

サーバーレスAWSのセキュリティ 設定チェック40観点

Lambda ・ API Gateway ・ DynamoDB ・ SQS ・ EventBridge ・ S3 ・ Step Functions を運用する会社のために

発行 株式会社Forest Book (Forest Book Inc.)

Forest Security (株式会社Forest Book)

2026年10月 版1.0

本資料の複製・社内配布は自由。再配布時は出典を明記のこと。

この資料について

項目	内容
題名	サーバーレスAWSのセキュリティ設定チェック40観点
副題	Lambda・API Gateway・DynamoDB・SQS・EventBridge・S3・Step Functions を運用する会社のために
発行	株式会社Forest Book（Forest Book Inc.）Forest Security
版	1.0（2026年10月）
想定読者	Webサービス／ECの外部連携やカスタムアプリをAWS サーバーレスで運用する中小企業・開発会社の開発責任者、テックリード、情報システム担当
想定構成	API Gateway、Lambda、DynamoDB、SQS、EventBridge、S3、Step Functions を1つのAWS アカウントで運用し、デプロイはSAM / CDK / Serverless Framework 等の IaC と CI で行っている構成
一次情報の確認日	2026年10月22日。本文中のAWS 公式ドキュメントは同日に直接確認した。出典一覧は末尾
本資料が示さないこと	40観点を満たせば脆弱性が存在しないという判断。観点は設定の確認に限り、コードの脆弱性診断や動的検査は含まない

この資料の使い方

CIS AWS Foundations Benchmark や AWS Security Hub の標準は、アカウント全体の基本設定（ルートユーザー、CloudTrail、パスワードポリシー、公開されたリソース）を広く点検するためのものである。サーバーレス構成を運用する会社がそれらを有効にすると、数百件の検出結果のうちサーバーレスに直接関係するものは一部で、肝心の「Lambda の実行ロールが広すぎる」「Webhook 受信用の関数 URL に認証がない」「Step Functions の実行履歴に個人情報が残る」といった構成固有の問題は、検出結果の中に埋もれるか、そもそもコントロールが存在しない。

本資料は、その隙間を埋めるために、サーバーレス構成に固有の確認観点を40に絞って並べたものである。各観点到「確認すること」「なぜ危ないか」「確認方法」「推奨設定」「出典」を付け、確認方法はAWS CLI（読み取り系のみ）またはコンソールの操作で書いた。Security Hub の該当コントロール ID がある観点は、出典一覧に併記した。

使い方は3通りある。

立場	使い方
自社のAWSを自分で点検する	40観点を上から順に実行し、CLI 出力を日付付きで保存する。「該当なし」も理由とともに記録する
委託先が構築した環境の状態を知りたい	委託先に40観点の記入を依頼し、出力の提出を求める。読めない場合は本資料の「なぜ危ないか」の列で判断する
作った人がいなくなった環境を引き継いだ	観点01～05（IAM）と32～35（ログ・監視）から始め、現状の把握を優先する

本資料は Security Hub や CIS の代わりではない。それらが扱うアカウント基盤の設定は観点01～03・32～33で最小限触れるにとどめ、残りをサーバーレス固有の観点に使っている。

サーバーレス構成で事故が起きる典型パターン4つ

いずれも架空の構成であり、実在の顧客・案件とは関係がない。Forest Book が自社検証環境で再現できる範囲の設定に限り書いている。

パターン	構成	起きること	設定上の原因	関係する観点
1 認証なしの受信口	EC プラットフォームからの注文 Webhook を Lambda 関数 URL (AuthType: NONE) で受け、在庫 DynamoDB を更新する	関数 URL を推測または漏えいした第三者が、偽の注文イベントを送り込み、在庫数と出荷指示が書き換わる	関数側で署名 (HMAC) 検証を実装していない。関数 URL が意図せず公開のまま。スロットリングもないため大量送信で課金も増える	06、07、10、12、13
2 共有実行ロール	15個の Lambda がすべて1つの実行ロールを使い、ロールには <code>dynamodb:*</code> ・ <code>s3:*</code> ・ <code>Resource: "*"</code> が付いている	画像変換 Lambda の依存パッケージの脆弱性を突かれ、その関数の権限で顧客テーブルと受注 CSV のバケットが丸ごと読み出される	関数ごとに権限を分けていない。読むだけでよい関数に書き込み・削除権限がある。CloudTrail のデータイベントがなく、読み出しの事実が後から追えない	04、05、09、20、32
3 平文の秘密情報	外部 API のキーと EC プラットフォームのアクセストークンを Lambda の環境変数に平文で置き、開発者は長期アクセスキーで作業している	開発者のアクセスキーが個人リポジトリに混入し、 <code>GetFunctionConfiguration</code> で環境変数を読まれる。外部 API が第三者に使われ、月末の請求で気づく	秘密情報を Secrets Manager / SSM に置いていない。人のアクセスに長期キーを使っている。GuardDuty と Budgets のアラートがない	02、08、33、35、36、39
4 止まる非同期処理	決済確定イベントを EventBridge → SQS → Lambda → Step Functions で処理し、会計システムへ連携する	会計 API の一時障害で Step Functions が失敗し、SQS にも DLQ がいないため、確定済みの決済 200件が会計に届かないまま消える。調査のためログを開くと、実行履歴に氏名とカード下4桁が無期限に残っている	Retry / Catch と DLQ がない。Step Functions のログに実行データを含めていない。CloudWatch Logs の保持期間が未設定	22、24、30、31、34

4つに共通するのは、個々のサービスの設定はどれも「初期値のまま」または「動かすために開けた」状態であり、誰かが悪意をもって弱くしたわけではない点である。サーバーレスは構成要素が多く、1つ1つの設定の既定値が安全側とは限らない。だから設定を観点として列挙して確かめる。

40観点のチェック表

表の読み方：「確認方法」の CLI は読み取り系のみで、`<...>` は環境に合わせて置き換える。`--region` は運用リージョンを指定し、IAM と CloudTrail はグローバルまたは全リージョンで確認する。「出典」の略称は末尾の出典一覧に対応する。Security Hub のコントロール ID は、該当するものがある観点にのみ出典一覧で示した。

IAM・実行ロール

No.	確認すること	なぜ危ないか	確認方法 (CLI または コンソール)	推奨設定	出典
01	ルートユーザーに MFA があり、アクセスキーが存在しないこと	ルートはすべてのリソースと請求に到達できる。キーが漏えいするとアカウント全体を失う	<code>aws iam get-account-summary --query 'SummaryMap.[AccountMFAEnabled,AccountAccessKeysPresent]'</code> (期待値 1 と 0)	ルートはハードウェア MFA を設定し、日常作業に使わない。キーは削除	IAM-BP
02	人が使う IAM ユーザーの長期アクセスキーの有無、最終使用日、作成からの経過日数	長期キーはリポジトリ・端末・チャットに残りやすく、漏えいしても気づきにくい	<code>aws iam generate-credential-report</code> の後 <code>aws iam get-credential-report --query Content --output text base64 -d</code> で <code>access_key_1_last_used_date</code> 、 <code>access_key_1_last_rotated</code> 列を確認	人は IAM Identity Center 等の ID 連携と一時認証情報を使う。残す場合は90日以内に更新し、未使用は削除	IAM-BP、IAM-CR
03	"Action": "*" と "Resource": "*" を同時に許すカスタマー管理ポリシー、IAM ユーザーへの直接アタッチ	管理者相当の権限が複数の経路に散らばり、棚卸しできなくなる	<code>aws iam list-policies --scope Local --only-attached --query 'Policies[.Arn]'</code> の各 ARN について <code>aws iam get-policy-version</code> で既定バージョンの本文を確認。 <code>aws iam list-attached-user-policies --user-name <name></code>	管理者権限はグループまたはロール経由に限定。ユーザーへの直接アタッチはなくす	IAM-BP
04	Lambda 関数ごとに実行ロールが分かれ、各ロールが実際に呼ぶ API とリソースに限定されていること	1関数の脆弱性が、その関数が触る必要のないテーブル・パケットへの到達経路になる (パターン2)	<code>aws lambda list-functions --query 'Functions[.FunctionName,Role]'</code> --output table でロールの共有を確認。各ロールで <code>aws iam list-attached-role-policies</code> と <code>aws iam list-role-policies</code> → <code>aws iam get-role-policy</code>	1関数1ロール。dynamodb:*・s3:*・Resource: "*" を使わず、テーブル ARN・バケット ARN 単位で許可。IAM Access Analyzer のポリシー生成を補助に使う	L-ROLE、IAM-BP
05	IAM Access Analyzer (外部アクセス) が有効で、アクティブな検出結果が放置されていないこと	ロール、S3、SQS、Lambda、KMS、Secrets Manager の外部アカウント・公開への開放を、個別に目視するのは現実的でない	<code>aws accessanalyzer list-analyzers --type ACCOUNT</code> → <code>aws accessanalyzer list-findings-v2 --analyzer-arn <arn> --filter '{"status":{"eq":["ACTIVE"]}}'</code>	アカウント単位で有効化 (外部アクセスアナライザーは追加料金なし)。検出結果は「意図した／意図しない」を判定して記録	AA、AA-PRICE

Lambda

No.	確認すること	なぜ危ないか	確認方法 (CLI または コンソール)	推奨設定	出典
06	AuthType: NONE の関数 URL の有無。NONE が必要な場合、関数内で署名検証をしていること	NONE では Lambda は呼び出し前に認証を行わない。URL を知る誰でも実行できる (パターン1)	<code>for f in \$(aws lambda list-functions --query 'Functions[.FunctionName]' --output text); do aws lambda get-function-url-config --function-name \$f 2>/dev/null && echo \$f; done</code>	原則 AWS_IAM。Webhook 受信で NONE が必要なら、HMAC 検証・タイムスタンプ検証・予約済み同時実行数を組み合わせる	L-URL
07	関数のリソーススペースポリシーに Principal: "*" や想定外アカウントの <code>lambda:InvokeFunction</code> がないこと	他アカウントや匿名からの呼び出しを許すと、関数の実行ロールの権限をそのまま使われる	<code>aws lambda get-policy --function-name <name> --query Policy --output text jq .</code>	呼び出し元を特定のサービス・ARN・aws:SourceArn 条件に限定	L-RBP

No.	確認すること	なぜ危ないか	確認方法 (CLI または コンソール)	推奨設定	出典
08	環境変数にアクセストークン、API キー、DB パスワードが平文で置かれていないこと	保存時は KMS で暗号化されるが、GetFunctionConfiguration 権限があれば平文で読める。コンソールでも表示される (パターン3)	<code>aws lambda get-function-configuration --function-name <name> --query 'Environment.Variables' jq 'keys'</code> (値は出力せずキー名だけ記録)	秘密情報は Secrets Manager または SSM Parameter Store (SecureString) に置き、実行時に取得。環境変数には参照名のみ	L-ENV
09	サポート終了 (非推奨) ランタイムの関数がないこと	非推奨ランタイムにはセキュリティパッチが提供されず、依存パッケージも更新できなくなる	<code>aws lambda list-functions --query 'Functions[][FunctionName,Runtime]' --output table</code> の結果を公式の非推奨日表と突合	サポート中のランタイムへ移行し、laC でランタイムを固定して定期確認	L-RT
10	インターネットから呼ばれる関数に予約済み同時実行数が設定されていること	大量呼び出しで同時実行数がアカウント上限まで使われ、他の関数が止まる。課金も増える	<code>aws lambda get-function-concurrency --function-name <name></code>	公開面の関数に上限を設定し、下流 (DB、外部 API) の処理能力に合わせる	L-CONC
11	非同期呼び出し (EventBridge、S3 イベント等) の失敗先 (On-failure 送信先または DLQ) があること	再試行後に失敗したイベントは記録先がないと破棄され、処理漏れに気づけない (パターン4)	<code>aws lambda get-function-event-invoke-config --function-name <name></code> 、 <code>aws lambda get-function-configuration --function-name <name> --query DeadLetterConfig</code>	On-failure 送信先 (SQS 等) を設定。送信先キューの滞留をアラーム監視	L-ASYNC

API Gateway

No.	確認すること	なぜ危ないか	確認方法 (CLI または コンソール)	推奨設定	出典
12	各ルート・メソッドの認可タイプ。NONE のものが意図したもの (Webhook 受信など、アプリ側で署名検証するもの) だけであること	認可なしのルートは、裏の Lambda と実行ロールの権限を匿名に開く	REST: <code>aws apigateway get-resources --rest-api-id <id> --embed-methods --query 'items[].{path:path,methods:resourceMethods}'</code> 。HTTP: <code>aws apigatewayv2 get-routes --api-id <id> --query 'Items[][RouteKey,AuthorizationType,AuthorizationId]' --output table</code>	JWT オーソライザー、IAM 認可、Lambda オーソライザーのいずれかを既定にし、NONE は一覧表で管理	APIGW-AUTH
13	ステージまたはルート単位のスロットリングと使用量プラン	上限がないと、1つの呼び出し元が API 全体の性能と課金を支配する	REST: <code>aws apigateway get-stage --rest-api-id <id> --stage-name <stage> --query methodSettings</code> 。HTTP: <code>aws apigatewayv2 get-stage --api-id <id> --stage-name <stage> --query '[DefaultRouteSettings,RouteSettings]'</code>	ステージ既定と、公開ルートの個別上限を設定。API キー配布先には使用量プラン	APIGW-THR
14	ステージのアクセスログが有効で、送信元 IP、ルート、ステータス、requestId を含むこと	ログがないと、不正な呼び出しの有無も、いつから起きていたかも分からない	REST: <code>... get-stage ... --query accessLogSettings</code> 。HTTP: <code>... get-stage ... --query AccessLogSettings</code>	JSON 形式で CloudWatch Logs に出力し、保持期間を設定 (観点34)	SH-REF
15	公開 REST API に WAF Web ACL が関連付けられているか。ない場合、スロットリングとアプリ側検証で代替しているか	既知の攻撃パターンや特定 IP からの大量アクセスを、アプリに届く前に止める手段がない	<code>aws wafv2 list-web-acls --scope REGIONAL</code> 、 <code>aws wafv2 get-web-acl-for-resource --resource-arn arn:aws:apigateway:<region>::/restapis/<id>/stages/<stage></code>	WAF は要件ではなく選択肢。公開面の広さと予算で決め、使わない判断も記録する。HTTP API は直接関連付けの対象外のため「該当なし」になりうる	APIGW-WAF

No.	確認すること	なぜ危ないか	確認方法（CLI またはコンソール）	推奨設定	出典
16	CORS の許可オリジンが * でなく、固定値または許可リストであること	* と認証情報付きリクエストの組み合わせは、第三者サイトからの API 呼び出しを許す	HTTP : <code>aws apigatewayv2 get-api --api-id <id> --query CorsConfiguration</code> 。REST は OPTIONS メソッドの統合レスポンス、または Lambda 側のヘッダー設定	許可オリジンを自社ドメインに限定。ステージングで <code>curl -H "Origin: https://evil.example" -I</code> の応答ヘッダーを確認	APIGW-CORS

DynamoDB

No.	確認すること	なぜ危ないか	確認方法（CLI またはコンソール）	推奨設定	出典
17	暗号化鍵の種類（AWS 所有／AWS マネージド／カスタマーマネージド）と、顧客データ・トークンを置くテーブルの鍵方針	すべてのテーブルは既定で保存時暗号化されるが、鍵の管理主体と監査可能性は種別で異なる	<code>aws dynamodb describe-table --table-name <name> --query 'Table.SSEDescription'</code> （空なら AWS 所有キーによる既定暗号化）	取引先や監査から鍵管理の説明を求められるテーブルはカスタマーマネージドキー。それ以外は既定でよい。判断を記録する	DDB-ENC
18	ポイントインタイムリカバリ（PITR）、削除保護、バックアッププランの対象	誤操作・不正なスク립ト・ランサム的な削除から戻す手段がない	<code>aws dynamodb describe-continuous-backups --table-name <name></code> 、 <code>aws dynamodb describe-table --table-name <name> --query 'Table.DeletionProtectionEnabled'</code> 、 <code>aws backup list-protected-resources</code>	本番テーブルは PITR と削除保護を有効化。復元手順を一度は実行して記録	DDB-PITR、BACKUP
19	テーブルのリソーススペースポリシーに他アカウントの許可がないこと	テーブル単位で他アカウントに直接アクセスを許せるため、IAM ロールの棚卸しだけでは見落とす	<code>aws dynamodb get-resource-policy --resource-arn <table-arn></code> （未設定ならエラー。その旨を記録）	他アカウント連携が必要な場合も、特定ロール ARN と操作に限定	DDB-COND
20	マルチテナント（複数の店舗・顧客企業）のデータを1テーブルに置く場合、パーティションキーにテナント識別子を含め、テナント境界をアプリと IAM の両方で守っていること。保持期限のある項目に TTL があること	アプリ側のバグ1つで別テナントのデータが読める。TTL がないと削除の約束を守れない	テーブル設計書とキー設計を確認。IAM 条件 <code>dynamodb:LeadingKeys</code> の使用有無。 <code>aws dynamodb describe-time-to-live --table-name <name></code>	キーの先頭にテナント ID。可能なら実行ロールの条件で <code>dynamodb:LeadingKeys</code> を絞る。保持期限のある属性に TTL	DDB-COND、DDB-TTL

SQS・EventBridge

No.	確認すること	なぜ危ないか	確認方法（CLI またはコンソール）	推奨設定	出典
21	キューの保存時暗号化と、キューポリシーに <code>Principal: "*" </code> がないこと	キューには注文・顧客情報を含むメッセージが滞留する。公開ポリシーは第三者による投入・取り出しを許す	<code>aws sqs get-queue-attributes --queue-url <url> --attribute-names SqsManagedSseEnabled KmsMasterKeyId Policy RedrivePolicy</code>	SSE-SQS または SSE-KMS を有効化。送信元は <code>aws:SourceArn</code> 条件で限定	SQS-SSE
22	主要キューに <code>RedrivePolicy</code> （DLQ）があり、DLQ の滞留をアラームで監視していること	処理に失敗したメッセージが保持期間の経過後に消え、業務データの欠落になる（パターン4）	上記の <code>RedrivePolicy</code> 。 <code>aws cloudwatch describe-alarms --query "MetricAlarms[?MetricName=='ApproximateNumberOfMessagesVisible']"</code>	<code>maxReceiveCount</code> を設定した DLQ。DLQ のメッセージ数が1以上で通知	SQS-DLQ

No.	確認すること	なぜ危ないか	確認方法（CLI またはコンソール）	推奨設定	出典
23	カスタムイベントバスのリソースポリシーが他アカウント・公開を許していないこと	偽のイベントを投入され、下流の Lambda や Step Functions がそれを正規のものとして処理する	<code>aws events describe-event-bus --name <bus> の Policy</code>	投入元を特定アカウント・ロールに限定。既定バスへの他アカウント書き込みも確認	EB-RBP
24	重要ルールターゲットに DLQ があること	ターゲットの呼び出し失敗が再試行後に破棄され、イベント欠落に気づけない	<code>aws events list-rules --event-bus-name <bus> → aws events list-targets-by-rule --rule <rule> --event-bus-name <bus> --query 'Targets[].DeadLetterConfig'</code>	業務上失ってはならないイベントのルールに SQS の DLQ を設定し、観点22と同じ監視	EB-DLQ

S3

No.	確認すること	なぜ危ないか	確認方法（CLI またはコンソール）	推奨設定	出典
25	アカウントレベルと各バケットのパブリックアクセスブロック4設定がすべて有効であること	1つのバケットポリシーの誤りで、受注 CSV や帳票が公開される	<code>aws s3control get-public-access-block --account-id <id>、for b in \$(aws s3api list-buckets --query 'Buckets[].Name' --output text); do echo \$b; aws s3api get-public-access-block --bucket \$b 2>&1; done</code>	アカウントレベルで4設定を有効化。静的サイト配信は CloudFront 経由にし、バケットは非公開のまま	S3-BPA
26	デフォルト暗号化、顧客データを置くバケットのバージョンing、 <code>aws:SecureTransport</code> による TLS 強制	平文 HTTP での取得を許すと経路上で読まれる。バージョンingがないと上書き・削除から戻せない	<code>aws s3api get-bucket-encryption --bucket 、aws s3api get-bucket-versioning --bucket 、aws s3api get-bucket-policy --bucket --query Policy --output text jq .</code>	SSE-S3 または SSE-KMS。業務データのバケットはバージョンingとライフサイクル。 <code>aws:SecureTransport: false</code> を Deny	S3-ENC、S3-VER
27	プリサインド URL の有効期限が用途に対して短く、署名に使う認証情報が限定されていること	プリサインド URL は持っている人が誰でも使えるベアトークン。長期有効な URL がメール・チャットで転送されると、取り消す手段がない	発行コードの <code>expiresIn</code> (SDK) または <code>--expires-in</code> (CLI) の値。署名に使うロールと、そのロールの S3 権限	用途に合わせ数分～数時間。発行専用のロールに対象プレフィックだけの権限。一時認証情報で署名すればセッション期限で失効する	S3-PRESIGN、S3-PRESIGN2
28	Object Ownership が「Bucket owner enforced」で ACL が無効であること	ACL が有効だと、オブジェクト単位の公開設定がバケットポリシーの外に残り、棚卸しから漏れる	<code>aws s3api get-bucket-ownership-controls --bucket </code>	新規バケットの既定である Bucket owner enforced に統一。古いバケットは ACL 依存を確認してから切り替え	S3-OWN

Step Functions

No.	確認すること	なぜ危ないか	確認方法（CLI またはコンソール）	推奨設定	出典
29	ステートマシンの IAM ロールが、呼び出す Lambda・サービス統合に限定されていること	コンソールの自動生成ロールや共用ロールに広い権限が残り、ステートマシンの定義変更だけで任意のリソースに到達できる	<code>aws stepfunctions describe-state-machine --state-machine-arn <arn> --query roleArn → 観点04と同じ手順でロールのポリシーを確認</code>	ステートマシンごとにロールを分け、 <code>lambda:InvokeFunction</code> 等を対象 ARN に限定	SFN-ROLE

No.	確認すること	なぜ危ないか	確認方法 (CLI または コンソール)	推奨設定	出典
30	Task 状態に Retry・Catch・TimeoutSeconds があること	一時障害が実行全体の失敗になり、途中で進んだ処理（決済確定、在庫引当）が宙に浮く。タイムアウトがないと実行が止まったまま残る（パターン4）	<pre>aws stepfunctions describe-state-machine --state-machine-arn <arn> --query definition --output text jq '.States[] select(.Type=="Task") {Retry,Catch,TimeoutSeconds}'</pre>	外部呼び出しの Task に指数バックオフの Retry、失敗を記録・通知する Catch、現実的な TimeoutSeconds	SFN-ERR
31	実行履歴のログ設定（ログレベル、実行データの包含）と、入力・出力に個人情報が含まれていないか	実行履歴には状態ごとの入力・出力が記録される。個人情報を含むペイロードをそのまま渡すと、ログと履歴に長期間残る（パターン4）	<pre>aws stepfunctions describe-state-machine --state-machine-arn <arn> --query loggingConfiguration (includeExecutionData の値)。定義で各状態に渡しているデータの範囲</pre>	ペイロードには ID と参照のみを載せ、本体は DynamoDB / S3 から取得。includeExecutionData は本番で false を基本にし、必要なら保持期間を短く	SFN-LOG、SFN-LOGCFG

ログ・監視

No.	確認すること	なぜ危ないか	確認方法 (CLI または コンソール)	推奨設定	出典
32	マルチリージョンの CloudTrail 証跡が管理イベント（読み取り・書き込み）を記録し、ログファイル検証が有効で、保存先バケットが非公開であること	証跡がないと、誰がいつ設定を変えたか、秘密情報を読んだかを事後に追えない（パターン2）	<pre>aws cloudtrail describe-trails --query 'trailList[].[Name,IsMultiRegionTrail,LogFileValidationEnabled,S3BucketName]' --output table、aws cloudtrail get-event-selectors --trail-name <trail></pre>	マルチリージョン証跡1本、ログファイル検証有効。顧客データを置く DynamoDB・S3 はデータイベントも検討	CT-CLI、CT-VAL
33	運用リージョンで GuardDuty が有効で、High 以上の未対応検出がないこと。Lambda Protection の有効化	漏えいした認証情報の利用や Lambda からの不審な通信は、ログを人が眺めても見つからない（パターン3）	<pre>aws guardduty list-detectors → aws guardduty get-detector --detector-id <id>、aws guardduty list-findings --detector-id <id> --finding-criteria '{"Criterion":{"severity":{"Gate":7},"service.archived":{"Eq":["false"]}}}'</pre>	有効化し、通知先を決める。検出の確認担当と期限を決める	GD
34	CloudWatch Logs の各ロググループに保持期間があり、関数ログにメール・電話・トークン・ペイロード全体が出ていないこと	既定では無期限保持。個人情報を含むログが消えずに蓄積し、閲覧権限を持つ全員が読める（パターン4）	<pre>aws logs describe-log-groups --query 'logGroups[].[logGroupName,retentionInDays]' --output table。aws logs filter-log-events --log-group-name <lg> --filter-pattern '@' --limit 20（本文はマスクして保存）。aws logs get-data-protection-policy --log-group-identifier <lg></pre>	保持期間を業務要件で設定（例：90日～1年）。console.log(event) を禁止し、構造化ログで出す項目を列挙。データ保護ポリシーでマスク	CWL-RET、CWL-MASK
35	AWS Budgets のアラートと、Lambda エラー率・API 5xx・スロットルの CloudWatch アラーム	サーバーレスは攻撃を受けると性能ではなく請求に現れる。アラームがないと月末の請求で知ることになる（パターン3）	<pre>aws budgets describe-budgets --account-id <id>、aws cloudwatch describe-alarms --query 'MetricAlarms[].[AlarmName,MetricName,Namespace]' --output table</pre>	月額予算の50/80/100%で通知。公開面の関数と API に異常検知アラーム	BUDGET

秘密情報

No.	確認すること	なぜ危ないか	確認方法（CLI または コンソール）	推奨設定	出典
36	DB・外部 API・連携先の秘密情報が Secrets Manager または SSM Parameter Store（SecureString）にあり、環境変数・コード・IaC テンプレートに書かれていないこと	コードと設定ファイルは複製されやすく、Git 履歴からは消しにくい（パターン3）	<code>aws secretsmanager list-secrets --query 'SecretList[][Name,LastAccessedDate]' --output table</code> 、 <code>aws ssm describe-parameters --parameter-filters Key=Type,Values=SecureString --query 'Parameters[][Name]'</code> （値は取得しない）。リポジトリの秘密情報スキャン	秘密情報は1か所に集約し、実行ロールに <code>secretsmanager:GetSecretValue</code> を対象 ARN 単位で許可	SM-ROT
37	シークレットのローテーション設定、リソースポリシーの範囲、未使用シークレット	漏えいに気づかない期間が長いほど被害が広がる。使われていないシークレットは管理から漏れる	<code>aws secretsmanager list-secrets --query 'SecretList[][Name,RotationEnabled,LastRotatedDate]' --output table</code> 、 <code>aws secretsmanager get-resource-policy --secret-id <name></code>	ローテーションできるもの（DB 資格情報等）は自動化。外部発行トークンのように自動化できないものは、手動更新の期限と手順を記録	SM-ROT
38	カスターマネージド KMS キーの自動ローテーションと、キーポリシーに公開・他アカウント許可がないこと	キーポリシーの誤りは、暗号化したすべてのデータの読み出しを許す	<code>aws kms list-keys</code> → <code>aws kms get-key-rotation-status --key-id <id></code> 、 <code>aws kms get-key-policy --key-id <id> --policy-name default --output text jq .</code>	自動ローテーション有効。キー管理者と利用者を分け、利用者は実行ロール単位	SH-REF

CI/CD

No.	確認すること	なぜ危ないか	確認方法（CLI または コンソール）	推奨設定	出典
39	CI からのデプロイが長期アクセスキーではなく、OIDC 連携による期限付きロールで行われ、ロールが対象スタックに限定されていること	CI に置いた長期キーは、CI 設定の閲覧権限を持つ全員と、CI サービス側の漏えい事故の影響を受ける（パターン3）	<code>aws iam list-open-id-connect-providers</code> 。CI 設定ファイル内の <code>AWS_ACCESS_KEY_ID</code> の有無。デプロイロールの信頼ポリシーの <code>sub</code> 条件（リポジトリ・ブランチ指定）	OIDC プロバイダーを登録し、信頼ポリシーでリポジトリとブランチを限定。デプロイロールの権限は対象スタックのリソースに絞る	IAM-OIDC
40	本番の設定が IaC と一致していること（コンソールでの手変更が残っていないこと）	手変更は次のデプロイで巻き戻るか、IaC に書かれていない「開いた設定」として残り続ける	CloudFormation / SAM： <code>aws cloudformation detect-stack-drift --stack-name <stack></code> → <code>aws cloudformation describe-stack-resource-drifts --stack-name <stack></code> 。CDK も同じスタックに対して実行	本番のコンソール書き込み権限を限定し、緊急変更は期限付きの昇格で行い、翌営業日までに IaC に反映。ドリフト検出を月次で実行	CFN-DRIFT

優先順位の付け方（3軸）

40観点をすべて同時に直すことはできない。Forest Security の診断では、CVSS の数値だけで並べず、次の3軸で判定している。自己点検でも同じ軸を使うと、報告書を受け取る側と話が合う。

軸	問い	高い例	低い例
攻撃の成立しやすさ	インターネットから認証なしで到達できるか。特別な前提（内部アカウント、別の脆弱性）が必要か	認証なしの関数 URL（06）、 <code>NONE</code> ルート（12）、公開バケット（25）	他の権限を得た後にのみ使える設定（19、38）

軸	問い	高い例	低い例
影響を受ける資産	個人データ、決済、アクセストークン、取引先データに届くか。読めるだけか、書き換え・削除もできるか	全テーブルに書き込める共有ロール（04）、平文トークン（08）	保持期間未設定のログ（34）のうち個人情報を含まないもの
業務影響	事業停止、課金、取引先への波及、法令上の報告義務につながるか	決済イベントの欠落（22、24、30）、同時実行数の枯渇（10）	監査性の不足（17、40）

3軸のうち2つ以上が「高い」観点を先に直す。同じ段階にある観点が複数あれば、修正規模の小さいもの（設定変更1つで済むもの）から着手し、設計変更を要するものは期限と担当を決めて別線で進める。この進め方の様式は、Forest Security の資料「診断の後が本番 — 脆弱性を直し切るための是正管理テンプレート」に台帳と記入例を載せている。

自己点検後の進め方

段階	すること	残すもの
記録	40観定の判定（実施／該当なし／未実施）と CLI 出力を、日付とアカウント ID を付けて保存する	判定表、出力ファイル
棚卸し	「未実施」と「該当なし（理由が曖昧）」を一覧にし、3軸で優先順位を付ける	優先順位付きの一覧
修正	設定変更は IaC に反映してからデプロイする。コンソールで直した場合は観点40で差分を確認する	PR、デプロイ記録
再確認	同じ CLI を再実行し、修正前と修正後の出力を並べて保存する	修正前後の出力
継続	月1回、観点02・05・09・33・35・40を再実行する。これらは時間の経過で状態が変わる	月次の判定表
第三者の確認	取引先や元請けから提出を求められた場合、自社の判定表に加えて、構築に関わっていない第三者の確認を取る	第三者の報告書

自社で構築した環境を自社だけで点検すると、「動かすために開けた設定」は見えにくい。40観定のうち、04・08・12・20・31 は構築時の判断がそのまま残りやすく、第三者の目を入れる価値が高い。

Forest Security について

Forest Security は、Webサービス／ECの外部連携・カスタムアプリを AWS で運用する中小企業・開発会社に対して、ソースコードとクラウド設定の両方からセキュリティ上の問題を確認し、直す順番を決め、直るまでの進捗を管理するサービスである。Shopify アプリと AWS サーバーレスを開発してきた知見を使う。自動スキャンの結果を転売するサービスではなく、報告書を納品して終わるサービスでもない。

商品	内容
Forest Security Review	AWS アカウント1つ（または Shopify アプリ1本）を対象に、本資料の40観点を含む設定点検を行い、指摘一覧（ID、証拠、影響、重大度、推奨修正）と60分の報告会を納める。資料・権限受領から5営業日

商品	内容
Forest Security Assessment	AWS 1アカウント+アプリ1本を対象に、AWS サーバーレス、Web/API、CI/CD、Shopify 固有の4領域を確認し、技術報告書、リスク台帳、修正チケット案を納める。是正後の再確認1回を含む
Forest Security Care	月額で、AWS 設定差分、依存パッケージ、GuardDuty 等の検出結果のレビューと是正チケットの進捗管理を行い、月次報告書を出す

確認は顧客が作成した読み取り専用のクロスアカウントロール（外部 ID 付き、期限付き）で行い、書き込み系の操作はしない。

やらないこと

24時間365日の監視、インシデント発生時の復旧作業やフォレンジック、網羅的なペネトレーションテスト、AWS や Shopify のプラットフォーム本体への検査、脆弱性が存在しないと断定すること、法令・ガイドライン・認証への適合を請け合うこと、自社が開発した資産に対する独立した第三者としての監査意見。

金額は個別見積。範囲と「含まれないもの」は提案時に書面で示す。

問い合わせ	info@forest-book.jp
サービスサイト（予定）	https://security.forest-book.jp
会社	株式会社Forest Book（Forest Book Inc.）神奈川県横浜市西区北幸二丁目10番48号 むつみビル3階

出典一覧

すべて AWS 公式ドキュメント。2026年10月22日に直接確認した。Security Hub のコントロール ID は AWS Security Hub controls reference (SH-REF) に基づく。

略称	文書	URL	関係する Security Hub コントロール
IAM-BP	Security best practices in IAM	https://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html	IAM.1、IAM.2、IAM.3、IAM.4、IAM.9
IAM-CR	Getting credential reports	https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_getting-report.html	IAM.8、IAM.22
AA	IAM Access Analyzer	https://docs.aws.amazon.com/IAM/latest/UserGuide/what-is-access-analyzer.html	IAM.28
AA-PRICE	IAM Access Analyzer pricing	https://aws.amazon.com/iam/access-analyzer/pricing/	—
IAM-OIDC	Create an OpenID Connect (OIDC) identity provider in IAM	https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_providers_create_oidc.html	—
L-ROLE	Lambda execution role	https://docs.aws.amazon.com/lambda/latest/dg/lambda-intro-execution-role.html	IAM.21
L-URL	Lambda function URL auth	https://docs.aws.amazon.com/lambda/latest/dg/urls-auth.html	—
L-RBP	Lambda resource-based policies	https://docs.aws.amazon.com/lambda/latest/dg/access-control-resource-based.html	Lambda.1
L-ENV	Lambda environment variable encryption	https://docs.aws.amazon.com/lambda/latest/dg/configuration-envvars-encryption.html	—
L-RT	Lambda runtimes	https://docs.aws.amazon.com/lambda/latest/dg/lambda-runtimes.html	Lambda.2
L-CONC	Lambda concurrency	https://docs.aws.amazon.com/lambda/latest/dg/configuration-concurrency.html	—
L-ASYNC	Capturing records of Lambda asynchronous invocations	https://docs.aws.amazon.com/lambda/latest/dg/invocation-async-retain-records.html	—
APIGW-AUTH	Control access to API Gateway	https://docs.aws.amazon.com/apigateway/latest/developerguide/apigateway-control-access-to-api.html	APIGateway.8
APIGW-THR	API Gateway throttling	https://docs.aws.amazon.com/apigateway/latest/developerguide/api-gateway-request-throttling.html	—
APIGW-WAF	API Gateway with AWS WAF	https://docs.aws.amazon.com/apigateway/latest/developerguide/apigateway-control-access-aws-waf.html	APIGateway.4
APIGW-CORS	HTTP API CORS	https://docs.aws.amazon.com/apigateway/latest/developerguide/http-api-cors.html	—

略称	文書	URL	関係する Security Hub コントロール
DDB-ENC	DynamoDB encryption at rest	https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/EncryptionAtRest.html	—
DDB-PITR	DynamoDB point-in-time recovery	https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/PointInTimeRecovery.html	DynamoDB.2、DynamoDB.4、DynamoDB.6
DDB-COND	Using IAM policy conditions for fine-grained access control	https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/specifying-conditions.html	—
DDB-TTL	Using time to live (TTL) in DynamoDB	https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/TTL.html	—
SQS-SSE	SQS server-side encryption	https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-server-side-encryption.html	SQS.1、SQS.3
SQS-DLQ	SQS dead-letter queues	https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-dead-letter-queues.html	—
EB-RBP	EventBridge resource-based policies	https://docs.aws.amazon.com/eventbridge/latest/userguide/eb-use-resource-based.html	EventBridge.3
EB-DLQ	EventBridge rule DLQ	https://docs.aws.amazon.com/eventbridge/latest/userguide/eb-rule-dlq.html	—
S3-BPA	S3 Block Public Access	https://docs.aws.amazon.com/AmazonS3/latest/userguide/access-control-block-public-access.html	S3.1、S3.8
S3-ENC	S3 default encryption	https://docs.aws.amazon.com/AmazonS3/latest/userguide/default-bucket-encryption.html	S3.5、S3.17
S3-VER	S3 versioning	https://docs.aws.amazon.com/AmazonS3/latest/userguide/Versioning.html	S3.14
S3-PRESIGN	Sharing objects with presigned URLs	https://docs.aws.amazon.com/AmazonS3/latest/userguide/ShareObjectPreSignedURL.html	—
S3-PRESIGN 2	Download and upload objects with presigned URLs	https://docs.aws.amazon.com/AmazonS3/latest/userguide/using-presigned-url.html	—
S3-OWN	Controlling ownership of objects and disabling ACLs	https://docs.aws.amazon.com/AmazonS3/latest/userguide/about-object-ownership.html	—
SFN-ROLE	Creating an IAM role for your state machine in Step Functions	https://docs.aws.amazon.com/step-functions/latest/dg/procedure-create-iam-role.html	—
SFN-ERR	Handling errors in Step Functions workflows	https://docs.aws.amazon.com/step-functions/latest/dg/concepts-error-handling.html	—
SFN-LOG	Using CloudWatch Logs to log execution history in Step Functions	https://docs.aws.amazon.com/step-functions/latest/dg/cw-logs.html	—

略称	文書	URL	関係する Security Hub コントロール
SFN-LOGCFG	LoggingConfiguration (API リファレンス)	https://docs.aws.amazon.com/step-functions/latest/apireference/API_LoggingConfiguration.html	—
CT-CLI	CloudTrail trail with CLI	https://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudtrail-create-and-update-a-trail-by-using-the-aws-cli.html	CloudTrail.1、CloudTrail.6
CT-VAL	CloudTrail log file validation (CLI)	https://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudtrail-log-file-validation-cli.html	CloudTrail.4
GD	What is GuardDuty	https://docs.aws.amazon.com/guardduty/latest/ug/what-is-guardduty.html	GuardDuty.1、GuardDuty.6
CWL-RET	CloudWatch Logs log groups (保持期間)	https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/Working-with-log-groups-and-streams.html	—
CWL-MASK	CloudWatch Logs data protection	https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/mask-sensitive-log-data.html	—
BUDGET	AWS Budgets	https://docs.aws.amazon.com/cost-management/latest/userguide/budgets-managing-costs.html	—
BACKUP	AWS Backup	https://docs.aws.amazon.com/aws-backup/latest/devguide/what-is-backup.html	—
SM-ROT	Secrets Manager rotation	https://docs.aws.amazon.com/secretsmanager/latest/userguide/rotating-secrets.html	SecretsManager.1、SecretsManager.3、SecretsManager.4
CFN-DRIFT	Detect unmanaged configuration changes with drift detection	https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/using-cfn-stack-drift.html	—
SH-REF	AWS Security Hub controls reference	https://docs.aws.amazon.com/securityhub/latest/userguide/securityhub-controls-reference.html	KMS.4、KMS.5、APIGateway.1、APIGateway.9 ほか
WA-SEC	AWS Well-Architected Framework セキュリティの柱	https://docs.aws.amazon.com/wellarchitected/latest/security-pillar/welcome.html	—
WA-SLS	Serverless Applications Lens セキュリティの柱	https://docs.aws.amazon.com/wellarchitected/latest/serverless-applications-lens/security-pillar.html	—

本資料の観点では AWS Well-Architected Framework セキュリティの柱と Serverless Applications Lens の領域に対応付けて選んだ。AWS の各設定の既定値は変わることがあるため、実施時点の公式ドキュメントで確認すること。