MarkLogic サーバー

MarkLogic Server on Amazon EC2 Guide

MarkLogic 8
2015 年 2 月

最終更新：8.0-1、2015 年 2 月

Copyright © 2015 MarkLogic Corporation.All rights reserved.

目次

MarkLogic Server on Amazon EC2 Guide

1.0 EC2 上の MarkLogic サーバーの概要 4
1.1 注意：作業を行う前に 4
1.2 EC2 用の MarkLogic サーバーについて 4
1.2.1 Amazon EC2 の用語 5
1.2.2 管理対象クラスタ機能 7
1.2.3 CloudFormation の外部での MarkLogic AMI の起動 9
1.3 HealthCheck アプリケーションサーバー 10
1.4 一般的なアーキテクチャ 11

2.0 EC2 上の MarkLogic サーバーの基本的な使用方法 14
2.1 セキュリティ 14
2.2 導入手順のまとめ 15
2.3 Amazon EC2 アカウントの作成 15
2.4 EC2 AMI 用の MarkLogic サーバーの有効化 16
2.5 初期セットアップ手順 16
2.5.1 AWS Management Console へのアクセス 17
2.5.2 IAM ロールの作成 18
2.5.3 キーペアの作成 23
2.5.4 SNS（Simple Notification Service）トピックの作成 23
2.6 AWS 設定変数 26

3.0 CloudFormation を使用した EC2 での MarkLogic の導入 30
3.1 概要 30
3.2 導入と起動 32
3.3 AWS Console を使用した CloudFormation スタックの作成 35
3.4 AWS コマンドラインインターフェイスを使用した CloudFormation スタックの作成 45
3.5 CloudFormation テンプレートのサンプル 46
3.5.1 Parameters 宣言 46
3.5.2 Mappings 宣言 50
3.5.3 Resources 宣言 53
3.5.4 Outputs 宣言 63
3.6 CloudFormation スタックの削除 64

4.0 EC2 上の MarkLogic サーバーの管理 65
4.1 MarkLogic サーバーインスタンスへのアクセス 65
4.1.1 ELB を介した MarkLogic サーバーへのアクセス 66

4.1.2 インスタンスのパブリック DNS を介した MarkLogic サーバーへのアクセス 67
4.2 EC2 インスタンスへのアクセス 68
4.3 EC2 エラーの検出 70
4.4 mlcmd スクリプトの使用 70
4.4.1 sync-volumes-from-mdb 71
4.4.2 sync-volumes-to-mdb 71
4.4.3 init-volumes-from-system 72
4.4.4 leave-cluster 72
4.5 S3（Amazon Simple Storage Service）に対応した MarkLogic の設定 73
4.5.1 S3 バケットのセットアップ 74
4.5.2 グループの S3 エンドポイントの設定 74
4.5.3 S3 資格情報の設定 75
4.5.3.1 セキュリティデータベースでの S3 資格情報の設定 75
4.5.3.2 環境変数での S3 資格情報の設定 76
4.5.3.3 S3 アクセスポリシーを使用した IAM ロールの設定 76
4.5.4 フォレストのデータディレクトリでの S3 パスの設定 77
4.5.5 テストを目的とした MarkLogic へのコンテンツのロード 78
4.6 EC2 でクラスタリソースのスケーリング 78
4.7 MarkLogic AMI のアップグレード 79
4.8 モニタリング（CloudWatch）....80
4.9 エンタープライズデータセンターから EC2 への移行 80
4.10 EBS ボリュームの作成およびインスタンスへのアタッチ 81
4.10.1 EBS ボリュームの作成 81
4.10.2 EBS ボリュームのインスタンスへのアタッチ 83
4.11 MarkLogic クラスタの一時停止または終了 84

1.0 EC2 上の MarkLogic サーバーの概要

この章では、MarkLogic Amazon マシンイメージ（AMI）を使用する Amazon Elastic Compute Cloud（EC2）上の MarkLogic サーバーの概要と、Amazon EC2 アカウントの作成方法および EC2 AMI 用 MarkLogic サーバーの注文方法について説明します。この章は、次のセクションで構成されています。

- 注意：作業を行う前に
- EC2 用の MarkLogic サーバーについて
- HealthCheck アプリケーションサーバー
- 一般的なアーキテクチャ

Amazon EC2 の詳細については、次の URL にアクセスして Amazon のドキュメントを参照してください。

http://aws.amazon.com/documentation/

1.1 注意：作業を行う前に

MarkLogic AMI を起動して、AWS 環境内に MarkLogic クラスタまたは単一の MarkLogic インスタンスを作成する方法はいくつかあります。ただし、最初は CloudFormation テンプレートを使用して MarkLogic AMI を起動し、このガイドに記載されている手順に従うことをお勧めします。他の方法は、その後で試してみてください。CloudFormation テンプレートを使用して MarkLogic AMI を起動する方法の詳細については、「CloudFormation を使用した EC2 での MarkLogic の導入」（30 ページ）を参照してください。MarkLogic CloudFormation テンプレートは、http://developer.marklogic.com/products/aws から入手できます。

CloudFormation テンプレートを使用して MarkLogic AMI を起動しない場合は、「管理対象クラスタ機能」（7 ページ）で説明されている管理対象クラスタ機能に自動的にアクセスできません。それでも AMI を起動することはできますが、「CloudFormation の外部での MarkLogic AMI の起動」（9 ページ）で説明されている手順に従う必要があります。

1.2 EC2 用の MarkLogic サーバーについて

MarkLogic では、Amazon Linux と MarkLogic サーバーが事前にパッケージ化された AMI を提供しています。MarkLogic がこのような AMI に追加しているスクリプトを使用すると、MarkLogic サーバーインスタンスを稼動させるために必要な手順を単純化できます。

このセクションでは、次の内容を取り上げます。

- Amazon EC2 の用語
- 管理対象クラスタ機能
- CloudFormation の外部での MarkLogic AMI の起動

1.2.1 Amazon EC2 の用語

次に、このガイドで使用する用語の定義について説明します。

*Elastic Compute Cloud（EC2）*は、API や使用可能なツールとユーティリティを使用して Amazon のデータセンター内でサーバーインスタンスを起動して管理できるようにする web サービスです。Amazon EC2 web サイトには、次の URL からアクセスできます。
http://aws.amazon.com/ec2/

CentOS は、Red Hat Enterprise Linux をベースにした無料のオープンソースオペレーティングシステムであり、コミュニティでサポートされています。

Elastic Load Balancer（ELB）は、複数の EC2 インスタンス間でアプリケーショントラフィックを自動的に分散させて負荷のバランスを取るサービスです。詳細については、http://docs.aws.amazon.com/gettingstarted/latest/wah/getting-started-create-lb.html を参照してください。

*Amazon マシンイメージ（AMI）*は暗号化されたマシンイメージであり、ソフトウェアのインスタンスを起動するために必要なすべての情報が含まれています。MarkLogic サーバーのインスタンスは、ストックの Amazon Linux AMI から作成されますが、MarkLogic と必要な依存関係がプレインストールされています。

タイプ	料金
実稼動	時間単位の EC2 プレミアム料金
BYOL（Bring Your Own License）	追加料金なし

*Elastic Block Store（EBS）*は、Amazon EC2 インスタンス用に専用設計されたストレージのタイプです。Amazon EBS を使用すると、Amazon EC2 インスタンスでデバイスとしてマウントできるボリュームを作成できます。Amazon EBS ボリュームは、何も手を加えずフォーマットされていない外部ブロックデバイスと同様に機能します。このようなボリュームはユーザーが指定したブロックデバイスにアタッチされ、ブロックデバイスインターフェイスを提供します。Amazon EBS ボリュームにはファイルシステムをロードしたり、ブロックデバイスを使用するのと同じようにそれらのボリュームを使用したりできます。Amazon EBS ボリュームは実際のインスタンスとは別に存在し、削除するまで存在し続けます。このため、Amazon EC2 インスタンスの実行状態を維持していなくても、データを格納できます。各 Amazon EBS ボリュームは、最大 1TiB のサイズまで使用できます。

*インスタンス*は、AMI の起動後に実行されるシステムです。インスタンスは障害が発生するか終了されるまで実行し続けます。障害発生時または終了時は、インスタンスのデータを使用できなくなります。起動後のインスタンスは、従来のホストとよく似ています。

アマゾンウェブサービス（AWS）は、Amazon のクラウドコンピューティングサービスです。詳細については、http://aws.amazon.com/ を参照してください。

AWS Cloud Storage（S3）は、アマゾンウェブサービスのインターフェイスであり、これを使用すると、web 上のどの場所からでもいつでも必要な量のデータを格納および取得できます。詳細については、「S3（Amazon Simple Storage Service）に対応した MarkLogic の設定」（73 ページ）および https://aws.amazon.com/s3/ を参照してください。

CloudFormation（CF）は、最初の AWS リソースをプロビジョニングするための AWS CloudFormation サービスです。詳細については、「CloudFormation を使用した EC2 での MarkLogic の導入」（30 ページ）および http://aws.amazon.com/cloudformation/ を参照してください。MarkLogic CloudFormation テンプレートは、http://developer.marklogic.com/products/aws から入手できます。

管理対象クラスタ（Managed Clusters）は MarkLogic の機能であり、AWS の機能と連携して必要な AWS リソースを自動的に作成およびプロビジョニングし、クラスタを管理するために必要な情報を MarkLogic に提供します。詳細については、「管理対象クラスタ機能」（7 ページ）を参照してください。

MarketPlace は AWS のサービスであり、Amazon 上で従量課金制または無料（追加料金なし）のパブリック AMI を公開する目的で使用します。詳細については、https://aws.amazon.com/marketplace を参照してください。

*インスタンスタイプ*は、Amazon EC2 インスタンスのサイズを定義します。MarkLogic サーバーインスタンスのタイプについては、「AWS Console を使用した CloudFormation スタックの作成」（35 ページ）の手順 5 の最後に示されている表を参照してください。

インスタンスストア（*エフェメラルストレージ*と呼ばれることもある）は、容量が固定されたインスタンスのストレージ領域です。インスタンスストアは、永続的なストレージソリューションとしては設計されていません。意図的にまたは誤ってインスタンスが再起動された場合、インスタンスストア上のデータは失われません。ただし、基盤となるドライブに障害が発生した場合やインスタンスが終了された場合は、データが失われます。

*EC2 Compute Unit（ECU）*は、1.0 ～ 1.2 GHz の 2007 Opteron または 2007 Xeon プロセッサと同等の CPU 処理能力を提供します。

*メタデータデータベース*は、1 つあるいは複数の MarkLogic サーバーのクラスタを管理するために必要なすべての設定データを格納してインデックス付けを行うデータベースです。AWS では、SimpleDB サービスを使用してメタデータデータベースを実装します。詳細については、http://docs.aws.amazon.com/AmazonSimpleDB/latest/GettingStartedGuide/Welcome.html を参照してください。

1.2.2 管理対象クラスタ機能

AWS で MarkLogic サーバーを実行する場合は、従来の IT データセンターでは発生しないようないくつかの課題が発生します。ただし管理対象クラスタ（Managed Clusters）機能を利用することで、信頼性、拡張性、高可用性のサポート、およびより困難ないくつかの問題を自動的に処理するツールによって、このような課題を軽減できます。AWS で MarkLogic AMI を起動する場合は、CloudFormation テンプレートを使用し、このガイドの手順に従うことをお勧めします。これは、このようなテンプレートが、確実かつ容易にクラウドを展開できるように特別に設計されている AWS と MarkLogic の機能を活用するうえで役立つためです。

AWS では、MarkLogic のノードまたはクラスタを展開する前に次のことを検討する必要があります。

- インスタンスのホスト名 — EC2 インスタンスの使用は、一意のホスト名を付けることから始まります。何らかの理由（ハードウェアまたはソフトウェアの障害、EC2 の課金を避けるための手動による停止など）によってインスタンスが停止した場合、インスタンスの再起動時には新しいホスト名が付けられます。これは、特にクラスタで（または単一ノードのインストールでも）設定の問題が発生する原因になります。クラスタを停止した後に、再設定せずにオンラインに戻した場合、すべてのホスト名が異なるため、そのノードはクラスタに再び参加しません。

  管理対象クラスタ機能は、ホスト名の変更を自動的に検出し、その変更をクラスタ全体にプロパゲートします。

- 一時的なデータ — EC2 インスタンスが終了するとルートボリュームが解放され、そのボリューム上のすべてのデータが失われます。これにはプレインストールされたソフトウェアや OS の設定も含まれます。 MarkLogic のインストールは、ルートボリューム上ではなく、アタッチされている EBS ボリューム上に置く必要があります。

  管理対象クラスタ機能は、各 EBS ボリューム、それに関連する EC2 インスタンス、およびマウントディレクトリを自動的にトラッキングします。EC2 インスタンスを再起動すると、管理対象クラスタ機能によりボリュームが適切な場所に自動的に再アタッチおよびマウントされ、フォレストおよびデータベースは必ずそれまでの状態が維持されます。

- AWS セキュリティ — EC2 ノード内から AWS サービスにアクセスする必要がある場合は認証が必要です。その方法の 1 つとして、AWS 資格情報を入力し、それを EC2 インスタンスに格納する方法が挙げられます。ただし、この方法は利便性が低く、セキュリティ上のリスクも伴います。管理対象クラスタ機能および CloudFormation テンプレートは IAM ロールを使用するので、AWS 資格情報が漏えいすることはありません。

- インフラストラクチャのセキュリティ保護 — EC2 インスタンスはデフォルトでは公開されています。そのため、EC2 インスタンスを起動して機密データを格納する前に、セキュリティ要件を考慮しておく必要があります。CloudFormation テンプレートでは、簡単に使用を開始できるようにするため、基本的なセキュリティグループが作成されます。このようなセキュリティグループは、各自の特定のセキュリティニーズに合わせて編集できます。
- 揮発性のインスタンス — EC2 インスタンスでは常に障害が発生する可能性があります。これは、オンプレミスのハードウェアにも当てはまりますが、クラウドでは、自分でその場で修正できません。代わりに、AWS の機能を利用して障害が発生したインスタンスを再起動し、常にデータベースの稼動状態を維持する必要があります。CloudFormation テンプレートでは、推奨されるトポロジが、必要に応じて複数のゾーンにわたる AutoScaling グループを使用して作成されます。これによりクラスタが分散され、問題のあるインスタンスが自動的に再起動されます。管理対象クラスタのサポートにより、再起動したインスタンスを確実にクラスタに再参加させることができます。
- データセンターの障害 — データセンターでは障害が発生する可能性があります。また、台風や地震などの予測できない問題が原因になることも考えられます。高可用性が必要な場合は、障害対策計画を作成し、複数のデータセンターにクラスタを分散させて、単一の障害でクラスタがオフラインにならないようにする必要があります。CloudFormation テンプレートでは、耐障害性を最大限に高めるために単一リージョン内の複数のゾーンに分散された AutoScaling グループを使用して、推奨されるトポロジが作成されます。
- 負荷分散とルーティング — MarkLogic クラスタは正常に稼動しているときにのみアクセスできるようにする必要があります。単一ノードの場合でも（または特にクラスタを実行する場合は）AWS ELB を使用することを強くお勧めします。ELB は、正常なノード間でトラフィックを分散させるだけでなく、ハードウェアおよび OS が正常に実行されている場合でも、ノードに問題が発生した場合は AutoScaling グループに自動的に通知します。これにより、トラフィックが正常なノードに転送され、問題のあるノードが終了および再起動されるようにします。

  管理対象クラスタ機能が各サーバー上で提供する稼動状況チェックアプリケーションは、ロードバランサーによって MarkLogic インスタンスが正常に稼動しているかどうかを検出するために使用されます。

- スケールアップとスケールダウン — 処理能力は、大量に必要になる場合もあれば少なくても済む場合もあります。クラウドコンピューティングでは、スケールアップとスケールダウンを実現する基本的なツールが提供されますが、その場合はソフトウェアがプロビジョニングの変更を認識して統合できる必要があります。CloudFormation テンプレートを使用すると、簡単な 1 ステップの処理でクラスタの処理能力をスケールアップしたりスケールダウンしたりできます。

- クラスタでのライセンスの適用 — 従来は、クラスタを立ち上げる場合は、初めてノードを起動した後で各ノードに対して手動でライセンスを適用する必要がありました。管理対象クラスタ機能では、クラスタ内のすべてのノードにライセンスキーが自動的に適用されます。
- クラスタのフォーメーション — MarkLogic クラスタを実行するときは、管理者の資格情報を使用して最初のノードを設定し、その後でクラスタに加わるように残りのノードを設定する必要があります。管理対象クラスタ機能では、これが自動的に処理されるので、手動で操作しなくてもクラスタの（1 つのクラスタでも）稼動準備が整います。
- クラスタの一時停止 — クラウドコンピューティングの大きな利点は、使った分だけ料金を払うという点です。開発用のサーバーを実行している場合や、サイトを 24 時間無休で稼動させる必要がない場合は、クラスタ全体を一時停止し、稼動していないときに EC2 の料金が発生しないようにすることができます。管理対象クラスタ機能と CloudFormation を使用すると、クラスタを素早く再起動してリソースをすべてのデータに再アタッチし、クラスタを停止した状態から稼動させることができます。

管理対象クラスタ機能を使用するうえで CloudFormation は必須ではありません。代わりに他のツールを使用して手動またはプログラムによって AWS リソースを設定することもできます。ただし、強力なクラウド調整ツールや管理ツールがないと作業は困難を伴うことになります。 CloudFormation を使用すると、ニーズに合わせて拡張させることができるシンプルな宣言型のテンプレートを使用して、管理対象クラスタの設定の文書化と実装の両方を実行できます。

また、管理対象クラスタ機能を使用せずに（提供される AMI を使用または使用せずに）MarkLogic を実行する最もシンプルな設定もサポートされています。ただし、この場合は非常に信頼性が低くなるので推奨されません。

1.2.3 CloudFormation の外部での MarkLogic AMI の起動

このセクションでは、CloudFormation や独自の管理ツールまたはサードパーティのクラウドコンピューティングツールを使用せずに、MarkLogic を稼動させなければならない場合に実行する必要がある最小限の手順について説明します。 管理対象クラスタ機能が有効にならないため、この手順は推奨されません。

- OS 固有のインストール手順に従い、MarketPlace で提供されている EC2 AMI または MarkLogic をプレインストールした独自の AMI を起動します。
- 「キーペアの作成」（23 ページ）の説明に従ってキーペアを作成します。 インスタンスを起動するには、キーペア名が必要になります。
- ポート 22（SSH 用）および Admin API 用のポート（8000 ～ 8002）で開かれるセキュリティグループを作成します。場合によっては、独自のアプリケーションま

たは外部クラスタ用に追加のポートを開くようにセキュリティグループを設定する必要があります。

- データ用の EBS ボリュームを作成し、そのボリュームを EC2 デバイス `/dev/sdf` に関連付けます。 MarkLogic は、起動時にこのボリュームを検出し（またはアタッチされるまで待機し）、ファイルシステムを作成して `/var/opt/MarkLogic` としてマウントします。

これにより、ポート 8001 上で管理ページに移動し、MarkLogic インスタンスの設定を続けることができるようになります。

1.3 HealthCheck アプリケーションサーバー

Elastic Load Balancer（ELB）は、各インスタンスに定期的にハートビートを送信し、稼動状況をモニタリングします。MarkLogic サーバーの各インスタンスは、ポート 7997 上で HealthCheck アプリケーションサーバーを実行します。ELB は認証を使用するように設定できないので、HealthCheck アプリケーションサーバーの URL では認証は必要ありません。

1.4 一般的なアーキテクチャ

このセクションでは、EC2 環境内の MarkLogic クラスタの一般的な設定について、いくつかの例を基に説明します。

『*Scalability, Availability, and Failover Guide*』で説明されているように、エバリュエータノード（E ノード）は、集計、計算（ユーザー定義関数を含む）などのデータ処理演算を実行します。またデータノード（D ノード）は、フォレストのデータ操作を管理します。E ノードは、D ノードのセキュリティグループとは別にグループ化でき、デプロイメントによってはこのようなグループ化が推奨されます。

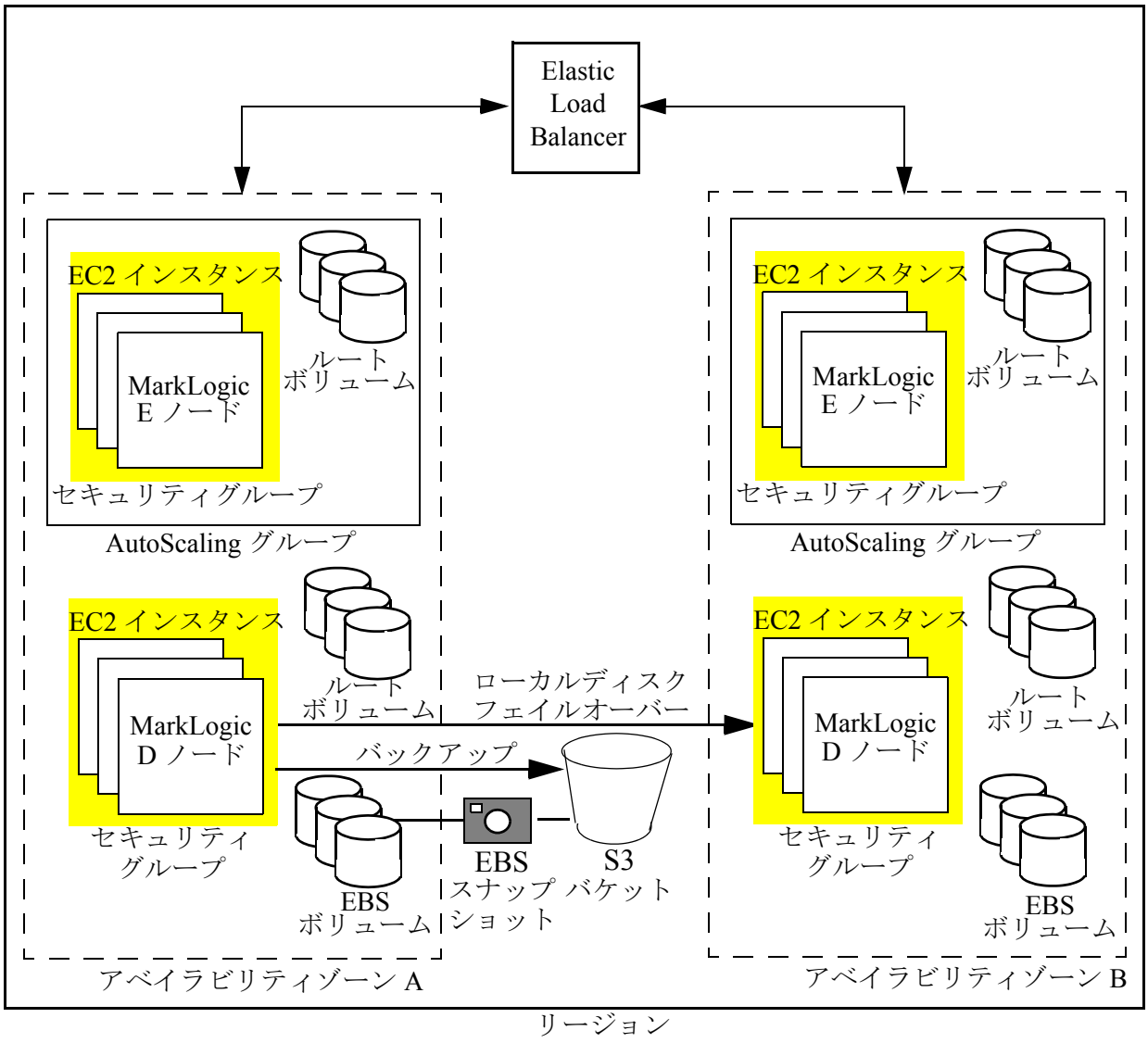
エンドユーザーまたはアプリケーションレベルのクエリは、ロードバランサーを介して E ノードにルーティングする必要があります。より多くのクエリ、ユーザー、計算を処理できるように E ノードを追加してクラスタの処理能力を向上させることができます。

高可用性を確保するには、同じリージョン内の異なるアベイラビリティゾーンに D ノードを配置し、ローカルディスクの障害に対応するように設定して、各トランザクションが 1 つあるいは複数のレプリカに書き込まれるようにします。EC2 では、同じリージョン内のゾーン間のレイテンシは短くなります（約 2 ミリ秒）。最適な可用性を実現するには、D ノードと E ノードを 2 つのアベイラビリティゾーンに均等に分割する必要があります。また、災害復旧を目的として D ノードを異なるリージョンに配置し、各リージョンの D ノード間でデータベースのレプリケーションを使用できます。

ストレージリソースとしては、フォレスト用には EBS ボリュームが、バックアップ用には S3 が推奨されます。すべてのボリュームは、16K のブロックでフォーマットする必要があります。このフォーマットは MarkLogic の大規模なシーケンシャル IO プロファイル用に最適化され、さらに Amazon の pIOPS の実装に合わせたフォーマットです。MarkLogic 上で設定されたフォレストには、それぞれ 20mb/ 秒以上が必要です。つまり 20mb/ 秒と 16K ブロックで、1280 IOPS になります。MarkLogic サーバーの各インスタンスは、最大 5 Hi-IO ボリューム / フォレストに設定する必要があります。また、セキュリティデータベースやスキーマデータベースで使用される EBS など、起動および低 IO フォレスト用の EBS を追加できます。プロビジョンド IOPS ボリューム上にフォレストを作成すると、保守期間中に移行したり、レプリカを使用して稼動中にスクリプトで移行したりできます。

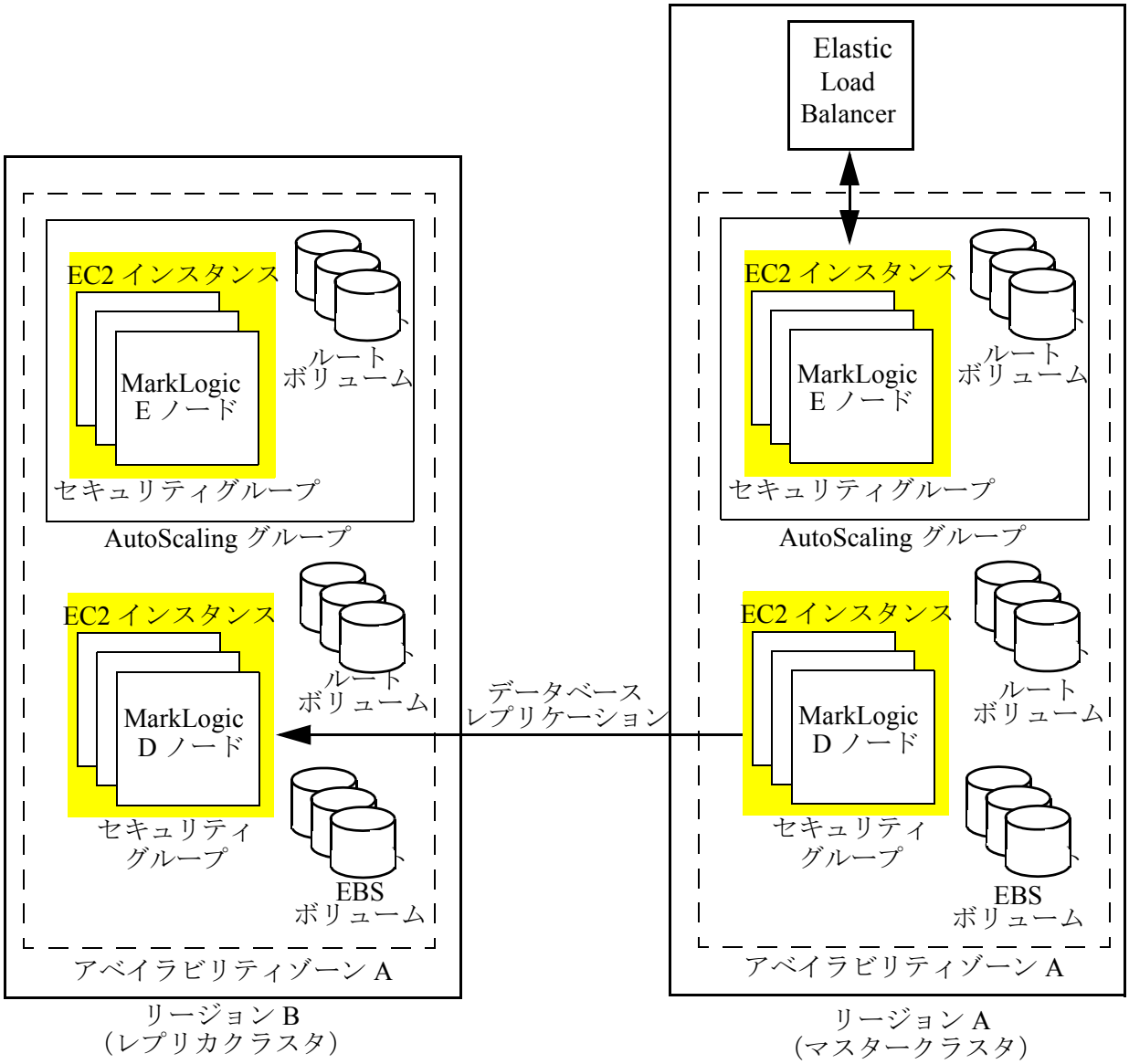
hi1.4xlarge インスタンスは、最高のパフォーマンスを提供します。MarkLogic は、SSD ボリュームを利用してすべての操作を高速化できます。ただし、SSD ボリュームは寿命が短いため、データベースのレプリケーションが必要です。また、m2.4xlarge は大量の RAM を提供し、1GB/ 秒の IO が保証された比較的高速な IO を実現します。m1.xlarge も同程度の IO を実現しますが、15GB/RAM 程度であり、キャッシュも少なくなります。

以下の図は、同じリージョン内の異なるゾーンに 2 つのクラスタが存在するアーキテクチャを示したものです。各クラスタ内の D ノードは、ローカルディスクフェイルオーバー用に設定されています。フォレストのデータは IOPS EBS ボリュームに格納され、バックアップと EBS ボリュームのスナップショットは S3 に格納されます。



以下の図は、データベースのレプリケーションを使用して 2 つのクラスタを設定し、異なるリージョンに導入するアーキテクチャを示したものです。このタイプのアーキテクチャは、Amazon Route 53 で機能します。これは、ユーザートラフィックを安全かつ信頼性の高い方法で MarkLogic クラスタにルーティングできるスケーラブルなドメインネームシステム（DNS）web サービスです。Route 53 では、マスタークラスタに障害が発生した場合は、数秒以内にトラフィックをレプリカクラスタにリダイレクトできます。レプリカの状態は、マスターよりもある程度（一般的に 5 ～ 20 秒）遅延しています。

Amazon Route 53 の詳細については、
http://docs.aws.amazon.com/Route53/latest/DeveloperGuide/Welcome.html を参照してください。



2.0 EC2 上の MarkLogic サーバーの基本的な使用方法

この章では、MarkLogic サーバー AMI を起動し、MarkLogic サーバー Admin 画面にアクセスする方法について説明します。この章は、次のセクションで構成されています。

- セキュリティ
- 導入手順のまとめ
- Amazon EC2 アカウントの作成
- EC2 AMI 用の MarkLogic サーバーの有効化
- 初期セットアップ手順
- AWS 設定変数

2.1 セキュリティ

MarkLogic サーバーへのアクセスは、『*Understanding and Using Security Guide*』で説明するメカニズムによって制御されます。EC2 環境内では、EC2 インスタンスへのアクセスは次の 3 つのメカニズムによって制御されます。

- 「キーペアの作成」（23 ページ）で説明されているキーペア
- 「IAM ロールの作成」（18 ページ）で説明されている AWS アイデンティティおよびアクセス管理（IAM）
- 「CloudFormation を使用した EC2 での MarkLogic の導入」（30 ページ）で説明されている CloudFormation テンプレートによって作成されるセキュリティグループ

  注： Amazon はセキュリティリソースを定期的に更新しています。MarkLogic サーバーの新しいインスタンスを作成するたびに、そのインスタンスには最新のセキュリティ更新が適用されます。古いインスタンスは自動的に更新されないので、クラスタ全体で統一された最新のセキュリティを維持するためには手動で更新する必要があります。またオプションで、新しいインスタンスの自動セキュリティ更新を無効にすることもできます。セキュリティ更新の詳細については、http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AmazonLinuxAMIBasics.html#security-updates を参照してください。

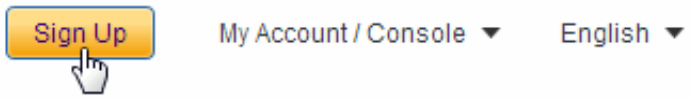
2.2 導入手順のまとめ

次の表は、EC2 上に MarkLogic サーバーを導入する手順をまとめたものです。

手順	詳細の参照先
Amazon EC2 アカウントを作成する（まだ所有していない場合）。	「Amazon EC2 アカウントの作成」（15 ページ）
MarkLogic サーバー AMI を有効にする。	「EC2 AMI 用の MarkLogic サーバーの有効化」（16 ページ）
Amazon AWS Management Console を開く。	「AWS Management Console へのアクセス」（17 ページ）
IAM ロールを作成する。	「IAM ロールの作成」（18 ページ）
キーペアを作成する（まだ作成していない場合）。	「キーペアの作成」（23 ページ）
SNS（Simple Notification Service）トピックを作成する。	「SNS（Simple Notification Service）トピックの作成」（23 ページ）
CloudFormation テンプレートから CloudFormation スタックを作成する。	「CloudFormation を使用した EC2 での MarkLogic の導入」（30 ページ）
MarkLogic サーバー Admin 画面を開く。	「MarkLogic サーバーインスタンスへのアクセス」（65 ページ）

2.3 Amazon EC2 アカウントの作成

EC2 AMI 用の MarkLogic サーバーは、Amazon EC2 アカウントをセットアップしていないと注文できません。Amazon EC2 アカウントをセットアップするには、http://aws.amazon.com にアクセスし、［サインアップ］をクリックします。



クリックしたら、新しいアカウントを作成する手順に従います。アカウントの作成では、メールアドレスを入力し、パスワードを作成して、クレジットカード情報を入力する必要があります。

2.4 EC2 AMI 用の MarkLogic サーバーの有効化

AMI は、MarkLogic が提供するものを使用できます。また、標準の Amazon ツールを使用して独自のカスタム AMI を作成することもできます。このガイドでは、AWS MarketPlace で入手可能な MarkLogic が提供する AMI を中心に説明します。

MarkLogic AMI を有効にするには、次の手順を実行します。

- https://aws.amazon.com/marketplace にアクセスします。
- MarkLogic を検索します。
- MarkLogic 製品ページで、［Accept Terms］ボタンをクリックします。

警告 推奨される CloudFormation 手順を使用せずに MarkLogic クラスタを手動で導入する計画がない限り、どの［Launch EC2 Instance］ボタンもクリックしないでください。

2.5 初期セットアップ手順

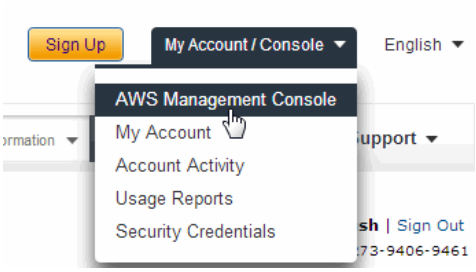
このセクションでは、AWS Management Console にアクセスして、セキュリティグループとキーペアを作成する方法について説明します。一般的に、セキュリティグループとキーペアは一度だけ作成し、その後に作成する各インスタンスではそれらを再利用することになります。このセクションで取り上げるトピックは次のとおりです。

- AWS Management Console へのアクセス
- IAM ロールの作成
- キーペアの作成
- SNS（Simple Notification Service）トピックの作成

2.5.1 AWS Management Console へのアクセス

このセクションでは、Amazon AWS Management Console にアクセスする方法について説明します。

1. Amazon EC2 アカウントにログインし、［My Account/Console］プルダウンメニューから［AWS Management Console］を選択します。



2. ［Sign into the AWS Management Console］をクリックし、EC2 アカウントのメールアドレスとパスワードを入力します。

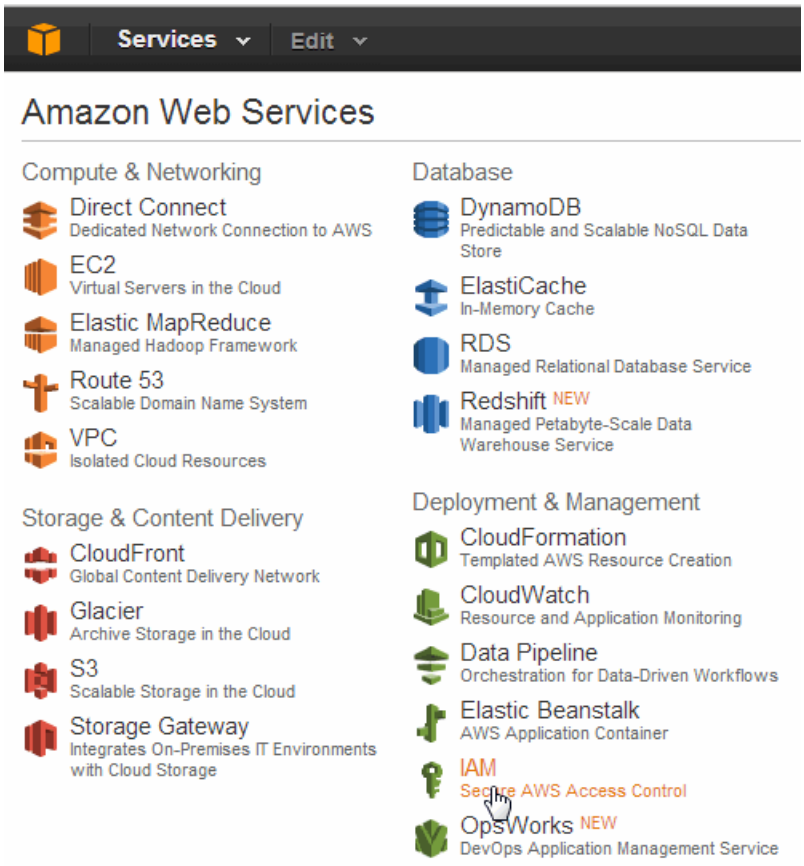


2.5.2 IAM ロールの作成

AWS Identity and Access Management（IAM）は、AWS でユーザーとユーザーパーミッションを管理できるようにする web サービスです。このサービスは、Amazon EC2、Amazon SimpleDB、および AWS Management Console を使用する複数のユーザーまたはシステムを抱える組織を対象にしています。IAM を使用すると、ユーザー、アクセスキーなどのセキュリティ資格情報、および AWS リソースへのユーザーアクセスを制御するパーミッションを集中管理できます。

このセクションでは、IAM ロールを作成する方法について説明します。また、各手順について説明しますが、各手順のすべてのオプションを説明するわけではありません。詳細については、http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-roles-for-amazon-ec2.html を参照してください。

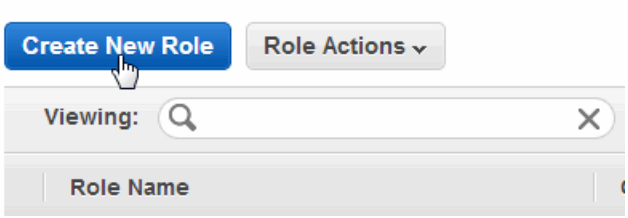
1. Amazon Web Services ページで、［IAM］をクリックします。



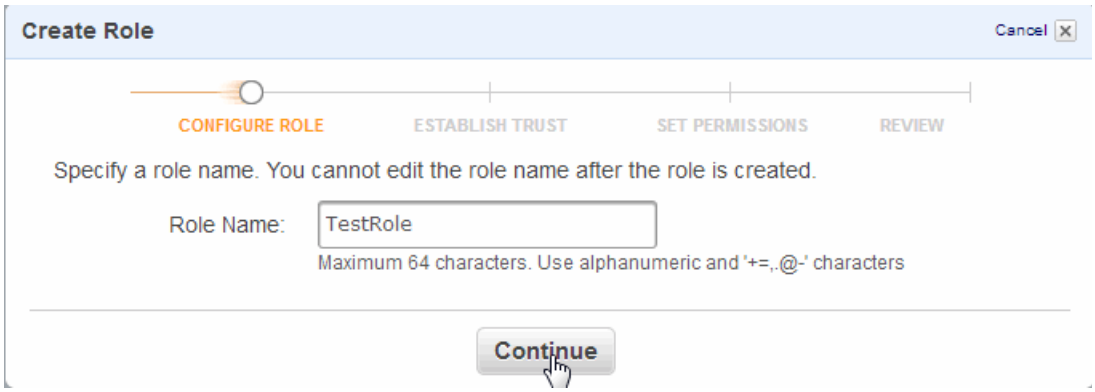
2. ［Getting Started］ページの［IAM Resources］セクションで［Roles］をクリックします。



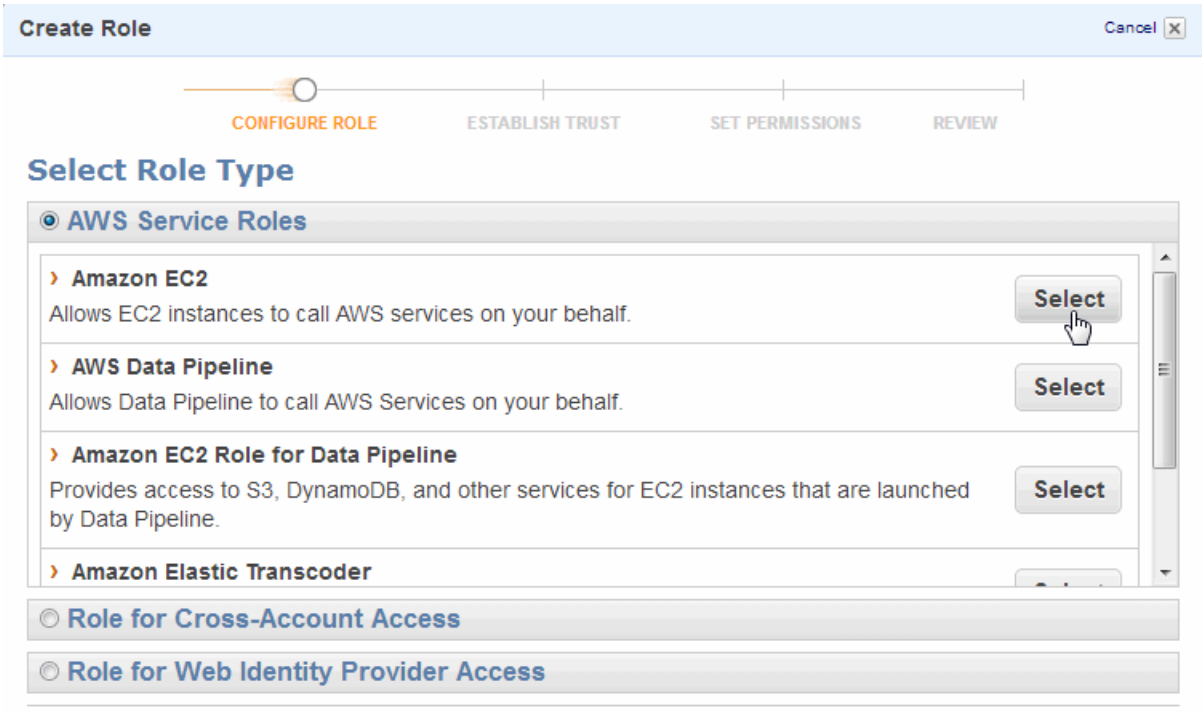
3. ［Roles］ページで［Create New Role］をクリックします。



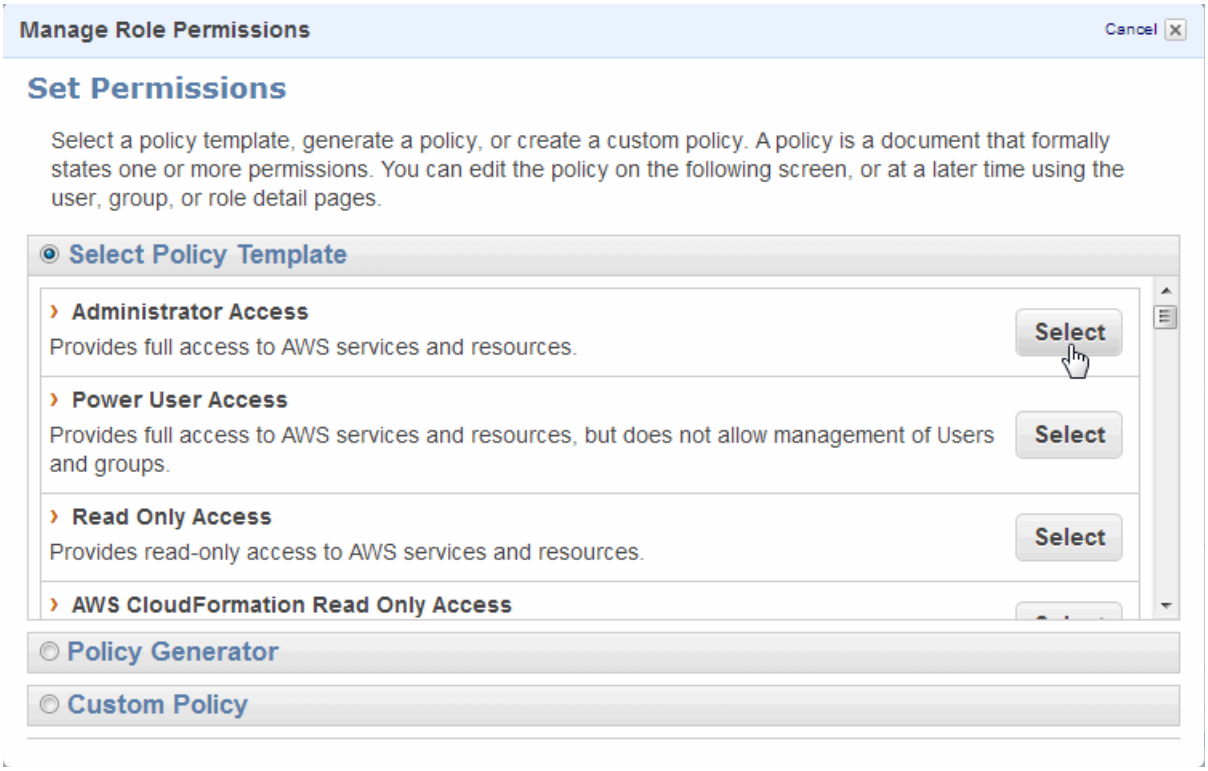
4. ［Create Role］ウィンドウで、新しいロールの名前を入力します。



5. ［Configure Role］ウィンドウで、［Amazon EC2］を選択します。



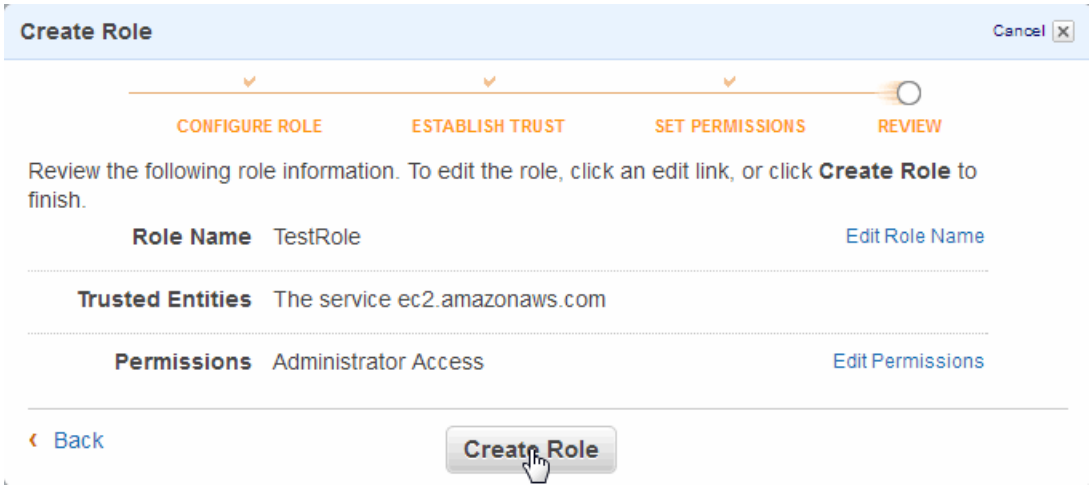
6. ［Set Permissions］ウィンドウで、ロールのアクセスポリシーを選択します。IAM ポリシーの詳細については、http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-policies-for-amazon-ec2.html を参照してください。



7. http://docs.aws.amazon.com/IAM/latest/UserGuide/PoliciesOverview.html の説明に従って、選択したポリシーのパーミッションを編集します。操作が終了したら、［Apply Policy］をクリックします。



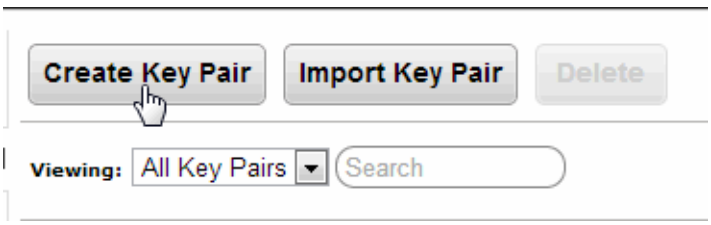
8. ［Review］ウィンドウで設定を確認し、変更する必要がある場合は編集します。操作が終了したら、［Create Role］をクリックします。



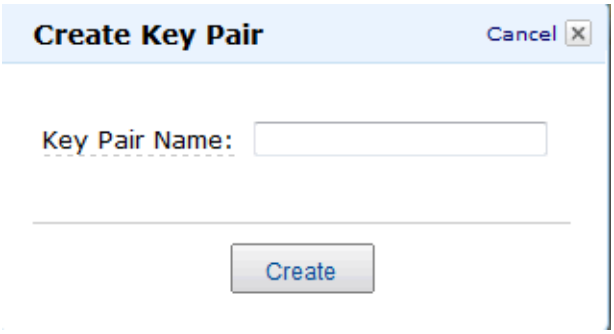
2.5.3 キーペアの作成

キーペアを使用すると、確実に自分のインスタンスだけにアクセスできます。Amazon EC2 キーペアは、1 つまたは複数作成できます。キーペアを使用すると、自分のインスタンスに SSH でアクセスできます。

1. AWS Management Console で、左側のナビゲーションセクションから［Key Pairs］を選択し、［Key Pairs］ページで［Create Key Pair］をクリックします。



2. キーペアの名前を入力し、［Create］をクリックします。



3. キーペアがローカルシステムにダウンロードされます。キーペアのダウンロードが完了したら、［Save File］をクリックします。

   注： キーペアをダウンロードしたローカルシステム上の場所は、覚えておく必要があります。このキーペアは、「EC2 インスタンスへのアクセス」（68 ページ）の説明に従って MarkLogic サーバーインスタンスへの SSH 接続を作成する場合に必要になります。

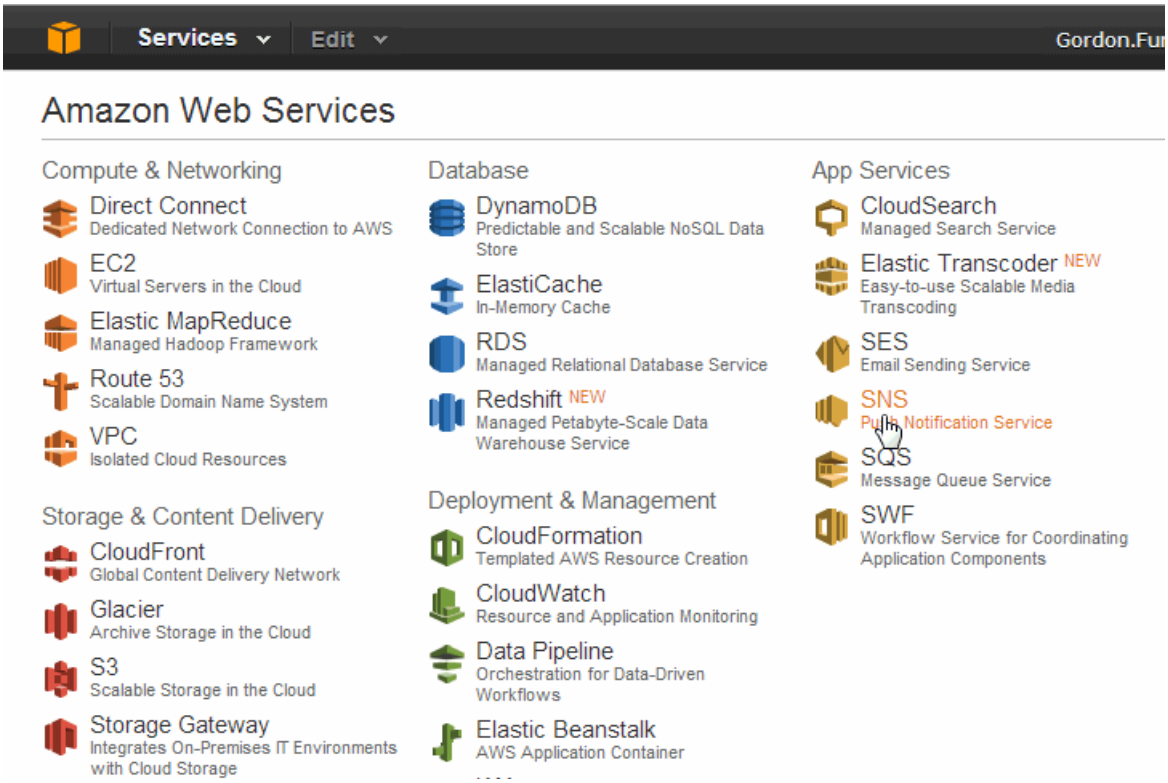
2.5.4 SNS（Simple Notification Service）トピックの作成

Amazon SQS（Simple Queue Service）は、EC2 インスタンスによって生成されたメッセージをキューに格納できるようにするキューシステムです。インスタンスからのメッセージを取得するには、SNS（Simple Notification Service）トピックを作成し、CloudFormation テンプレートでユーザーデータの一部としてそのトピックを指定する必要があります。

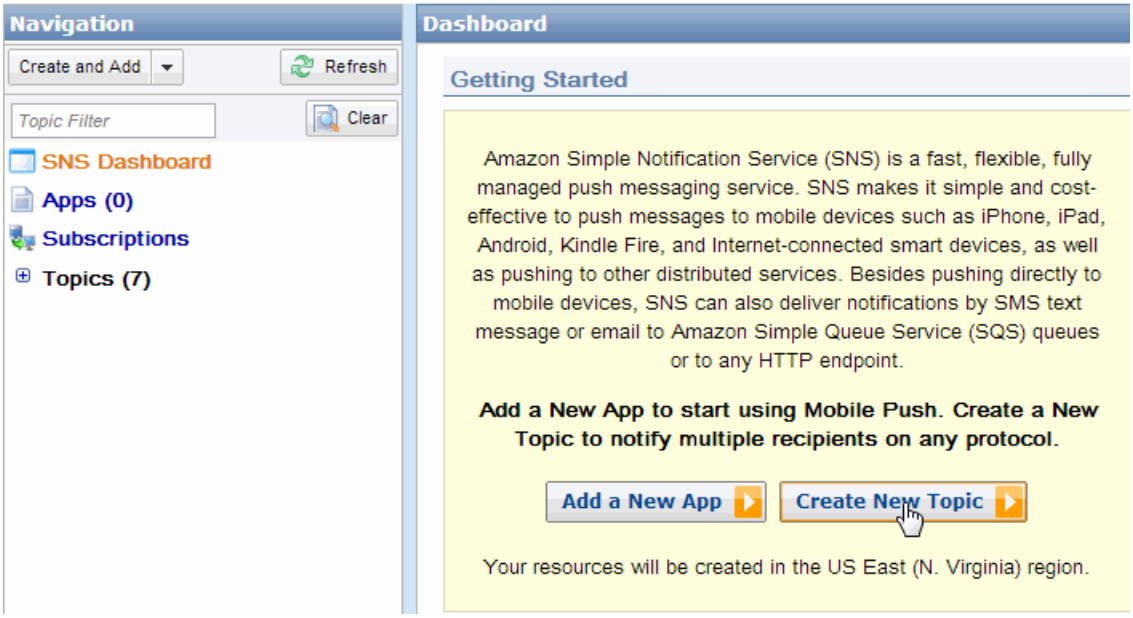
SQS キューシステムおよび SNS トピック作成の詳細については、http://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqssubscribe.html を参照してください。

SNS トピックを作成する方法はいくつかあります。次にその中の 1 つについて説明します。

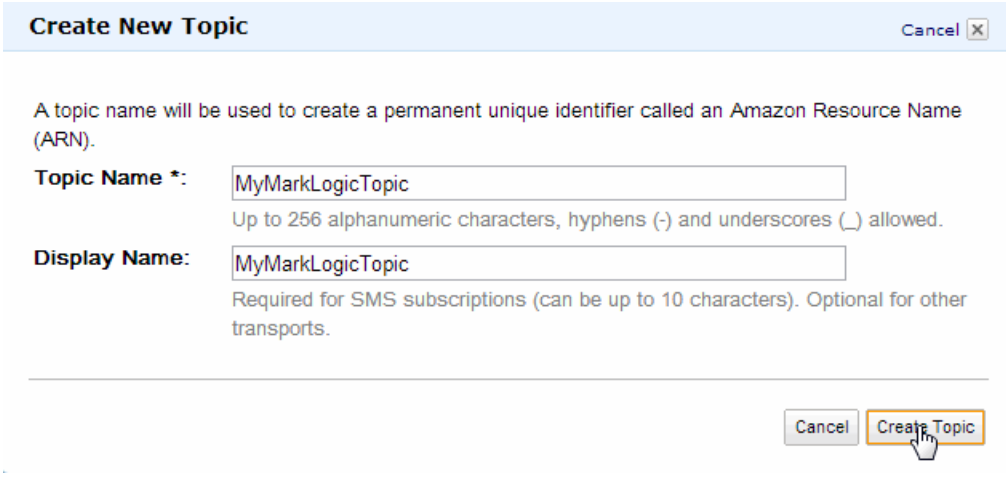
1. ［Amazon Web Services］メニューから［SNS Dashboard］を開きます。



2. ［Create New Topic］をクリックします。

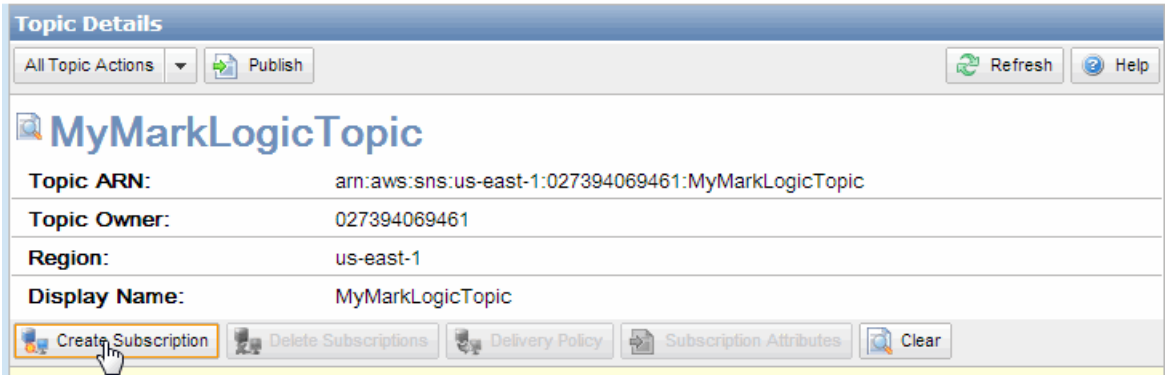


3. トピック名を入力し、オプションで表示名を入力します。［Create Topic］をクリックします。



4. ［Topic Details］ウィンドウ内の、トピックの ARN（Topic ARN）をメモしておきます。この情報は、「AWS Console を使用した CloudFormation スタックの作成」（35 ページ）の説明に従ってスタックを作成するときに、［LogSNS］フィールドに入力します。

5. メッセージを表示するには、SNS トピックをサブスクライブする必要があります。トピックをサブスクライブするには、［Topic Details］ページで［Create Subscription］をクリックします。http://docs.aws.amazon.com/sns/latest/dg/welcome.html で説明されているように、SNS トピックをサブスクライブする方法はいくつかあります。



2.6 AWS 設定変数

MarkLogic は、起動時に環境変数のセットを使用してカスタマイズできます。これは、外部で管理される単一ノードから完全なクラスタ管理機能を使用する大規模な分散クラスタに至るまで、すべての設定に適用されます。

これらの変数は、サーバーの起動方法や起動時期に関係なく、環境内に一貫性がある値が存在することを保証する任意の方法で指定できます。また、管理対象クラスタのサポートに関連する変数も、インスタンスごとに適切に設定する必要があります。EC2 ユーザーデータとして値を渡す手法は、単純で信頼性の高い手法であり、同じ AMI をすべてのインスタンスで再利用できるようになり、AMI 自体をカスタマイズする必要がなくなります。その他の手法としては、変数の割り当てを、初期起動中に `/etc/marklogic.conf` に配置する方法や、クラスタ内の各同等ノード専用のカスタム AMI に組み込む方法が挙げられます。

別の手法を使用して環境を設定することもできますが、起動、ランレベルの変更、および手動サービス操作（起動 / 停止 / 再起動）中にシステムがグローバルルート環境を設定する方法に違いがあれば、慎重に検討およびテストする必要があります。

システムを初期化するために使用する導入ツール、RPM インストールのプロセスと順序、システムの設定と起動によっては、インスタンス上での MarkLogic の最初の起動の前に、システムが正しく設定され、グループ内のすべてのインスタンスが一貫性のある設定になっていることを確認するために、別の設定手法が必要になることがあります。

サンプルの CloudFormation テンプレートが実装するアーキテクチャと戦略は、慎重に定義およびテストされています。これらは、実装に使用するツールに関係なく、デザインパターンとして模倣できる有効なモデルです。

次の環境変数は、MarkLogic の起動時に認識されるか、複数の設定場所から自動的に設定されます。

- MARKLOGIC_EC2_HOST — 0 に設定されている場合、すべての EC2 固有の設定、起動、管理の機能が無効になります（以降の残りの変数は使用されません）。

  これは、MarkLogic を外部から管理する場合に役に立ちます。

- MARKLOGIC_LICENSE_KEY — この MarkLogic インスタンスで使用するライセンスキー。
- MARKLOGIC_LICENSEE — MARKLOGIC_LICENSE_KEY に対応するライセンシー。
- MARKLOGIC_AWS_ACCESS_KEY — このインスタンス上のすべての AWS サービスで使用する AWS アクセスキー。

- MARKLOGIC_AWS_SECRET_KEY — このインスタンス上のすべての AWS サービスで使用する AWS 秘密キー。
- MARKLOGIC_CLUSTER_NAME — インスタンスとクラスタを自動的に設定するために使用する MarkLogic クラスタ名。

  SimpleDB の場合は、SimpleDB で使用される「ドメイン」に対応します。

- MARKLOGIC_CLUSTER_MASTER — 設定する必要があり、クラスタ内の正確に 1 つのノードに対して「1」と等しくなります。 マスターノードは初期データベースを作成し、クラスタのブートストラップホストになります。

  末尾に「#」が付いている名前の（MARKLOGIC_NODE_NAME を参照）複数のノードに対して「1」を設定できます。この場合、解決された名前が「1」で終わるノードのみが、クラスタブートストラップホストのロールとなります。

- MARKLOGIC_NODE_NAME — クラスタ内のノードの個別の名前。MARKLOGIC_CLUSTER_NAME が指定されている場合は必須です。末尾には「#」を付けることができます。「MyNode-#」のようにノード名の末尾に「#」がついている場合は、可変のノード名として識別されます。
- MARKLOGIC_ADMIN_USERNAME — MarkLogic 管理者のユーザー名。
- MARKLOGIC_ADMIN_PASSWORD — MarkLogic 管理者のパスワード。
- MARKLOGIC_EBS_VOLUME — プライマリ EBS ボリュームのボリューム仕様。このボリュームは論理デバイス `/dev/sdf` にアタッチされます。また、必要に応じてファイルシステムが作成され、`/var/opt/MarkLogic` にマウントされます。この値は volspec[,volspec ...] という形式になり、volspec は次のいずれかになります。
  - vol-xxxx：既存の EBS ボリュームにアタッチします
  - snap-xxxx：ボリュームを作成するために使用される AWS スナップショット
  - < 数字 > : ボリュームのサイズを示す 1 ～ 1024 までの整数（GB 単位）。新規のボリュームが作成されます。
  - < 仕様文字列 > : V1 EC2 CLI ツールと互換性がある形式のボリューム仕様文字列。この形式は現在、EC2 ユーザーデータまたは `/etc/marklogic.conf` を使用する場合のみサポートされます。
  - [snapshot-id]:[volume-size]:[delete-on-termination]:[volume-type[:iops]]:[encrypted]

各値の意味は次のとおりです。

パラメータ	説明
snapshot-id	ボリュームのソースとして使用する既存のスナップショット
volume-size	ボリュームのサイズ（GB 単位）
delete-on-termination	< 無視されます >
volume-type	EBS のボリュームタイプ。「standard」、「gp2」、「io1」のいずれかになります。
iops	プロビジョニングされた IOP（PIOP）- ボリュームタイプが「iops」の場合のみ許可されます。
encrypted	REST で EBS 暗号化を使用します。

例：

:20::gp2:true - 暗号化された D ストレージタイプの 20GB のボリューム

snap-abcde:200:::- スナップショット「snap-abcde」からボリュームを作成し、サイズを 200GB に変更します。デフォルトは gp2 ボリュームタイプです。

:1000::io1:2000:- 2000 PIOP の 1000GB PIOP ボリューム

注：

- 組み合わせた場合、一部の値のみが有効です。詳細については、EC2 EBS のドキュメントを参照してください。
- snapshot-id または volume-size のいずれかは必須です。
- 暗号化は snapshot-id と組み合わせて使用可能であり、対応するスナップショットが暗号化されている必要があります。
- iops は、ボリュームタイプ「io1」の場合のみ許可されます。
- 指定されていない場合のデフォルトのボリュームタイプは「gp2」です。
- 2 番目以降の指定は、前の volspec の繰り返しを示します。 例えば、「10,20,*」は、最初のノードでは 10GB のボリュームを作成し、同じ名前を持つ 2 番目以降のノードでは 20GB のボリュームを作成することを示します。

- MARKLOGIC_EBS_VOLUME1 ...MARKLOGIC_EBS_VOLUME9 — MARKLOGIC_EBS_VOLUME と同じ形式の最大 9 個の追加の EBS ボリューム。これらのノードについて、初期化、アタッチ、ファイルシステムの作成、およびマウントが行われます。
- MARKLOGIC_LOG_SNS — SQS（Simple Queue Service）からのメッセージを取得するために使用される SNS（Simple Notification Service）トピック。
- MARKLOGIC_BOOT_WAIT — （デフォルト = 30）。設定されている場合、この値は、初期データボリューム（$MARKLOGIC_EBS、デフォルトは /dev/sdf）がオンラインになるまでに待機する最大時間（秒数）です。これは、MARKLOGIC_EBS_VOLUME が指定されていない場合にのみ使用され、MarkLogic はボリュームが手動または外部プロセスによってアタッチされるのを待機します。

  ボリュームがアタッチされずにタイムアウトになった場合、起動がアボートされます。

3.0 CloudFormation を使用した EC2 での MarkLogic の導入

この章では、CloudFormation テンプレートを使用して MarkLogic サーバーを導入する方法について説明します。

- 概要
- 導入と起動
- AWS Console を使用した CloudFormation スタックの作成
- AWS コマンドラインインターフェイスを使用した CloudFormation スタックの作成
- CloudFormation テンプレートのサンプル
- CloudFormation スタックの削除

3.1 概要

管理対象クラスタは、自動的に初期化され、推奨されるトポロジを使用して事前設定されます。

CloudFormation は、*スタック*を作成するために必要なコンポーネントのセットを指定できるようにする AWS の技術です。管理対象クラスタは、提供されているいずれかの Amazon CloudFormation テンプレートを使用して作成できます。また、管理対象クラスタテンプレートでは次の項目が作成されます。

- AWS サービスを実行するために必要な IAM ロール（セキュリティ資格情報で渡す必要はありません）
- インスタンスに配信される受信ネットワークトラフィックを制御するセキュリティグループ
- ノードあたり 1 つずつの AutoScaling グループ
- AutoScaling グループの起動設定
- すべてのノードに対応するロードバランサー
- 各ノード用の EBS ボリューム

CloudFormation テンプレートを使用する場合は、複数のパラメータを入力する必要があります（AWS Console または CloudFormation スタックを起動できる任意のサードパーティ製コマンドラインツールを使用）。各パラメータでは、次の情報を指定します。

- 各ノードの実行場所となるゾーン
- セキュリティデータベースを最初に作成するための管理ユーザーとパスワード
- SSL キー名（インスタンス起動後にインスタンスにログインするために使用）

- 初期データボリューム /var/opt/MarkLogic 用に作成するボリュームのサイズ（GB 単位）
- AutoScaling グループおよび管理対象クラスタサポートの起動手順からメッセージを取得するために使用する SNS（Simple Notification Service）トピック
- 作成されるインスタンスの EC2 インスタンスタイプ

起動すると、CloudFormation によってすべての必要なリソースが作成されます。そのようなリソースは、起動時に Amazon EC2 ノードによって管理対象クラスタの一部であることが認識され、次のアクションが実行されます。ユーザーが介入する必要はありません。

- そのノードに関連付けられているすべてのボリュームをアタッチする
- 必要に応じてファイルシステムを作成する
- ファイルシステムをマウントする
- MarkLogic を起動する
- EC2 ライセンスを適用して受け入れる
- 初期ノード（マスター）を作成し、管理ユーザー名とパスワードを設定するか、クラスタにアタッチする
- ノードをロードバランサーに関連付ける

ロードバランサーは、ポート 7997 上で HealthCheck アプリケーションサーバーを使用して MarkLogic が適切に実行されていることを検出します。また、MarkLogic インスタンスが稼動していることが確認された場合のみ、そのノードにトラフィックを転送します。

各 AutoScaling グループ（ASG）はシステムの安定性を検出し、オペレーティングシステムに問題がある場合はそのノードを終了して再起動します。すべてのノードについて ASG `NodesPerZone` 値を 0 に設定することで、いつでもクラスタを一時停止できます。その後、各 ASG の `NodesPerZone` を値 1 ～ 20 に再設定することでノードを再起動できます。再起動時は、一時停止状態から再開するか、障害を検出した ASG を再起動してサーバーを再起動することで、システムが次の処理を自動的に実行します。

- 以前にアタッチされていたすべてのボリュームを検出し、それらを再アタッチする
- 前回の起動後にホスト名が変更されたかどうかを検出し、変更されている場合は、MarkLogic クラスタ内で新しいホスト名に変更する
- クラスタに再アタッチする

3.2 導入と起動

MarkLogic は、システムサービスとして `/etc/init.d` から起動するか、「service MarkLogic start」などによって手動で起動します。 標準インストールの場合、インストール後の次の再起動時に MarkLogic が起動しますが、スクリプトまたはシステム設定を使用して任意の時点で起動することもできます。

起動環境のカスタマイズは、インストール後の最初の MarkLogic 起動の前に完全に設定しておき、ロール（単一、クラスタマスター、クラスタ結合子）を正しく設定し、正しいデータボリューム、Java JVM、パス、および他の設定可能な情報を検出できるようにしておく必要があります。このセクションでは、AWS 固有の設定変数について説明します。

MarkLogic は、一般的にシステムの起動時に起動するように設定されますが、手動で起動することもできます。一貫性のある動作を確保するため、同じ環境を継承するようにすべての起動パスを設定する必要があります。 最も大きな違いは、MarkLogic が AMI にプレインストールされているかどうかによって生じます。

MarkLogic に影響する関連起動プロセスは、初期化プロセス中に実行されます。MarkLogic がプレインストールおよび設定されていない状況で AMI を使用する場合は、特に他のサービスとの連携および依存関係を考慮する必要があります。

次の表は、AWS Linux システムでの一般的なサービスの起動順序を示したものです。

順序	サービス
02	lvm2-monitor
08	ip6tables
08	iptables
10	network
11	auditd
12	rsyslog
13	irqbalance
15	mdmonitor
22	messagebus
25	blk-availability
25	netfs

順序	サービス
26	acpid
26	udev-post
50	cloud-init-local
51	cloud-init
52	cloud-config
55	sshd
57	ntpdate
58	ntpd
80	sendmail
85	MarkLogic（バージョン 7）
86	tomcat-jsvc
90	crond
95	atd
95	jexec
97	yum-updatesd
98[c]	cloud-final（すべてのユーザー定義の upstart および cloud-init スクリプト）
98[M]	MarkLogic（バージョン 8）
99	local（/etc/rc.local）

`cloud-init` には複数のコンポーネントがあります。cloud-config 状態（52）で生成されるファイルおよび設定データは非常に詳細に設定できますが、導入ツールは独自の目的でこれを使用する点に注意してください。 最も一般的なのは、「cloud-final」（98[c]）で実行される「ユーザースクリプト」です。

MarkLogic はバージョン 8 で LSB 初期設定形式に移行したため、`cloud-final` の後に実行されるという依存関係が追加されました。そのため、MarkLogic がプレインストールされているかどうかに関係なく、MarkLogic の前にユーザー設定を適用できます。

MarkLogic の起動時には次のプロセスが実行されます。

1. `/etc/init.d/MarkLogic` が呼び出されます。これは、初期化（/etc/rc5.d/S98MarkLogic など）または手動（service MarkLogic start など）によって実行されます。

2. `/etc/sysconfig/MarkLogic` が読み込まれます（以降の処理を実行）。

3. 主要な環境変数のデフォルト値が設定されます。

4. `/etc/marklogic.conf` が読み込まれます（存在する場合）。ここで変数を変更または追加できます。

5. MARKLOGIC_EC2_HOST!=1 の場合、EC2 固有の追加の処理は実行されません

6. 定義されていない場合は、次の順序で EC2 メタデータを使用して、MARKLOGIC_HOSTNAME が計算されます。

    - public-hostname
    - public-ipv4
    - local-hostname
    - local-ipv4
    - hostname

7. インスタンスに関連付けられている IAM ロールから MARKLOGIC_AWS_ROLE が取得されます。

8. まだ設定されていない場合、MARKLOGIC_EBS は `/dev/sdf` に設定されます。

9. MARKLOGIC_EC2_USERDATA != 0 の場合、EC2 ユーザーデータが読み取られて解析されます。既存の設定は、任意の名前 / 値のペアによって上書きされます。

10. 再起動時の値の再計算によるオーバーヘッドを回避するため、解決済みのすべての変数は `/usr/local/mlcmd.conf` に書き込むことでキャッシュされます。

11. MARKLOGIC_CLUSTER_NAME、MARKLOGIC_NODENAME、および MARKOGIC_CLUSTER_MASTER が定義されている場合、管理対象クラスタのロジックが実行されます。

    - クラスタの形成または参加
    - データボリュームの作成 / アタッチ
    - ホスト名の変更の解決
    - クラスタ設定の更新

    注： このプロセスは、起動およびサービス開始のたびに繰り返されます。

3.3 AWS Console を使用した CloudFormation スタックの作成

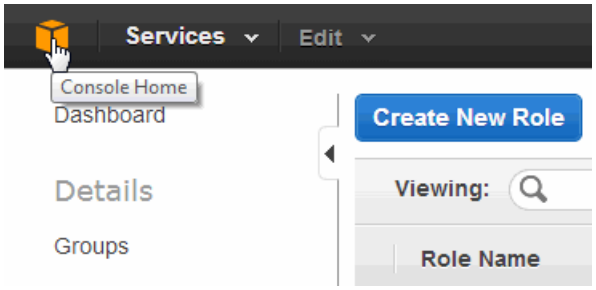
このセクションでは、AWS Console を使用して、テンプレートから CloudFormation スタックを作成する方法について説明します。また、各手順について説明しますが、各手順のすべてのオプションを説明するわけではありません。詳細については、http://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/cfn-console-create-stack.html を参照してください。

CloudFormation スタックを作成するには、次のものが必要です。

- Amazon MarketPlace から AMI を購入するか、MarkLogic から AMI を入手します。
- CloudFormation テンプレート。テンプレートは MarkLogic から入手できます。また、「CloudFormation テンプレートのサンプル」（46 ページ）の説明に従って独自のテンプレートを作成することもできます。MarkLogic CloudFormation テンプレートは、http://developer.marklogic.com/products/aws から入手できます。
- IAM ロール。「IAM ロールの作成」（18 ページ）で説明されています。
- キーペア。「キーペアの作成」（23 ページ）で説明されています。
- SNS トピック。「SNS（Simple Notification Service）トピックの作成」（23 ページ）で説明されています。

次の手順は、テンプレートから CloudFormation スタックを作成する方法について説明したものです。

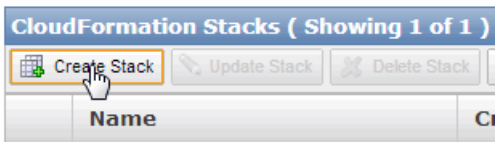
1. ［Services］ページの左上部分にあるボックスをクリックし、アマゾンウェブサービスのホームページにアクセスします。



2. アマゾンウェブサービスのホームページで、［CloudFormation］をクリックします。

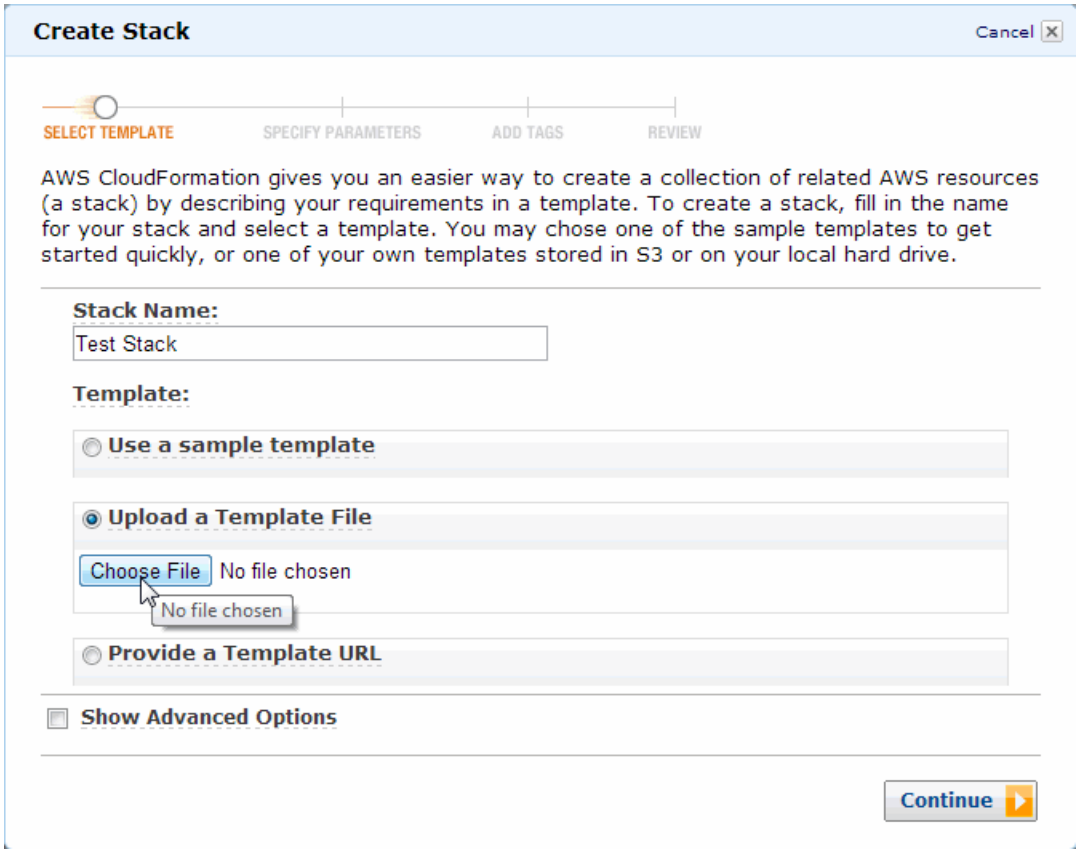


3. ［CloudFormation Stacks］ページで、［Create Stack］をクリックします。

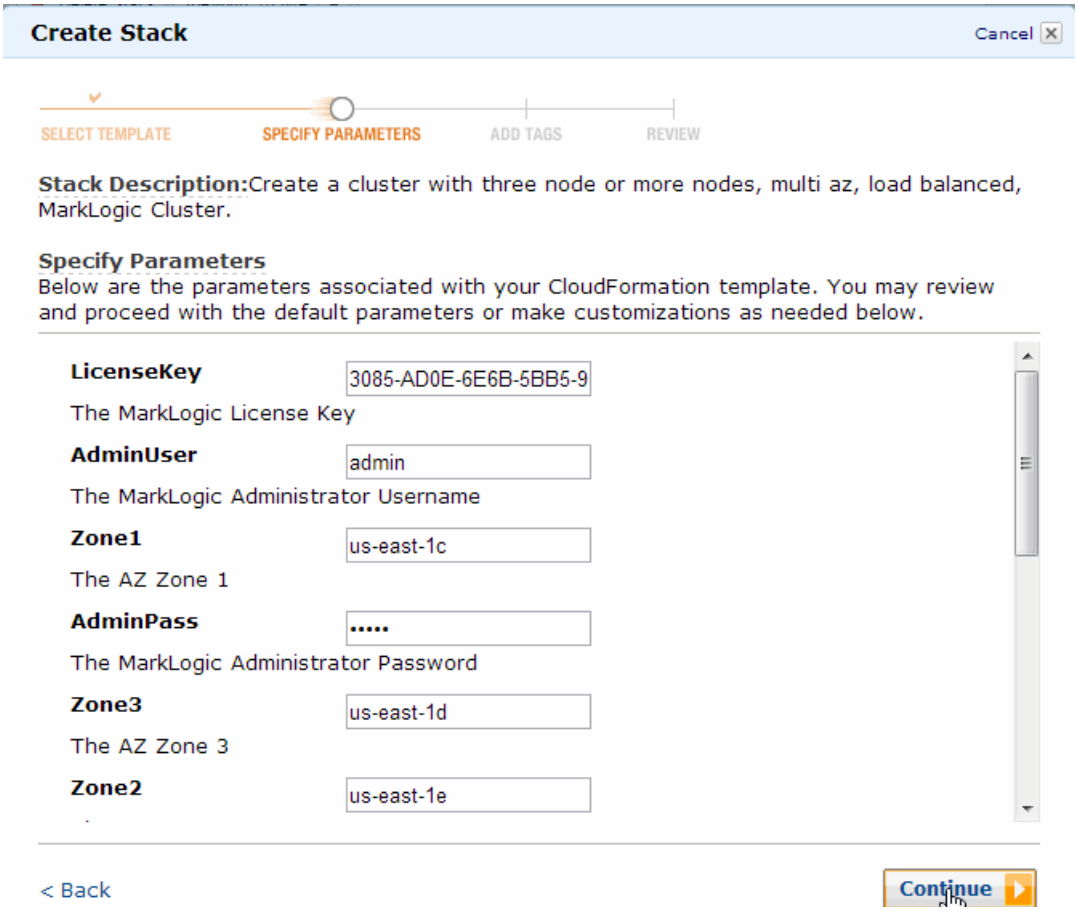


4. ［Select Template］ウィンドウで、スタックの名前を入力します。［Choose File］をクリックし、「CloudFormation テンプレートのサンプル」（46 ページ）で作成した CloudFormation スクリプトファイルを選択します。また、［Provide a Template URL］をクリックし、http://developer.marklogic.com/products/aws に記載されているいずれかの URL を入力することもできます。操作が終了したら、［Continue］をクリックします。

   注： スタック名は、EBS ボリュームの名前を含め、スタックのすべてのリソースを識別する目的で使用されます。スタック名には、自分のユーザー名など、簡単に識別できる名前を付けることをお勧めします。各ゾーンの最初のノードを除くすべてのノードの EBS ボリュームは、スタックを削除しても削除されません。そのため、スタックを削除した後でそれらのボリュームを削除する必要がある場合に備え、簡単に識別できるようにしておく必要があります。



5. ［Specify Parameters］ウィンドウで、次の表に示す情報を入力します。操作が終了したら、［Continue］をクリックします。

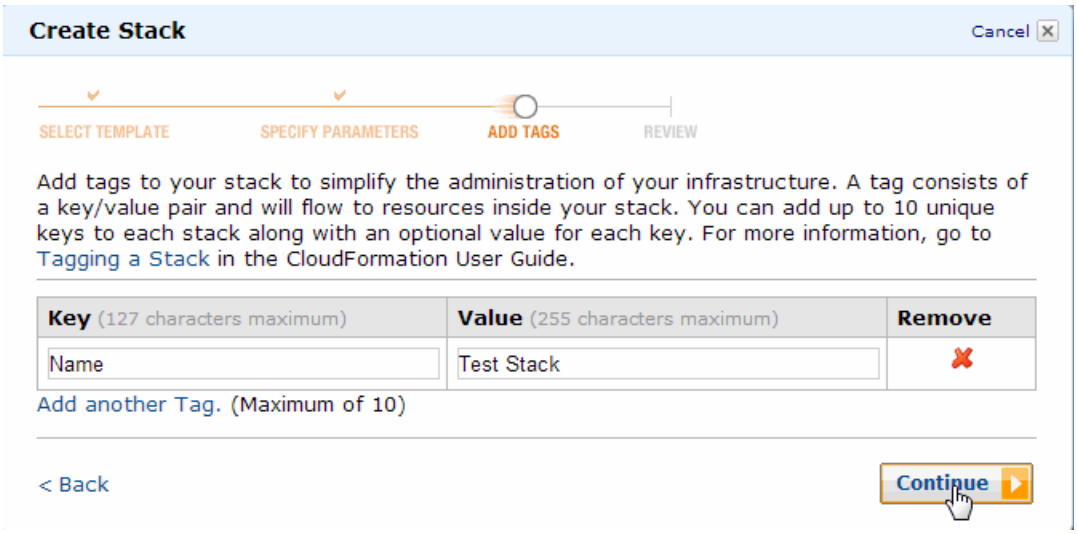


パラメータ名	説明
AdminUser	MarkLogic 管理者としてログインするために使用するユーザー名。
AdminPass	MarkLogic 管理者としてログインするために使用するパスワード。
IAMRole	「IAM ロールの作成」（18 ページ）で作成した IAM ロールの名前。

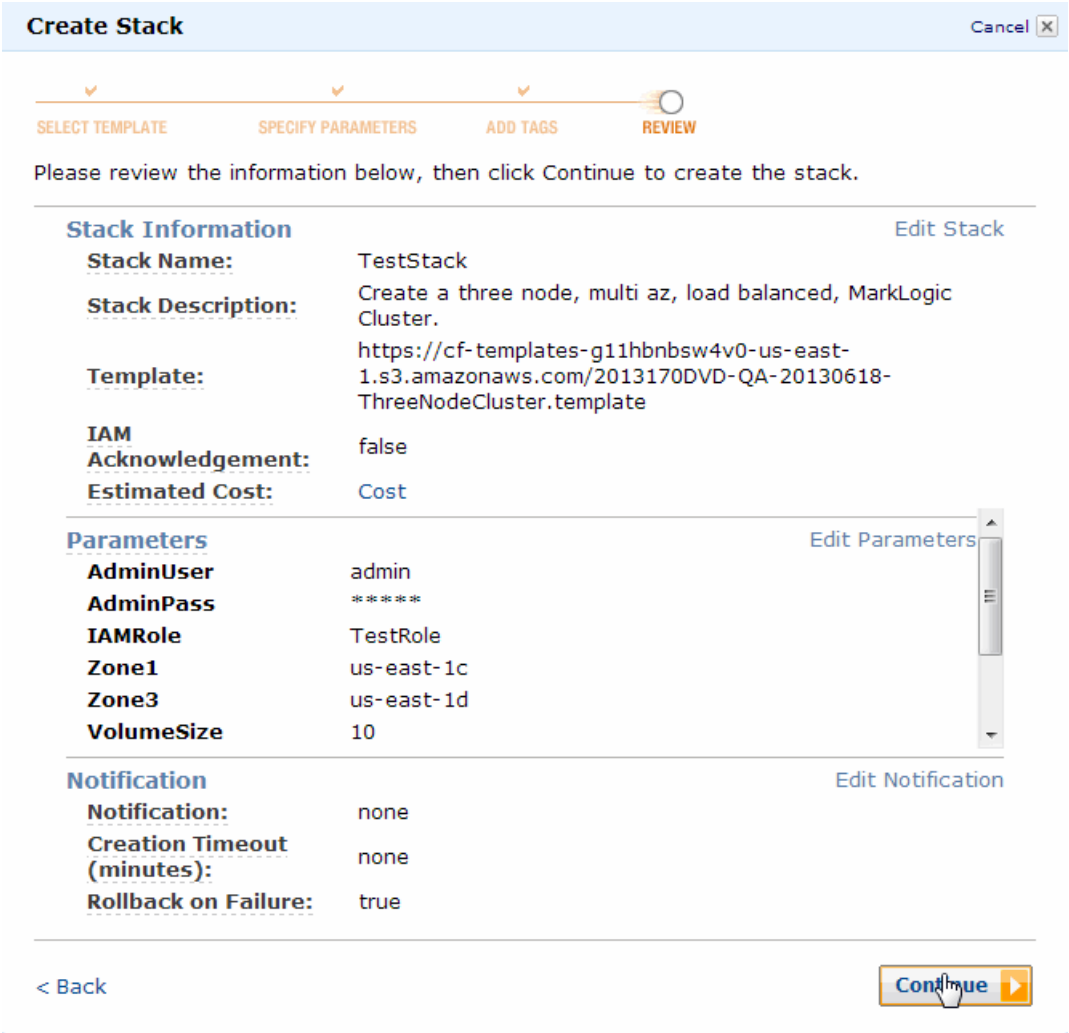
パラメータ名	説明
InstanceType	起動する EC2 インスタンスのタイプ。次のタイプを指定できます。 準仮想化（PVM）： • m1.small（デフォルト） • m1.medium • c1.medium • m1.large • m1.xlarge • m2.4xlarge ハードウェア仮想マシン（HVM）： • m3.xlarge • m3.2xlarge • cc1.4xlarge • cc2.8xlarge • cr1.8xlarge • cg1.4xlarge • hi1.4xlarge • hs1.8xlarge 各インスタンスタイプの詳細については、http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/instance-types.html を参照してください。
KeyName	「キーペアの作成」（23 ページ）で作成したキーペアの名前。
Licensee	MarkLogic の担当者から入手したライセンシーの名前。ライセンス情報を後で入力する場合は、「none」と入力します。
LicenseKey	MarkLogic の担当者から入手したライセンスキー。ライセンス情報を後で入力する場合は、「none」と入力します。
LogSNS	ログ記録に必要な SNS（Simple Notification Service）。SNS Dashboard に表示されるトピックの ARN 全体を入力します。SNS トピックの取得方法の詳細については、「SNS（Simple Notification Service）トピックの作成」（23 ページ）を参照してください。

パラメータ名	説明
NodesPerZone	各ゾーンで作成するノード（ホスト）の数。例えば値 `1` を入力すると、ゾーンごとに 1 つのノードが作成され、クラスタの合計は 3 ノードになります。値 `0` を入力すると、すべてのノードが一時停止されます。
SpotPrice	インスタンスのスポット料金（USD/ 時間）。デフォルト値は 0 です。0 以外の場合、指定した金額はオンデマンドではなくインスタンスのスポットリクエストとして使用されます。
VolumeSize	初期の EBS ボリュームサイズ（GB）。有効な値の範囲は 10 ～ 1000 です。デフォルトは 10 です。
Zone1 Zone2 Zone3	各ホストを作成するクラスタ内のゾーン。クラスタ内の各ゾーンは、`us-east` や `us-west` などの同じリージョンに含まれている必要があります。 `us-east` リージョンでは次のゾーンを使用できます。 • `us-east-1a` • `us-east-1b` • `us-east-1c` • `us-east-1d` • `us-east-1e` `us-west` リージョンでは次のゾーンを使用できます。 • `us-west-2a` • `us-west-2b` • `us-west-2c` `eu-west` リージョンでは次のゾーンを使用できます。 • `eu-west-1a` • `eu-west-1b` • `eu-west-1c`

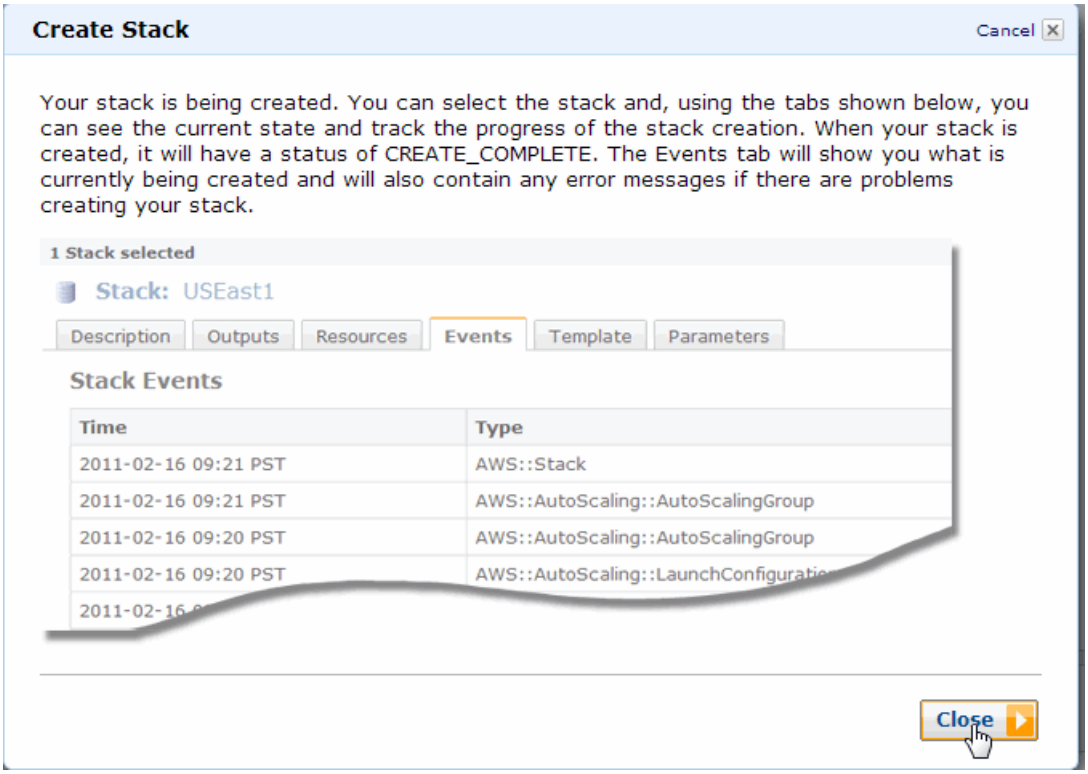
6. ［Add Tags］ウィンドウで、スタックのタグを追加します。入力するタグは、EC2 ダッシュボードで EC2 リソースを識別します。例えば、キーを `Name` として識別した場合、指定した値（例えば `Test Stack`）は、EC2 ダッシュボードの［Instance］リストの［Name］列に表示されます。タグの詳細については、http://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/cfn-console-add-tags.html を参照してください。操作が終了したら、［Continue］をクリックします。



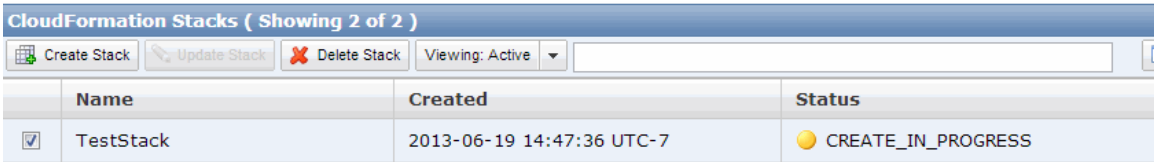
7. ［Review］ウィンドウで、設定を確認します。［Edit］をクリックして変更を加えます。操作が終了したら、［Continue］をクリックします。



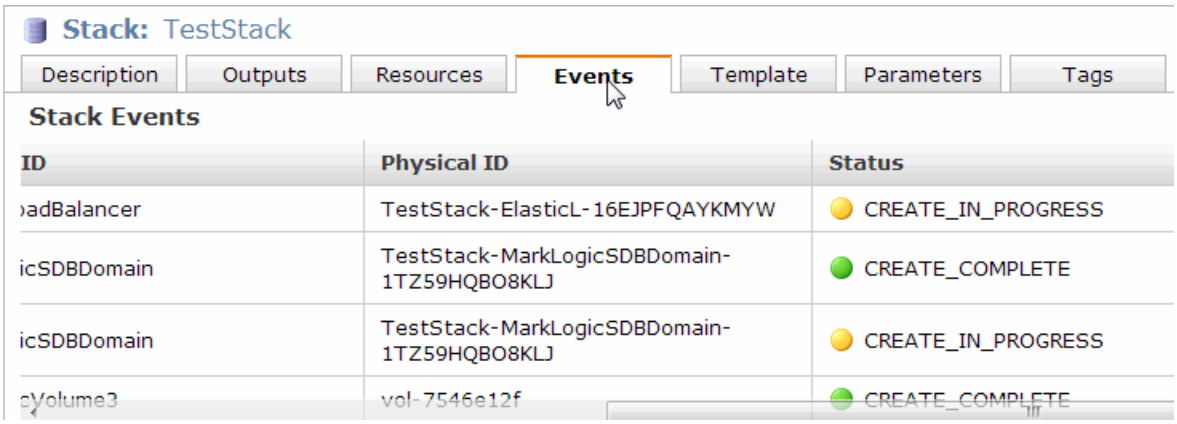
8. スタックが作成されていることを知らせるメッセージが表示されます。[Close] をクリックします。



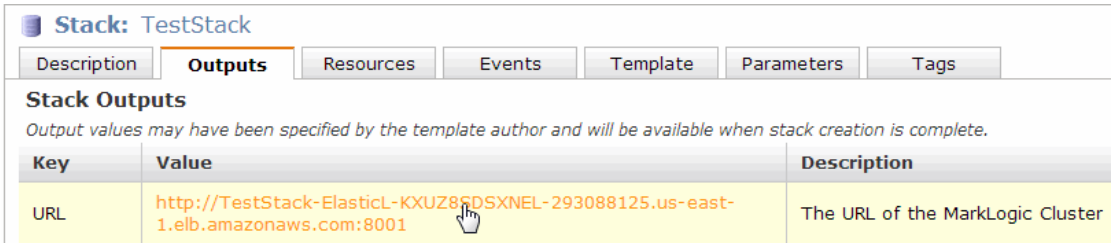
9. ページの上部にスタックの名前、作成日、およびステータスが表示されます。



10. AWS の処理速度およびスタック内で作成するリソースの数によっては数分かかることがあります。スタック作成の進行状況は、ページの下部にある［Events］タブを使用して表示できます。最新のステータスを確認するには、［Refresh］をクリックします。



11. 「CREATE_COMPLETE」というステータスは、AutoScaling グループが作成されたことを示します。EC2 インスタンスが起動するまで約 5 ～ 10 分待ってから［Outputs］タブに移動し、［Value］列でロードバランサーの URL をクリックします。これにより、使用可能なインスタンスに関する MarkLogic の Admin 画面が開きます。



注： ［Outputs］タブの URL が機能しない場合は、さらに 5 ～ 10 分待ってからもう一度試してください。

12. 手順 5 で指定した管理者のユーザー名とパスワードを使用してログインします。

注： すべてのホストが作成されてクラスタに参加するまで、Admin 画面では何も変更しないでください。スタックのステータスについて疑わしい点がある場合は、「SNS（Simple Notification Service）トピックの作成」（23 ページ）の説明に従って SNS トピックからログを確認します。

3.4 AWS コマンドラインインターフェイスを使用した CloudFormation スタックの作成

CloudFormation スタックは、AWS CloudFormation Console だけでなく、AWS CloudFormation コマンドラインインターフェイス（CLI）を使用して作成することもできます。AWS CloudFormation CLI の詳細については、http://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/cfn-using-cli.html を参照してください。

> **注：** AWS コマンドラインツールでは、CloudFormation パラメータ値でスペースを使用することはできません。スペースが含まれるパラメータ値を使用するとエラーになります。

CLI コマンドのリストについては、http://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/CFN_CMD.html を参照してください。

AWS CloudFormation CLI を使用してスタックを作成する方法をまとめると、次のようになります。

1. http://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/cfn-installing-cli.html の説明に従い、自社のシステム用に AWS CloudFormation CLI 環境をインストールして設定します。

2. 「AWS Console を使用した CloudFormation スタックの作成」（35 ページ）に示すようなパラメータを指定して cfn-create-stack 関数を呼び出します。

3. cfn-create-stack 関数は非同期で実行されるため、スタックが作成される前にスタックの ID が返されます。スタック ID を指定して cfn-describe-stack-events コマンドを使用すると、そのスタックのステータスを確認できます。

4. スタックが作成されたら、cfn-describe-stacks 関数を使用して、MarkLogic Admin 画面の URL を取得できます。

3.5 CloudFormation テンプレートのサンプル

CloudFormation テンプレートは、*スタック*と呼ばれる AWS リソースのコレクションを作成する目的で使用する JSON コードで構成されています。CloudFormation テンプレートの詳細については、http://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/template-guide.html を参照してください。このセクションでは、3 つ以上のノードを持つ MarkLogic クラスタで構成されたスタックを作成するために使用する CloudFormation テンプレートについて説明します。このテンプレートはそのまま使用することもできますが、変更を加えて、AWS で独自の MarkLogic クラスタを作成することもできます。

> **注：** このテンプレートを変更する前に、「AWS Console を使用した CloudFormation スタックの作成」（35 ページ）の説明に従って変更されていないテンプレートを実行し、クラウドスタックの作成手順を習得しておくことをお勧めします。

CloudFormation テンプレートの主なセクションは、次のとおりです。

- Parameters 宣言
- Mappings 宣言
- Resources 宣言
- Outputs 宣言

3.5.1 Parameters 宣言

テンプレートの `Parameters` ブロックでは、MarkLogic クラスタを構築するために必要なパラメータを定義しています。ここでは 3 つのゾーンが、クラスタ内でサーバーが作成されるホスト対象のゾーンを定義しています。http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-regions-availability-zones.html で説明されているように、すべてのゾーンが同じリージョンに置かれている必要があります。

各パラメータの説明については、「AWS Console を使用した CloudFormation スタックの作成」（35 ページ）の手順 5 の最後に示されている表を参照してください。

```
{
  "AWSTemplateFormatVersion":"2010-09-09",
  "Description":"Create a cluster with three node or more nodes,
multi az, load balanced, MarkLogic Cluster.
MarkLogic-7.0-20140529.x86_64.rpm",
  "Parameters":
   {
    "AdminUser":
     {
      "Description":"The MarkLogic Administrator Username",
      "Type":"String"
     },
```

```
    "AdminPass":
     {
      "Description":"The MarkLogic Administrator Password",
      "Type":"String",
      "NoEcho":"true"
     },
    "InstanceType":
     {
      "Description":"Type of EC2 instance to launch",
      "Type":"String",
      "Default":"m1.medium",
      "AllowedValues":
["m1.small","m1.medium","m1.large","m1.xlarge","m2.xlarge","m2.2xlarge
","m2.4xlarge","m3.medium","m3.large","m3.xlarge","m3.2xlarge",
"c1.medium","c1.xlarge","cc1.4xlarge","cc2.8xlarge","cr1.8xlarge",
"cg1.4xlarge","hi1.4xlarge","hs1.8xlarge","i2.xlarge","i2.2xlarge",
"i2.4xlarge","i2.8xlarge","r3.large","r3.xlarge","r3.2xlarge",
"r3.4xlarge","r3.8xlarge"]
     },
    "IAMRole":
     {
      "Description":"IAM Role",
      "Type":"String"
     },
    "Licensee":
     {
      "Description":"The MarkLogic Licensee or 'none'",
      "Type":"String",
      "Default":"none"
     },
    "LicenseKey":
     {
      "Description":"The MarkLogic License Key or 'none'",
      "Type":"String",
      "Default":"none"
     },
    "KeyName":
     {
      "Description":"Name of and existing EC2 KeyPair to enable
SSH access to the instance",
      "Type":"String"
     },
    "LogSNS":
     {
      "Description":"SNS Topic for logging - optional/advanced",
      "Type":"String",
      "Default":"none"
     },
```

```
    "NodesPerZone":
     {
      "Description":"Total number of nodes per Zone. (3 zones).
Set to 0 to shutdown/hibernate",
      "Type":"Number",
      "MinValue":"0",
      "MaxValue":"20",
      "Default":"1"
     },
    "SpotPrice":
     {
      "Description":"Spot price for instances in
USD/Hour - Optional/advanced",
      "Type":"Number",
      "MinValue":"0",
      "MaxValue":"2",
      "Default":"0"
     },
    "VolumeSize":
     {
      "Description":"The EBS Data volume size (GB) for all nodes",
      "Type":"Number",
      "MinValue":"10",
      "MaxValue":"1000",
      "Default":"10"
     },
    "Zone1":
     {
      "Description":"The AZ Zone 1 (e.g. us-west-2a)",
      "Type":"String",
      "AllowedValues":
       ["us-east-1a","us-east-1b","us-east-1c","us-east-1d",
"us-east-1e","us-west-2a","us-west-2b","us-west-2c","eu-west-1a",
"eu-west-1b","eu-west-1c"]
     },
    "Zone2":
     {
      "Description":"The AZ Zone 2",
      "Type":"String",
      "AllowedValues":
       ["us-east-1a","us-east-1b","us-east-1c","us-east-1d",
"us-east-1e","us-west-2a","us-west-2b","us-west-2c","eu-west-1a",
"eu-west-1b","eu-west-1c"]
     },
    "Zone3":
     {
      "Description":"The AZ Zone 3",
      "Type":"String",
      "AllowedValues":
       ["us-east-1a","us-east-1b","us-east-1c","us-east-1d",
"us-east-1e","us-west-2a","us-west-2b","us-west-2c","eu-west-1a",
"eu-west-1b","eu-west-1c"]
     }
   },
```

```
"Conditions":
 {
  "UseLogSNS":
   {
    "Fn::Not":
     [
      {
       "Fn::Equals":
        [
         {
          "Ref":"LogSNS"
         },"none"]
      }]
   },
  "UseSpot":
   {
    "Fn::Not":
     [
      {
       "Fn::Equals":
        [
         {
          "Ref":"SpotPrice"
         },0]
      }]
   }
 },
```

3.5.2 Mappings 宣言

テンプレートの Mappings ブロックでは、表から値を見つける方法を提供します。

AWSInstanceType2Arch マップでは、すべての使用可能なインスタンスタイプの値を定義します。PVM 値がインスタンスタイプを準仮想マシンとして定義するのに対し、HVM 値はインスタンスタイプをハードウェア仮想マシンとして定義します。また、AWSRegionArch2AMI マップでは、各リージョンの AMI を定義します。各リージョンは、PVM と HVM AMI の両方を持ちます。

HVM AMI の詳細については、
http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using_cluster_computing.html を参照してください。

```
"Mappings":
  {
    "AWSInstanceType2Arch":
     {
      "m1.small":
       {
        "Arch":"PVM"
       },
      "m1.medium":
       {
        "Arch":"PVM"
       },
      "m1.large":
       {
        "Arch":"PVM"
       },
      "m1.xlarge":
       {
        "Arch":"PVM"
       },
      "m2.xlarge":
       {
        "Arch":"PVM"
       },
      "m2.2xlarge":
       {
        "Arch":"PVM"
       },
      "m2.4xlarge":
       {
        "Arch":"PVM"
       },
      "m3.medium":
       {
        "Arch":"HVM"
       },
      "m3.large":
```

```
    {
     "Arch":"HVM"
    },

   "m3.xlarge":
    {
     "Arch":"HVM"
    },
   "m3.2xlarge":
    {
     "Arch":"HVM"
    },
   "c1.medium":
    {
     "Arch":"PVM"
    },
   "c1.xlarge":
    {
     "Arch":"HVM"
    },
   "cc1.4xlarge":
    {
     "Arch":"HVM"
    },
   "cc2.8xlarge":
    {
     "Arch":"HVM"
    },
   "cr1.8xlarge":
    {
     "Arch":"HVM"
    },
   "cg1.4xlarge":
    {
     "Arch":"HVM"
    },
   "hi1.4xlarge":
    {
     "Arch":"HVM"
    },
   "hs1.8xlarge":
    {
     "Arch":"HVM"
    },
   "i2.xlarge":
    {
     "Arch":"HVM"
    },
   "i2.2xlarge":
    {
     "Arch":"HVM"
    },
   "i2.4xlarge":
    {
```

```
      "Arch":"HVM"
     },

    "i2.8xlarge":
     {
      "Arch":"HVM"
     },
    "r3.large":
     {
      "Arch":"HVM"
     },
    "r3.xlarge":
     {
      "Arch":"HVM"
     },
    "r3.2xlarge":
     {
      "Arch":"HVM"
     },
    "r3.4xlarge":
     {
      "Arch":"HVM"
     },
    "r3.8xlarge":
     {
      "Arch":"HVM"
     }
   },
```

`AWSRegionArch2AMI` には、使用される AMI を記述します。これらは、MarkLogic の新しいリリースごとに変更されます。

```
  "AWSRegionArch2AMI":
   {
    "us-east-1":
     {
      "PVM":"ami-41633528",
      "HVM":"ami-4363352a"
     },
    "us-west-2":
     {
      "PVM":"ami-e85bc5d8",
      "HVM":"ami-ea5bc5da"
     },
    "eu-west-1":
     {
      "PVM":"ami-68fa1c1f",
      "HVM":"ami-56fa1c21"
     }
   }
 },
```

3.5.3 Resources 宣言

テンプレートの `Resources` ブロックでは、このテンプレートによってスタック用に作成されるすべての AWS リソースを定義します。各リソースは特定の AWS タイプとして定義されます。各リソースタイプの詳細については、
http://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-template-resource-type-ref.html
を参照してください。

このテンプレートでは次のようなリソースが定義されています。

- Elastic Block Store（EBS）ボリューム
- SimpleDB ドメイン（SimpleDB はメタデータデータベースの Amazon による実装です）
- AutoScaling グループ（ASG）。ASG ごとに、次のリソースが存在します。
    - セキュリティグループ
    - インスタンスタイプ
    - アイデンティティおよびアクセス管理（IAM）インスタンスプロファイル
    - 起動設定
    - UserData
    - Elastic Load Balancer（ELB）
- ELB ポート
- 稼動状況チェック値
- 各 EC2 インスタンスのセキュリティグループ

`Resources` ブロックの最初の部分では、Zone1、Zone2、および Zone3 の最初のノードについて `/var/opt/MarkLogic` が使用する EBS ボリュームを定義します。`AWS::EC2::Volume` タイプの詳細については、http://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-properties-ec2-ebs-volume.html を参照してください。

以下に示すように、すべての EBS ボリュームの定義は、Zone1 の `MarklogicVolume1` と同様です。

```
"Resources":
 {
  "MarklogicVolume1":
   {
    "Type":"AWS::EC2::Volume",
    "Properties":
     {
      "AvailabilityZone":
       {
        "Ref":"Zone1"
       },
      "Size":
       {
        "Ref":"VolumeSize"
       },
      "Tags":
       [
        {
         "Key":"Name",
         "Value":"MarkLogicData 1"
        }],
      "VolumeType":"standard"
     }
   },
```

`MarkLogicSDBDomain` では、「Amazon EC2 の用語」（5 ページ）で説明されているようにメタデータデータベースとして使用される SimpleDB データベースを作成し、SimpleDB ドメインの名前を返します。`AWS::SDB::Domain` タイプの詳細については、http://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-properties-simpledb.html を参照してください。

```
  "MarkLogicSDBDomain":
   {
    "Type":"AWS::SDB::Domain",
    "Properties":
     {
      "Description":"Simple DB auto-created and used as cluster name"
     }
   },
```

MarkLogicServerGroup1、MarkLogicServerGroup2、および MarkLogicServerGroup3 は、Zone1、Zone2、および Zone3 の AutoScaling グループ（ASG）です。AWS::AutoScaling::AutoScalingGroup タイプの詳細については、http://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-properties-as-group.html を参照してください。以下に示すように、これらはすべて Zone1 の MarkLogicServerGroup1 と同様です。

```
        "MarkLogicServerGroup1":
             {
              "Type":"AWS::AutoScaling::AutoScalingGroup",
              "Properties":
               {
                "AvailabilityZones":
                 [
                  {
                   "Ref":"Zone1"
                  }],
                "LaunchConfigurationName":
                 {
                  "Ref":"LaunchConfig1"
                 },
                "MinSize":"0",
                "MaxSize":
                 {
                  "Ref":"NodesPerZone"
                 },
                "DesiredCapacity":
                 {
                  "Ref":"NodesPerZone"
                 },
                "Cooldown":"300",
                "HealthCheckType":"EC2",
                "HealthCheckGracePeriod":"300",
                "LoadBalancerNames":
                 [
                  {
                   "Ref":"ElasticLoadBalancer"
                  }],
                "NotificationConfiguration":
                 {
                  "TopicARN":
                   {
                    "Ref":"LogSNS"
                   },
```

NotificationTypes には、SNS トピックに送信される通知を記述します。これは Cloud Formation スクリプトに提供され、AutoScaling グループのアクションを監視できるようになります。

```
                "NotificationTypes":

        ["autoscaling:EC2_INSTANCE_LAUNCH","autoscaling:EC2_INSTANCE_LAUNCH_
        ERROR","autoscaling:EC2_INSTANCE_TERMINATE","autoscaling:EC2_INSTANCE_
        TERMINATE_ERROR"]
                }
              }
            },
```

LaunchConfig1、LaunchConfig2、LaunchConfig3 は、ASG 1、ASG 2、および ASG 3 の起動設定です。これらには、リージョンに関連付けられている AMI ID、インスタンスタイプ、およびアーキテクチャ（PVM または HVM）を検索する方法を記述します。これらはすべて、下の ASG 1 用のコードと同様です。AWS::AutoScaling::LaunchConfiguration タイプの詳細については、http://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-properties-as-launchconfig.html を参照してください。

```
              "LaunchConfig1":
               {
                "Type":"AWS::AutoScaling::LaunchConfiguration",
                "Properties":
                 {
                  "KeyName":
                   {
                    "Ref":"KeyName"
                   },
                  "ImageId":
                   {
                    "Fn::FindInMap":
                     ["AWSRegionArch2AMI",
                      {
                       "Ref":"AWS::Region"
                      },
                      {
                       "Fn::FindInMap":
                        ["AWSInstanceType2Arch",
                         {
                          "Ref":"InstanceType"
                         },"Arch"]
                      }]
                  },
```

以下に示すように、各起動設定には UserData および SecurityGroups プロパティを記述します。

UserData プロパティには、「AWS 設定変数」（26 ページ）で説明されている変数に割り当てられるデータを入力します。ASG 1 の UserData プロパティは、次に示すとおりです。

注： VolumeSize の ,* では、各 ASG の 2 番目のノードおよび追加のノードのボリュームサイズを定義します。また # は、ノードの名前が動的に付けられ、数値のサフィックスが 1 - ManxNodesPerZone から追加されていることを示します。

```
"UserData":
{
 "Fn::Base64":
  {
   "Fn::Join":
    ["",
     ["MARKLOGIC_CLUSTER_NAME=",
      {
       "Ref":"MarkLogicSDBDomain"
      },"\n","MARKLOGIC_EBS_VOLUME=",
      {
       "Ref":"MarklogicVolume1"
      },",",
      {
       "Ref":"VolumeSize"
      },",*\n","MARKLOGIC_NODE_NAME=NodeA#\n",
       "MARKLOGIC_ADMIN_USERNAME=",
      {
       "Ref":"AdminUser"
      },"\n","MARKLOGIC_ADMIN_PASSWORD=",
      {
       "Ref":"AdminPass"
      },"\n","MARKLOGIC_CLUSTER_MASTER=1\n",
       "MARKLOGIC_LICENSEE=",
      {
       "Ref":"Licensee"
      },"\n","MARKLOGIC_LICENSE_KEY=",
      {
       "Ref":"LicenseKey"
      },"\n","MARKLOGIC_LOG_SNS=",
      {
       "Ref":"LogSNS"
      },"\n"]]
  }
},
```

各起動設定に記述されている `SecurityGroups` プロパティでは、`InstanceSecurityGroup` によって定義されているセキュリティグループを AutoScaling グループ内の Amazon EC2 インスタンスに割り当てます。各プロパティは次のようになります。

```
"SecurityGroups":
          [
           {
            "Ref":"InstanceSecurityGroup"
           }],
         "InstanceType":
          {
           "Ref":"InstanceType"
          },
         "IamInstanceProfile":
          {
           "Ref":"IAMRole"
          },
         "SpotPrice":
          {
           "Fn::If":
            ["UseSpot",
             {
              "Ref":"SpotPrice"
             },
             {
              "Ref":"AWS::NoValue"
             }]
          }
       }
     },
```

`ElasticLoadBalancer` は、すべての ASG のロードバランサーです。`AWS::ElasticLoadBalancing::LoadBalancer` タイプの詳細については、http://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-properties-ec2-elb.html を参照してください。

```
        "ElasticLoadBalancer":
             {
              "Type":"AWS::ElasticLoadBalancing::LoadBalancer",
              "Properties":
               {
                "AppCookieStickinessPolicy":
                 [
                  {
                   "CookieName":"SessionID",
                   "PolicyName":"MLSession"
                  }],
                "AvailabilityZones":
                 [
                  {
                   "Ref":"Zone1"
                  },
                  {
                   "Ref":"Zone2"
                  },
                  {
                   "Ref":"Zone3"
                  }],
                "ConnectionDrainingPolicy":
                 {
                    "Enabled":"true",
                    "Timeout":"60"
                 },
                "CrossZone":"true",
```

`Listeners` では、Elastic Load Balancer（ELB）が公開するすべてのポートを定義します。

```
            "Listeners":
                 [
                  {
                   "LoadBalancerPort":"8000",
                   "InstancePort":"8000",
                   "Protocol":"HTTP",
                   "PolicyNames":
                    ["MLSession"]
                  },
                  {
                   "LoadBalancerPort":"8001",
                   "InstancePort":"8001",
                   "Protocol":"HTTP",
                   "PolicyNames":
                    ["MLSession"]
                  },
                  {
                   "LoadBalancerPort":"8002",
                   "InstancePort":"8002",
                   "Protocol":"HTTP",
                   "PolicyNames":
                    ["MLSession"]
                  },
                  {
                   "LoadBalancerPort":"8003",
                   "InstancePort":"8003",
                   "Protocol":"HTTP",
                   "PolicyNames":
                    ["MLSession"]
                  },
                  {
                   "LoadBalancerPort":"8004",
                   "InstancePort":"8004",
                   "Protocol":"HTTP",
                   "PolicyNames":
                    ["MLSession"]
                  },
                  {
                   "LoadBalancerPort":"8005",
                   "InstancePort":"8005",
                   "Protocol":"HTTP",
                   "PolicyNames":
                    ["MLSession"]
                  },
                  {
                   "LoadBalancerPort":"8006",
                   "InstancePort":"8006",
                   "Protocol":"HTTP",
                   "PolicyNames":
                    ["MLSession"]
                  },
```

```
        {
         "LoadBalancerPort":"8007",
         "InstancePort":"8007",
         "Protocol":"HTTP",
         "PolicyNames":
          ["MLSession"]
        },
        {
         "LoadBalancerPort":"8008",
         "InstancePort":"8008",
         "Protocol":"HTTP",
         "PolicyNames":
          ["MLSession"]
        }],
```

HealthCheck はポート 7997 上で、Interval によって指定された間隔（秒単位）で HealthCheck アプリケーションサーバーに接続することにより、各 MarkLogic インスタンスの稼動状況をチェックします。Timeout 時間（秒単位）以内に「200 OK」以外の応答があった場合は、正常に稼動していないものと見なされ、そのインスタンスが ELB から削除されます。HealthCheck パラメータの詳細については、http://docs.aws.amazon.com/ElasticLoadBalancing/latest/APIReference/API_HealthCheck.html を参照してください。

```
      "HealthCheck":
       {
        "Target":"HTTP:7997/",
        "HealthyThreshold":"3",
        "UnhealthyThreshold":"5",
        "Interval":"10",
        "Timeout":"5"
       }
     }
},
```

InstanceSecurityGroup では、各 EC2 インスタンスのセキュリティグループを定義します。AWS::EC2::SecurityGroup タイプの詳細については、http://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-properties-ec2-security-group.html を参照してください。

```
"InstanceSecurityGroup":
{
 "Type":"AWS::EC2::SecurityGroup",
 "Properties":
  {
   "GroupDescription":"Enable SSH access and HTTP
    access on the inbound port",
   "SecurityGroupIngress":
    [
     {
      "IpProtocol":"tcp",
      "FromPort":"22",
```

```
          "ToPort":"22",
          "CidrIp":"0.0.0.0/0"
         },
         {
          "IpProtocol":"tcp",
          "FromPort":"7998",
          "ToPort":"7998",
          "CidrIp":"0.0.0.0/0"
         },
         {
          "IpProtocol":"tcp",
          "FromPort":"8000",
          "ToPort":"8010",
          "SourceSecurityGroupOwnerId":
           {
            "Fn::GetAtt":
             ["ElasticLoadBalancer","SourceSecurityGroup.OwnerAlias"]
           },

          "SourceSecurityGroupName":
           {
            "Fn::GetAtt":
             ["ElasticLoadBalancer","SourceSecurityGroup.GroupName"]
           }
         },
         {
          "IpProtocol":"tcp",
          "FromPort":"8000",
          "ToPort":"8010",
          "CidrIp":"0.0.0.0/0"
         },
         {
          "IpProtocol":"tcp",
          "FromPort":"7997",
          "ToPort":"7997",
          "SourceSecurityGroupOwnerId":
           {
            "Fn::GetAtt":
             ["ElasticLoadBalancer","SourceSecurityGroup.OwnerAlias"]
           },
          "SourceSecurityGroupName":
           {
            "Fn::GetAtt":
             ["ElasticLoadBalancer","SourceSecurityGroup.GroupName"]
           }
         }]
    }
  },
```

InstanceSecurityGroupIngress では、インスタンスに到達することが許可される受信トラフィックを制御する受信ルールを定義します。AWS::EC2::SecurityGroupIngress タイプの詳細については、http://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-properties-ec2-security-group-ingress.html を参照してください。

注： InstanceSecurityGroupIngress が InstanceSecurityGroup から分けられているのは、セキュリティグループ自体を参照する必要があるためです。

```
"InstanceSecurityGroupIngress":
     {
      "Type":"AWS::EC2::SecurityGroupIngress",
      "Properties":
       {
        "IpProtocol":"tcp",
        "GroupName":
         {
          "Ref":"InstanceSecurityGroup"
         },
        "FromPort":"7999",
        "ToPort":"7999",
        "SourceSecurityGroupName":
         {
          "Ref":"InstanceSecurityGroup"
         }
       }
     }
   },
```

3.5.4 Outputs 宣言

CloudFormation の起動に成功すると、Outputs によって、MarkLogic の Admin 画面のポート（8001）をポイントする ELB の URL が生成されます。

```
"Outputs":
 {
  "URL":
   {
    "Description":"The URL of the MarkLogic Cluster",
    "Value":
     {
      "Fn::Join":
       ["",
        ["http://",
         {
          "Fn::GetAtt":
           ["ElasticLoadBalancer","DNSName"]
         },":8001"]]
     }
   }
 }
```

3.6 CloudFormation スタックの削除

CloudFormation スタックを削除するには、
http://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/cfn-console-delete-stack.html で説明されている手順に従います。

CloudFormation スタックを削除すると、CloudFormation テンプレートによって作成されたほとんどの EC2 リソース（インスタンス、セキュリティグループなど）も削除されます。しかし EBS ボリュームは例外であり、削除されません。スタックの削除後に EBS ボリュームを削除する必要がある場合は、
http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-deleting-volume.html で説明されている手順に従って手動で削除してください。

> **注：** スタックを削除すると、各ゾーン内の最初のノード用に作成された EBS ボリュームも削除されます。ただし、各ゾーン内の追加ノード用の EBS ボリュームは削除されません。これは、そのようなボリュームが CloudFormation スタック内で直接作成されたものではなく、追加ノードの起動プロセスの一部として作成されているためです。

4.0 EC2 上の MarkLogic サーバーの管理

この章では、MarkLogic サーバー AMI を起動し、MarkLogic サーバー Admin 画面にアクセスする方法について説明します。この章は、次のセクションで構成されています。

- MarkLogic サーバーインスタンスへのアクセス
- EC2 インスタンスへのアクセス
- EC2 エラーの検出
- mlcmd スクリプトの使用
- S3（Amazon Simple Storage Service）に対応した MarkLogic の設定
- EC2 でクラスタリソースのスケーリング
- MarkLogic AMI のアップグレード
- モニタリング（CloudWatch）
- エンタープライズデータセンターから EC2 への移行
- EBS ボリュームの作成およびインスタンスへのアタッチ
- MarkLogic クラスタの一時停止または終了

4.1 MarkLogic サーバーインスタンスへのアクセス

このセクションでは、EC2 で MarkLogic サーバーのインスタンスにアクセスする方法について説明します。

MarkLogic サーバーインスタンスには、次の 2 種類の方法でアクセスできます。

- Elastic Load Balancer（ELB）URL を使用する
- インスタンスのパブリック DNS を使用する

相違点は、ELB URL では、クラスタ内で使用可能な任意の MarkLogic サーバーのインスタンスに接続されることです。「mlcmd スクリプトの使用」（70 ページ）で説明されている `mlcmd` スクリプトを実行するときと同様に、特定のインスタンスにアクセスする必要がある場合は、そのインスタンスのパブリック DNS を使用します。

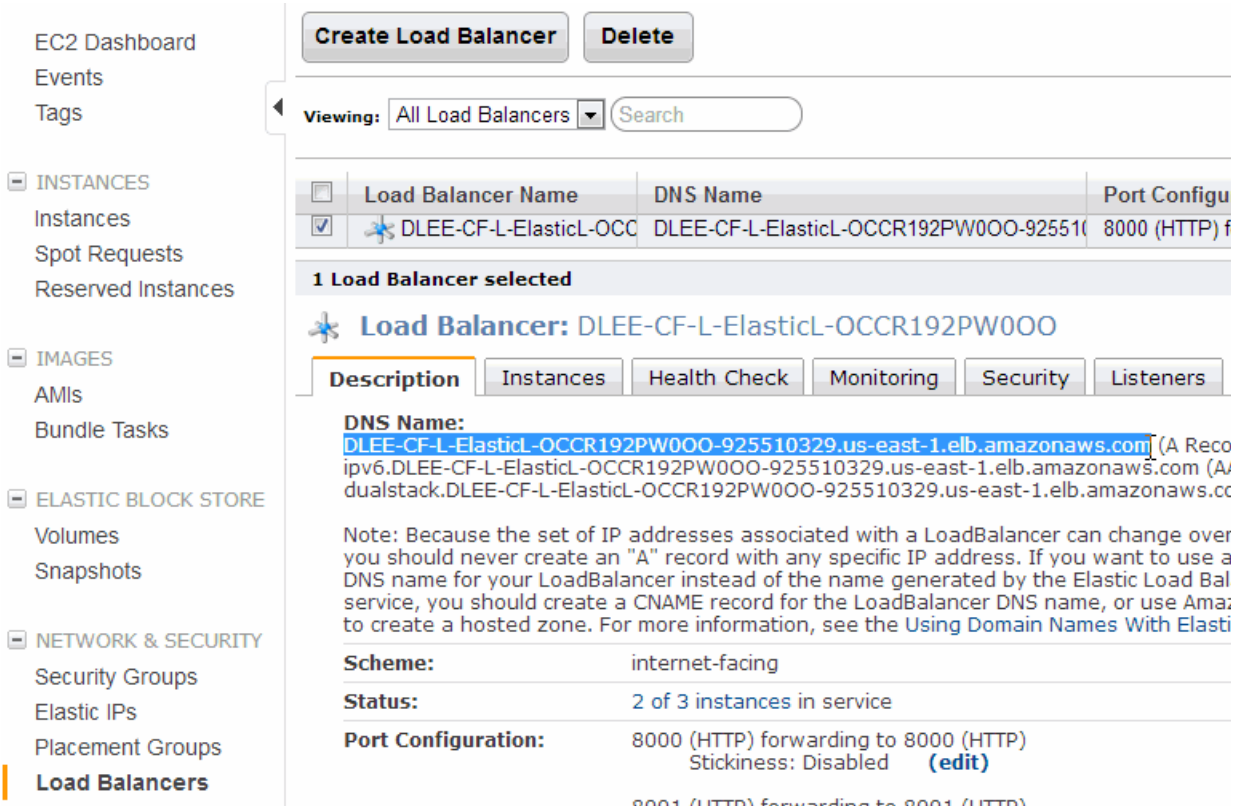
4.1.1 ELB を介した MarkLogic サーバーへのアクセス

「AWS Console を使用した CloudFormation スタックの作成」（35 ページ）のステップ 11 で説明されているように、CloudFormation Console の［Outputs］部分で URL をクリックすることにより、ELB を介して MarkLogic Admin 画面にアクセスできます。

このセクションでは、EC2 ダッシュボードから ELB を介して MarkLogic サーバーにアクセスする方法について説明します。

1. EC2 ダッシュボードで、左側のナビゲーションメニューにある［Load Balancers］をクリックします。

2. DNS 名から URL をコピーします。



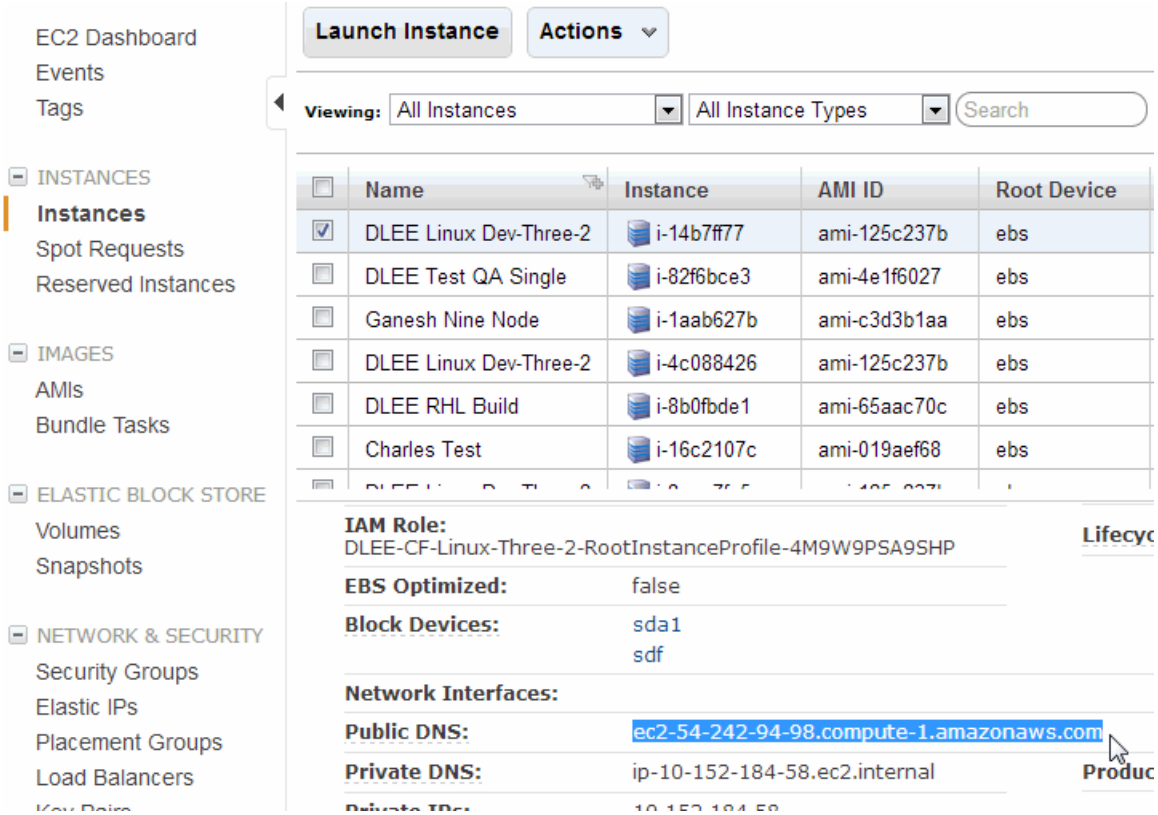
この URL を使用して MarkLogic サーバーにアクセスします。ELB ポートとして定義した任意のポートにアクセスできます。例えば、URL が `DLEE-CF-L-ElasticL-OCCR192PW0OO-925510329.us-east-1.elb.amazonaws.com` である場合、Admin 画面にアクセスするには、ブラウザで次の URL を入力します。

```
http://DLEE-CF-L-ElasticL-OCCR192PW0OO-925510329.us-east-
1.elb.amazonaws.com:8001
```

4.1.2 インスタンスのパブリック DNS を介した MarkLogic サーバーへのアクセス

このセクションでは、インスタンスのパブリック DNS を介して MarkLogic サーバーにアクセスする方法について説明します。

1. EC2 ダッシュボードで、左側のナビゲーションメニューにある［Instances］をクリックします。目的の MarkLogic サーバーインスタンスを選択し、パブリック DNS 値をコピーします。



パブリック DNS は、MarkLogic サーバーにアクセスするための URL の一部として使用します。例えば、パブリック DNS が `ec2-54-242-94-98.compute-1.amazonaws.com` である場合、Admin 画面にアクセスするには、ブラウザで次の URL を入力します。

```
http://ec2-54-242-94-98.compute-1.amazonaws.com:8001
```

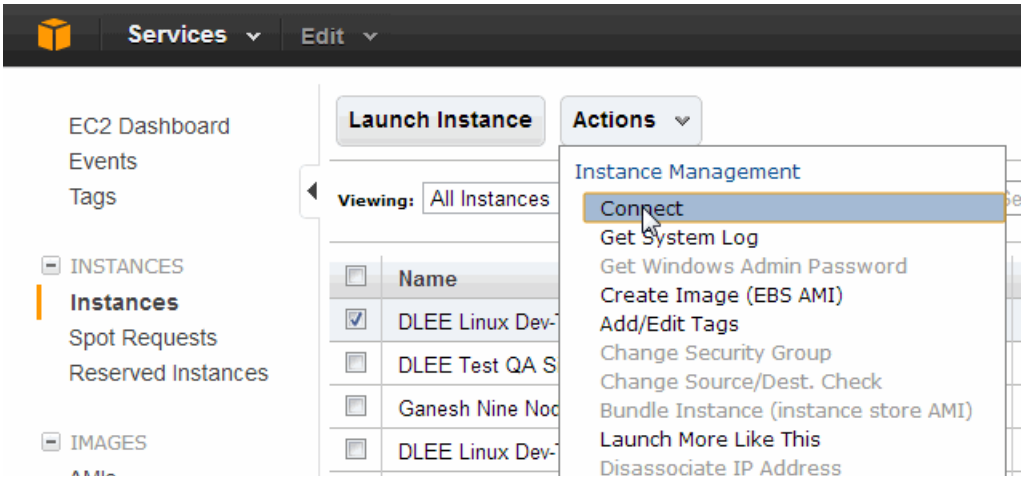
4.2 EC2 インスタンスへのアクセス

EC2 インスタンスのログファイルをチェックする場合など、実行するタスクの内容によっては、「EC2 エラーの検出」（70 ページ）で説明されているように、EC2 インスタンスに SSH で接続する必要があります。

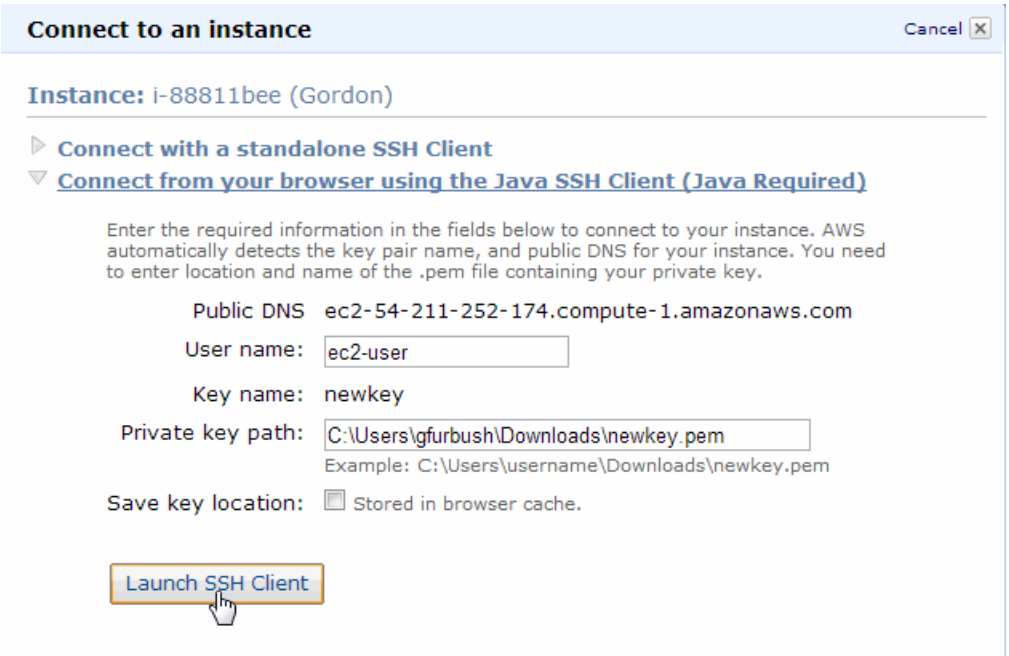
> **注：** ロードバランサーには SSH で接続できません。特定の EC2 インスタンスに SSH で接続する必要があります。EC2 インスタンスに SSH で接続するには、ローカルホストにダウンロードした、そのインスタンスで使用されているキーペアが必要です。

インスタンスに SSH で接続するには、次の手順を実行します。

1. EC2 ダッシュボードを開きます。
2. 左側のナビゲーションセクションで［Instances］を選択します。
3. 接続先のインスタンスを選択します。
4. ［Actions］プルダウンメニューから［Connect］を選択します。



5. ユーザー名として ec2-user を指定し、ローカルホストにダウンロードしたキーペアのコピーのパスを入力します。［Launch SSH Client］をクリックします。



6. これにより、EC2 インスタンスを対象としたシェルウィンドウが表示されます。この方法で初めて接続するときには、各種ディレクトリを作成するように要求するメッセージが表示されることがあります。その場合は、各メッセージに対して［Yes］をクリックします。

またはシェルウィンドウを開き、次のコマンドを使用して SSH でインスタンスに接続することもできます。

```
ssh -i /path/to/keypair.pem ec2-user@<Public DNS>
```

例えば、newkey.pem という名前のキーペアが c:/stuff/ ディレクトリに格納されている場合は、次のように ec2-54-242-94-98.compute-1.amazonaws.com というパブリック DNS を使用してインスタンスにアクセスできます。

```
ssh -i c:/stuff/newkey.pem ec2-user@ec2-54-242-94-98.compute-1.amazonaws.com
```

4.3 EC2 エラーの検出

起動エラーは、各インスタンスの /var/log/messages ファイルに格納されます。messages ファイルを表示するには、「EC2 インスタンスへのアクセス」（68 ページ）の説明に従ってインスタンスに SSH で接続します。

messages ファイルにアクセスするには、スーパーユーザーになる必要があります。例えば、messages ファイルの末尾を表示するには、次のように入力します。

```
sudo tail -f /var/log/messages
```

「SNS（Simple Notification Service）トピックの作成」（23 ページ）で説明されているように、SNS トピックを使用して CloudFormation スタックに関連するエラーを取得することもできます。

4.4 mlcmd スクリプトの使用

mlcmd スクリプトは、起動操作および管理対象クラスタ機能の高度な使用方法をサポートしています。mlcmd スクリプトは、実行可能スクリプトとして /opt/MarkLogic/bin/mlcmd にインストールされます。

mlcmd を実行するには、ホストにログインし、ルートとして、またはルート権限を使用して実行する必要があります。さらに Java をインストールし、java コマンドを PATH に含めるか、JAVA_HOME を JRE または JDK ホームディレクトリに設定する必要があります。サーバーでの MarkLogic の初回起動時は、mlcmd スクリプトを使用するために必要な /var/local/mlcmd.conf ファイルが作成されます。/var/local/mlcmd.conf ファイルが作成されている場合は、mlcmd スクリプトを使用するために MarkLogic を起動する必要はありません。

mlcmd のシンタックスは次のとおりです。

```
mlcmd command
```

`mlcmd` コマンドのリストは、次に示すとおりです。

mlcmd コマンド	説明
sync-volumes-from-mdb	このインスタンスに現在アタッチされていない EBS ボリュームをアタッチします。
sync-volumes-to-mdb	システムに現在アタッチされている EBS ボリュームを同期し、それらをメタデータデータベースに格納します。
init-volumes-from-system	起動時に実行されたプロセスと同じボリュームを初期化します。
leave-cluster	実行場所のホストをクラスタから削除します。

4.4.1 sync-volumes-from-mdb

このコマンドは、メタデータデータベース内を検索し、次の処理を実行します。

- このインスタンスに現在アタッチされていない EBS ボリュームを見つけて、それらをアタッチします。
- ボリュームにファイルシステムが含まれていない場合は、ファイルシステム（ext4）を作成します。
- メタデータデータベース内で示されているマウントポイントにデバイスをマウントします。
- `marklogic:` というプレフィックスが付いた現在の EC2 インスタンスのすべてのタグを EBS ボリュームに適用します。

4.4.2 sync-volumes-to-mdb

このコマンドは初回起動後であればいつでも実行できます。システムに現在アタッチされている EBS ボリュームを同期してメタデータデータベースに格納して、次の再起動時にそのようなボリュームがアタッチおよびマウントされるようにします。次の手順が実行されます。

- システムに対するすべての EBS ボリュームを見つけます。
- 管理対象範囲内のすべてのボリュームについて、次の情報を示すエントリをメタデータデータベースに入力します。
    - EBS ボリューム ID
    - EBS マウントデバイス

    - オペレーティングシステムのマウントデバイス
    - オペレーティングシステムのマウントポイント（ディレクトリ）
- アタッチされていてマウントされていないボリュームの場合、マウントポイントは、そのボリュームのデフォルトのマウントポイントに設定されます。

既存の添付ファイル、ファイルシステム、またはマウントポイントに対する変更は実行されません。

4.4.3 init-volumes-from-system

このコマンドは、現在のシステムを参照し、起動時に実行されたプロセスと同じボリュームの初期化を試みます。

- ユーザーデータ変数 MARKLOGIC_EBS_VOLUME<N> として一覧表示される各ボリュームについて次の処理が実行されます。
    - 必要に応じてボリュームをシステムにアタッチします。
    - 必要に応じてファイルシステムを作成します。
- 管理対象範囲内のシステムにアタッチされている各 EBS ボリュームについて次の処理が実行されます。
    - 必要に応じてファイルシステムを作成します。
    - デバイスをデフォルトのマウントポイント（またはメタデータデータベース内で現在指定されているマウントポイント）にマウントします。
    - 現在の EBS ボリューム、OS デバイスおよびマウントポイントによってメタデータデータベースを更新します。

4.4.4 leave-cluster

このコマンドをホスト上で実行すると、そのホストをクラスタから削除できます。また、メタデータデータベース内に格納されているクラスタ設定情報からも、そのホストが削除されます。 このコマンドでは、ホストサーバーが事前に初期化された状態（新規インストールと同じ）に維持されます。そのサーバーが再起動されると、最初の起動時と同じ方法でクラスタに再び参加します。

オプションの `-terminate` 引数を使用してインスタンスを終了すると、クラスタから削除した後で AutoScaling グループの `DesiredCount` 属性が 1 だけ減少します。

4.5 S3（Amazon Simple Storage Service）に対応した MarkLogic の設定

Amazon S3 は、使用可能なファイルシステムタイプとして MarkLogic サーバーでサポートされています。S3 アクセスはグローバルレベルで設定します。S3 用にグループを設定すると、S3 パスを指定することで、グループ内の任意のフォレストを S3 上に配置できるようになります。さらに、グループ内の任意のホストで、S3 へのバックアップ、S3 からのリストア、S3 上のディレクトリおよびファイルの読み取りと書き込みを実行できます。

> **注：** S3 ファイルシステムは、ジャーナルの維持に必要なファイル操作を実行できないので、トランザクションジャーナリングは S3 上では機能しません。S3 フォレストが高速データディレクトリを使用して設定されている場合や、許可されている更新が `read-only` に設定されている場合を除き、フォレストをデータベースにアタッチする前にデータベースのジャーナリングオプションを `off` に設定しておく必要があります。ただし、これはデータベースに対するバックアップ / リストア操作では必須ではありません。

Amazon S3 にアクセスできるように MarkLogic を設定するには、次の操作を実行します。

- S3 バケットのセットアップ
- グループの S3 エンドポイントの設定
- S3 資格情報の設定
- フォレストのデータディレクトリでの S3 パスの設定
- テストを目的とした MarkLogic へのコンテンツのロード

4.5.1 S3 バケットのセットアップ

http://docs.aws.amazon.com/gettingstarted/latest/wah/getting-started-create-bucket.html の指示に従って、S3 バケットをセットアップします。

警告 バケット名にピリオド（.）が含まれている場合、複数の問題が発生する可能性があります。S3 との互換性を最大限に高めるため、代わりにダッシュ（-）を使用してください。

バケット名はグローバルであり、その範囲はアカウントに限定されません。ユニバーサルに一意である可能性が高いバケット名を選択する必要があります。例えば次のようになります。

- 悪い例：`test`
- 良い例：`zippy-software-org-test`

注： コンテンツを S3 にアップロードする場合は、S3 Management Console を使用しないでください。その代わりに、設定手順を完了した後で、『*Loading Content Into MarkLogic Server Guide*』で説明されている手順に従ってください。

4.5.2 グループの S3 エンドポイントの設定

S3 エンドポイントは、MarkLogic グループの S3 プロパティを指定することによって設定します。

1. Admin 画面にログインします。
2. 左側のツリーメニューで、［Groups］アイコンの下にあるグループの名前をクリックします。
3. ［Group Configuration］ページで一番下までスクロールし、S3 フィールドを見つけます。



S3 フィールドを次のように設定します。

設定	説明
s3 domain	S3 エンドポイントで使用されるドメイン。デフォルト値はリージョン（地域）ごとに設定されます。ただし、必要に応じて変更できます。各リージョンのエンドポイントの一覧については、http://docs.aws.amazon.com/general/latest/gr/rande.html#s3_region を参照してください。
s3 protocol	S3 との通信用に `http` または `https` を選択できます。デフォルトは `http` です。
s3 server side encryption	S3 上のストレージは、サーバー側の暗号化に参加できます。デフォルトは `none` ですが、`aes256` に設定してサーバー側の暗号化を有効にできます。

4.5.3 S3 資格情報の設定

S3 を使用するには、S3 資格情報を指定する必要があります。S3 資格情報は次の 3 つのうちのいずれかの方法で設定できます。

- セキュリティデータベースでの S3 資格情報の設定
- 環境変数での S3 資格情報の設定
- S3 アクセスポリシーを使用した IAM ロールの設定

S3 資格情報は次の優先順位で検出されます。

1. サーバーのセキュリティデータベースで設定されている資格情報
2. 環境変数
3. IAM ロール

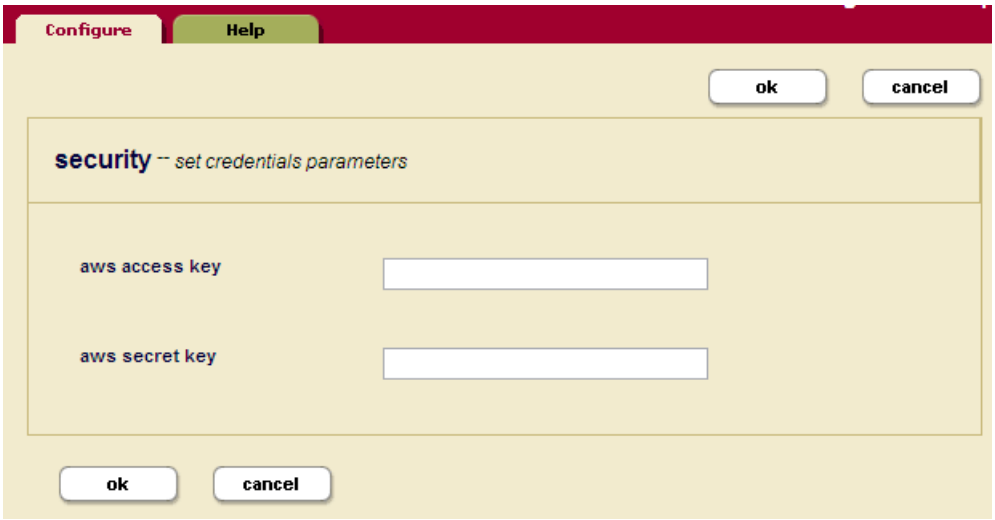
4.5.3.1 セキュリティデータベースでの S3 資格情報の設定

S3 資格情報は、［Security］、［Credentials］、［Configure］タブのフィールドで指定します。

1. Admin 画面にログインします。
2. 左側のツリーメニューで、［Security］アイコンをクリックします。

3. ［Credentials］をクリックして、［Credential Configuration］ページを開きます。

4. AWS アカウント用に提供されている AWS アクセスキーと AWS 秘密キーを入力します。



4.5.3.2 環境変数での S3 資格情報の設定

サーバーが S3 AWS 資格情報として使用する環境変数のペアを設定できます。この情報は、EC2 ユーザーデータとして渡すか、MarkLogic が実行される環境に対して設定できます。

- MARKLOGIC_AWS_ACCESS_KEY -- AWS アクセスキー
- MARKLOGIC_AWS_SECRET_KEY -- AWS 秘密キー

4.5.3.3 S3 アクセスポリシーを使用した IAM ロールの設定

IAM ロールが関連付けられている EC2 インスタンスを実行する場合は、「Amazon S3 Full Access」、「Amazon S3 Read Only Access」などの S3 アクセスを提供するポリシーテンプレートを選択できます。

IAM ロールは、セキュリティ資格情報で使用されるので、S3 リソースにアクセスするために、AWS 資格情報を MarkLogic または EC2 インスタンスに格納する必要はありません。これは最も安全な S3 へのアクセス方法です。

4.5.4 フォレストのデータディレクトリでの S3 パスの設定

フォレストのデータディレクトリを有効な S3 パスに設定します。フォレストのデータディレクトリの設定方法の詳細については、『*Administrator's Guide*』の「Creating a Forest」を参照してください。同一バケットに対して複数のフォレストを設定できます。

S3 パスの形式は次のとおりです。

```
s3://bucket/directory/file
```

各値の意味は次のとおりです。

項目	説明
bucket	S3 バケットの名前。
directory	スラッシュ（/）で区切られた 0 個以上のディレクトリ名。
file	ファイル名（特定のファイルに対応するパスの場合）。

ディレクトリパス（フォレストのデータディレクトリなど）の場合は、バケット自体で十分であり、ファイルはバケットのルートに配置されます

S3 ディレクトリのパスの例：

```
s3://my-company-bucket

s3://my-company-bucket/directory

s3://my-company-bucket/dir1/dir2/dir3
```

S3 ファイルのパスの例：

```
s3://my-company-bucket/file.xml

s3://my-company-bucket/directory/file.txt

s3://my-company-bucket/dir1/dir2/dir3/file.txt
```

警告 S3 フォレストが高速データディレクトリを使用して設定されている場合や、許可されている更新が `read-only` に設定されている場合を除き、フォレストをデータベースにアタッチする前にデータベースのジャーナリングを `off` に設定しておく必要があります。このようにしないとフォレストエラーになり、データベースでジャーナリングを無効にした後でフォレストを再起動する必要があります。

4.5.5 テストを目的とした MarkLogic へのコンテンツのロード

『*Loading Content Into MarkLogic Server Guide*』で説明されているいずれかの手法を使用して S3 データベースにコンテンツをロードし、クエリを実行して MarkLogic サーバーが S3 上で正常に設定されていることを確認します。

> **注：** S3 Management Console を使用してバケットに直接アップロードされたコンテンツは、MarkLogic サーバーによって認識されません。

4.6 EC2 でクラスタリソースのスケーリング

3+ CloudFormation テンプレートを使用してスタックを作成している場合は、使用率が高い期間に対応するためにノードとフォレストを一時的に追加してクラスタをスケールアップし、後で必要なリソースが低減した場合に削除できます。

クラスタにホストを追加する方法はシンプルです。［Update Stack］機能を使用して 3+ CloudFormation テンプレートを再適用し、`NodesPerZone` 設定の数値を増やすだけです。また、AutoScaling グループを使用してホストを追加することもできます。クラスタのデータ容量をスケールアップする場合は、「EBS ボリュームの作成およびインスタンスへのアタッチ」（81 ページ）の説明に従ってボリュームを追加する方法が推奨されます。

クラスタをスケールダウンする場合は、いくつかの手動による操作が必要です。次の手順を実行します。

1. MarkLogic および AWS ツールを使用して、各 ASG 内で削除対象となる同じ個数のホストを特定します。Security、Meters、App-Services、Modules などのビルトイン MarkLogic データベースが搭載されているホストは削除しないでください。

2. 削除するホストからデータを削除するか他のホストにデータを移動します。この操作を行うには、REST 管理 API または XQuery の `tieredstorage` API を使用して、パーティションまたはフォレストを他のホスト上のボリュームに移行します。データの移行の詳細については、『*Administrator's Guide*』の「Migrating Forests and Partitions」を参照してください。

3. 削除する各ホスト上でスーパーユーザーとして leave-cluster `-terminate` コマンドを実行します。これによりノードがクラスタから削除され、AutoScaling グループの `DesiredCount` 設定が調整されます。詳細については、「leave-cluster」（72 ページ）を参照してください。

4. 未使用のボリュームを削除します。

5. サイズを縮小したクラスタが反映されるように CloudFormation テンプレートを更新し、［Update Stack］機能を使用してテンプレートをスタックに再適用し、AWS に設定の更新を通知します。

4.7 MarkLogic AMI のアップグレード

このセクションでは、MarkLogic AMI をアップグレードする手順について説明します。ここで説明するのは、CloudFormation を使用して作成されたインスタンスおよび AMI から直接組み込まれたインスタンスに対応できるように AMI をアップグレードする手順です。

次の手順は、3 つ以上のノードの CloudFormation テンプレートを使用して作成されたインスタンスに対応できるように AMI をアップグレードする方法について説明したものです。

1. アップグレードの前に、重要なデータをすべてバックアップしておきます。

2. 新しい AMI を使用する更新された CloudFormation テンプレートを使用してスタックを更新します。スタックの更新が完了していることを確認します。

3. EC2 ダッシュボードでインスタンスを 1 つずつ停止し、新しいインスタンスに置き換えられるまで待ちます。

   注： アップグレードの前に CloudFormation の外部で変更が加えられると、アップグレードが失敗する原因になります。

4. EC2 ダッシュボードで、他のノードを終了します（できれば 1 つずつ終了してください。これらは UI を操作しなくても起動して再接続されます）。

5. 新しいインスタンスでポート 8001 にアクセスすると、アップグレードのメッセージが表示されます（`security-upgrade.xqy` 画面）。

6. ［OK］をクリックし、インスタンスに対するアップグレードが完了するまで待ちます。

次の手順は、AMI から直接組み込まれたインスタンスをアップグレードする方法について説明したものです。クラスタ内の各 MarkLogic インスタンスについて、次の手順を実行します。

1. インスタンスを終了します。

2. アップグレードされた AMI から新しいインスタンスを起動します。

3. 元のインスタンスに関連付けられている EBS データボリュームをアタッチします。

   注： アップグレードの前にインスタンスまたは AMI をカスタマイズすると、アップグレードが失敗する原因になる場合があります。

4.8 モニタリング（CloudWatch）

AWS では、EC2 インスタンス、ESB ボリューム、および他のサービスに対する強固なモニタリング機能を CloudWatch サービスによって実現しています。CloudWatch を使用して AWS サービスごとにしきい値を設定し、それらのしきい値を超えたときに SMS またはメールを使用して通知を送信できます。例えば、過度のストレージスループットに対してしきい値を設定できます。また、CloudWatch でモニタリングするための独自のメトリックを作成することもできます。例えば、インスタンスの現在の空きメモリをモニタリングするためのカスタムメトリックを作成し、メモリのしきい値を超えた場合に、アラームまたは自動応答をトリガーできます。

CloudWatch の使用方法の詳細については、http://docs.aws.amazon.com/AmazonCloudWatch/latest/DeveloperGuide/CHAP_UsingCloudWatch.html を参照してください。

4.9 エンタープライズデータセンターから EC2 への移行

このセクションでは、データおよび設定をデータセンターから EC2 に移行する手順を説明します。

ローカルのデータセンターから EC2 に移行する方法はいくつかあります。次に示す手順は、その一例です。

1. S3 ストレージにデータベースをバックアップします。この作業を実行するには、ローカルの MarkLogic クラスタで「S3 資格情報の設定」（75 ページ）の説明に従って S3 セキュリティ資格情報を設定し、「フォレストのデータディレクトリでの S3 パスの設定」（77 ページ）の説明に従って、バックアップ ディレクトリ用に S3 バケットへのパスを指定します。

2. 『*Administrator's Guide*』の「Exporting a Configuration」の説明に従い、Configuration Manager を使用して、クラスタのすべての設定データをエクスポートします。

3. 「CloudFormation を使用した EC2 での MarkLogic の導入」（30 ページ）の説明に従い、CloudFormation テンプレートを作成して、EC2 上にクラスタのホストを再作成します。

4. 『*Administrator's Guide*』の「Importing a Configuration」の説明に従い、設定データを EC2 クラスタにインポートします。

5. バックアップしたデータを S3 から設定済みの EC2 フォレストにリストアします。

4.10 EBS ボリュームの作成およびインスタンスへのアタッチ

このセクションでは、EBS ボリュームを作成して、MarkLogic サーバーインスタンスにアタッチする方法について説明します。

一般的に、ノードごとに 1 つのボリュームを作成し、ボリュームごとに 1 つのフォレストを使用することをお勧めします。

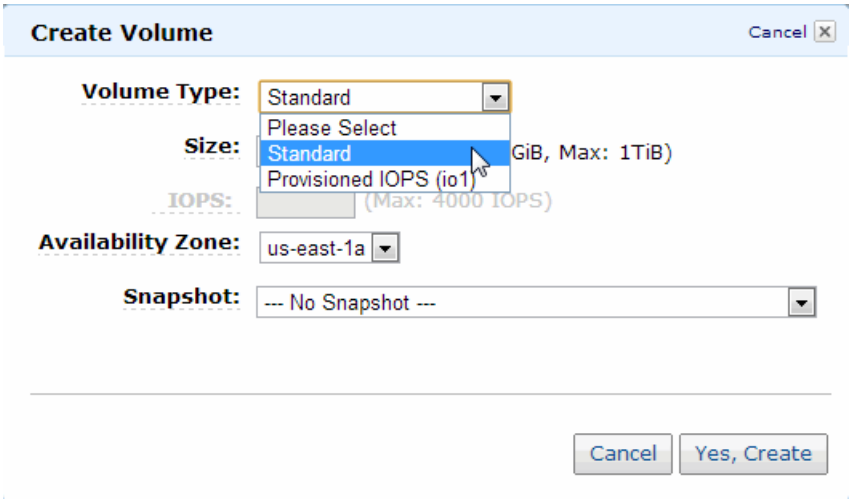
4.10.1 EBS ボリュームの作成

EBS ボリュームを作成するには、次の手順を実行します。

1. EC2 ダッシュボードを開き、左側のナビゲーションセクションで［Volumes］を選択します。［EBS Volumes］ページで、［Create Volume］を選択します。



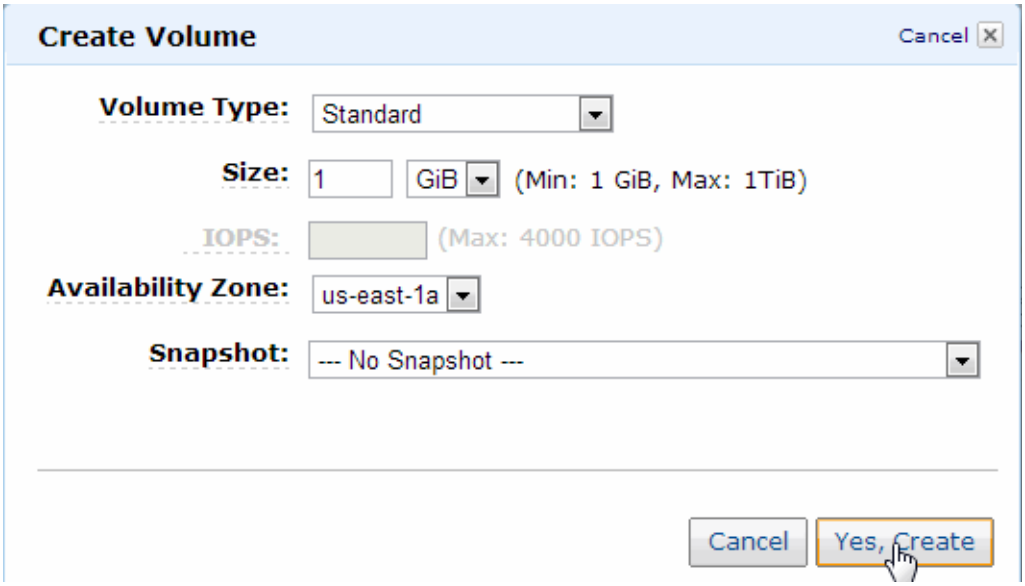
2. ［Create Volume］ウィンドウで、［Volume Type］プルダウンメニューからボリュームタイプを指定します。



3. ニーズを満たし、インスタンスに関連付けられている同じアベイラビリティゾーンに対して十分な大きさのボリュームサイズを指定します。ボリュームをアタッチする予定のインスタンスと同じゾーンを指定します。オプションで EBS スナップショットを指定することもできます。スナップショットの作成方法の詳細については、EBS スナップショットに関するページをヘルプで参照してください。

   警告 デフォルトでは、インスタンスと EBS ボリュームのゾーンは異なる場合があります。

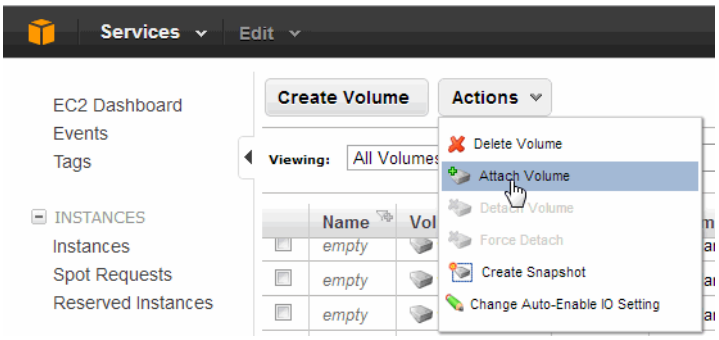
   操作が完了したら、[Yes, Create] をクリックします。管理コンソールの右側のセクションでこの新しいボリュームへの参照を見つけ、[State] が [`available`] になっていることを確認します。



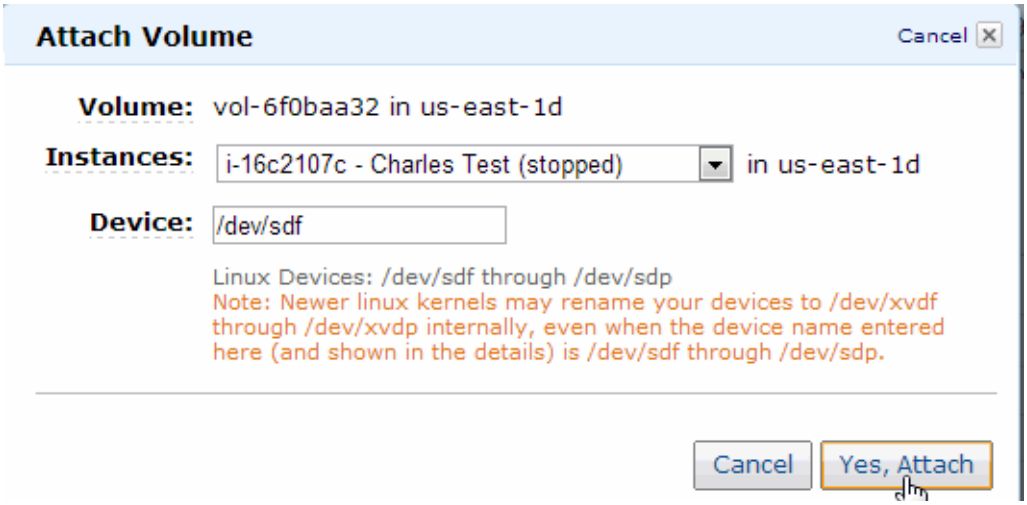
4.10.2 EBS ボリュームのインスタンスへのアタッチ

このセクションでは、EC2 ダッシュボードを使用して、ボリュームをインスタンスにアタッチする方法について説明します。

1. 左側のナビゲーションセクションでボリュームを選択し、[Attach Volume] をクリックします。



2. [Attach Volume] ウィンドウで、MarkLogic サーバー AMI から起動するインスタンスを指定します。[Device] セクションでは、`/dev/sdf` を使用します。操作が完了したら、[Yes, Attach] をクリックします。管理コンソールの右側のセクションでこのボリュームへの参照を見つけ、ステータスが [in-use] になっていることを確認します。ステータスが [in-use] になっていない場合は、ステータスが [in-use] に変わるまで [Refresh] をクリックし続けます。



3. インスタンスに SSH で接続し、init-volumes-from-system コマンドを実行してボリュームのファイルシステムを作成し、新しいボリュームの設定を使用してメタデータデータベースを更新します。init-volumes-from-system コマンドを実行すると、実行中の処理に関する詳細なレポートが出力されます。このレポートからボリュームのマウントディレクトリをメモします。

4. ボリュームがアタッチされてインスタンスにマウントされたら、そのホストで Admin 画面にログインしてフォレストを作成し、インスタンスのホスト名を指定し、さらにフォレストのデータディレクトリとしてボリュームのマウントディレクトリを指定します。フォレストの作成方法の詳細については、『*Administrator's Guide*』の「Creating a Forest」を参照してください。

4.11 MarkLogic クラスタの一時停止または終了

［Update Stack］機能を使用して CloudFormation テンプレートをスタックに再適用し、すべてのノードについて ASG `NodesPerZone` 値を 0 に設定することで、いつでもクラスタを一時停止できます。その後で、各 ASG の `NodesPerZone` を値 1 ～ 20 にリセットすることでノードを再起動できます。

警告 MarkLogic インスタンスは、EC2 ダッシュボードから手動で停止しないでください。各 AutoScaling グループはインスタンスが停止されたことを検出し、それらを自動的に再起動します。これは、Admin 画面からまたは MarkLogic API 呼び出しを使用して MarkLogic をシャットダウンする場合にも当てはまります。

MarkLogic クラスタを削除するには、「CloudFormation スタックの削除」（64 ページ）の説明に従ってスタックを削除します。