AWS 上のネットワークトラフィックのためのファイアウォールオートメーション

AWS 実装ガイド

2021 年 2 月

目次

概要

「AWS 上のネットワークトラフィックのためのファイアウォールオートメーション」ソリューションは、ネットワークトラフィックのフィルタリングに必要な AWS リソースを設定します。このソリューションを使用すると、数百または数千の Amazon VPC およびアカウントを 1 つの場所で検査できます。このソリューションでは、Amazon VPC 間のトラフィックを検査するための一元的な AWS Network Firewall のプロビジョニングプロセスを自動化して時間を節約できます。AWS Network Firewall、ファイアウォールポリシー、ルールグループを一元的に設定および管理することもできます。

このソリューションでは、AWS Network Firewall を使用して、ネットワークトラフィックのきめ細かな可視化と制御を実現します。これにより、イベント駆動型のログ記録を通じて、ネットワークのセグメンテーション、ドメインの Egress フィルタリング、侵入防止を行うことができます。AWS マネジメントコンソールで数回クリックするだけで、Amazon VPC 環境で AWS Network Firewall を有効化できます。AWS Network Firewall は、基盤となるインフラストラクチャのセットアップや維持を必要とせずに、ネットワークトラフィック量に応じて自動的にスケールし、高可用性の保護を提供します。このソリューションは、GitOps ワークフローを使用した AWS Network Firewall 設定に関する共同作業や変更管理にも役立ちます。

このガイドでは、AWS クラウドで「AWS Network Firewall Deployment Automations for AWS Transit Gateway」ソリューションを計画およびデプロイするためのインフラストラクチャおよび設定に関する情報を提供します。

この実装ガイドでは、「AWS 上のネットワークトラフィックのためのファイアウォールオートメーション」ソリューションをアマゾン ウェブ サービス (AWS) クラウド内にデプロイするためのアーキテクチャに関する考慮事項と設定手順について説明します。これには、セキュリティと可用性に関する AWS のベストプラクティスを使用してこのソリューションをデプロイするために必要な AWS のサービスを起動および設定する AWS CloudFormation テンプレートへのリンクが含まれています。

このガイドは、AWS クラウド内での実践的なアーキテクチャの設計経験を持つ IT アーキテクト、DevOps プロフェッショナル、テクノロジープロフェッショナル、ネットワークエンジニア、セキュリティエンジニアを対象としています。

コスト

このソリューションの実行中に使用した AWS のサービスのコストは、お客様の負担となります。2021 年 2 月時点で、このソリューションを米国東部 (バージニア北部) リージョンにおいて、2 つのアベイラビリティーゾーン、2 つのネットワークファイアウォールエンドポイント、1 日当たり 5 GB のトラフィック、デフォルト設定で実行する推定コストは、**1 か月あたり約 620.55 USD** です。これには、AWS CodePipeline、AWS CodeBuild、Amazon Simple Storage Service (Amazon S3) に対する推定請求額が含まれます。

AWS のサービス	ディメンション	総コスト (1 か月あたり)
AWS Network Firewall (エンドポイント)	2 つのエンドポイント/24 時間 (0.395 USD/エンドポイント/時間)	568.80 USD
AWS Network Firewall (データ処理)	5 GB (0.65 USD/GB)	9.75 USD
AWS Transit Gateway (VPC アタッチメント)	24 時間 (0.05 USD/時間)	36.00 USD
AWS Transit Gateway (データ処理)	10 GB (0.02 USD/GB)	6.00 USD
Amazon Code サービス (AWS CodePipeline、AWS CodeBuild、AWS CodeCommit)		AWS CodePipeline の実行回数に基づく
Amazon S3		AWS CodePipeline の実行回数と AWS Network Firewall のログアクティビティに基づく
	合計	**620.55 USD**

料金は変更される可能性があります。詳細については、このソリューションで使用する AWS の各サービスの料金表ウェブページを参照してください。

アーキテクチャの概要

このソリューションをデフォルトのパラメータを使用してデプロイすると、AWS クラウド内に次の環境が構築されます。

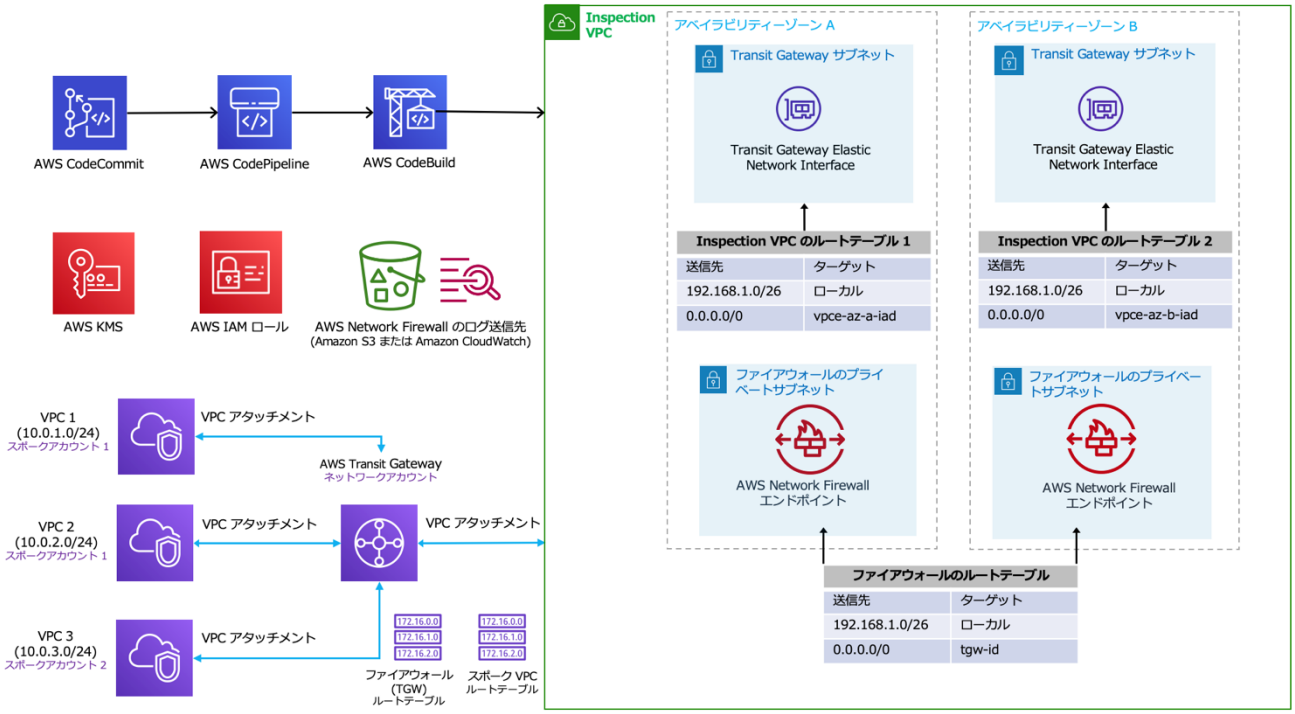

図 *1:* 「*AWS* 上のネットワークトラフィックのためのファイアウォールオートメーション」ソリューションのアーキテクチャ

AWS CloudFormation テンプレートは、ソリューションのデプロイ先である AWS リージョンのランダムに選択されたアベイラビリティーゾーンに、合計 4 つのサブネットを持つ Inspection VPC をデプロイします。既存の AWS Transit Gateway ID を指定すると、2 つのサブネットは AWS VPC の AWS Transit Gateway のアタッチメントを作成するために使用されます。他の 2 つのサブネットは、ランダムに選択された 2 つのアベイラビリティーゾーンに AWS Network Firewall エンドポイントを作成するために使用されます。このテンプレートは、新しい AWS CodeCommit リポジトリと、すべてのネットワークトラフィックを許可するデフォルトの AWS Network Firewall の設定を作成します。このテンプレートには、新しいルールグループの作成に役立つ一連の例も含まれています。設定パッケージは、AWS CodeCommit リポジトリで変更できます。これにより、AWS CodePipeline が開始され、以下のステージが実行されます。

検証ステージ - AWS Network Firewall の設定は、ドライランモードを有効にした AWS Network Firewall の API で検証されます。これにより、ユーザーは実際の変更を試みる前に、予期しない問題を見つけることができます。このステージでは、JSON ファイル構造をチェックし、設定で参照しているすべてのファイルがパッケージ内に存在するかどうかも確認します。

デプロイステージ - このステージでは、新しい AWS Network Firewall、AWS Netwoprk Firewall ポリシー、ルールグループが作成されます。リソースのいずれかが既に存在する場合、リソースは更新されます。このステージでは、変更を検出したり、AWS CodeCommit リポジトリから最新の設定を適用して修復したりすることもできます。いずれかのルールグループの変更が失敗すると、ルールグループの変更は元の状態にロールバックされます。非対称トラフィックを避けるために、TGW-VPC アタッチメントに対してアプライアンスモードがアクティブ化されます。詳細については、「Appliance in a shared services in VPC」を参照してください。

また、このソリューションでは、各アベイラビリティーゾーンに Amazon VPC ルートテーブルを作成し、AWS Network Firewall の Amazon VPC エンドポイントをターゲットとするデフォルトのルーティング先を設定します。ファイアウォールサブネットを持つ 1 つの共有ルートテーブルも作成し、AWS Transit Gateway ID をターゲットとするデフォルトのルーティング先を設定します。このルートは、AWS Transit Gateway ID が AWS CloudFormation の入力パラメータに指定されている場合にのみ作成されます。

デプロイに関する考慮事項

Amazon VPC CIDR ブロック

Amazon VPC および関連リソースの設定は、AWS CloudFormation のスタック更新ワークフローを使用して更新することはできません。VPC CIDR ブロックを更新するには、Amazon VPC を削除して再作成する必要があります。ネットワークエンジニアリングチームに連絡して、Inspection VPC 専用の CIDR ブロックを取得することをお勧めします。

AWS Network Firewall の設定

このソリューションのデプロイには、デフォルトの AWS Network Firewall のポリシーを使用するため、既存のネットワークは中断されません。これにより、カスタムネットワークファイアウォールポリシーや、ステートフル/ステートレスなルールグループを設計およびデプロイできます。これには、既存の Suricata ステートフルルールも含まれます。Suricata の詳細については、*AWS Network Firewall* デベロッパーガイドの「Stateful Suricata compatible IPS rule groups in AWS Network Firewall」を参照してください。

> **注意:** AWS Firewall Manager を使用して、このソリューションのファイアウォールルールを一元的に設定および管理することもできます。

デプロイ環境

このソリューションは同じ AWS リージョンで複数回デプロイできるため、ユーザーは既存の AWS Transit Gateway 用に新しい AWS Network Firewall および関連リソースをセットアップできます。このソリューションを AWS Transit Gateway を使用せずにデプロイし、ネットワークを変更する前にテストできます。AWS Transit Gateway ID を指定しない場合、このソリューションは TGW-VPC アタッチメントを作成しません。これにより、ネットワークエンジニアは AWS Network Firewall の設定をカスタマイズし、ネットワークを変更する前にファイアウォールポリシーを更新できます。

AWS Transit Gateway

このソリューションは、AWS Transit Gateway ID を指定した場合、既存の AWS Transit Gateway を使用して AWS VPC の AWS Transit Gateway のアタッチメントを作成します。ルートテーブル ID と AWS Transit Gateway ID を指定すると、既存の AWS Transit Gateway ルートテーブルへの関連付けと伝播も作成されます。詳細については、AWS CloudFormation のパラメータを参照してください。

AWS Transit Gateway を作成し、Amazon VPC とピアリングアタッチメントを管理するには、「サーバーレス転送ネットワークオーケストレーター」ソリューションを使用することをお勧めします。

ソリューションコンポーネント

AWS CodeCommit

このソリューションでは、デフォルトの設定と例を使用して AWS CodeCommit リポジトリを作成します。

AWS CodePipeline

AWS CodePipeline は、AWS CodeCommit リポジトリの設定パッケージの更新に基づいて変更を検証、テスト、実装します。

Amazon CloudWatch

`Log Destination for the Network Firewall` パラメータで `CloudWatchLogs` を選択すると、このソリューションはログのロググループを作成します。アラートログとフローログは、ログレコードを収集し、これらをログファイルに統合します。詳細については、AWS Network Firewall デベロッパーガイドを参照してください。

Amazon Simple Storage Service

このソリューションは、次の Amazon Simple Storage Service (Amazon S3) バケットを作成します。

ソースコードバケット - このバケットは、AWS CodeBuild ステージで使用するソースコードのバージョンをホストし、AWS Network Firewall のリソースを検証およびデプロイして、関連リソースを更新します。

AWS CodePipeline のアーティファクトバケット - このバケットは、AWS CodePipeline のステージで作成した入力アーティファクトと出力アーティファクトを保存します。AWS CodePipeline は、ステージのアクションタイプに応じて、入力アーティファクトまたは出力アーティファクトのファイルを圧縮して転送します。

AWS Network Firewall のログの送信先バケット - この Amazon S3 バケットは、`Log Destination for the Network Firewall` パラメータで `Amazon S3` を選択した場合にのみ作成されます。

Amazon VPC

このソリューションは、AWS Transit Gateway のアタッチメントと AWS Network Firewall のエンドポイントをサポートするために、4 つのサブネットを持つ Inspection VPC を作成します。

セキュリティ

AWS インフラストラクチャでシステムを構築する場合、セキュリティ上の責任はお客様と AWS の間で共有されます。この責任共有モデルにより、AWS がホストオペレーティングシステムと仮想化レイヤーからサービスが運用されている施設の物理的なセキュリティに至るまでの要素を運用、管理、および制御するため、お客様の運用上の負担を軽減するのに役立ちます。セキュリティの詳細については、AWS クラウドセキュリティを参照してください。

AWS Key Management Service

このソリューションは、AWS Key Management Service (AWS KMS) の 2 つの暗号化キーを作成します。1 つのキーは、Amazon S3 アーティファクト、ソースコードバケット、AWS CodeBuild プロジェクトでオブジェクトの暗号化に使用します。2 番目のキーは、AWS Network Firewall のログの送信先の暗号化に使用します。送信先は、`Amazon CloudWatch` または `Amazon S3` バケットのどちらを選択するかによって異なります。デフォルトでは、このソリューションでプロビジョニングした IAM ロールのみが、このキーを使用した暗号化または復号の操作を許可されます。自動キーローテーションはデフォルトで有効になっています。

AWS Identity and Access Management のロール

AWS Identity and Access Management (IAM) のロールを使用すると、AWS クラウド内のサービスとユーザーに対してアクセスポリシーおよび許可をきめ細かく割り当てることができます。このソリューションで作成した IAM ロールでは、AWS CodePipeline ステージおよび AWS CodeBuild ステージ

に対し、Amazon S3 バケットにアクセスして AWS Network Firewall リソースを管理するための許可を付与します。

設計に関する考慮事項

デプロイ可能な AWS リージョン

このソリューションで使用する AWS Network Firewall は、現在、特定の AWS リージョンでのみ利用可能です。このソリューションは、AWS Network Firewall が利用可能な AWS リージョンで起動する必要があります。AWS リージョン別に利用可能な AWS のサービスの最新情報については、「AWS リージョン別のサービス」のリストを参照してください。

AWS CloudFormation テンプレート

このソリューションでは、AWS CloudFormation を使用して AWS クラウドへの「AWS Network Firewall Deployment Automations for AWS Transit Gateway」ソリューションのデプロイを自動化します。このソリューションには次の AWS CloudFormation テンプレートが含まれており、デプロイ前にダウンロード可能です。

aws-network-firewall-deployment-automations-for-aws-transit-gateway.template: このテンプレートを使用してソリューションおよびすべての関連コンポーネントを起動します。デフォルト設定では、AWS Network Firewall、Amazon VPC、AWS Transit Gateway、AWS CodeCommit リポジトリ、AWS CodeBuild、AWS CodePipeline、Amazon CloudWatch がデプロイされます。このテンプレートは特定のニーズに合わせてカスタマイズすることもできます。

> **注意:** AWS CloudFormation のリソースは、AWS Cloud Development Kit (AWS CDK) のコンストラクトで作成されています。

自動デプロイ

ソリューションを起動する前に、このガイドに記載しているアーキテクチャの概要、コンポーネント、デプロイに関する考慮事項を確認してください。このセクションの手順に従い、ソリューションを設定して AWS アカウント内にデプロイします。

デプロイ時間: 約 7 分。AWS CodePipeline は、AWS Network Firewall リソースのデプロイに最大 10 分かかることがあります。

前提条件

このソリューションでは、AWS リージョンで AWS Network Firewall が利用可能であることが必要です。詳細については、「デプロイに関する考慮事項」を参照してください。

AWS Network Firewall のログの送信先を更新する

このソリューションをデプロイ済みである場合は、スタックに対する更新ごとに AWS CodePipeline を手動で開始して、AWS Network Firewall のログの送信先を更新する必要があります。AWS Network Firewall の設定を更新して手動で変更をリリースしないでください。AWS CodePipeline を手動で開始するには、*AWS CodePipeline* ユーザーガイドの「パイプラインを手動で開始する」を参照してください。

AWS Network Firewall、ファイアウォールポリシー、ルールグループを変更するには、「AWS Network Firewall のリソースの設定」を参照してください。

デプロイの概要

次の手順を使用して、このソリューションを AWS にデプロイします。詳細は、各ステップのリンクを参照してください。

ステップ 1. スタックの起動

- AWS アカウントで AWS CloudFormation テンプレートを起動します。

- 必須パラメータ (Stack Name、Transit Gateway ID、Transit Gateway route table(s)、Firewall Logging Configuration) の値を入力します。
- 他のテンプレートパラメータを確認し、必要に応じて調整します。

ステップ 2. AWS Network Firewall、ファイアウォールポリシー、ルールグループの変更

- スタックが正常に作成されると、AWS CodePipeline が AWS CloudFormation によって開始されます。
- AWS Network Firewall、ファイアウォールポリシー、ルールグループを変更します。詳細については、「AWS Network Firewall のリソースの設定」を参照してください。

ステップ 1. スタックの起動

この自動化された AWS CloudFormation テンプレートは、AWS クラウドに「AWS 上のネットワークトラフィックのためのファイアウォールオートメーション」ソリューションをデプロイします。

> **注意:** このソリューションの実行中に使用した AWS のサービスのコストは、お客様の負担となります。詳細については、このガイドの「コスト」セクションで、このソリューションで使用されている各 AWS サービスの料金表ウェブページを参照してください。

1. AWS マネジメントコンソールにサインインし、`aws-network-firewall-deployment-automations-for-aws-transit-gateway` AWS CloudFormation テンプレートを選択して起動します。または、独自にカスタマイズするためにテンプレートをダウンロードすることもできます。
2. このテンプレートは、デフォルトで米国東部 (バージニア北部) リージョンで起動されます。別の AWS リージョンでこのソリューションを起動するには、コンソールのナビゲーションバーのリージョンセレクターを使用します。

> **注意**: このソリューションで使用する AWS Network Firewall は、現在、特定の AWS リージョンでのみ利用可能です。このソリューションは、このサービスをサポートしている AWS リージョンで起動する必要があります。AWS リージョン別に利用可能な AWS のサービスの最新情報については、「AWS リージョン別のサービス」のリストを参照してください。

3. **スタックの作成**ページで、正しいテンプレート URL が **Amazon S3 URL** テキストボックスに表示されていることを確認し、[**次へ**] を選択します。
4. **スタックの詳細を指定**ページで、このソリューションのスタックに名前を割り当てます。名前の文字数制限に関する詳細は、*AWS Identity and Access Management* ユーザーガイドの「IAM および AWS STS クォータ」を参照してください。
5. **パラメータ**で、このソリューションのテンプレートのパラメータを確認し、必要に応じて変更します。このソリューションでは、次のデフォルト値を使用します。

VPC の設定

パラメータ	デフォルト	説明
Provide the CIDR bock for the inspection VPC	`192.168.1.0/26`	Amazon VPC の CIDR ブロック。`/26` 以上の CIDR ブロックである必要があります。

AWS Transit Gateway の設定

パラメータ	デフォルト	説明
Provide the existing AWS Transit Gateway ID you wish to attach to the Inspection VPC	<入力は任意>	現在の AWS リージョンの既存の AWS Transit Gateway ID。(例: `tgw-a1b2c3d4e5`) **注意**: AWS Transit Gateway ID を削除/更新してスタックを更新している場合、AWS Transit Gateway のアタッチメントはアカウントから削除されません。AWS Transit Gateway のアタッチメントは手動で削除する必要があります。

パラメータ	デフォルト	説明
Provide the AWS Transit Gateway Route Table to be associated with the Inspection VPC TGW Attachment	<入力は任意>	既存の AWS Transit Gateway のルートテーブル ID。(例: ファイアウォールのルートテーブル) (例: `tgw-rtb-0a1b2c3d`) **注意**: AWS Transit Gateway のルートテーブル ID を削除してスタックを更新している場合、AWS Transit Gateway のアタッチメントはアカウントから削除されません。AWS Transit Gateway のアタッチメントは手動で削除する必要があります。
Provide the AWS Transit Gateway Route Table to receive 0.0.0.0/0 route to the Inspection VPC TGW Attachment	<入力は任意>	伝播用の既存の AWS Transit Gateway のルートテーブル ID。(例: スポーク VPC のルートテーブル) (例: `tgw-rtb-183ae12f`) **注意**: AWS Transit Gateway ID/AWS Transit Gateway ルートテーブル ID、およびデフォルトルートの AWS Transit Gateway ルートテーブル ID を削除してスタックを更新している場合、AWS Transit Gateway ルートテーブルのデフォルトルート、0.0.0.0/0 のルートエントリは削除されません。ルートは手動で削除する必要があります。

ファイアウォールのログ記録の設定

パラメータ	デフォルト	説明
Select the type of log destination for the Network Firewall	`CloudWatchLogs`	ログのストレージ送信先のタイプ。Amazon S3 バケットまたは Amazon CloudWatch のロググループにログを送信できます。 **注意**: デフォルト値は `CloudWatchLogs` です。このソリューションは、ファイアウォールログのロググループを作成します。Amazon S3 バケットにログを保存することもできます。 ログ記録を設定する必要がない場合は、`ConfigureManually` を選択します。 最初のデプロイ後にこのパラメータを更新する場合は、AWS CodePipeline を手動で開始してログの送信先を更

パラメータ	デフォルト	説明
		新する必要があります。詳細については、「ソリューションのコンポーネント」を参照してください。
Select the type of log to send to the defined log destination.	FLOW	送信するログのタイプ。アラートログは、ステートフルルールに一致するトラフィックと、アラートログメッセージを送信するアクションの設定をレポートします。フローログは、標準のネットワークトラフィックフローログです。 **注意**: これを ALERT ログに設定するか、両方のタイプのログを有効にすることができます。詳細については、*AWS Network Firewall* デベロッパーガイドの「Logging network traffic from AWS Network Firewall」を参照してください。
Select the log retention period for Network Firewall Logs.	90	ログの保持期間 (日数)。この設定は、Inspection VPC のフローログの保持期間にも適用されます。

6. [**次へ**] を選択します。
7. **スタックオプションの設定**ページで、[**次へ**] を選択します。
8. **レビュー**ページで、設定を確認します。テンプレートが AWS Identity and Access Management (IAM) リソースを作成することを承認するチェックボックスを必ずオンにします。
9. [**スタックの作成**] を選択してスタックをデプロイします。

   スタックのステータスは、AWS CloudFormation コンソールの**ステータス**列で確認できます。約 7 分で CREATE_COMPLETE ステータスが表示されます。

   AWS CloudFormation がスタックの作成を完了すると、ソリューションで作成した AWS CodePipeline が引き続き実行して、すべての AWS Network Firewall リソースを作成します。

ステップ 2. AWS Network Firewall、ファイアウォールポリシー、ルールグループの変更

スタックを正常にデプロイすると、AWS CodePipeline は AWS Code Build のステージを開始します。各ステージでは、AWS Network Firewall コンポーネントを検証してデプロイします。デプロイステージが完了すると、AWS Network Firewall とファイアウォールポリシーを確認できます。

デフォルトの AWS Network Firewall、ファイアウォールポリシー、および作成したルールグループを変更するには、「AWS Network Firewall のリソースの設定」を参照してください。

その他のリソース

AWS のサービス

- AWS CloudFormation
- AWS CodeCommit
- AWS CodeBuild
- AWS CodePipeline
- Amazon CloudWatch
- AWS Network Firewall
- Amazon S3
- AWS Transit Gateway
- Amazon VPC

AWS Network Firewall のリソースの設定

ソリューションをデプロイしたら、ネットワークのリソースをカスタマイズできます。このソリューションでは、AWS CodeCommit リポジトリを作成して、すべての AWS Network Firewall の設定ファイルを保存します。これらのファイルは更新可能であり、新しいリソースを対応する各フォルダ内に作成できます。変更をコミットして AWS CodeCommit リポジトリにプッシュすると、このソリューションは設定パッケージを使用して AWS Network Firewall のリソースを作成または更新します。ファイアウォール、ファイアウォールポリシー、ルールグループに対する変更は、AWS CodePipeline の実行が正常に完了した後で確認できます。パイプラインのステータスをモニタリングして、変更が正常に

デプロイされたことを確認することをお勧めします。AWS CodePipeline で AWS CodeBuild ステージのログを確認することもできます。

> **注意:** `FirewallPolicyArn` および `ResourceARN` の属性へのすべての参照には、実際の JSON ファイルへの参照パスを含める必要があります。これらの値は、このソリューションで設定を取得するために使用します。AWS CodeCommit リポジトリに示している設定の例を参照してください。

AWS Network Firewall およびファイアウォールポリシーに固有の文字列が追加され、1 つの AWS リージョンでソリューションを複数回デプロイできるようになります。デプロイしたリソースには、AWS リージョンごとに固有の名前が付きます。

ソリューションで参照しているのと同じ名前のリソースが AWS Network Firewall 内に既にある場合、これらのリソースは AWS CodeCommit リポジトリに提供されている設定で更新されます。変更をコミットする前に、アカウントと AWS リージョンにおいて以前に AWS Network Firewall コンソールで作成したリソースのリソース名を確認することをお勧めします。

AWS CodeBuild の検証ステージ

このソリューションでは、2 つの AWS CodeBuild ステージを作成します。最初のステージでは、設定ファイル (ファイアウォール、ファイアウォールポリシー、ルールグループ) を検証し、JSON 形式が有効かどうかを確認します。このソリューションでは、これらのファイルを使用して AWS Network Firewall の API を検証し、ファイルに定義されている属性に有効なデータがあることを確認します。ファイルにフォーマットの問題や無効なデータがある場合、AWS CodeBuild ステージは `Failed` 状態になり、AWS Network Firewall へのファイルのデプロイは続行されません。AWS CodeBuild の検証ステージは、次のログの例に示すような、ファイルのエラーの詳細を表示します。

```
[TIMESTAMP] : "-----------INVALID FILES START-----------"
[TIMESTAMP]: {
   “path”: “./firewallPolicies/firewall-policy-1.json”,
   “error”: “Unexpected key ‘key’ found in params.FirewallPolicy”
}
[TIMESTAMP]: "-----------INVALID FILES END-----------"
[TIMESTAMP]: “Validation failed.”
[TIMESTAMP]: “Error in firewall config validation” : “Validation
failed.”
```

ソリューションがデプロイされると、AWS CodeCommit リポジトリは次のデフォルトのディレクトリ構造になります。

- **Examples** - このディレクトリには、設定ファイルの例が含まれています。
- **Firewalls** - このディレクトリには、JSON 形式のファイアウォール設定が含まれています。属性は CreateFirewallAPI アクションにドキュメントとして追加されます。

> **注意**: `FirewallPolicyArn` には、コードコミットリポジトリ内のファイアウォールポリシーファイルのファイルパスと完全に一致する値があります。

次の JSON ファイルの例に示すように、このソリューションでは、`./firewallPolicies/firewall-policy-1.json` コミットリポジトリパスでファイアウォールポリシーとして `firewall-policy-1.json` を使用しています。

```
{
   “FirewallName”: “Firewall-1”,
   “FirewallPolicyARN”: “./firewallPolicies/firewall-policy-1.json”,
   “Description”: “Network Firewall 1”.
   “DeleteProtection”: true,
   “SubnetChangeProtection”: true
}
```

- **FirewallPolicies** - このディレクトリには、ファイアウォールポリシーが JSON 形式が含まれています。これらのポリシーの属性は CreateFirewallPolicy に説明されています。属性 `ResourceArn` には、コードコミットリポジトリ内のルールグループファイルのファイルパスと正確に一致する値があります。AWS Network Firewall ポリシーの例を次に示します。

```
{
"FirewallPolicyName": "Firewall-Policy-1",
"Description": "Firewall Policy 1",
"FirewallPolicy": {
  "StatelessDefaultActions": [
    “aws:drop”
  ],
  "StatelessRuleGroupReferences": [
    {
      "Priority": 30,
      “ResourceArn”:“./ruleGroups/stateless-fwd-to-
stateful.example.json”
```

```
    },
    {
      "Priority": 20,
      "ResourceArn":"./ruleGroups/stateless-pass-
action.example.json"

}
  ],
  "StatefulRuleGroupReferences":[
     {
      "ResourceArn":"./ruleGroups/stateful-domainblock.example.json"
    },
    {
      "ResourceArn":"./ruleGroups/suricata-rule-reference.json"
    }
  ]
 }
}
```

注意: ファイアウォールポリシーファイルの `ResourceArn` 属性には、AWS CodeCommit リポジトリのルールグループファイルへのファイルパスが必要です。

- **RuleGroup** - このディレクトリには、ルールグループの設定が JSON 形式で含まれています。これらの設定の属性は、CreateRuleGroup に説明されています。ルールグループは、次のステートフルルールグループファイルの例に示すように、`RuleGroup` 属性または rules (Suricata フラット形式) 属性に詳細を指定することで定義できます。

```
{
   "RuleGroupName": "StatefulRulesExample1",
   "RuleGroup": {
    "RulesSource": {
      "RulesSourceList": {
        "TargetTypes": ["HTTP_HOST"],
        "Targets": [
          "test.example.com",
          "test2.example.com"
        ],
        "GeneratedRulesType": "DENYLIST"
       }
      }
    },
    "Type": "STATEFUL",
    "Description": "Stateful Rule",
    "Capacity": 100
}
```

次の Suricata ファイルの例では、rules 属性は、ルールが定義されている `drop.rules` ファイルを参照しています。詳細については、Drop.rules ファイルの例を参照してください。

```
{
   “RuleGroupName”: “suricata-drop-rules”,
   “Rules”: “./ruleGroups/drop.rules”,
   “Type”: “STATEFUL”.
   “Description”: “Suricata rule group”,
   “Type”: 100
}
```

注意: `drop.rules` ファイルは設定パッケージに追加する必要があり、ローカルパスのみが許可されます。Amazon S3 および HTTP リンクは許可されていません。

トラブルシューティング

問題: AWS CloudFormation スタックが正常に完了しても、一部の AWS Network Firewall リソースが作成されない。

解決策: AWS CloudFormation スタックが完了した後でも、ソリューションで作成した AWS CodePipeline ステージが **In-Progress** 状態になることがあります。AWS CodePipeline ステージが完了すると、すべての AWS Network Firewall リソースが AWS Network Firewall コンソールで使用可能になります。

問題: AWS CodePipeline ステージが失敗する。

解決策: AWS CodePipeline ステージが **Failed** 状態の場合は、このソリューションが AWS Network Firewall リソースの作成または更新の操作を完了できなかったことを意味します。AWS CodePipeline ステージのログを参照して、AWS CodeBuild ステージが正常に完了したことを確認してください。JSON ファイルが有効でないか、情報が正しくない場合は、ファイルを検証する AWS CodeBuild ステージにエラーおよび該当のファイル名が表示されます。詳細については、AWS CodeBuild ユーザーガイドを参照してください。

ソリューションのアンインストール

「AWS 上のネットワークトラフィックのためのファイアウォールオートメーション」ソリューションは、AWS マネジメントコンソールを使用するか、AWS Command Line Interface (AWS CLI) を使用してアンインストールできます。

AWS マネジメントコンソールの使用

1. AWS CloudFormation コンソールにサインインします。
2. このソリューションのインストール用のスタックを選択します。
3. [**削除**] を選択します。
4. 次のリソースは、ソリューションの削除後も保持されます。リソースを手動で削除するには、次のリンクを参照してください。
    - AWS CodeCommit リポジトリ
    - Amazon CloudWatch のロググループ
    - Amazon S3 にある AWS CodePipeline のアーティファクトバケット
    - Amazon S3 にある AWS CodeBuild のソースコードバケット
    - AWS Network Firewall
    - AWS Network Firewall のファイアウォールポリシー
    - AWS Network Firewall のルールグループ
    - Inspection VPC
    - AWS Transit Gateway のアタッチメント

AWS コマンドラインインターフェイスの使用

AWS Command Line Interface (AWS CLI) がお客様の環境で使用できるかどうかを確認します。インストール手順については、*AWS CLI* ユーザーガイド の「AWS Command Line Interface とはどの

ようなものですか」を参照してください。AWS CLI が使用可能になったことを確認したら、次のコマンドを実行します。

```
$ aws cloudformation delete-stack --stack-name <installation-stack-name>
```

運用メトリクスの収集

このソリューションには、匿名の運用メトリクスを AWS に送信するオプションが含まれています。当社はこのデータを使用して、ユーザーがこのソリューション、関連サービスおよび製品をどのように使用しているかをよりよく理解し、提供するサービスや製品の改善に役立てます。有効にすると、次の情報が収集され、AWS に送信されます。

- **Solution ID:** AWS ソリューション識別子
- **Unique ID (UUID):**「AWS 上のネットワークトラフィックのためのファイアウォールオートメーション」ソリューションのデプロイごとにランダムに生成される固有の識別子
- **Timestamp:** データ収集タイムスタンプ
- **アカウントにデプロイされた AWS CloudFormation スタックの数**
- **管理対象のファイアウォールの数**
- **管理対象のファイアウォールポリシーの数**
- **デプロイされたステートフルルールグループの数**
- **デプロイされたステートレスルールグループの数**
- **デプロイされた Suricata ルールの数**
- **AWS Network Firewall の送信先タイプ**
- **AWS Network Firewall のログタイプ**

AWS は、このアンケートを通じて収集されたデータを所有します。データ収集には、AWS プライバシーポリシーが適用されます。この機能を無効にするには、次のように AWS CloudFormation テンプレートのマッピングセクションを変更します。

```
"Send" : {
"AnonymousUsage" : { "Data" : "Yes" }
},
```

次のように変更します。

```
"Send" : {
"AnonymousUsage" : { "Data" : "No" }
},
```

ソースコード

このソリューションのソースファイルをダウンロードし、カスタマイズ内容を他のユーザーと共有するには、GitHub リポジトリにアクセスしてください。「AWS 上のネットワークトラフィックのためのファイアウォールオートメーション」ソリューションのテンプレートは、AWS Cloud Development Kit (AWS CDK) を使用して作成しています。詳細については、README.md ファイルを参照してください。

改訂履歴

日付	変更
2021 年 2 月	初回リリース
2022 年 4 月	リリース v1.0.1: ブランチ名の更新。詳細については、GitHub リポジトリの CHANGELOG.md ファイルを参照してください。

寄稿者

- Lalit Grover
- Nikhil Reddy

注意

お客様は、この文書に記載されている情報を独自に評価する責任を負うものとします。このドキュメントは、(a) 情報提供のみを目的としており、(b) AWS の現行製品とプラクティスを表したものであり、予告なしに変更されることがあり、(c) AWS およびその関連会社、サプライヤー、またはライセンサーからの契約義務や確約を意味するものではありません。AWS の製品やサービスは、明示または暗示を問わず、いかなる保証、表明、条件を伴うことなく「現状のまま」提供されます。お客様に対する AWS の責任は、AWS 契約により規定されます。本書は、AWS とお客様の間で行われるいかなる契約の一部でもなく、そのような契約の内容を変更するものでもありません。

「AWS 上のネットワークトラフィックのためのファイアウォールオートメーション」ソリューションは、Apache Software Foundation で利用可能な Apache ライセンスバージョン 2.0 の条件に基づいてライセンスされます。