



AWS を用いた耐障害性の高いアプリケーションの構築

2011 年 5 月

ジェフ・バー、アティラ・ナリン [著]

玉川 憲 [監訳]

序論

ソフトウェアは、全世界のほとんどで毎日の生活に欠かせないものになっています。私達はどこにしようと、携帯電話の利用、現金自動引出機からの現金の引き出し、または信号で停止するだけでも、ソフトウェアと緊密に関係しています。

ソフトウェアはそれほどにまで日常生活に不可欠な要素となっているため、ソフトウェアが機能している状態、利用可能な状態を保つために、非常に多くの努力を払う必要があります。

一般に、この分野の研究は耐障害性(フォールト・トレランス)として知られており、システムを構築するコンポーネントの一部が故障しても、システムが作動可能であり続けることを意味します。

必須であるシステムが常に利用可能であることは当然ですが、それ以上に多くの多彩なソフトウェアも利用可能であって欲しいと願うのが常です。たとえば、製品を購入する E コマースサイトにアクセスします。月曜日の午前 9 時だろうと休日の午前 3 時だろうと、サイトが利用可能で、ショッピングを受け入れてくれることを期待します。E コマースサイトがこうした期待に応えられないと、多数の取引チャンスを失います。繁盛している E コマースサイトの場合は、控え目に見積もっても、利用不可能な期間に毎分何千ドルも失う可能性があります。これが、企業や組織が、故障に耐えられるソフトウェアシステムの開発に力を注いでいる理由のひとつです。

Amazon Web Services (AWS) は、耐障害性の高いソフトウェアシステムを構築するために最適なプラットフォームを提供します。ただし、この性質は、AWS に固有のものではありません。十分なリソースや時間が与えられれば、大半のプラットフォーム上で耐障害性の高いソフトウェアシステムの構築が可能です。AWS プラットフォームでは、最低限の人的作業と、最低限の事前投資により、耐障害性の高いシステムの構築が可能である点がユニークなのです。

故障は関心を集めるべきではない

業務用のデータセンター環境でサーバーがクラッシュした、またはハードディスクの容量がなくなった場合、直ちにシステム管理者に状況が通知されます。それらは、注目すべきイベントであり、システム管理者の作業を中断はせずとも、少なくとも気を引く必要があるからです。従来の業務用データセンター環境における理想的な状況というのは、問題解決のために迅速に行動できる管理スタッフに、故障が確実に通知されることと理解されがちです。多数の組織が、この IT の理想境に到達することは可能です。しかし、通常そうするには、多彩な経験、事前投資、かなりのスタッフが必要となります。

Amazon Web Services が提供するプラットフォームを使用した場合には、このケースにあてはまらないでしょう。理想的には、AWS プラットフォーム上に構築されたアプリケーションの故障はシステム自体によって自動的に対処されます。その結果、極めて退屈なイベントとなります。

Amazon Web Services は、コンピューティング、ストレージ、通信手段などの膨大な量の IT インフラへのアクセスを提供します。これらの IT インフラは、自動的に(または、ほぼ自動的に)割り当てられ、ほとんどの種類の故障に対処します。実際に使用するリソースに対してのみ請求されるため、事前投資が必要ありません。

Amazon Machine Images

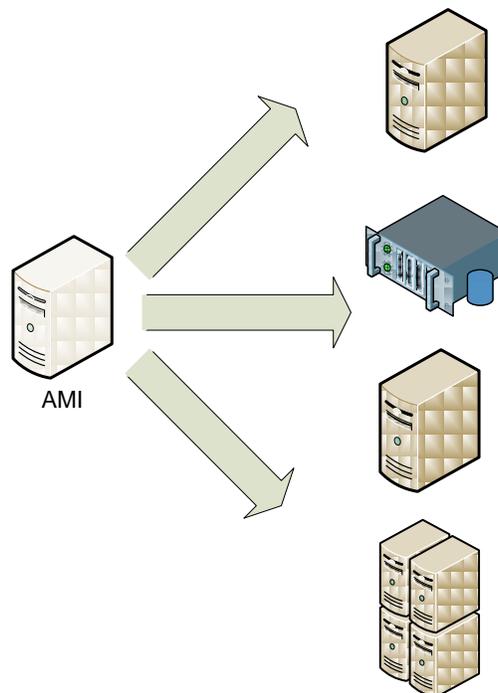
Amazon Elastic Compute Cloud (Amazon EC2) は、Amazon Web Services が提供するウェブサービスの一つであり、ソフトウェアシステムの構築とホスティングのためのコンピューティングリソース、いわゆる、サーバーインスタンスを提供します。Amazon EC2 は、アプリケーション開発を行うための Amazon Web Services の最初の入り口となるでしょう。複数の EC2 インスタンスを使用し、Auto Scaling や Elastic Load Balancing などのツールや補助サービスを利用することで、極めて信頼性が高く耐障害性の高いシステムを構築できます。

Amazon EC2 インスタンスは、みかけは従来のハードウェアサーバーと非常に類似しています。Amazon EC2 インスタンスは、Linux、Windows または OpenSolaris などのなじみ深いオペレーティングシステムを使用します。そのため、これらのオペレーティングシステムで実行可能なほとんどあらゆる種類のソフトウェアを構成することが可能です。Amazon EC2 インスタンスは IP アドレスを備えているため、リモートマシンと更新する通常の方法 (SSH または RDP) が使用可能です。

サーバーインスタンスの定義に使用するテンプレートは、Amazon Machine Image (AMI) と呼ばれます。このテンプレートは、基本的にソフトウェアコンフィギュレーション (オペレーティングシステム、アプリケーションサーバー、アプリケーション) を含んでおり、望みの**インスタンスタイプ**¹を適用することができます。

Amazon EC2 のインスタンスタイプは、基本的にハードウェアの型で、メモリー量 (RAM)、計算処理能力 (CPU 数) に一致するインスタンスタイプをアプリケーションの必要性に合わせて選びます。

単一の AMI を、異なるインスタンスタイプのサーバーリソースの構築に使用することができます。この関係を下で説明します。



¹インスタンスタイプ - <http://aws.amazon.com/jp/ec2/instance-types/>

図 1 – Amazon Machine Image

Amazon は、共通のソフトウェアコンフィギュレーションを含んでいる多数の AMI を公開しています。さらに、AWS 開発者コミュニティの多数のメンバーが独自のカスタム AMI を作成してきました。これらの AMI はすべて、AWS ウェブサイトの Amazon Machine Image リソースページ²に掲載されています。

ただし、AWS 上で耐障害性の高いアプリケーションを構築する最初のステップは、御社の AMI ライブラリの構築です。御社のアプリケーションは、少なくとも1つの AMI で構成されるでしょう。そのアプリケーションを起動するには、AMI を起動するだけで良いのです。

たとえば、御社のアプリケーションがウェブサイト、またはウェブサービスである場合、その AMI はウェブサーバー (Apache または Microsoft Internet Information Server)、関連の静的コンテンツ、すべての動的ページ向けのコードを含んで設定します。代わりに、シンプルな AMI を用いて、インスタンスが起動次第ブートストラップ スクリプトを実行して、すべての必要なソフトウェアコンポーネントとコンテンツをインストールすることもできます。その結果、AMI を起動後、ウェブサーバーが開始し、アプリケーションはリクエストを受け付けを開始します。

いったん AMI を構築すると、故障したインスタンスの交換はとても簡単です。テンプレートと同じ AMI を使用して、代替のインスタンスを起動するだけです。

この操作は、下記で説明するように、スクリプト可能なコマンドラインツール、または AWS Management Console を通じて、API を起動します。この資料の後半で、Auto Scaling サービスを紹介します。このサービスを使用して、故障した、または劣化したインスタンスを、自動的に新しいインスタンスと交換できます。

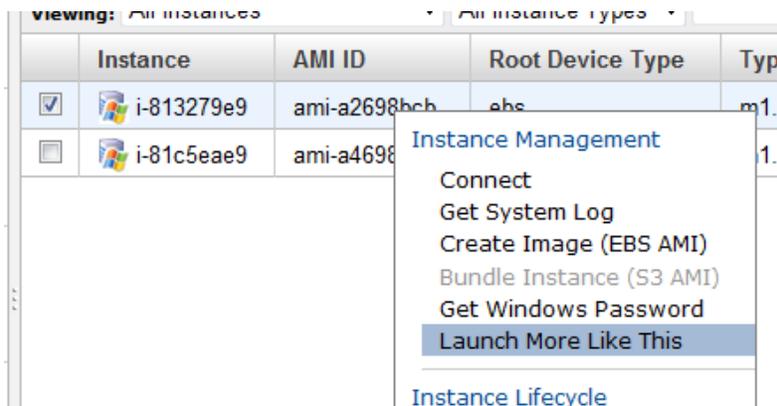


図 2 – Amazon EC2 インスタンスの起動

これは、耐障害性を高めるほんの第一歩です。この時点で、故障から迅速に回復できます。インスタンスが故障した場合、または要求通りに動かない場合、同じテンプレート上にある別のインスタンスを起動できます。ダウンタイムを最小化するには、常にスペアのインスタンスを実行し、故障が生じた時に切り替える用意をします。この操作は、**Elastic IP Address** (監注: Elastic IP Address には、瞬時に割り当てられるので、伸縮自在という意味の Elastic とついているが、実際には、固定のパブリック IP アドレスです) を使用して効率的に実施できます。Elastic IP Address を新しいインスタンスに再度マッピングするだけで、代替インスタンスまたはスペアインスタンスに容易にフェールオーバーが可能です。Elastic IP Address については、本書の後半で詳しく説明します。

² Amazon Machine Images Resources ページ - <http://developer.amazonwebservices.com/connect/kbcategory.jspa?categoryID=171>

定義済みの AMI に基づく代替インスタンスの迅速な起動は、耐障害性の高める重要な第一歩です。次のステップは、サーバーインスタンスがアクセスする永続的データを保管することです。

Elastic Block Store

Amazon Elastic Block Store (Amazon EBS) は、Amazon EC2 インスタンスで使用するためのブロックレベルのストレージボリュームを提供します。Amazon EBS ボリュームは、EC2 インスタンスの運用状況から独立して永続することが可能 (オフインスタンス) なストレージです。

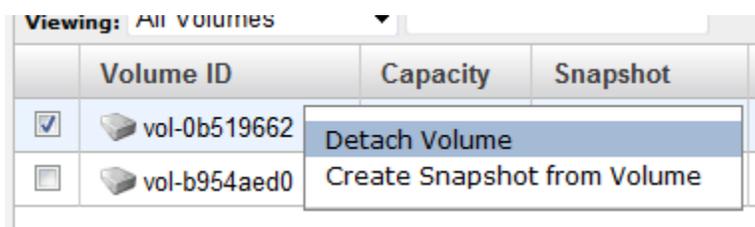
Amazon EBS ボリュームは、基本的には、実行中の Amazon EC2 インスタンスに取り付けることができるハードディスクといえるでしょう。Amazon EBS は、特に、データベース、ファイルシステム、またはブロックレベルストレージへの直接アクセスが必要なアプリケーションに適しています。EBS ボリュームは、データを冗長に保管し、通常のハードドライブよりさらに耐久性を高めています。EBS ボリュームの年間平均故障率 (AFR) は、商用ハードディスクの 4%であるのに比べ、0.1–0.5%です。

AWS プラットフォーム上に耐障害性の高いアプリケーションを構築する際には、Amazon EBS と Amazon EC2 を、一緒に使用することができます。永続する必要があるデータは、各 Amazon EC2 インスタンスに関連する、いわゆる「一時ストレージ(ephemeral storage)」ではなく、Amazon EBS ボリュームに保管します。Amazon EC2 インスタンスが故障し、交換する必要がある場合、Amazon EBS ボリュームを新しい Amazon EC2 インスタンスに取り付けるだけで済みます。この新しいインスタンスは、基本的にはオリジナルの複製であるため、データや機能の損失がありません。

Amazon EBS は、極めて信頼性が高いですが、さらに故障の可能性を軽減するには、**スナップショット**と呼ばれる機能を用いて、EBS ボリュームのバックアップを作成します。強固なバックアップ戦略には、間隔(バックアップ間の時間、一般には毎日。特定のアプリケーションではさらに頻繁に行われる)、保存期間(ロールバックに対するアプリケーションやビジネス要件により異なる)、復旧計画などがあります。スナップショットは、高耐用性を実現する Amazon S3 に保管されます。

スナップショットは、新規 Amazon EBS ボリュームの作成に使用できます。そうして出来た EBS ボリュームは、スナップショットの取得時のオリジナルボリュームの完全な複製です。バックアップは、アプリケーションのオンディスク状態を表すため、スナップショットを実施する前に、インメモリデータをディスクに書き出すように注意が必要です。

Amazon EBS の操作は、以下の図のように、API 経由、また AWS Management Console から実行可能です。



Viewing: All volumes			
	Volume ID	Capacity	Snapshot
<input checked="" type="checkbox"/>	vol-0b519662		Detach Volume
<input type="checkbox"/>	vol-b954aed0		Create Snapshot from Volume

図 3 – Amazon EBS

Elastic IP Address

Elastic IP Address は、特定の EC2 リージョン内で EC2 インスタンスにマップ(経由)が可能なパブリック IP アドレスです。このアドレスは、一度取得するとそれを取得した AWS アカウントに関連することになり、特定のインスタンスやインスタンスの寿命には関連しません。つまり、耐障害性の高いアプリケーションの構築を支援するように設計されているのです。Elastic IP Address は、極めて短い時間内に、故障したインスタンスから取り外して、交換するインスタンスに

紐付けることが可能です。Amazon EBS ボリューム (およびこの問題に関するその他の EC2 リソース) と同様に、Elastic IP Address 上のすべての操作は、API を通してプログラムで実行することもできますし、AWS Management Console から手動で実行できます。



図 4 – Elastic IP addresses

故障は役に立つ

「私は、実際はプログラマーではありません。プログラムが作動するまでいろいろかき集めてから、先に進みます。プロのプログラマーは、「確かに動いているけど、メモリーがいろいろなところにもれている。この問題を解決しなくては。」と言うでしょう。私は、10 回のリクエストのうち 1 回は Apache を再起動します。」

Rasmus Lerdorf (PHP のクリエイター)

実際は認めたくありませんが、ほとんどのソフトウェアシステムは、長い間に品質が低下します。これは、以下の理由のいくつかが原因です。

1. ソフトウェアは、メモリーおよび/またはリソースをリークします。自身で書いたソフトウェアだけでなく、そのソフトウェアが依存するソフトウェア (アプリケーションフレームワーク、オペレーティングシステム、デバイスドライバなど) もリークし得るのです。
2. ファイルシステムが長期間にわたってフラグメントし、パフォーマンスに影響を及ぼす。
3. ハードウェア (特にストレージ) デバイスが、長期間にわたって物理的な質が低下していく。

規律あるソフトウェアエンジニアリングを用いると、こうした問題をある程度軽減しますが、結局は最新のソフトウェアシステムは、制御がきかない沢山のコンポーネント (オペレーティングシステム、ファームウェア、ハードウェア) に左右されます。結局、ハードウェア、システムソフトウェア、御社のソフトウェアのなんらかの組み合わせにより故障が生じ、アプリケーションの可用性を妨害するのです。

従来 of IT 環境では、ハードウェアに定期的に保守サービスを行います。どの程度積極的に行うかは、実際には金銭的な限度があります。しかし、Amazon EC2 を使用すれば、必要なリソースをいつでも終了し、再度作成することが可能です。

AWS プラットフォームの利点を最大限に活かすアプリケーションは、新規サーバーインスタンスで定期的にリフレッシュが可能です。この操作により、全体として潜在的に生じる品質低下がシステムに悪影響を与えないようにできます。ある意味で、故障とみなされるイベント(例えば、サーバーの終了)を、このリソースをリフレッシュするための強制機能として使用しているともいえるでしょう。

このアプローチを利用して、AWS アプリケーションは、サーバーインスタンスの構成要素としてではなく、クライアントに提供するサービスとしてさらに明確に定義することができます。この考えに基づく、サーバーインスタンスそれ自体は、重要ではなく、むしろ簡単に処分できるものとなるのです。

Auto Scaling

コンピューターリソースの自動プロビジョニングや自動スケーリングの概念は、Amazon Web Services プラットフォームでよく設計された耐障害性の高いアプリケーションを稼働させるのに必要不可欠です。Auto Scaling³は、この概念をアプリケーションに極めて容易に適用できる強力なオプションです。

Auto Scaling を使用すれば、Amazon EC2 の容量を自動的にアップまたはダウンできます。必要なのは、サーバーインスタンスがさらに必要となる(または、もっと少なくなる)時期を決定するためのルールを定義することです。

1. 稼働サーバーインスタンス数が、一定数より上昇(または下降)した場合、サーバーインスタンスを起動(または終了)します。
2. サーバーインスタンスのリソースの使用状況(CPU、ネットワークまたはディスク)が一定のしきい値以上(または以下)である場合、サーバーインスタンスを開始(または終了)します。このようなメトリクスは、Amazon EC2 インスタンスをモニターする Amazon CloudWatch サービスから収集できます。

Auto Scaling を使用すれば、サーバーインスタンスを何時でも終了できます。代替インスタンスが自動的に起動されることがわかっているからです。また、ロードの上昇に対応してインスタンスを増加できます。増加したインスタンスがこれ以上必要なくなる時は、自動的に終了します。

これらのルールにより、従来からある多数の冗長性パターンを極めて容易に実践できます。

たとえば、「N+1冗長」⁴は、リソース(データベースなど)を常に利用可能にするための、よく知られた戦略です。「N+1」とは、予想されるロードの取り扱いに N 個のリソースをして十分である場合には、N+1 個のリソースを作動可能にしておくべきであることを示します。

このアプローチは、Auto Scaling にとって最適です。N+1 を Auto Scaling で実践するには、利用可能な任意の AMI の最低2つのインスタンスが常に存在するというルールを定義します。インスタンスを Elastic Load Balancing と関連して使用する場合、各インスタンスは、受信するリクエストの一部を処理することになります。必要に応じて、各インスタンスにおいて、全体のロードを完全に処理するのに十分な空間(未使用の容量)を準備します。1つのインスタンスが故障すると、2つのインスタンスという、最低しきい値に違反するため、Auto Scaling は直ちに代替インスタンスを起動します。Auto Scaling は、正常なサーバーインスタンスが 2 つ利用可能であることを常に確認し続けるのです。

³Auto Scaling は、多数のシナリオに適用できます。本書では、特に故障耐障害性の達成方法の例について説明しています。

⁴http://en.wikipedia.org/wiki/N%2B1_redundancy

Auto Scaling は、自動的に故障を検知し、代替インスタンスを起動するため、インスタンスが予想通りに動かない(低パフォーマンスで実行されている)場合でも、そのインスタンスを終了し、新しいインスタンスを起動します。

Auto Scaling を使用して、メモリークや品質の劣化がアプリケーションに影響しないことを確認するため、定期的にインスタンスを入れ替えることが可能です(入れ替えすべきです)。すなわち、サーバーインスタンスに失効日を設定し、常に「フレッシュ」に保ちます。

「N+1」アプローチを用いることで、リクエストを受け付ける追加のサーバーを持つことが可能でしょう。これにより、アプリケーションは、プライマリサーバーが故障しても、シームレスに移行できます。Amazon EC2 の Elastic Load Balancing 機能は、サーバー間のロードバランスを取る最適な方法です。

Elastic Load Balancing

Elastic Load Balancing は、受信トラフィックを複数の Amazon EC2 インスタンス上のアプリケーションに分散する AWS 製品です。Elastic Load Balancing を使用すると、DNS ホスト名を与えられます。このホスト名に送信されるリクエストは、Amazon EC2 インスタンスのプールに伝達されるのです。

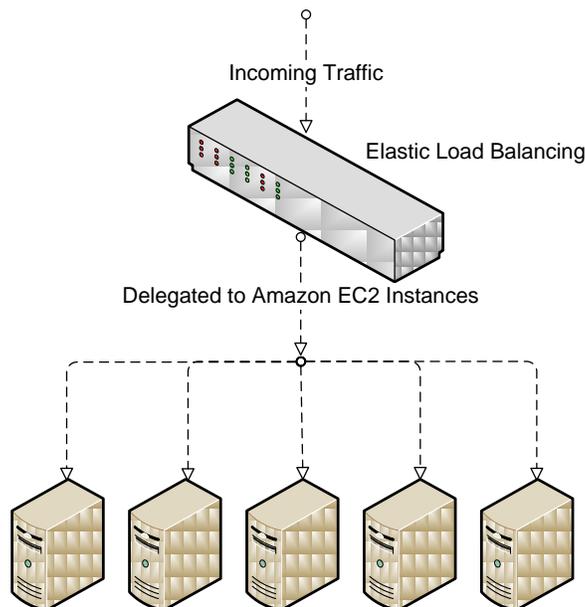


図 5 – Elastic Load Balancing

Elastic Load Balancing はプール内で不具合のあるインスタンスを検出し、不具合が修復されるまで、これらのインスタンスに対するトラフィックを健全なインスタンスに自動的にルート変更します。

Auto Scaling と Elastic Load Balancing は、理想的な組み合わせです。Elastic Load Balancing は、処理のために単一 DNS 名を提供し、Auto Scaling は、リクエストを受諾する正常な Amazon EC2 インスタンスが、常に適度な数存在することを保証するのです。

Reserved Instance

これまでみてきた検討テクニックは、いつでも必要な時に Amazon EC2 インスタンスを調達できるという前提に立っています。

Amazon Web Services には、自由に利用できる大量のハードウェアリソースがありますが、他のクラウドコンピューティング プロバイダと同様に、リソースは有限です。ユーザーがリソースへのアクセスを最大化する最良の方法は、必要なコンピューティング能力の一部を予約することです。この操作は、Reserved Instance と呼ばれる機能で実行可能です。

Reserved Instance を使用すれば、Amazon Web Services クラウドで文字通りコンピューティング能力を予約できます。そうすることにより、価格を低く抑えることができます。ただし、耐障害性の状況でさらに大切なことは、必要なコンピューティング能力を取得するチャンスを最大化できることです。

リージョンと利用可能なゾーン

さらに優れた耐障害性を高めるのに鍵となる要素は、アプリケーションを地理的に分散させることです。単一の Amazon Web Services データベースが何らかの理由で故障した場合、地理的に遠方のデータセンターで同時に実行して、アプリケーションを保護することが可能です。

Amazon Web Services は、地理的な分散しているリージョン (Region) を選択して利用可能です。AWS を使用するとき、データの保管、インスタンスの実行、クエリの開始、データベースのインスタンス化を実行するリージョンを指定できるということです。Amazon EC2 を含む大半の AWS インフラストラクチャサービスは、米国東部 (北部バージニア)、米国西部 (北部カリフォルニア)、EU (アイルランド)、アジアパシフィック (シンガポール)、アジアパシフィック (東京) の 5 リージョンで実施されています (監注: 2011 年 5 月時点)。Amazon S3 の地理的な配置は、若干異なります。米国標準は米国全土のデータセンターを網羅します。その他、米国東部 (北部バージニア)、米国西部 (北部カリフォルニア)、EU (アイルランド)、アジアパシフィック (シンガポール)、アジアパシフィック (東京) のリージョンが存在します。

各リージョンには、Availability Zone (以下、アベイラビリティ・ゾーン) が存在します。アベイラビリティ・ゾーンは、他のゾーンからの影響を受けないように各々独立したロケーションであり、同一リージョンの他のアベイラビリティ・ゾーンに対して待ち時間の少ない安価なネットワーク接続を提供します。異なるアベイラビリティ・ゾーンでインスタンスを起動することにより、アプリケーションをゾーン全体に影響を与える (起こることは稀ですが) 故障から保護できます。

リージョンは、複数のアベイラビリティ・ゾーンから構成され、それぞれ別々のエリアまたは国に分散して、存在しています。Amazon EC2 のサービスレベルアグリーメント (SLA) は、各 Amazon EC2 リージョンにおいて、99.95% の可用性を誇っています。

Elastic IP Address は、複数のアベイラビリティ・ゾーンに及ぶ耐障害性の高いアプリケーションの設計で重要な役割を担っています。フェールオーバーメカニズムを採用することで、IP アドレス (と受信トラフィック) を故障したインスタンスやゾーンからはずし、代替インスタンスに付け直すことができます。

機能をアベイラビリティ・ゾーンに適切に分割することが重要になります。アプリケーションの各ティアの冗長化したインスタンス (ウェブサイト、アプリケーション、データベース) は、異なるアベイラビリティ・ゾーンに配置しなければなりません。望ましい目標は、複数のアベイラビリティ・ゾーンで各アプリケーションスタックの独立したコピーを持つことです。以下のアーキテクチャダイアグラムでは、望ましい機能分割を示しています。

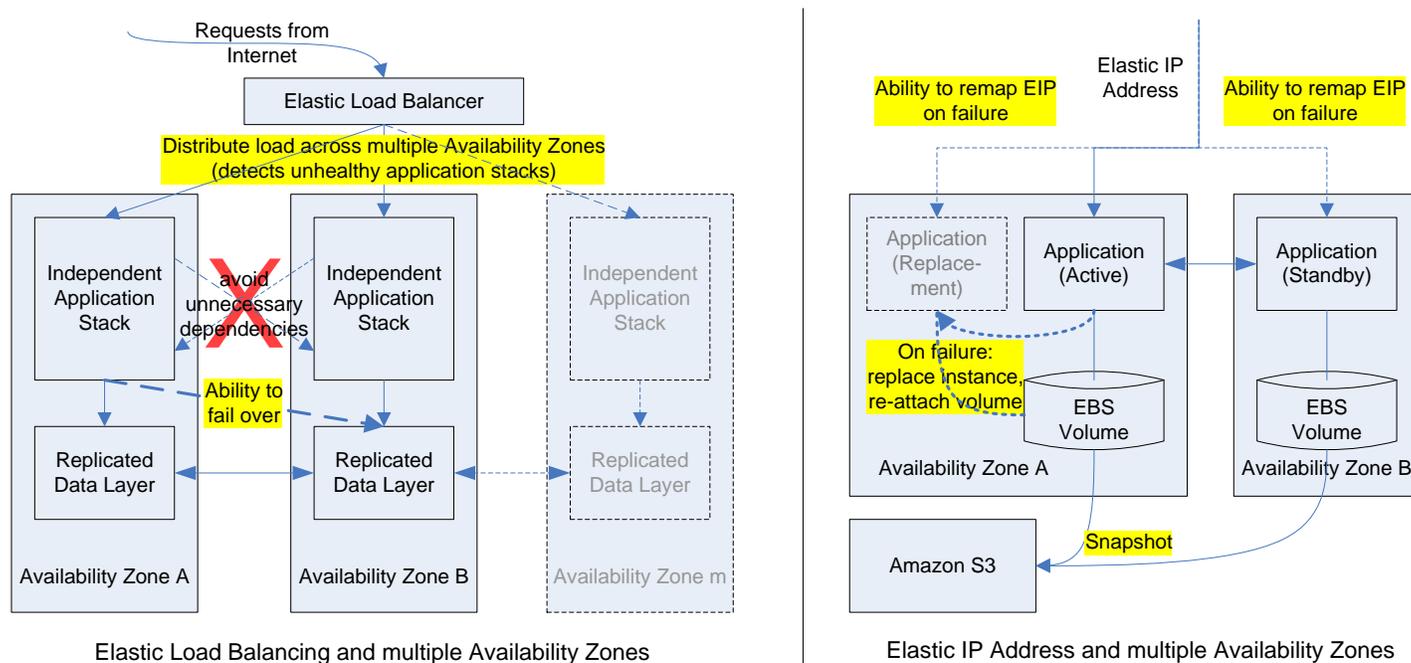


図 6: 複数のアベイラビリティ・ゾーンの活用

複数のアベイラビリティ・ゾーンに Amazon EC2 インスタンスを配置することによって、耐障害性の高いアプリケーションを構築できます。手動介入を少なくして耐障害性を高めるために、Elastic Load Balancing を使用することができます。処理インスタンスを Elastic Load Balancer の背後に配置することにより、耐障害性を改善するのです。Elastic Load Balancer は、複数のアベイラビリティ・ゾーンに存在する複数のインスタンス全体に自動的にトラフィックのバランスをとり、健全な Amazon EC2 インスタンスだけがトラフィックを受け取るよう調整するからです。Elastic Load Balancer を使えば、単一のアベイラビリティ・ゾーンのみでロードバランスすることも、複数のアベイラビリティ・ゾーン全体でロードバランスすることも可能です。Elastic Load Balancing は Amazon EC2 インスタンスの健全さを検出することができます。不具合のあるインスタンスが検出されると、そのインスタンスに対して、トラフィックの割り当ては行われなくなります。代わりに、残りの正常なインスタンス全体に、ロードが分散されます。もし、特定のアベイラビリティ・ゾーンにあるすべてのインスタンスに不具合がある場合、複数のアベイラビリティ・ゾーンにおいて Amazon EC2 インスタンスをセットアップしている場合なら、Elastic Load Balancing は、そのもう一つのゾーン内で正常な Amazon EC2 インスタンスに対してトラフィックのルート割り当てを行いません。不具合のあった Amazon EC2 インスタンスが正常な状態に復帰すると、それらに対するロードバランシングが再開されるでしょう。

耐障害性を高める構成要素

Amazon EC2 とその関連機能は、アプリケーションの導入、構築を行うための、パワフルで経済的なプラットフォームを提供します。ただし、この特色は、全体から見れば、Amazon Web Services のほんの一面です。

Amazon Web Services は、アプリケーション開発に統合することのできる多数の製品を提供します。これらのウェブサービスは、絶対的な耐障害性であり、これらのサービスを使用することにより、自社のアプリケーションの耐障害性が高めることが可能です。

Amazon Simple Queue Service

Amazon Simple Queue Service (SQS) は、耐障害性アプリケーションのバックボーンとして機能する、信頼性が極めて高い分散メッセージシステムです。

メッセージは、ユーザーが作成するキューに保管されます。各キューは URL で定義されるため、キューの Access Control List (ACL) を条件として、インターネット接続があるどのサーバーからでもアクセス可能です。Amazon SQS を使用すれば、キューがいつでも利用可能かどうかを確認できます。キューに送信するメッセージは、最高 4 日間保管されます(または、メッセージが読まれ、アプリケーションにより削除されるまで)。

Amazon SQS を使用する標準的なシステムアーキテクチャ

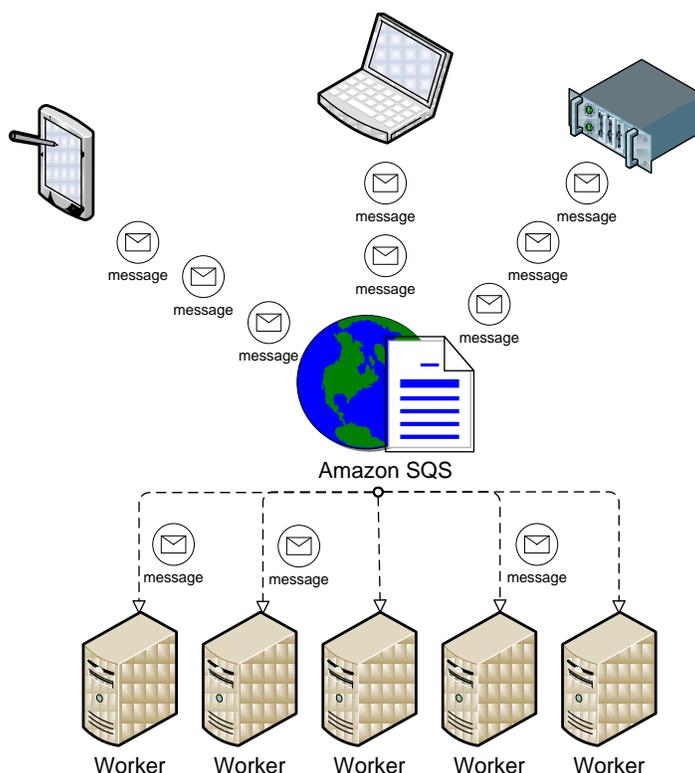


図 7 – Amazon SQS システムアーキテクチャ

この例では、リクエストを受信するために Amazon SQS キューが使用されます。多数の Amazon EC2 インスタンスが、リクエストを探して、絶えずこのキューをポーリングします。リクエストを受信すると、Amazon EC2 のひとつがそのリクエストをピックアップして処理します。このインスタンスがリクエストの処理を終了すると、ポーリングに戻ります。

キューのメッセージ数が増加するか、またはメッセージ処理平均時間が高くなりすぎると、追加の Amazon EC2 インスタンスに作業プロセスをさらに追加するだけで、スケールを拡大できます。

キューからメッセージを消費する「作業プロセス」を実施する EC2 インスタンスが、適度に供給されることを確認するために、Auto Scaling を統合して Amazon EC2 インスタンスを管理するというのは、よくあることです。全ての作業プロセスが落ちてしまうという極端なケースにおいても、Amazon SQS は、受信メッセージを単に保管し続けています。メッセージは最高 4 日間保管されるため、代替 Amazon EC2 インスタンスを起動する時間がたっぷりあります。

メッセージが SQS キューから引き上げられると、**可視タイムアウト**と呼ばれる設定可能な時間間隔の間、その他の消費者には表示されません。消費者がメッセージを処理した後、キューからメッセージが削除されます。可視タイムアウト機能で指定された時間間隔が過ぎた後、メッセージが削除されていないと、再びキューに表示され、別の消費者が引き上げ、処理することが可能です。この二相モデルは、消費アプリケーションがメッセージを処理中に失敗しても、キューアイテムが失われないことを保証します。

Amazon Simple Storage Service

Amazon Simple Storage Service (Amazon S3) は、高い耐久性、高い耐障害性を持つデータストレージを提供する、一見非常にシンプルなウェブサービスです。Amazon Web Services が、可用性と耐障害性を維持する責任を負います。ユーザーは、使用するストレージの料金を支払うだけです。

Amazon S3 は、Amazon S3 リージョンの複数の施設にある複数のデバイスに冗長にオブジェクトを保管するため、Amazon Web Service データセンターが故障した場合でも、データにアクセスできます。

Amazon S3 は、アプリケーションが持ついずれの種類のオブジェクトデータ保管にも最適です。Amazon S3 は、Amazon SQS と同様に URL でアクセスするため、インターネット接続があるあらゆるコンピューティングリソースから S3 を利用できます。

Amazon S3 のバージョンング機能により、S3 に保管されているオブジェクトの以前のバージョンを保存し、アプリケーションの誤った動きによりオブジェクトを間違えて削除した場合でも、オブジェクトを保護します。Versioning は、どの S3 バケットでも有効にできます。

Amazon S3 を使用すれば、耐障害性の重要な面である、データストレージの責任を Amazon Web Service に委任できます。

Amazon SimpleDB

Amazon SimpleDB は、高い耐障害性、高い耐久性を持つ構造化されたデータストレージソリューションです。Amazon SimpleDB を使用すれば、データに属性を付加して保存でき、属性値に基づいてクエリを実行できます。多数の使用シナリオにおいて、Amazon SimpleDB は、MySQL や Microsoft SQL Server などの従来のリレーショナルデータベースを補ったり、時に、リレーショナルデータベースと置き換えて使用することができます。

Amazon SimpleDB は、Amazon S3 やその他のサービスと同様に、非常に利用価値が高い製品です。Amazon SimpleDB を使用して、高い可用性と高い耐障害性で設計されたスケーラブルなサービスを有効活用することが可能です。Amazon SimpleDB に保管されているデータは、単一故障点を持たず、冗長に保管されるのです。

Amazon Relational Database Service

Amazon Relational Database Service (Amazon RDS) は、クラウド上でリレーショナルデータベースを容易に実行できるウェブサービスです。耐障害性と高可用性のアプリケーションの構築という観点では、Amazon RDS は、クリティカルなデータベースの信頼性を向上するためのいくつかの機能を提供します。

RDS では、データベースを自動的にバックアップするため、データベースインスタンスを任意の時点で復旧できます。Amazon RDS は、お客様のデータベースとトランザクションログをバックアップし、ユーザが指定した保持期間、両方を保存しています。この機能は、デフォルトでご利用になれます。

Amazon EBS スナップショットと同様に、Amazon RDS を使用して DB インスタンスのスナップショットを取得できます。この完全なバックアップは、お客様が明示的にそれらを削除するまで、Amazon RDS によって保存されます。望めばいつでも、DB スナップショットから新しい DB インスタンスを作成することができます。これにより、オペレータのミス、アプリケーションのバグ等の意図しないデータ変更などの高レベルの故障から回復するのに役立ちます。

また、Amazon RDS Multi-AZ 配備をサポートします。この機能が有効になると、別のアベイラビリティ・ゾーンデータベースの同期スタンバイレプリカが設定されます。両方のデータベースの同期を保つために、DB インスタンスへのアップデートがアベイラビリティ・ゾーンをまたがってスタンバイに同期的に複製されます。フェールオーバーが実施される際には、スタンバイはプライマリに昇進し、データベースの運用を継続します。DB インスタンスを Multi-AZ 配備として実行することにより、DB インスタンスのコンポーネントの故障や1つのアベイラビリティゾーンでのサービス故障といった予期せぬイベント時に、お客様のデータを守ることができます。

結論

Amazon EC2 は、アプリケーション開発の入り口です。サーバーインスタンスは従来のサーバーと概念上極めて似ています。このため、クラウド向けのアプリケーション開発の学習曲線を大幅に削減します。ただし、従来のハードウェアのサーバーインスタンスと同様の方法で Amazon EC2 サーバーインスタンスを使用するのは、開始点に過ぎません。このままでは、耐障害性、パフォーマンス、全体的なコストを大幅に改善することはないでしょう。

Amazon Web Services プラットフォームの完全な利点は、Amazon EC2 やその他の Amazon Web Services 製品を統合するときに発揮されます。

Amazon EC2 を用いて故障性の高いアプリケーションを構築するには、代替インスタンスを迅速に稼働できるようにする、永続性のあるデータ保管のために Amazon EBS を使用する、複数のアベイラビリティ・ゾーンや Elastic IP Address を利用する、などのベストプラクティスに従うことが重要です。

Auto Scaling を使用すれば、サービスを監視するのに必要な時間やリソースを大幅に削減できます。故障が生じた場合、代替インスタンスが自動的に起動します。サーバーに問題があると診断されると、サーバーが停止し、Auto Scaling が新しいインスタンスを起動します。

Elastic Load Balancing を使用すれば、ユーザーのアプリケーションに、単一の公開エンドポイントを作成します。Amazon EC2 インスタンスの起動、故障、停止、再開は、ユーザーから見えないところで行われるでしょう。

Amazon SQS、Amazon S3、Amazon SimpleDB は、アプリケーションに統合可能なより高いレベルの構成要素といえるでしょう。これらのサービス自体が、高い耐障害性を達成している好例であり、これらを組み込むことで、御社のアプリケーションの耐障害性を高めることが可能です。Amazon RDS を使用すれば、自動バックアップ、スナップショット、Multi-AZ 配備を含む、耐障害性の高いデータベースに簡単にアクセスすることができます。

そして何よりも、Amazon Web Services の価格設定は、実証実験を行うことを可能にします。事前投資は必要ありません。使用した分だけお支払い頂くだけで良いのです。Amazon Web Services プラットフォームのある側面が御社のアプリケーションに適応しないことが判明したら、できるだけ速やかに使用を停止することで、投資を抑えることができます。

Amazon Web Services は、パワフルで洗練されたサービス、そして、その経済的な透明性を提供します。耐障害性の高いソフトウェアを構築するという観点では、最適なプラットフォームに違いありません。

詳細情報

1. Elastic IP と アベイラビリティ・ゾーンのベストプラクティス – http://support.rightscale.com/09-Clouds/AWS/02-Amazon_EC2/Designing_Failover_Architectures_on_EC2/00-Best_Practices_for_using_Elastic_IPs_%28EIP%29_and_Availability_Zones
2. Amazon のアベイラビリティ・ゾーンを使用して耐障害性の高いサイトを設定 – <http://blog.rightscale.com/2008/03/26/setting-up-a-fault-tolerant-site-using-amazons-availability-zones/>
3. Scalr – <https://scalr.net/login.php>
4. Scalr と Amazon Web Service を使用してバーチャルデータセンターを構築 – <http://scottmartin.net/2009/07/11/creating-a-virtual-datacenter-with-scalr-and-amazon-web-services/>
5. Amazon Elastic Load Balancing – <http://aws.amazon.com/jp/elasticloadbalancing/>
6. Auto Scaling Service – <http://aws.amazon.com/jp/autoscaling>
7. インスタンスタイプ – <http://aws.amazon.com/jp/ec2/instance-types/>
8. Elastic Block Store – <http://aws.amazon.com/jp/ebs/>
9. Amazon Machine Images Resources – <http://developer.amazonwebservices.com/connect/kbcategory.jspa?categoryID=171>
10. Amazon Relational Database Service – <http://aws.amazon.com/jp/rds/>