキュリティ ベストプラクティスご紹介

アマゾンウェブサービスジャパン株式会社

Senior Partner Solutions Architect
Takanori Ohba

2021 Sep 16

Who am I ?

@takaohba

- 大場 崇令（オオバ タカノリ）
  - Senior Partner Solutions Architect @Amazon Web Services Japan K.K. (Joined 2015/12)

- Background
  - AWS テクニカルトレーナー@AWSJ K.K.
  - Web サービスのインフラエンジニア
  - 国内クラウドベンダーにてテクニカルサポート

- 好きな AWS サービス
  - AWS Well-Architected Tool
  - AWS Systems Manager
  - AWS Service Catalog

Well-Architected Lead

本日のゴールと対象者

- ゴール
    - Amazon S3 のセキュリティベストプラクティスを理解し、活用できるようになること

- 想定対象者
    - Amazon S3 のセキュリティ対策に悩まれている方
    - AWS サービス (AWS IAM等) の基本的機能、概念について理解されている方

aws

本日の内容

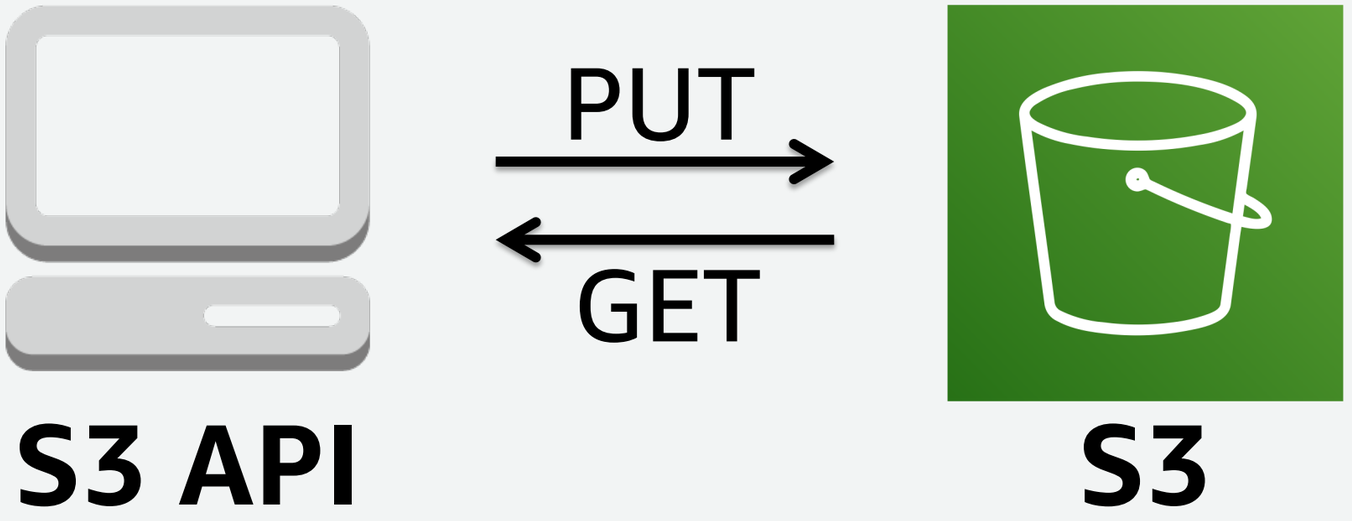
- Amazon S3 にデータが格納されるシーン
- Amazon S3 をセキュアに活用するにあたっての基本要素
- Amazon S3 のセキュリティとそのベストプラクティス
- まとめ

aws

Amazon S3にデータが格納されるシーン

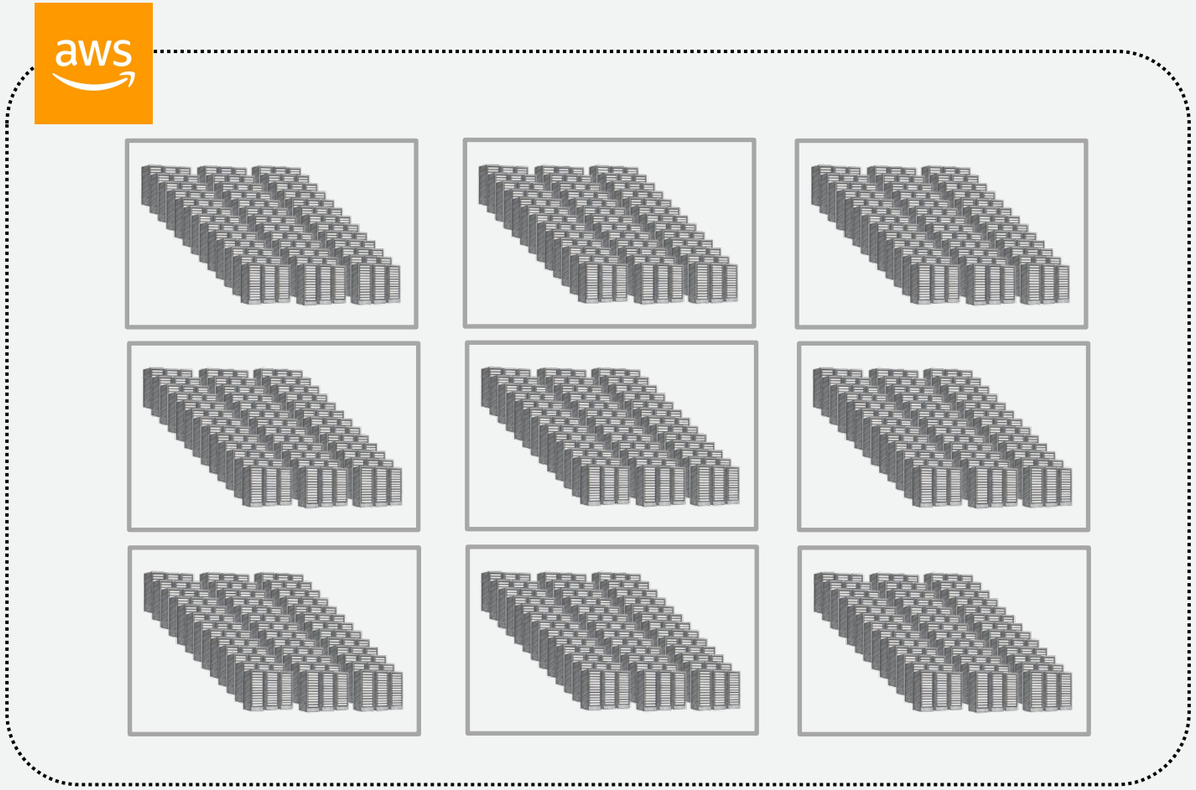
Amazon S3 とは

Amazon Simple Storage Service (S3) は、ユーザがデータを安全に、容量制限なく、データ保存が可能な、クラウド時代のオブジェクトストレージです。



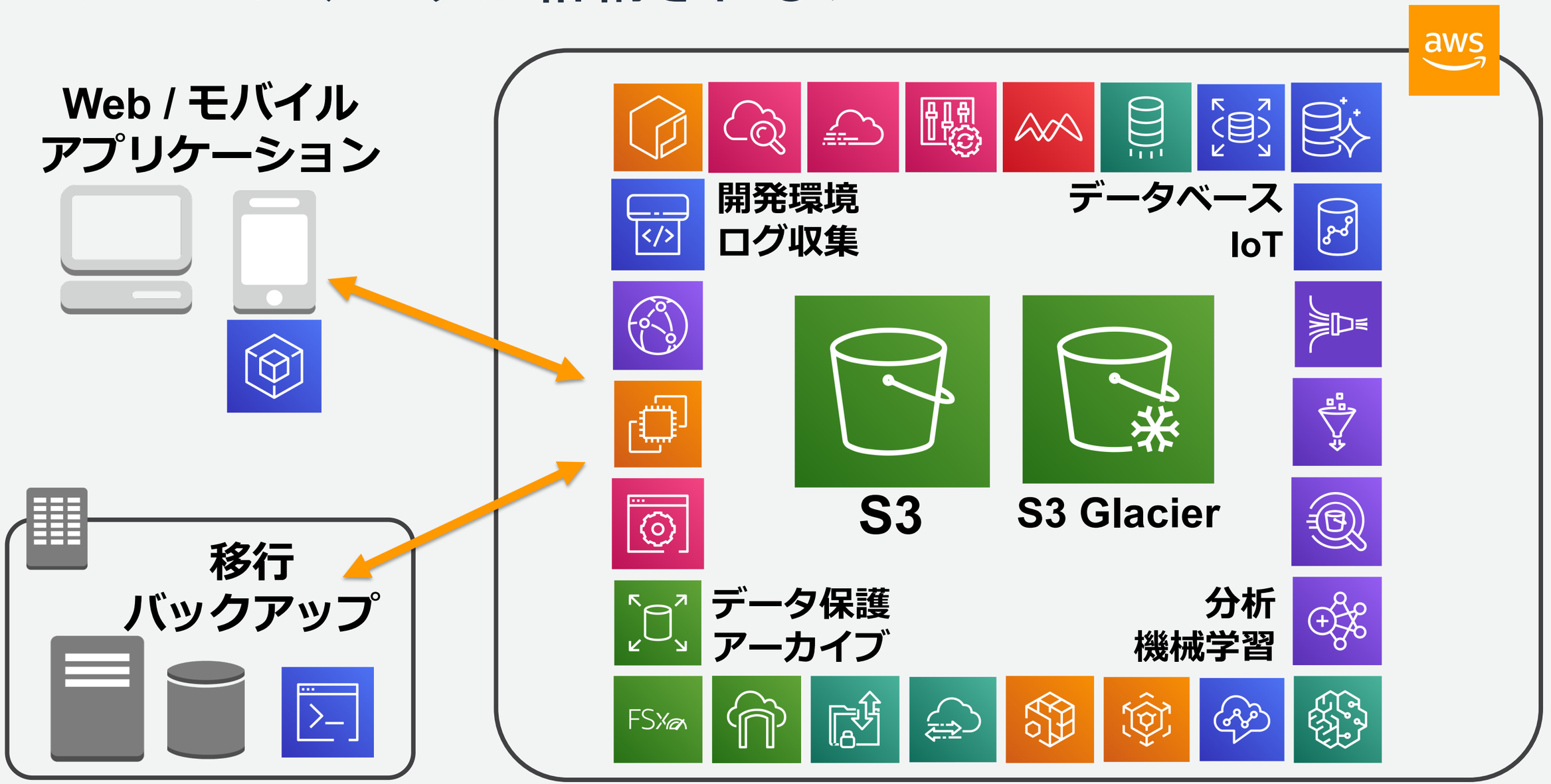
Amazon S3 の特徴

- **容量無制限**
  - 1 ファイル最大 5TB まで
- **高い耐久性**
  - 99.999999999%
- **安価なストレージ**
  - 容量単価:月額 1GB / 約 3 円※
- **スケーラブルで安定した性能**
  - データ容量に依存しない性能
    - ユーザが、サーバ台数、媒体本数や RAID、RAID コントローラを考える必要がない



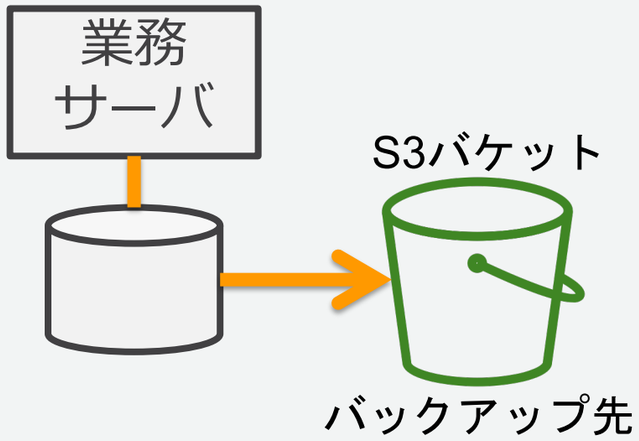
※2021年9月 https://aws.amazon.com/jp/s3/pricing/
東京リージョンにおけるスタンダードが、US$0.025/GB

Amazon S3にデータが格納されるシーン



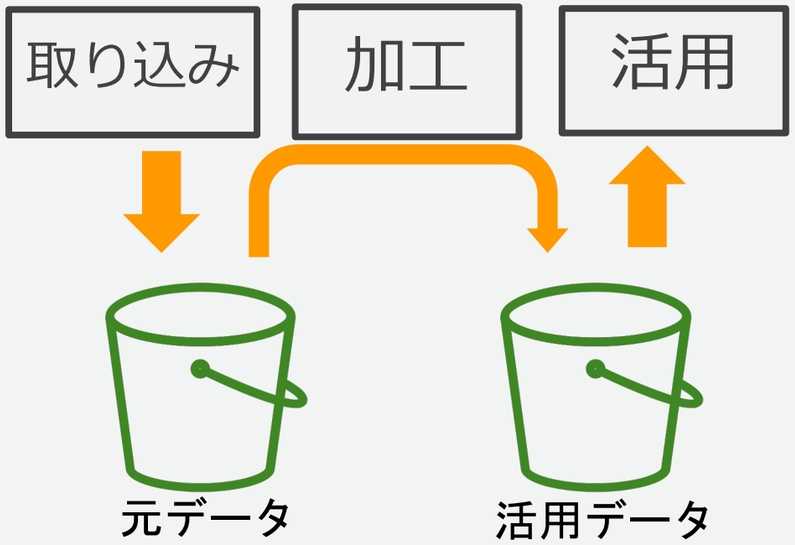
Amazon S3のユースケースの分類

データ保護・移行



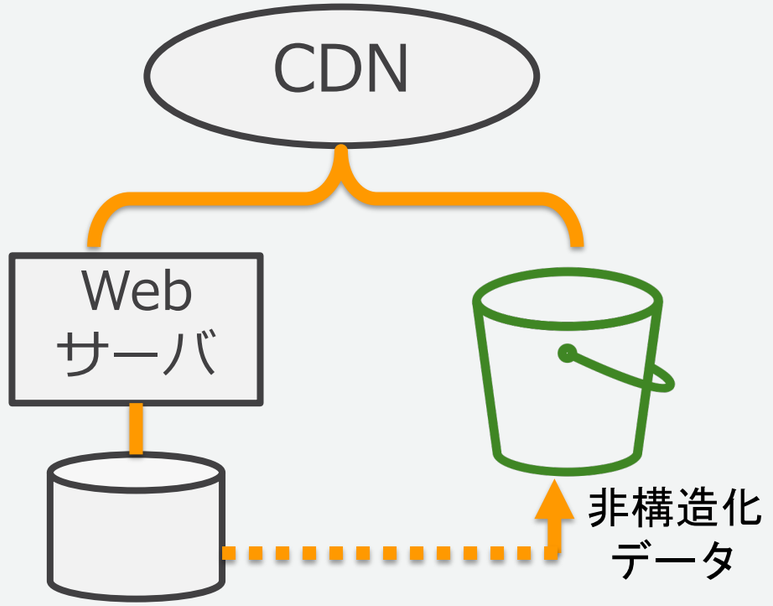
- バックアップによる活用
- データ移行
- サーバマイグレーション

データレイク



- データ分析基盤
- 機械学習・IoT
- HPC・映像処理

Webサービスのコンテンツオフロード



- 写真管理サイト
- 動画サイト
- ECサイト

Amazon S3をセキュアに活用するにあたっての基本要素

aws

AWSのセキュリティ方針

セキュリティは AWS における**最優先事項**

セキュリティに対する継続的な投資

- セキュリティ専門部隊の設置

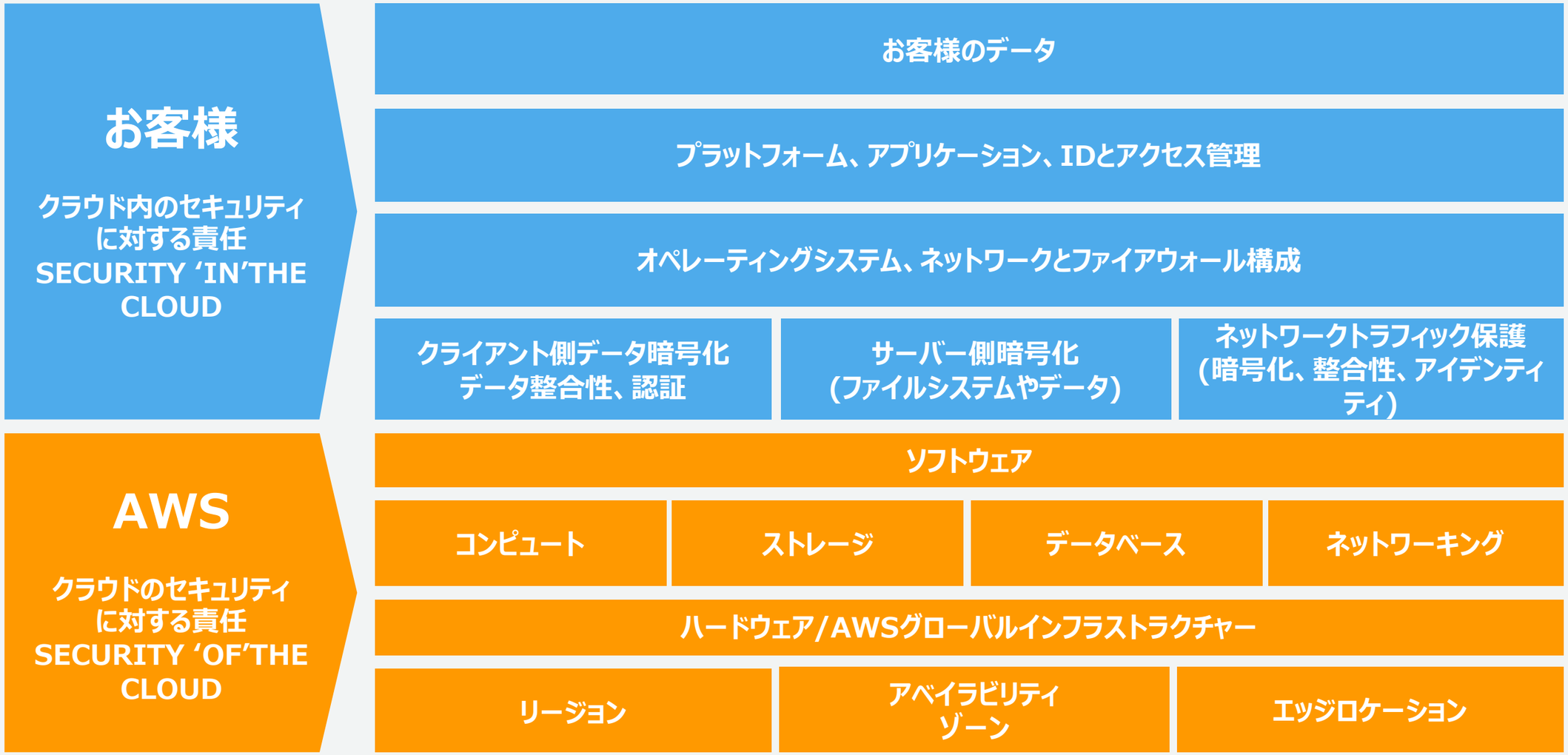
セキュリティ関連**ホワイトペーパー**の公開

- AWS セキュリティセンター
- https://aws.amazon.com/jp/security/

複数の**第三者認証**を取得

- AWS コンプライアンス
- https://aws.amazon.com/jp/compliance/

責任共有モデル



https://aws.amazon.com/jp/compliance/shared-responsibility-model/

AWSでデータの安全確保をサポートする方法

- 最小権限の原則 - セキュリティのベストプラクティス
  - アクセス許可は最小限のセットから開始する
  - 必要に応じて、アクセス許可を追加付与する

- 適切なアクセス許可を定義するために整理しておくこと
  - 利用するAWSサービスがサポートするアクションは何か
  - 目的のタスクにおいて、必要な API 要件
  - それらのアクション実行に必要なアクセス許可設定

aws

AWS Well-Architected(W-A) Framework とは?

クラウドにおけるシステム設計・運用のベストプラクティス集

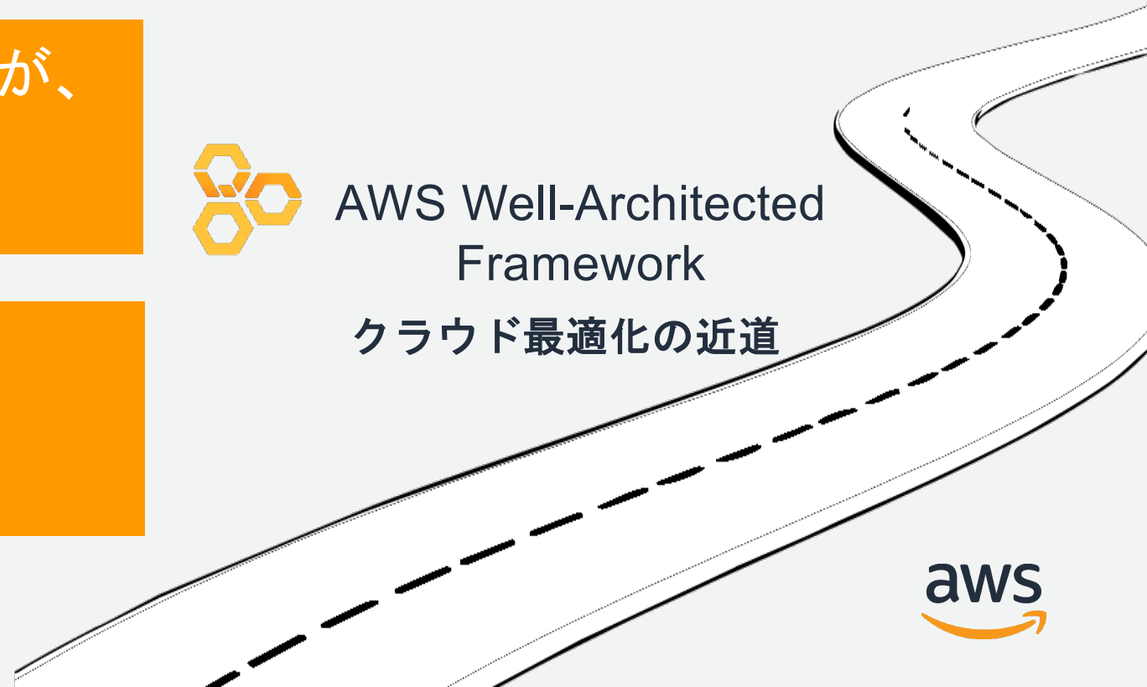
運用性 | セキュリティ | 信頼性 | パフォーマンス効率 | コストの最適化

AWSのソリューションアーキテクト(SA)が、
10年以上に渡り、数多くのお客様へ
技術支援をしてきた経験の集大成

効率的なTransformationを
Well-Architected FrameworkとAWSのSAがお手伝い

AWS Well-Architected Framework

クラウド最適化の近道



W-A Framework 質問項目 と ベストプラクティス

必ず全てにOK(パス)しなければならないものではない
ステークホルダー間で現状が共有され、認識・検討されていることが重要

SEC 1. SEC 2. … SEC10.

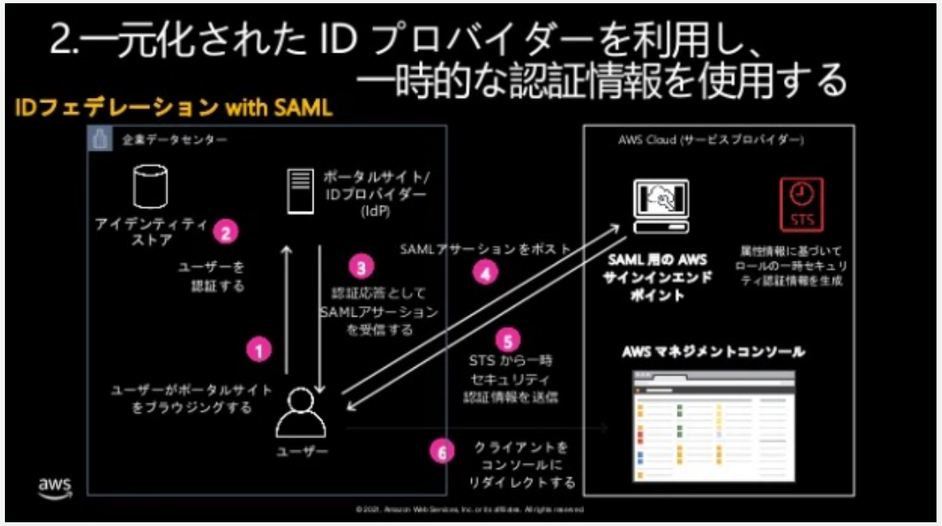
柱ごとに質問項目が定義

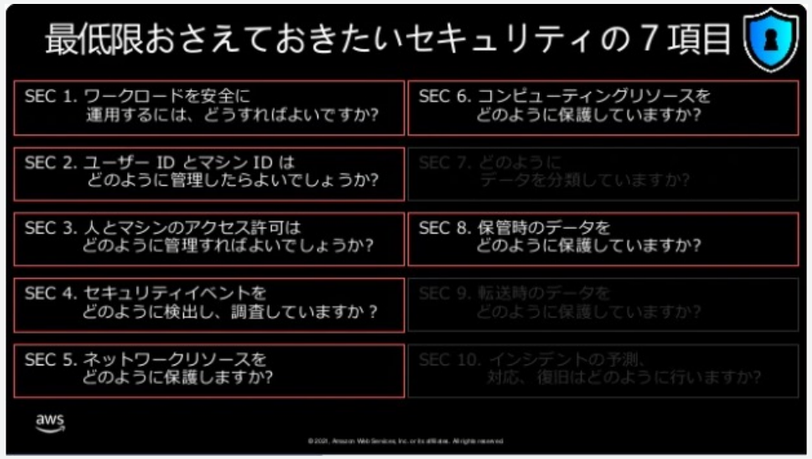
SEC 1: ワークロードを安全に運用するには、どうすればよいですか?

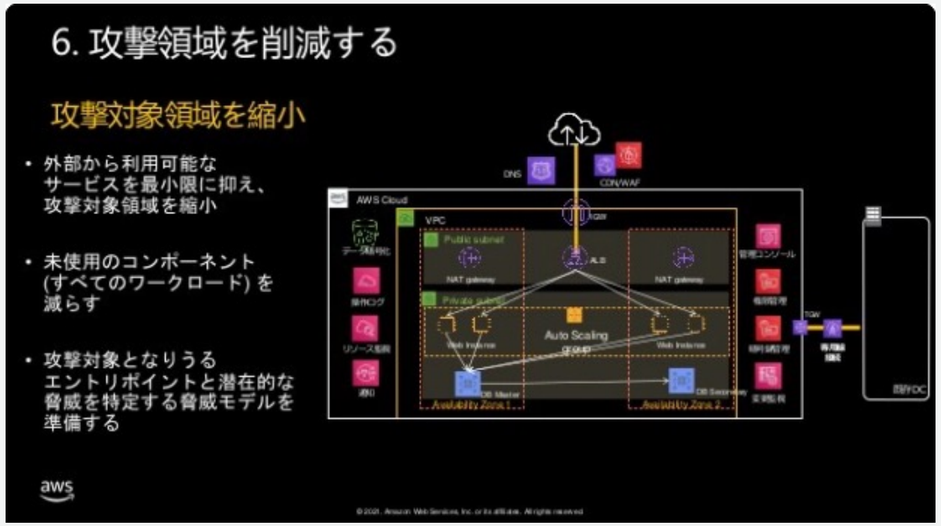
- アカウントを使用してワークロードを分ける
- AWS アカウントのセキュリティを確保する
  :
- 新しいセキュリティサービスと
  機能を定期的に評価および実装する

それぞれ、ベストプラクティスが
そのままチェック項目となっている
(それに沿っているか?の観点で「レビュー」
& ON/OFFすることで「回答」とする)

W-A Framework セキュリティの資料







https://www.slideshare.net/AmazonWebServicesJapan/aws-wellarchitected-security

W-A Framework：セキュリティ対策ドメイン

	ドメイン	内容
1	セキュリティの基礎	AWS アカウントをワークロード毎に分ける 最新情報を入手 セキュリティプロセス、テスト、検証を自動化
2	アイデンティティとアクセス管理	IAM による論理的アクセス制御 パスワードポリシー 最小権限
3	検出	ログの取得・管理 自動化された脅威検出
4	インフラストラクチャ保護	ネットワーク設計（セグメンテーション等） DDoS 対策
5	データ保護	データの暗号化 KMS を使ったキーの管理
6	インシデント対応	インシデントの発生、通知、対応の自動化 DevSecOps

AWS Well-Architected フレームワーク – セキュリティの柱
https://docs.aws.amazon.com/ja_jp/wellarchitected/latest/security-pillar/welcome.html

Amazon S3のセキュリティとそのベストプラクティス

想定される課題と対策

課題	対策
偶発的な情報の開示	IAMによる適切な権限管理 VPCネットワークの理解 Block Public Access 暗号化
データの不整合	IAMによる適切な権限管理 データの整合性チェック バージョニング / Object Lock
過失による削除	バージョニング/ Object Lock MFA Delete
証跡の取得 ユーザー行動の変化を観察	AWS CloudTrail によるログ取得 Amazon Macie の利用

aws

想定される課題と対策

課題	対策
偶発的な情報の開示	**IAMによる適切な権限管理** VPCネットワークの理解 **Block Public Access** **暗号化**
データの不整合	IAMによる適切な権限管理 データの整合性チェック バージョニング / Object Lock
過失による削除	バージョニング/ Object Lock MFA Delete
証跡の取得 ユーザー行動の変化を観察	AWS CloudTrail によるログ取得 Amazon Macie の利用

不正アクセスを防ぐ

IAM ユーザー/アクセスキーは共有しない

- IAM ユーザーは一人に一つ
- git リポジトリ、ファイル共有サービス、ファイルサーバーなどに置かない
  - git に置かれていないかは git-secrets でチェックする
  - S3 に置かれていないかはMacieで確認可能
- メールや Slack で送りあわない

aws

アクセスキーではなくIAMロールを活用

- AWS 環境内でのアクセスはすべてIAMロールで管理可能
- IAM ユーザーのアクセスキーが必要なケース

1. **開発時の AWS CLI / SDK によるアクセス**
    - 本番環境と開発環境はアカウントを分け、本番環境にはアクセスキーを発行しない。開発環境は一時的なアクセスキーを極力利用する。
2. **外部サーバー / 管理サービスからの AWS リソースへのアクセス**
    - 提供するアクセスキーの権限は
      最小限に絞り、定期的にローテーションする

- 不要なアクセスキーが作られていないか定期的に
  IAM 認証レポートでチェック

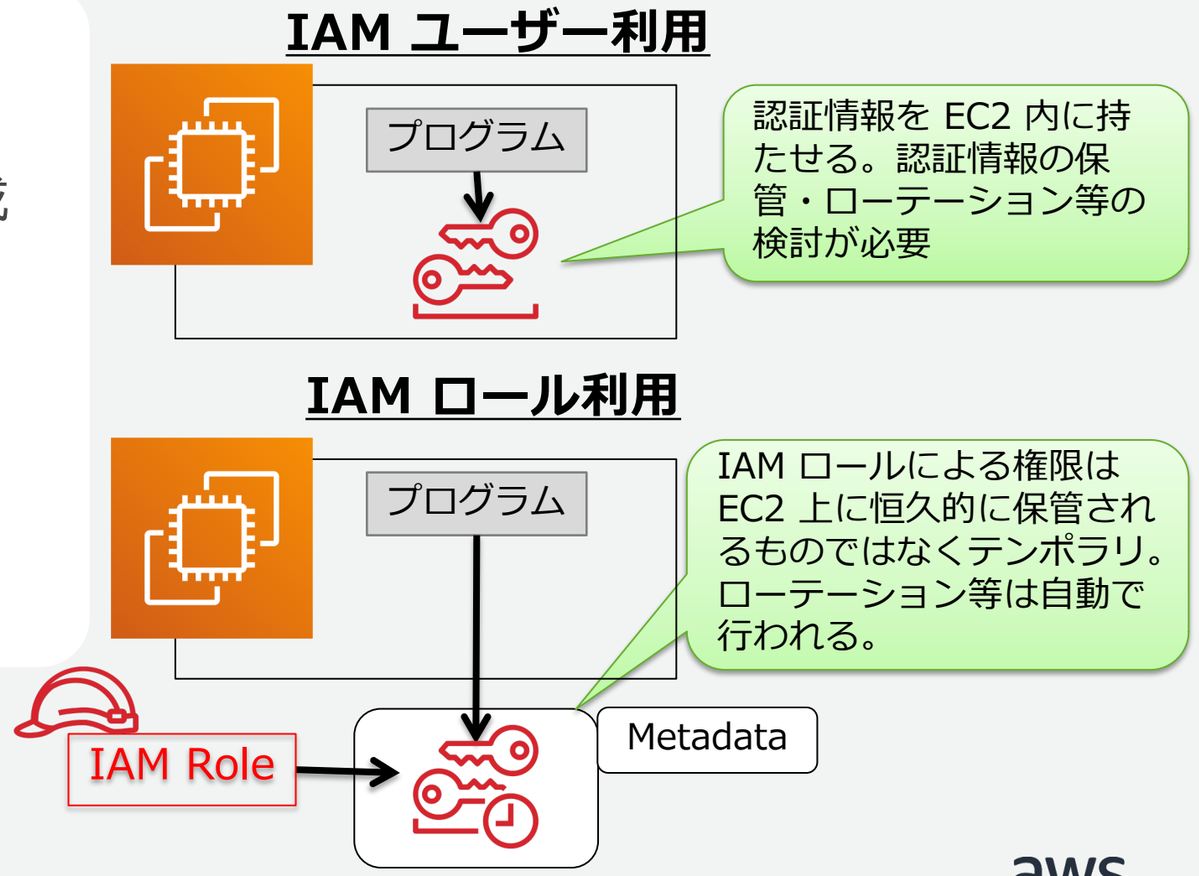
aws

EC2 には IAM ロールを利用

EC2 のような AWS サービスに対して AWS 操作権限を付与するための仕組み。
IAM ユーザーの認証情報のようなものを OS/アプリケーション側に持たせる必要がなく、認証情報の漏えいリスクを低減可能。IAM ロールによる認証情報は AWS が自動的にローテーション。

IAM ロール利用の利点

- EC2 上のアクセスキーの管理が容易
- 認証情報は STS(Security Token Service) で生成
- 自動的に認証情報のローテーションが行われる
- EC2 上のアプリケーションに最低権限を与えることに適している
- AWS SDK 及び AWS CLI のサポート
- IAM ユーザーの認証情報を外部に漏えいしてしまうリスクを低減させる



http://docs.aws.amazon.com/ja_jp/IAM/latest/UserGuide/roles-resourcebasedpolicies-compare.html

IAM ポリシーは最小権限を付与

AWS アカウントのセキュリティを向上させるに、**IAM ポリシーを定期的に確認**し、モニタリングを実施

「アクセスアドバイザー」を活用し、ポリシーにおいて、必要なアクションにのみ、必要な最小限の権限が付与されていることを確認

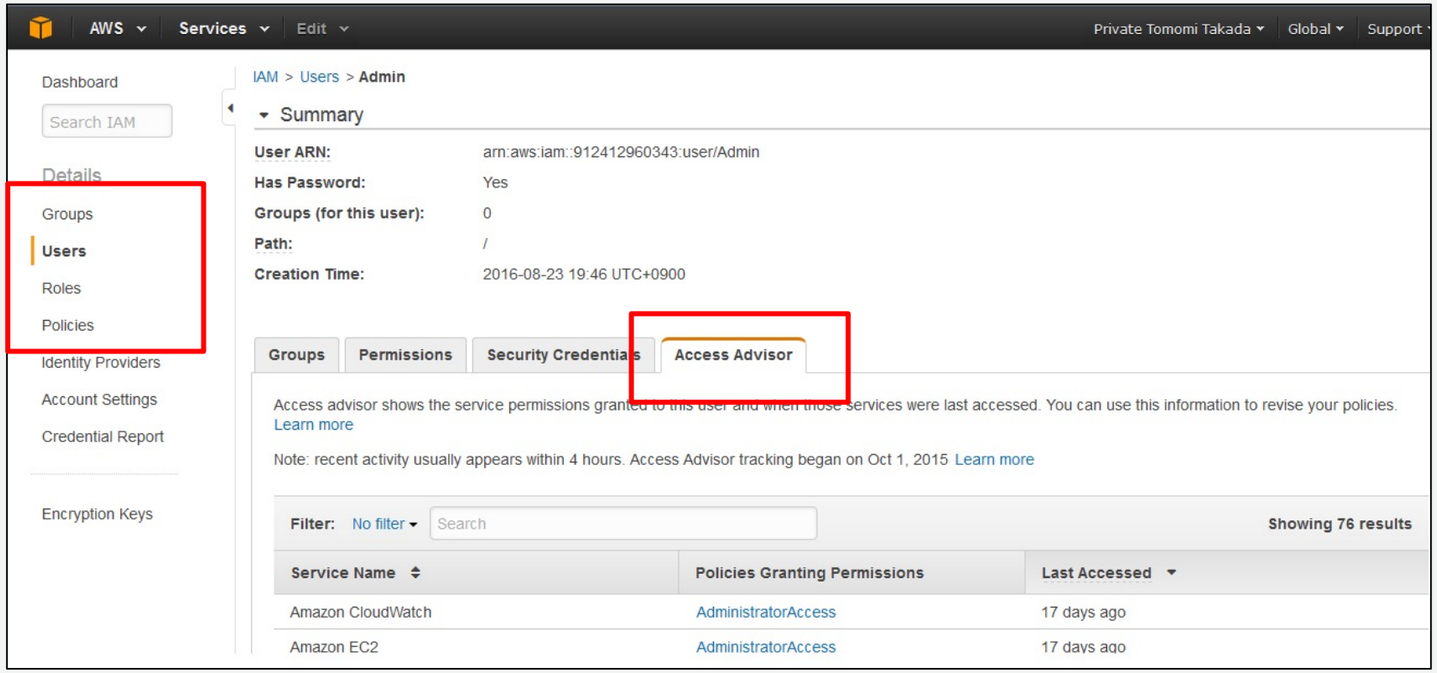
- 「ポリシー」の「ポリシーの使用状況」を確認し、適用されているユーザやグループ、ロールを確認
- 「アクセス権限」で、最小権限かを確認

Access Advisor と
Service Last Accessed Data

IAM エンティティ (ユーザー、グループ、ロール) が、
最後に AWS サービスにアクセスした日付と時刻を表示する機能

IAM の**最小限の特権**に関する設定に利用

- IAM ポリシー内で未使用または最近使用されていないアクセス許可を識別
- 未使用のサービスに関するアクセス許可を削除したり、類似の使用パターンを持つユーザーをグループに再編成
- アカウントのセキュリティを改善



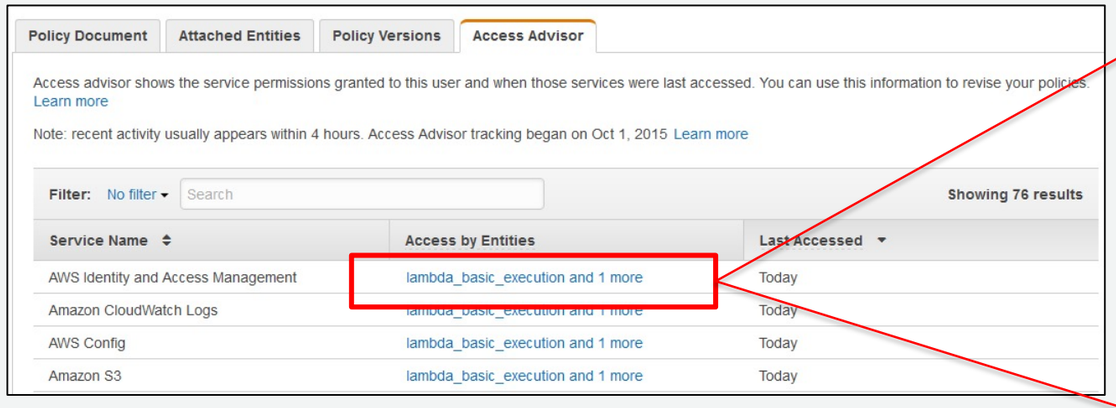
https://docs.aws.amazon.com/ja_jp/IAM/latest/UserGuide/access_policies_access-advisor.html

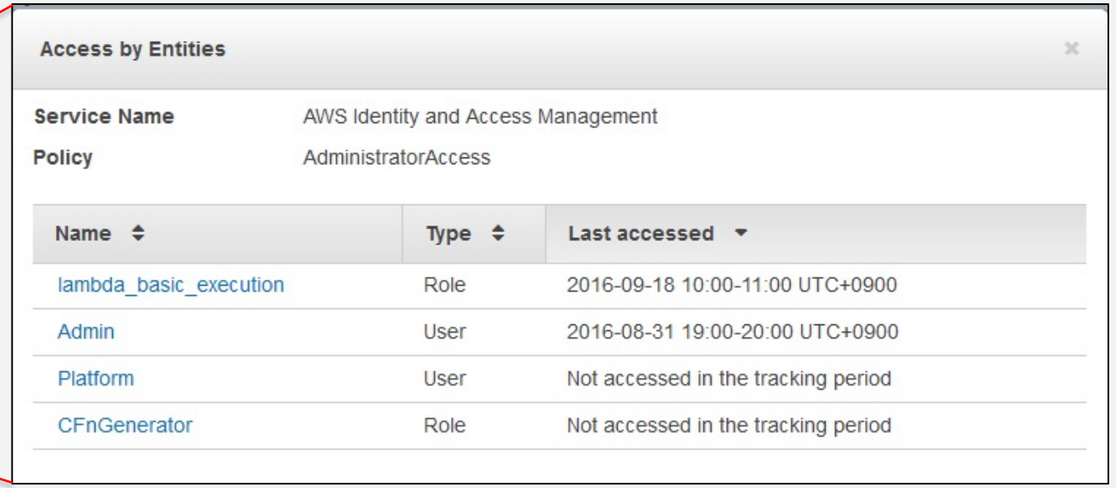
Service Last Accessed Data の利用例

ユーザーや、グループ、ロールに与えられた権限で利用されていないものを発見

AWS Directory Service	AdministratorAccess	19 days ago
Elastic Load Balancing	AdministratorAccess	19 days ago
AWS Key Management Service	AdministratorAccess	19 days ago
AWS Billing	AdministratorAccess	24 days ago
AWS IoT	AdministratorAccess	Not accessed in the tracking period
Amazon API Gateway	AdministratorAccess	Not accessed in the tracking period
AWS Certificate Manager	AdministratorAccess	Not accessed in the tracking period
AWS Database Migration Service	AdministratorAccess	Not accessed in the tracking period
Amazon Storage Gateway	AdministratorAccess	Not accessed in the tracking period

- IAMポリシーの利用状況と利用しているエンティティの識別



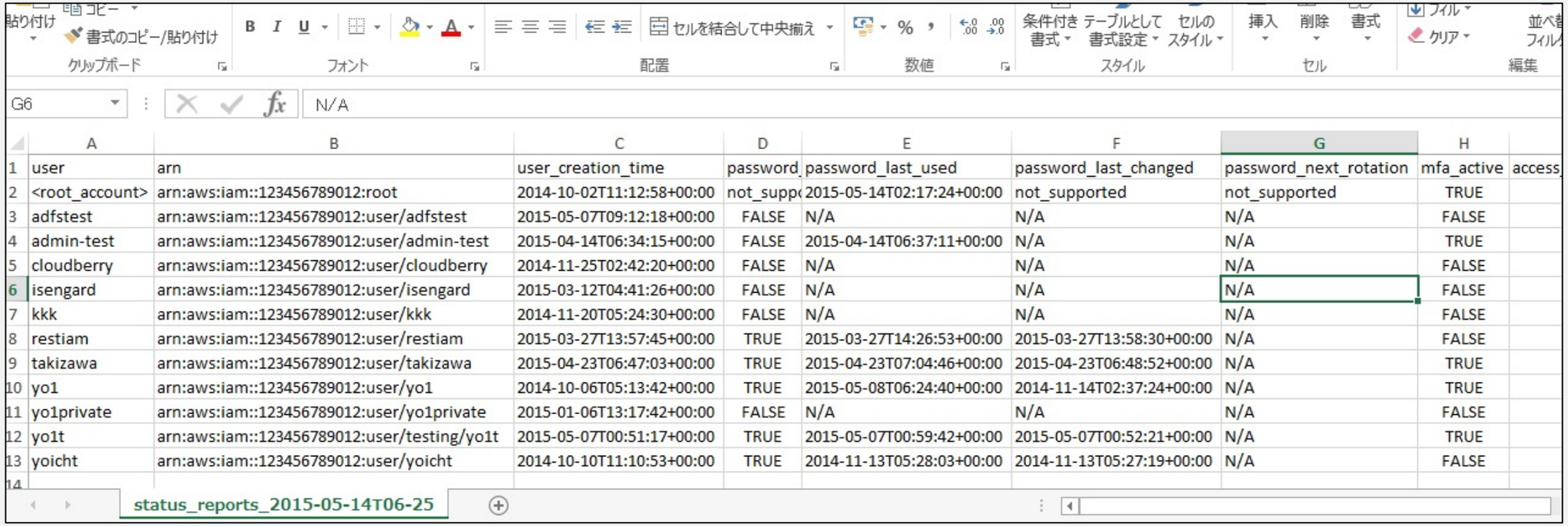


IAMポリシーを利用しているのが誰で最後にアクセスしたのがいつか容易に識別可能

IAM 認証情報レポート

ユーザーの作成日時
最後にパスワードが使われた日時
最後にパスワードが変更された日時
MFA を利用しているか
Access Key が Active か

- Access Key のローテートした日時
- Access Key を最後に使用した日時
- Access Key を最後に利用した AWS サービス
- 証明書は Active か
- 証明書のローテートした日時



G6 | N/A

	A	B	C	D	E	F	G	H	
1	user	arn	user_creation_time	password	password_last_used	password_last_changed	password_next_rotation	mfa_active	access
2	<root_account>	arn:aws:iam::123456789012:root	2014-10-02T11:12:58+00:00	not_supp	2015-05-14T02:17:24+00:00	not_supported	not_supported	TRUE	
3	adfstest	arn:aws:iam::123456789012:user/adfstest	2015-05-07T09:12:18+00:00	FALSE	N/A	N/A	N/A	FALSE	
4	admin-test	arn:aws:iam::123456789012:user/admin-test	2015-04-14T06:34:15+00:00	FALSE	2015-04-14T06:37:11+00:00	N/A	N/A	TRUE	
5	cloudberry	arn:aws:iam::123456789012:user/cloudberry	2014-11-25T02:42:20+00:00	FALSE	N/A	N/A	N/A	FALSE	
6	isengard	arn:aws:iam::123456789012:user/isengard	2015-03-12T04:41:26+00:00	FALSE	N/A	N/A	N/A	FALSE	
7	kkk	arn:aws:iam::123456789012:user/kkk	2014-11-20T05:24:30+00:00	FALSE	N/A	N/A	N/A	FALSE	
8	restiam	arn:aws:iam::123456789012:user/restiam	2015-03-27T13:57:45+00:00	TRUE	2015-03-27T14:26:53+00:00	2015-03-27T13:58:30+00:00	N/A	FALSE	
9	takizawa	arn:aws:iam::123456789012:user/takizawa	2015-04-23T06:47:03+00:00	TRUE	2015-04-23T07:04:46+00:00	2015-04-23T06:48:52+00:00	N/A	TRUE	
10	yo1	arn:aws:iam::123456789012:user/yo1	2014-10-06T05:13:42+00:00	TRUE	2015-05-08T06:24:40+00:00	2014-11-14T02:37:24+00:00	N/A	TRUE	
11	yo1private	arn:aws:iam::123456789012:user/yo1private	2015-01-06T13:17:42+00:00	FALSE	N/A	N/A	N/A	FALSE	
12	yo1t	arn:aws:iam::123456789012:user/testing/yo1t	2015-05-07T00:51:17+00:00	TRUE	2015-05-07T00:59:42+00:00	2015-05-07T00:52:21+00:00	N/A	TRUE	
13	yoicht	arn:aws:iam::123456789012:user/yoicht	2014-10-10T11:10:53+00:00	TRUE	2014-11-13T05:28:03+00:00	2014-11-13T05:27:19+00:00	N/A	FALSE	
14									

status_reports_2015-05-14T06-25

git-secrets を使ったクレデンシャル管理

- コミットの防止

```
$ git secrets --register-aws
$ git add git-secret.py
$ git commit -m "This is a test commit for git-secret"
git-secret.py:1:AWSAccessKeyId = "AKIAIOSFODNN1EXAMPLE"
git-secret.py:2:AWSSecretKey = "wJalrXUtnFEMI/K1MDENG/bPxRfiCYEXAMPLEKEY"
```

[ERROR] Matched one or more prohibited patterns

- 既存レポジトリのスキャン

```
$ git secrets --scan
```

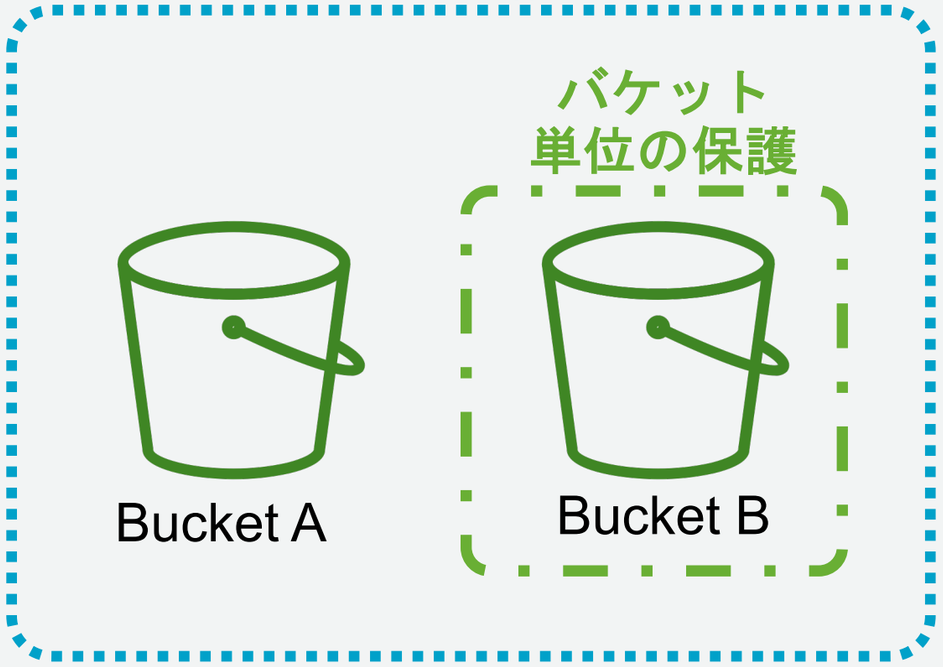

https://github.com/awslabs/git-secrets

意図せずバケットがパブリックになるのを防ぐ

Amazon S3 Block Public Access

アカウントレベルまたはバケットレベルで「あらかじめ」
意図せずバケットがパブリックアクセスになるのを抑制する

アカウント単位の保護



パブリックなアクセスを許すバケットポリシー

```
{
   "Version": "2012-10-17",
   "Id": "Policy153130929
   "Statement": [
      {
         "Sid": "Allow      bje      ny",
         "Effect": "All
         "Principal": "
         "Action": "s3:      bject",
         "Resource": "a      s3:::sum      et/*”
      }
   ]
}
```

新規で作成されるアカウントと新規で作成されるバケットにデフォルトで適用される

aws

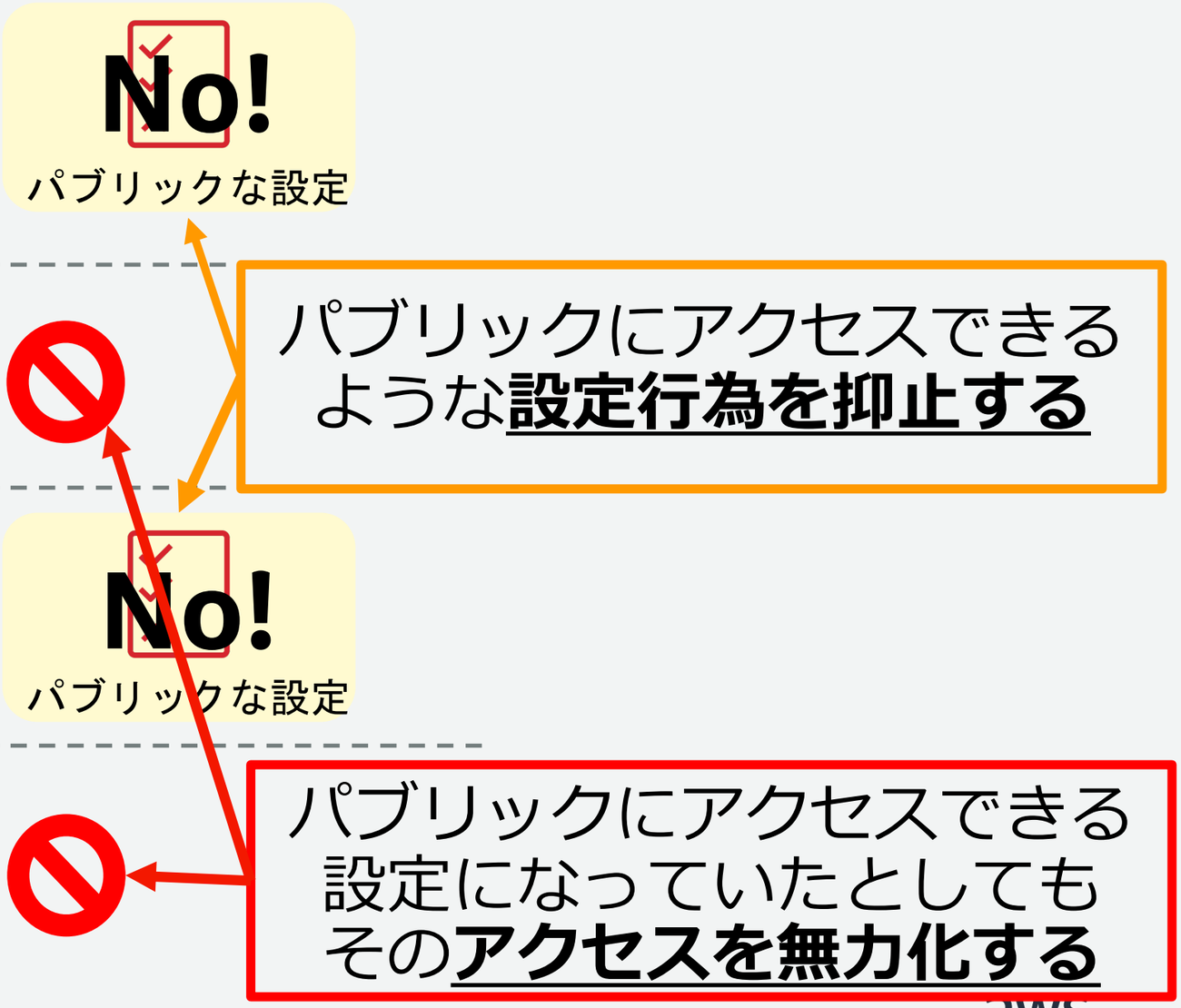
Amazon S3 Block Public Access（続き）

1. 新しいパブリック **ACL** および
   パブリックオブジェクトのアップロード
   をブロック

2. パブリック **ACL** により付与された
   パブリックアクセスを無効化

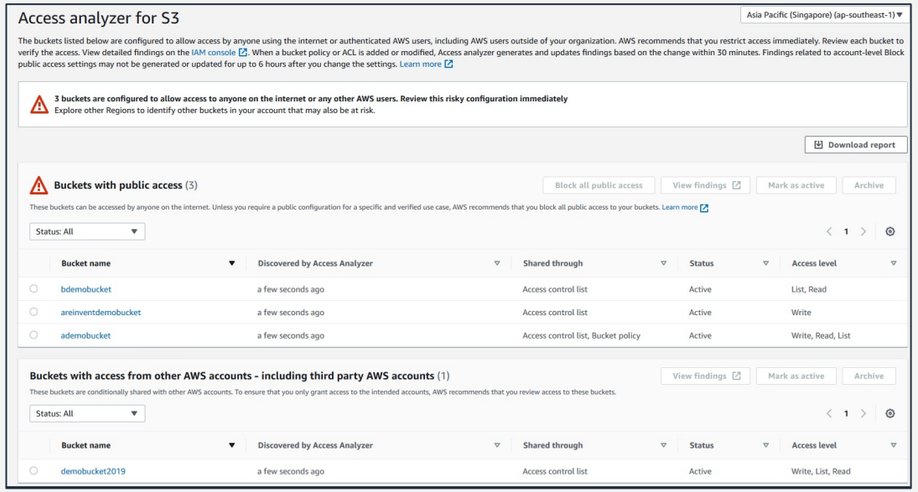
3. 新しいパブリック**バケットポリシー**を
   ブロック

4. パブリックの**バケットポリシー**を持つ
   バケットへのパブリックアクセスおよび
   クロスアカウントアクセスを無効化



Access Analyzer for Amazon S3

- アクセスポリシーを解析し、その内容が意図したものであることを確認するための機能
- IAM Access Analyzerをベースに実装されており、有効にするとAccess Analyzer for S3が自動的にコンソールに表示される
- パブリックアクセス可能なバケットと、他のアカウントに共有されたバケットが一覧表示され、後者はアクセス権も確認可能
- ステータスがActiveなものは要注意。不要な権限が付与されていないかを確認し、対処する必要がある。妥当なものであれば問題なしというフラグを付与すればOK
- 中国を除く全リージョンで利用可能



暗号化を強制したい

データの暗号化で守れる範囲

	想定シナリオ	暗号化の効果
紛失・盗難	• データを入れたUSBメモリの紛失 • HDDの盗難	暗号化の効果：⭕
内部犯行	• 社内サーバのディスク持ち出し • 社内データの外部送信	暗号化の効果：🔺 復号済みのデータを外部に送信されると漏洩する。
オペレーションミス	• 内部情報が入ったコードを外部公開 • 設定の不備でファイルの外部公開 • メールの誤送信	暗号化の効果：🔺 復号済みのデータを外部に送信されると漏洩する。
サイバー攻撃	• 社内ネットワークに侵入されてサーバ内のデータを外部にアップロード	暗号化の効果：🔺 復号済みのデータを外部に送信されると漏洩する。

aws

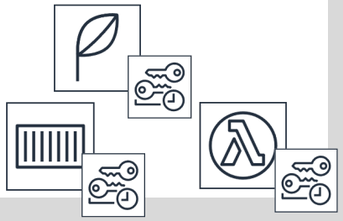
暗号化はもう一つのアクセス管理メカニズム

アクセス主体

ユーザー
(またはグループ)

ロール
(assumed, delegated, federated, service-linked)

アプリケーション



認証

Authentication

認可

Authorization

IDベースポリシー/リソースベースポリシー

AWS Identity and Access Management (IAM)

リソース

Instances

Endpoints

Bucket with objects

復号

Decryption

キーポリシー

AWS Key Management Service

データ

aws

AWS Key Management Service (AWS KMS)

暗号鍵の作成、管理、運用のためのサービス

- 可用性、物理的セキュリティ、ハードウェアの管理をAWS が担当するマネージドサービス
- 暗号鍵を保存、暗号鍵を使用するための安全なロケーションを提供
- マスターキーはFIPS 140-2検定済暗号化モジュールによって保護

AWSサービスとの統合 (S3, EBS, Redshift, RDS, Snowball等)

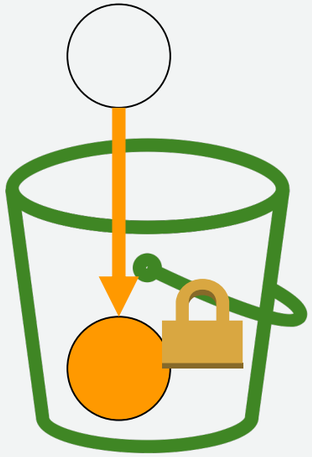
SDKとの連携でお客様の独自アプリケーションデータも暗号化

AWS CloudTrailと連動したログの生成による組み込み型監査

全てのリージョンで利用可能

aws

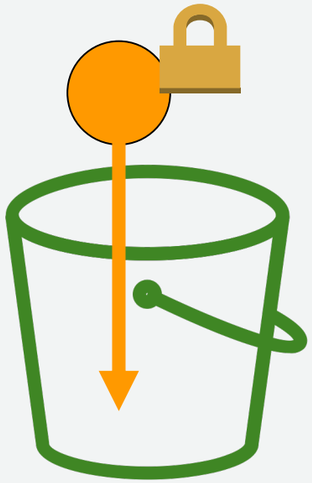
Amazon S3 における暗号化(at rest)



サーバーサイド暗号化

- SSE-S3　：Amazon S3 が管理する鍵を利用した暗号化
- SSE-KMS　：AWS KMS でユーザが管理する鍵を利用した暗号化
- SSE-C　：ユーザが独自に鍵管理を行う

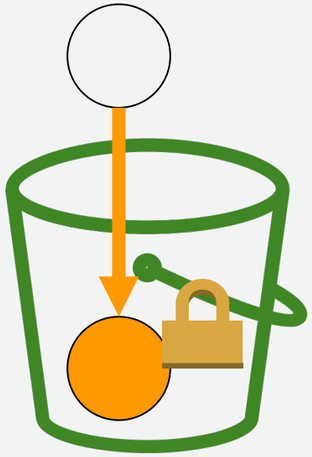
AWS Key Management Service



クライアントサイド暗号化

クライアント側で暗号化したデータをS3に格納する

Amazon S3 における暗号化(at rest)

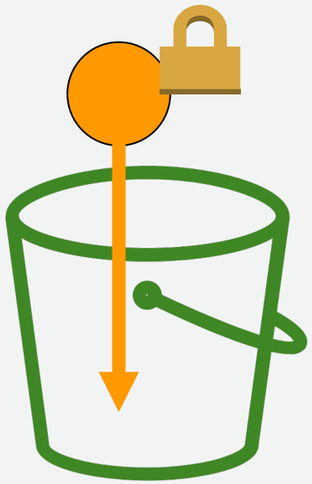


デフォルト暗号化の強制が可能

サーバーサイド暗号化

- SSE-S3　：Amazon S3 が管理する鍵を利用した暗号化
- SSE-KMS　：AWS KMS でユーザが管理する鍵を利用した暗号化
- SSE-C　：ユーザが独自に鍵管理を行う

デフォルト暗号化を適用し、Block Public Accessを利用することで
意図しないパブリックアクセスを防ぐ、二重のガードレールとなる。



クライアントサイド暗号化

クライアント側で暗号化したデータをS3に格納する

aws

想定される課題と対策

課題	対策
偶発的な情報の開示	IAMによる適切な権限管理 VPCネットワークの理解 Block Public Access 暗号化
データの不整合	IAMによる適切な権限管理 データの整合性チェック **バージョニング / Object Lock**
過失による削除	**バージョニング/ Object Lock** MFA Delete
証跡の取得 ユーザー行動の変化を観察	AWS CloudTrail によるログ取得 Amazon Macie の利用

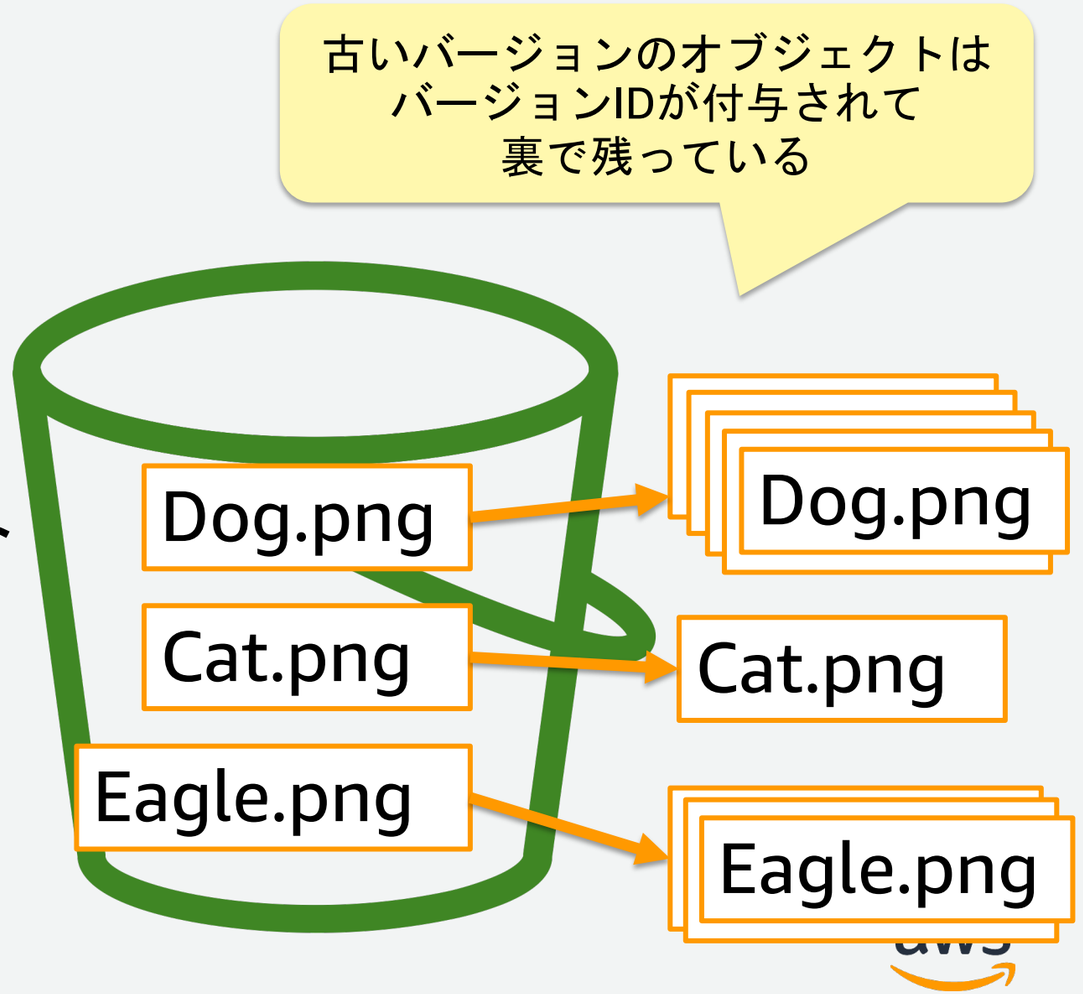
aws

改ざん・過失による削除を防止したい

バージョン管理機能 (Versioning)

ユーザやアプリケーションの誤操作による削除対策に有効

- バケットに対して設定(Enable/Disable)
- キー名ごとにバージョンを管理する
- バージョン保管されている任意のオブジェクトを参照可能
- バージョン保管されているオブジェクトも課金が発生する
- バケットを削除したい場合は、古いバージョンのオブジェクトも削除する



Amazon S3 Object Lock (改ざん防止機能)

S3 オブジェクトを変更できないようにする

いわゆる WORM 機能(Write Once Read Many)
オブジェクトまたはバケットに対して適用

リテンション指定

改ざんを防止するロックの期間の定義
リーガルホールド

データ保護とコンプライアンス

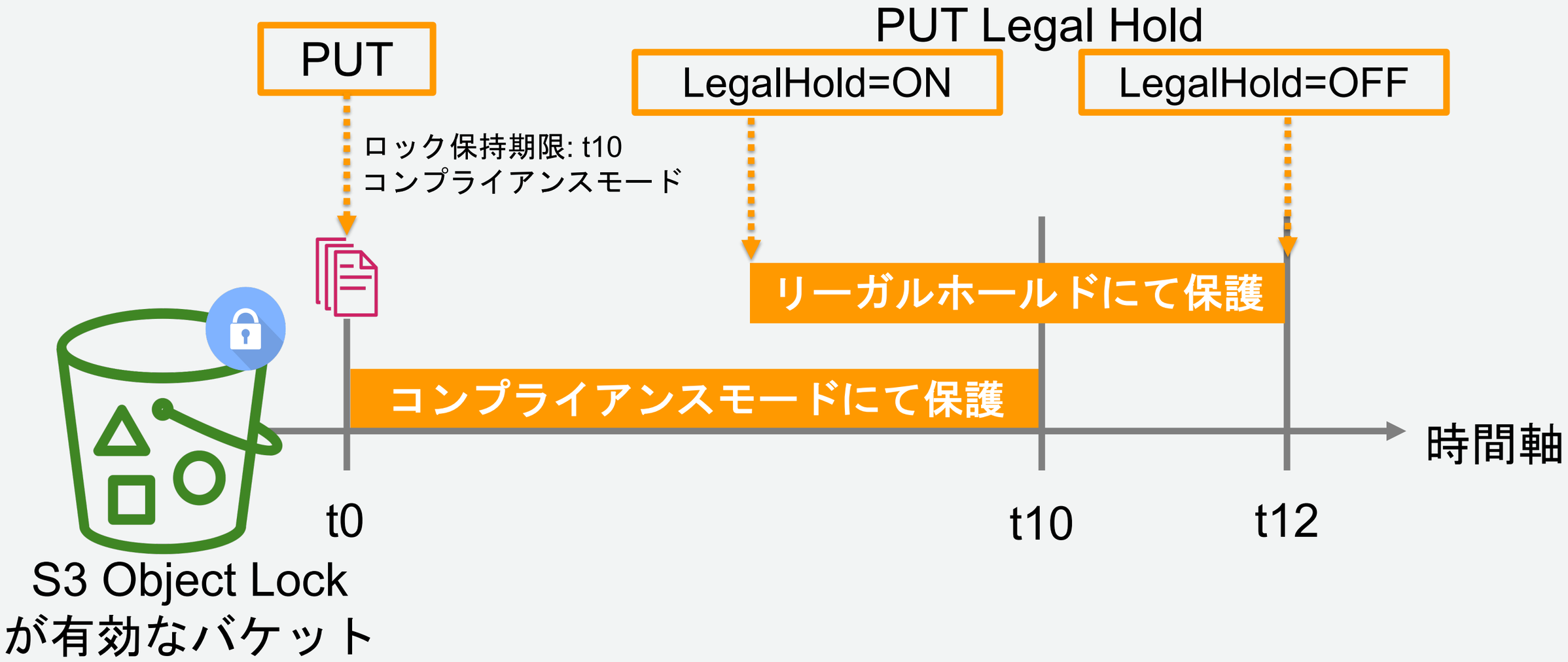
第三者機関によるSEC 17a-4アセスメント済み
意図しない削除からの保護

二つのモード

コンプライアンスモードとガバナンスモード

aws

Amazon S3 Object Lock 利用例

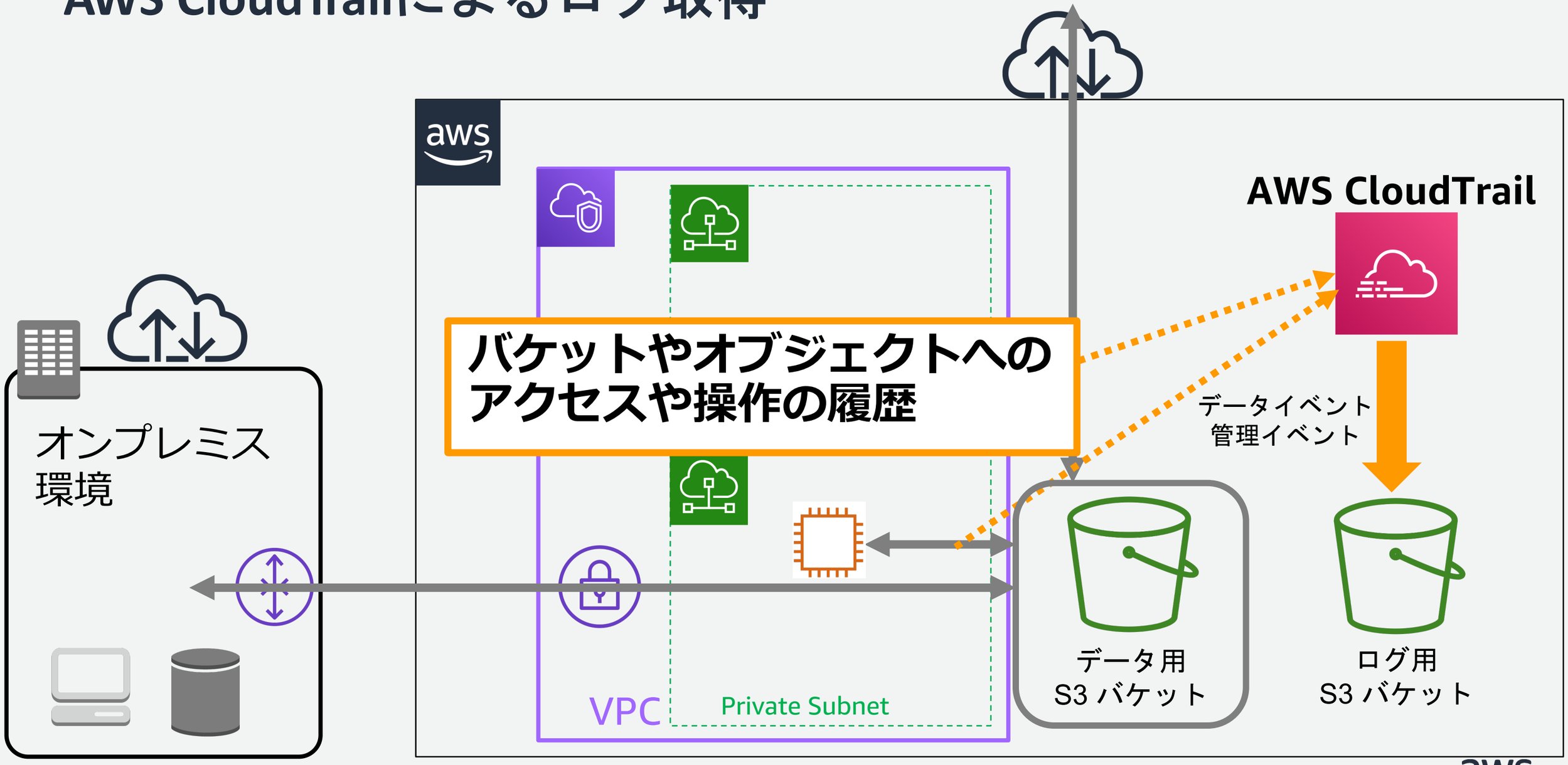


想定される課題と対策

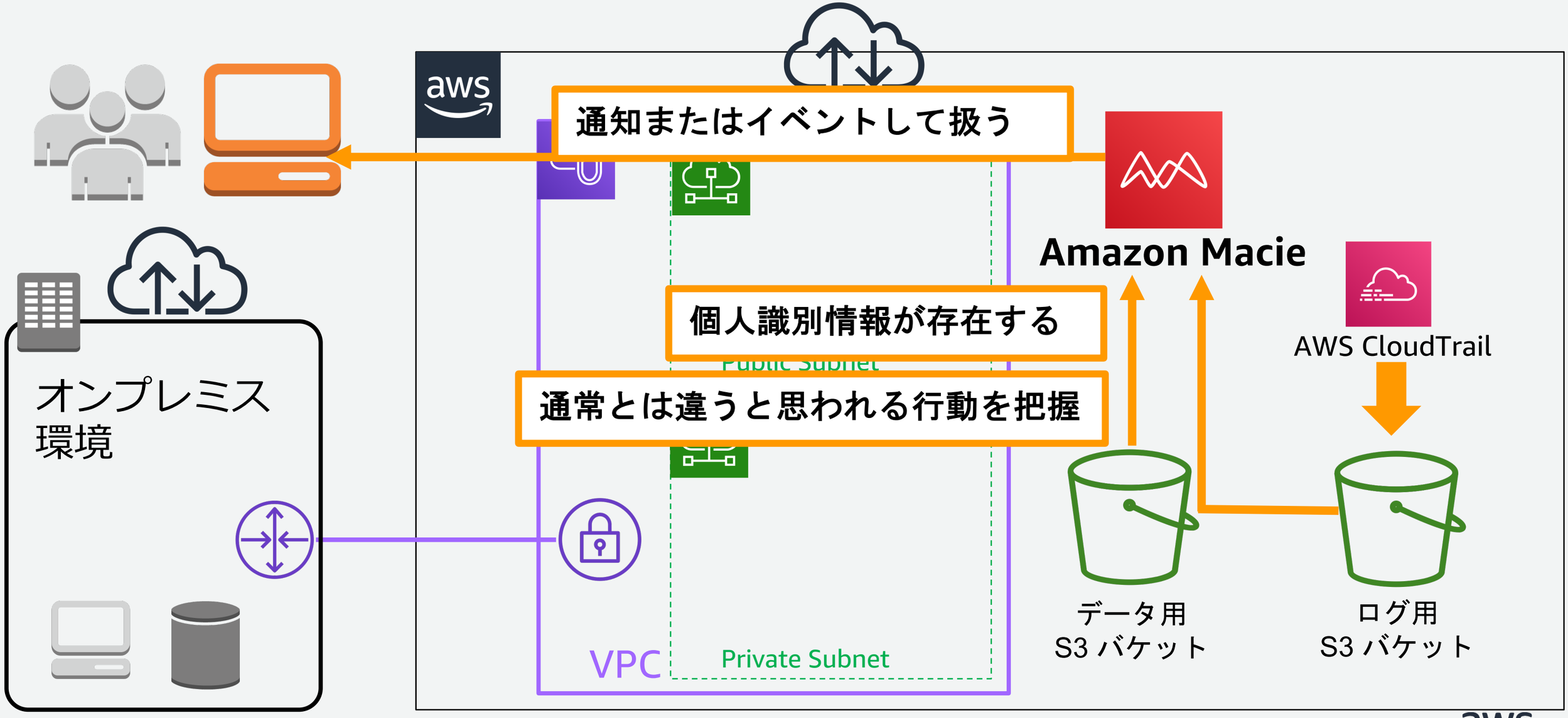
課題	対策
偶発的な情報の開示	IAMによる適切な権限管理 VPCネットワークの理解 Block Public Access 暗号化
データの不整合	IAMによる適切な権限管理 データの整合性チェック バージョニング / Object Lock
過失による削除	バージョニング/ Object Lock MFA Delete
証跡の取得 ユーザー行動の変化を観察	**AWS CloudTrail によるログ取得** **Amazon Macie の利用**

aws

AWS CloudTrailによるログ取得



Amazon Macieによるユーザ行動の追跡



その他のモニタリングや管理に有効な機能

Logging

- バケット単位でバケットに対するアクセスログの出力設定が可能
- 出力先としてS3バケットを指定
- ログフォーマットは下記を参照
  http://docs.aws.amazon.com/ja_jp/AmazonS3/latest/dev/LogFormat.html

Tag管理

- バケット、オブジェクトに対してタグの指定が可能
- タグ指定によりリソースグループにて関連するAWSサービスとの紐付けが可能
- オブジェクトに対してのタグは、ライフサイクル、分析、モニタリング、
  クロスリージョンレプリケーション機能で活用可能

aws

まとめ

まとめ

- Amazon S3 にデータが格納されるシーン
  - Amazon S3 振り返り
  - ユースケースの分類
    - データ保護・移行、データレイク、Web サービスのコンテンツオフロード

- Amazon S3 をセキュアに活用するにあたっての基本要素
  - AWS のセキュリティ方針
  - 責任共有モデル
  - AWS Well-Architected Framework

- Amazon S3 のセキュリティとそのベストプラクティス
  - IAM による適切な権限管理
  - Block Public Access
  - 暗号化
  - バージョニング／Object Lock
  - AWS CloudTrail によるログ取得
  - Amazon Macie の利用

aws

想定される課題と対策 => ベストプラクティス

課題	対策
偶発的な情報の開示	**IAMによる適切な権限管理** VPCネットワークの理解 **Block Public Access** **暗号化**
データの不整合	IAMによる適切な権限管理 データの整合性チェック **バージョニング / Object Lock**
過失による削除	**バージョニング/ Object Lock** MFA Delete
証跡の取得 ユーザー行動の変化を観察	**AWS CloudTrail によるログ取得** **Amazon Macie の利用**

※ 太字箇所が本日ご紹介したベストプラクティスです

Thank you!

Appendix

Amazon S3 をセキュアに活用するために抑えておくポイント

ここできっちり整理しておくAWSサービス

AWS Identity and Access Management (IAM)

Amazon Virtual Private Cloud(VPC)

Amazon Simple Storage Service (S3)

aws

ここできっちり整理しておくAWSサービス

IAM

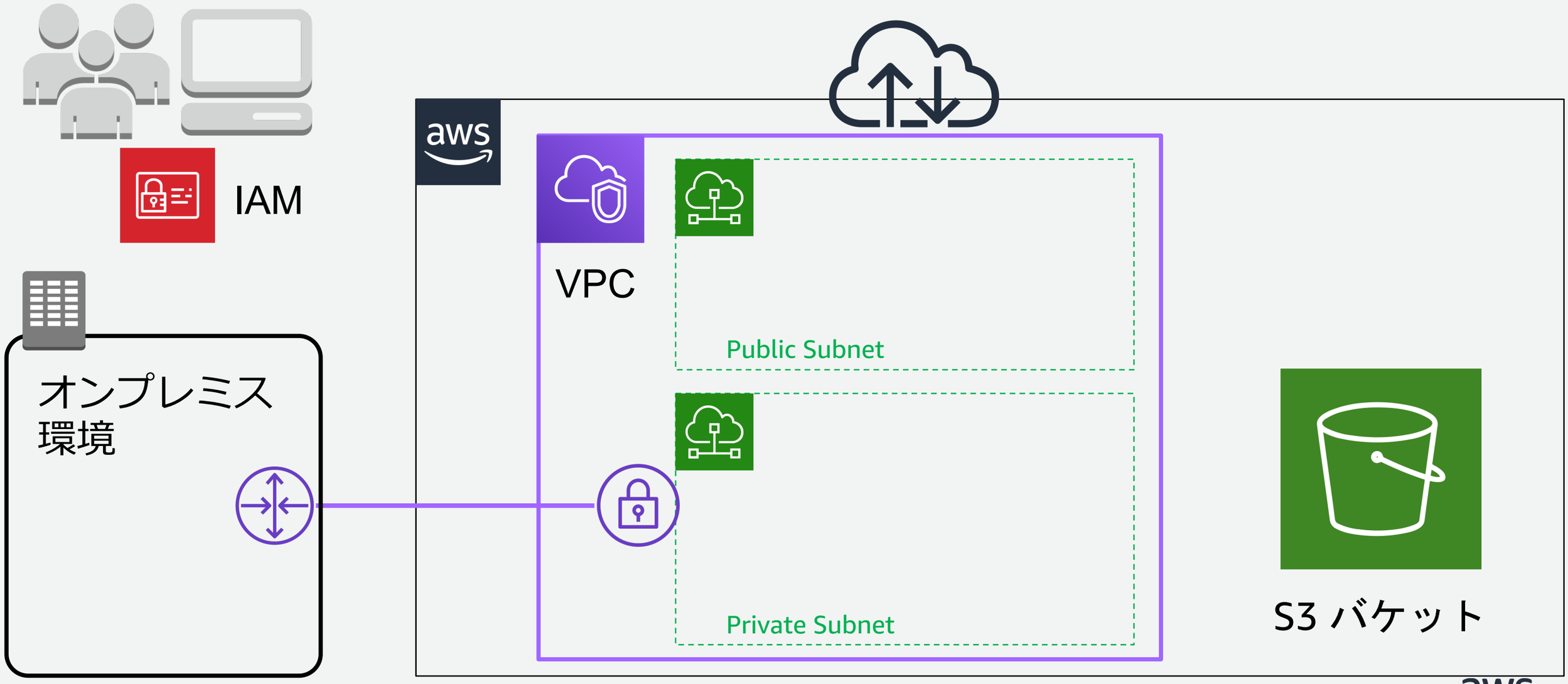
VPC

S3

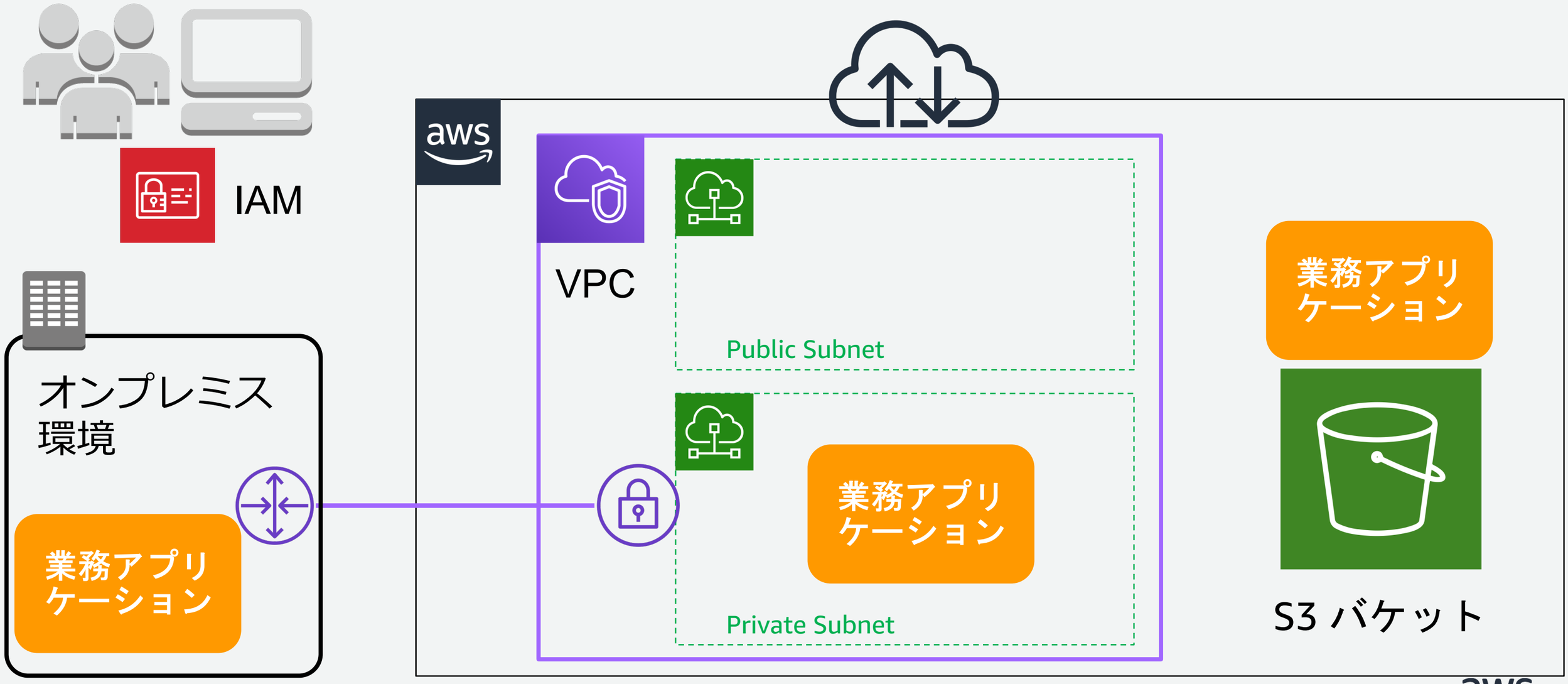
S3 をセキュアに活用するにあたっての基本要素

aws

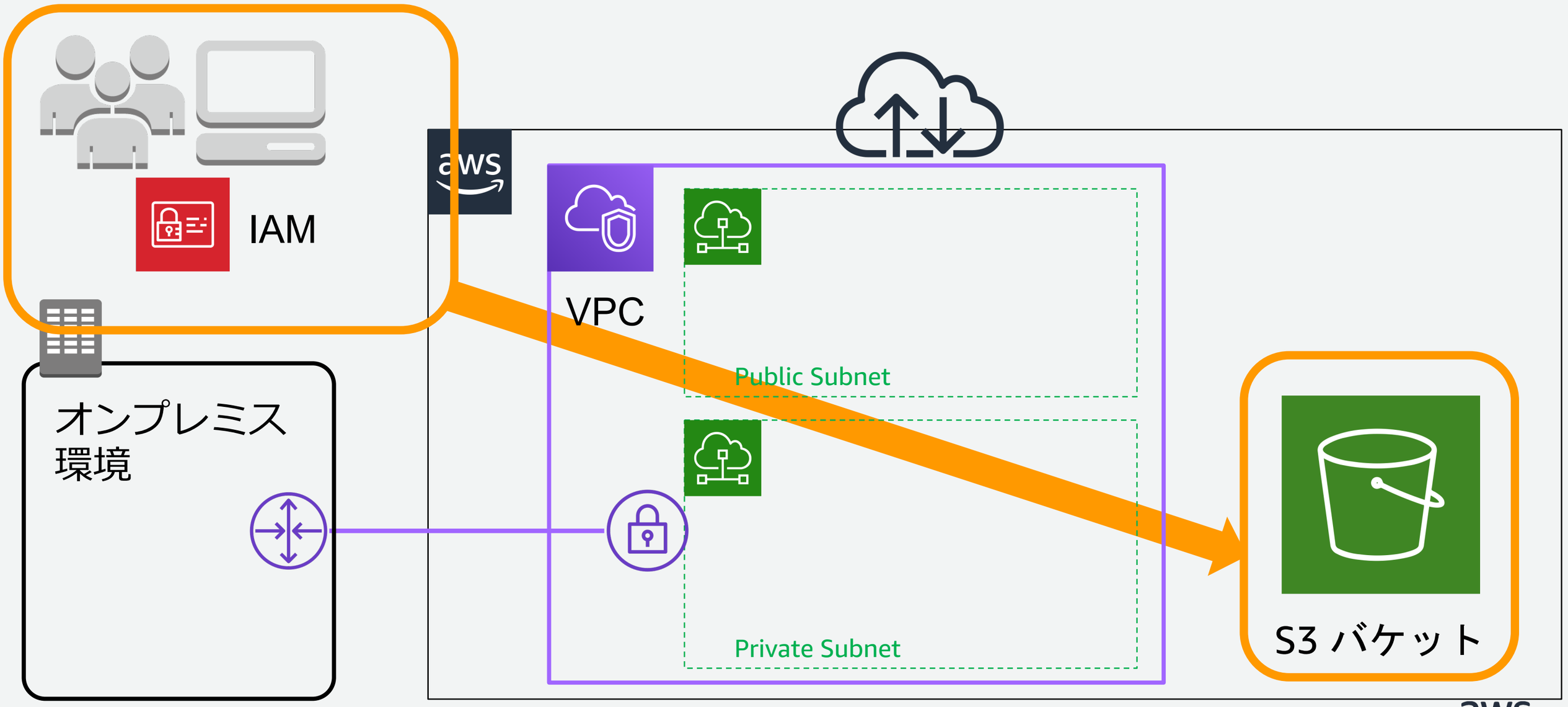
IAM / VPC / S3 の立ち位置の図解



IAM / VPC / S3 の立ち位置の図解 (続き)



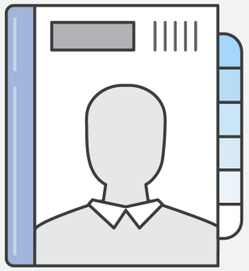
適切なアクセス許可を定義するための整理1(IAM)



IAMの用語

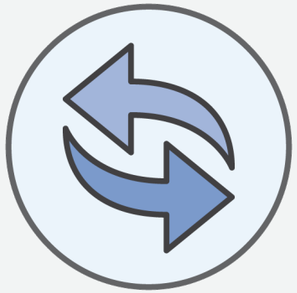
アイデンティティ(Identity)

- プリンシパル(Principal)と呼ばれる実行者のアイデンティティ（ユーザー / グループ / ロール）



API アクション(Action)

- 目的のAPIが何をするもので、どんなレスポンスをするか

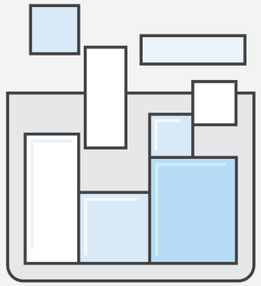


リソース(Resource)

- アクションが行われる対象のAWSリソース

条件(Condition)

- そのアクションを許可するためのアイデンティティ、アクション、リソース、その他リクエストパラメータの要求する特徴



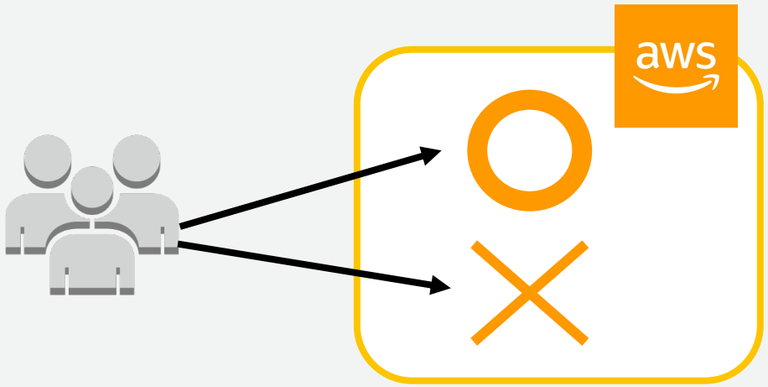
aws

ユーザーポリシーとリソースポリシーの比較

IAM ユーザーポリシー

このユーザーは AWS で
何ができるのか?

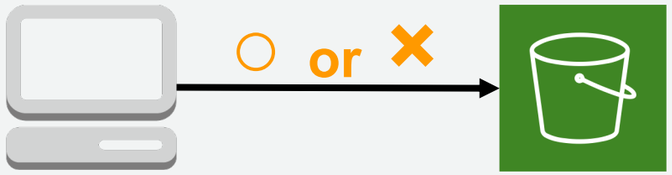
- IAM 環境にアクセスコントロール
ポリシーを保持したい場合
- AWS のサービスすべてに対する
コントロール



Amazon S3 バケットポリシー

このS3リソースに
誰がアクセスできるのか？

- S3 環境にアクセスコントロール
ポリシーを保持したい場合
- IAM ロールを使用せず、S3 バケット
に対するクロスアカウントアクセスも
付与できる



aws

ユーザーポリシーとリソースポリシーの比較

IAM ユーザーポリシー

この特定のユーザーは summitbucket への
オブジェクトの PUT および GET が許可されます

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
              ”Sid":"Allow-write-and-read",
              “Effect": ”Allow",
              "Action":[
                 "s3:PutObject",
                 "s3:GetObject",
              ],
      "Resource":"arn:aws:s3:::summitbucket/*"
    }
  ]
}
```

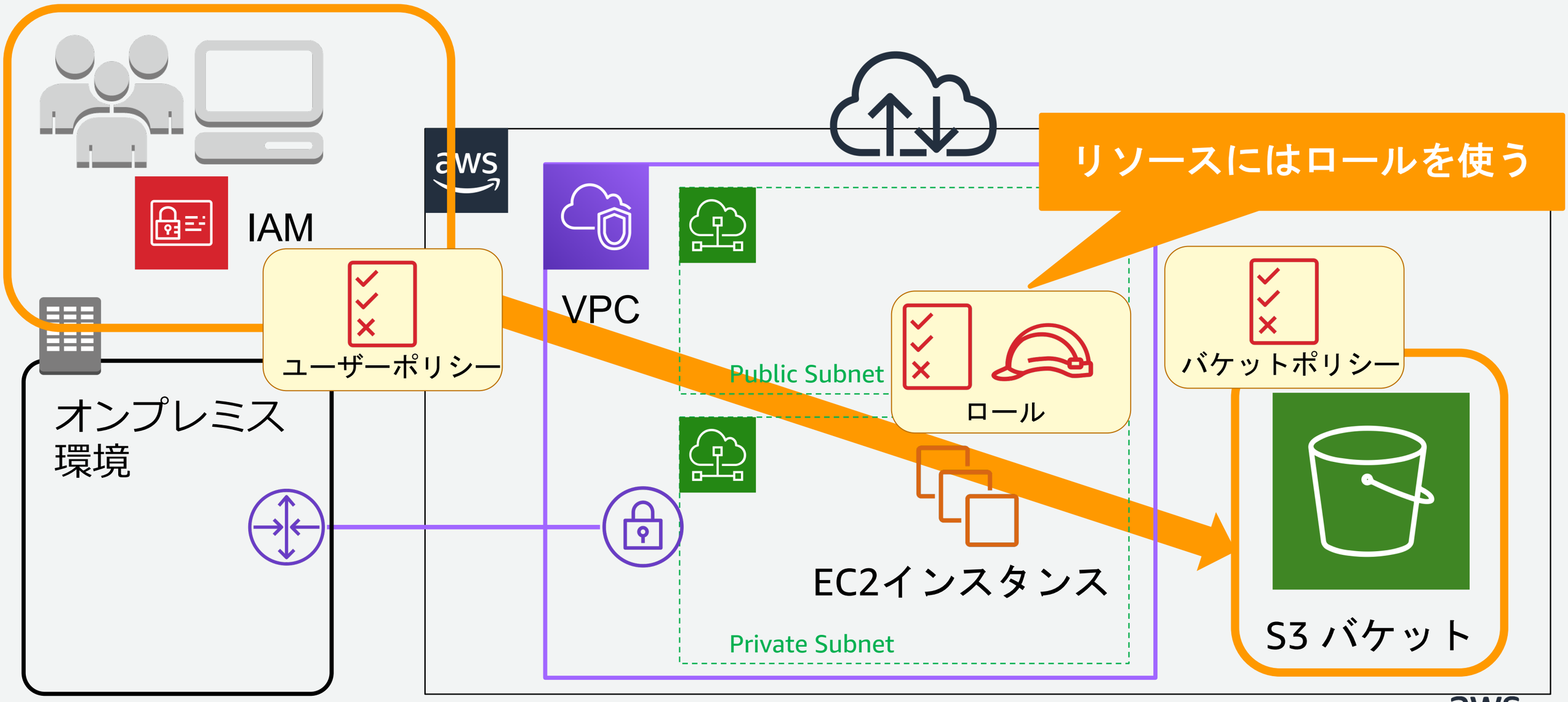
Amazon S3 バケットポリシー

AWS アカウント 1111111111 のプリンシパルは
特定のタグ値を持つsummitbucket のオブジェクトのみ
読み込みを許可されます

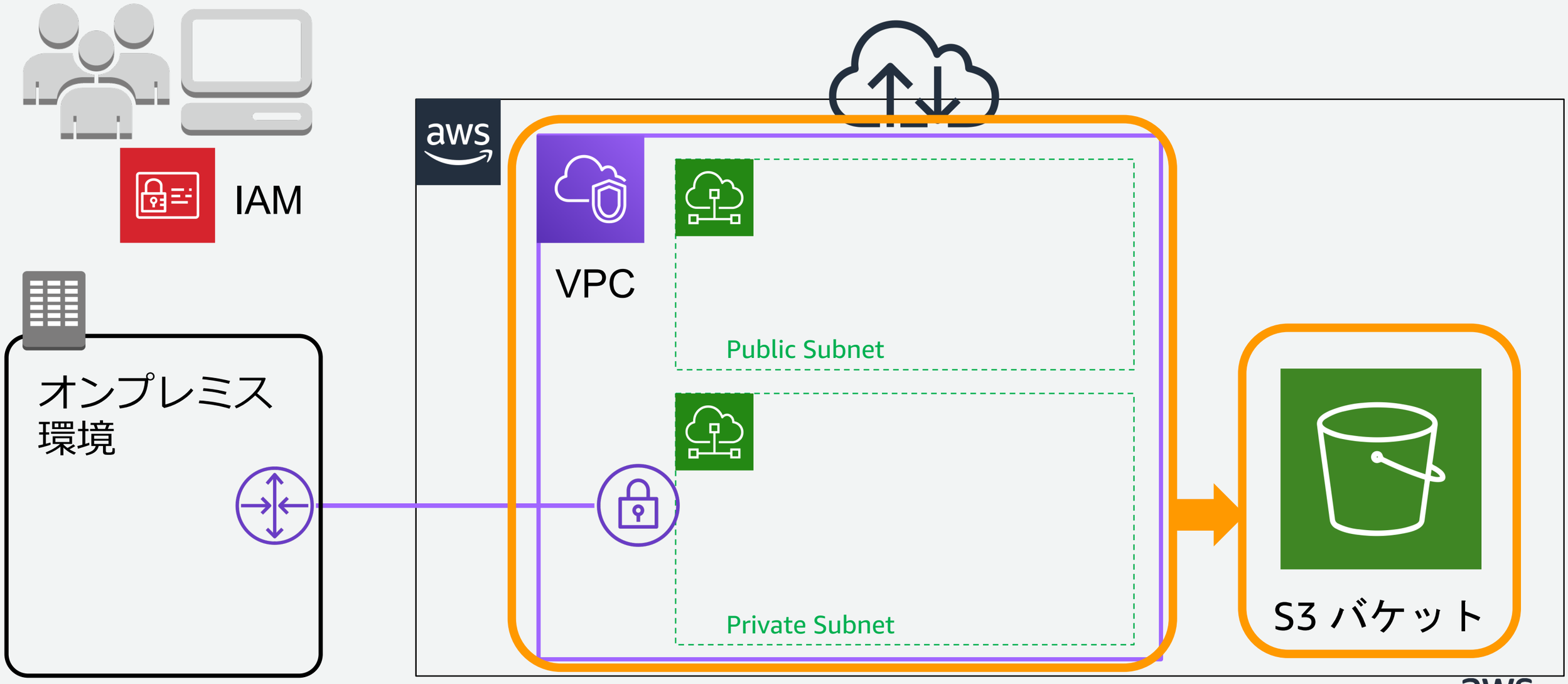
```
{
   "Version": "2012-10-17",
   "Id": "123",
   "Statement": [
{
              "Sid": ”Allowing Read Permission",
              "Effect": "Allow",
              "Principal": {"AWS":"1111111111"},
              "Action": ["s3:GetObject"],
              "Resource": ["arn:aws:s3:::summittbucket/*”],
              "Condition": {"StringEquals":
{"s3:ExistingObjectTag/Project": "X"}}
              }
   ]
}
```


aws

適切なアクセス許可を定義するための整理1(IAM)

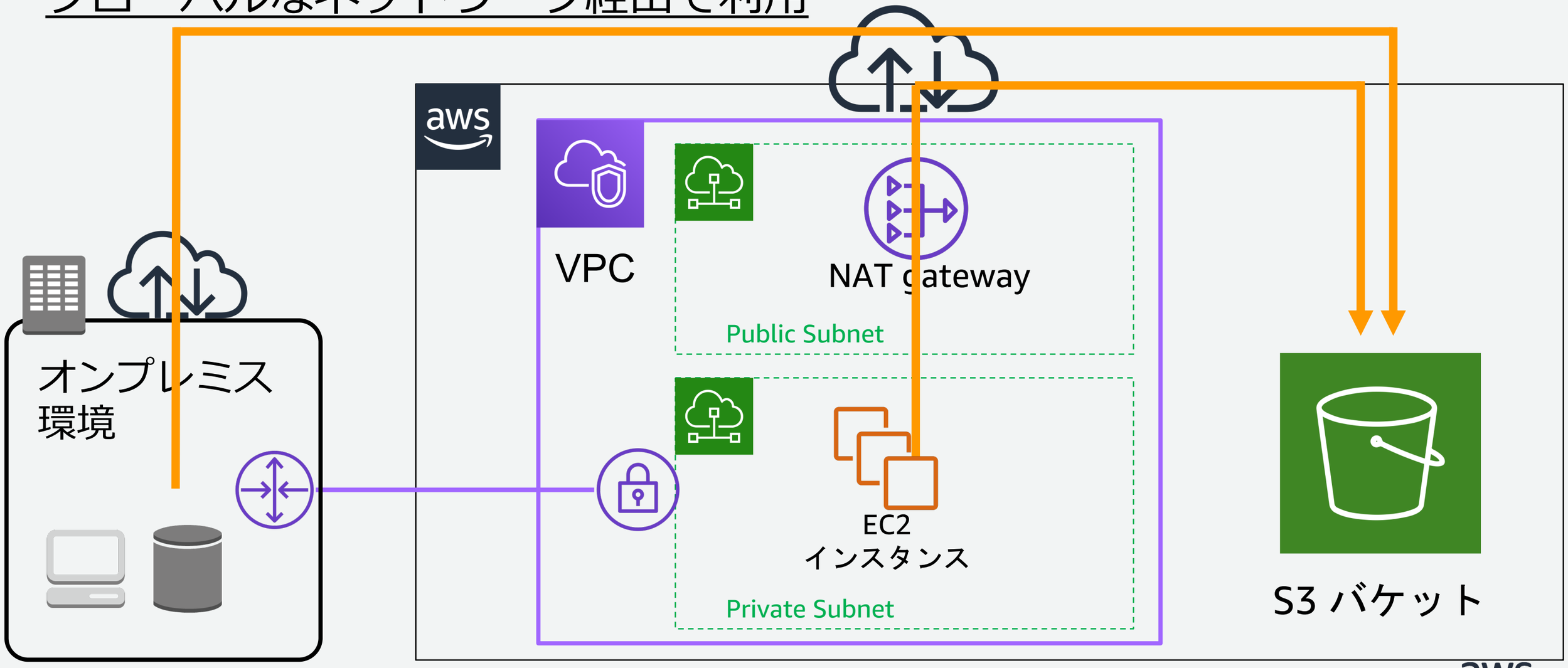


適切なアクセス許可を定義するための整理2(VPC)



Amazon S3へのアクセスパス

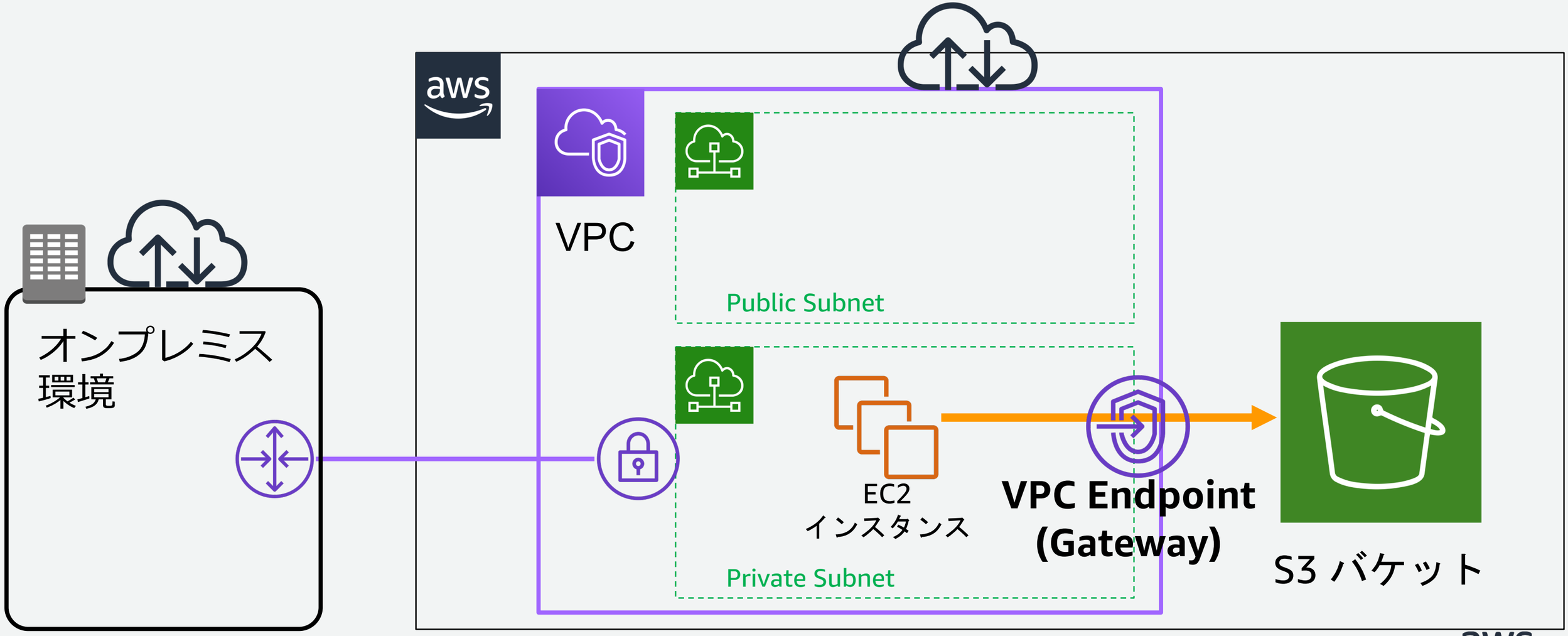
グローバルなネットワーク経由で利用



aws

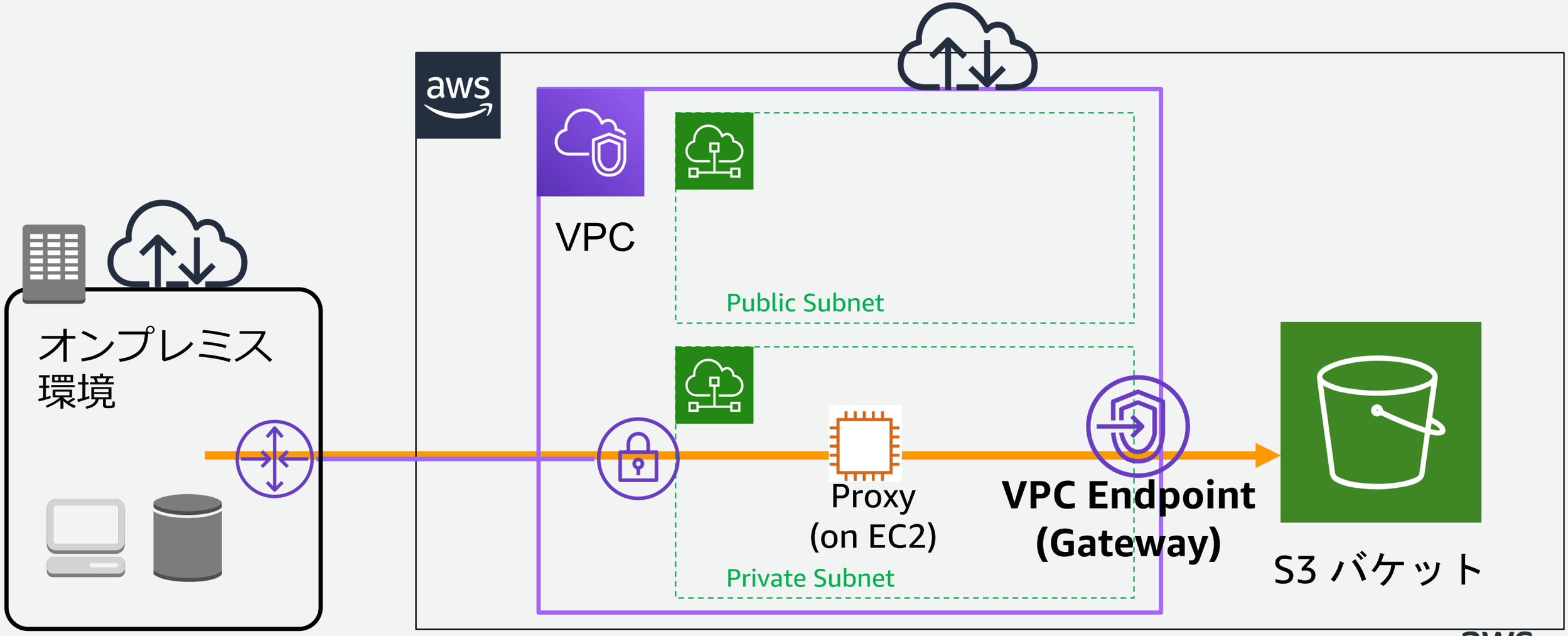
Amazon S3へのアクセスパス(続き)

閉じたネットワーク経由で利用



Amazon S3へのアクセスパス(続き)

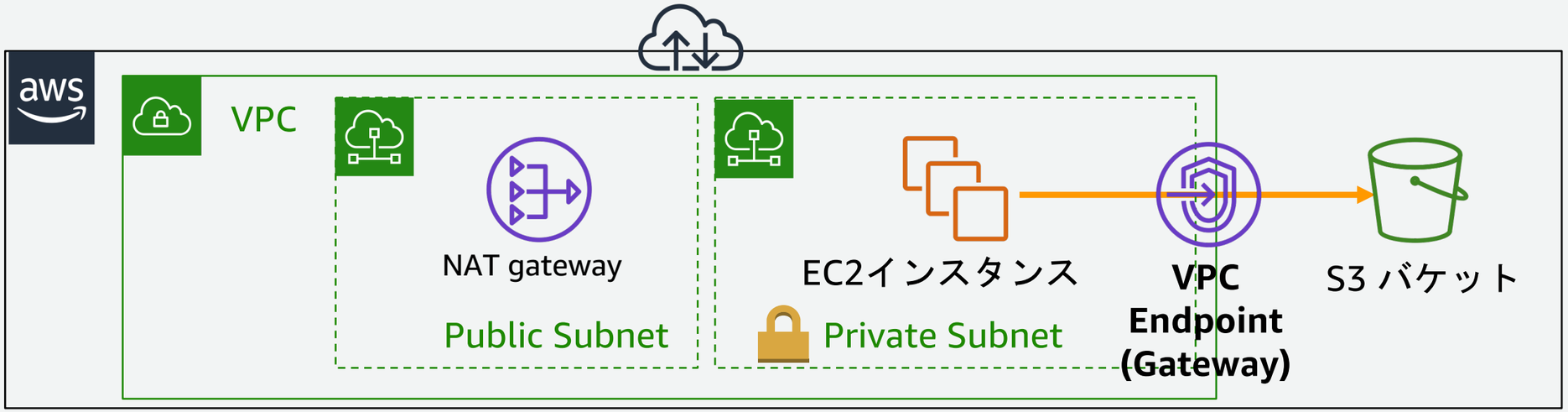
閉じたネットワーク経由で利用



VPC Endpoint

VPC内の**Private Subnet**上で稼働するサービスからNAT Gatewayを経由せずに直接S3とセキュアに通信させることが可能

- 同一リージョンのS3のみ通信可能
- アクセスポリシーを適用して通信可能なBucketや通信元のVPCの指定が可能
  - ➢ バケットポリシーやIAMポリシーを利用したSource IP / VPC CIDRによる制限は不可
- 別のVPCやSubnetを跨いだ直接のEndpointの利用は不可



http://docs.aws.amazon.com/ja_jp/AmazonVPC/latest/UserGuide/vpc-endpoints.html
http://aws.typepad.com/aws_japan/2015/05/vpcendpointfors3.html

S3 Access Points

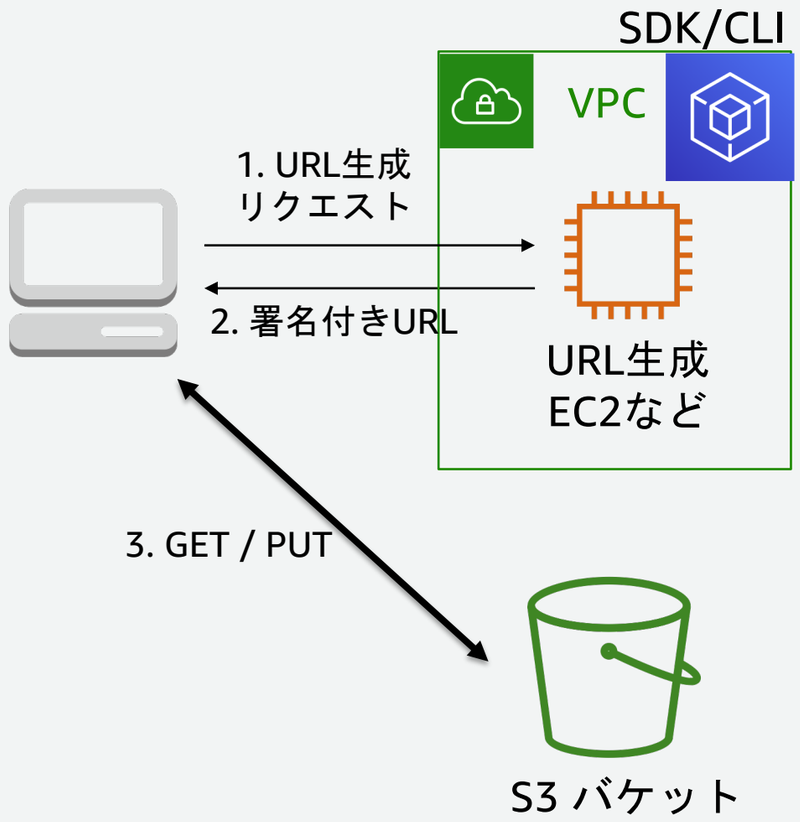
ホスト名・ARN・IAMリソースを持つ新しいS3リソース

- アプリケーションはバケット内のオブジェクトにアクセスするためにアクセスポイントを利用
- アクセスポイント毎にポリシーを適用可能
- アクセスポイントは特定のVPCに限定することが可能
- アクセスポイント名はアカウントとリージョンに固有のプライベート名前空間

Pre-signed Object URL (署名付きURL)

AWS SDK またはAWS CLI を利用して生成
S3上のプライベートなオブジェクトに対して**一定時間のアクセスを許可**

- セキュアにS3とのデータのやり取りが可能
- GET / PUTオペレーションで利用可能
  - 任意のユーザへの一時的なオブジェクト共有
  - 任意のユーザに一時的なS3へのアップロード権限の付与
- URLを生成したIAMユーザ/ロールの権限が用いられる
- 生成されたURLはExpireする前までが有効

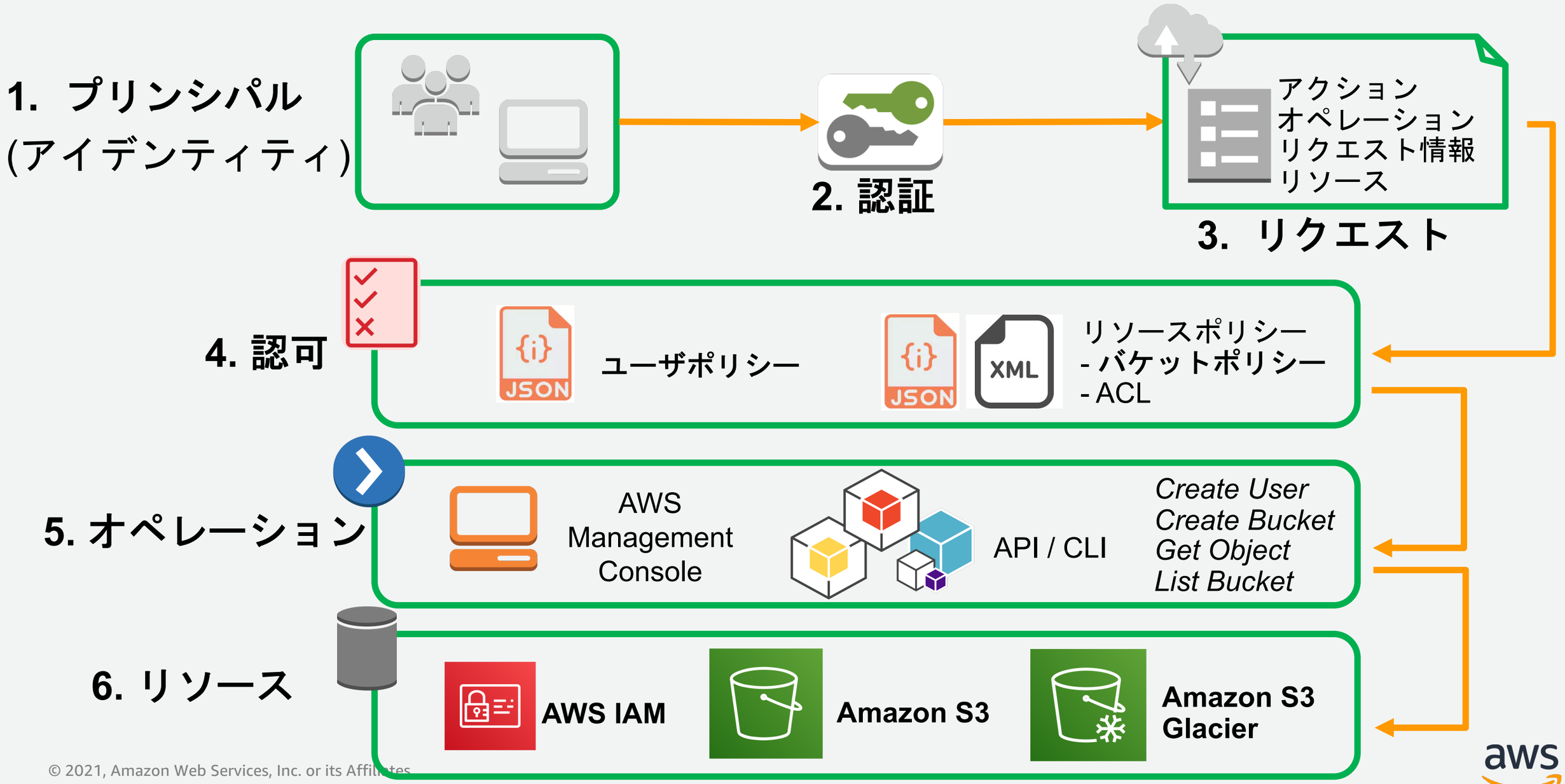


注意

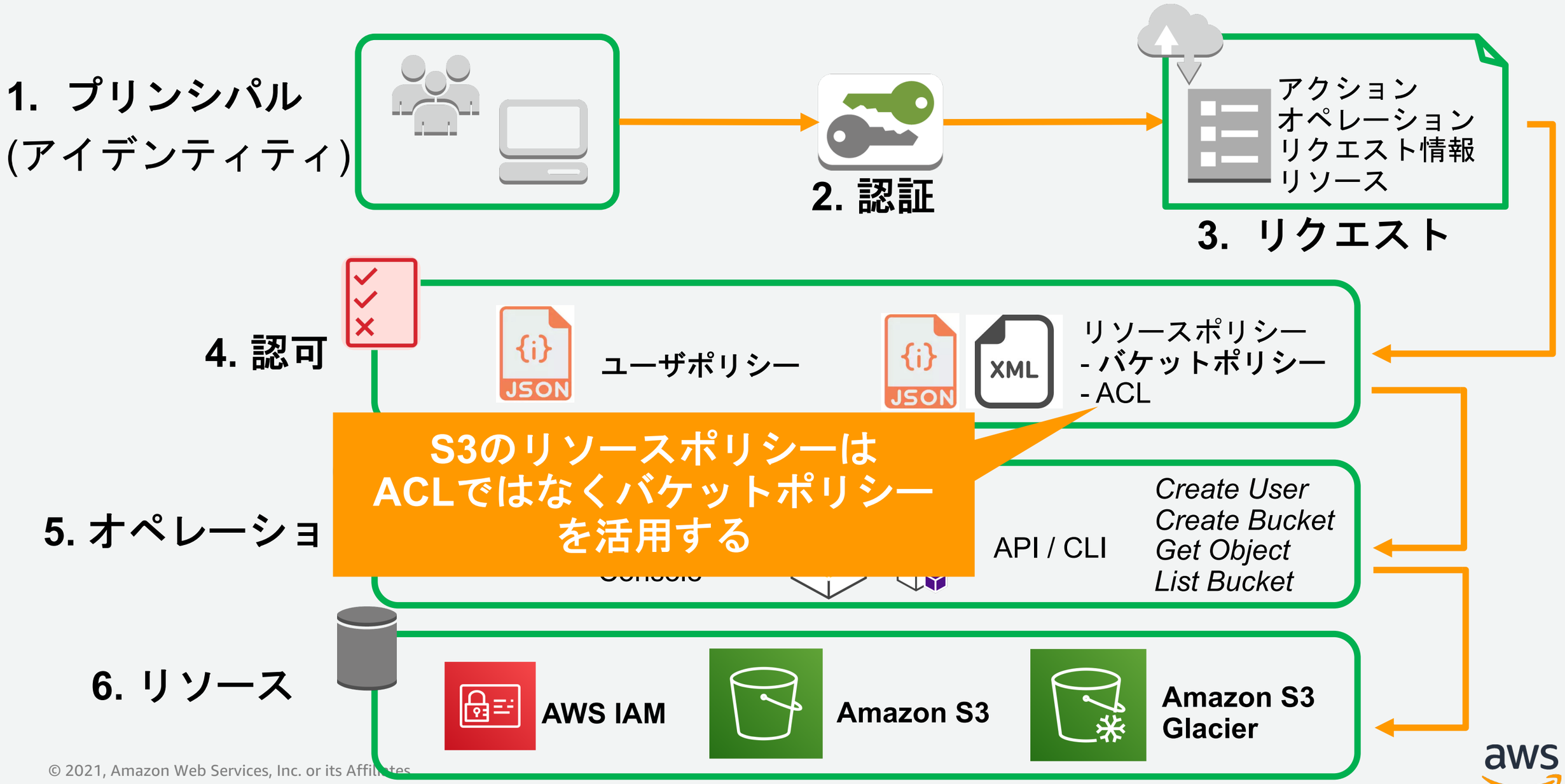
URLを知るユーザー全員がアクションを実行できる

Amazon S3における認可

Amazon S3にアクセスできる権限

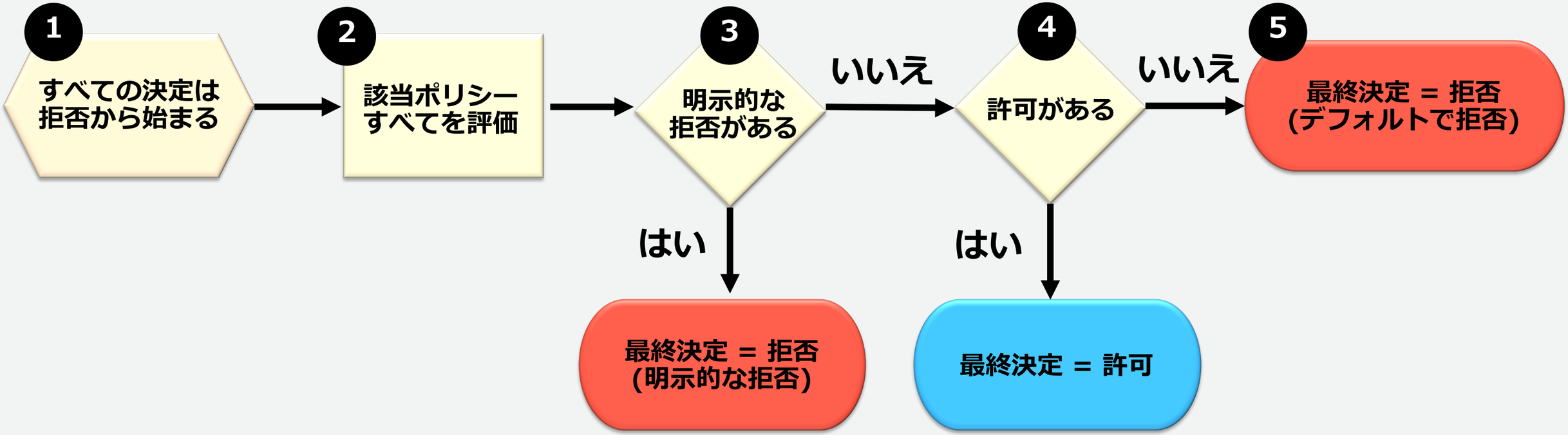


Amazon S3にアクセスできる権限



Amazon S3 がリクエストを承認する方法

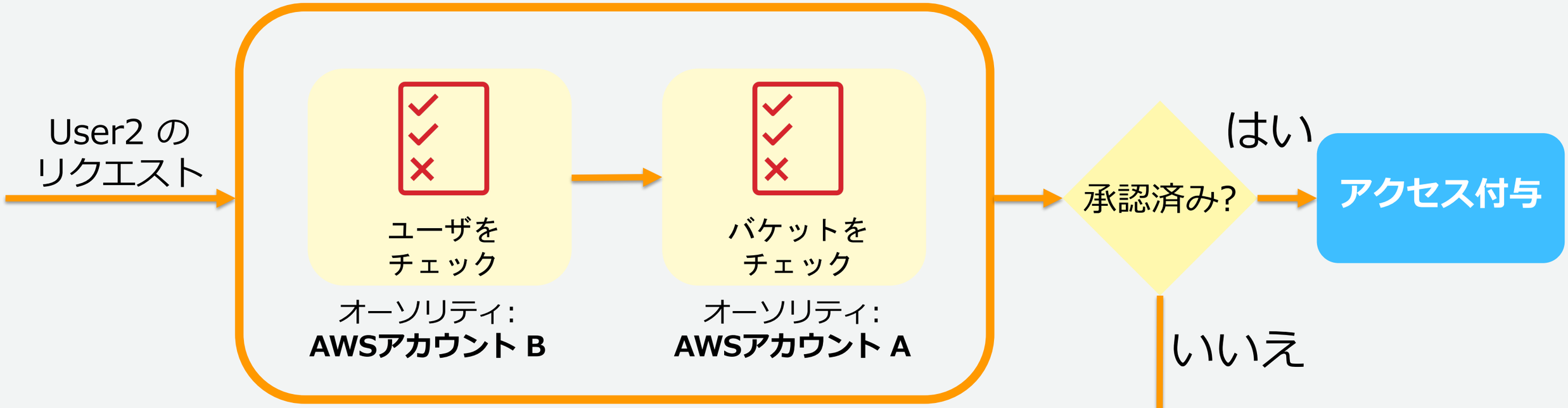
- すべてのポリシーにおける明示的な拒否は全ての許可を上書きする
  - ユーザーチェック – このユーザにアクセス許可を付与したかどうかをチェック
  - バケットチェック – バケット所有者がアクセス許可を付与したかどうかをチェック
  - オブジェクトチェック – 明示的な「許可」を探す



例1:　バケット所有者のAWSアカウントで完結する バケットオペレーションのリクエスト

リクエスタ： User1 (IAMユーザー)
User1の親アカウント： AWSアカウント A
バケット所有者： AWSアカウント A

例2: バケット所有者ではないAWSアカウントからのリクエスト (クロスアカウントアクセス)

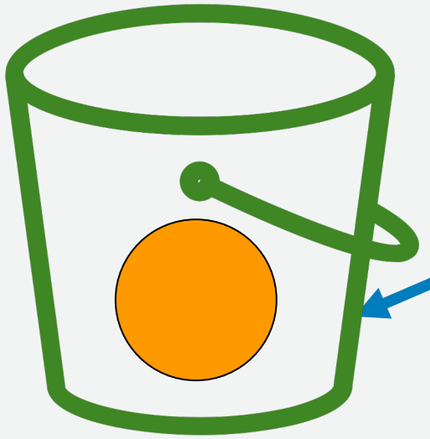


リクエスタ： User2 (IAMユーザー)
User2の親アカウント： AWSアカウント B
バケット所有者： AWSアカウント A

クロスアカウントアクセスの管理例 (バケットポリシーとIAMユーザポリシー)

アカウント - A

バケットポリシー



アカウント - A のS3バケット

アカウントBのPrincipalに、アカウントAのバケットへのアクセスを許可するポリシー

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": { "AWS": "arn:aws:iam::AccountB:user/AccountBUserName”},
      "Action": ["s3:GetObject", "s3:PutObject", "s3:PutObjectAcl”],
      "Resource": [ "arn:aws:s3:::AccountABucketName/*" ]
    }
  ]
}
```

アカウント - B

ユーザーポリシー

アカウントBのIAMユーザーに、アカウントAのバケットへアクセスを許可するポリシー

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": { "AWS": "arn:aws:iam::AccountB:user/AccountBUserName" },
      "Action": [ "s3:GetObject", "s3:PutObject", "s3:PutObjectAcl" ],
      "Resource": [ "arn:aws:s3:::AccountABucketName/*" ]
    }
  ]
}
```

クロスアカウントアクセスの管理例 (AssumeRole)

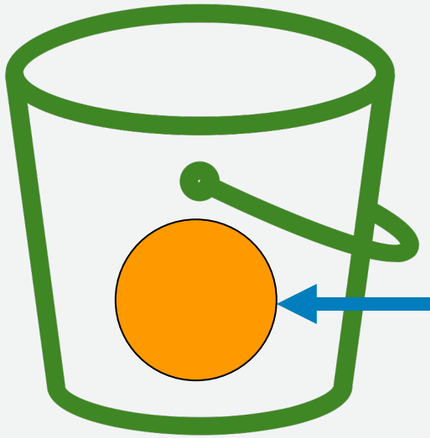
アカウントB のIAMユーザーがこのロールに対して AssumeRoleすることを許可するポリシー

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": { "AWS": "arn:aws:iam::AccountB:user/AccountBUserName" },
      "Action": "sts:AssumeRole”
    }
  ]
}
```

アカウント - A

AccountARole

取得



アカウント - A のS3バケット

アカウント - B

アカウントBのIAMユーザーに AccountARole に対するAssumeRoleを許可するポリシー

ユーザー
ポリシー

```
{
 "Version": "2012-10-17",
 "Statement": {
  "Effect": "Allow",
  "Action": "sts:AssumeRole",
  "Resource":"arn:aws:iam::AccountA:role/AccountARole"
                }
 }
```