DDoS に対する AWS の

ベストプラクティス

2015 年 6 月

注意

本文書は、情報提供の目的のみのために提供されるものです。本書の発行時点における AWS の現行製品と慣行を表したものであり、それらは予告なく変更されることがあります。お客様は本文書の情報および AWS 製品の使用について独自に評価する責任を負うものとします。これらの情報は、明示または黙示を問わずいかなる保証も伴うことなく、「現状のまま」提供されるものです。本文書内のいかなるものも、AWS、その関係者、サプライヤ、またはライセンサーからの保証、表明、契約的なコミットメント、条件や確約を意味するものではありません。お客様に対する AWS の責任は AWS 契約によって規定されています。また、本文書は、AWS とお客様との間の契約に属するものではなく、また、当該契約が本文書によって修正されることもありません。

目次

要約 3

はじめに 4

分散型サービス妨害攻撃 4

緩和テクニック 6

- 攻撃対象領域を削減する 7
- スケールして攻撃を吸収できるようにする 9
- 公開されたリソースを保護する 16
- 通常時の動作について学習する 21
- 攻撃に対する計画を作成する 25

まとめ 26

要約

このホワイトペーパーは、アマゾン ウェブ サービス (AWS) で実行されるアプリケーションの分散型サービス妨害攻撃（DDoS）に対する耐障害性を向上させる必要のあるお客様向けに書かれたものです。このホワイトペーパーでは、分散型サービス妨害攻撃の概要と可用性を維持するための手法を説明し、耐障害性を向上させることを目的としたアーキテクチャのガイダンスとなるリファレンスアーキテクチャを提示します。

本書の対象者は IT に関する意思決定者およびセキュリティ担当者であり、ネットワーキング、セキュリティ、AWS におけるセキュリティの基本的な概念を理解していることを前提として書かれています。各セクションには、示されているタスクの実行方法に関する具体的な情報が記載された AWS ドキュメントへのリンクがあります。詳細については、AWS re:Invent の SEC307 および Sec305 セッションも参照してください。

はじめに

サービス拒否 (DoS) 攻撃は、お客様のウェブサイトやアプリケーションをエンドユーザーが利用できないようにすることを目的とした攻撃です。攻撃者は、これを行うために、ネットワークやその他のリソースを消費するさまざまな手法を用いて、正規のエンドユーザーのアクセスを中断させます。DoS 攻撃の最もシンプルな形式では、図 1 に示すように攻撃者が 1 つのチャネルを通じて被害者を標的にします。

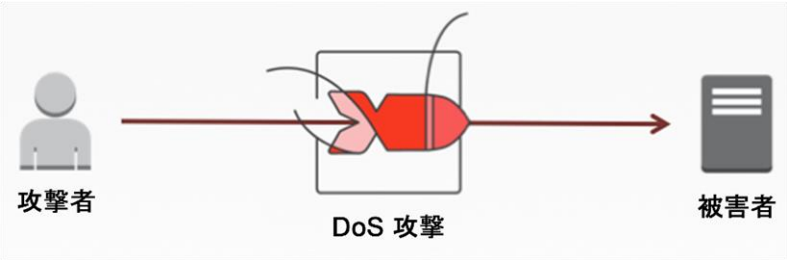

図 1: DoS 攻撃の図

分散型サービス妨害攻撃

分散型サービス妨害 (DDoS) 攻撃では、攻撃者は複数のシステムを使用して標的に対する攻撃を指揮します。これらのシステムは、協力者のグループによって侵害または制御されたホストである場合があります。図 2 に示すように、DDoS 攻撃では、侵害された各ホストが攻撃に参加し、標的に対して大量のパケットやリクエストを生成します。

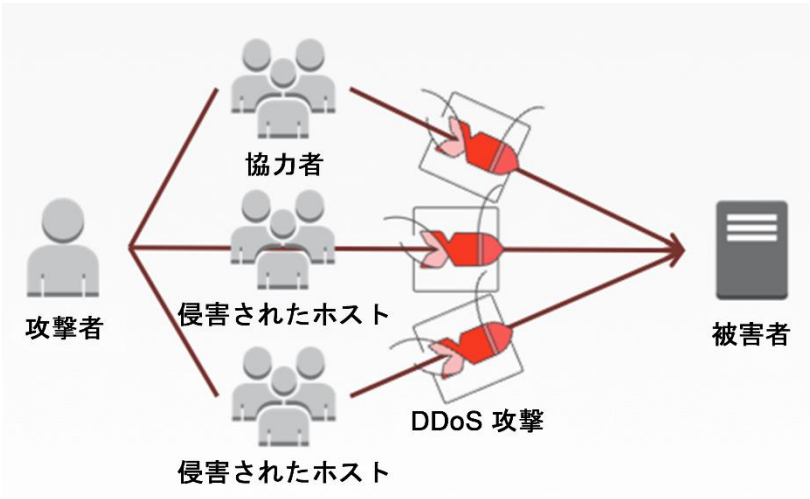

図 2: DDoS 攻撃の図

攻撃者が標的を攻略するためには、数多くの方法があります。たとえば、反射と増幅の手法を組み合わせて大量のパケットを生成したり、大きなボットネットを使用したりする方法があります。リフレクター攻撃では、サーバーから、なりすました IP アドレスへの応答を引き出し、これにより侵害を受けたサーバーは反射物のように動作します。アンプ攻撃とともにリフレクター攻撃を使用すると、侵害を受けたサーバーは元のリクエストに対して不均等な応答を送信します。図 3 に示すように、攻撃者はこれらの手法を組み合わせることで、限られた数のホストからの少量の帯域幅を、標的を対象とした大量のトラフィックに変えることができます。

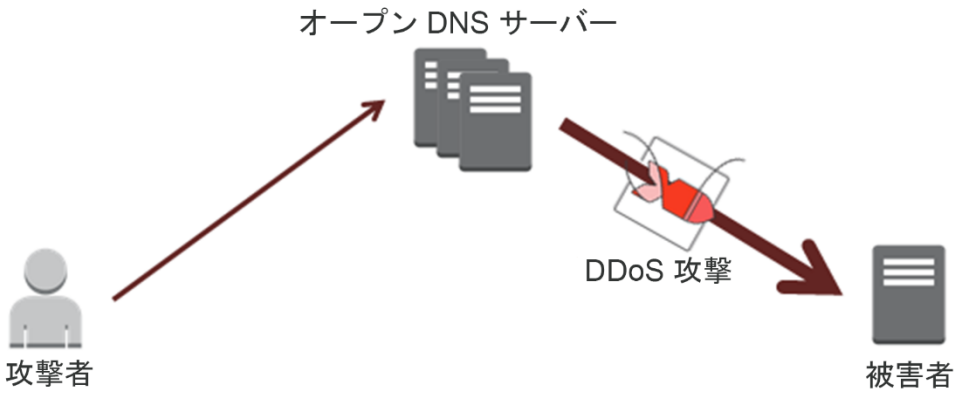

図 3: DDoS のフラッド、リフレクター、アンプ攻撃

増幅係数（応答サイズに対するリクエストサイズの比率）は、使用されているプロトコル（DNS、NTP、SSDP など）によって異なります。たとえば、DNS の平均増幅係数は 28～54 の範囲です。この場合、攻撃者は 64 バイトのリクエストペイロードを DNS サーバーに送信して、3,456 バイトの不要なトラフィックを生成することができます。[1]

[1] https://www.us-cert.gov/ncas/alerts/TA14-017A

望まれないトラフィックが標的に到達すると、トラフィックはネットワーキングハードウェア、オペレーティングシステム、およびアプリケーションの複数の各レイヤーを通過し、いずれのレイヤーでもリソースを使い尽くすことができます。消費されたリソースにより、攻撃は帯域幅の使い尽くし（UDP フラッドなど）、プロトコルの使い尽くし（SYN フラッドなど）、またはアプリケーションの使い尽くし（HTTP GET/POST フラッドなど）に分類されます。

攻撃のタイプとは無関係に、DDoS 攻撃は根本的には可用性の問題であり、攻撃者の目的は正規のエンドユーザーに対してリソースを使用できない状態にすることです。したがって、AWS 内でフェイルオーバー機能を利用して、DDoS 攻撃によって発生する可用性に関する脆弱性の問題を減らすことができます。

その効果は、使用中のサービスやその設定方法によって変わる可能性があります。したがって、ここで説明する手法によって特定のレベルの可用性は保証されません。AWS の料金は使用量に基づいており、本書で説明している手法を使用する際は、これについて考慮してください。

緩和テクニック

従来、セキュリティとは攻撃者が目的を達成できないようにするために、フィルタやその他の障壁を設置することでした。障壁の設置に加えて、変化する状況に合わせてスケールできる、柔軟性のあるアーキテクチャがあると有益です。以下では、攻撃に対する脆弱性を減らすために使用できる 5 つの手法について説明します。

- 攻撃対象領域を削減する
- スケールして攻撃を吸収できるようにする
- 公開されたリソースを保護する
- 通常時の動作について学習する
- 攻撃に対する計画を作成する

本書では、これらの各手法についてと、AWS を使用した弾力性の構築方法を示すために、図 4 に示したリファレンスアーキテクチャについて説明します。AWS アーキテクチャセンターの全般的なガイドラインに従って、「耐障害性と高可用性」で説明されているようにすぐにフェイルオーバーするシステムを構築することもできます。

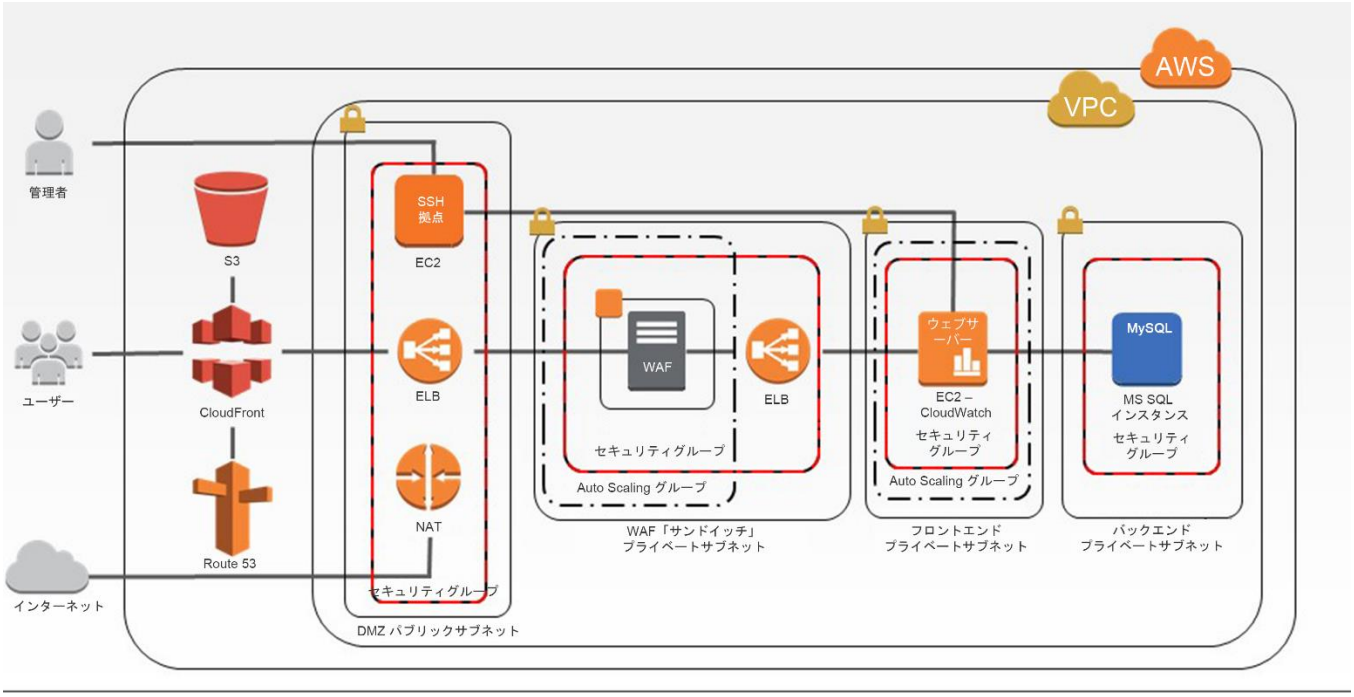

図 4: DDoS 弾力性のリファレンスアーキテクチャ

攻撃対象領域を削減する

攻撃領域は、お客様のアプリケーションへのアクセスを許可するさまざまなインターネットエントリポイントの組み合わせです。攻撃領域を最小化するには、(a) 必要なインターネットエントリポイントの数を減らす、(b) 重要でないインターネットエントリポイントを排除する、(c) 管理トラフィックからエンドユーザートラフィックを分離する、(d) 必要なインターネットエントリポイントを、信頼できないエンドユーザーがアクセスできないレベルまで難読化する、(e) インターネットエントリポイントを分離して攻撃の効果を最小化する、という戦略があります。この戦略は、Amazon Virtual Private Cloud で達成できます。

Amazon Virtual Private Cloud

Amazon Virtual Private Cloud (VPC) を使用すると、AWS 内の独自の論理的に分離された領域の仮想ネットワーク（仮想プライベートクラウドとも呼ばれます）を定義できます。VPC では、セキュリティが確保された環境で EC2 インスタンスのような AWS リソースを起動するために、ルートテーブル、ネットワークゲートウェイ、およびセキュリティを設定することができます。

さらに重要なことは、VPC では、パブリック以外のインスタンスをプライベートサブネットのみで利用可能にしたり、プライベート DNS エントリを内部アプリケーションによってのみアクセス可能にするために、インターネットでインスタンスを非表示にすることができます。これは、セキュリティグループとネットワークアクセスコントロールリスト (ACL) を作成することにより達成できます。セキュリティグループは関連する EC2 インスタンスに対するファイアウォールとして機能し、ネットワーク ACL は関連するサブネットに対するファイアウォールとして機能します。VPC インスタンスはセキュリティグループを防御の最初のレイヤーとして使用して保護し、第 2 のレイヤーとしてネットワーク ACL を追加することができます。セキュリティグループとネットワーク ACL の詳細については、「VPC のセキュリティ」を参照してください。

以下のリンクに従って、Amazon マネジメントコンソールで VPC を設定します。ステップ 3 で選択するインスタンスがわからない場合は、次のステップに進む前に、「Amazon Elastic Compute Cloud」というセクションに進んでください。

ステップ 1: VPC とインターネットゲートウェイをセットアップする

ステップ 2: VPC のセキュリティグループをセットアップする

ステップ 3: インスタンスを VPC 内で起動する

ステップ 4: Elastic IP アドレスをインスタンスに割り当てる

ステップ 5: ネットワークアクセスコントロールリストをセットアップする

以下に示す Amazon VPC の一般的な設定シナリオのステップに従うこともできます。

シナリオ 1: 1 つのパブリックサブネットのみを持つ VPC

シナリオ 2: パブリックサブネットとプライベートサブネットを持つ VPC

シナリオ 3: パブリックサブネットとプライベートサブネット、およびハードウェア VPN アクセスを持つ VPC

シナリオ 4: 1 つのプライベートサブネットのみ、およびハードウェア VPN アクセスを持つ VPC

リンクの手順に従うと、VPC の EC2 インスタンスに接続できます。Linux インスタンスに接続する方法については、『Linux インスタンス用 Amazon EC2 ユーザーガイド』の「Linux インスタンスへの接続」を参照してください。Windows インスタンスに接続する方法については、『Microsoft Windows インスタンス用 Amazon EC2 ユーザーガイド』の「Windows インスタンスへの接続」を参照してください。

スケールして攻撃を吸収できるようにする

DDoS 攻撃は、スケールに関するものです。ほとんどの攻撃者は、アプリケーションが対応できないレベルのトラフィックを送信することで目的を達成します。攻撃のスケールを上回るアーキテクチャを実装することで、攻撃者側でより多くの時間とリソースを必要とする障壁を作成し、それによりアプリケーションの弾力性を高めることができます。

AWS 内では、水平スケーリングと垂直スケーリングという 2 つの形式のスケーリングを活用できます。水平スケーリングは、インフラストラクチャにインスタンスやサービスを追加することで機能します。垂直スケーリングは、より多くのメモリ、CPU、およびキャパシティーを持つインスタンスを選択することで機能します。スケールのこれら 2 つのディメンションを使用すると、DDoS 攻撃に対する 4 つの直接的なメリットが得られます。

- 攻撃は、より広い領域に分散され、影響する範囲については最小化される。
- 攻撃者は、攻撃をスケールアップするためにさらに多くにリソースを消費する必要がある。
- スケーリングによって、DDoS 攻撃を分析し、対策を実行するまでの時間を稼ぐことができる。
- スケーリングによって、その他の障害シナリオでの冗長性のセキュリティレイヤーが追加される。

DDoS の観点からは、AWS でスケーリングを活用するために、(1) アプリケーションに対して適切なインスタンスタイプを選択する、(2) Elastic Load Balancing や Auto Scaling といったサービスを自動的にスケールするよう設定する、(3) Amazon CloudFront や Amazon Route 53 など、AWS グローバルサービスに組み込まれた固有のスケールを使用する、という 3 つの方法があります。

Amazon Elastic Compute Cloud

インスタンスタイプ

Amazon Elastic Compute Cloud (EC2) は、スケーラブルなコンピューティング容量をクラウド上で提供し、ハードウェアへの事前投資を不要にします。EC2 を使用して、インスタンスと呼ばれる仮想サーバーを起動し、トラフィックスパイクの変化に応じてスケールアップまたはスケールダウンすることができます。インスタンスを起動するときは、指定するインスタンスタイプによって、アプリケーションに使用されるホストコンピュータのハードウェアコンポーネント（コンピューティング、メモリ、ストレージなど）が決まります。

アプリケーションの弾力性を高めるには、スケールしてアプリケーションや予期しないスパイクをサポートできるインスタンスタイプを選択します。コストに対して最適化された一部の EC2 インスタンスは、保証されたネットワークリソースを提供せず、DDoS 攻撃の可用性への影響を受けやすくなる場合があります。インフラストラクチャにとって不可欠のアプリケーションでは、EBS 最適化インスタンスまたはホストコンピュータで 10 ギガビットのネットワーク接続を持つインスタンスタイプを選択します。詳細については、「Amazon EC2 インスタンスの構成」を参照してください。

拡張ネットワーキング

C3、C4、R3、D2、および I2 インスタンスでは、拡張ネットワーキング機能を有効にできます。これにより、より高いネットワークパフォーマンス（秒あたりのパケット数）が提供されます。この機能では、従来の実装と比較し、I/O パフォーマンスが高く、CPU 利用率が低くなるネットワーク仮想化スタックが使用されます。拡張ネットワーキングでは、アプリケーションは高いパフォーマンス（秒あたりのパケット数）、低いネットワークレイテンシー、スケーラビリティの向上というメリットが得られます。これらは、DDoS 攻撃に対する弾力性を構築するうえで役立つ機能です。

拡張ネットワーキングを有効にするには、SR-IOV ドライバを備えた、適切なハードウェアアシスト仮想マシン (HVM) である Amazon マシンイメージ (AMI) を起動する必要があります。Amazon HVM Linux AMI にはデフォルトで SR-IOV ドライバが含まれています。デフォルトで SR-IOV ドライバーを含まない AMI の場合は、該当するドライバをダウンロードおよびインストールする必要があります。次の AMI に対して拡張ネットワーキングをダウンロードして有効にする手順については、以下のリンク先を参照してください。

Amazon Linux での拡張ネットワーキングの有効化

Ubuntu での拡張ネットワーキングの有効化

他の Linux ディストリビューションでの拡張ネットワーキングの有効化

Windows での拡張ネットワーキングの有効化

Elastic Load Balancing

Elastic Load Balancing (ELB) を使用してインフラストラクチャへのトラフィックを管理すると、複数のアベイラビリティーゾーン (AZ) にある複数の EC2 インスタンスにトラフィックを分散させて、1 つのインスタンスのみが過負荷になるリスクを最小限に抑えるというメリットが得られます。ELB を使用すると、トラフィック全体の流れを中断することなく、ニーズの変化に応じて EC2 インスタンスを追加および削除することができます。たとえば、1 つの EC2 インスタンスが失敗した場合、ELB は自動的に残りの実行中の EC2 インスタンスにトラフィックを再ルーティングします。失敗した EC2 インスタンスが回復すると、ELB はそのインスタンスへのトラフィックも復元します。

また、ELB はクライアントにとって単一の管理先となるだけでなく、ネットワークへの攻撃に対する最前線の防御機能も持ちます。すべての EC2 インスタンスを ELB の背後に配置し、インターネットのみに ELB を公開します。これにより、攻撃対象領域を最小化することができます。ELB のインスタンスを参照し、必要に応じて規模を拡張または縮小（処理能力を追加または削除）することができます。個別のインスタンスを管理する必要はありません。VPC とともに使用すると、セキュリティグループやネットワーク ACL を ELB に関連付けて、追加の弾力性レイヤーを提供できます。ELB は有効な TCP リクエストのみをサポートするため、UDP や SYN フラッドなどの DDoS 攻撃は、インスタンスに到達することはできません。

次のセクションでは、2 つの基本的なロードバランサー（インターネット接続ロードバランサーと内部ロードバランサー）を設定する方法を示します。これらの 2 つのロードバランサー（図 4 に示す）は、セクション「ウェブアプリケーションファイアウォール」セクションで示したレイヤー 7 の保護のスケーリングに必要です。

ステップ 1: デフォルト VPC での基本的なロードバランサーの作成

ステップ 2: Amazon VPC での基本的な内部向けロードバランサーの作成

EC2 インスタンスがある他のリージョンで、これらのステップを繰り返します。

Auto Scaling

Auto Scaling により、アプリケーションの可用性を維持できると同時に、お客様が定義する条件に応じて EC2 の能力を自動的に縮小あるいは拡張することができます。たとえば、ネットワークトラフィックが多い（典型的な DDoS 攻撃）ときに、新しいインスタンスを Auto Scaling グループに段階的に追加する条件を設定できます。また、ネットワークトラフィックが少ないときに同じようにインスタンスを除去する条件を設定できます。Amazon CloudWatch を使用して規模の拡大や縮小をトリガーし、ELB が Auto Scaling グループ内でインスタンスにトラフィックを分散させるようにできます。

Auto Scaling グループを初めて本稼働させる前に、検討が必要なアクションがいくつかあります。開始する前に、AWS クラウドでの実行を想定してアプリケーションを徹底的に再確認する時間を取り、以下を検討します。

- サーバーの起動と設定に要する時間。アプリケーションの起動に 5 分以上かかる場合は、複数のインスタンスにおいてアプリケーションを事前に実行状態にしておくか、スケーリングのしきい値を低くすることをお勧めします。
- アプリケーションのパフォーマンスと最も関連性が高いメトリックスの特定。DDoS 攻撃を示すメトリックスの例として、CPUUtilization、NetworkIn、StatusCheckFailed などがあります。
- Auto Scaling グループの一部として使用する可能性がある既存リソース（EC2 インスタンスや AMI など）。Auto Scaling グループに対して攻撃を受けている状態でもアプリケーションを実行しつづけるためには、同じタイプのインスタンスまたはより高いキャパシティーが必要です。
- Auto Scaling グループに含める AZ の数。最低でも 2 つの AZ をお勧めします。
- スケールアップおよびスケールダウンを行う速度。DDoS 攻撃は波のように押し寄せることに注意してください。最初の攻撃の後でスケールダウンしたが、結局再度スケールアップしなければならなくなる状況は避けたいものです。
- Auto Scaling グループでの EC2 インスタンスの最大数。インスタンスが追加されると、コストも増える可能性があります。Auto Scaling ポリシーを作成するときは、インスタンスの最大数を設定できます。この最大数に達したときのアラームを設定することもできます。アラームを設定するステップについては、「Amazon CloudWatch」を参照してください。

インフラストラクチャとアプリケーションに関する理解を深めれば、Auto Scaling の実装はさらに効果的になります。インフラストラクチャについて十分な情報が得られたら、Auto Scaling グループの作成を開始できます。

ステップ 1: 起動設定を作成する

ステップ 2: Auto Scaling グループを作成する

ステップ 3: Auto Scaling グループを確認する

Amazon CloudFront

Amazon CloudFront は、他の Amazon サービスと統合しながら、エンドユーザーにコンテンツを配信するための簡単で効果的な手段を提供するコンテンツ配信ネットワーク (CDN) です。CDN は、オリジンサーバーとエンドユーザーの間のプロキシレイヤーとして機能します。CDN はコンテンツをキャッシュし、複数のエッジ (PoP) からサイトへの接続を最適化してパフォーマンスを向上させます。その効果として、すべてのトラフィックリクエストをオリジンに送信する代わりに、リクエストはエンドユーザーに直接応答する複数の PoP に分散されます。

Amazon CloudFront は、複数の PoP を使用することで、複数の場所のトラフィックを分散させて、インフラストラクチャと一部のアプリケーションレイヤー DDoS 攻撃の両方から保護する能力を備えています。これらのそれぞれの場所に、AWS はキャパシティーと冗長性のために複数のインターネット接続を持ち、これにより Amazon CloudFront は正規のエンドユーザーにコンテンツを提供しながら、攻撃のトラフィックを分離することができます。

Amazon CloudFront には、無効なリクエストを削除しながら、有効な TCP 接続と HTTP リクエストのみが行われるようにするフィルタリング機能もあります。これにより、オリジンから無効なトラフィック（UDP フラッド、SYN フラッド、およびスローリードでよく使用されます）を処理する負荷が取り除かれます。

Amazon CloudFront を使用するには、ディストリビューションを作成してオリジン（EC2 インスタンス、S3 バケット、ELB、またはカスタムウェブサーバー）を指定します。ディストリビューションを設定すると、Amazon CloudFront は、エンドユーザーリクエストへの応答とパフォーマンスを向上させるためのコンテンツのキャッシュを開始します。

次の手順では、ELB をオリジンとして Amazon CloudFront ディストリビューションを作成する方法を示します（図 4 に示します）。別のディストリビューション用に Amazon CloudFront を設定することができます。詳細については、「Amazon EC2 と他のカスタムディストリビューションを使用するための要件と推奨事項」を参照してください。

ステップ 1: ウェブディストリビューションを作成する

ステップ 2: ウェブディストリビューションをテストする

Amazon Route 53

DDoS 攻撃の最も一般的な標的は、ドメインネームシステム (DNS) です。DNS はドメイン名を見つけて IP アドレスに変換するために使用されるため、攻撃者は頻繁に DNS を単一障害点と見なします。たとえば、アプリケーションが実際に利用可能であっても、アプリケーションの DNS がダウンしている場合、エンドユーザーは正しくルーティングされません。こうすることで、アプリケーションは実質的に利用できなくなります。DNS は UDP を使用し、クエリに応答するために使用されるため、高いボリュームによるリフレクター攻撃およびアンプ攻撃を受けやすくなります。このような理由から、DNS の弾力性を高めるためのリソースが必要になります。

Amazon Route 53 は、可用性に優れ、スケーラブルな DNS サービスで、AWS の内部または外部で実行しているインフラストラクチャにエンドユーザーをルーティングすることを念頭に設計されています。Amazon Route 53 は、さまざまなルーティングタイプを通じてグローバルにトラフィックを管理できるようにします。これには、レイテンシーによるルーティング、Geo DNS、加重ラウンドロビンが含まれます。これらのルーティングタイプは、Route 53 フェイルオーバーと関連付けて、低レイテンシーの耐障害性に優れたアーキテクチャを可能にすることもできます。

Amazon Route 53 には、シャッフルシャーディングと anycast ルーティングという、DDoS 攻撃を受けていても連携してエンドユーザーがアプリケーションにアクセスできるようにする 2 つの機能があります。

シャッフルシャーディング
シャッフルシャーディングはデータベースシャーディングの概念に似ていて、データの水平パーティションが異なるデータベースサーバー間で分散され、それにより負荷が分散されるとともに冗長性が提供されます。同様に、Amazon Route 53 はシャッフルシャーディングを使用して数多くのエンドポイントに DNS リクエストを分散させ、アプリケーションの複数のパスとルートを提供します。

Anycast ルーティング
Anycast ルーティングでは、複数の PoP から同じ IP アドレスをアドバタイズすることで冗長性を高めます。DDoS 攻撃が 1 つのエンドポイントを攻略した場合、シャッフルシャーディングによって障害が分離され、インフラストラクチャに追加のルートが提供されます。

また、Amazon Route 53 は、ヘルスチェックと DNS フェイルオーバーを提供し、DNS クエリで正常なリソースのみが使用されるようにします。たとえば、example.com が、世界各地の AZ にそれぞれ 2 台ずつ、合計 10 個のインスタンスでホストされているとします。それらのインスタンスの正常性をチェックし、正常なインスタンスのみを使って example.com の DNS クエリに応答するように Amazon Route 53 を設定することができます。Amazon Route 53 のエイリアス、加重、レイテンシー、位置情報ルーティング、フェイルオーバーの各リソースレコードセットを使用して、さまざまなフェイルオーバー構成をセットアップできます。

DNS に Amazon Route 53 を使用するには、ドメイン名とサブドメインを Amazon Route 53 に登録または移行できます。このセクションでは、Amazon Route 53 のエイリアスレコードセットとして Amazon CloudFront ディストリビューションを使用します。「AWS リソースへのクエリのルーティング」のステップに従って、トラフィックを他のサービスにルーティングできます。

ステップ 1: ドメイン名を登録し、DNS サービスとして Amazon Route 53 を設定する

ステップ 2: Amazon CloudFront ディストリビューションにクエリをルーティングする（パブリックホストゾーンのみ）

ステップ 3: ヘルスチェックと DNS フェイルオーバーを作成する

Amazon Route 53 の使用の詳細については、以下のリンク先を参照してください。

- 既存のドメインの DNS サーバーを Amazon Route 53 に移行する
- 親ドメインを移行しないで Amazon Route 53 を DNS サービスとして使用するサブドメインを作成する
- 親ドメインを移行しないでサブドメインの DNS サービスを Amazon Route 53 に移行する

公開されたリソースを保護する

アプリケーションへのインターネットエントリポイントを減らすことができない状況では、正規のエンドユーザートラフィックを中断させることなく、これらのエントリポイントへのアクセスを制限し、保護するための追加の対策が必要です。この制御と柔軟性を提供できる 3 つのリソースは、Amazon CloudFront、Amazon Route 53、およびウェブアプリケーションファイアウォールです。

Amazon CloudFront

Amazon CloudFront は、コンテンツへのアクセスを制限するための 2 つのメカニズム（Geo Restriction およびオリジンアクセスアイデンティティを備えています。

Geo Restriction

Amazon CloudFront では、特定の地理的場所からのコンテンツへのアクセスを制限する Geo Restriction（地域制限とも呼ばれます）がサポートされます。エンドユーザーがコンテンツをリクエストすると、通常 Amazon CloudFront はリクエストの発信場所に関係なく、リクエストされたコンテンツを提供します。ただし、エンドユーザーが特定の国にいる場合、Geo Restriction を使用して、通常はエンドユーザーをサポートしない他の国に公開レベルを下げることができます。Geo Restriction では、以下のシナリオがサポートされます。

- ホワイトリストで承認された国に基づいて、コンテンツへのアクセスを許可する
- ブラックリストで拒否された国に基づいて、コンテンツへのアクセスを防止する

Geo Restriction を使用するには、組み込みの Amazon CloudFront 機能を使用するか、国レベルよりも詳細なレベルを提供するサードパーティの位置情報サービスを使用します。組み込みの Amazon CloudFront 機能を使用するには、「コンテンツの地理的なディストリビューションを制限する」に記載されているステップに従います。サードパーティのサービスの設定については、「サードパーティの位置情報サービスを使用する」を参照してください。

オリジンアクセスアイデンティティ (OAI)
通常、Amazon S3 バケットを Amazon CloudFront のオリジンとして使用している場合、バケット内のオブジェクトの読み取りアクセス権限をすべてのユーザーに付与していることになります。つまり、Amazon CloudFront の URL または Amazon S3 の URL のいずれかを使用して、だれでもコンテンツにアクセスできます。Amazon CloudFront は Amazon S3 の URL を公開しませんが、アプリケーションサーバーが何らかのコンテンツを Amazon S3 から直接供給したり、直接リンクが公開されたりした場合、攻撃者は Amazon S3 の URL を見つけることができます。

特別な Amazon CloudFront ユーザーを意味する OAI を作成することで、Amazon S3 へのアクセスを制限できます。Amazon CloudFront に対するアクセス権限を OAI に付与し、その他のすべてのアクセス権限を除くように、Amazon S3 アクセス権限を変更します。エンドユーザーが Amazon CloudFront を使用して Amazon S3 オブジェクトにアクセスすると、OAI はユーザーの代わりにオブジェクトを取得します。Amazon S3 の URL を使用してオブジェクトにアクセスしようとした場合は、アクセスは拒否されます。

ステップ 1: CloudFront OAI を作成し、それをディストリビューションに追加する

ステップ 2: Amazon S3 バケット内のオブジェクトの読み取り権限を OAI に付与する

ステップ 3: S3 バケットのアクセス権限を編集する

Amazon Route 53

Amazon Route 53 には、インフラストラクチャのスケーリングと、DDoS 攻撃への応答を簡単にする 2 つの機能があります。これらの機能は、エイリアスレコードセットとプライベート DNS です。

エイリアスリソースレコードセット
通常の Amazon Route 53 リソースレコードセットとは異なり、エイリアスリソースレコードセットは、DNS 機能に対する Amazon Route 53 独自の拡張機能を提供します。エイリアスリソースレコードセットは、IP アドレスまたはドメイン名の代わりに、Amazon CloudFront ディストリビューション、ELB ロードバランサー、Amazon S3 バケット、または同じホストゾーン内の別の Amazon Route 53 リソースレコードセットへのポインタを格納します。

アマゾンレコードセットは時間を節約し、攻撃中に追加のツールを提供します。たとえば、example.com などのエイリアスリソースレコードセットが ELB のロードバランサー（アプリケーションを実行している複数の EC2 インスタンス間のトラフィックを分散させている）を指し示しているとします。アプリケーションが攻撃された場合、Amazon CloudFront ディストリビューションを指し示すようにエイリアスレコードセットを変更するか、ウェブアプリケーションファイアウォールを実行している、より高いキャパシティーの EC2 インスタンスを持つ異なる ELB ロードバランサーまたはお客様独自のセキュリティツールを指し示すようにエイリアスレコードセットを変更できます。この場合、example.com のリソースレコードセットを含むホストゾーンに変更を加えることなく、Amazon Route 53 は、example.com の DNS 応答に自動的にその変更を反映します。

エイリアスレコードセットを作成すると、常には必要なく、攻撃時のみ役立つ追加のリソースを通じてトラフィックをリダイレクトできる柔軟性が提供されます。エイリアスリソースレコードセットの詳細については、「エイリアスおよび非エイリアスリソースレコードセットの選択」を参照してください。

エイリアスレコードセットを作成するには、「Amazon Route 53 を使用してエイリアスレコードセットを作成する」を参照してください。

プライベート DNS
プライベート DNS により、アプリケーションリソース（ウェブサーバー、アプリケーションサーバー、データベースサーバー、データベースなど）の内部 DNS 名を管理できるようになります。この情報をパブリックインターネットに公開することがありません。たとえば、CNAME を使ってルーティングしたいが、外部には公開する必要がない内部リソースがある場合は、VPC 内でプライベート DNS を使用できます。

また、Amazon Route 53 を使用して、スプリットビュー DNS（別名、スプリットホライズン DNS）を設定できます。同じウェブサイトまたはアプリケーションの内部バージョンと外部バージョンを保持する場合（たとえば、管理ユーザートラフィックとエンドユーザートラフィックを分離する）、同じドメイン名に対して異なる内部 IP アドレスと外部 IP アドレスを返すパブリックホストゾーンとプライベートホストゾーンを設定できます。同じドメイン名のパブリックホストゾーンとプライベートホストゾーンを作成し、両方のホストゾーンに同じサブドメインを作成します。

ステップ 1: プライベートホストゾーンを作成する

ステップ 2: プライベートホストゾーンを一覧表示する

ステップ 3: プライベートホストゾーンに Amazon VPC を関連付ける

ウェブアプリケーションファイアウォール

アプリケーションレイヤーで発生する DDoS 攻撃では、インフラストラクチャ攻撃と比較して、トラフィックのボリュームが低いウェブアプリケーションがよく標的になります。これらのタイプの攻撃を軽減するには、インフラストラクチャの一部にウェブアプリケーションファイアウォール (WAF) を含めます。

WAF は、ウェブトラフィックにルールセットを適用するフィルターとして機能します。通常、これらのルールはクロスサイトスクリプト (XSS) や SQL インジェクション (SQLi) といった弱点に対応しますが、HTTP GET または POST フラッドを軽減して、DDoS に対する弾力性を構築するためにも役立ちます。

HTTP はエンドユーザーとアプリケーション間のリクエスト応答プロトコルとして機能します。ここで、エンドユーザーはデータをリクエスト (GET) し、処理のためにデータを送信 (POST) します。GET フラッドは、高いレートで同じ URL をリクエストするか、アプリケーションからすべてのオブジェクトをリクエストします。POST フラッドは、ログインやデータベース検索などのアプリケーションプロセスをトリガーして負荷をかけます。

WAF には、これらのタイプの攻撃がアプリケーションの可用性に影響しないようにするいくつかの機能があります。その 1 つの機能が HTTP レート制限で、特定の期間にわたりエンドユーザーごとにサポートされる HTTP リクエストのしきい値を確立できます。エンドユーザーがそのしきい値を超えると、WAF は新しいリクエストをブロックまたはバッファリングし、他のエンドユーザーがアプリケーションにアクセスできるようにします。

また、WAF は HTTP リクエストを検査し、通常のパターンに準拠しないものを識別します。たとえば、文字制限を超えているログインを防止したり、全件検索を防止するなどです。アプリケーションレイヤーの DDoS に役立つ他の WAF 機能として、応答がコンピュータでないことを確認するテスト（Completely Automated Public Turning test to tell Computers and Humans Apart (CAPTCHA) テスト）や IP 評価リストがあります。

WAF を AWS インフラストラクチャにデプロイするには、最初に AWS Marketplace で利用可能ないずれかの WAF アプリケーションを選択する必要があります。AWS Marketplace は、AWS のお客様が EC2 インスタンス上で実行するソフトウェアを検索、購入し、そして、インストールすることが可能な、オンラインストアです。WAF ソリューションは、AWS Marketplace「Security」カテゴリ、または、「Web Application Firewall」で検索すると見つけることができます。

WAF ソリューションを選択したら、EC2 インスタンスにソフトウェアをデプロイし、トラフィックに合わせてスケールするようにインスタンスを設定する必要があります。それを行う前に、AWS Marketplace WAF のスケーリングに関する追加の要件について説明します。

すべての HTTP リクエストを検査するため、WAF は、アプリケーショントラフィックの流れの中に設置します。残念ながら、これにより WAF が障害点またはボトルネックとなるシナリオが作成されます。この問題を軽減するには、トラフィックスパイク中に複数の WAF をオンデマンドで実行する必要があります。WAF 用のこのタイプのスケーリングは、「WAF サンドイッチ」を通じて行われます。

「WAF サンドイッチ」では、WAF ソフトウェアを実行している EC2 インスタンスは Auto Scaling グループに含まれ、2 つの ELB ロードバランサー間に配置されます。「Elastic Load Balancing」セクションでは、2 つのロードバランサーを作成しました。それは、デフォルト VPC の基本的なロードバランサーと内部ロードバランサーです。デフォルト VPC の基本的なロードバランサーは、すべての着信トラフィックを WAF EC2 インスタンスに分散する、フロントエンドのパブリック側のロードバランサーになります。ELB の背後の Auto Scaling グループで WAF EC2 インスタンスを実行することで、高いレベルでのトラフィックのスパイクが発生したときに WAF EC2 インスタンスは追加され、インスタンスはスケールできます。

トラフィックが検査およびフィルタリングされると、WAF EC2 インスタンスは内部のバックエンドロードバランサーにトラフィックを転送し、このロードバランサーがトラフィックをアプリケーションの EC2 インスタンス間に分散します。この設定（図 5 に示す）により、WAF EC2 インスタンスは、アプリケーションの EC2 インスタンスの可用性に影響することなく、スケールしてキャパシティーの要求を満たすことができます。

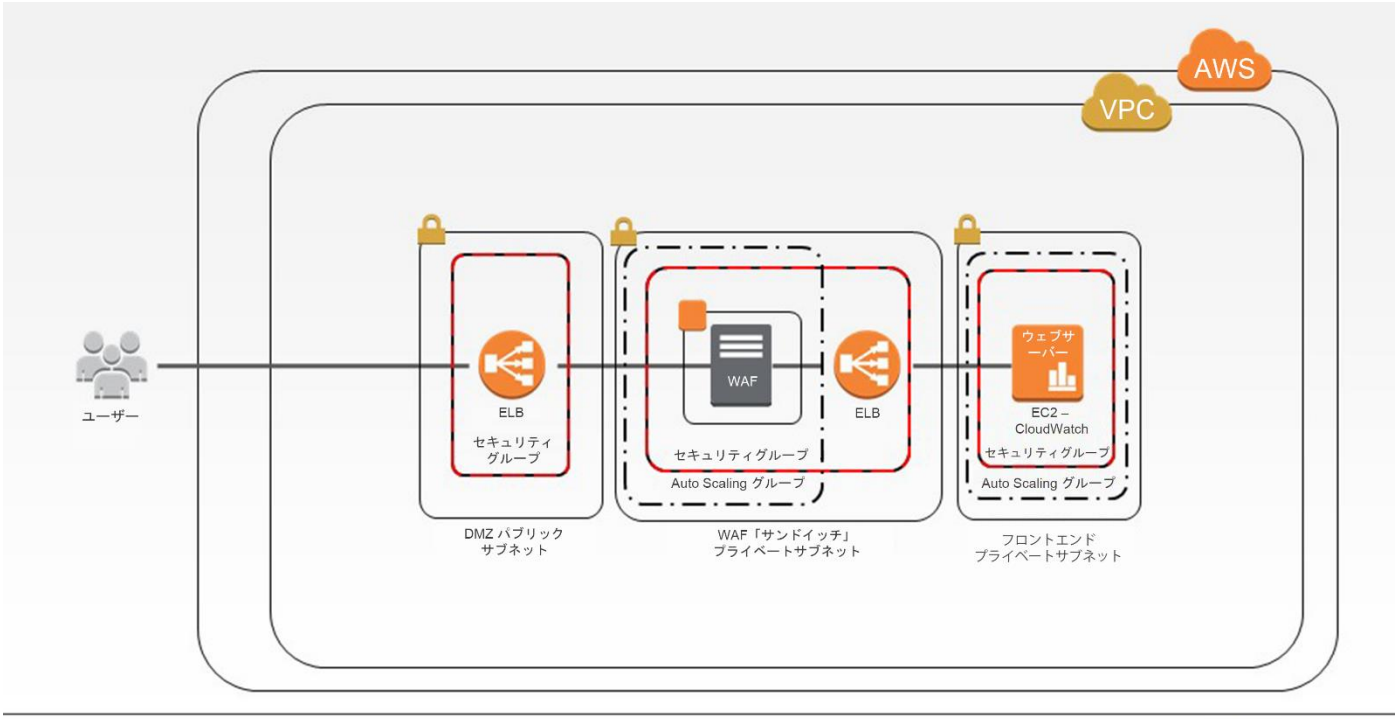

図 5: 「WAF サンドイッチ」

基本的な ELB ロードバランサーと内部的な ELB ロードバランサーは既に作成されているため、次のステップでは 2 つの ELB ロードバランサー間で EC2 インスタンスに WAF をデプロイし、設定します。AWS Marketplace のソリューションで、「WAF サンドイッチ」用の Auto Scaling をサポートするために、異なる構成を採用します。AWS Marketplace から WAF を選択したら、AWS パートナーに連絡し、ソフトウェアのデプロイとスケーリングに関するドキュメントまたは支援をリクエストしてください。AWS プロフェッショナル サービスに連絡して、インフラストラクチャでの WAF のデプロイに関するヘルプを求めることもできます。

通常時の動作について学習する

弾力性のあるアーキテクチャを確保するための 1 つの方法は、アプリケーションやインフラストラクチャが攻撃を受けているときに検知することです。しばしば、お客様はアプリケーションが実行中であるかダウンしているかによって、攻撃の状態を判断します。しかし、より良い戦略では、アプリケーションで予期されているトラフィックのレベルやパターンを理解し、それを異常なレベルやパターンと比較するためのベンチマークとして使用します。

通常のトラフィックパターンを知ることが重要であるのには多くの理由がありますが、DDoS 攻撃での 1 つの理由として、通常、攻撃者はしきい値を判断しようとして、アプリケーションを調査またはテストすることが挙げられます。攻撃者は、何回か攻撃しても成功しない場合、アプリケーションの可用性に影響を与えるために使用できるトラフィックまたはベクターの量を推定できます。こうした状況では、予期できることとそうでないことを知ることで、異常を検出し、それが発生した時は警告して、状況を認識できるようになります。

Amazon CloudWatch

Amazon CloudWatch を使用して、AWS で実行中のインフラストラクチャとアプリケーションをモニタリングすることができます。Amazon CloudWatch はメトリックスとログファイルを収集し、それらのメトリックスが事前に決定されたしきい値を超えたときのアラームを設定できます。また、通常時の 1 日の状況がわからない場合は、Amazon CloudWatch を使用して、リソース使用率、アプリケーションパフォーマンス、およびオペレーションの状態においてシステム全体の可視性を得られます。これらの確認項目を使用して、DDoS 攻撃に対応し、いつ変更が必要か評価することができます。

アラートを作成する前に、アプリケーションの正常な動作について理解する必要があります。このためには、Amazon CloudWatch のメトリックスの表示、選択、グラフ表示を行います。

ステップ 1: 利用可能なメトリックスを表示する

ステップ 2: 利用可能なメトリックスを検索する

ステップ 3: メトリックスを選択する

ステップ 4: メトリックスの統計を取得する

ステップ 5: メトリックスをグラフ化する

Amazon CloudFront レポートと Amazon Route 53 ヘルスチェックには、トラフィックとアプリケーションをより良く理解するために確認できる別のメトリックスがあります。

アプリケーションのベースラインを確立したら、Amazon CloudWatch を使用してそれらのメトリックスでアラームを作成し、攻撃の可能性やその他の異常を警告することができます。次の表に、DDoS 攻撃の推奨のアラートメトリックスを示します。

トピック	メトリックス	説明
Auto Scaling	GroupMaxSize	Auto Scaling グループの最大サイズ。
AWS の請求	EstimatedCharges	AWS 使用量に対する予想請求額。
Amazon CloudFront	Requests	すべての HTTP/S リクエストに対するリクエストの数
Amazon CloudFront	TotalErrorRate	HTTP ステータスコードが 4xx または 5xx であるすべてのリクエストの割合。
Amazon EC2	CPUUtilization	割り当てられた EC2 コンピュートユニットのうち、現在使用されているものの比率。
Amazon EC2	NetworkIn	すべてのネットワークインターフェースでの、このインスタンスによって受信されたバイトの数。
Amazon EC2	StatusCheckFailed	StatusCheckFailed_Instance と StatusCheckFailed_System の組み合わせで、どちらかのステータスチェックが失敗したら報告します。

ELB	UnHealthyHostCount	各アベイラビリティーゾーンの異常なインスタンス数。
ELB	RequestCount	受信され、登録されたインスタンスにルーティングされた、完了したリクエスト数。
ELB	Latency	リクエストがロードバランサーから送信され、応答を受信するまでの経過時間（秒）。
ELB	HTTPCode_ELB_4xx HTTPCode_ELB_5xx	ロードバランサーで生成される HTTP 4XX または 5XX エラーコードの数。
ELB	BackendConnectionErrors	成功しなかった接続の数。
ELB	SpilloverCount	キューがいっぱいなため、拒否されたリクエストの数。
Amazon Route 53	HealthCheckStatus	ヘルスチェックのエンドポイントのステータス。

表 1: 推奨される CloudWatch のメトリックス

これらのデフォルトのメトリックスに加えて、Amazon CloudWatch Logs を使用して、EC2 インスタンスで実行しているアプリケーションからログファイルをモニタリングおよびアクセスできます。Amazon CloudWatch ログは、Amazon Linux、Ubuntu、および Windows で利用できるインストール可能なエージェントであり、ログを CloudWatch に送信します。アプリケーションログに存在するエラーの数をトラッキングし、エラー率が指定のしきい値を超えたときに管理者に通知を送ることができます。

また、アプリケーションログの特定の文字列（例:「NullReferenceException」）をモニタリングしたり、ログデータの特定の場所での任意の文字列（例: Apache アクセスログの「404」ステータスコード）の発生数をカウントしたりできます。目的の語句が見つかると、CloudWatch ログは指定の CloudWatch メトリックスにデータをレポートします。

このセクションでは Amazon Linux または Ubuntu Server を実行する Amazon EC2 インスタンスに CloudWatch ログエージェントをインストールするために必要なステップを説明します。Windows を実行する Amazon EC2 インスタンスで CloudWatch ログを開始するには、『Microsoft Windows インスタンス用 Amazon EC2 ユーザーガイド』の「CloudWatch へのパフォーマンスカウンタの送信と CloudWatch ログへのログの送信」を参照してください。

ステップ 1: クイックスタート: 既存の EC2 インスタンスへ CloudWatch ログエージェントのインストールと設定を行う

ステップ 2: Amazon Simple Notification Service をセットアップする

ステップ 3: アラームを作成する

利用できるアラームの詳細については、「Amazon CloudWatch の名前空間、ディメンション、メトリックスのリファレンス」を参照してください。

攻撃に対する計画を作成する

攻撃されているときに対応戦略を立てても、効果は期待できません。攻撃される前に対応を計画することで、以下のことが確実になります。

- アーキテクチャを検証し、インフラストラクチャにとって有効な手法を選択する。
- 弾力性を高めるためのコストを評価し、防御の目標を理解する。
- 攻撃が発生したときの連絡先を明確にする。

このプランの作成の一部として、必要なレベルのサポートを考慮する必要があります。AWS は、お客様それぞれに合わせたレベルのサポートを提供しています。ただし、DDoS 攻撃の際により高いレベルのサポートを必要とするお客様は、次のメリットに基づいてエンタープライズサポートレベルをご検討ください。

- テクニカルアカウントマネージャ (TAM): TAM は AWS サービス全範囲における技術的な専門知識を提供し、お客様のアーキテクチャの詳細を理解します。TAM は AWS ソリューションアーキテクトと連携して、お客様がベストプラクティスに従うよう支援し、継続的なサポートニーズに関する主要な連絡先となる直通電話を提供します。
- ホワイトグローブケースルーティング: ケースは、緊急の問題を短時間で正確に解決するために、特別なトレーニングを受けたエンジニアに直接転送されます。

これらのサポート機能およびさまざまなレベルのサポートの詳細については、AWS サポートセンターを参照してください。

まとめ

AWS では、DDoS 攻撃に対する耐性を高めるために使用できる数多くの機能やサービスを提供しています。これらを使用し、また必要に応じてその他の手段を使用して、クラウド上のお客様のインフラストラクチャやアプリケーションを保護し、DDoS からの保護に必要な要件を満たすことができます。AWS では、本書のベストプラクティスを使用して、一連のセキュリティポリシーおよび対応を作成することをお勧めします。これにより、DDoS 攻撃に対する耐性を高めることができます。