



このコンテンツは公開から3年以上経過しており内容が古い可能性があります  
最新情報については[サービス別資料](#)もしくはサービスのドキュメントをご確認ください

# [AWS Black Belt Online Seminar]

## AWS Config Update

サービスカットシリーズ

Archived

Security Solutions Architect 桐谷 彰一  
2020/12/08

AWS 公式 Webinar  
<https://amzn.to/JPWebinar>



過去資料  
<https://amzn.to/JPArchive>



# 自己紹介



名前：桐谷 彰一（きりたに しょういち）

所属：ソリューションアーキテクト セキュリティスペシャリスト

経歴：セキュリティベンダー、ネットワークベンダーのプリセールスエンジニア  
エンタープライズ、官公庁のお客様のセキュリティ対策のご支援

好きなAWSサービス：



Amazon GuardDuty



AWS Security Hub

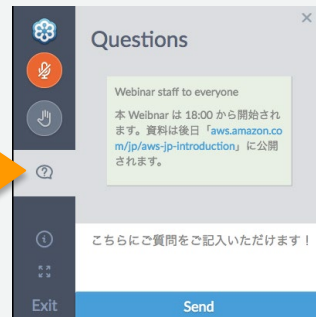
# AWS Black Belt Online Seminar とは

「サービス別」「ソリューション別」「業種別」のそれぞれのテーマに分かれて、アマゾンウェブ サービス ジャパン株式会社が主催するオンラインセミナーシリーズです。

## 質問を投げることができます！

- 書き込んだ質問は、主催者にしか見えません
- 今後のロードマップに関するご質問は  
お答えできませんのでご了承下さい

- ① 吹き出しをクリック
- ② 質問を入力
- ③ Sendをクリック



Twitter ハッシュタグは以下をご利用ください  
#awsblackbelt

# 内容についての注意点

- 本資料では2020年12月8日時点のサービス内容および価格についてご説明しています。最新の情報はAWS公式ウェブサイト(<http://aws.amazon.com>)にてご確認ください。
- 資料作成には十分注意しておりますが、資料内の価格とAWS公式ウェブサイト記載の価格に相違があった場合、AWS公式ウェブサイトの価格を優先とさせていただきます。
- 価格は税抜表記となっております。日本居住者のお客様には別途消費税をご請求させていただきます。
- AWS does not offer binding price quotes. AWS pricing is publicly available and is subject to change in accordance with the AWS Customer Agreement available at <http://aws.amazon.com/agreement/>. Any pricing information included in this document is provided only as an estimate of usage charges for AWS services based on certain information that you have provided. Monthly charges will be based on your actual use of AWS services, and may vary from the estimates provided.

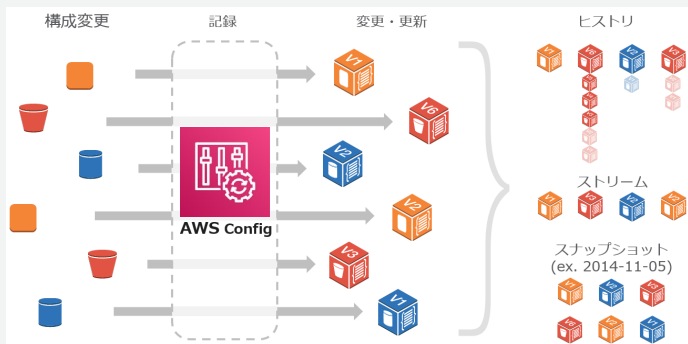
# 本日のアジェンダ

1. AWS Config のおさらい
2. 新機能：適合パックの概要
3. 新機能：サードパーティリソースサポートの概要
4. 組織のセキュリティ管理をより効率化するその他のアップデート
5. まとめ

# AWS Config のおさらい

## 構成情報の記録、評価を行うマネージドサービス

### 特徴 [\(http://aws.amazon.com/jp/config/\)](http://aws.amazon.com/jp/config/)



- AWS リソースの構成情報、変更履歴を記録
  - 構成情報を定期的にスナップショットとして保存
  - 必要に応じ SNS を使った通知も可能
- 構成情報を元に、現在のシステムがあるべき状態になっているか評価できる (Config Rules)

### 価格体系 [\(http://aws.amazon.com/jp/config/pricing/\)](http://aws.amazon.com/jp/config/pricing/)

- 1 回の設定項目の記録につき 0.003 USD
- Config Rules ルール評価ごとに 0.001USD
- ログが保存される Amazon S3 の料金

# AWS Config の利用例

特定のセキュリティグループを利用しているリソースを検索  
(SSH/RDPがフルオープンなSGが発見された！ 影響は？)

## サンプル SQL クエリ

List all EC2 instances currently running in my account	<a href="#">クエリの使用</a>
List all EC2 instances with AMI ID "ami-2a69aa47"	<a href="#">クエリの使用</a>
List all EBS volumes that are not in use	<a href="#">クエリの使用</a>
List all resources that are related to security group "sg-12345"	<a href="#">クエリの使用</a>
List all DynamoDB tables where server-side encryption is disabled	<a href="#">クエリの使用</a>
List all IAM users created between date "2018-12-01T00:00" and date "2019-02-28T00:00"	<a href="#">クエリの使用</a>
List all RDS instances running data	
List all RDS DB Instances that are p	
List all Lambda functions using runti	
List all S3 buckets where versioning	

## 高度なクエリ

下記の SQL クエリエディタを使用して、リソース設定データをクエリします。サンプルクエリ

### SQL クエリエディタ

```
1 SELECT
2   resourceId,
3   resourceName,
4   resourceType,
5   relationships
6 WHERE
7   relationships.resourceId = 'sg-e7k-9b'
```

## 結果

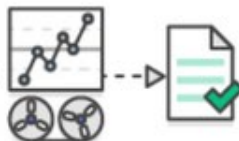
resourceId	resourceName	resourceType	relationships
eni-060e	i10765db7	-	4 個の項目
eni-07f22	i62125ef3	-	4 個の項目
i-081d1e	i2548c0	-	5 個の項目
i-098590	i2bff2ed	-	5 個の項目
test_inVf	test_inVPC	AWS::Lambda::Function	4 個の項目
vpc-30c3		AWS::EC2::VPC	23 個の項目



# AWS Config による構成管理



AWSリソースの  
構成変更



AWS Config  
による取得



構成管理/記録の保存  
過去の構成状況  
依存関係の確認  
コンプライアンス準拠

# AWS Config Rules の利用例

## パブリック読み込みが許可されたS3バケットを把握

s3-bucket-public-read-prohibited

再評価 結果の削除 編集

説明 S3 バケットが読み取りパブリックアクセスを許可していないことを確認します。S3 バケットポリシーまたはバケット ACL で読み取りパブリックアクセスを許可している場合、そのバケットは準拠していません。

コンプライアンス状況 非準拠

リソース ID リソースタイプ リソースのコンプライアンス状況 アクション

リソース ID	リソースタイプ	リソースのコンプライアンス状況	アクション
sk-test	S3 Bucket	非準拠	該当なし

AWSが提供するマネージドルールを利用して、リスクがある設定を簡単に把握  
(160以上のルールを提供) ※2020/12/08現在  
+カスタムルールで特定の評価にも対応

## 必要があれば修復アクションで1click/自動で対応

修復アクションを選択

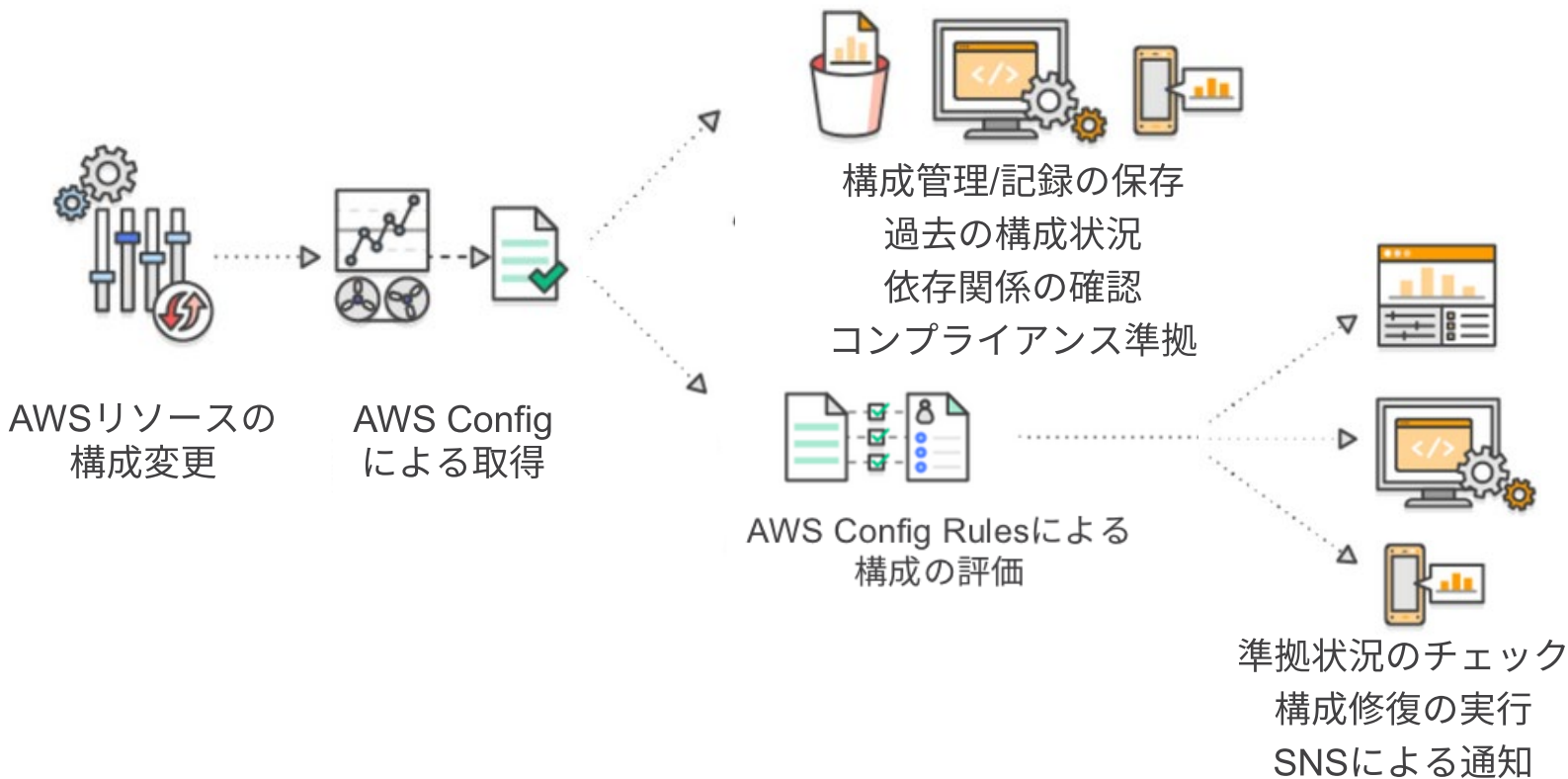
修復アクションの実行は、AWS Systems Manager Automationを使用して達成されます。AWS が推奨する一連の修復アクションまたはカスタムの修復アクションから選択します。ルールを修復するには、テーブルから範囲内のすべての非準拠リソースを選択します。

修復アクション AWS-DisableS3BucketPublicReadWrite  
Disable S3-Bucket's public WriteRead access via private ACL

リソース ID パラメータ S3BucketName

修復アクションを実行し、パブリック読み込みアクセスを許可を“無効”にすることも可能

# AWS Config Rules で構成情報を評価



# 基本的な機能については、前回の BlackBelt をご確認ください

[AWS Black Belt Online Seminar] AWS Config 資料及び QA 公開

<https://aws.amazon.com/jp/blogs/news/webinar-bb-aws-config-2019/>

The screenshot shows the top navigation bar of the AWS Japan website with the AWS logo and links for 'お問い合わせ', 'サポート', and 'アカウント'. Below the navigation bar, there are links for 're:Invent', '製品', 'ソリューション', '料金', 'ドキュメント', '学ぶ', and 'パートナー'. A search bar is visible on the right. The main content area features the title '[AWS Black Belt Online Seminar] AWS Config 資料及び QA 公開' by AWS Japan Staff, dated June 30, 2019. A paragraph of text describes the seminar and the availability of materials. Below the text is a video player with a play button and a thumbnail image of the seminar content. The thumbnail includes the AWS logo, the title, the speaker's name (Shinya Tani), and QR codes for downloading materials.

The screenshot shows a slide from the seminar. It features the AWS logo at the top left, followed by the title '[AWS Black Belt Online Seminar] AWS Config' and the subtitle 'サービスカットシリーズ'. Below the title, it identifies the speaker as 'Security Solutions Architect 棚谷 彰一' and provides the date '2019/06/18'. There are two QR codes and two URLs: 'https://amazon.jp/JP/Webinar' and 'https://amazon.jp/JP/Archives'. A footer contains the copyright notice '© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.' At the bottom of the slide, there is a navigation bar with a back arrow, a '1 of 50' indicator, a forward arrow, and a LinkedIn icon.

20190618 AWS Black Belt Online Seminar AWS Config from Amazon Web Services Japan

# 新機能のご紹介

# AWS Config のアップデート

## 適合パック



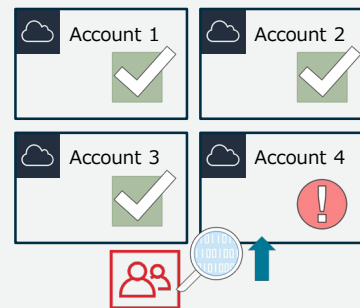
組織全体への展開と  
コンプライアンスチェックを  
より簡単に

## サードパーティ リソースへの対応



AWS リソース以外の  
設定変更や  
追跡が可能に

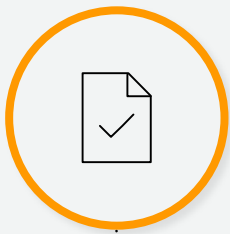
## その他の アップデート



組織のセキュリティ管理を  
より効率的に

# 適合パックの概要

# AWS Config 適合パック (Conformance Pack)



## 構成管理のための共通コンプライアンスフレームワーク

- 複数の Config Rule と修復アクションをまとめて用途に応じてパッケージ化
- 単一AWSアカウント、AWS Organizations の組織全体に対して適用可能

## 不変性(immutable)

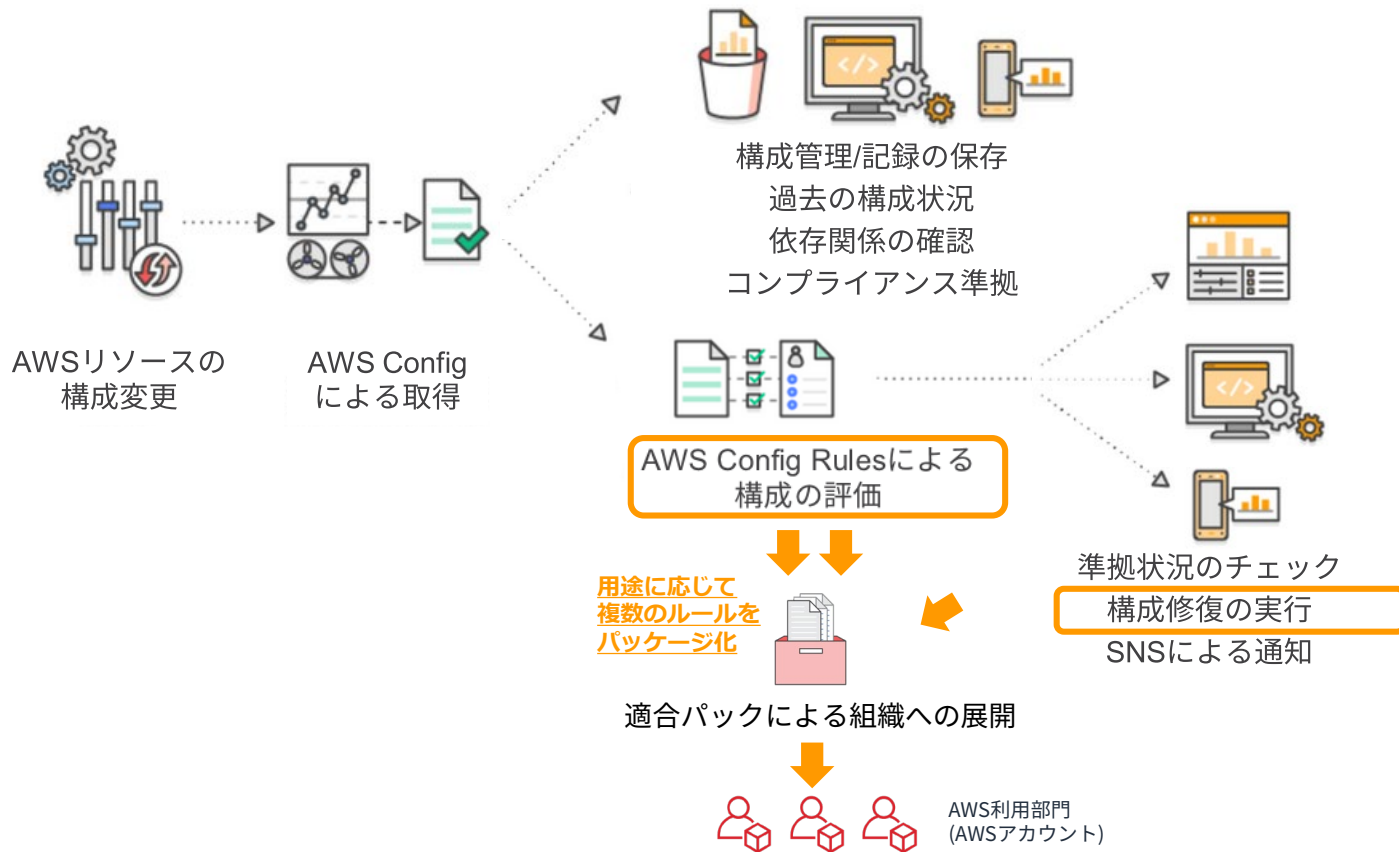
- 個々のルールは、アクセス権限やアカウントの権限に関わらず、デプロイされた適合パックの外部から変更不可
- 組織のマスターアカウントから展開した適合パックは、メンバーアカウントから変更不可

## 価格体系

- 適合パックによるルール評価ごとに 0.0012USD



# AWS Config 適合パック



# 適合パックの詳細

AWS Config > 適合パック > ControlTowerSample

## ControlTowerSample

ルール

設定

### ルール (12)

🔍 名前またはコンプライアンスステータスでルールをフィルタリングする

< 1 > ⚙️

名前	Remediation action	Compliance
↓ 1つの適合パックに複数の Config Rule		
CheckForRestrictedSshPolicy-conformance-pack-beqouusak	設定されていません	⚠️ 非準拠
CheckForEbsOptimizedInstance-conformance-pack-beqouusak	設定されていません	✅ 準拠
CheckForS3PublicWrite-conformance-pack-beqouusak	設定されていません	✅ 準拠
CheckForRestrictedCommonPortsPolicy-conformance-pack-beqouusak	設定されていません	⚠️ 非準拠
CheckForRootMfa-conformance-pack-beqouusak	設定されていません	⚠️ 非準拠
CheckForS3PublicRead-conformance-pack-beqouusak	設定されていません	✅ 準拠
CheckForRdsPublicAccess-conformance-pack-beqouusak	設定されていません	✅ 準拠

# 適合パックのサンプルテンプレート

「運用のベストプラクティス」など、様々なサンプルを提供  
50個を超えるテンプレートを用意（2020/12/08現在）

ドキュメント > AWS Config > 開発者ガイド フィードバック

## コンフォーマンスパックのサンプルテンプレート

PDF

AWS Configコンソールに表示されるコンフォーマンスパックのYAMLテンプレートを次に示します。コンフォーマンスパックテンプレート内では、1つ以上のAWS Configルールと修正アクションを使用できます。コンフォーマンスパックに一覧表示されているAWS Configルールは、AWS Config管理ルールまたはAWS Configカスタムルールにすることができます。すべてのコンフォーマンスパックテンプレートは、からダウンロードできます[GitHub](#)。

### トピック

- AWS Control Tower 発券的ガードレールコンフォーマンスパック
- ABS CCI 2.0 マテリアルワークロードの運用のベストプラクティス
- ABS CCI 2.0 標準ワークロードの運用のベストプラクティス
- ACSC 基本的な 8 運用のベストプラクティス
- ACSC ISM 運用のベストプラクティス
- AI と ML 運用のベストプラクティス
- Amazon の運用上のベストプラクティス DynamoDB
- Amazon S3の運用に関するベストプラクティス
- APRA CPG 234 運用のベストプラクティス
- アセット管理の運用のベストプラクティス
- AWS IDとアクセス管理の運用に関するベストプラクティス
- AWS Well-Architected フレームワークの信頼性の柱運用のベストプラクティス
- AWS Well-Architected フレームワークセキュリティ柱の運用のベストプラクティス
- BCP と DR 運用のベストプラクティス

GitHub にて公開 + 日々追加中

awsconfigs / aws-config-rules Watch 100 Star 966 Fork 463

<> Code Issues 36 Pull requests 23 Actions Projects ...

master - aws-config-rules / aws-config-conformance-packs / Go to file

tysodotcom Updated names of the CIS AWS Foundation benchmark y... 24 days ago History

AWS-Control-Tower-Dete...	Added Control Tower Conformance Pack	2 months ago
Operational-Best-Practice...	Additional conformance packs	2 months ago
Operational-Best-Practice...	Additional conformance packs	2 months ago
Operational-Best-Practice...	Updated ACSC Related Conformance Packs	last month
Operational-Best-Practice...	Updated ACSC Related Conformance Packs	last month

# 適合パックのサンプルテンプレート (1/2)

各国や業界のセキュリティ要件、ガイドライン、レギュレーションなどに対するテンプレート

テンプレート名	テンプレート名
ABS CCIG 2.0 マテリアルワークロードの運用のベストプラクティス	K-ISMS 運用のベストプラクティス
ABS CCIG 2.0 標準ワークロードの運用のベストプラクティス	MAS 通知 655 運用のベストプラクティス
ACSC 基本的な 8 運用のベストプラクティス	MAS TRMG 2013 運用のベストプラクティス
ACSC ISM 運用のベストプラクティス	NC TRMG の運用のベストプラクティス
APRA CPG 234 運用のベストプラクティス	NERC CIP 運用のベストプラクティス
BNM 運用のベストプラクティス RMIIT	NCSC クラウドセキュリティ原則の運用のベストプラクティス
CIS 運用のベストプラクティス	NIST 800-53 リビジョン 4 運用のベストプラクティス
CMMC レベル 1 運用のベストプラクティス	NIST 800 171 運用のベストプラクティス
CMMC レベル 2 運用のベストプラクティス	NIST CIS 運用のベストプラクティス
運用のベストプラクティス FedRAMP (低)	NYDFS 23 運用のベストプラクティス
運用のベストプラクティス FedRAMP (中)	PCI DSS 3.2.1 運用のベストプラクティス
FFIEC の運用のベストプラクティス	RBI MD-ITF 運用のベストプラクティス
HIPAA セキュリティ運用のベストプラクティス	

# 適合パックのサンプルテンプレート (2/2)

各AWSサービスのベストプラクティスや、Well-Architected フレームワークの視点で用意されたテンプレート

テンプレート名	テンプレート名
AWS Control Tower 発見的ガードレールコンフォーマンスパック	クエリに関する運用のベストプラクティス - コーディングのベストプラクティス
AI と ML 運用のベストプラクティス	アセット管理の運用のベストプラクティス
Amazon の運用上のベスト プラクティス DynamoDB	ロードバランシング運用のベストプラクティス
Amazon S3の運用に関するベストプラクティス	ログ記録運用のベストプラクティス
EC2 運用のベストプラクティス	管理とガバナンスサービスの運用のベストプラクティス
AWS IDとアクセス管理の運用に関するベストプラクティス	モニタリング運用のベストプラクティス
AWS Well-Architected フレームワークの信頼性の柱運用のベストプラクティス	ネットワーキングとコンテンツ配信サービスの運用のベストプラクティス
<b>AWS Well-Architected フレームワーク セキュリティ柱の運用のベストプラクティス</b>	パブリックにアクセス可能なリソース運用のベストプラクティス
BCP と DR 運用のベストプラクティス	セキュリティ、アイデンティティ、およびコンプライアンスサービスの運用のベストプラクティス
コンピューティングサービス運用のベストプラクティス	サーバーレス運用のベストプラクティス
データの耐障害性に関する運用のベストプラクティス	ストレージサービス運用のベストプラクティス
データベースサービス運用のベストプラクティス	修正アクションを含むテンプレートの例
データレイクおよび分析サービスの運用のベストプラクティス	カスタムコンフォーマンスパック
暗号化とキー管理の運用のベストプラクティス	

# 適合パックの設定イメージ (1/3)

## AWS Well-Architected フレームワークセキュリティ柱の運用のベストプラクティス

**AWS Config** ×

ダッシュボード  
**適合パック**  
ルール  
リソース  
▼ アグリゲータ  
ルール  
リソース  
認証  
高度なクエリ  
設定

AWS Config > 適合パック

### 適合パック

適合パックは、AWS アカウントで単一のエンティティとしてデプロイおよびモニタリングできる AWS Config ルールおよび修復アクションのコレクションです。 [詳細はこちら](#)

適合パック  アクション ▼

🔍 名前またはコンプライアンスステータスでルールをフィルタリングする

< 1 > ⚙️

	名前 ▲	デプロイ ▼	コンプライアンス ▼
<input type="radio"/>	MyS3CPack	完了しました	⚠️ 非準拠
<input type="radio"/>	OrgConformsPack-CISPack-gahzupvp	完了しました	⚠️ 非準拠

最新情報 [🔗](#)  
ドキュメント [🔗](#)  
パートナー [🔗](#)  
よくある質問 [🔗](#)  
料金表 [🔗](#)

### テンプレートを指定

#### テンプレートの詳細

##### 適合パックテンプレート

すべての適合パックはテンプレートに基づいています。テンプレートは、AWS Config ルールと修復アクションをデプロイする AWS アカウントとリージョンに関する設定情報を含む YAML ファイルです。

サンプルテンプレートを使用

テンプレートをアップロード

#### サンプルテンプレート

🔍 well

Operational Best Practices for AWS Well Architected Reliability Pillar

Operational Best Practices for AWS Well Architected Security Pillar

サンプルテンプレートを選択

サンプルテンプレートを表示するには、次を参照してください [適合パックのサンプルテンプレート](#) です。 [🔗](#)

キャンセル

# 適合パックの設定イメージ (2/3)

## AWS Well-Architected フレームワークセキュリティ柱の運用のベストプラクティス

### 適合パックの詳細を指定

#### 適合パックの詳細

リージョン  
Asia Pacific (Tokyo)

適合パック名  
このテンプレートのデプロイに名前を付けます。

mycpack-WASEC

適合パック名には、文字 (A~Z および a~z)、数字 (0~9)、  
できません。

#### パラメータ - オプション

パラメータはテンプレートで定義され、適合パックを作成

パラメータが入力されていません

パラメータを追加

### 適合パックの確認とデプロイ

#### テンプレートの詳細

サンプルテンプレート  
Operational Best Practices for AWS Well Architected Security Pillars

#### 適合パックの詳細

リージョン  
Asia Pacific (Tokyo)

適合パック名  
mycpack-WASEC

#### パラメータ - オプション

パラメータが入力されていま

キャンセル 前へ 適合パックをデプロイ

### 適合パック

適合パックは、AWS アカウントで単一のエンティティとしてデプロイおよびモニタリングできる AWS Config ルールおよび修復アクションのコレクションです。 [詳細はこちら](#)

適合パック

🔍 名前またはコンプライアンスステータスでルールをフィルタリングする

< 1 > ⚙️

	名前 ▲	デプロイ ▼	コンプライアンス ▼
<input type="radio"/>	mycpack-WASEC	完了しました	⚠️ 非準拠
<input type="radio"/>	MyS3CPack	完了しました	⚠️ 非準拠
<input type="radio"/>	OrgConformsPack-CISPack-gahzupvp	完了しました	⚠️ 非準拠

# 適合パックの設定イメージ (3/3)

## AWS Well-Architected フレームワークセキュリティ柱の運用のベストプラクティス

### mycpack-WASEC

デプロイ:  
🟢 完了しました

**ルール** | 設定

#### ルール (50)

🔍 名前またはコンプライアンスステータスでルールをフィルタリングする

名前	修復アクション	コンプライアンス
lambda-inside-vpc-conformance-pack-komqljuhb	設定されていません	⚠️ 非準拠
s3-bucket-default-lock-enabled-conformance-pack-komqljuhb	設定されていません	⚠️ 非準拠
account-part-of-organizations-conformance-pack-komqljuhb	設定されていません	🟢 準拠
lambda-function-public-access-prohibited-conformance-pack-komqljuhb	設定されていません	🟢 準拠
guardduty-enabled-centralized-conformance-pack-komqljuhb	設定されていません	🟢 準拠
elasticsearch-encrypted-at-rest-conformance-pack-komqljuhb	設定されていません	⚠️ 非準拠
s3-bucket-public-write-prohibited-conformance-pack-komqljuhb	設定されていません	🟢 準拠
elasticsearch-node-to-node-encryption-check-conformance-pack-komqljuhb	設定されていません	⚠️ 非準拠
<b>vpc-sg-open-only-to-authorized-ports-conformance-pack-komqljuhb</b>	設定されていません	⚠️ 非準拠

### vpc-sg-open-only-to-authorized-ports-conformance-pack-komqljuhb

アクション ▼

編集

#### ▼ ルールの詳細

説明  
Checks whether any security groups with inbound 0.0.0.0/0 have TCP or UDP ports accessible. The rule is NON\_COMPLIANT when a security group with inbound 0.0.0.0/0 has a port accessible which is not specified in the rule parameters.

トリガータイプ  
• オーバーサイジングの設定変更  
• 設定変更

最後に成功した評価  
🟢 2020年11月25日 17:27

変更範囲  
リソース

リソースタイプ  
EC2 SecurityGroup

Config ルール ARN  
arn:aws:config:ap-northeast-1:27425-122:config-rule/aws-service-rule/config-conforms.amazonaws.com/config-rule-dfsyre

#### パラメータ

キー	タイプ	値	説明
authorizedTcpPorts	String	443	Comma-separated list of TCP ports authorized to be open to 0.0.0.0/0. Ranges are defined by dash, for example, "443,1020-1025".
authorizedUdpPorts	String		Comma-separated list of UDP ports authorized to be open to 0.0.0.0/0. Ranges are defined by dash, for example, "500,1020-1025".

#### ▼ 対象範囲内のリソース

詳細を表示 | 修復 | 🔄

非準拠 ▼

ID	タイプ	ステータス	注釈
sg-03ef42b1c1262	EC2 SecurityGroup	-	One or more TCP ports (22) are not in range of the authori
sg-06617e537da	EC2 SecurityGroup	-	One or more TCP ports (3389) are not in range of the autho
sg-08ce0932a1f	EC2 SecurityGroup	-	One or more TCP ports (80) are not in range of the authori



# 適合パック : AWS Well-Architected フレームワークセキュリティ柱の運用のベストプラクティス

各質問に関連した Config Rule がまとめられている

質問	含まれる Config Rule
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？	account-part-of-organizations, codebuild-project-envvar-awscred-check, iam-root-access-key-check, root-account-hardware-mfa-enabled, root-account-mfa-enabled,
SEC 2. ユーザー ID とマシン ID はどのように管理したらよいでしょうか？	access-keys-rotated, emr-kerberos-enabled, iam-password-policy, iam-user-group-membership-check, iam-user-mfa-enabled, iam-root-access-key-check, root-account-hardware-mfa-enabled, root-account-mfa-enabled, iam-user-unused-credentials-check, mfa-enabled-for-iam-console-access, secretsmanager-rotation-enabled-check, secretsmanager-scheduled-rotation-success-check,
SEC 3. 人とマシンのアクセス許可はどのように管理すればよいでしょうか？	elb-deletion-protection-enabled, emr-kerberos-enabled, iam-group-has-users-check, iam-no-inline-policy-check, iam-policy-no-statements-with-admin-access, iam-user-no-policies-check, rds-instance-deletion-protection-enabled,
SEC 4. セキュリティイベントをどのように検出し、調査していますか？	api-gw-execution-logging-enabled, cloud-trail-cloud-watch-logs-enabled, cloudtrail-enabled, cloud-trail-encryption-enabled, cloud-trail-log-file-validation-enabled, cloudtrail-s3-dataevents-enabled, cloudtrail-security-trail-enabled, cloudwatch-alarm-action-check, cw-loggroup-retention-period-check, elb-logging-enabled, guardduty-enabled-centralized, multi-region-cloudtrail-enabled, rds-logging-enabled, redshift-cluster-configuration-check, s3-bucket-logging-enabled, securityhub-enabled, vpc-flow-logs-enabled, wafv2-logging-enabled
SEC 5. ネットワークリソースをどのように保護しますか？	alb-waf-enabled, dms-replication-not-public, ebs-snapshot-public-restorable-check, ec2-instance-no-public-ip, ec2-security-group-attached-to-eni, elasticsearch-in-vpc-only, emr-master-no-public-ip, restricted-ssh, ec2-instances-in-vpc, internet-gateway-authorized-vpc-only, lambda-function-public-access-prohibited, lambda-inside-vpc, rds-instance-public-access-check, rds-snapshots-public-prohibited, redshift-cluster-public-access-check, restricted-common-ports, s3-account-level-public-access-blocks, sagemaker-notebook-no-direct-internet-access, vpc-default-security-group-closed, vpc-sg-open-only-to-authorized-ports

# 適合パック : AWS Well-Architected フレームワークセキュリティ柱の運用のベストプラクティス

質問	含まれる Config Rule
SEC 6. コンピューティングリソースをどのように保護していますか?	ec2-imdsv2-check, ec2-instance-managed-by-systems-manager, ec2-managedinstance-association-compliance-status-check, ec2-managedinstance-patch-compliance-status-check
SEC 7. どのようにデータを分類していますか?	cw-loggroup-retention-period-check, guardduty-non-archived-findings
SEC 8. 保管時のデータをどのように保護していますか?	api-gw-cache-enabled-and-encrypted, cloud-trail-encryption-enabled, cloudwatch-log-group-encrypted, cmk-backing-key-rotation-enabled, dms-replication-not-public, dynamodb-table-encrypted-kms, ebs-snapshot-public-restorable-check, ec2-ebs-encryption-by-default, ec2-instance-no-public-ip, efs-encrypted-check, elasticsearch-encrypted-at-rest, elasticsearch-in-vpc-only, emr-master-no-public-ip, encrypted-volumes, ec2-instances-in-vpc, kms-cmk-not-scheduled-for-deletion, lambda-function-public-access-prohibited, lambda-inside-vpc, rds-instance-public-access-check, rds-snapshot-encrypted, rds-snapshots-public-prohibited, rds-storage-encrypted, redshift-cluster-configuration-check, redshift-cluster-public-access-check, s3-account-level-public-access-blocks, s3-bucket-default-lock-enabled, s3-bucket-public-read-prohibited, s3-bucket-public-write-prohibited, s3-bucket-server-side-encryption-enabled, s3-bucket-versioning-enabled, s3-default-encryption-kms, sagemaker-endpoint-configuration-kms-key-configured, sagemaker-notebook-instance-kms-key-configured, sagemaker-notebook-no-direct-internet-access, sns-encrypted-kms
SEC 9. 転送時のデータをどのように保護していますか?	acm-certificate-expiration-check, ulb-http-drop-invalid-header-enabled, alb-http-to-https-redirection-check, elasticsearch-node-to-node-encryption-check, elb-acm-certificate-required, elb-tls-https-listeners-only, redshift-require-tls-ssl, s3-bucket-ssl-requests-only
SEC 10. インシデントの予測、対応、復旧はどのように行いますか?	-

存在する AWS リソース「のみ」「自動で」評価 = 利用状況に即した準拠状況を確認

# 各サンプルテンプレートの詳細説明

## AWS Well-Architected フレームワークセキュリティ柱の運用のベストプラクティス

PDF

コンフォーマンスパックは、マネージド型またはカスタムの AWS Config ルールと AWS Config 修復アクションを使用して、セキュリティ、運用中、またはコスト最適化ガバナンスチェックを作成できるように設計された汎用コンプライアンスフレームワークを提供します。コンフォーマンスパックは、サンプルテンプレートとして、特定のガバナンスまたはコンプライアンス標準への準拠を完全に保証するように設計されていません。サービスの使用が、適用可能な法的および規制の要件を満たしているかどうかは、お客様が評価してください。

以下は、アマゾンウェブサービスの Well-Architected フレームワークセキュリティ柱と AWS マネージド Config ルール間のマッピングのサンプルです。各 Config ルールは、特定の AWS リソースに適用され、柱の設計原則の 1 つ以上に関連しています。Well-Architected フレームワークカテゴリは、複数の Config ルールに関連している場合があります。これらのマッピングに関する詳細およびガイダンスについては、以下の表を参照してください。

このコンフォーマンスパックは、AWS Security Assurance Services LLC (AWS SAS) によって検証されました。AWS SAS は、Payment Card Industry Qualified Security Assessors (QSA)、HITRUST 認定 共通セキュリティフレームワークプロプライナー (CCSFP)、およびさまざまな業界フレームワークのガイダンスと評価を提供することを認定されたコンプライアンスプロフェッショナルのチームです。AWS SAS プロフェッショナルはこのコンフォーマンスパックを設計し、お客様が Well-Architected フレームワークセキュリティの柱設計原則のサブセットにアクセスできるようにしました。

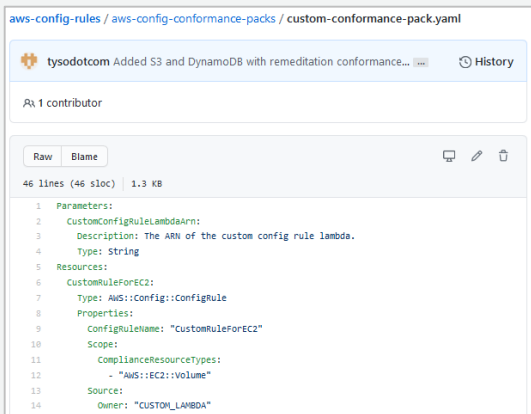
**AWS リージョン:** を除く、サポートされているすべての AWS リージョン中東 (バレーン)

コントロール ID	コントロールの説明	AWS Config ルール	ガイダンス
SEC-1	ワークロードはどのように安全に運用しますか? ワークロードを安全に運用するには、セキュリティのすべての領域に対してベストプラクティスの上書きを適用する必要があります。組織レベルとワークロードレベルで運用上の優劣性において定義した要件とプロセスを取得し、すべての領域に適用します。AWS や業界の推奨事項、脅威インテリジェンスに関する最新情報入手して、脅威モデルとコントロール目的の進化に役立てることができます。セキュリティプロセス、テスト、検証を自動化することで、セキュリティオペレーションをスケールすることが可能になります。	アカウント/パート of organizations	AWS Organizations 内の AWS アカウントの一元管理は、アカウントが準拠していることを確認するのに役立ちます。一元化されたアカウントガバナンスがないと、アカウント設定が不整合になり、リソースと機密データが公開される可能性があります。
SEC-1	ワークロードはどのように安全に運用しますか? ワークロードを安全に運用するには、セキュリティのすべての領域に対してベストプラクティスの上書きを適用する必要があります。組織レベルとワークロードレベルで運用上の優劣性において定義した要件とプロセスを取得し、すべての領域に適用します。AWS や業界の推奨事項、脅威インテリジェンスに関する最新情報入手して、脅威モデルとコントロール目的の進化に役立てることができます。セキュリティプロセス、テスト、検証を自動化することで、セキュリティオペレーションをスケールすることが可能になります。	codebuild-project-envvar-awscred-check	認証情報 AWS_ACCESS_KEY_ID および AWS_SECRET_ACCESS_KEY が AWS Codebuild プロジェクト環境に存在しないことを確認します。これらの変数はクリアテキストで保存しないでください。これらの変数をクリアテキストに保存すると、意図しないデータ漏えいや不正アクセスの原因になります。
SEC-1	ワークロードはどのように安全に運用しますか? ワークロードを安全に運用するには、セキュリティのすべての領域に対してベストプラクティスの上書きを適用する必要があります。組織レベルとワークロードレベルで運用上の	iam-root-access-key-check	ルートユーザーに AWS Identity and Access Management (IAM) ロールにアタッチされたアクセスキーがないことを確認することにより、システムとアセットへのアクセスを制御できます。ルートアクセスキーが削除されていることを確認します。代わりに、ロールベースの AWS アカウントを作成して使用し、最小機能の原則を組み込みます。

# 適合パック : カスタムテンプレート

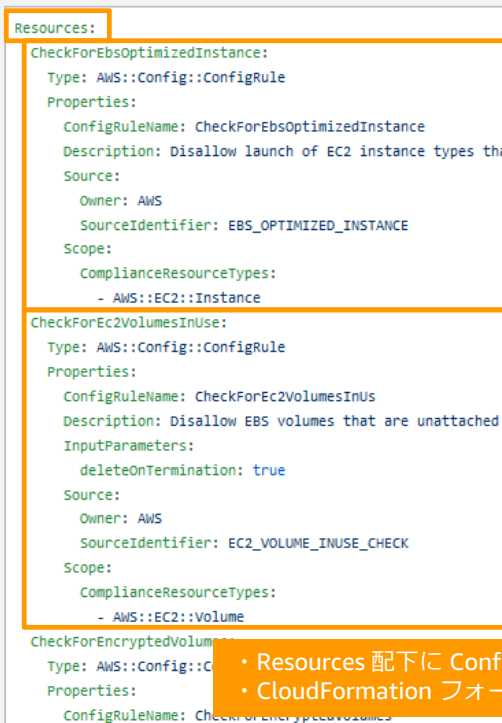
## ユーザー独自の評価内容をカスタムテンプレートとして作成可能

1. GitHubからテンプレートをダウンロード  
(yaml形式)



```
aws-config-rules / aws-config-conformance-packs / custom-conformance-pack.yaml
tysodotcom Added S3 and DynamoDB with remediation conformance... History
Rt 1 contributor
Raw Blame
46 lines (46 sloc) | 1.3 KB
1 Parameters:
2   CustomConfigRuleLambdaArn:
3     Description: The ARN of the custom config rule lambda.
4     Type: String
5 Resources:
6   CustomRuleForEc2:
7     Type: AWS::Config::ConfigRule
8     Properties:
9       ConfigRuleName: "CustomRuleForEc2"
10      Scope:
11        ComplianceResourceTypes:
12          - "AWS::EC2::Volume"
13      Source:
14        Owner: "CUSTOM_LAMBDA"
```

2. 修正してカスタムテンプレート化



```
Resources:
  CheckForEbsOptimizedInstance:
    Type: AWS::Config::ConfigRule
    Properties:
      ConfigRuleName: CheckForEbsOptimizedInstance
      Description: Disallow launch of EC2 instance types that
      Source:
        Owner: AWS
        SourceIdentifier: EBS_OPTIMIZED_INSTANCE
      Scope:
        ComplianceResourceTypes:
          - AWS::EC2::Instance
  CheckForEc2VolumesInUse:
    Type: AWS::Config::ConfigRule
    Properties:
      ConfigRuleName: CheckForEc2VolumesInUse
      Description: Disallow EBS volumes that are unattached
      InputParameters:
        deleteOnTermination: true
      Source:
        Owner: AWS
        SourceIdentifier: EC2_VOLUME_INUSE_CHECK
      Scope:
        ComplianceResourceTypes:
          - AWS::EC2::Volume
  CheckForEncryptedVolume:
    Type: AWS::Config::C
    Properties:
      ConfigRuleName: CheckForEncryptedVolumes
```

- Resources 配下に Config Rule を記載
- CloudFormation フォーマットの Config Rules 定義

3. 適合パックのデプロイ時にアップロードして利用



テンプレートを指定

テンプレートの詳細

適合パックテンプレート  
すべての適合パックはテンプレートに基づいています。テンプレートは、AWS Config ルールと修復アクションをデプロイする AWS アカウントとリージョンに関する設定情報を含む YAML ファイルです。

サンプルテンプレートを使用  テンプレートをアップロード

テンプレートの場所

テンプレートは、AWS Config ルールと修復アクションを含む YAML ファイルです。

テンプレートソースを指定

Amazon S3 バケットの既存のテンプレートを使用するか、ローカルマシンからテンプレートをアップロードします。テンプレートが 50 KB を超える場合は、Amazon S3 バケットにアップロードして、そのバケットを選択します。

Amazon S3 バケット  テンプレートファイルをアップロード

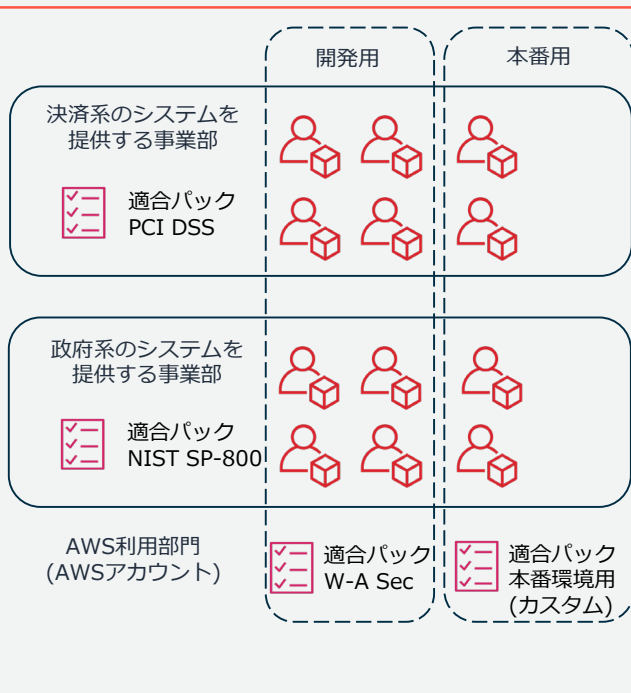
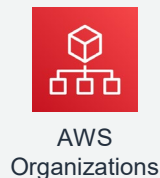
テンプレートファイルをアップロード

YAML 形式のファイル

MyCustomCPack\_Org.yaml

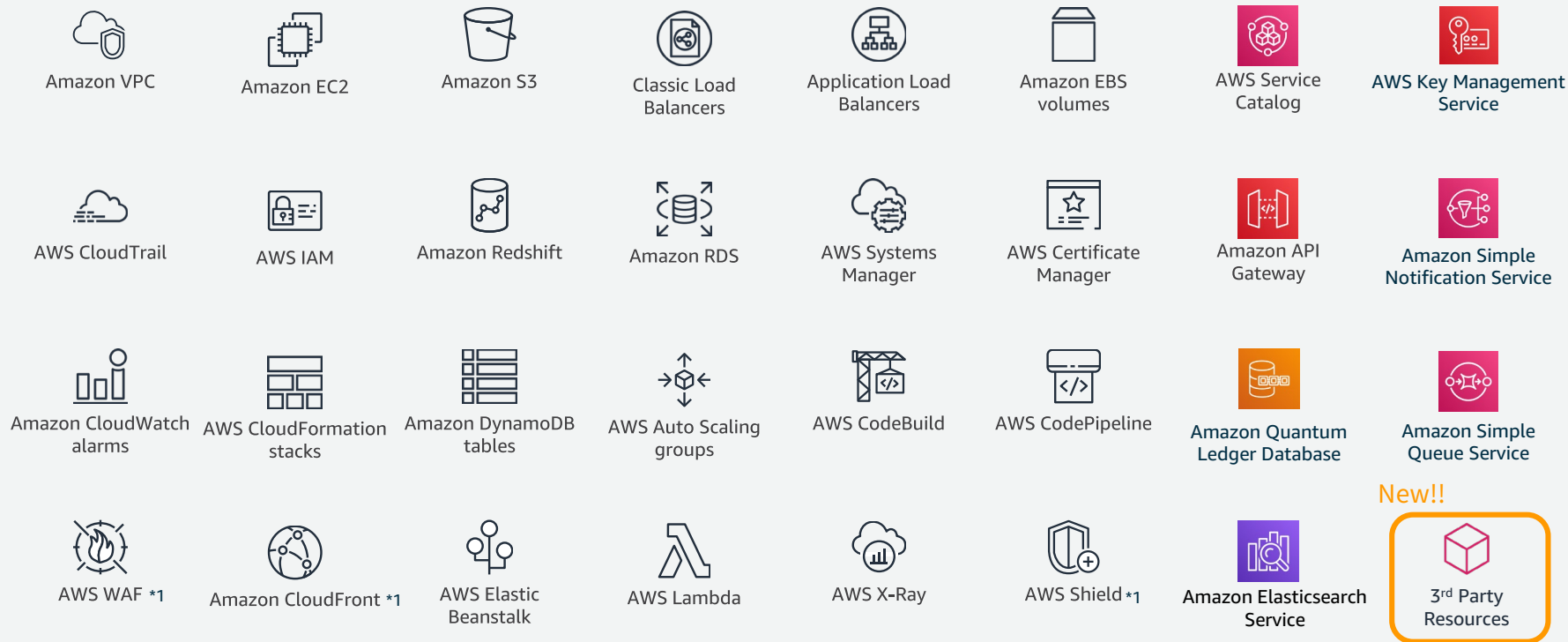
キャンセル 次へ

# 適合パックによるアカウント特性や組織特性に応じた評価



# サードパーティリソースへの対応

# AWS Config が対応しているリソース



\*1: グローバルサービスは米国東部（バージニア北部）リージョンでサポート

[https://docs.aws.amazon.com/ja\\_jp/config/latest/developerguide/resource-config-reference.html](https://docs.aws.amazon.com/ja_jp/config/latest/developerguide/resource-config-reference.html)

# サードパーティリソース例： WordPress の構成情報を Config で管理

The screenshot shows the AWS Config console interface. On the left is a navigation sidebar with options like 'ダッシュボード', 'ルール', 'リソース', and 'アグリゲータ'. The main content area is titled 'リソースのインベントリ' (Resources Inventory). Below the title is a search and filter section with dropdown menus for 'リソースカテゴリ' (set to 'すべてのリソース') and 'リソースタイプ' (set to 'すべてのリソースタイプ'). A table below lists resources with columns for 'リソース識別子' (Resource ID), 'タイプ' (Type), and 'コンプライアンス' (Compliance). Two rows are highlighted with an orange box: 'mywordpress-01' and 'resource-001', both with the type 'Testing WordPress'.

リソース識別子	タイプ	コンプライアンス
○ mywordpress-01	Testing WordPress	-
○ resource-001	Testing WordPress	-
○ i-03b7[redacted]7bcd	EC2 Instance	-
○ arn:aws:acm:ap-northeast-1:...	ACM Certificate	-
○ subnet-Of[redacted]:7	EC2 Subnet	-



# サードパーティリソースの設定や履歴を Config で管理

AWS Config > リソース > mywordpress-01

## mywordpress-01

リソースタイムライン [🔗](#)

▶ 詳細

▼ 設定項目 (JSON) の表示

```
{
  "version": "1.3",
  "accountId": "274251360022",
  "configurationItemCaptureTime": "2020-11-23T13:37:54.624Z",
  "configurationItemStatus": "OK",
  "configurationStateId": "1606138674624",
  "configurationItemMD5Hash": "",
  "resourceType": "MyCustomNamespace::Testing::WordPress",
  "resourceId": "mywordpress-01",
  "awsRegion": "ap-northeast-1",
  "tags": {},
  "relatedEvents": [],
  "relationships": [],
  "configuration": {
    "InstanceId": "i-03b[redacted]67bcd",
    "PublicIp": "3.[redacted].14.57",
    "SubnetId": "subnet-84[redacted]3df",
    "Name": "MyCustomResourceWordPress"
  },
  "supplementaryConfiguration": {},
  "resourceTransitionStatus": "None"
}
```

設定の詳細情報

設定タイムライン

コンプライアンスタイムライン



23 11月 2020

10:16:17 午後

23 11月 2020

10:37:54 午後

3 変更

▶ 構成の詳細

## 設定タイムラインで、設定状態の履歴も確認可能

▼ 変更 3

設定変更 3

フィールド	開始	終了
Configuration.SubnetId	"subnet-ala[redacted]ala"	"subnet-8[redacted]3df"
Configuration.PublicIp		"3[redacted].14.57"
Configuration.InstanceId		"i-03b79daf[redacted]67bcd"

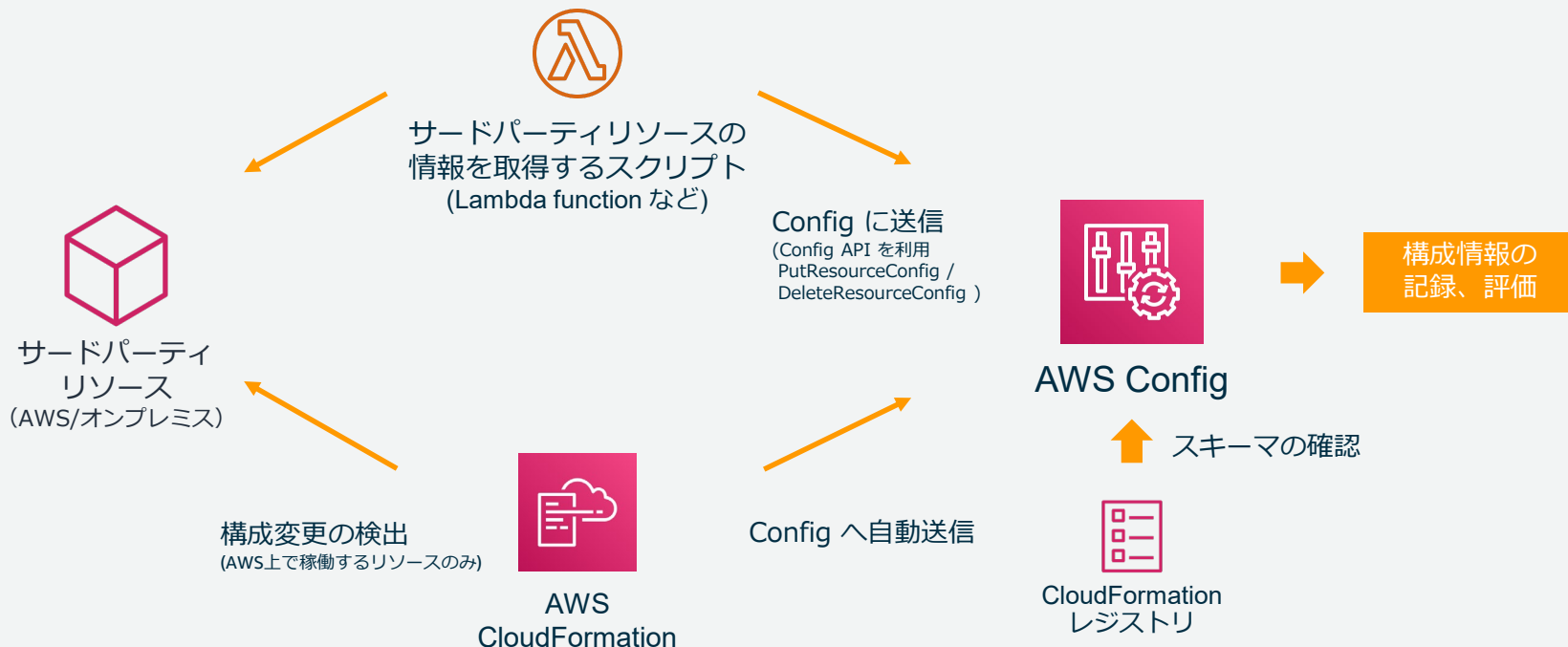
# 仕組み : CloudFormation リソースプロバイダ

CloudFormation レジストリにサードパーティリソースのスキーマを登録  
(スキーマをJSON定義して、cloudformation-cli で submit )

The screenshot shows the AWS CloudFormation console interface. On the left is a navigation sidebar with options like 'スタック', 'StackSets', 'エクスポート', 'デザイナー', 'CloudFormation レジストリ', and 'フィードバック'. The main content area is titled 'リソースタイプ' (Resource Type) and shows a list of resource types under the heading 'リソースタイプ (1)'. A dropdown menu is set to 'プライベート'. A resource type named 'MyCustomNamespace::Testing::WordPress' is highlighted with an orange box. An orange arrow points from this resource type to the 'スキーマ' (Schema) panel on the right. The schema panel displays a JSON definition for the resource type.

```
{
  "typeName": "MyCustomNamespace::Testing::WordPress",
  "description": "An example resource that creates a website based on WordPress 5.2.2.",
  "sourceUrl": "https://github.com/aws-cloudformation/aws-cloudformation-rpdk.git",
  "properties": {
    "Name": {
      "description": "A name associated with the website.",
      "type": "string",
      "pattern": "^[a-zA-Z0-9]{1,219}\\Z",
      "minLength": 1, "maxLength": 219
    },
    "SubnetId": {
      "description": "A subnet in which to host the website.",
      "pattern": "^(subnet-[a-f0-9]{13})|(subnet-[a-f0-9]{8})\\Z",
      "type": "string"
    },
    "InstanceId": {
      "description": "The ID of the instance that backs the WordPress site.",
      "type": "string"
    },
    "PublicIp": {
      "description": "The public IP for the WordPress site.",
      "type": "string"
    }
  },
  "required": [ "Name", "SubnetId" ],
  "primaryIdentifier": [ "/properties/PublicIp", "/properties/InstanceId" ],
  "readOnlyProperties": [ "/properties/PublicIp", "/properties/InstanceId" ],
  "additionalProperties": false
}
```

# サードパーティリソースの構成情報の流れ



# ご参考：

# CloudFormation レジストリ、リソースプロバイダの詳細情報

[AWS Black Belt Online Seminar] AWS CloudFormation deep dive 資料及び QA 公開

<https://aws.amazon.com/jp/blogs/news/webinar-bb-aws-cloudformation-deep-dive-2020/>

The screenshot shows the AWS Japan blog page for the article "[AWS Black Belt Online Seminar] AWS CloudFormation deep dive 資料及び QA 公開". The page includes the AWS logo, navigation tabs for "ブログホーム", "カテゴリ", and "エディション", and a search bar. The main content area features the article title, author "by AWS Japan Staff", and date "on 13 OCT 2020". Below the title, there is a section for "先日 (2020/10/06) 開催しました AWS Black Belt Online Seminar 「AWS CloudFormation deep dive」の資料を公開しました。" and a video player for the seminar recording. The video player has a play button and a title "[AWS Black Belt Online Seminar] AWS CloudFormation deep dive".

The screenshot shows the AWS CloudFormation Registry console. The main heading is "CloudFormationレジストリ". There are two main sections: "独自に作成したCFnリソース定義を登録する" and "リソースプロバイダスキーマ". The first section lists: "3rd PartyリソースがCFnで管理できる", "パブリック (AWSのネイティブ) リソースも移行中", and "既存のテンプレートやスタックは変更不要" (現在519 (東京リージョン)). The second section lists: "リソースの設計書に相当するスキーマ", "設定可能なプロパティなどを定義する", and "マネジメン". To the right, there is a small screenshot of the console interface showing a resource type being registered.

The diagram is titled "リソースプロバイダ実装の流れ" (Flow of Resource Provider Implementation). It lists the following steps: 1. スキーマを定義する (Define schema), 2. ハンドラを実装する (Implement handler), 3. ビルドする (Build), 4. テストする (Test), 5. レジストリに登録する (Register to registry), 6. CFnで利用する (Use in CFn). Below the steps, it says "実装例 (サンプル)" (Implementation example (sample)) and lists "Unicom Maker" with links to its GitHub repository and documentation.

The screenshot shows terminal output for "Pythonによる実装例 1" (Python-based implementation example 1). It is divided into two main sections: "1. 事前準備" (Preparation) and "2. コード、CFn CLIとプラグイン" (Code, CFn CLI and plugins). The preparation section lists: "Cloud9環境 (Ubuntu)", "Python 3.7 (RPDK設定と合わせる)", "AWS CLI", and "AWS SAM CLI". The code section lists: "unicorn-makerのclone", "venvの設定", and "cloudformation-cli\*の導入 (バージョン整合注意)". The terminal output shows commands like "sudo apt install python3", "python3 -m venv .env", "pip install cloudformation-cli-python-plugin", and "cfm --version".

# その他のアップデート

組織のセキュリティ管理をより効率化する機能拡張

# アドバンスドクエリ： マルチアカウント、マルチリージョンの検索に対応

AWS Config > 高度なクエリ > クエリエディタ

## クエリエディタ

次の SQL クエリエディタを使用して、AWS リソースの設定を照会します。プロパティとそのデータ型のリストは [GitHub](#) にあります。クエリ スコープを選択して、この AWS アカウントまたは複数のアカウントとリージョンに対してデータをクエリします。 [詳細はこちら](#)

### クエリスコープ

アグリゲータを選択して、このアカウントとリージョン、または複数のアカウントとリージョンに対してクエリを実行するクエリ範囲を定義します。

skorg-config-aggregator ▲

Q

このアカウントとリージョンのみ

skorg-config-aggregator

### Count EC2 Instances

クエリスコープ: skorg-config-aggregator

```
1 SELECT
2   configuration.instanceType,
3   COUNT(*)
4 WHERE
5   resourceType = 'AWS::EC2::Instance'
6 GROUP BY
7   configuration.instanceType
```

実行

クリア

実行 (Ctrl+Enter)

### 出力

名前を付けてエクスポート ▼

< 1 > ⚙

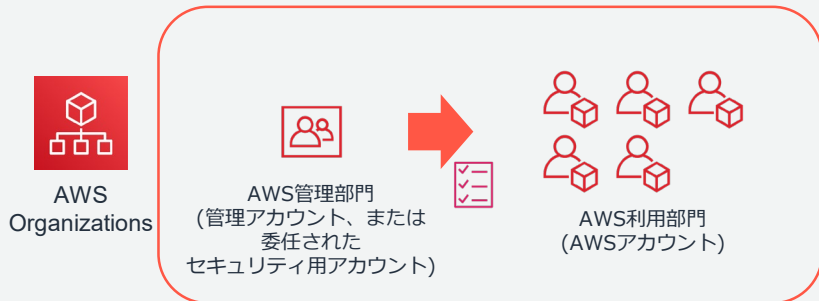
configuration.instanceType	COUNT(*)
t2.micro	3
t3.small	2
c4.xlarge	1
c5.2xlarge	1
c5.large	

アグリゲータが選択可能に

複数のアカウント、リージョンを  
横ぐしで検索して状況を確認

# AWS Organizations 連携 :

## AWS Organizations で委任管理者のサポートと Config Rule/ 適合パックの一括配布



## 評価結果をまとめて確認



```
$aws organizations enable-aws-service-access --service-principal=config-multiaccountsetup.amazonaws.com
```

```
$aws configservice put-organization-config-rule ¥  
--organization-config-rule-name my-cloudtrail-enabled ¥  
--organization-managed-rule-metadata ¥  
(snip) RuleIdentifier="CLOUD_TRAIL_ENABLED"
```

```
$aws configservice put-organization-conformance-pack ¥  
--organization-conformance-pack-name="CISPack" ¥  
--template-body="file://CISConformancePack.yaml" ¥  
--delivery-s3-bucket="{awsconfigconforms-your-bucket}"
```

集約ビュー > ルール

### ルール

ルールは、必要な構成設定を表します。AWS Config は、リソース設定が該当するルールに準拠しているかどうかを評価し、結果の概要を次の

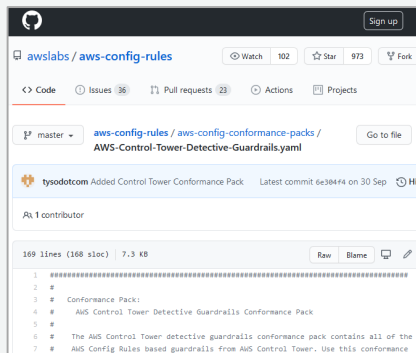
アグリゲータ	コンプライアンス状況	リージョン	アカウント
skorg-config-aggregator	すべて	すべてのリージョン	すべてのアカウント
ルール名	コンプライアンス	リージョン	アカウント
OrgConfigRule-org-cloudtrail-en...	1 非準拠リソース	ap-northeast-1	27...022
OrgConfigRule-org-s3-bucket-s...	4 非準拠リソース	ap-northeast-1	27...022

# マルチアカウント環境へのガードレールの適用

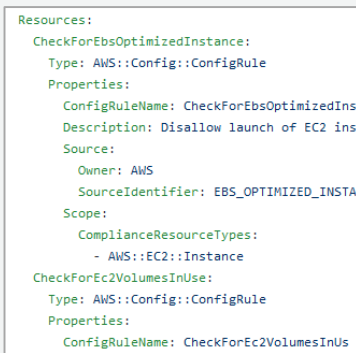
## 適合パック : AWS Control Tower Detective Guardrails

- AWS Control Tower のガードレールに含まれる、Config Rule のパッケージ
- AWS Control Tower の機能を切り出して利用
  - Control Tower の発見的ガードレールを既存 AWS アカウントに適用
  - Control Tower がサポートしていない AWS リージョンのセキュリティ統制
  - 組織のセキュリティポリシーに沿って、ガードレールをカスタマイズして利用

1. GitHubからテンプレートをダウンロード



2. 修正してカスタムテンプレート化



3. 組織にセキュリティポリシーに沿った評価

名前	修繕アクション	コンプライアンス
CheckForEc2VolumesInUse-conformance-pack-bty6nzs2	設定されていません	🟢 準拠
CheckForS3VersioningEnabled-conformance-pack-bty6nzs2	設定されていません	🔴 非準拠
CheckForRootMfa-conformance-pack-bty6nzs2	設定されていません	🔴 非準拠
CheckForRestrictedSshPolicy-conformance-pack-bty6nzs2	設定されていません	🔴 非準拠
CheckForRdsStorageEncryption-conformance-pack-bty6nzs2	設定されていません	🟢 準拠
CheckForRestrictedCommonPortsPolicy-conformance-pack-bty6nzs2	設定されていません	🔴 非準拠
CheckForEncryptedVolumes-conformance-pack-bty6nzs2	設定されていません	🔴 非準拠
CheckForRdsPublicAccess-conformance-pack-bty6nzs2	設定されていません	🟢 準拠
CheckForS3PublicWrite-conformance-pack-bty6nzs2	設定されていません	🟢 準拠
CheckForS3PublicRead-conformance-pack-bty6nzs2	設定されていません	🟢 準拠

Landing Zone の手動適用がより簡単に



# 適合パック : AWS Control Tower Detective Guardrails

推奨/選択的	Config Rule	内容
強く推奨	ebs-optimized-instance	Amazon EBS 最適化以外のタイプの Amazon EC2 インスタンスを禁止
強く推奨	ec2-volume-inuse-check	Amazon EC2 インスタンスにアタッチされていない Amazon EBS ボリュームを禁止
強く推奨	encrypted-volumes	Amazon EC2 インスタンスにアタッチされた Amazon EBS ボリュームの暗号化を有効
強く推奨	rds-instance-public-access-check	Amazon RDS データベースインスタンスへのパブリックアクセスを禁止
強く推奨	rds-snapshots-public-prohibited	Amazon RDS データベーススナップショットへのパブリックアクセスを禁止
強く推奨	rds-storage-encrypted	ストレージが暗号化されていない Amazon RDS データベースインスタンスを禁止
強く推奨	restricted-common-ports	RDP を介したインターネット接続を禁止
強く推奨	restricted-ssh	SSH を介したインターネット接続を禁止
強く推奨	root-account-mfa-enabled	root ユーザーに対して MFA を有効
強く推奨	s3-bucket-public-read-prohibited	Amazon S3 バケットへのパブリック読み取りアクセスを禁止
強く推奨	s3-bucket-public-write-prohibited	Amazon S3 バケットへのパブリック書き込みアクセスを禁止
選択的	s3-bucket-versioning-enabled	バージョンングが有効になっていない Amazon S3 バケットを禁止
選択的	iam-user-mfa-enabled	MFA なしの IAM ユーザーへのアクセスを禁止
選択的	mfa-enabled-for-iam-console-access	MFA なしの IAM ユーザーへのコンソールアクセスを禁止

# 本日のまとめ

1. AWS Config のおさらい
  - AWSリソースの構成管理、評価を行うマネージドサービス
  - Config でリソースの構成記録、Config Rules で構成評価
2. 新機能：適合パックの概要
  - Config Rules を用途に応じてパッケージ化
  - 組織のセキュリティ管理やコンプライアンス準拠がより簡単に！
3. 新機能：サードパーティリソースサポートの概要
  - AWSリソース以外にも対象に (WordPressなど)
4. 組織のセキュリティ管理をより効率化するその他のアップデート
  - マルチアカウント環境での展開、ルール管理、評価、レポートニング
  - 組織へのガードレールの適用、カスタマイズ性の向上

**利用シーンが広がった AWS Config を有効活用して  
AWS の利用をより安全・快適に！**

# Q&A

ご質問については

AWS Japan Blog 「<https://aws.amazon.com/jp/blogs/news/>」にて

後日掲載します。

# AWS の日本語資料の場所「AWS 資料」で検索



The screenshot shows the AWS Japanese website header with the logo, navigation links for '日本語', 'アカウント', and 'サポート', and a 'サインイン' button. The main content area features the title 'AWS クラウドサービス活用資料集トップ' and a paragraph describing the resource collection. Below the text are four buttons: 'AWS Webinar お申込', 'AWS 初心者向け', '業種・ソリューション別資料', and 'サービス別資料'.

aws

日本担当チームへお問い合わせ サポート 日本語 ▼ アカウント ▼ [コンソールにサインイン](#)

製品 ソリューション 料金 ドキュメント 学習 パートナー AWS Marketplace その他 🔍

## AWS クラウドサービス活用資料集トップ

アマゾン ウェブ サービス (AWS) は安全なクラウドサービスプラットフォームで、ビジネスのスケールと成長をサポートする処理能力、データベースストレージ、およびその他多種多様な機能を提供します。お客様は必要なサービスを選択し、必要な分だけご利用いただけます。それらを活用するために役立つ日本語資料、動画コンテンツを多数ご提供しております。(本サイトは主に、AWS Webinar で使用した資料およびオンデマンドセミナー情報を掲載しています。)

[AWS Webinar お申込 »](#) [AWS 初心者向け »](#) [業種・ソリューション別資料 »](#) [サービス別資料 »](#)

<https://amzn.to/JPArchive>

# ご視聴ありがとうございました

AWS 公式 Webinar

<https://amzn.to/JPWebinar>



過去資料

<https://amzn.to/JPArchive>

