AWS ホワイトペーパー

AWSクラウドでのウェブアプリケーションのホスティング

AWSクラウドでのウェブアプリケーションのホスティング: AWS ホワイトペーパー

Table of Contents

AWSクラウドでのウェブアプリケーションのホスティング

公開日: 2021 年 8 月 20 日 (改訂履歴)

要約

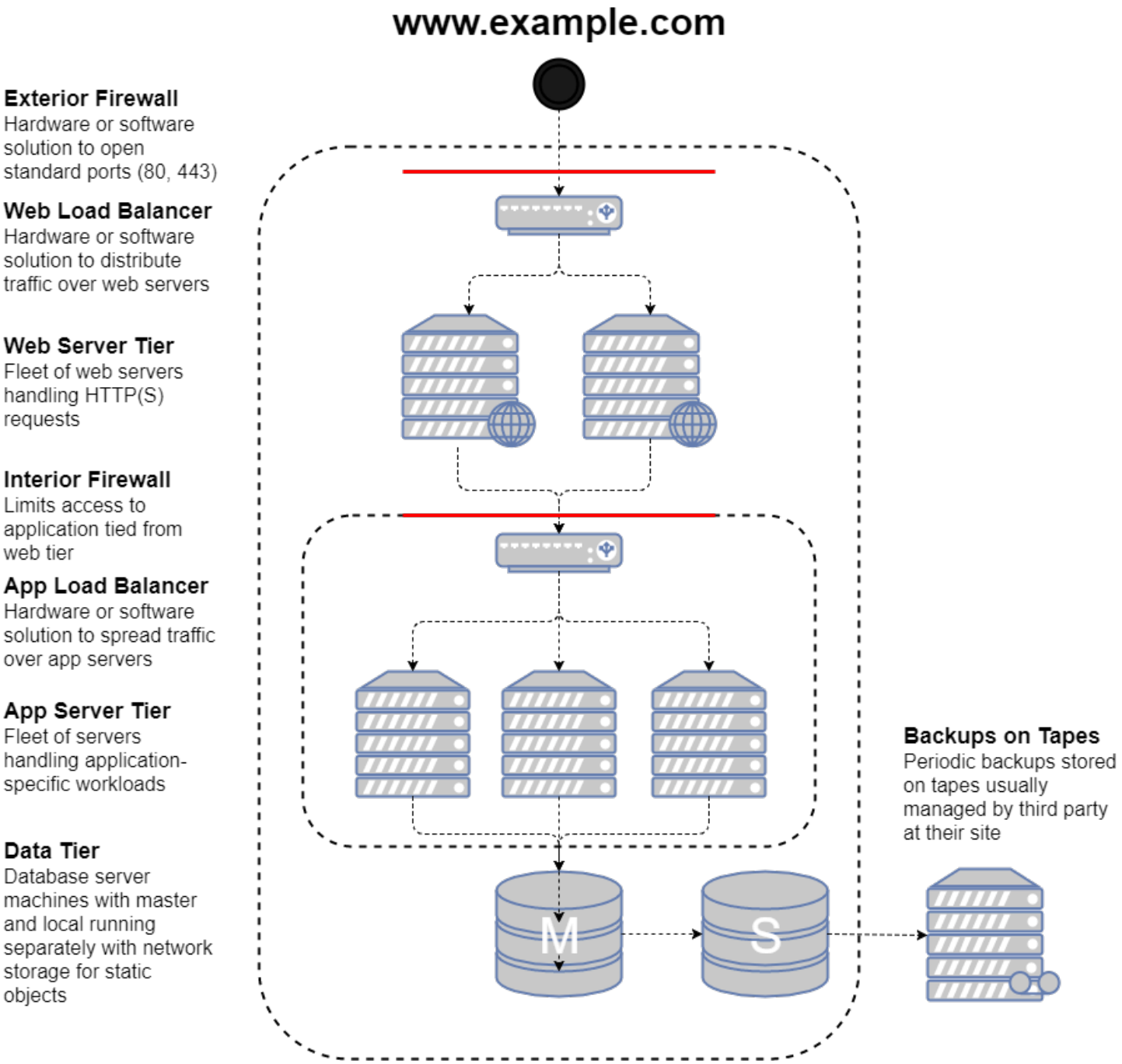
従来のオンプレミスのウェブアーキテクチャでは、信頼性を確保するために、複雑なソリューションと正確なリザーブドキャパシティ予測が必要でした。トラフィックが集中するピーク期間や激しく揺れ動くトラフィックパターンにより、高価なハードウェアの稼働率が低くなることがあります。これにより活用されていないハードウェアを維持するための運用コストが高くなり、あまり使用されていないハードウェアのために、資本の活用が非効率的になります。

アマゾン ウェブ サービス (AWS) は、最も要求度の高いウェブアプリケーションに対して、信頼性が高く、スケーラブルで、安全性に優れ、高い性能のインフラストラクチャを提供します。このインフラストラクチャは、IT コストをほぼリアルタイムの顧客のトラフィックパターンと一致させます。

このホワイトペーパーは、従来のウェブアーキテクチャをクラウドで実行して伸縮性、スケーラビリティ、信頼性を実現する方法を理解したいと考えている IT 管理者およびシステムアーキテクトを対象としています。

従来のウェブホスティングの概要

スケーラブルなウェブホスティングは、よく知られた問題空間です。次の図は、共通の 3 層ウェブアプリケーションモデルを実装する従来型のウェブホスティングアーキテクチャを表しています。このモデルでは、アーキテクチャをプレゼンテーション、アプリケーション、持続性の各レイヤーに分けることができます。これらのレイヤーにホストを追加することで、スケーラビリティが提供されます。また、アーキテクチャには、パフォーマンス、フェイルオーバー、可用性の機能も組み込まれています。従来型のウェブホスティングアーキテクチャは、いくつかの変更だけで、簡単に AWS クラウドに移植できます。



従来型のウェブホスティングアーキテクチャ

以下のセクションでは、このようなアーキテクチャを AWS クラウドにデプロイするべき理由と、その方法について説明します。

AWS を使用したクラウドでのウェブアプリケーションのホスティング

まず問題とすべきなのは、従来型のウェブアプリケーションホスティングソリューションを AWS クラウドに移行する価値についてです。クラウドがお客様に適切であると判断した場合は、適切なアーキテクチャが必要となります。このセクションは、AWS クラウドのソリューションを評価する上で役立ちます。クラウドでのウェブアプリケーションのデプロイをオンプレミスのデプロイと比較し、アプリケーションをホスティングする AWS クラウドアーキテクチャについて示し、AWS クラウドアーキテクチャソリューションの主要コンポーネントについて説明します。

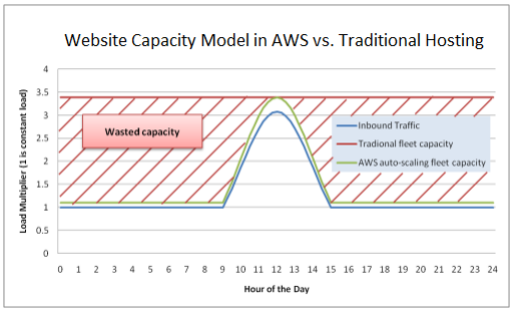
ウェブアプリケーションのホスティングに関する一般的な問題を AWS が解決する方法

ウェブアプリケーションの実行責任者である場合、さまざまなインフラストラクチャとアーキテクチャの問題に直面する場合があります。これらの問題に AWS はシームレスでコスト効率に優れたソリューションを提供しています。以下に、従来型のホスティングモデルよりも AWS の方が優れているいくつかのメリットを示します。

ピーク処理が必要な過剰サイズのフリートに対する費用効果の高い代替案

従来型のホスティングモデルでは、ピーク容量を処理できるサーバーのプロビジョニングを行う必要があります。未使用のサイクルは、ピーク期間外では無駄になります。AWS がホストするウェブアプリケーションでは、追加のサーバーをオンデマンドでプロビジョニングできるため、実際のトラフィックパターンに合わせて容量とコストを常に調整できます。

例えば、次のグラフは、使用量のピークが午前 9 時から午後 3 時までで、その日の残りの時間は使用量が少ないウェブアプリケーションを示しています。実際のトラフィック傾向に基づき必要な時にのみリソースをプロビジョニングするオートスケーリングアプローチは、結果として、無駄な容量が少なくなり、コストも 50% 以上削減されます。



従来型のホスティングモデルで無駄に消費された容量の例

予期しないトラフィックピークを扱うスケーラブルなソリューション

従来のホスティングモデルでプロビジョニングが遅くなると、予期しないトラフィックの急増にタイミングよく対応できないという悲惨な結果になります。人気のあるメディアでサイトが取り上げられた後、トラフィックが予期せず急増したためにウェブアプリケーションが利用できなくなるという話が数多くあります。AWS クラウドでは、通常のトラフィックスパイクに合わせてウェブアプリケーションをスケールするのに役立つ同じオンデマンド機能で、予期しない負荷を処理することもできます。新しいホストはわずか数分で起動して使用できるようになり、トラフィックが通常の状態に戻ったときに素早くオフラインにできます。

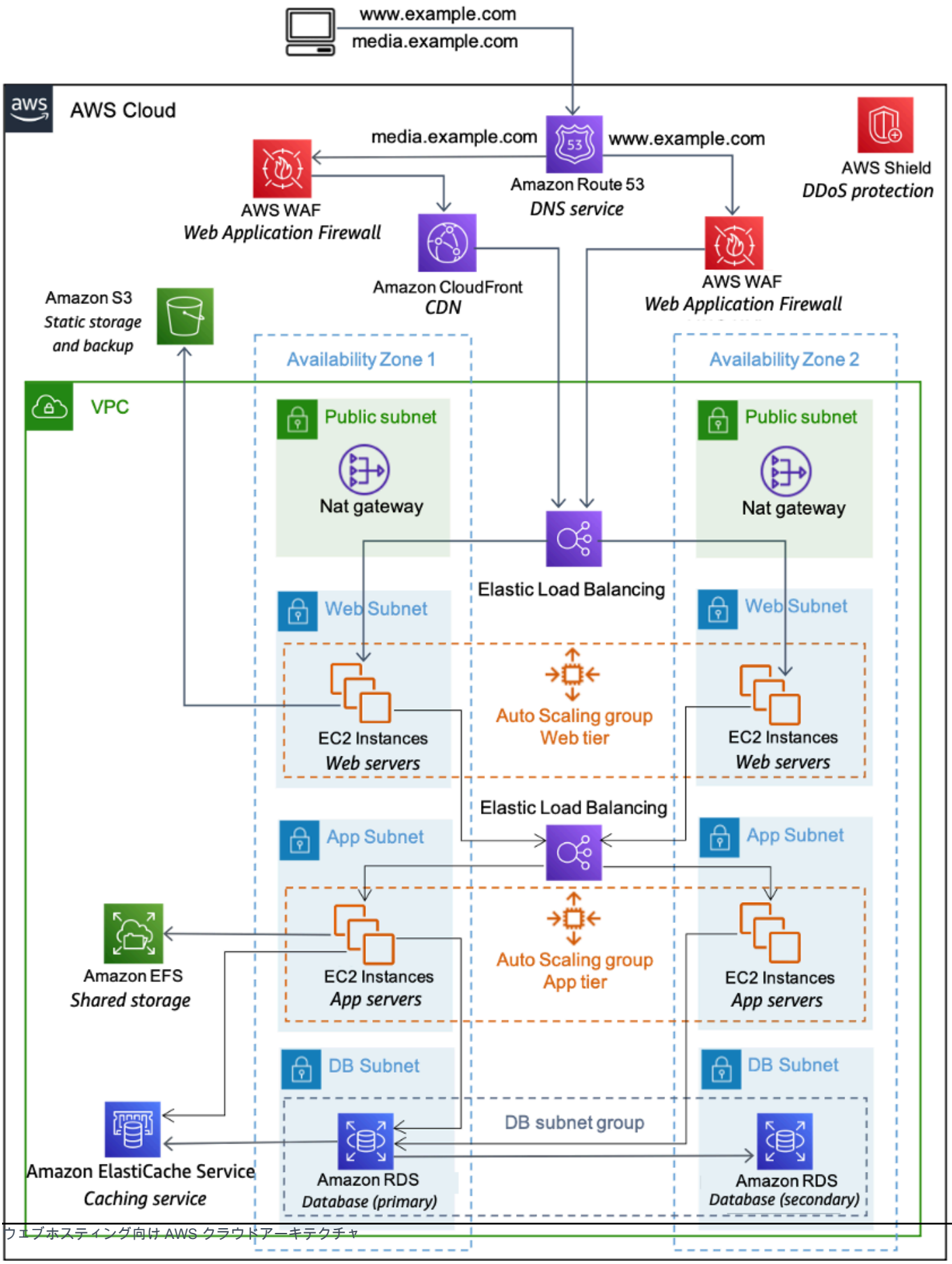
テスト、ロード、ベータ、本番前環境のためのオンデマンドソリューション

本稼働ウェブアプリケーション用に従来型のホスティング環境を構築するハードウェアのコストは、本稼働フリートでは停止しません。多くの場合、開発ライフサイクルの各段階でウェブアプリケーションの品質を確保するために、運用前、ベータ、およびテスト用の各フリートを作成する必要があります。このテストハードウェアを最大限に活用するためにさまざまな最適化を行うことはできますが、これらのパラレルフリートが最適に使用されるとは限りません。また、高価なハードウェアの多くは長期間使用されません。

AWS クラウドでは、テストフリートの必要に応じて、必要なときに、それらをプロビジョンできます。これにより、実際に使用する数日前または数か月前にリソースを事前にプロビジョニングする必要がなくなるだけでなく、必要のないインフラストラクチャコンポーネントを柔軟に破棄できます。さらに、ロードテスト中に AWS クラウドでユーザートラフィックをシミュレートできます。これらのパラレルフリートも、新製品リリースのステージング環境として使用できます。これにより、現在の製品から新しいアプリケーションバージョンに、ほとんど、またはまったくサービスを停止せずに迅速に切り替えることができます。

ウェブホスティング向け AWS クラウドアーキテクチャ

次の図は、従来型のウェブアプリケーションアーキテクチャと、それが AWS クラウドのコンピューティングインフラストラクチャを利用する方法を、もう一度見直しています。


www.example.com
media.example.com
AWS Cloud
media.example.com
www.example.com
AWS Shield
DDoS protection
AWS WAF
Web Application Firewall
Amazon Route 53
DNS service
Amazon CloudFront
CDN
AWS WAF
Web Application Firewall
Amazon S3
Static storage and backup
Availability Zone 1
Availability Zone 2
VPC
Public subnet
Nat gateway
Public subnet
Nat gateway
Elastic Load Balancing
Web Subnet
Web Subnet
EC2 Instances
Web servers
Auto Scaling group
Web tier
EC2 Instances
Web servers
Elastic Load Balancing
App Subnet
App Subnet
Amazon EFS
Shared storage
EC2 Instances
App servers
Auto Scaling group
App tier
EC2 Instances
App servers
DB Subnet
DB Subnet
DB subnet group
Amazon ElastiCache Service
Caching service
Amazon RDS
Database (primary)
Amazon RDS
Database (secondary)

AWS でのウェブホスティングアーキテクチャの例

1. Amazon Route 53 による DNS サービス - ドメイン管理を簡素化するために、DNS サービスを提供します。
2. Amazon CloudFront によるエッジキャッシング - 大容量コンテンツをエッジキャッシュして、お客様のレイテンシーを短くします。
3. AWS WAF による Amazon CloudFront のエッジセキュリティ - クロスサイトスクリプティング (XSS) や SQL インジェクションを含む悪意のあるトラフィックを、お客様が定義したルールでフィルターします。
4. Elastic Load Balancing (ELB) によるロードバランシング - 複数のアベイラビリティーゾーンと AWS Auto Scaling グループに負荷を分散し、サービスの冗長性とデカップリングを実現できます。
5. AWS Shield による DDoS 保護 - 最も一般的なネットワークおよびトランスポートレイヤー DDoS 攻撃に対して自動的にインフラストラクチャを保護します。
6. セキュリティグループを使用したファイアウォール - セキュリティをインスタンスに移動して、ステートフルなホストレベルのファイアウォールをウェブとアプリケーションサーバーの両方に提供します。
7. Amazon ElastiCache によるキャッシング - Redis または Memcached によるキャッシングを行い、アプリケーションとデータベースから負荷を取り除き、頻繁なリクエストに対するレイテンシーを短くします。
8. Amazon Relational Database Service (Amazon RDS) によるマネージドデータベース - 6 つの可能な DB エンジンにより、高い可用性のマルチ AZ データベースアーキテクチャを作成します。
9. Amazon Simple Storage Service (Amazon S3) による静的ストレージとバックアップ - バックアップや画像や動画のような静的アセット用に、シンプルな HTTP ベースのオブジェクトストレージを可能にします。

AWS ウェブホスティングアーキテクチャの主要コンポーネント

以下のセクションでは、AWS クラウドでデプロイされるウェブホスティングアーキテクチャの主要なコンポーネントについていくつか概説し、従来型のウェブホスティングアーキテクチャとの違いを説明します。

ネットワーク管理

AWS クラウドでは、他のお客様のネットワークから自分のネットワークを分離する機能で、より安全でスケーラブルなアーキテクチャが可能になります。セキュリティグループがホストレベルのセキュリティ (ホストセキュリティセクションを参照) を提供する一方で、Amazon Virtual Private Cloud (Amazon VPC) は、お客様が定義する論理的に分離された仮想ネットワークでリソースを起動できます。

Amazon VPC は AWS でのネットワークのセットアップの詳細を完全にコントロールするサービスです。こうしたコントロールの例としては、ウェブサーバーのパブリック側のサブネットの作成や、データベースのインターネット接続がないプライベートサブネットの作成があります。さらに、Amazon VPC では、ハードウェアバーチャルプライベートネットワーク (VPN) を使用してハイブリッド型アーキテクチャを作成でき、また固有のデータセンターの拡張機能として AWS クラウドを使用できます。

Amazon VPC には、お使いのネットワークの従来型の IPv4 サポートに加えて、IPv6 サポートも含まれます。

コンテンツ配信

ウェブトラフィックが地理的に分散している場合、インフラストラクチャ全体を世界中にレプリケートすることは常に実現可能であるとは限らず、費用効果も間違いなく高くありません。コンテンツ配信ネットワーク (CDN) は、エッジロケーションのグローバルネットワークを利用して、動画、ウェブページ、画像などのウェブコンテンツのキャッシュされたコピーを顧客に配信する機能を提供します。CDN は、応答時間を短縮するために、顧客または発信元のリクエストロケーションに最も近いエッジロケーションを利用します。ウェブアセットがキャッシュから配信されると、スループットが劇的に向上します。動的データの場合、多くの CDN を、オリジンサーバーからデータを取得するように設定できます。

CloudFront を使用すると、エッジロケーションのグローバルネットワークを使用して、動的、静的、ストリーミングコンテンツを含むウェブサイトを配信できます。CloudFront は可能な限り最高のパフォーマンスでコンテンツが配信されるように、コンテンツのリクエストを、最も近いエッジロケーションに自動的にルーティングします。CloudFront は、Simple Storage Service (Amazon S3) や Amazon Elastic Compute Cloud (Amazon EC2) など、他の AWS のサービスと連携するように最適化されています。CloudFront はまた、オリジナルの最終ファイルが格納されている AWS 以外のオリジンサーバーともシームレスに連携します。

他の AWS のサービスと同様に、CloudFront を使用するための契約やは月々の最低使用料金は不要です。サービスを通じて実際に配信されたコンテンツの量に応じてお支払いいただくだけです。

また、ウェブアプリケーションインフラストラクチャのエッジキャッシュに対する既存のソリューションは、AWS クラウドでも問題なく動作するはずです。

パブリック DNS の管理

ウェブアプリケーションを AWS クラウドに移行するには、ドメインネームシステム (DNS) を多少変更する必要があります。DNS ルーティングの管理をサポートするために、AWS は、高い可用性を備え、スケーラブルなクラウド DNS ウェブサービスである Amazon Route 53 を提供しています。Route 53 は、「www.example.com」などの名前を、コンピュータが相互に接続するための数字の IP アドレス (192.0.2.1 など) に変換することで、エンドユーザーをインターネットアプリケーションにルーティングできるきわめて信頼性が高く経済的な方法をデベロッパーや企業に提供するように設計されています。Route 53 は IPv6 にも完全準拠しています。

ホストセキュリティ

AWS では、エッジでのインバウンドネットワークトラフィックのフィルタリングに加えて、ウェブアプリケーションではホストレベルでネットワークトラフィックフィルタリングを適用することも推奨しています。Amazon EC2 は、セキュリティグループという名前の機能を備えています。セキュリティグループはインバウンドネットワークファイアウォールと似ており、これに対して、EC2 インスタンスに到達することを許可するプロトコル、ポート、ソース IP 範囲を指定できます。

1 つ以上のセキュリティグループを各 EC2 インスタンスに割り当てることができます。各セキュリティグループは、各インスタンスへの適切なトラフィックを許可します。セキュリティグループは、特定のサブネットまたは IP アドレス、リソースのみが EC2 インスタンスにアクセスできるように設定できます。また、他のセキュリティグループを参照して、特定のグループに属する EC2 インスタンスへのアクセスを制限することもできます。

図 3 の AWS ウェブホスティングアーキテクチャでは、ウェブサーバークラスターのセキュリティグループが、ウェブレイヤーのロードバランサーからのみ、かつポート 80 と 443 (HTTP と HTTPS) の TCP 経由でのアクセスのみを許可します。一方、アプリケーションサーバーのセキュリティグループは、アプリケーションレイヤーのロードバランサーからのアクセスのみを許可します。このモデルでは、サポートエンジニアも EC2 インスタンスにアクセスする必要があります。これは AWS Systems Manager Session Manager で実現できます。セキュリティに関するさらに詳しい説明については、AWS クラウドセキュリティを参照してください。セキュリティ冊子、認定情報、AWS のセキュリティ機能について説明する、セキュリティに関するホワイトペーパーが含まれています。

クラスター全体のロードバランシング

ハードウェアロードバランサーは、従来型のウェブアプリケーションアーキテクチャで使用される一般的なネットワークアプライアンスです。AWS では、Elastic Load Balancing (ELB) サービスを通じてこの機能を提供します。ELB は、受信アプリケーショントラフィックを、EC2 インスタンス、コンテナ、IP アドレス、AWS Lambda 関数、仮想アプライアンスなどの複数のターゲットに自動的に分散させます。変動するアプリケーショントラフィックの負荷を、1 つのアベイラビリティーゾーンまたは複数のアベイラビリティーゾーンで処理できます。Elastic Load Balancing では、4 種類のロードバランサーが用意されています。これらはすべて、アプリケーションの耐障害性を高めるのに必要な高い可用性、オートスケーリング、堅牢なセキュリティを特徴としています。

その他のホストとサービス

従来型のウェブホスティングアーキテクチャでは、ホストの大半が静的 IP アドレスを持ちます。AWS クラウドでは、ホストの大半がダイナミック IP アドレスを持ちます。それぞれの EC2 インスタンスには、パブリックとプライベートの両方の DNS エントリがあり、インターネットを通じてアドレス指定されますが、DNS エントリと IP アドレスは、インスタンスを起動するときに動的に割り当てられます。手動で割り当てることはできません。静的 IP アドレス (AWS 用語で Elastic IP アドレス) は、起動後の実行中インスタンスに割り当てることができます。プライマリデータベース、集中ファイルサーバー、EC2 がホストするロードバランサーなど、一貫したエンドポイントを必要とするインスタンスとサービスでは、Elastic IP アドレスを使用する必要があります。

ウェブアプリケーション内でのキャッシング

インメモリアプリケーションキャッシュは、頻繁に使用される情報をキャッシュすることにより、サービスの負荷を減らし、データベースレイヤーのパフォーマンスとスケーラビリティを向上させることができます。Amazon ElastiCache は、クラウド内でインメモリキャッシュを簡単にデプロイ、運用、スケールできるウェブサービスです。作成するインメモリキャッシュは、負荷に応じて自動的にスケールし、障害のあるノードを自動的に置き換えるように設定できます。ElastiCache は、Memcached と Redis のプロトコルに準拠しており、現在のオンプレミスソリューションからの移行を簡素化します。

データベース構成、バックアップ、フェイルオーバー

多くのウェブアプリケーションには、通常はリレーショナルまたは非リレーショナルデータベース形式で、何らかの永続性が含まれます。AWS では、リレーショナルデータベースサービスと非リレーショナルデータベースサービスの両方を提供しています。また、EC2 インスタンスで独自のデータ

ベースソフトウェアをデプロイできます。次の表に、これらのオプションの概要を示します。詳細はこのセクションで説明します。

表 1 - リレーショナルデータベースソリューションと非リレーショナルデータベースソリューション

	リレーショナルデータベースソリューション	NoSQL ソリューション
マネージドデータベースサービス	Amazon RDS for MySQL、Oracle、SQL Server、MariaDB、PostgreSQL、Amazon Aurora	Amazon DynamoDB、Amazon Keyspaces、Amazon Neptune、Amazon QLDB、Amazon Timestream
セルフマネージド	Amazon EC2 インスタンス上でのリレーショナルデータベース管理システム (DBMS、database management system) のホスティング	EC2 インスタンス上での非リレーショナルデータベースソリューションのホスティング

Amazon RDS

Amazon Relational Database Service (Amazon RDS) では、使い慣れた MySQL、PostgreSQL、Oracle、Microsoft SQL Server データベースエンジンの機能を引き続き利用できます。既に使用しているコード、アプリケーション、ツールを、Amazon RDS で使用できます。Amazon RDS では、データベースソフトウェアに自動的にパッチを当て、データベースをバックアップし、ユーザーが定義した保持期間、バックアップを保存します。また、ポイントインタイムリカバリもサポートします。1 回の API コールで、お客様のリレーショナルデータベースインスタンスに関連するコンピューティングリソースまたはストレージ容量をスケールできるという柔軟性が得られます。

Amazon RDS マルチ AZ 配置により、データベースの可用性が向上し、予期しない停止からデータベースが保護されます。Amazon RDS リードレプリカは、データベースの読み取り専用のレプリカを提供するため、読み取りの多いデータベースワークロードに単一のデータベースデプロイの容量が対応できない場合に、スケールアウトできます。AWS のすべてのサービスと同様に、先行投資は必要なく、お支払いいただくのは使用したリソース分のみです。

Amazon EC2 インスタンス上でのリレーショナルデータベース管理システム (RDBMS、relational database management system) のホスティング

マネージド型 Amazon RDS サービスに加えて、お好みの RDBMS (MySQL、Oracle、SQL Server、DB2 など) を EC2 インスタンスにインストールし、ご自身で管理できます。Amazon EC2 上でデータベースをホストする AWS のお客様は、読み取り専用コピーのミラリング、常時対応のパッシブスレーブのログシッピングなど、さまざまなプライマリ/スタンバイとレプリケーションモデルをうまく使用しています。

Amazon EC2 で独自のデータベースソフトウェアを直接管理している場合、耐障害があり永続的なストレージの可用性も考える必要があります。このため、Amazon EC2 で実行されているデータベースは、ネットワーク接続ストレージに似た Amazon Elastic Block Store (Amazon EBS) ボリュームを使用することをお勧めします。

データベースを実行する EC2 インスタンスでは、データベースのすべてのデータと口グを EBS ボリュームに配置する必要があります。これらは、データベースのホストに障害が発生した場合でも引き続き利用できます。この設定では、ホストに障害が発生した場合に新しい EC2 インスタンスを開始でき、既存の EBS ボリュームを新しいインスタンスに添付できる単純なフェイルオーバーシナリオに対して使用できます。その後、データベースは中断したところから再開できます。

EBS ボリュームは、アベイラビリティーゾーン内で自動的に冗長性を提供します。1 つの EBS ボリュームのパフォーマンスが、データベースニーズに対して十分でない場合、ボリュームをストライピングしてデータベースの 1 秒あたりの入出力オペレーション (IOPS) パフォーマンスを向上させることができます。

ワークロードの要求が厳しい場合は、EBS プロビジョンド IOPS を使用して、必要な IOPS を指定することもできます。Amazon RDS を使用する場合、サービスが独自のストレージを管理するため、お客様はデータの管理に集中できます。

非リレーショナルデータベース

AWS では、リレーショナルデータベースのサポートに加えて、マネージド型の非リレーショナルデータベースも多数提供しています。

- Amazon DynamoDB は、フルマネージド型の NoSQL データベースサービスであり、高速で予測可能なパフォーマンスとシームレスなスケーラビリティが特長です。AWS Management Consoleまたは DynamoDB API を使用すると、ダウンタイムやパフォーマンス低下を発生させずに容量をスケールできます。DynamoDB では、分散データベースの運用とスケーリングに伴う管理作業を AWS に任せることができるため、ハードウェアのプロビジョニング、設定と構成、レプ

リケーション、ソフトウェアパッチ適用、クラスターのスケーリングなどを気にする必要はなくなります。

- Amazon DocumentDB (MongoDB 互換) は、大規模な JSON データ管理のためのデータベースサービスで、AWS 上で完全に管理実行されており、高い耐久性を備えたエンタープライズ対応のサービスとなっています。
- Amazon Keyspaces (Apache Cassandra 用) は、スケーラブルで可用性の高い、Apache Cassandra 互換のマネージドデータベースサービスです。Amazon Keyspaces では、現在使用しているのと同じ Cassandra アプリケーションコードとデベロッパーツールを使用して、AWS で Cassandra ワークロードを実行できます。
- Amazon Neptune は、高速かつ信頼性の高いフルマネージドグラフデータベースサービスです。このサービスでは、高度に接続されたデータセットと連携するアプリケーションを簡単に構築および実行できます。Amazon Neptune の核となるのは、数十億のリレーションシップの保存とミリ秒台のレイテンシーでのグラフのクエリに最適化された、専用の高パフォーマンスグラフデータベースエンジンです。
- Amazon Quantum Ledger Database (Amazon QLDB) (QLDB) は、信頼された機関によって所有されている透過的でイミュータブル、かつ暗号的に検証可能なトランザクションログを提供する、フルマネージド型台帳データベースです。QLDB を使用すると、アプリケーションデータの全変更を追跡し、完全で検証可能な変更履歴を長期間維持できます。
- Amazon Timestream は、IoT および運用アプリケーションに適した、高速かつスケーラブルなサーバーレス時系列データベースサービスです。リレーショナルデータベースの最大 1,000 倍の速度と 10 分の 1 のコストで、1 日あたり数兆ものイベントを、簡単に保存し、分析できます。

さらに、Amazon EC2 を使用して、使用している他の非リレーショナルデータベーステクノロジーをホストすることもできます。

データとアセットのストレージとバックアップ

お使いのウェブアプリケーションのデータとアセットの保存、アクセス、バックアップのために、AWS クラウド内には多くのオプションがあります。Simple Storage Service (Amazon S3) は、可用性が高く、冗長性のあるオブジェクトストアを提供します。Simple Storage Service (Amazon S3) は、画像、動画、その他の静的メディアなど、いくらか静的またはゆっくり変化するオブジェクトのための優れたストレージソリューションです。Simple Storage Service (Amazon S3) はまた、CloudFront とやり取りすることで、これらのアセットのエッジキャッシングとストリーミングもサポートしています。

添付ファイルシステムのようなストレージの場合、EC2 インスタンスには EBS ボリュームを添付できます。これらは、EC2 インスタンスを実行するためのマウント可能なディスクのように動作します。Amazon EBS は、ブロックストレージとしてアクセスする必要があるデータ、およびデータベースパーティションやアプリケーションログなど、実行中のインスタンスの運用期間を超えた永続性を必要とするデータに最適です。

EC2 インスタンスから独立したライフタイムを持つことに加えて、EBS ボリュームのスナップショットを作成して Simple Storage Service (Amazon S3) に保存することもできます。EBS スナップショットは、前回のスナップショット以降の変更のみをバックアップするため、スナップショットを頻繁に行うことでスナップショットにかかる時間を短縮できます。EBS スナップショットは、複数の EBS ボリューム全体でデータをレプリケートし、それらのボリュームをほかの実行中のインスタンスに添付するためのベースラインとしても使用できます。

EBS ボリュームのサイズは最大 16 TB で、複数の EBS ボリュームはさらに大型の容量、または増加した入出力 (I/O) パフォーマンスに対してストリッピングできます。I/O を多用するアプリケーションのパフォーマンスを最大化するために、プロビジョンド IOPS ボリュームを使用できます。プロビジョンド IOPS ボリュームは、I/O を多用するワークロード、特にストレージのパフォーマンスとランダムアクセス I/O スループットにおける一貫性が重要であるデータベースワークロードのニーズを満たすように設計されています。

ボリュームの作成時に IOPS レートを指定すると、Amazon EBS によって、ボリュームの運用期間中においてそのレートがプロビジョニングされます。Amazon EBS は現在、ボリュームあたり最大 16,000 (すべてのインスタンスタイプ) から 64,000 (Nitro System で構築されたインスタンス) までの範囲で、ボリュームごとの IOPS をサポートしています。複数のボリュームをまとめてストライピングして、インスタンスごとに数千の IOPS をアプリケーションに配信できます。これとは別に、スループットが高く、ミリ秒未満のレイテンシーを必要とするミッションクリティカルなワークロードには、最大 256,000 IOPS をサポートできる io2 Block Express ボリュームタイプを使用できます。最大ストレージ容量は 64 TB です。

フリートをオートスケーリングする

AWS クラウドアーキテクチャと従来型のホスティングモデルの間の主な違いの 1 つは、AWS はウェブアプリケーションフリートを自動的にオンデマンドでスケールして、トラフィックの変化に対応できます。従来のホスティングモデルでは、通常、トラフィック予測モデルを使用して、予測されるトラフィックよりも先にホストをプロビジョニングします。AWS では、フリートをスケールアウトおよびスケールインするための一連のトリガーに従って、インスタンスをオンザフライでプロビジョニングできます。

Auto Scaling サービスでは、必要に応じて拡張または縮小できるサーバーの容量グループを作成できます。また、Auto Scaling は、CloudWatch と連動してメトリクスデータを取得し、Elastic Load Balancing と連動してホストの追加と削除を実行し負荷を分散します。例えば、ウェブサーバーが一定期間、80 パーセントを超える CPU 利用率を報告する場合、追加のウェブサーバーが迅速にデプロイされ、自動的にロードバランサーに追加されて、ロードバランシングのローテーションに直ちに含められます。

AWS ウェブホスティングアーキテクチャモデルで示しているように、アーキテクチャの異なるレイヤーに複数の Auto Scaling グループを作成して、各レイヤーが独立してスケールするようにできます。例えば、ウェブサーバーの Auto Scaling グループは、ネットワークの I/O の変化に応じてスケールインやスケールアウトをトリガーし、アプリケーションサーバーの Auto Scaling グループは、CPU の使用状況に従ってスケールアウトやスケールインされるなどです。最小値と最大値を設定することで、24 時間 365 日の可用性を確保し、グループ内での使用量を制限できます。

Auto Scaling トリガーは、特定のレイヤーでフリート全体の拡大と縮小の両方を設定し、リソース使用率を実際の需要に合わせることができます。Auto Scaling サービスに加えて、Amazon EC2 API を介して、Amazon EC2 フリートを直接スケールできるため、インスタンスの開始、終了、検査が可能になります。

追加のセキュリティ機能

分散型サービス妨害 (DDoS) 攻撃の数と巧妙さが増しています。従来、これらの攻撃はかわすのが困難です。多くの場合、攻撃中にウェブサイトへの訪問が失われたことによる機会コストだけでなく、緩和にかかる時間と労力の両方でコストがかかります。このような攻撃に対する防御に役立つ AWS の要素とサービスは数多くあります。その 1 つが、AWS ネットワークの規模です。AWS のインフラストラクチャは非常に大きく、AWS の規模を活用して防御を最適化できます。Elastic Load Balancing、 Amazon CloudFront、 Amazon Route 53 などのいくつかのサービスは、トラフィックの大幅な増加に対応してウェブアプリケーションをスケーリングするのに効果的です。

特にインフラ保護サービスは、お客様の防衛戦略に役立ちます。

- AWS Shield は、さまざまな形態の DDoS 攻撃ベクトルに対する保護に役立つマネージド DDoS 保護サービスです。AWS Shield の標準サービスは無料で、お客様のアカウント全体で自動的に有効になります。この標準サービスは、最も一般的なネットワークおよびトランスポーテ–ションレイヤーの攻撃に対する防御に役立ちます。このレベルに加えて、高度な製品では、進行中の攻撃をほぼリアルタイムで可視化し、前述のサービスとより高いレベルで統合することで、ウェブアプリケーションに対するより高いレベルの保護を提供します。さらに、AWS DDoS Response Team (DRT) と連携することで、お客様のリソースに対する大規模で高度な攻撃を軽減できます。

- AWS WAF (ウェブアプリケーションファイアウォール) は、お客様のウェブアプリケーションの可用性やセキュリティを侵害する可能性がある攻撃、または過剰なリソースを消費を伴う攻撃から保護するように設計されています。AWS WAF は、カスタムルールに従って CloudFront または Application Load Balancer によりインラインで機能し、クロスサイトスクリプティング、SQL インジェクション、DDoS などの攻撃から保護します。ほとんどの AWS のサービスと同様に、AWS WAF にはフル機能の API が備えられており、セキュリティのニーズの変化に従って、AWS WAF インスタンスのルールを作成および編集を自動化する上で役立ちます。
- AWS Firewall Manager は、AWS Organizations でお客様のアカウントとアプリケーションの全体のファイアウォールのルールを一元的に設定および管理できるセキュリティ管理サービスです。新規アプリケーションが作成されると、AWS Firewall Manager はセキュリティルールの共通セットを適用することで、新規アプリケーションとリソースを簡単にこれらに準拠させることができます。

AWS によるフェイルオーバー

従来型のウェブホスティングに対する AWS のもう一つの重要な利点は、冗長なデプロイロケーションに簡単にアクセスできるアベイラビリティーゾーンです。アベイラビリティーゾーンは、物理的に区分された場所で、他のアベイラビリティーゾーンの障害から影響を受けないように設計されています。アベイラビリティーゾーンは、同じ AWS リージョン内の他のアベイラビリティーゾーンに低価格かつ低レイテンシーのネットワーク接続を提供します。AWS ウェブホスティングアーキテクチャの図表で示すように、AWS ではウェブアプリケーションの耐障害性を強化するために、複数のアベイラビリティーゾーン全体に EC2 ホストをデプロイすることをお勧めしています。

障害発生時にアベイラビリティーゾーン間で単一アクセスポイントを移行するための規定があることを確認することが重要です。例えば、データベーススタンバイを 2 番目のアベイラビリティーゾーンに設定して、あまり起こらない障害のシナリオ中でもデータの持続性の整合性が保たれ、可用性が高くなるようにする必要があります。Amazon EC2 または Amazon RDS では、ボタンをクリックするだけでこれを行うことができます。

既存のウェブアプリケーションを AWS クラウドに移行する際には、アーキテクチャ上の変更が必要になることがよくありますが、スケーラビリティ、信頼性、費用対効果が大幅に向上するため、AWS クラウドを利用する価値は十分にあります。次のセクションでは、これらの改善点について説明します。

ウェブホスティングに AWS を使用する際の重要な考慮事項

AWS クラウドと従来型のウェブアプリケーションホスティングモデルの間には、いくつかの重要な違いがあります。前のセクションでは、ウェブアプリケーションをクラウドにデプロイするときに考察すべき多くの重要な分野に焦点を当てました。このセクションでは、アプリケーションをクラウドに移動するときに考える必要がある重要なアーキテクチャ上の変更についていくつか取り上げます。

物理的なネットワークアプライアンスはもうありません

AWS では、物理的ネットワークアプライアンスをデプロイすることはできません。例えば、AWS アプリケーションのファイアウォール、ルーター、ロードバランサーは、物理的デバイスに常駐しなくなり、ソフトウェアソリューションに置き換える必要があります。ロードバランシングの場合、または VPN 接続を確立する場合のいずれでも、さまざまなエンタープライズ品質のソフトウェアソリューションがあります。これは、AWS クラウドで実行できるものの制限ではなく、今日、これらのデバイスを使用する場合のアプリケーションへのアーキテクチャの変更です。

どこでもファイアウォール

これまではシンプルな非武装地帯 (DMZ、demilitarized zone) 機能を持っていて、従来型のホスティングモデルでホスト間のコミュニケーションを開いた場合、AWS は各ホストをロックダウンするという、さらにセキュアなモデルを適用します。AWS デプロイ計画の手順の 1 つに、ホスト間のトラフィック分析があります。この分析により、どのポートを開く必要があるかを正確に判断できます。アーキテクチャのホストのタイプそれぞれに対して、セキュリティグループを作成できます。また、さまざまな種類のシンプルで階層式のセキュリティモデルを作成して、アーキテクチャ内のホスト間でのアクセスを最小限に抑えることができます。Amazon VPC 内のネットワークアクセスコントロールリストの使用は、サブネットレベルでのネットワークのロックダウンに役立ちます。

複数のデータセンターの可用性を検討する

AWS リージョン内のアベイラビリティーゾーンは、複数のデータセンターと考えることができます。異なるアベイラビリティーゾーンの EC2 インスタンスは、論理的にも物理的にも分離されており、高可用性で信頼性の高いアプリケーションをデータセンター全体でデプロイするための使いやすいモデルを提供します。Amazon VPC をリージョンサービスとして使用すると、すべてのリソースを同じ論理ネットワークに保持しながら、アベイラビリティーゾーンを活用できます。

ホストを一時的かつ動的なものとして扱う

おそらく、AWS アプリケーションを構築する方法について最も重要な変更は、Amazon EC2 ホストが一時的かつ動的であると考えなければならないことです。AWS クラウド用に構築されたアプリケーションは、ホストが常に利用可能であると想定してはならず、EC2 インスタンスに障害が発生すると、EC2 インスタンスに保存されたデータは失われるという知識を元に設計する必要があります。

新しいホストが起動するときに、ホストのアベイラビリティーゾーン内の IP アドレスや場所についての仮定は成り立ちません。設定モデルは柔軟でなければならず、ホストをブートストラップするアプローチには、クラウドの動的な性質を考慮に入れる必要があります。これらのテクニックは、非常にスケーラブルで耐障害性に優れたアプリケーションの構築と実行に不可欠です。

コンテナとサーバーレスを検討する

このホワイトペーパーは、主に従来型のウェブアーキテクチャにより重点を置いています。ただし、コンテナやサーバーレスのテクノロジーに移行してウェブアプリケーションをモダナイズすることも検討してください。AWS Fargate や AWS Lambda などのサービスを活用すると、仮想マシンを使用せずにコンピューティングタスクを実行できます。サーバーレスコンピューティングでは、容量のプロビジョニングやパッチなどのインフラストラクチャ管理タスクを AWS が処理するため、イノベーションや変化への適応をより迅速にする、より俊敏なアプリケーションを構築できます。

デプロイの自動化を検討する

- Amazon Lightsail は使いやすい仮想プライベートサーバー (VPS) であり、アプリケーションやウェブサイトの構築に必要なすべてのものに加えて、コスト効率が良い月額プランを提供します。Lightsail は、より単純なワークロード、迅速なデプロイ、および AWS の使用開始に最適です。小規模から始めて、成長に合わせて拡張できるように設計されています。
- AWS Elastic Beanstalk は、Java、.NET、PHP、Node.js、Python、Ruby、Go、および Docker で開発されたウェブアプリケーションやサービスを、Apache、NGINX、Passenger、IIS などの使い慣れたサーバー上にデプロイしたり、スケーリングしたりできる使いやすいサービスです。お客様はコードをアップロードするだけで、Elastic Beanstalk が、デプロイ、容量のプロビジョニング、ロードバランシング、オートスケーリング、アプリケーションのヘルスモニタリングを自動的に処理します。同時に、お客様はアプリケーションが稼動している AWS リソースの完全な制御を保持でき、いつでも基盤となるリソースにアクセスできます。
- AWS App Runner は、デベロッパーによるコンテナ化されたウェブアプリケーションや API の迅速なデプロイを簡単にするフルマネージドサービスです。大規模に、しかも事前のインフラスト

ラクチャ経験を必要とせずにデプロイできます。ソースコードからでも、コンテナイメージからでも始めることができます。App Runner がウェブアプリケーションを自動的に構築およびデプロイし、暗号化しつつトラフィックのロードバランスを実行します。また App Runner は、トラフィックのニーズに応じて自動的にスケールアップまたはスケールダウンします。

- AWS Amplify は、それぞれを連携させたり個別で使用したりできる、ツールとサービスのセットです。これらの機能により、フロントエンドウェブおよびモバイルのデベロッパーが、AWS によるスケーラブルなフルスタックアプリケーションを構築できるようにします。Amplify を使用すると、数分の内にアプリケーションバックエンドを構成し、アプリケーションを接続できます。また、静的なウェブアプリケーションのデプロイは数クリックだけで実行できます。さらに、AWS Management Console の外部でも、簡単にアプリケーションコンテンツの管理が行えます。

結論と寄稿者

まとめ

AWS クラウドへのウェブアプリケーションを検討しているときには、多くのアーキテクチャと概念的な考慮事項があります。ビジネスと共に成長するコスト効率が良く、非常にスケーラブルで、耐障害性に優れているというインフラストラクチャのメリットは、AWS クラウドへの移行の労力を打ち消します。

寄稿者

本ドキュメントは、次の人物および組織が寄稿しました。

- AWS、シニアソリューションアーキテクト、Amir Khairalomoum
- AWS、シニアソリューションアーキテクト、Dinesh Subramani
- AWS、シニアソリューションアーキテクト、Jack Hemion
- AWS、クラウドサポートエンジニア、Jatin Joshi
- AWS、シニアソリューションアーキテクト、Jorge Fonseca
- AWS、ソリューションアーキテクト、Shinduri K S

その他の資料

- Django ベースのアプリケーションを Amazon LightSail にデプロイする
- 高可用性の Drupal ウェブサイトを Elastic Beanstalk にデプロイする
- 高可用性の PHP アプリケーションを Elastic Beanstalk にデプロイする
- DynamoDB を使用して Node.js アプリケーションを Elastic Beanstalk にデプロイする
- AWS クラウドでの Linux ウェブアプリケーション入門
- 静的ウェブサイトをホスティングする
- Amazon S3 を使用した静的ウェブサイトのホスティング
- チュートリアル: Elastic Beanstalk を使用した ASP.NET Core アプリケーションのデプロイ
- チュートリアル: Elastic Beanstalk を使用して.NET サンプルアプリケーションをデプロイする方法

改訂履歴

このホワイトペーパーの更新に関する通知を受け取るには、RSS フィードをサブスクライブしてください。

update-history-change	update-history-description	update-history-date
ホワイトペーパーの更新	新しいサービス、機能、更新されたサービスの制限により、複数のセクションと図表が更新されました。	2021 年 8 月 20 日
ホワイトペーパーの更新	図 3 の「ElastiCache によるキャッシュ」のアイコンラベルが更新されました。	2019 年 9 月 29 日
ホワイトペーパーの更新	新しいサービスの複数のセクションが追加され、更新されました。さらなる明確性とサービスのために図表が更新されました。「ネットワーク管理」における AWS の標準ネットワーキング方式としての VPC の追加。 DDoS 保護と「追加セキュリティ機能」の軽減に関するセクション。 ウェブホスティングのサーバーレスアーキテクチャに関する小さなセクションが追加されました。	2017 年 7 月 1 日
ホワイトペーパーの更新	複数のセクションが更新され、より明確になりました。AWS アイコンを使用する図表が更新されました。Amazon Route 53 の詳細に関する「パブリック DNS	2012 年 9 月 1 日

	の管理」セクションの追加。明確性のために「他のホストとサービスの検索」セクションが更新されました。明確性のためと DynamoDB に関する「データベースの構成、バックアップ、およびフェイルオーバー」セクションが更新されました。「データとアセットのストレージとバックアップ」セクションは、EBS プロビジョンド IOPS ボリュームを含めるように拡張されました。	
初版公開	ホワイトペーパーを公開しました。	2010 年 5 月 1 日

注意

本文書は、情報提供の目的でのみ作成されています。本文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 製品またはサービス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。これらの製品またはサービスは、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。