
MarkLogic サーバー

MarkLogic Server on Amazon EC2 Guide

MarkLogic 9
2017 年 5 月

最終更新 : 9.0-2、2017 年 7 月

目次

MarkLogic Server on Amazon EC2 Guide

1.0	EC2 上の MarkLogic サーバーの概要	3
1.1	注意：作業を行う前に	3
1.2	EC2 用の MarkLogic サーバーについて	3
1.2.1	Amazon EC2 の用語	4
1.2.2	管理対象クラスタ機能	6
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	9
1.4	一般的なアーキテクチャ	10
2.0	EC2 上の MarkLogic サーバーの基本的な使用方法	13
2.1	セキュリティ	13
2.2	導入手順のまとめ	14
2.3	Amazon EC2 アカウントの作成	14
2.4	EC2 AMI 用の MarkLogic サーバーの有効化	15
2.5	初期セットアップ手順	15
2.5.1	AWS Management Console へのアクセス	16
2.5.2	IAM ロールの作成	17
2.5.3	キーペアの作成	22
2.5.4	SNS (Simple Notification Service) トピックの作成	22
2.6	AWS 設定変数	25
2.7	EC2 ユーザーデータ	29
2.8	/etc/marklogic.conf ファイルを使用した設定	30
2.9	その他の設定方法	31
2.10	セキュリティ設定の考慮事項	31
3.0	CloudFormation を使用した EC2 での MarkLogic の導入	33
3.1	使用する CloudFormation テンプレートのバージョン	33
3.2	概要	34
3.3	導入と起動	36
3.4	AWS Console を使用した CloudFormation スタックの作成	40
3.5	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	49
3.6	CloudFormation テンプレートのサンプル	49
3.6.1	Parameters 宣言	50
3.6.2	Mappings 宣言	54
3.6.3	Resources 宣言	57
3.6.4	Outputs 宣言	65

3.7	安全な資格情報による CloudFormation の使用	65
3.8	CloudFormation スタックの削除	69
4.0	EC2 上の MarkLogic サーバーの管理	70
4.1	MarkLogic サーバーインスタンスへのアクセス	70
4.1.1	ELB を介した MarkLogic サーバーへのアクセス	71
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	72
4.2	EC2 インスタンスへのアクセス	73
4.3	EC2 エラーの検出	75
4.4	mlcmd スクリプトの使用	75
4.4.1	sync-volumes-from-mdb	76
4.4.2	sync-volumes-to-mdb	76
4.4.3	init-volumes-from-system	77
4.4.4	leave-cluster	78
4.5	Amazon S3 (Simple Storage Service) に対応した MarkLogic の設定	78
4.5.1	S3 バケットのセットアップ	79
4.5.2	グループの S3 エンドポイントの設定	79
4.5.3	S3 資格情報の設定	80
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	80
4.5.3.2	環境変数での S3 資格情報の設定	81
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	81
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	82
4.5.5	テストを目的とした MarkLogic へのコンテンツのロード	83
4.6	EC2 でクラスターソースのスケーリング	83
4.7	MarkLogic AMI のアップグレード	84
4.8	モニタリング (CloudWatch)	86
4.9	エンタープライズデータセンターから EC2 への移行	87
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	87
4.10.1	EBS ボリュームの作成	87
4.10.2	EBS ボリュームのインスタンスへのアタッチ	89
4.11	MarkLogic クラスターの一時停止または終了	91

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」(33 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」(6 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働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、「Amazon S3 (Simple Storage Service) に対応した MarkLogic の設定」(78 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」(33 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」(6 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」(40 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にはインスタンスストア上のデータは失われませんが、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、DynamoDB 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/amazondynamodb/latest/developerguide/GettingStartedDynamoDB.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できません。

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なります。

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。

注： AWS で実行している MarkLogic、XCC クライアント（mlcp など）を使用している場合、AWS ELB で動作するように `xcc.httpcompliant` 設定を有効にする必要があります。詳細については、『XCC Developer’s Guide』の「[Using a Load Balancer or Proxy Server with an XCC Application](#)」を参照してください。

- スケールアップとスケールダウン — 処理能力は、大量に必要な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する必要があります。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となります。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 を稼働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」（22 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、そのボリュームを論理デバイス `/dev/sdf` にアタッチします。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。MarkLogic の AMI は事前定義されたデータボリュームを持ちますが、これは `/dev/sdf` と関連付けられている必要があります。

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、稼働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。

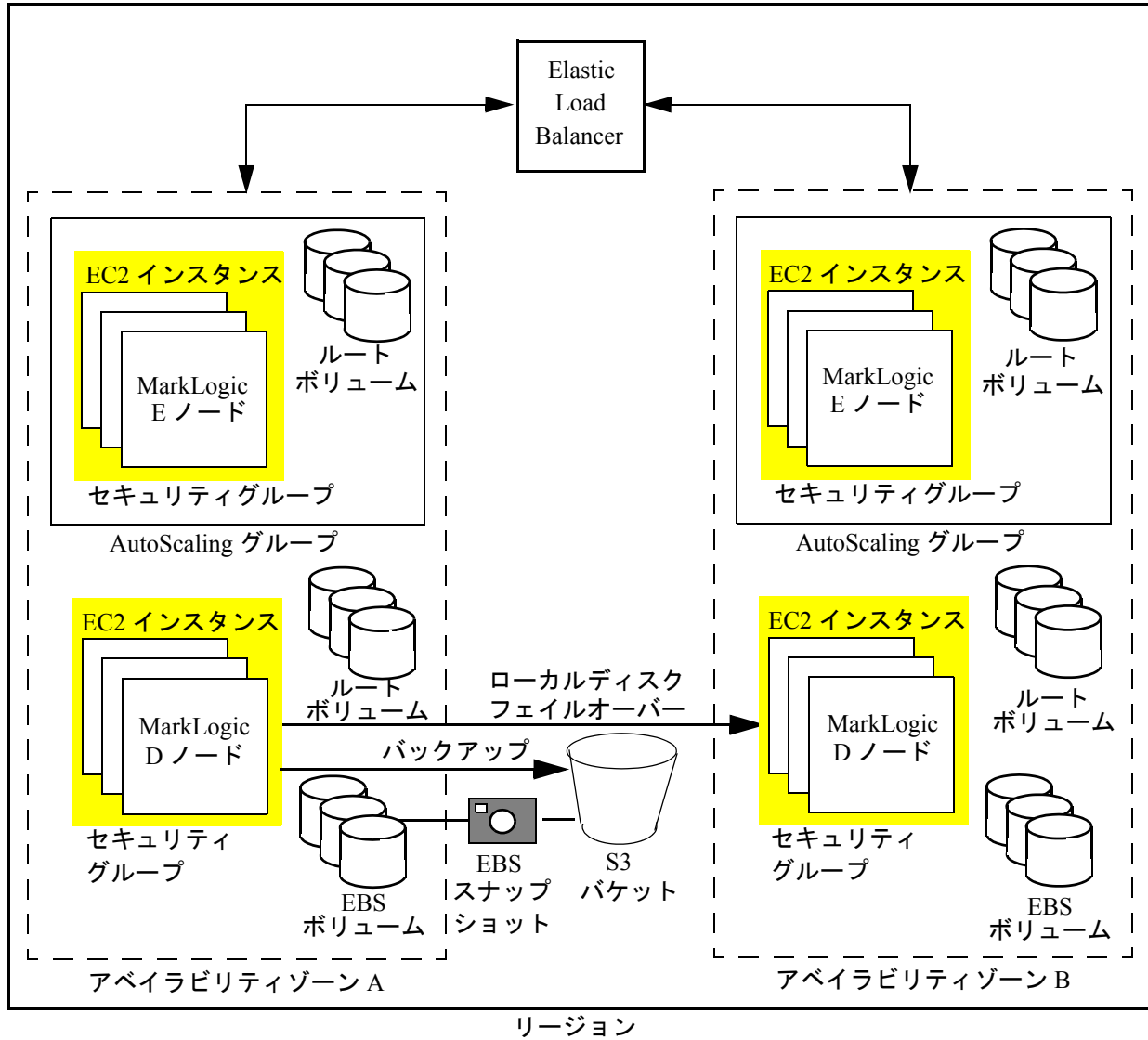
エンドユーザーまたはアプリケーションレベルのクエリは、ロードバランサーを介して E ノードにルーティングする必要があります。より多くのクエリ、ユーザー、計算を処理できるように E ノードを追加してクラスタの処理能力を向上させることができます。

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。

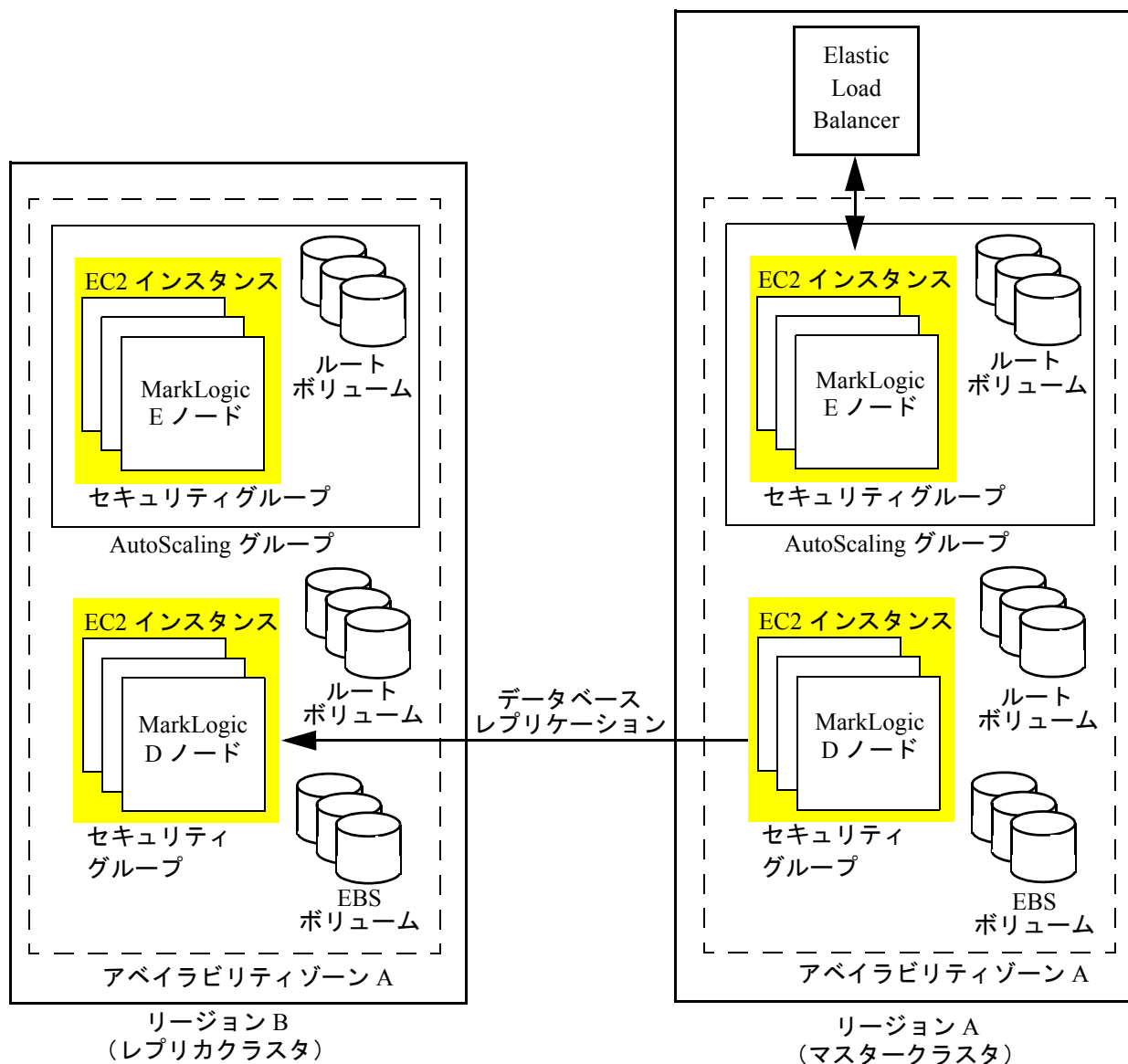
以下の図は、同じリージョン内の異なるゾーンに2つのクラスタが存在するアーキテクチャを示したものです。各クラスタ内のDノードは、ローカルディスクフェイルオーバー用に設定されています。フォレストのデータは IOPS EBS ボリュームに格納され、バックアップと EBS ボリュームのスナップショットは S3 に格納されます。



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 サーバーの管理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](#)
- [EC2 ユーザーデータ](#)
- [/etc/marklogic.conf ファイルを使用した設定](#)
- [その他の設定方法](#)
- [セキュリティ設定の考慮事項](#)

2.1 セキュリティ

MarkLogic サーバーへのアクセスは、『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」(22 ページ) で説明されているキーペア
- 「IAM ロールの作成」(17 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」(33 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ため、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。

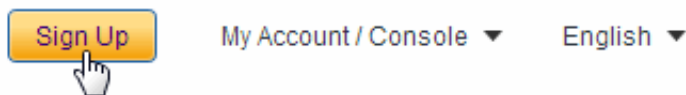
2.2 導入手順のまとめ

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」 (14 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」(15 ページ)
Amazon AWS Management Console を開く。	「AWS Management Console へのアクセス」(16 ページ)
IAM ロールを作成する。	「IAM ロールの作成」(17 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」(22 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」(22 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」(33 ページ)
MarkLogic サーバーの管理画面を開く。	「MarkLogic サーバーインスタンスへのアクセス」(70 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、[Sign Up] 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](#)

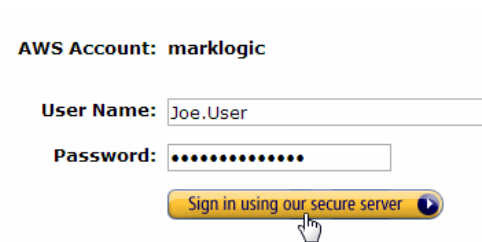
2.5.1 AWS Management Console へのアクセス

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。

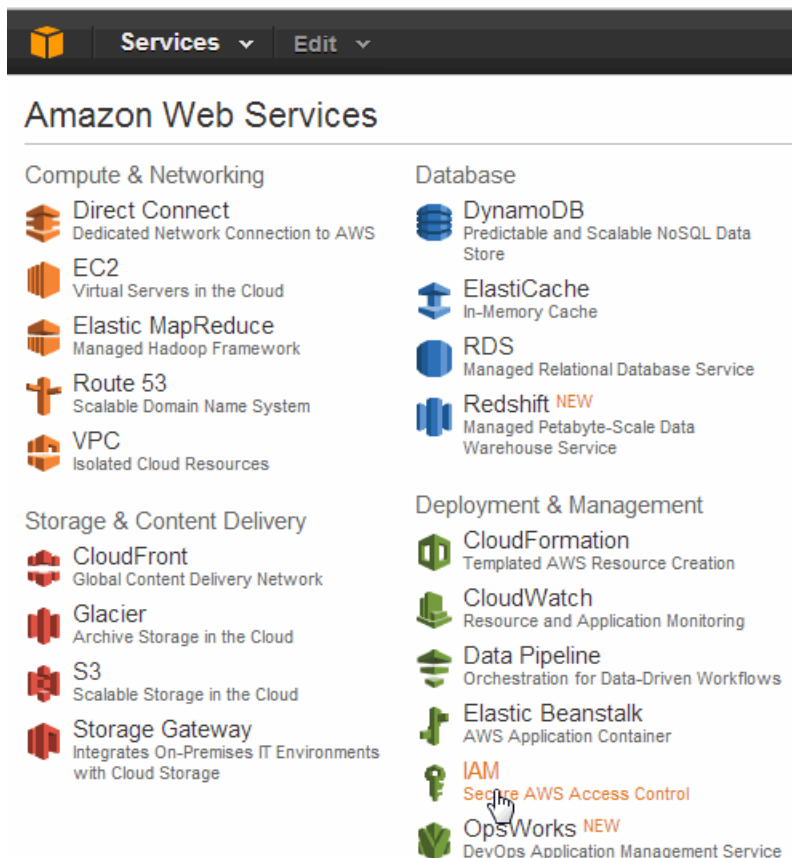


2.5.2 IAM ロールの作成

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 DynamoDB、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。

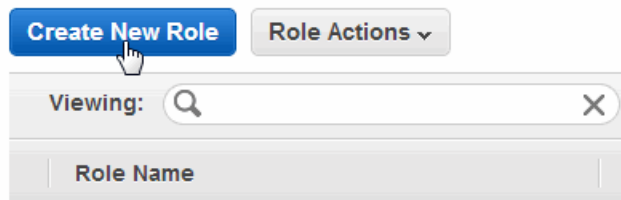
1. Amazon Web Services ページで、[IAM] をクリックします。



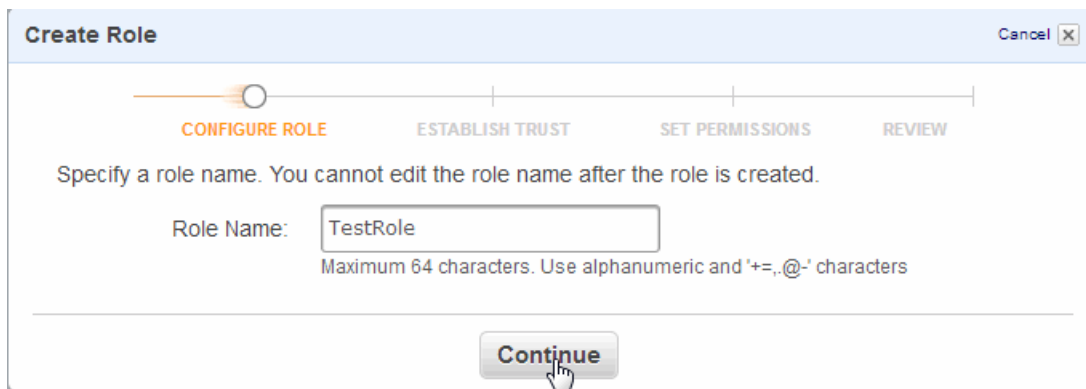
2. [Getting Started] ページの [IAM Resources] セクションで [Roles] をクリックします。



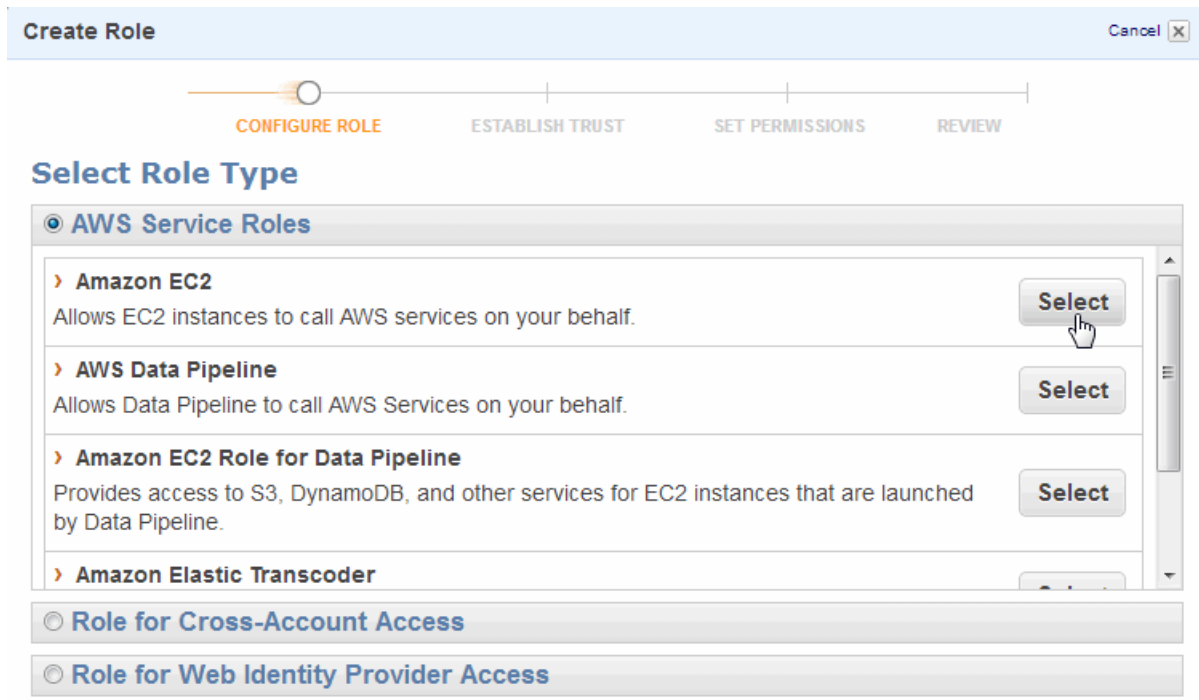
3. [Roles] ページで [Create New Role] をクリックします。



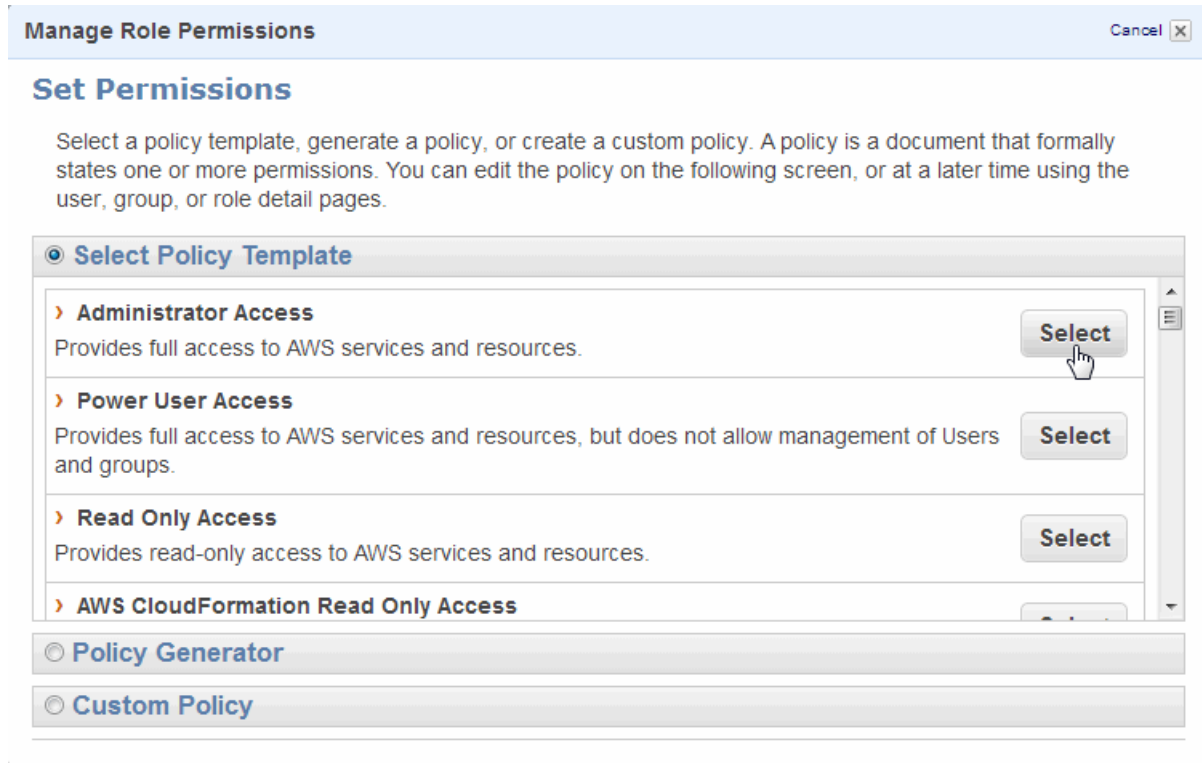
4. [Create Role] ウィンドウで、新しいロールの名前を入力します。



5. [Select Role Type] 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、[Continue] をクリックします。

Manage Role Permissions Cancel

Set Permissions

You can customize permissions by editing the following policy document. For more information about the access policy language, see [Overview of Policies](#) in Using IAM.

Policy Name

AdministratorAccess-TestRole-201310111303

Policy Document

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "*",
      "Resource": "*"
    }
  ]
}
```

< Back Apply Policy

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。

Create Role Cancel

CONFIGURE ROLE ESTABLISH TRUST SET PERMISSIONS REVIEW

Review the following role information. To edit the role, click an edit link, or click **Create Role** to finish.

Role Name TestRole Edit Role Name

Trusted Entities The service ec2.amazonaws.com

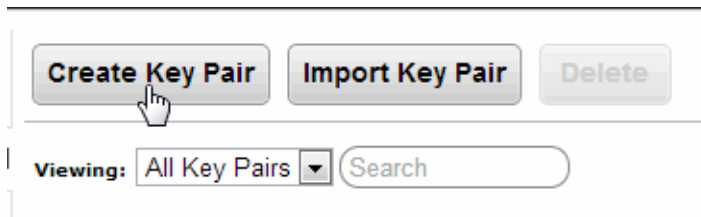
Permissions Administrator Access Edit Permissions

< Back Create Role

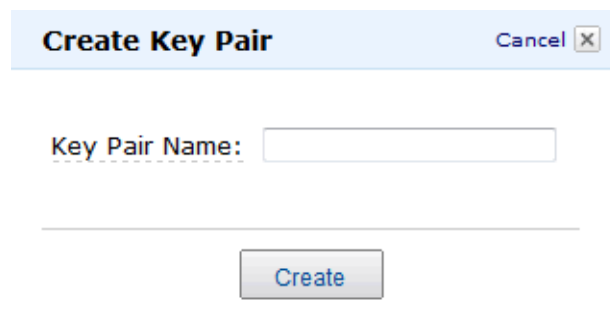
2.5.3 キーペアの作成

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」（73 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。

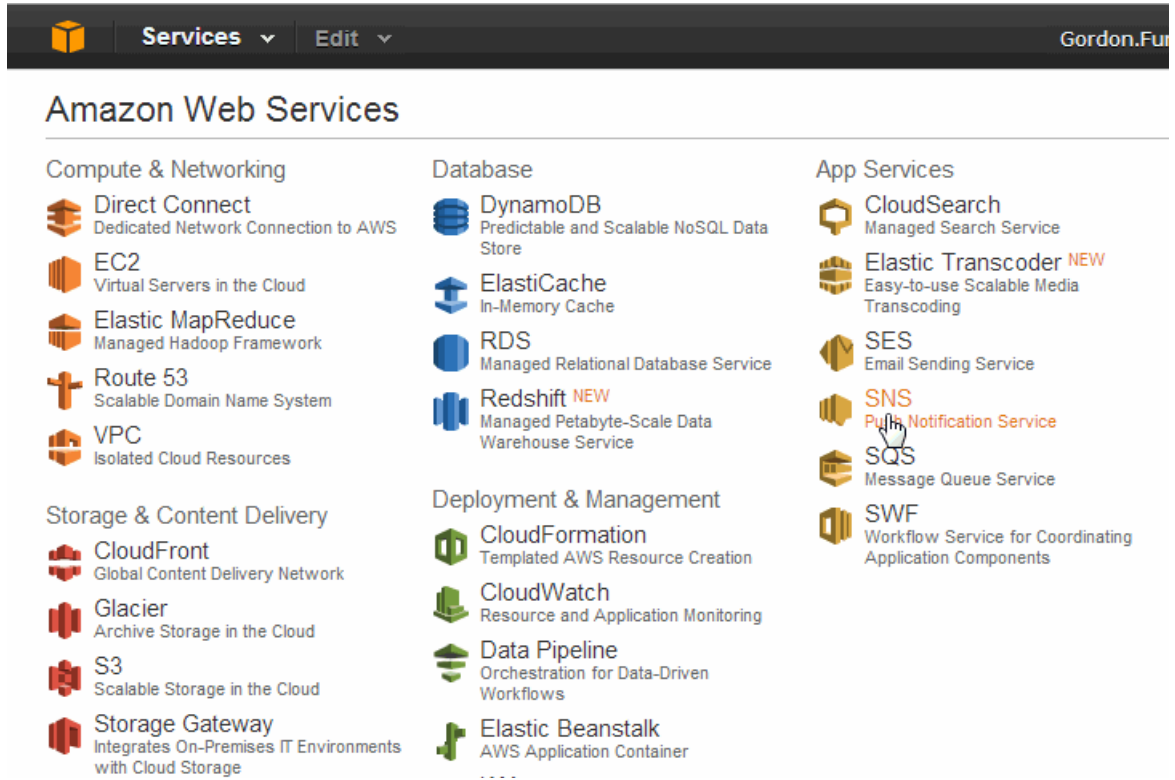
2.5.4 SNS（Simple Notification Service）トピックの作成

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。

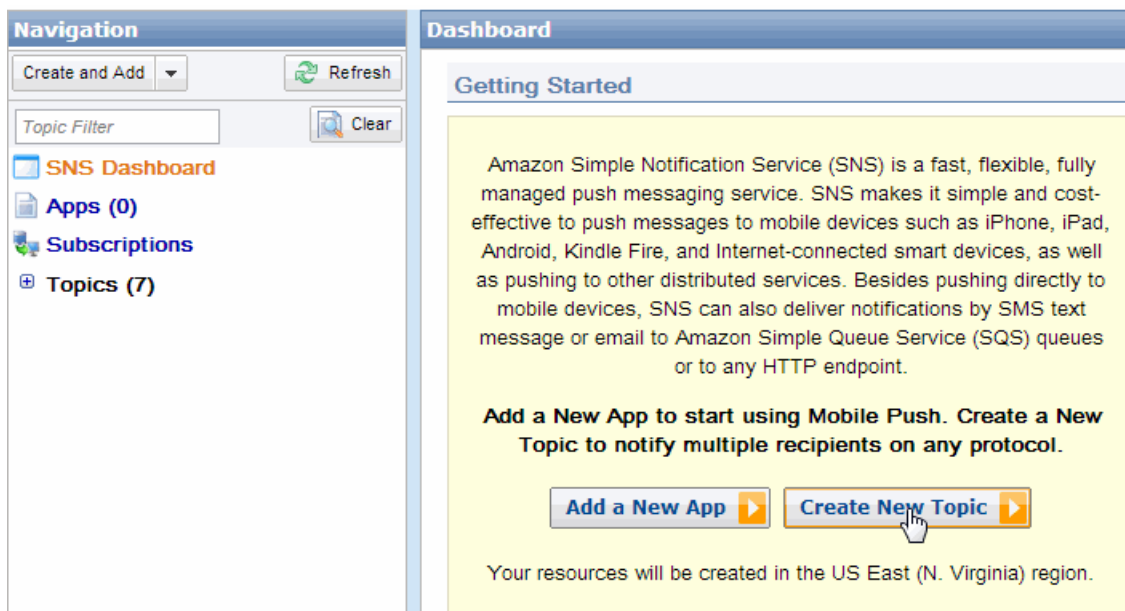
SQS キューシステムおよび SNS トピック作成の詳細については、http://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs_subscribe.html を参照してください。

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] ダッシュボードを開きます。



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。

Create New Topic Cancel X

A topic name will be used to create a permanent unique identifier called an Amazon Resource Name (ARN).

Topic Name *:
Up to 256 alphanumeric characters, hyphens (-) and underscores (_) allowed.

Display Name:
Required for SMS subscriptions (can be up to 10 characters). Optional for other transports.

Cancel Create Topic

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」(40 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。

Topic Details

All Topic Actions Publish Refresh Help

MyMarkLogicTopic

Topic ARN:	arn:aws:sns:us-east-1:027394069461:MyMarkLogicTopic
Topic Owner:	027394069461
Region:	us-east-1
Display Name:	MyMarkLogicTopic

Create Subscription Delete Subscriptions Delivery Policy Subscription Attributes Clear

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 `/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。

CloudFormation を使用する場合、導入および設定には `AWS::CloudFormation::Init` リソース（およびヘルパー `cfn-init` コマンド）を推奨します。詳細については、<http://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-resource-init.html> を参照してください。

CloudFormation を使用しない場合、より詳細な `cloud-init` サービスを直接使用できます。詳細については、<http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/user-data.html> を参照してください。

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が異なっているため、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。クラスタ内のすべてのノードで同じにする必要がある値と、インスタンスごとに異なってもかまわない値があります。サンプルテンプレートおよびリファレンスアーキテクチャでは、インスタンス変数の初期化に ASG (AutoScaling グループ) 起動設定を使用します。ASG はゾーン単位で使用できます。これによりゾーンごとに異なる設定を指定できますが、各ゾーン (または ASG) 内で同じ値を使用できます。

- 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_BOOT_WAIT — 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 (デフォルト = 30)。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。
- MARKLOGIC_LICENSE_KEY — この MarkLogic インスタンスで使用するライセンスキー。このライセンスキーは、BYOL (Bring Your Own License) AMI またはユーザー作成 AMI に対してのみ有効です。

注： 標準機能を有効にするときにライセンスキーは必要ありません。

- MARKLOGIC_LICENSEE — MARKLOGIC_LICENSE_KEY に対応するライセンシー。
- MARKLOGIC_AWS_ACCESS_KEY — Amazon S3 (Simple Storage Service) にアクセスするときに使用される AWS アクセスキー。詳細については、「S3 資格情報の設定」(80 ページ) を参照してください。
- MARKLOGIC_AWS_SECRET_KEY — Amazon S3 (Simple Storage Service) にアクセスするときに使用される AWS シークレットキー。詳細については、「S3 資格情報の設定」(80 ページ) を参照してください。
- MARKLOGIC_AWS_SESSION_TOKEN — Amazon S3 (Simple Storage Service) にアクセスするときに使用されるオプションの AWS セッショントークン。詳細については、「S3 資格情報の設定」(80 ページ) を参照してください。

- `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 (V8.0.3 以前) で使用される「ドメイン」に対応します。DynamoDB の場合、これは DynamoDB テーブル名 (V8.0.4+) に対応します。このクラスタ名は、単一ノードクラスタを含めて、すべての管理対象クラスタ機能に必要です。
- `MARKLOGIC_CLUSTER_MASTER` — 設定する必要がある、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` の項を参照) 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、クラスタマスター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「#」が付いている場合は、可変のノード名として識別されます。詳細については、「導入と起動」(36 ページ) での `/sbin/service` に関する説明を参照してください。
- `MARKLOGIC_ADMIN_USERNAME` — 初期インストールに使用する MarkLogic 管理者ユーザー名。
- `MARKLOGIC_ADMIN_PASSWORD` — 初期インストールに使用する MarkLogic 管理者ユーザーパスワード。

EC2 UserData という場所は、AWS 上の安全な場所ではなく、インスタンスの実行中にクリアすることはできません。EC2 UserData 内に設定された変数は文字列リテラルとして評価されます。これは、`/etc/marklogic.conf` 内の値がシェル「ソース」としてパースされ、常に平文 (または Base64 エンコードのテキスト) であるのと対照的です。

設定変数の推奨される場所は、`/etc/marklogic.conf` です。MarkLogic 資格情報の安全な格納場所を使用した例については、「セキュリティ設定の考慮事項」(31 ページ) を参照してください。

- `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。
arn:aws:sns:us-east-1:1234567890123456:mytopic など、SNS ログトピックのフル ARN を入力します。
- MARKLOGIC_LOG_SQS — MARKLOGIC_LOG_SNS の代替変数。起動メッセージを投稿する AWS SQS キューのエンドポイント。クラスタの起動の進捗を監視するために使用できます。存在しない、空、または「none」に設定されている場合は、使用されません。
- MARKLOGIC_ADMIN_AUTOCREATE — この変数が設定されており、クラスタ管理が設定されていない場合、値は EC2 メタデータキーとして使用され、メタデータ値は自動作成機能の初期パスワードに使用されます。MarketPlace AMI では、これは、デフォルトで「instance-id」になるように事前設定されています。

2.7 EC2 ユーザーデータ

簡単な設定方法は、EC2 UserData 内にすべての変数を配置することです。この方法では、追加のソフトウェアやインフラストラクチャが不要で、AWS Console GUI、コマンドラインツール、AWS SDK、CloudFormation、およびほとんどのサードパーティ製導入ツールを使用して入力できます。ただし、EC2 UserData は安全なデータ格納場所ではないので、機密データ以外のデータにのみ使用してください。

CloudFormation で CloudInit 機能を利用すると、EC2 ユーザーデータに最小の「stub」設定を配置し、テンプレートのリソース MetaData セクションに残りのデータを配置できます。これは非常に安全で柔軟です。

MarkLogic の起動 (`/sbin/service MarkLogic <command>`) では、EC2 UserData はテキスト行として読み取られ、「MARKLOGIC_」で始まる行の場合は `name=value` のペアとして解析されます。それぞれの `name=value` のペアは、`<name>=<value>` として環境にエクスポートされます。たとえば、`MARKLOGIC_CLUSTER_NAME` ユーザーデータ変数は、`MARKLOGIC_CLUSTER_NAME` シェル環境変数になりますが、`MYNAME=MYVALUE` は無視されます。`MARKLOGIC_prefix` は、`PATH` などのシステム環境変数をユーザーが渡せないようにするためのセキュリティ対策です。同様に、コードインジェクションが起きないように、UserData は解析され、評価されているテキストではなく環境変数が明示的に作成されます。

`MARKLOGIC_` 以外で始まる UserData 行はすべて無視されるので、ユーザーは自由に UserData 内の追加 `name=value` ペアを渡すことができ、行の先頭が `MARKLOGIC_` でない限り、他の目的に行全体を使用できます。

2.8 /etc/marklogic.conf ファイルを使用した設定

何らかの理由で、CloudFormation テンプレートを使用して、「AWS 設定変数」(25 ページ) で説明した MarkLogic 設定変数で UserData を設定できない場合、代替方法は `/etc/marklogic.conf` ファイルを作成することです。このファイルは起動時に MarkLogic により読み取られます。このファイルは、AMI 上または RPM 内で明示的に提供されません。これは、AMI または RPM のいずれかのアップグレードでカスタマイズが上書きされないようにするためです。MarkLogic の最初の起動前に、このファイルを作成および入力した場合、そのファイルが読み込まれます (`/etc/sysconfig/MarkLogic` を呼び出すシェルによって評価されます)。`/etc/marklogic.conf` を読み込んだ結果として設定されたサポート対象の設定環境変数はすべて、「導入と起動」(36 ページ) で説明されている順序および優先順位で、エクスポートされ評価されます。

「AWS 設定変数」(25 ページ) で説明しているように、`MARKLOGIC_EC2_HOST=0` を `/etc/marklogic.conf` ファイルに追加することによって、起動および管理機能が無効になります。

注： 安全な資格情報を提供するための推奨される方法については、「セキュリティ設定の考慮事項」(31 ページ) を参照してください。

`/etc/marklogic.conf` ファイルは、カスタム AMI の作成、EC2 UserData の使用を困難にしている導入ツールとの統合、および手動のカスタマイズに役立ちます。ファイルは、MarkLogic RPM のインストール前に作成でき、RPM をアンインストールしても削除されません。

2.9 その他の設定方法

グローバルプロファイル (/etc/profile) の修正、ルート起動スクリプト、/etc/sysconfig/MarkLogic の編集などのその他の設定方法は可能ですが、推奨されません。これらのファイルの変更内容が、OS または MarkLogic の更新後も失われないことや、これらのファイルを変更していなくても、後で同じように機能することは保証されません。OS をアップグレードすると、たびたび、ルートまたは初期環境の設定が変更され、また起動中に有効な一連のエクスポート済み変数が変更されます。/sbin/service MarkLogic <command> を直接呼び出すスクリプトでは、初期環境と同じ環境が必要です。

2.10 セキュリティ設定の考慮事項

初期管理者ユーザーの自動作成に資格情報を与えるために、「導入と起動」(36 ページ) で説明されている起動プロセス中に、MARKLOGIC_ADMIN_USERNAME 変数と MARKLOGIC_ADMIN_PASSWORD 変数を設定する必要があります。これは、最初のインストール時、およびノードの終了と再起動でクラスタを再参加させるために必要です。パスワードは、最初の起動プロセスでのみ使用され、MarkLogic プロセスにエクスポートされたりディスクに保存されたりすることはありません。

既知のパスワードをシステムに安全に提供するために、平文のパスワードを /etc/marklogic.conf に格納して EC2 UserData に渡すことはやめてください。AWS が推奨する 1 つの単純な方法は、クラスタ内の EC2 インスタンスへの読み取り専用アクセスを認める AMI ロールと組み合わせて、暗号化したストレージおよびデータ転送でプライベート S3 バケットを利用することです。AWS CLI を使用すると、パスワードを安全に取得し、オンデマンドで MarkLogic に渡すことができます。このコマンドは MARKLOGIC_ADMIN_PASSWORD 変数として /etc/marklogic.conf に配置する必要があります。

詳細については、AWS CLI (<http://docs.aws.amazon.com/cli/latest/reference/s3/index.html>) を参照してください。

次に、S3 から安全に資格情報を取得する完全な /etc/marklogic.conf ファイルの例を示します。

```
MARKLOGIC_CLUSTER_NAME=JOE-CFN-JOESecure5x-MarkLogicDDBTable-164OK8LD6ARMY
MARKLOGIC_EBS_VOLUME=vol-11111111
MARKLOGIC_NODE_NAME=NodeA#
MARKLOGIC_ADMIN_USERNAME=admin
##
MARKLOGIC_ADMIN_PASSWORD=\
$(aws s3 --region us-east-1 cp s3://marklogic.joesbucket/secret-password -)
##
MARKLOGIC_CLUSTER_MASTER=1
MARKLOGIC_LICENSEE=none
MARKLOGIC_LICENSE_KEY=none
MARKLOGIC_LOG_SNS=arn:aws:sns:us-east-1:02344343341:JOE-LOG-NOTIFY
```

注： スペースを含む変数は引用符で囲む必要があります。次に例を示します。
MARKLOGIC_LICENSEE="Carp Corporation".

複数のゾーンクラスタの場合、EC2 インスタンスは、ASG ごとに単一の LaunchConfiguration を使用する AutoScalingGroup で作成されるので、環境は、そのゾーン内に作成されたすべての EC2 インスタンスで同一です。この設定変数は、各ゾーン内のノードで同一の設定値にできる設計です。同じ概念により、ゾーンごとにノードの数を変更できます。上記の例の設定は、単一ゾーン内のすべてのノードに使用できません。追加ゾーンごとに、次の 3 つの値は変える必要がありますが、残りは同一にする必要があります。

```
# ...Same as Zone except for ...
MARKLOGIC_EBS_VOLUME=vol-22222222
MARKLOGIC_NODE_NAME=NodeB#
MARKLOGIC_CLUSTER_MASTER=0
#....
```

安全なキーマネージャーに接続してディスク上に格納された暗号化パスワードを復号するなど、同様のメカニズムを使用できます。

/etc/marklogic.conf ファイルは、ホストに対し MarkLogic を初めて起動する前に作成する必要があります。外部でユーザー名とパスワードが変更された場合、/etc/marklogic.conf で取得したパスワードは現在のパスワードを返す必要があります。そうでない場合、ノードは再起動時にクラスタに再参加できません。

CloudFormation を使用した /etc/marklogic.conf の作成例については、「安全な資格情報による CloudFormation の使用」（65 ページ）を参照してください。

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。

- [使用する CloudFormation テンプレートのバージョン](#)
- [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](#)
- [安全な資格情報による CloudFormation の使用](#)
- [CloudFormation スタックの削除](#)

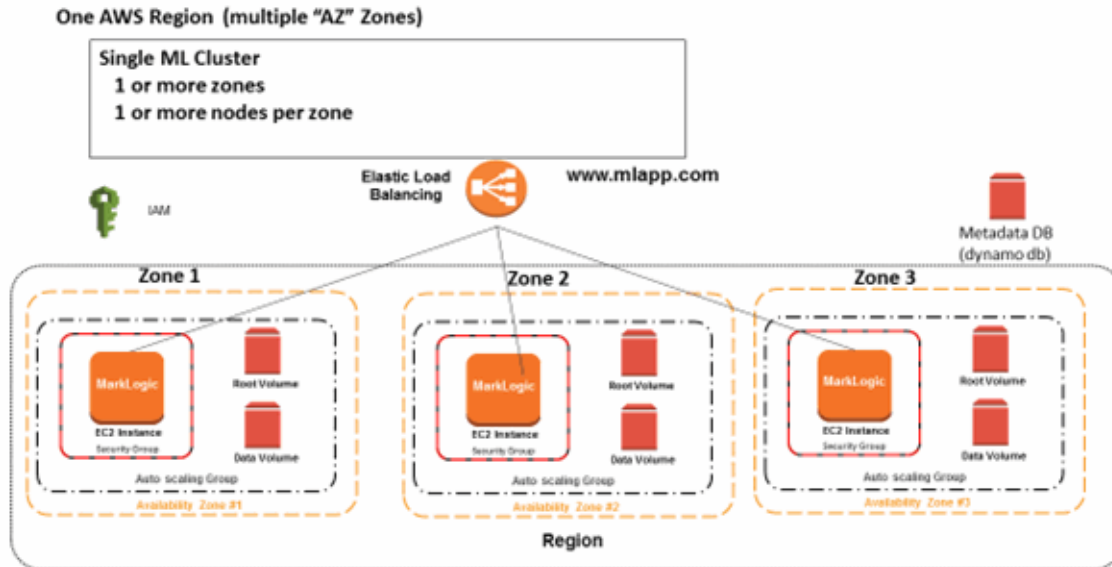
3.1 使用する CloudFormation テンプレートのバージョン

MarkLogic CloudFormation テンプレートには、バージョン 1 とバージョン 2 の 2 つの基本バージョンがあります。2 つのテンプレートバージョンの根本的な違いは、メタデータの保存に使用するデータベースにあります。

- バージョン 1.0 — このバージョンはメタデータを SimpleDB に保存します。バージョン 1 は、新規インストールの場合、MarkLogic 7.x ~ 8.0.1 と互換性があります。既存の EC2 クラスタを更新する場合は、「MarkLogic AMI のアップグレード」(84 ページ) の説明に従い、このバージョンを使用してください。
- バージョン 2.0 — このバージョンはメタデータを DynamoDB に保存します。バージョン 2 は MarkLogic 8.0.3 以降と互換性がありますが、それ以前のリリースとは互換性がありません。

3.2 概要

管理対象クラスタは、自動的に初期化され、以下の図に示すような推奨されるトポロジを使用して事前設定されます。



サンプルの CloudFormation テンプレートでは、このリファレンスアーキテクチャの単純な例を実装し、管理対象クラスタ機能を利用します。どのようにクラスタが作成されているかにかかわらず、必要なコンポーネントを、制御された方法で作成、設定、および導入する必要があります。

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）および EBS タイプ
- 作成されるインスタンスの EC2 インスタンスタイプ
- オプション：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

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 の `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

3.3 導入と起動

MarkLogic は、システムサービスとして `/sbin/service` 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、ルール (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。

初期プロセス中、特に 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、MarkLogic と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
58	ntpd
80	sendmail
85	MarkLogic (バージョン 7)
86	tomcat-jsvc
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. `/sbin/service 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. `MARKLOGIC_CLUSTER_NAME`、`MARKLOGIC_NODENAME`、および `MARKLOGIC_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。

11. 手順 10 が実行されると、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。
12. 手順 10 が実行されない場合、次の処理が行われます。
 - `MARKLOGIC_ADMIN_AUTOCREATE` が設定され、空ではない場合、
 - `MARKLOGIC_ADMIN_PASSWORD` は、キーが `$MARKLOGIC_ADMIN_AUTOCREATE` である EC2 メタデータの値に設定されます。これは、`MARKLOGIC_ADMIN_PASSWORD` の以前の設定をすべて上書きします。
 - `MARKLOGIC_ADMIN_PASSWORD` が空でなく、`MARKLOGIC_ADMIN_USERNAME` が空の場合、`MARKLOGIC_ADMIN_USERNAME="admin"` に設定します。
 - `MARKLOGIC_ADMIN_PASSWORD` と `MARKLOGIC_ADMIN_USERNAME` の両方が空の場合は、
 - `MARKLOGIC_EBS_VOLUME` 設定に指定されているすべてのボリュームを初期化してマウントします。
 - MarkLogic サーバーを起動します。
 - 初期管理者ユーザーを作成し、セキュリティデータベースを初期化します。
 - サーバーが再起動するまで待ち、ログインを検証します(管理対象クラスタの起動での現在の実装に従いタイムアウトした場合は再試行します)。
 - システムログおよびコンソールに成功または失敗のログが記録されます。

3.4 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、

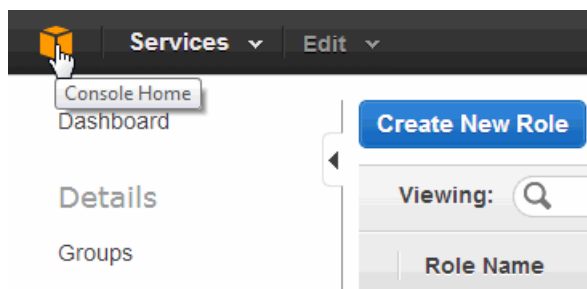
<http://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/cfn-console-create-stack.html> を参照してください。

CloudFormation スタックを作成するには、次のものがが必要です。

- Amazon Marketplace で AMI を購入するか、自身の AMI を作成します（お困りの場合はカスタマサポートにお問い合わせください）。サンプルの CloudFormation テンプレートには、最新の Marketplace AMI が組み込まれており、適切な AMI ID を使用してこれらを編集する必要があります。
- 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」（49 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」（17 ページ）で説明されています。
- キーペア。「キーペアの作成」（22 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」（22 ページ）で説明されています。
- オプション：SNS トピック。「SNS（Simple Notification Service）トピックの作成」（22 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。

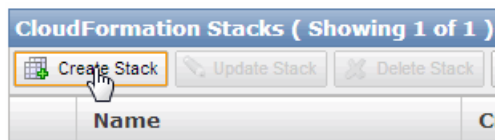
1. [Services] ページの左上部分にあるボックスをクリックし、アマゾンウェブサービスのホームページにアクセスします。



- 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。



- [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」(49 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。

Create Stack Cancel X

SELECT TEMPLATE SPECIFY PARAMETERS ADD TAGS REVIEW

AWS CloudFormation gives you an easier way to create a collection of related AWS resources (a stack) by describing your requirements in a template. To create a stack, fill in the name for your stack and select a template. You may choose one of the sample templates to get started quickly, or one of your own templates stored in S3 or on your local hard drive.

Stack Name:
Test Stack

Template:

Use a sample template

Upload a Template File

Choose File No file chosen
No file chosen

Provide a Template URL

Show Advanced Options

Continue ▶

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。

Create Stack Cancel

SELECT TEMPLATE **SPECIFY PARAMETERS** ADD TAGS REVIEW

Stack Description: Create a cluster with three node or more nodes, multi az, load balanced, MarkLogic Cluster.

Specify Parameters
Below are the parameters associated with your CloudFormation template. You may review and proceed with the default parameters or make customizations as needed below.

LicenseKey
The MarkLogic License Key

AdminUser
The MarkLogic Administrator Username

Zone1
The AZ Zone 1

AdminPass
The MarkLogic Administrator Password

Zone3
The AZ Zone 3

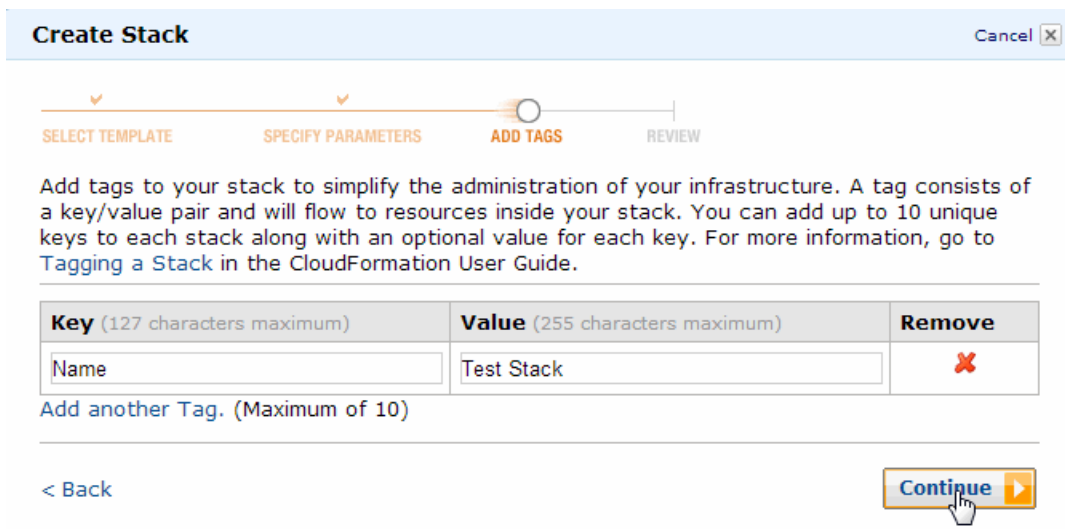
Zone2
.

[< Back](#) **Continue**

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」(17 ページ) で作成した IAM ロールの名前。

パラメータ名	説明
InstanceType	<p>起動する EC2 インスタンスのタイプ。これらは、リリース、製品タイプ、ゾーン、リージョン、可用性によって異なります。これらのフィールドで現在サポートされている値については、http://developer.marklogic.com/products/aws を参照してください。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。</p> <p>注： 現在、HVM インスタンスタイプだけが Marketplace AMI でサポートされており、PVM タイプはカスタム AMI で使用できます。</p>
KeyName	「キーペアの作成」(22 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)。トピックの ARN が SNS ダッシュボードに表示されたらそれをすべて入力します (例 : <code>arn:aws:sns:us-east-1:1234567890123456:mytopic</code>)。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」(22 ページ) を参照してください。
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、 <code>us-east</code> や <code>us-west</code> などの同じリージョンに含まれている必要があります。


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。



Create Stack Cancel

SELECT TEMPLATE SPECIFY PARAMETERS **ADD TAGS** REVIEW

Add tags to your stack to simplify the administration of your infrastructure. A tag consists of a key/value pair and will flow to resources inside your stack. You can add up to 10 unique keys to each stack along with an optional value for each key. For more information, go to [Tagging a Stack in the CloudFormation User Guide](#).

Key (127 characters maximum)	Value (255 characters maximum)	Remove
Name	Test Stack	

Add another Tag. (Maximum of 10)

< Back Continue

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。

Create Stack Cancel

SELECT TEMPLATE SPECIFY PARAMETERS ADD TAGS **REVIEW**

Please review the information below, then click Continue to create the stack.

Stack Information Edit Stack

Stack Name:	TestStack
Stack Description:	Create a three node, multi az, load balanced, MarkLogic Cluster.
Template:	https://cf-templates-g11hbnsw4v0-us-east-1.s3.amazonaws.com/2013170DVD-QA-20130618-ThreeNodeCluster.template
IAM Acknowledgement:	false
Estimated Cost:	Cost

Parameters Edit Parameters

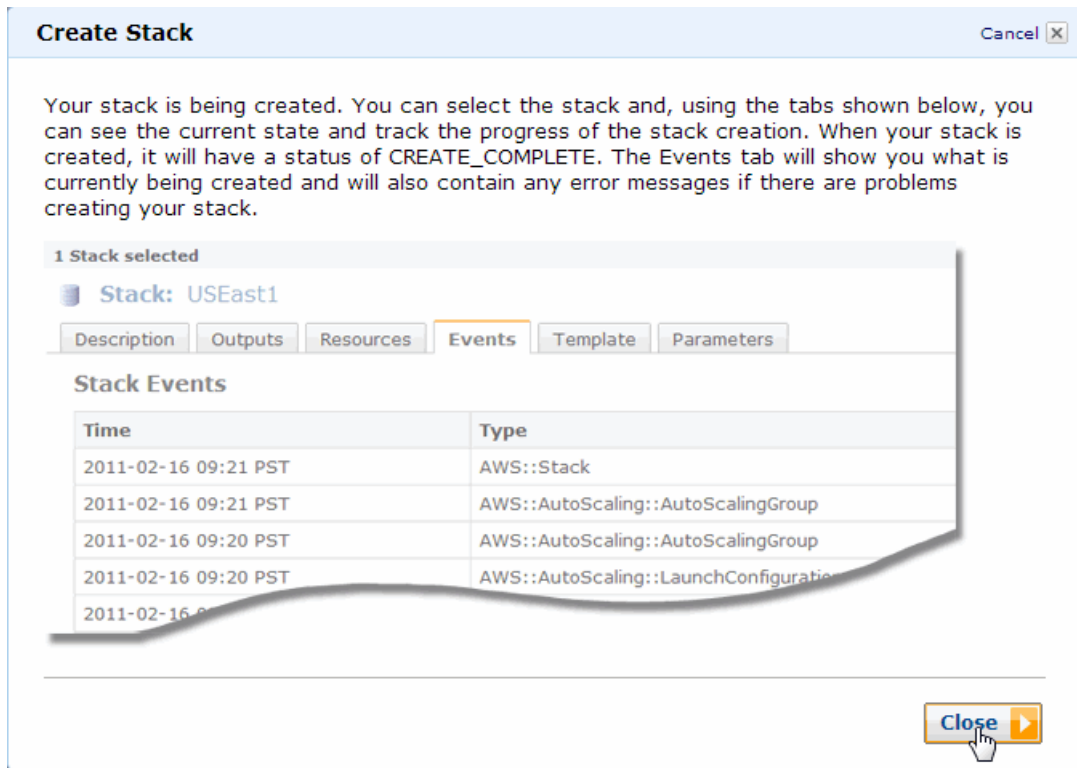
AdminUser	admin
AdminPass	*****
IAMRole	TestRole
Zone1	us-east-1c
Zone3	us-east-1d
VolumeSize	10

Notification Edit Notification

Notification:	none
Creation Timeout (minutes):	none
Rollback on Failure:	true

[< Back](#) Continue

- 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。



- 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。

CloudFormation Stacks (Showing 2 of 2)

Create Stack Update Stack Delete Stack Viewing: Active

	Name	Created	Status
<input checked="" type="checkbox"/>	TestStack	2013-06-19 14:47:36 UTC-7	● CREATE_IN_PROGRESS

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。

The screenshot shows the 'Stack: TestStack' page with the 'Events' tab selected. The 'Stack Events' table is displayed below the tabs.

ID	Physical ID	Status
oadBalancer	TestStack-ElasticL-16EJPFQAYKMYW	● CREATE_IN_PROGRESS
icSDBDomain	TestStack-MarkLogicSDBDomain-1TZ59HQBO8KLJ	● CREATE_COMPLETE
icSDBDomain	TestStack-MarkLogicSDBDomain-1TZ59HQBO8KLJ	● CREATE_IN_PROGRESS
cyolume3	vol-7546e12f	● CREATE_COMPLETE

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 の管理画面が開きます。

The screenshot shows the 'Stack: TestStack' page with the 'Outputs' tab selected. The 'Stack Outputs' table is displayed below the tabs.

Key	Value	Description
URL	http://TestStack-ElasticL-KXUZ8SDSXNEL-293088125.us-east-1.elb.amazonaws.com:8001	The URL of the MarkLogic Cluster

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、管理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」(22 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。

3.5 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://aws.amazon.com/cli/> 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」(40 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 の管理画面の URL を取得できます。

3.6 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。

<http://developer.marklogic.com/products/aws> で入手できるサンプルのテンプレートは、管理対象クラスタ機能に必要なアーキテクチャおよび IT の要件を示し、例としてのみそのまま使用できるように用意されています。実稼働テンプレートは、特有の IT 要件に対応するようにカスタマイズする必要があり、これらの例でパラメータおよびマッピングとして示された値の多くはハードコード化できます。例えば、あるリージョン内だけで実行する場合は、リージョンと AMI ID のマッピング表は必要ありません。

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」(40 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](#)

注： これらのサンプルテンプレートは ELB を作成するとともに、MarkLogic サーバーごとにパブリック IP を有効にします。スタックの出力には、ELB の URL が示されます。アプリケーションは一般に、そのエンドポイントとして ELB を使用する必要があります。mlcp などの XCC アプリケーションは、セッションアフィニティの問題の有無にかかわらず ELB を通じて接続するために、`xcc.httpcompliant=true` モードを設定する必要があります。詳細については、『XCC Developer's Guide』の「[Using a Load Balancer or Proxy Server with an XCC Application](#)」を参照してください。

3.6.1 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」(40 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-8.0-20150512.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": "r3.8xlarge",
      "AllowedValues":
      [ "t2.small", "t2.medium", "m1.medium", "m3.medium", "m3.large",
      "m3.xlarge", "m3.2xlarge", "cc1.4xlarge", "cc2.8xlarge", "c3.large",
      "c3.xlarge", "c3.2xlarge", "c3.4xlarge", "c3.8xlarge", "cr1.8xlarge",
      "r3.large", "r3.xlarge", "r3.2xlarge", "r3.4xlarge", "r3.8xlarge",
      "i2.xlarge", "i2.2xlarge", "i2.4xlarge", "i2.8xlarge", "hi1.4xlarge",
      "hs1.8xlarge" ]
    },

    "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"
    },
    "VolumeType" : {
      "Description" : "The EBS Data volume Type",
      "Type" : "String",
      "AllowedValues" : [ "standard", "gp2" ],
      "Default" : "gp2"
    },
    "Zone1": {
      "Description": "The AZ Zone 1 (e.g. us-west-2a)",
      "Type": "String",
      "AllowedValues":
        [ "ap-northeast-1a", "ap-northeast-1b", "ap-northeast-1c",
"ap-southeast-1a", "ap-southeast-1b", "ap-southeast-2a",
"ap-southeast-2b", "eu-west-1a", "eu-west-1b", "eu-west-1c",
"sa-east-1a", "sa-east-1b", "us-east-1a", "us-east-1b", "us-east-1c",
"us-east-1d", "us-east-1e", "us-west-1a", "us-west-1b", "us-west-1c",
"us-west-2a", "us-west-2b", "us-west-2c" ]
    },
    "Zone2": {
      "Description": "The AZ Zone 2",
      "Type": "String",
      "AllowedValues":
        [ "ap-northeast-1a", "ap-northeast-1b", "ap-northeast-1c",
"ap-southeast-1a", "ap-southeast-1b", "ap-southeast-2a",
"ap-southeast-2b", "eu-west-1a", "eu-west-1b", "eu-west-1c",

```

```
"sa-east-1a", "sa-east-1b", "us-east-1a", "us-east-1b", "us-east-1c",
"us-east-1d", "us-east-1e", "us-west-1a", "us-west-1b", "us-west-1c",
"us-west-2a", "us-west-2b", "us-west-2c" ]
    },

    "Zone3": {
        "Description": "The AZ Zone 3",
        "Type": "String",
        "AllowedValues":
            [ "ap-northeast-1a", "ap-northeast-1b", "ap-northeast-1c",
"ap-southeast-1a", "ap-southeast-1b", "ap-southeast-2a",
"ap-southeast-2b", "eu-west-1a", "eu-west-1b", "eu-west-1c",
"sa-east-1a", "sa-east-1b", "us-east-1a", "us-east-1b", "us-east-1c",
"us-east-1d", "us-east-1e", "us-west-1a", "us-west-1b", "us-west-1c",
"us-west-2a", "us-west-2b", "us-west-2c" ]
    }
},

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

3.6.2 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": {
    "c1.medium" : {
      "Arch" : "PVM"
    },
    "c1.xlarge" : {
      "Arch" : "PVM"
    },
    "c3.2xlarge" : {
      "Arch" : "HVM"
    },
    "c3.4xlarge" : {
      "Arch" : "HVM"
    },
    "c3.8xlarge" : {
      "Arch" : "HVM"
    },
    "c3.large" : {
      "Arch" : "HVM"
    },
    "c3.xlarge" : {
      "Arch" : "HVM"
    },
    "cc2.8xlarge" : {
      "Arch" : "HVM"
    },
    "cr1.8xlarge" : {
      "Arch" : "HVM"
    },
    "hi1.4xlarge" : {
      "Arch" : "HVM"
    },
    "hs1.8xlarge" : {
      "Arch" : "HVM"
    },
    "i2.2xlarge" : {
```

```
    "Arch" : "HVM"
  },
  "i2.4xlarge" : {
    "Arch" : "HVM"
  },
  "i2.8xlarge" : {
    "Arch" : "HVM"
  },
  "i2.xlarge" : {
    "Arch" : "HVM"
  },
  "m1.large" : {
    "Arch" : "PVM"
  },
  "m1.medium" : {
    "Arch" : "PVM"
  },
  "m1.small" : {
    "Arch" : "PVM"
  },
  "m1.xlarge" : {
    "Arch" : "PVM"
  },
  "m2.2xlarge" : {
    "Arch" : "PVM"
  },
  "m2.4xlarge" : {
    "Arch" : "PVM"
  },
  "m2.xlarge" : {
    "Arch" : "PVM"
  },
  "m3.2xlarge" : {
    "Arch" : "HVM"
  },
  "m3.large" : {
    "Arch" : "HVM"
  },
  "m3.medium" : {
    "Arch" : "HVM"
  },
  "m3.xlarge" : {
    "Arch" : "HVM"
  },
  "r3.2xlarge" : {
    "Arch" : "HVM"
  },
  "r3.4xlarge" : {
    "Arch" : "HVM"
  },
  "r3.8xlarge" : {
    "Arch" : "HVM"
  },
},
```



```
"r3.large" : {
  "Arch" : "HVM"
},
"r3.xlarge" : {
  "Arch" : "HVM"
},
"t2.medium" : {
  "Arch" : "HVM"
},
"t2.small" : {
  "Arch" : "HVM"
},
"cc1.4xlarge" : {
  "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" : {
    "HVM" : "ami-a5ee08ce"
  },
  "us-west-1" : {
    "HVM" : "ami-d313fb97"
  },
  "us-west-2" : {
    "HVM" : "ami-ed81bddd"
  },
  "eu-west-1" : {
    "HVM" : "ami-e9b9c99e"
  },
  "ap-southeast-1" : {
    "HVM" : "ami-181e264a"
  },
  "ap-southeast-2" : {
    "HVM" : "ami-a993ea93"
  },
  "ap-northeast-1" : {
    "HVM" : "ami-ae61b0ae"
  },
  "sa-east-1" : {
    "HVM" : "ami-35b13028"
  }
},
```

3.6.3 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
- DynamoDB テーブル (DynamoDB 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" : {
      "Ref" : "VolumeType"
    }
  }
},
},

```

MarkLogicDDBTable では、「Amazon EC2 の用語」(4 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 DynamoDB データベースを作成し、DynamoDB テーブルの名前を返します。

注： 読み取りと書き込みの容量はどちらも、3 ノードテンプレートでは 10、1 ノードテンプレートでは 2 に設定されます。大規模クラスタ内のインスタンスが同時に再起動した場合に起きるピーク期間に十分な容量を、確実にプロビジョニングしていることが重要です。十分な容量がない場合、終了後にノードが置き換えられたときに、クラスタは正しく再結合できません。容量に関する CloudWatch アラームを設定できます。これを使用すれば、手動でアラートすることも、容量を修正するスクリプトをトリガーすることもできます。

AWS::DynamoDB::Table タイプの詳細については、<http://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-resource-dynamodb-table.html> を参照してください。

```

"MarkLogicDDBTable" : {
  "Type" : "AWS::DynamoDB::Table",
  "Properties" : {
    "AttributeDefinitions" : [ {
      "AttributeName" : "node",
      "AttributeType" : "S"
    } ],
    "KeySchema" : [ {
      "AttributeName" : "node"
      "KeyType" : "HASH",
    } ],
    "ProvisionedThroughput" : {
      "WriteCapacityUnits" : "10",
      "ReadCapacityUnits" : "10"
    }
  }
},
},

```

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": {
      "Fn::If" : [ "UseLogSNS", {
        "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 CloudFormation 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になります。

```
  "NotificationTypes": ["autoscaling:EC2_INSTANCE_LAUNCH",
"autoscaling:EC2_INSTANCE_LAUNCH_ERROR",
"autoscaling:EC2_INSTANCE_TERMINATE",
"autoscaling:EC2_INSTANCE_TERMINATE_ERROR"]
  }, {
    "Ref" : "AWS::NoValue"
  } ]
}
}
},
```

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」（25 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 - MaxxNodesPerZone 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" : "MarkLogicDDBTable"
    }, "\n", "MARKLOGIC_EBS_VOLUME=", {
      "Ref" : "MarklogicVolume1"
    }, ":", {
      "Ref" : "VolumeSize"
    }, "::", {
      "Ref" : "VolumeType"
    }, ":", "*\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"
    }
  }
}
},

```

3.6.4 Outputs 宣言

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

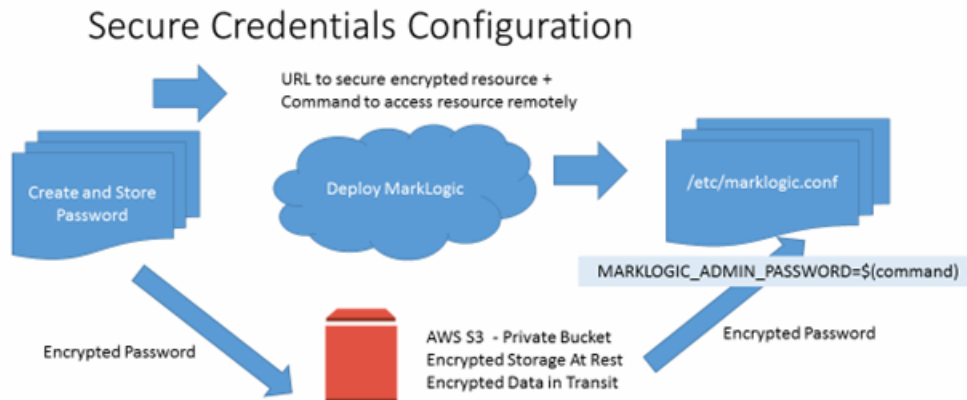
```
"Outputs": {
  "URL": {
    "Description": "The URL of the MarkLogic Cluster",
    "Value": {
      "Fn::Join": [ "", [ "http://", {
        "Fn::GetAtt": ["ElasticLoadBalancer", "DNSName"]
      }, ":8001" ] ]
    }
  }
}
```

3.7 安全な資格情報による CloudFormation の使用

このサンプルテンプレートは、実稼働環境向けには設計されていません。ほとんどの導入には、対処の必要な特定のインフラストラクチャおよび統合の要件があります。重要な問題は、自動化された「無人」プロセスで MarkLogic の安全な資格情報をどのように管理するかということです。このサンプルテンプレートでは、管理者パスワードが平文で CloudFormation パラメータとして渡され、続いて名前 / 値のペアである単純な EC2 ユーザーデータに変換されます。これは、資格情報を扱う場合に安全な方法ではありません。

「[/etc/marklogic.conf ファイルを使用した設定](#)」(30 ページ) で説明しているように、EC2 UserData に代わる方法として、導入中に `/etc/marklogic.conf` を作成できます。この方法は、CloudFormation 内で非常に簡単に行うことができます。

CloudFormation を使用した実稼働導入では、`AWS::CloudFormation::Init` リソース (およびヘルパー `cfn-init` コマンド) を導入および設定に推奨します。詳細については、<http://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-resource-init.html> を参照してください。



CloudFormation を使用しない場合は、CloudFormation が使用する詳細設定可能な API である `cloud-init` サービスを直接使用できます。詳細については、<http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/user-data.html> を参照してください。

CloudInit リソースを使用した場合、EC2 UserData は、`cfn-init` を介して安全にテンプレートメタデータリソースから設定変数にアクセスする小さな「ブートストラップ」スクリプトにのみ使用されます。資格情報自体の代わりに、資格情報の安全なチャンネルに対するリファレンスを渡すことによって、機密データが元の場所から EC2 インスタンスに直接渡されることはありません。このプロセスは AWS が推奨するものです。次の投稿を参照してください。

<http://blogs.aws.amazon.com/application-management/post/Tx3LKFZ27CWZBKO/Authenticated-File-Downloads-with-CloudFormation>

必要な認証を設定し、ストレージおよびアクセスを保護するためのオプションが多数存在します。適切な設定はユーザーの要件と統合方針によって異なり、IT およびセキュリティ計画全体に組み入れる必要があります。MarkLogic 導入と CloudFormation または他の調整ツールを統合する場合は、該当するインスタンス上で MarkLogic を初めて起動する前に、`/etc/marklogic.conf` ファイルを作成することのみが必要です。

以下のスニペットは、管理者パスワードに CloudInit と安全な S3 バケットを利用する CloudFormation テンプレートの例から抜粋した起動設定と AutoScalingGroup セクションです。なお、S3 ファイルの URL 自体に機密性は必要ありません。したがって、CloudFormation パラメータとして安全に渡し、インスタンスが存続する間格納できます。起動設定では、`cfn-init` の呼び出しに単純なスクリプトが使用され、ゾーンの AutoScalingGroup に関連付けられたメタデータリソースへのリファレンスを渡します。メタデータリソースは、AutoScalingGroup セクションの「Properties」タグの兄弟です。

AutoScalingGroup セクションの「files」エントリは、ルート所有者およびグループで `/etc/marklogic.conf` を作成します（所有者の読み取り専用）。

AutoScalingGroup セクションの「services」エントリは、CloudInit が完了した後に MarkLogic を起動し、`/etc/marklogic.conf` または `/etc/sysconfig/MarkLogic` が将来 CloudInit によって更新された場合に MarkLogic を再起動します。

起動設定スニペットの例：

```
"LaunchConfig1" : {
    "Type" : "AWS::AutoScaling::LaunchConfiguration",
    "Properties" : {
        ....},
  "UserData": {"Fn::Base64": {"Fn::Join": [
    "",
    [
      "#!/bin/bash\n",
      "function error_exit\n",
      "{\n",
      "logger -t MarkLogic \"$1\n",
      "exit 1\n",
      "}\n",
      "yum update -y aws-cfn-bootstrap\n",
      "yum update -y\n",
      "# Install application\n",
      "/opt/aws/bin/cfn-init -v -s ",
      {"Ref": "AWS::StackId"}, " -r ASG1 --region ",
      {"Ref": "AWS::Region"}, " || error_exit 'Failed to run cfn-
init'\n",
      "\n",
      "# All is well so signal success\n",
      "\n"
    ]
  ]}}}
```

AutoScalingGroup スニペットの例 :

```

"ASG1" : {
  "Type" : "AWS::AutoScaling::AutoScalingGroup",
  "Properties" : {
    .....
  },
  "Metadata": {
    "MarkLogic::MetaDataType": "2015-07-17-14:49:23",
    "AWS::CloudFormation::Init": {
      "config": {
        "files": {"/etc/marklogic.conf": {
          "content": {"Fn::Join": [
            "",
            [
              "MARKLOGIC_CLUSTER_NAME=", {"Ref": "MarkLogicDDBTable"}, "\n",
              "MARKLOGIC_EBS_VOLUME=", {"Ref": "MarkLogicVolume1"}, "\n",
              "MARKLOGIC_NODE_NAME=NodeA#\n",
              "MARKLOGIC_ADMIN_USERNAME=", {"Ref": "AdminUser"}, "\n",
              "# Password obtained via protected S3 file\n",
              "# MARKLOGIC_ADMIN_PASSWORD=\n",
              "# $(s3 cp --region us-west-2 s3://bucket/secret-password - ) \n",
              "MARKLOGIC_ADMIN_PASSWORD=$( aws s3 --region ",
              {"Ref": "AWS::Region"}, " cp ", {"Ref": "AdminPassS3URL"}, " - )\n",
              "MARKLOGIC_CLUSTER_MASTER=0\n"
            ] ]} ,
          "mode": "000400",
          "owner": "root",
          "group": "root"
        }},
        "services": {"sysvinit":
{"MarkLogic": {
  "enabled": "true",
  "ensureRunning": "true",
  "files": [
    "/etc/marklogic.conf",
    "/etc/sysconfig/MarkLogic"
  ] }
}}
}}

```

3.8 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 サーバーの管理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](#)
- [Amazon S3 \(Simple Storage Service\) 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」(75 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」(40 ページ) の [手順 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 の管理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。

The screenshot shows the AWS Management Console interface for an Elastic Load Balancing (ELB) instance. The left sidebar shows the navigation menu with 'Load Balancers' selected. The main content area shows the details of the selected load balancer, including the 'DNS Name' field which is highlighted and has the URL copied to the clipboard. The URL is: `http://DLEE-CF-L-ElasticL-OCCR192PW000-925510329.us-east-1.elb.amazonaws.com:7800`.

Load Balancer Name	DNS Name	Port Configuration
DLEE-CF-L-ElasticL-OCCR192PW000	DLEE-CF-L-ElasticL-OCCR192PW000-925510329.us-east-1.elb.amazonaws.com	8000 (HTTP) forwarding to 8000 (HTTP) Stickiness: Disabled

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-OCCR192PW000-925510329.us-east-1.elb.amazonaws.com` である場合、MarkLogic ポート 7800 にアクセスするには、ブラウザで次の URL を入力します。

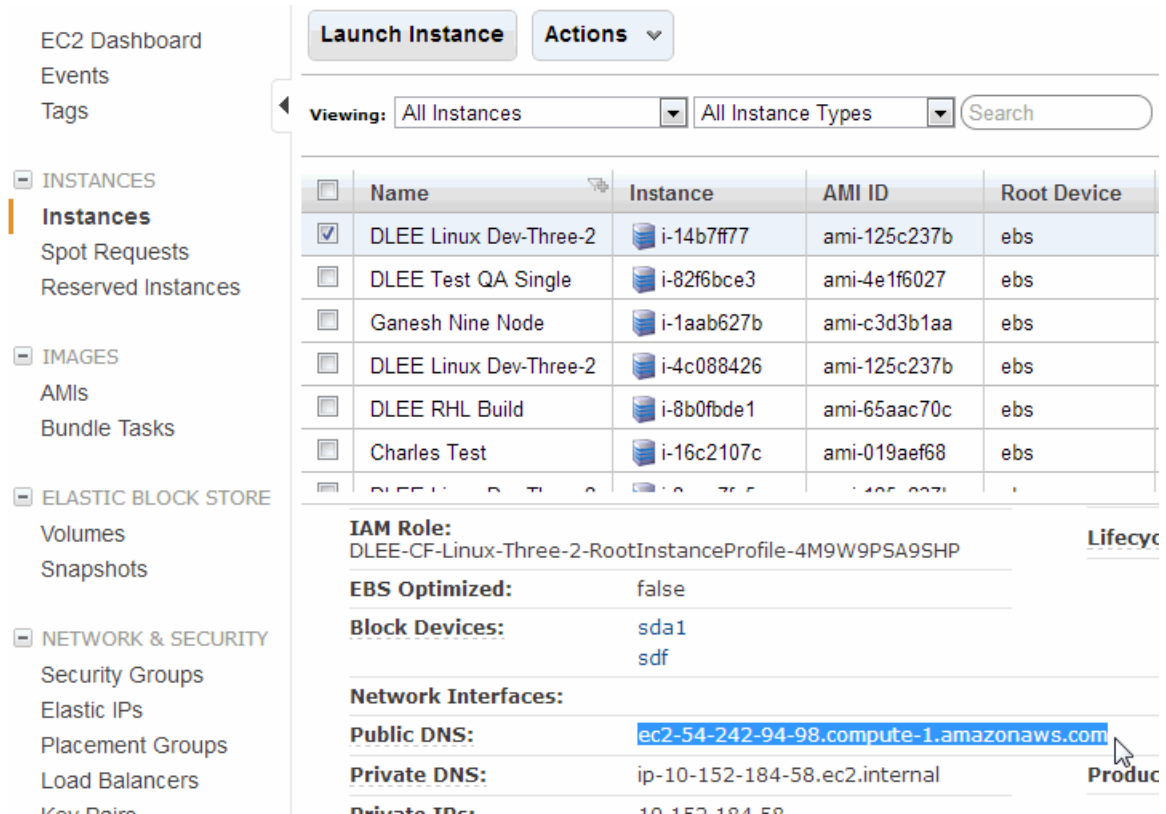
```
http://DLEE-CF-L-ElasticL-OCCR192PW000-925510329.us-east-1.elb.amazonaws.com:7800
```

注： MarkLogic ポート 8001 へのアクセスにロードバランサーを使用しないでください。管理画面は、ロードバランサーの背後で使用するには設計されていません。

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。



The screenshot shows the AWS Management Console interface for EC2 instances. On the left is a navigation menu with categories like INSTANCES, IMAGES, ELASTIC BLOCK STORE, and NETWORK & SECURITY. The main area displays a table of instances. The first instance, 'DLEE Linux Dev-Three-2', is selected. Below the table, the instance's configuration details are shown, including IAM Role, EBS Optimized status, Block Devices, and Network Interfaces. The 'Public DNS' field is highlighted in blue, showing the value 'ec2-54-242-94-98.compute-1.amazonaws.com'.

Name	Instance	AMI ID	Root Device
DLEE Linux Dev-Three-2	i-14b7ff77	ami-125c237b	ebs
DLEE Test QA Single	i-82f6bce3	ami-4e1f6027	ebs
Ganesh Nine Node	i-1aab627b	ami-c3d3b1aa	ebs
DLEE Linux Dev-Three-2	i-4c088426	ami-125c237b	ebs
DLEE RHL Build	i-8b0fbde1	ami-65aac70c	ebs
Charles Test	i-16c2107c	ami-019aef68	ebs

IAM Role: DLEE-CF-Linux-Three-2-RootInstanceProfile-4M9W9PSA9SHP **Lifecycle**

EBS Optimized: false

Block Devices: sda1, sdf

Network Interfaces:

Public DNS: ec2-54-242-94-98.compute-1.amazonaws.com

Private DNS: ip-10-152-184-58.ec2.internal **Product**

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、管理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
```

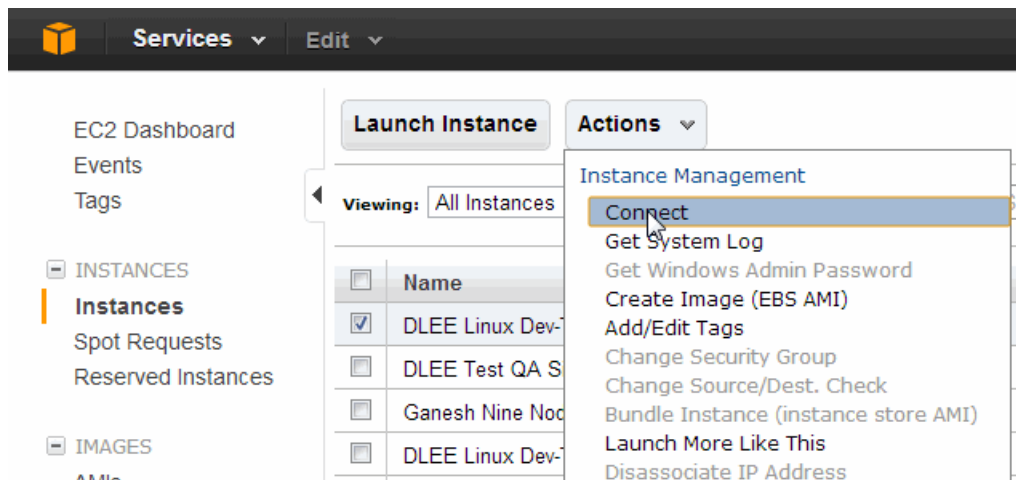
4.2 EC2 インスタンスへのアクセス

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」(75 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。

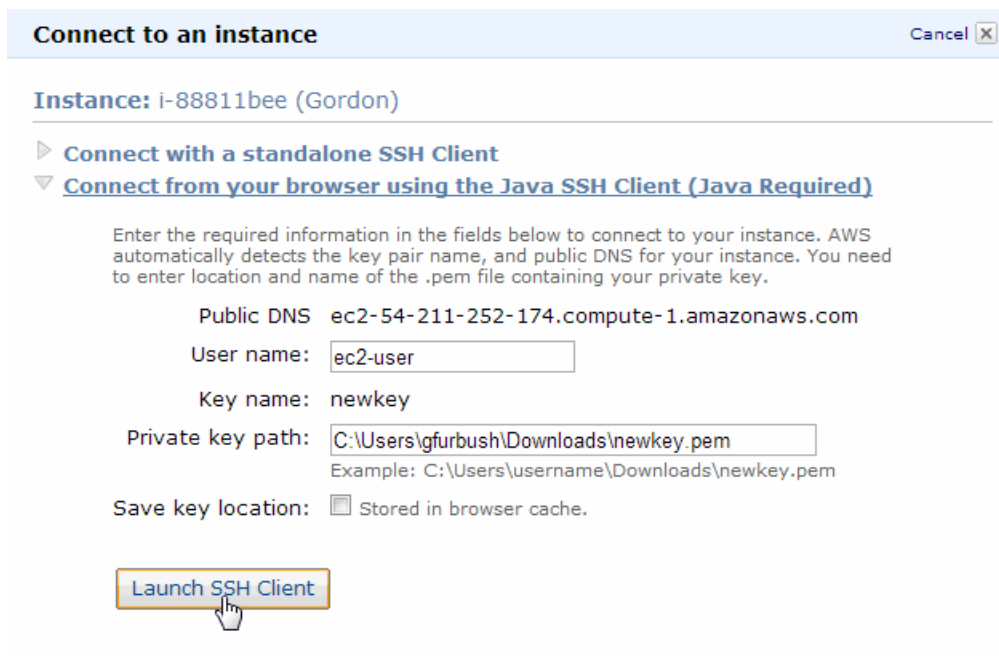
注： ロードバランサーには SSH で接続できません。特定の EC2 インスタンスに SSH で接続する必要があります。EC2 インスタンスに SSH で接続するには、ローカルホストにダウンロードした、そのインスタンスで使用されているキーペアが必要です。

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。



Connect to an instance Cancel X

Instance: i-88811bee (Gordon)

▶ **Connect with a standalone SSH Client**

▼ **Connect from your browser using the Java SSH Client (Java Required)**

Enter the required information in the fields below to connect to your instance. AWS automatically detects the key pair name, and public DNS for your instance. You need to enter location and name of the .pem file containing your private key.

Public DNS: ec2-54-211-252-174.compute-1.amazonaws.com

User name:

Key name: newkey

Private key path:
Example: C:\Users\username\Downloads\newkey.pem

Save key location: Stored in browser cache.

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」(73 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」(22 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（下表「デフォルトの EBS マウントポイント」を参照）。

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。

デフォルトの EBS マウントポイント

EC2 デバイス	RedHat デバイス	Linux デバイス	マウントポイント
/dev/sdf	/dev/xvdj	/dev/xvdf	/var/opt/MarkLogic
/dev/sdg	/dev/xvdk	/dev/xvdg	/var/opt/volume1
/dev/sdh	/dev/xvdl	/dev/xvdh	/var/opt/volume2
/dev/sdi	/dev/xvdm	/dev/xvdi	/var/opt/volume3
/dev/sdj	/dev/xvdn	/dev/xvdj	/var/opt/volume4
/dev/sdk	/dev/xvdo	/dev/xvdk	/var/opt/volume5
/dev/sdl	/dev/xvdp	/dev/xvdl	/var/opt/volume6
/dev/sdm	/dev/xvdq	/dev/xvdm	/var/opt/volume7
/dev/sdn	/dev/xvdr	/dev/xvdn	/var/opt/volume8
/dev/sdo	/dev/xvds	/dev/xvdo	/var/opt/volume9

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 Amazon S3 (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。次に例を示します。

- 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. 管理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 domain	<input type="text" value="s3.amazonaws.com"/>	The simple storage service internet domain name.
s3 protocol	<input type="text" value="http"/>	The simple storage service internet internet protocol.
s3 server side encryption	<input type="text" value="none"/>	Specifies server side encryption for data at rest on the simple storage service.

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。

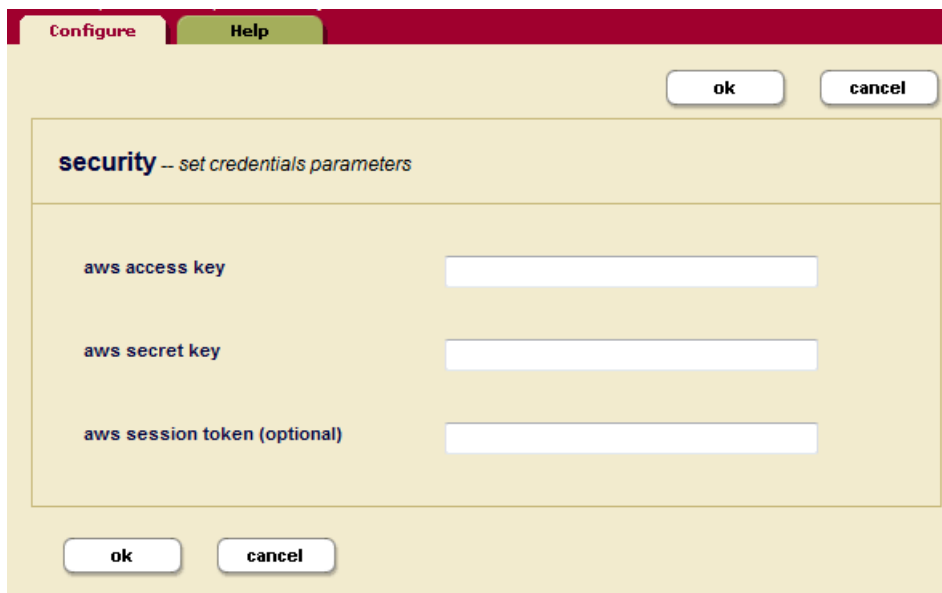
1. サーバーのセキュリティデータベースで設定されている資格情報
2. 環境変数
3. IAM ロール

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. 管理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、AWS 秘密キー、およびオプションの AWS セッショントークンを入力します。



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
- MARKLOGIC_AWS_SESSION_TOKEN -- オプションの AWS セッショントークン

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。

IAM ロールは、MARKLOGIC_AWS_ROLE 環境変数が設定されている場合にサーバー上でのみ使用されます。これは、EC2 設定を無効にしない限り (MARKLOGIC_EC2_HOST=0 を設定するなど)、自動的に行われます。この場合、サーバーは MARKLOGIC_AWS_ROLE 変数を使用しません。

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」(87 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`」(78 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

既存の CloudFormation テンプレートの更新は、ソフトウェアと IT アーキテクチャの互換性がある限り、またはその更新が、管理対象クラスタ機能またはデータで必要とされる AWS リソースへの破壊的な修正を必要としない場合に可能です。テンプレート更新の影響に関する一般的なガイダンスは、<http://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/using-cfn-updating-stacks.html> にあります。

最新のサンプルテンプレートには、V8.0.3 から始まるバージョンが付けられています。メジャーバージョン番号の変更は、管理対象クラスタ機能の互換性のない実装を表します。したがって、以前の CloudFormation テンプレートを使用して作成された EC2 スタックは、新しい CloudFormation テンプレートで更新できません。

同じメジャーテンプレートバージョン内では、元のテンプレート内の AMI ID を、該当する Marketplace の最新の AMI ID を使用して更新することにより、アップグレードがサポートされます。

- AWS リージョン
- 準仮想化タイプ (HVM 対 PVM)
- インスタンスタイプ
- EBS ボリュームタイプ

警告 カスタマイズによっては、同じバージョンおよび AMI を使用している場合でも、一部のスタック更新は、実行中のクラスタに対して破壊的になることがあり、実稼働の作業に適用する前にテストする必要があります。

次の手順は、同じメジャーテンプレートバージョン内で MarkLogic の新しいリリースに新しい AMI を使用するように、スタックをアップグレードする方法について説明しています。

1. 元のテンプレート内で AMI ID を探し、AWS Marketplace からの対応する更新済み AMI ID か、<http://developer.marklogic.com/products/aws> に示されている AMI ID を見つけます。例えば、元のテンプレート内の `AWSRegionArch2AMI` 定義は次のようになります。

```
"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"
    }
}
},
```

例えば、インスタンスが us-east-1 および us-west-2 リージョンに置かれている場合、新しいテンプレートを開き、AWSRegionArch2AMI 定義を探し、us-east-1 および us-west-2 リージョンの AMI ID をコピーします。例えば、新しいテンプレートには次の内容が含まれます。

```
"us-east-1" : {
    "HVM" : "ami-96ffe7fe"
},
"us-west-2" : {
    "HVM" : "ami-75d3f245"
},
```

続いて次のように AWSRegionArch2AMI 定義を更新できます。

```
"AWSRegionArch2AMI":
{
    "us-east-1":
    {
        "PVM": "ami-41633528",
        "HVM": "ami-96ffe7fe"
    },
    "us-west-2":
    {
        "PVM": "ami-e85bc5d8",
        "HVM": "ami-75d3f245"
    },
    "eu-west-1":
    {
        "PVM": "ami-68fa1c1f",
        "HVM": "ami-56fa1c21"
    }
}
},
```

2. 重要なデータをバックアップします。
3. 更新した CloudFormation テンプレートで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。

4. 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。セキュリティデータベースを含んだインスタンスまたはノードから始めます。

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。

5. EC2 ダッシュボードで、他のノードを終了します。ローリングアップグレードでは（また、推奨される手順として）、他のノードを1つずつ終了します。これらは UI を操作しなくても起動して再接続されます。
6. 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）。
7. [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」(80 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」(82 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」(33 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。

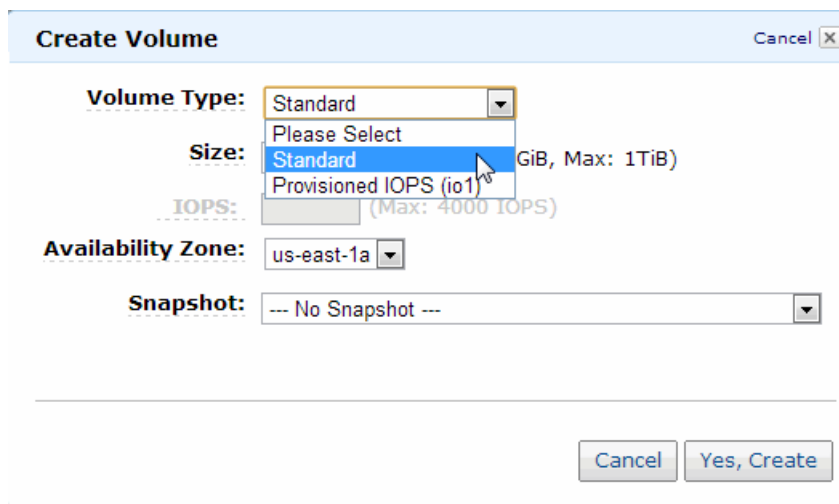
4.10.1 EBS ボリュームの作成

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。



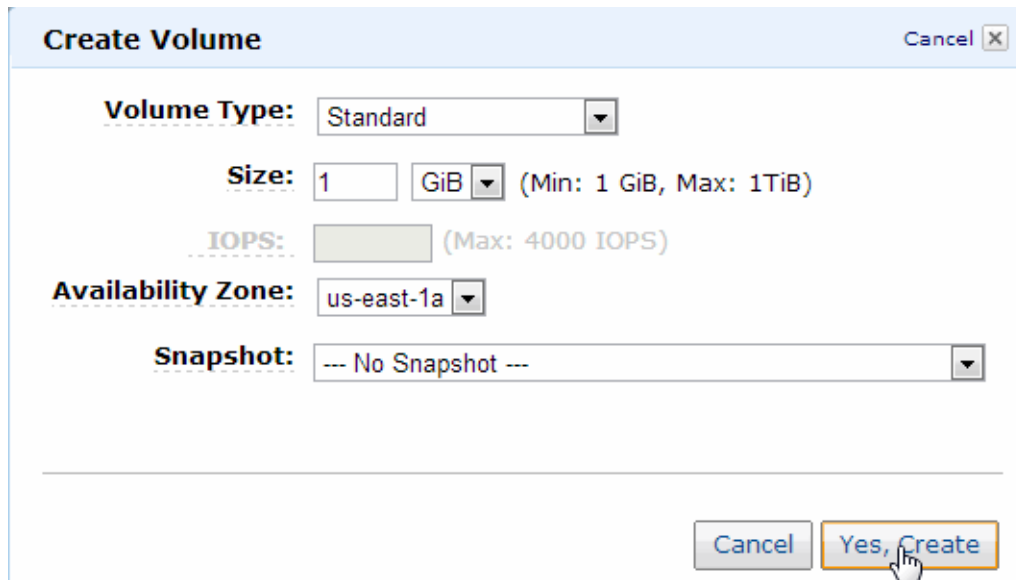
2. [Create Volume] ウィンドウで、[Volume Type] プルダウンメニューからボリュームタイプを指定します。



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。



Create Volume Cancel

Volume Type: Standard

Size: 1 GiB (Min: 1 GiB, Max: 1TiB)

IOPS: (Max: 4000 IOPS)

Availability Zone: us-east-1a

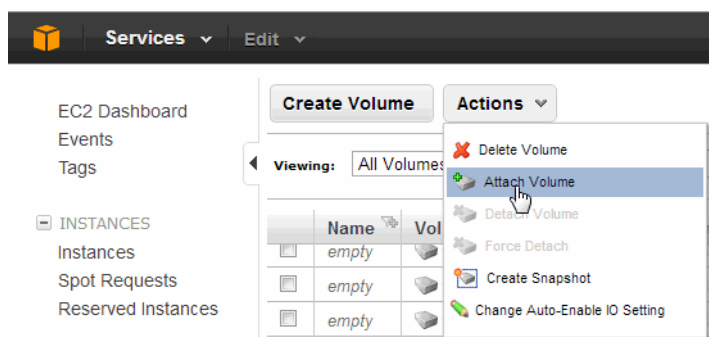
Snapshot: --- No Snapshot ---

Cancel Yes, Create

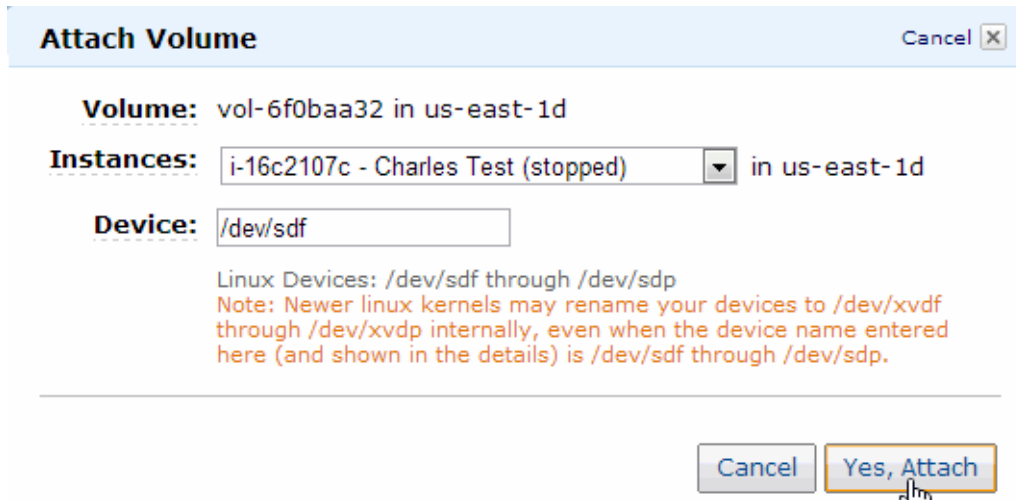
4.10.2 EBS ボリュームのインスタンスへのアタッチ

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、そのホストで管理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 の `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、管理画面からまたは 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」（69 ページ）の説明に従ってスタックを削除します。