

3 セキュリティグループの指定する

DBインスタンスが復元され、ステータスが「利用可能」になった後、インスタンス一覧から、復元されたDBインスタンスを選択し、「インスタンスの操作」をクリックして「変更」を選択して、変更を行います。

「個のセキュリティグループ」を必要に応じて変更します。ここでは、WordPressのデータベースを復元したので、「WordPress…」を指定して①、「次へ」をクリックします②。

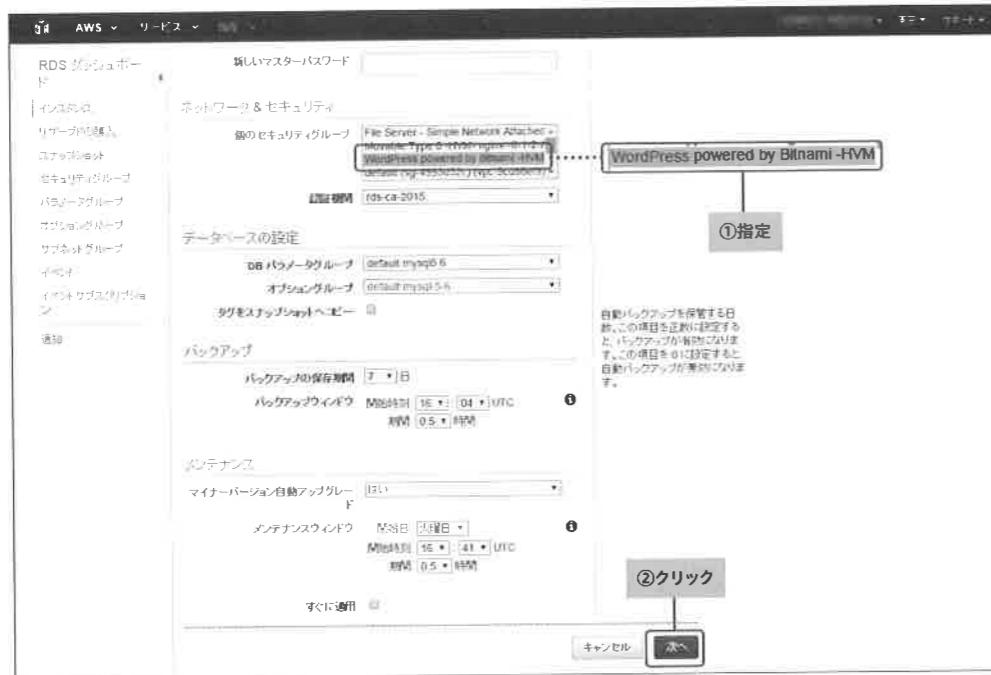


図9 セキュリティグループの指定

コスト効率的なバックアップを設定する

8-3-4 日時を指定して復元する

スナップショットからの復元以外に、ポイントインタイム復元機能もあります。復元したい特定の日時を指定することができます。

1 RDSマネージメントコンソールよりインスタンスを選択する

RDSマネージメントコンソールを開き、左サイドのメニューより「インスタンス」を選択します(図10①)。

一覧より復元したいインスタンスを選択し、「インスタンスの操作」をクリックして②、「特定時点への復元」を選択します③。

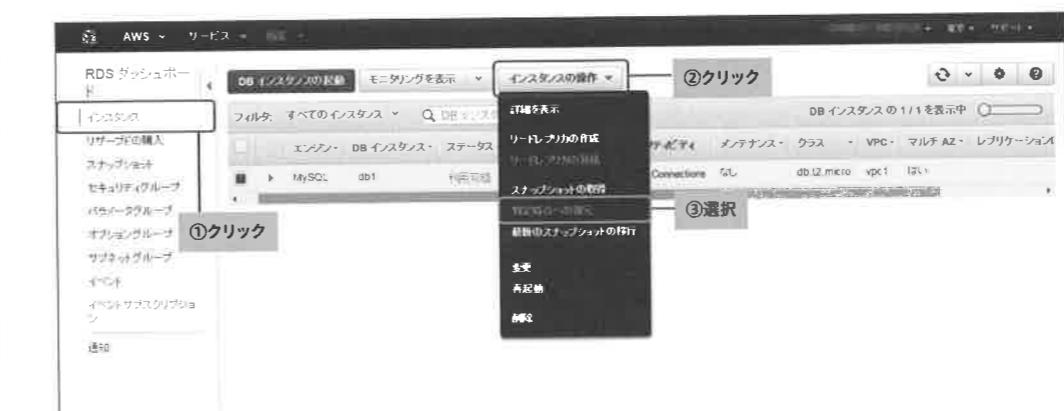


図10 特定時点の復元

2 復元可能な日時や任意の日時を入力する

復元可能な最新日時を選ぶか、任意の日時を入力することができます(図11①)。その他の入力項目は前述した内容を参考にして設定し②、最後に「DBインスタンスの起動」をクリックします。

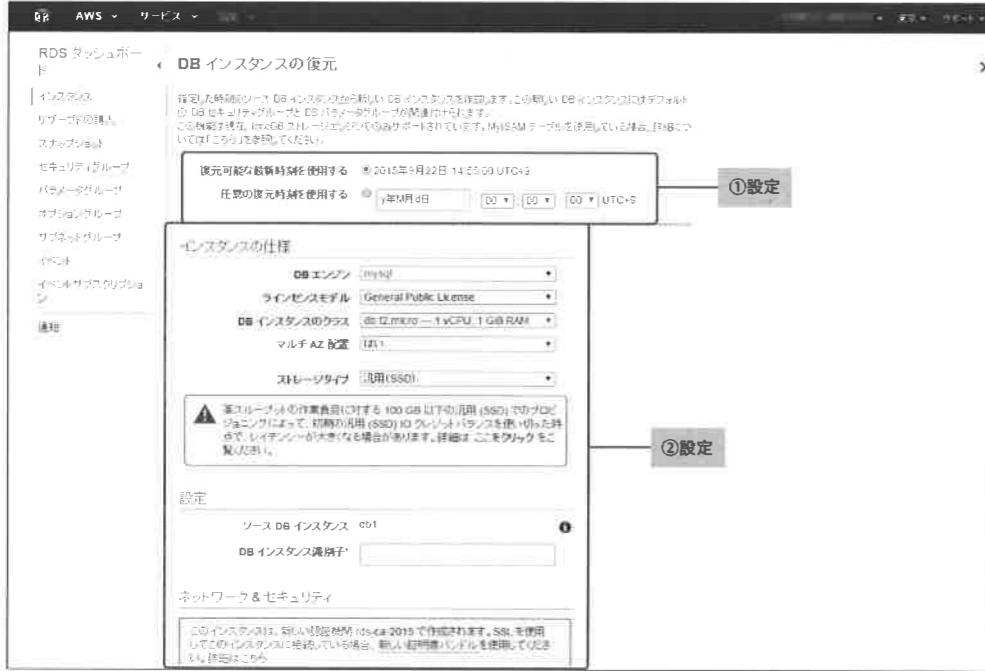


図11 復元日時の指定



CloudWatchを活用して リソースコストを監視する

AWSで利用している各サービスの状況をリアルタイムで監視することができるCloudWatchについて、その設定方法などを解説します。

- 1 メトリックスとアラームを利用する
- 2 SNSでメール配信を設定する
- 3 EC2のCPU使用率を監視する
- 4 S3のバケット使用量を監視する
- 5 RDSのデータベース接続数を監視する



9.1 メトリックスとアラームを利用する

↑ メトリックス（何を監視するのか）、アラーム（どのような状態になったら通知するのか）というCloudWatchの基本的な機能について解説します。

9-1-1 メトリックスとは

監視したい情報の種類のこと「メトリックス」と言います。利用している(EC2、RDSなど)サービスごとに、メトリックスは用意されています。CPUの使用率、空きストレージ容量、データベースへの接続数などが具体的なメトリックスです。主なメトリックスは、表1～4のとおりです。

MEMO / CloudWatchメトリックスリファレンス

CloudWatchについては以下も参考にしてください。

- CloudWatch メトリックスリファレンス
https://docs.aws.amazon.com/ja_jp/AmazonCloudWatch/latest/DeveloperGuide/CW_Support_For_AWS.html

表1 S3のメトリックス一覧

| メトリックス名 | 内容 |
|-----------------|--------------|
| NumberOfObjects | バケット内オブジェクト数 |
| BucketSizeBytes | バケット内使用バイト数 |

表2 EC2のメトリックス一覧

| メトリックス名 | 内容 | メトリックス名 | 内容 |
|----------------|--------------|----------------|--------------|
| CPUUtilization | CPUの使用率 | DiskWriteBytes | ディスク書き込みバイト数 |
| DiskReadBytes | ディスク読み込みバイト数 | DiskWriteOps | ディスク書き込み回数 |
| DiskReadOps | ディスク読み込み回数 | NetworkIn | 受信バイト数 |

(続き)

| メトリックス名 | 内容 |
|-------------------|----------|
| NetworkOut | 送信バイト数 |
| StatusCheckFailed | ステータスフラグ |

(続き)

| メトリックス名 | 内容 |
|----------------------------|----------------|
| StatusCheckFailed_Instance | インスタンスステータスフラグ |
| StatusCheckFailed_System | システムステータスフラグ |

表3 EBSのメトリックス一覧

| メトリックス名 | 内容 |
|-------------------|----------|
| VolumeIdleTime | アイドル秒数 |
| VolumeQueueLength | キュー残数 |
| VolumeReadBytes | 読み込みバイト数 |
| VolumeReadOps | 読み込み回数 |

| メトリックス名 | 内容 |
|----------------------|----------|
| VolumeTotalReadTime | 読み込み総秒数 |
| VolumeTotalWriteTime | 書き込み総秒数 |
| VolumeWriteBytes | 書き込みバイト数 |
| VolumeWriteOps | 書き込み回数 |

表4 RDSのメトリックス一覧

| メトリックス名 | 内容 |
|---------------------|------------------|
| BinLogDiskUsage | バイナリログディスク使用バイト数 |
| CPUCreditBalance | CPUクレジット累積数 |
| CPUCreditUsage | CPUクレジット消費数 |
| CPUUtilization | CPUの使用率 |
| DatabaseConnections | データベースへの接続数 |
| DiskQueueDepth | ディスクI/Oキュー残数 |
| FreeStorageSpace | 空きストレージ容量 |
| FreeableMemory | メモリ空き容量 |

| メトリックス名 | 内容 |
|---------------------------|------------|
| NetworkReceiveThroughput | 受信トラフィック量 |
| NetworkTransmitThroughput | 送信トラフィック量 |
| ReadIOPS | 読み込み平均回数 |
| ReadLatency | 読み込み平均時間 |
| ReadThroughput | 読み込み平均バイト数 |
| SwapUsage | スワップ使用量 |
| WriteIOPS | 書き込み平均回数 |
| WriteLatency | 書き込み平均時間 |
| WriteThroughput | 書き込み平均バイト数 |

MEMO / カスタムメトリックス

標準で用意されているメトリックス以外に、何か監視したい項目がある場合には、カスタムメトリックスを、コマンドライン(またはAPI)を使って、自分自身でメトリックスを追加することもできます。

9-1-2 アラームとは

アラームはメトリックスを監視します。設定した「しきい値」を超えた場合、SNSを通じて通知を行うことができます。

監視には、表5のようにどの位の時間を対象とするのか(間隔)、表6のようにその間のどの値を比較するのか(統計)を設定することができます。

表5 間隔

| 間隔 |
|-----|
| 5分間 |
| 15分 |
| 1時間 |
| 6時間 |
| 1日 |

9

表6 統計

| 統計 |
|---------|
| 平均 |
| 最小 |
| 最大 |
| 合計 |
| データサンプル |

しきい値の単位は表7のとおりです。しきい値は、しばらく運用してみてどの位か把握してから再設定するといいでしよう。

アラームは、指定した時間間隔で、前回の値と比較を行い、表8に掲げた状態に変化します。

表7 しきい値

| しきい値の単位 |
|---------|
| Bytes |
| Count |
| Percent |
| Seconds |

表8 アラームの状態

| アラームの状態 | |
|---------|-------------|
| OK | しきい値を下回っている |
| 警告 | しきい値を上回っている |
| 不足 | 判定できない状態 |

アラームの状態が「警告」になった場合、「9-2 SNSでメール配信を設定する」で解説するSNSを使用して通知を行います。



9-2 SNSでメール配信を設定する

メッセージの配信を管理できるSNSを使って、CloudWatchアラームをメールで送信できるように設定する方法を解説します。

9-2-1 SNSとは

SNS(Simple Notification Service)は(図1)、各サービスから出力される通知(Publish)を、設定された配信先(Subscriber)に送信します。

配信先はトピックで管理され、複数に配信することや、配信先を追加・変更することができます。通知を出力する側は、トピックを指定することにより、配信先を直接管理する必要がありません。

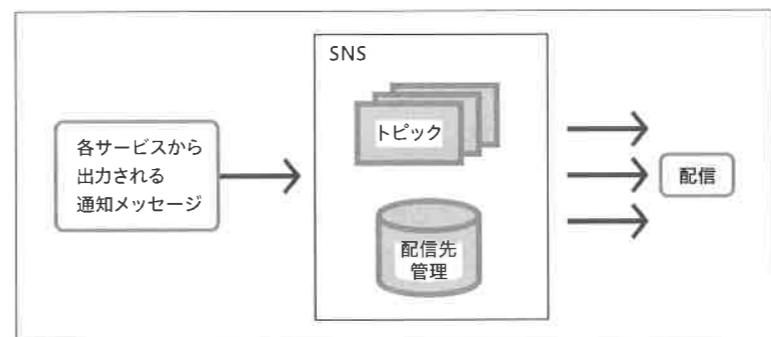


図1 SNS

本章では、配信先としてメールを使用しますが、表1のようなプロトコルがサポートされています。

表1 プロトコル

| プロトコル |
|------------|
| HTTP |
| HTTPS |
| Email |
| Email-JSON |

| プロトコル |
|-------------|
| Amazon SQS |
| Application |
| AWS Lambda |

1 SNSマネージメントコンソールを開く

コンソールのホームから「SNS」(プッシュ通知サービス)をクリックします(図2)。

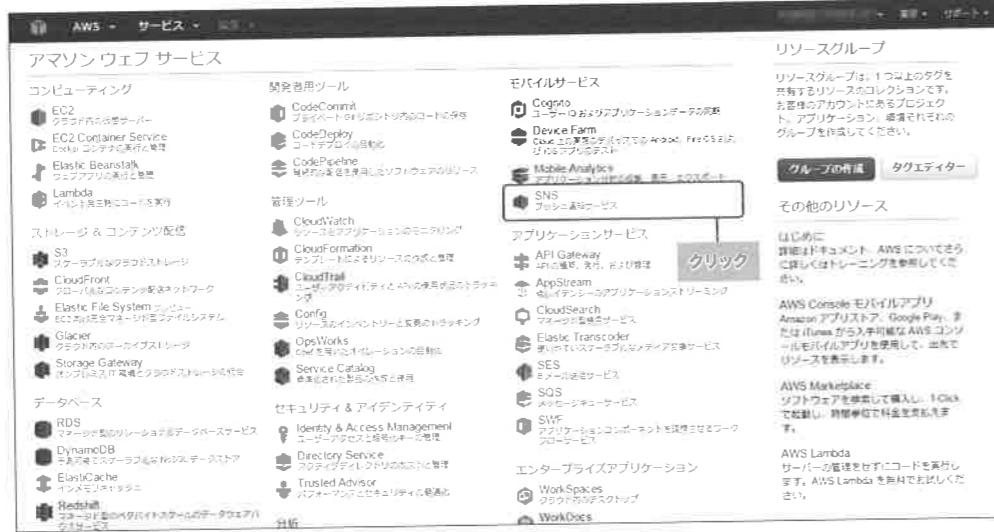


図2 コンソールのホームから「SNS」を選ぶ

2 「Create Topic」をクリックする

配信先を管理するトピックを作成します。「Get Startted」をクリックして(図3①)、SNS Homeから、「Create Topic」をクリックします②(左サイドのメニューより「Topics」を選択し、「Create new topic」をクリックしても同様)。



図3 トピックの作成

3 「Topic name」と「Display name」を入力する

「Topic name」と(図4①)、「Display name」を入力します②。ここでは両方とも「admin」と入力します。「Create topic」をクリックします③。

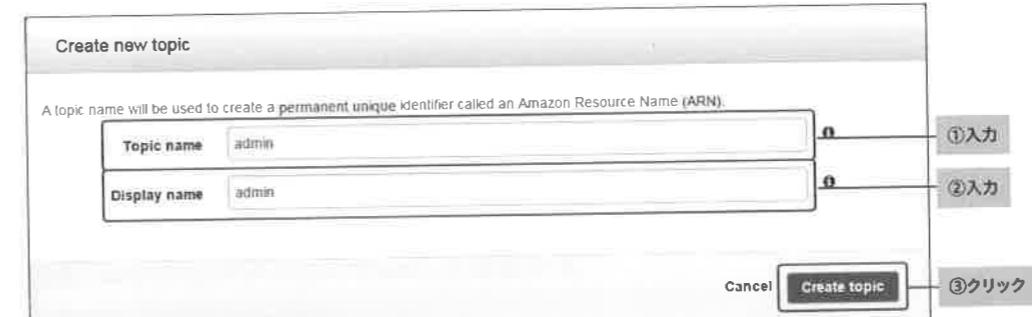


図4 トピックの作成

4 「Create Subscription」をクリックする

作成したトピックに配信先を追加します。配信先は複数登録できますが、ここでは「メール」を1つ登録します。まず「Create Subscription」をクリックします(図5)。

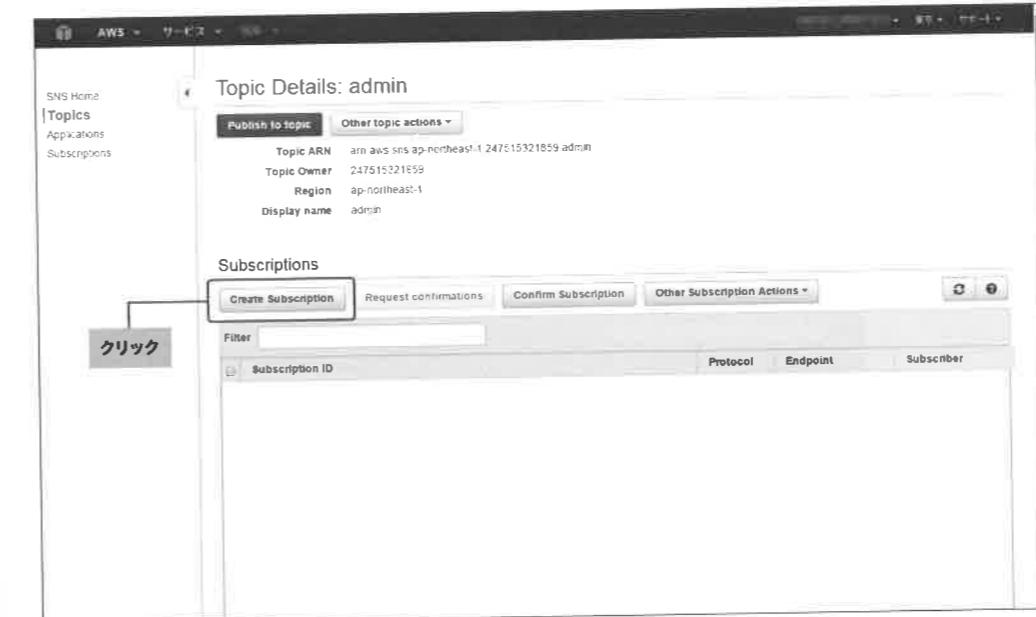


図5 配信先の追加

5 配信先を追加する

「Protocol」では、「Email」を選択して(図6①)、「Endpoint」には、配信先のメールアドレスを入力します②。最後に「Create Subscription」をクリックします③。

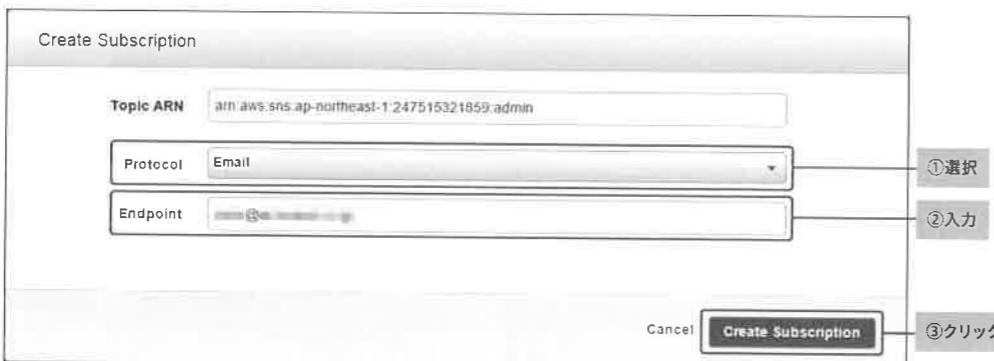


図6 プロトコルの選択とエンドポイントの入力

6 「Subscription ID」を確認する

配信先が一覧に追加されます(図7)。「Subscription ID」には、「PendingConfirmation」と表示されていて、確認メールでの承認待ち状態となっています。

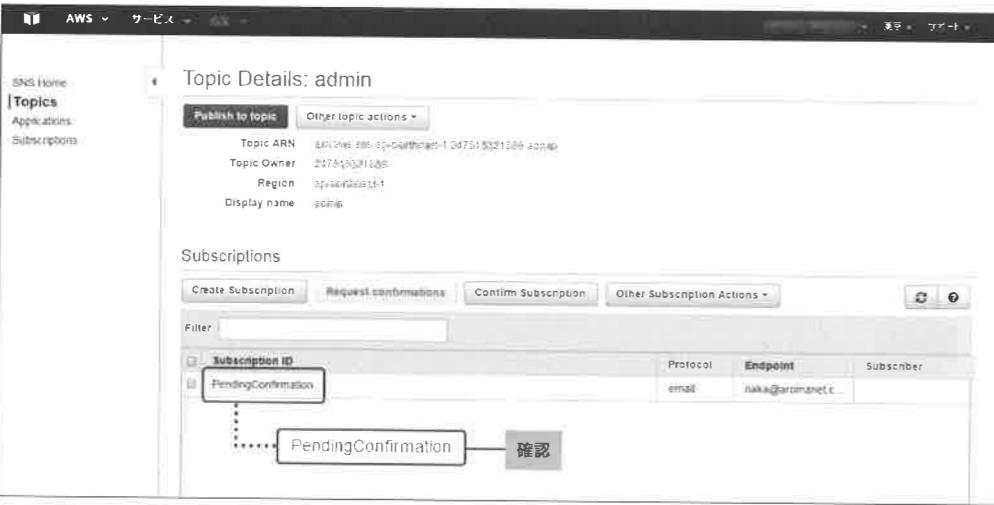


図7 配信先作成後

7 メールを確認する

配信先のメールを受信できる環境(メールソフトやWebメールなど)より、届いたメールを確認し、「Confirm subscription」をクリックします(図8)。



図8 届くメールの内容

8 「Subscription confirmed!」を確認する

図9の画面のように「Subscription confirmed!」と表示されれば、承認完了です。

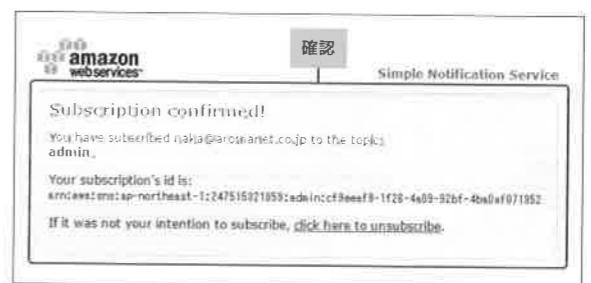


図9 メールアドレスの確認完了

9 「Subscription ID」を確認する

前述した承認待ち状態の画面で「リロード」をクリックします(図10①)。最新状態が反映され、「Subscription ID」には、「arn:」で始まるIDが表示されます②。



図10 リロードしてIDを確認

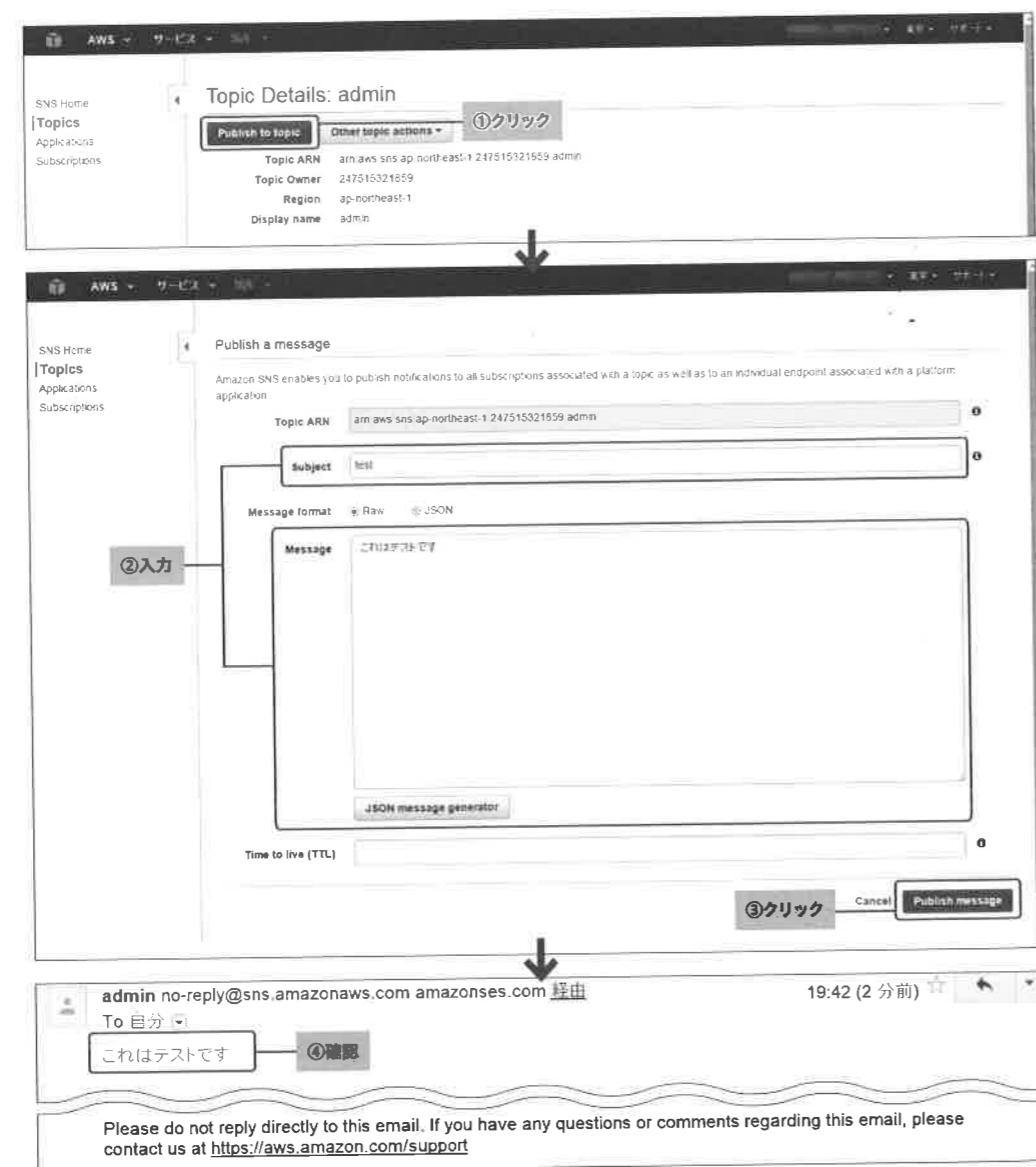


図11 メッセージの送信

10 メッセージ送信のテストを行う

作成したトピック「admin」が正しく動作するかどうかテストしてみましょう。

「Publish to topic」をクリックして(図11①)、Subject(タイトル)と、Message(メッセージ)を入力し②、最後に「Publish message」をクリックします③。

配信先メールアドレスに送信したメッセージが届いているかどうか確認してください④。

MEMO / リソースコストの削減について

CloudWatchは、各サービスのリソースを監視することができます。しきい値を超えるとアラートが表示されます。これがポピュラーな使い方ですが、アラートが出なくとも十分使い道があります。各リソースをあまり使っていないのであれば、何か最適化できないか検討する材料になります。CPU使用率が低ければ、インスタンスタイプをワンクラス下げる、ディスク使用量が少なければ、ディスクサイズを変更する、というように、AWSの使った分だけ課金されるしくみをうまく利用してコストを削減することができます。



9.3 EC2のCPU使用率を監視する

EC2のCPU使用率を監視するために、CloudWatchを使ってメトリックスを選び、アラームの設定を行います。

1 CloudWatchマネージメントコンソールを開く

コンソールのホームから「CloudWatch」(リソースとアプリケーションのモニタリング)をクリックします(図1)。



図1 コンソールのホームからCloudWatchを選ぶ

2 メトリックスを選ぶ

左サイドのメニューより「メトリックス」(図2①)→「EC2」②を選ぶと、インスタンス別にメトリックスの位置が表示されます③。CloudWatchでチェックしたいインスタンスのIDをあらかじめ調べておきましょう。ここでは、インスタンス「i-ec618a49」の「CPUUtilazation」の行にチェックを入れて④、メトリックスを選びます。

3 EC2のCPU使用率を監視する

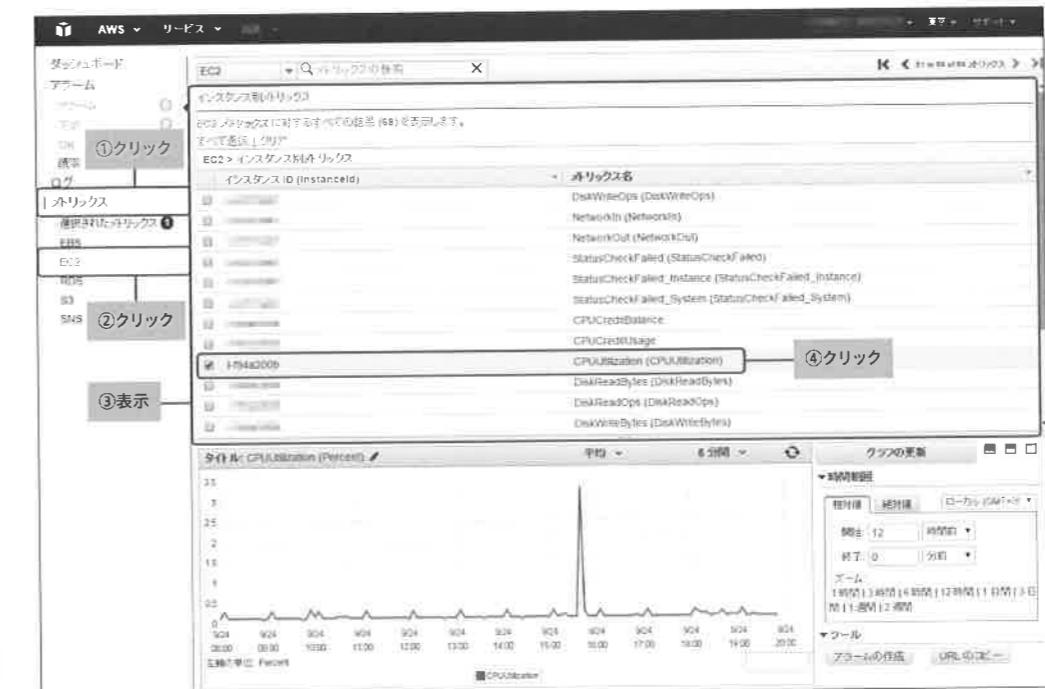


図2 メトリックスの選択

3 アラームを作成する

画面の右下にある「アラームの作成」をクリックすると(図3)、ウィンドウが表示されます。表1を参考にして入力し(図4①～⑧)、最後に「アラームの作成」をクリックします⑨。

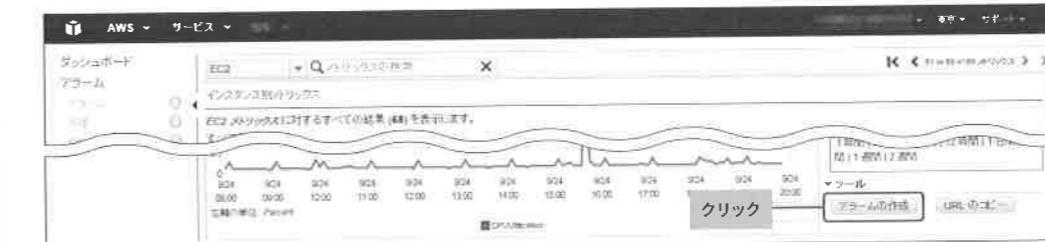


図3 「アラームの作成」をクリック

表1 アラームの設定

| 項目名 | 設定内容 | 備考 |
|-----------|-------------------|----|
| アラームのしきい値 | | |
| ①名前 | CPU 使用率が80%を超えたとき | — |

CloudWatch を活用してリソースコストを監視する

図4 アラームの作成

| 項目名 | 設定内容 | 備考 |
|---|---------------------|---------------|
| ②説明 | CPU 使用率が 80% を超えたとき | — |
| 次の時 | | |
| ③ [CPUUtilization] が | 80 | — |
| ④ を N 回連続した場合 | 1 | — |
| アクション | | |
| ⑤ アラームが次の時 | 状態: 警告 | — |
| ⑥ 通知の送信先 | admin | SNS で作成したトピック |
| アラームのプレビュー | | |
| ⑦ 間隔 | 5分間 | — |
| ⑧ 統計 | 平均 | — |
| <p>アラームの作成</p> <p>1. メトリックスの選択 2. アラームの定義</p> <p>アラームのしきい値</p> <p>アラームの詳細しきい値を指定します。右のグラフを参照しながら、連切れやすい値を設定してください。</p> <p>名前: CPU 使用率が 80% を超えたとき ①入力</p> <p>説明: CPU 使用率が 80% を超えたとき ②入力</p> <p>次の時: CPUUtilization (CPUUtilization) ③選択 & 入力</p> <p>値: >= 80</p> <p>を: 1 回連続した場合</p> <p>アクション</p> <p>アラームの状態が変わったときに、以下のアクションを実行するかを定義します。</p> <p>通知</p> <p>アラームが次の時: 警告</p> <p>通知の送信先: admin</p> <p>メーリングリスト:</p> <p>+ 通知 + Auto Scaling アクション + EC2 アクション</p> <p>アラームのプレビュー</p> <p>このアラームは、次の場合にトリガーされます 5 分間の間、青線が赤線を上回る。または上昇し続ける</p> <p>CPUUtilization >= 80</p> <p>名前空間: AWS/ECS</p> <p>インスタンス ID: i-084a2000</p> <p>InstanceId: i-084a2000</p> <p>メトリクス名: CPUUtilization</p> <p>間隔: 5 分間</p> <p>統計: 平均</p> <p>④入力</p> <p>⑤選択</p> <p>⑥選択</p> <p>⑦選択</p> <p>⑧選択</p> <p>⑨クリック</p> <p>キャンセル 戻る アラームの作成</p> | | |

(続)

4 状態を確認する①

4 状態を確認する①
アラームを作成した直後は左サイドのメニューにアラームの状態が「不足」と表示されます(図5)。

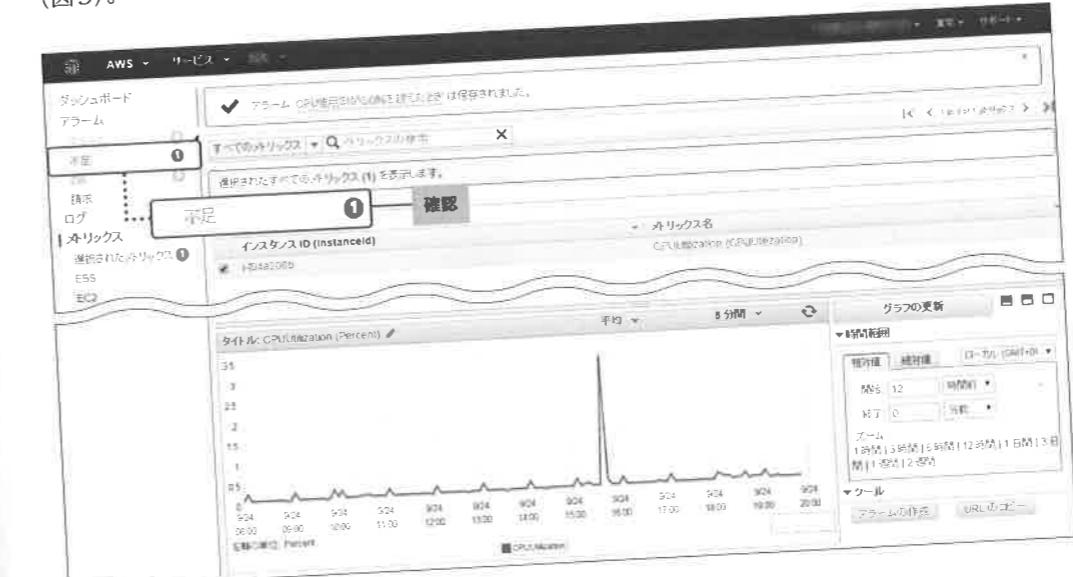


図5 アラームの状態：不足

5 状態を確認する②

図6のようにアラートの状態が「OK」に変われば、現在しきい値を超えておらず、正常であると言えます。

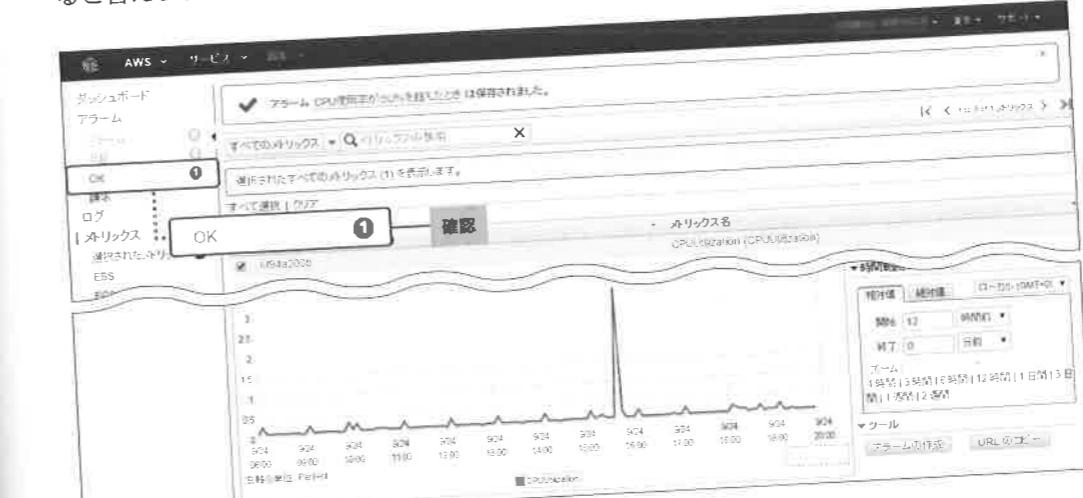


図6 アラームの状態:0

6 アラームの一覧を確認する

左サイドのメニューから「アラーム」をクリックすると(図7①)、登録されているアラームの一覧が表示され②、それぞれ個別に状態をチェックできます。例えば、アラームをクリックすれば、履歴や設定内容を確認できます。

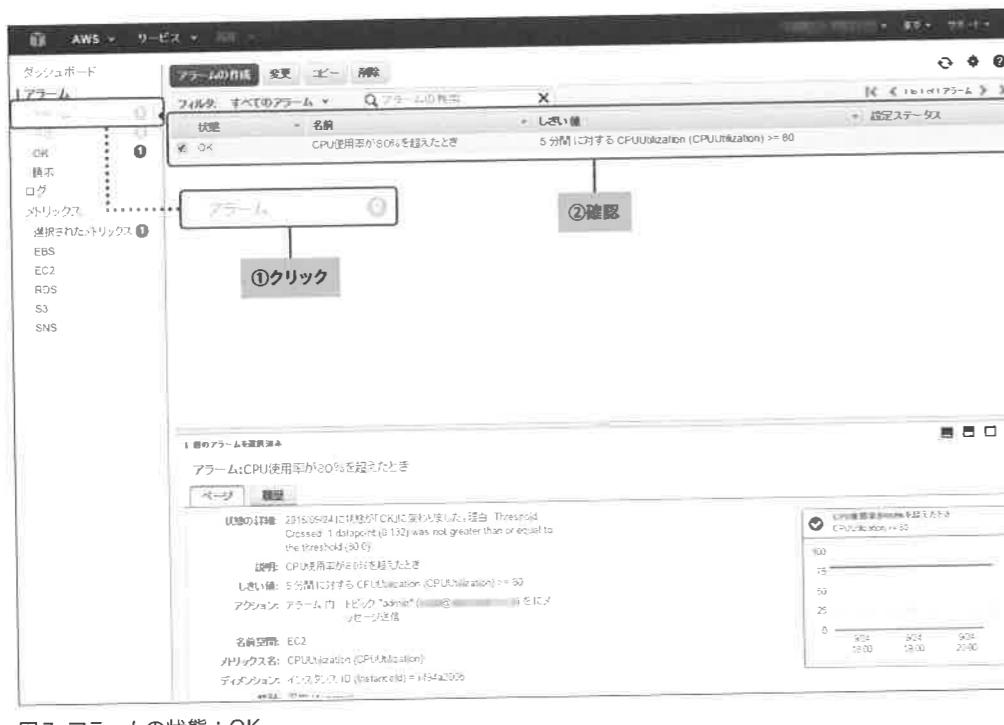


図7 アラームの状態:OK

MEMO / ダッシュボードをカスタマイズできる

CloudWatchのダッシュボードには複数のメトリックスを表示することができます。監視する項目が多くなってきたら、ダッシュボードでグラフや文字すべてをまとめてチェックでき、とても便利です。また、複数のダッシュボードも定義できるので、目的に応じて使い分けることもできます。

MEMO / CloudWatchの無料利用枠について

1ヶ月間で、10個のメトリックス、10個のアラーム、メール通知1,000件分が無料提供されます。

**9 4 S3のバケット使用量を監視する**

S3のバケット使用量を監視するために、CloudWatchを使って、メトリックスを選択、アラームの設定を行います。

1 メトリックスを選ぶ

CloudWatchマネジメントコンソールを開き、左サイドのメニューより「メトリックス」(図1①)→「S3」を選びます②。バケット別にメトリックスの位置が表示されます。ここでは、バケット「www.aws-book.com」の「BucketSizeBytes」の行にチェックを入れて③、メトリックスを選びます。

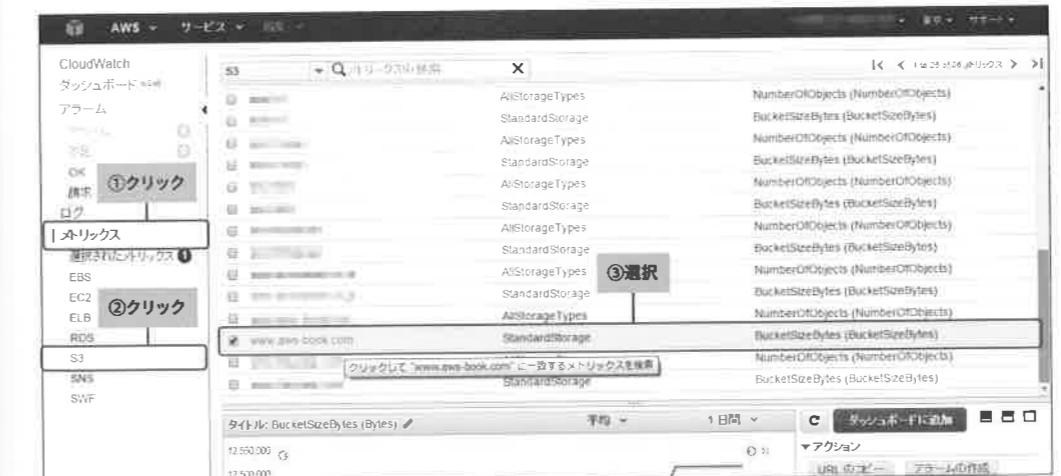


図1 メトリックスを選ぶ

2 アラームを作成する

「アラームの作成」をクリックすると(図2)、図3のようなウィンドウが表示されます。表1のように入力を行い①~⑧、最後に「アラームの作成」をクリックします⑨。



図2 「アラームの作成」をクリック

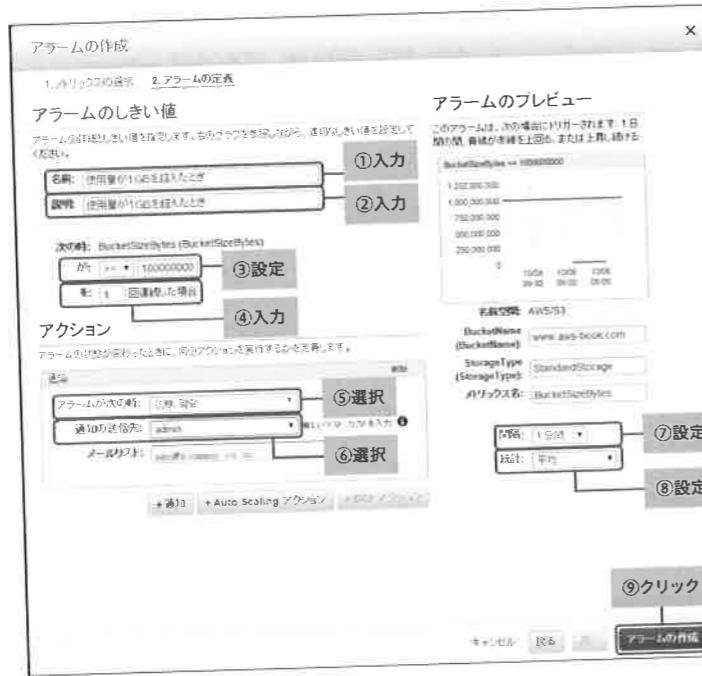


図3 アラームを作成する

表1 S3バケット使用量監視アラーム設定項目

| 項目名 | 設定内容 | 備考 |
|-----------------------|---------------|--------------|
| アラームのしきい値 | | |
| ①名前 | 使用量が1GBを超えたとき | — |
| ②説明 | 使用量が1GBを超えたとき | — |
| 次の時 | | |
| ③ [BucketSizeBytes] が | 1000000000 | カンマは入力しない |
| ④をN回連続した場合 | 1 | — |
| アクション | | |
| ⑤アラームが次の時 | 状態:警告 | — |
| ⑥通知の送信先 | admin | SNSで作成したトピック |
| アラームのプレビュー | | |
| ⑦間隔 | 1日間 | — |
| ⑧統計 | 平均 | — |



9.5 RDSのデータベース接続数を監視する

RDSのデータベース接続数を監視するために、CloudWatchを使って、メトリックスを選ぶ、アラームの設定を行います。

1 メトリックスを選ぶ

CloudWatchマネージメントコンソールを開き、左サイドのメニューより「メトリックス」(図1①)→「RDS」を選択すると②、DBインスタンス識別子別にメトリックスの位置が表示されます。ここでは、DBインスタンス識別子「db1」の「DatabaseConnections」の行にチェックを入れて③、メトリックスを選びます。

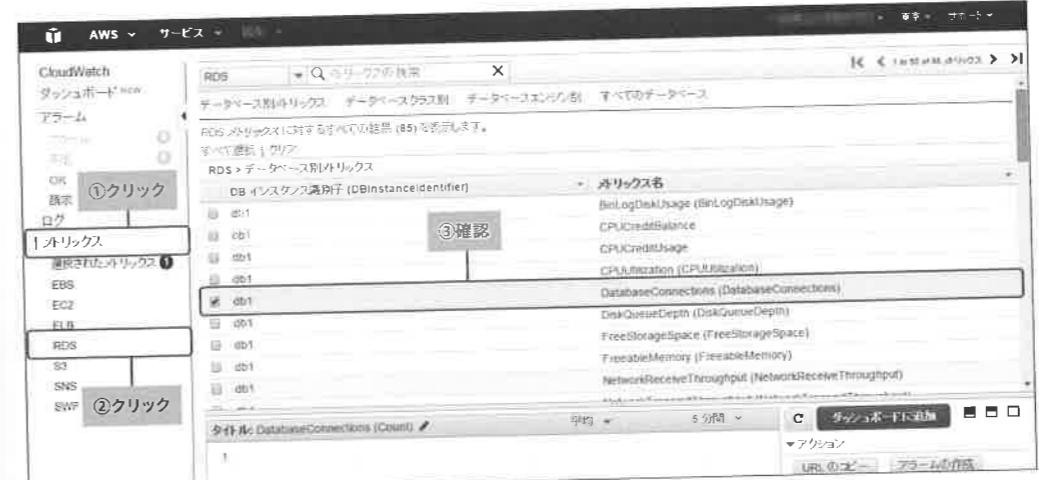


図1 メトリックスを選ぶ

2 アラームを作成する

「アラームの作成」ボタンをクリックすると(図2)、図3のようなウィンドウが表示されます。表1のように入力を行い①～⑧、最後に「アラームの作成」をクリックします⑨。



図2 「アラームの作成」をクリック

CHAPTER 10

処理を自動化して 運用費用を削減する

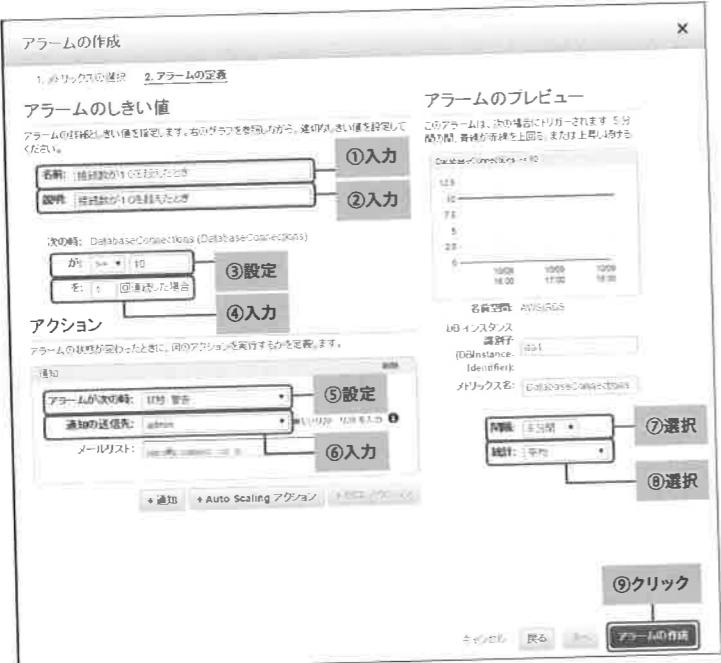


図3 アラームを設定

表1 アラームの設定

| 項目名 | 設定内容 | 備考 |
|---------------------------|----------------|---------------|
| アラームのしきい値 | | |
| ①名前 | 接続数が 10 を超えたとき | — |
| ②説明 | 接続数が 10 を超えたとき | — |
| 次の時 | | |
| ③ [DatabaseConnections] が | 10 | — |
| ④を N 回連続した場合 | 1 | — |
| アクション | | |
| ⑤アラームが次の時 | 状態:警告 | — |
| ⑥通知の送信先 | admin | SNS で作成したトピック |
| アラームのプレビュー | | |
| ⑦間隔 | 5 分間 | — |
| ⑧統計 | 平均 | — |

定期的に実行したいバックアップ処理(スナップショット)や、状態を監視し条件に一致した場合、自動実行するような設定について解説します。

- 1 EC2 のスナップショットを自動作成する
- 2 EC2 のインスタンスを自動停止する
- 3 EC2 から S3 に自動転送する

10.1 EC2のスナップショットを自動作成する

 EC2のスナップショットは残念ながら自動では作成されませんので、EC2サーバ内でcrontabを利用します。AWS CLIコマンドを定期的に動かす方法を解説します。

10-1-1 必要な環境や権限について

EC2のスナップショット(バックアップ)を自動で作成するには、アクセス権のあるIAMユーザー、CLIが利用できるような環境が必要となります。ここでは、第4章で作成したユーザー「aws-book」を使い、また、すでにAWS CLIをセットアップしている第6章で作成したMovable TypeのEC2サーバを使って説明を行います(図1)。

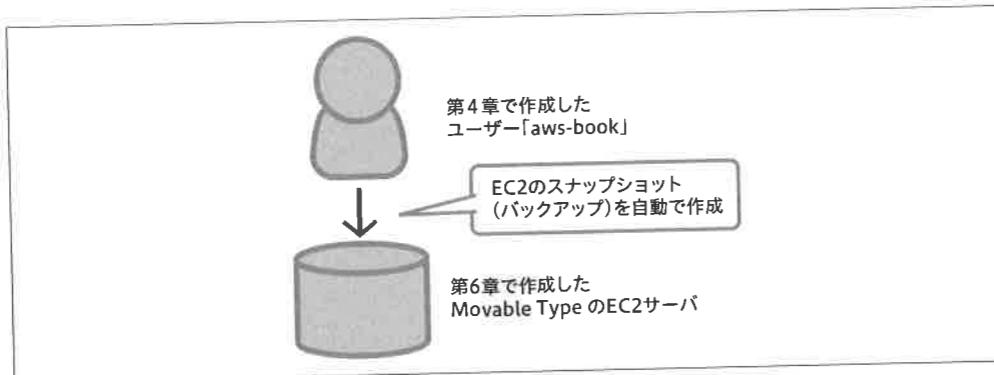


図1 利用する環境

10

10-1-2 IAMユーザーにアクセス許可を追加する

IAMユーザーにアクセス許可を追加します。

1 権限を付与したいユーザーを選択する

EC2上でスナップショットをAWS CLIから作成するためには、アクセス権限が必要です。IAMマネージメントコンソールを開き、左サイドのメニューからユーザーを選択します。登録されているユーザーが一覧されますので、権限を付与したいユーザー、ここでは、ユーザー「aws-book」をクリックします(図2)。

1 EC2のスナップショットを自動作成する

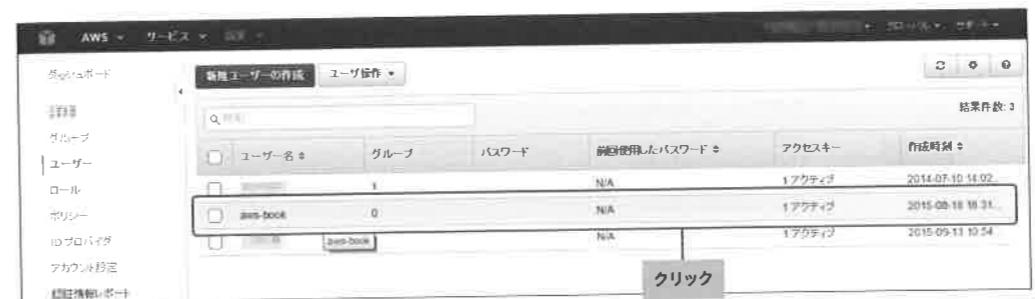


図2 IAMマネージメントコンソールからユーザーを選択

1

2 「ポリシーのアタッチ」をクリックする

「アクセス許可」(図3①)→「管理ポリシー」をクリックします②。「ポリシーのアタッチ」をクリックして③、アクセス権限を追加します。

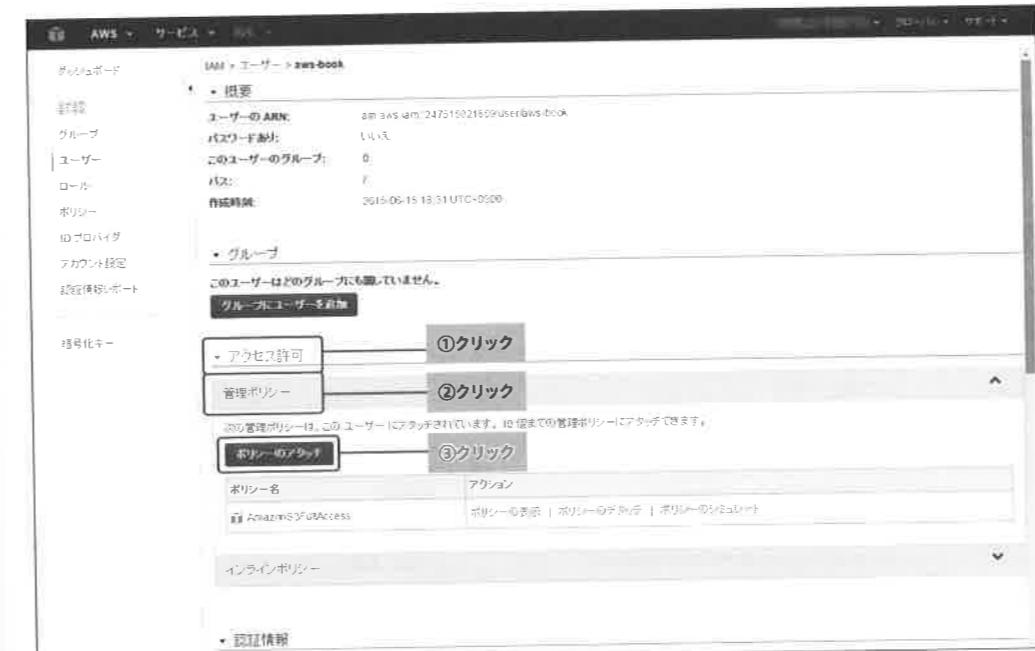


図3 「ポリシーのアタッチ」をクリック

3 「AmazonEC2FullAccess」を選択する

ポリシーの一覧が表示されますので、一覧上部の検索窓に「ec2」と入力して絞り込みます(図4①)。EC2へのアクセス権限を付与するために、その中から「AmazonEC2FullAccess」を選択します②。最後に「ポリシーのアタッチ」をクリックします③。

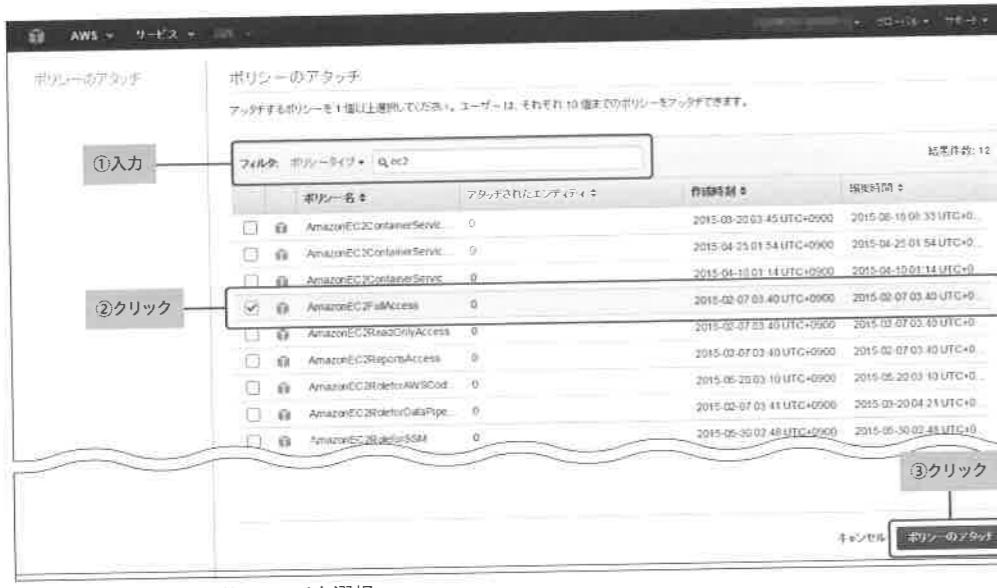


図4 「AmazonEC2FullAccess」を選択

4 追加されたポリシーを確認する

図5のようにEC2へのポリシーが追加されます。



図5 EC2へのポリシーが追加される

10-1-3 ボリュームIDを確認する

該当するボリュームIDを調べましょう。

1 該当インスタンスを確認する

スナップショットはボリュームに対して処理されますので、後述するコマンドラインでは、スナップショットをとりたいボリュームIDが必要になります。ここでは、第6章で構築したMovable TypeのEC2サーバのインスタンスより、ボリュームIDを調べます。EC2マネージメントコンソールを開き、左サイドのメニューより「インスタンス」を選びます(図6①)。該当インスタンスを選択して②、一覧下に表示される詳細情報より「説明」をクリックします③。

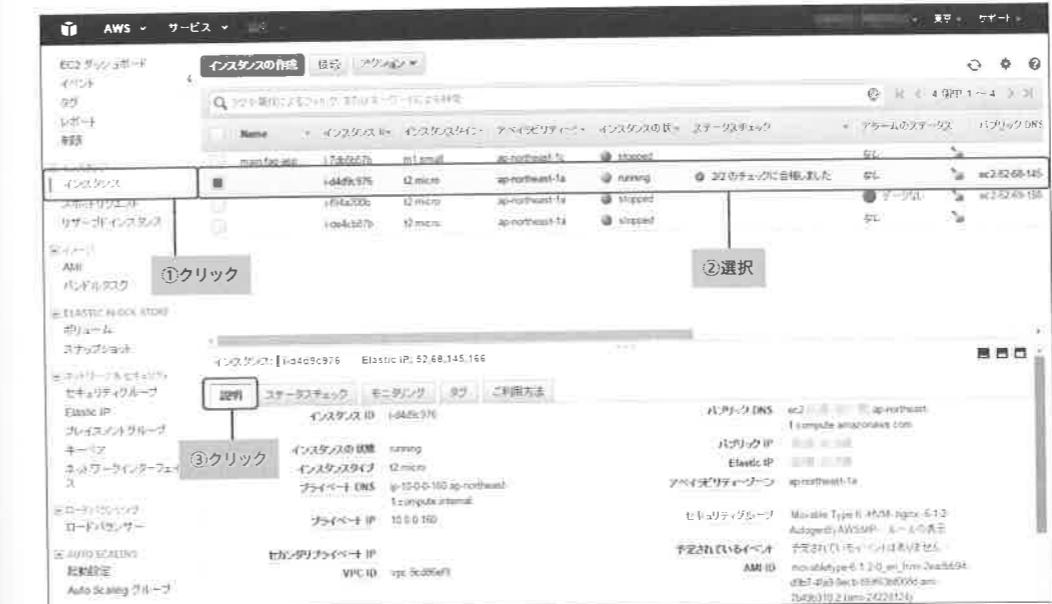


図6 インスタンスの説明

2 ボリュームIDを確認する

下にスクロールしていくと、「ルートデバイス」を見つけることができますので、その値をクリックします。ここでは「/dev/xvda」です(図7①)。ポップアップが表示され、EBS ID の値(vol-228a8fde)が、ボリュームIDとなります②。この値は後で使用しますのでメモをとるなど、覚えておいてください。



図7 ボリュームIDの確認

10-1-4 コマンドラインからスナップショットを作成する

1 EC2サーバに接続する

EC2サーバにログインして、まず、コマンドラインからスナップショットが正しく作成されるかどうか、動作の確認を行います。

ソフトウェア「TeraTerm」を起動して、ホストにEC2サーバのIPアドレス(第6章の「6-4 EC2からS3に転送する」で設定したIPアドレス)を入力し(図8①)、EC2サーバに接続します②。

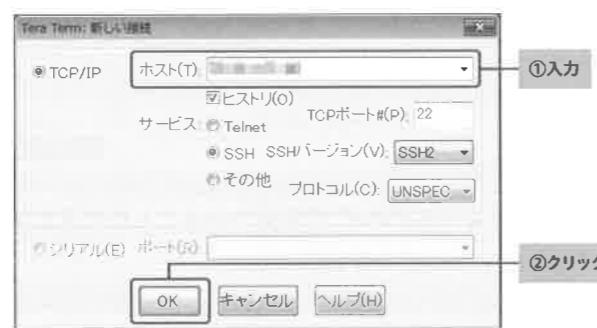


図8 EC2サーバに接続

2 EC2サーバに接続する

ユーザ名を「ec2-user」(図9①)、パスフレーズは入力せず、鍵(第5章で作成したキーペア)を指定して②、「OK」をクリックして認証を行います③。

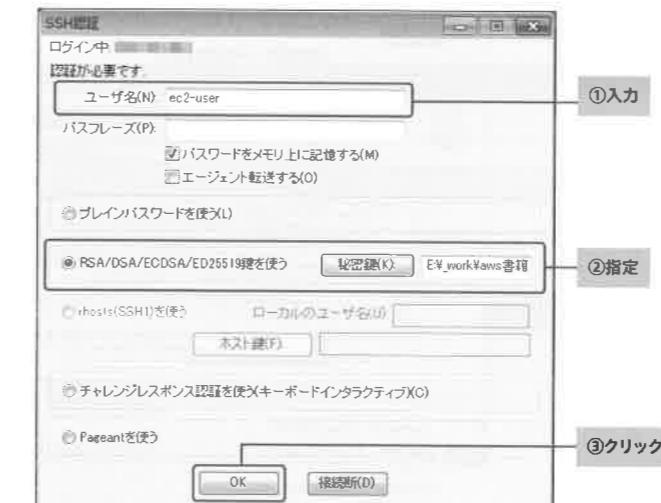


図9 SSH認証

1

EC2のスナップショットを自動作成する

3 ログイン画面を確認する

ログインできると、図10のような画面が表示されます。

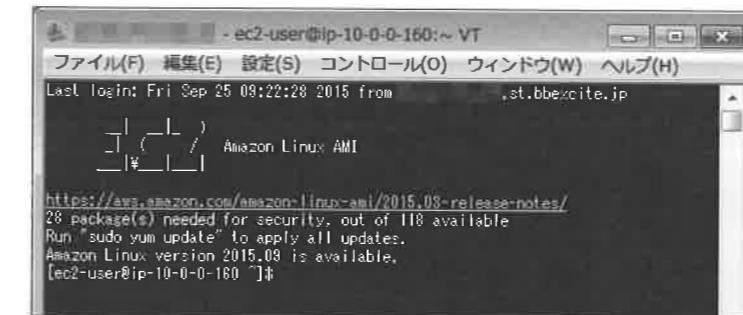


図10 ログイン後の画面

4 AWS CLIコマンドの書式を確認する

スナップショットを作成するAWS CLIコマンドの書式は構文1のとおりです。

構文1 スナップショットを作成するAWS CLIコマンド

```
aws ec2 create-snapshot --volume-id ボリュームID --description "説明文"
```

MEMO / 書式設定について

詳しい書式設定については図11のサイトで確認してください。

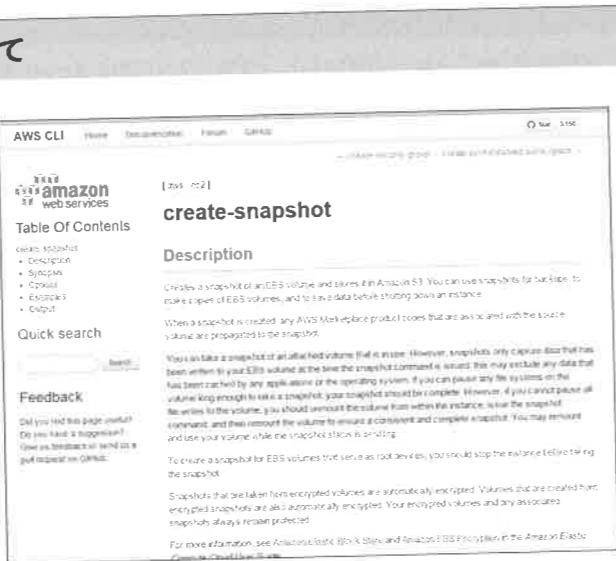


図11 create-snapshot
URL <http://docs.aws.amazon.com/cli/latest/reference/ec2/create-snapshot.html>

10

5 ボリュームIDを設定したAWS CLIコマンドの入力する

ここでは、前節で取得したボリュームIDを設定して、リスト1のようにコマンド入力します(図12①)。すると、②のような返答があります。

リスト1 AWS CLIコマンド

```
aws ec2 create-snapshot --volume-id vol-228a8fde --description "mtbackup"
```

ボリュームID

図12 スナップショットを作成するコマンドを実行

処理を自動化して運用費用を削減する

6 スナップショットの作成の開始を確認する

AWS EC2マネージメントコンソールを開き、左サイドのメニューから「スナップショット」を選び(図13①)、確認してみましょう。スナップショットの作成が開始されていることが確認できます②。

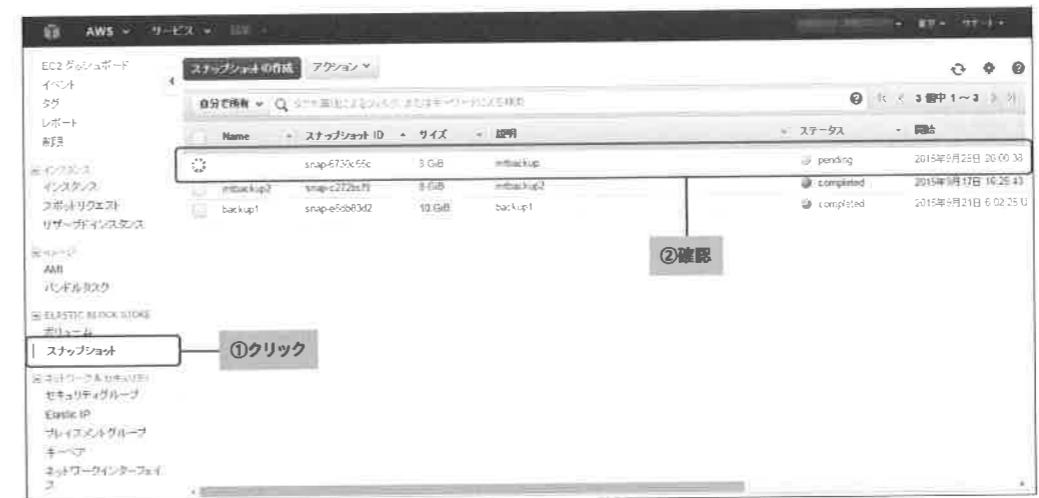


図13 スナップショット作成開始

7 スナップショットの作成の完了を確認する

ステータス「pending」が、図14のように「completed」に変われば、スナップショットの処理は完了となります。



図14 スナップショットの処理が完了

10-1-5 スクリプト、crontabを設定する

最後に、定期実行するために、シェルスクリプトを作成し、そのシェルスクリプトを定期的に起動するcrontabを設定します。

1 backup.shというファイルを作成する

入力するシェルスクリプトはリスト2のとおりです。「backup.sh」というファイル名にします(図15)。

リスト2 backup.shというファイルを作成

```
#!/bin/sh
aws ec2 create-snapshot --volume-id vol-228a8fde --description "mtbackup"
```

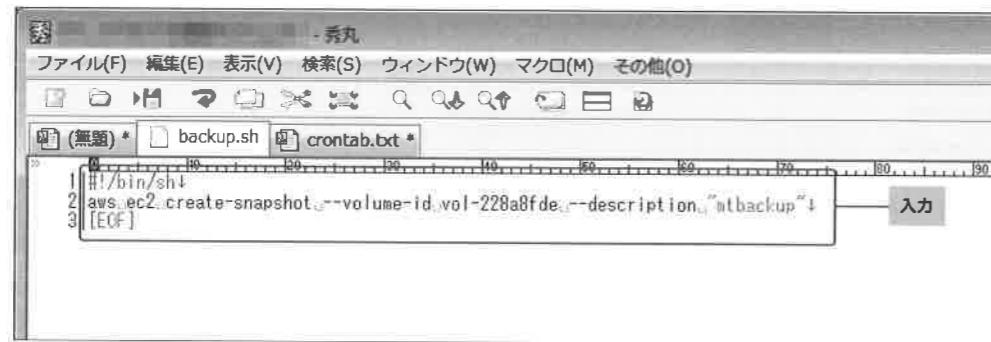


図15 シェルスクリプト「backup.sh」の作成

2 crontab.txtというファイルを作成する

リスト3のようなcrontabファイルを作成します(図16)。ファイル名は「crontab.txt」です。ファイルを保存する際に改行をしてください。改行コードをLFだけに設定します。この設定では、毎朝4時にスナップショット作成を実行します。

リスト3 crontabファイルを作成

```
0 4 * * * /bin/sh /home/ec2-user/backup.sh
```

処理を自動化して運用費用を削減する

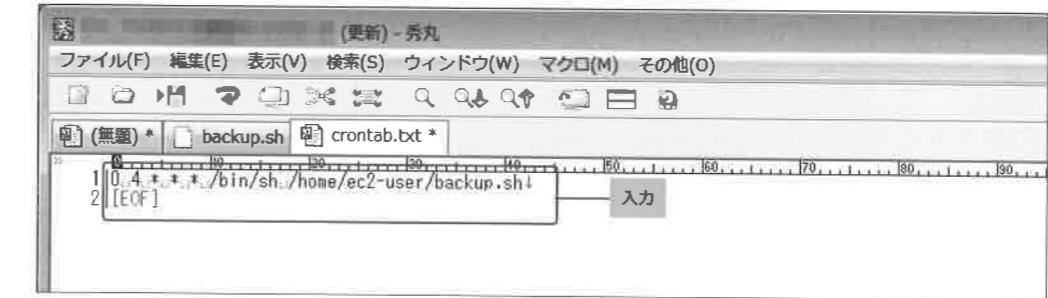


図16 「crontab」ファイルの作成

3 Cyberduckで接続する

作成した2ファイルをEC2サーバへアップロードします。ソフトウェア「Cyberduck」を起動し、「SFTP(SSHによる暗号化FTP)」にて接続を行います(図17①)。ServerにEC2サーバのIPアドレス②、ユーザ名に「ec2-user」③と入力して鍵を指定し④、「Connect」をクリックします⑤。

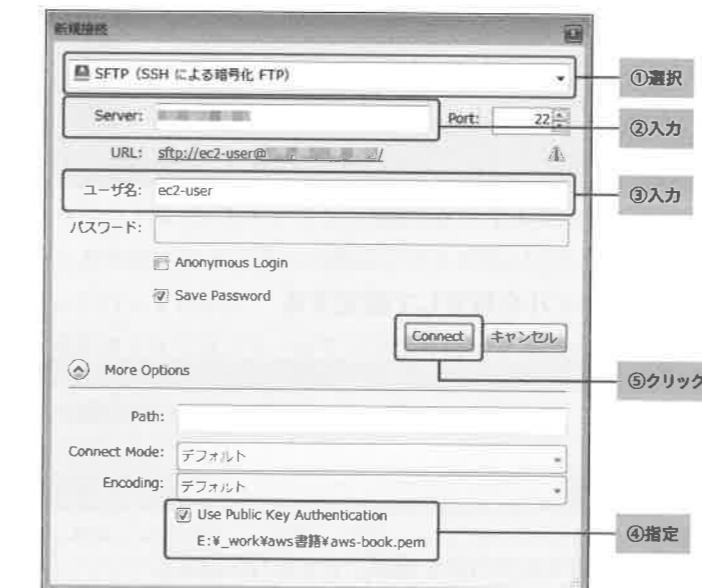


図17 SFTPにて接続

4 ファイルをアップロードする

接続後、「アップロード」をクリックして(図18①)、作成した2つのファイルを「/home/ec2-user」にアップロードします②③。

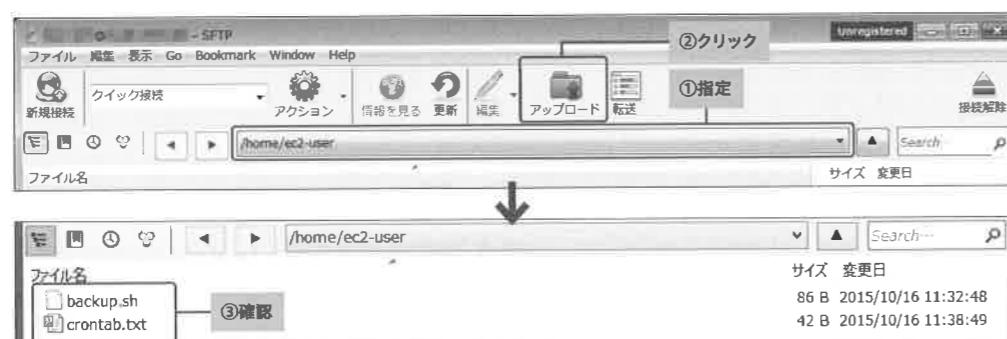


図18 Cyberduckよりファイルをアップロード

5 アップロードされたファイルを確認する

ソフトウェア「TeraTerm」に戻り、アップロードされたファイルをlsコマンドを入力して(図19①)、確認します②。

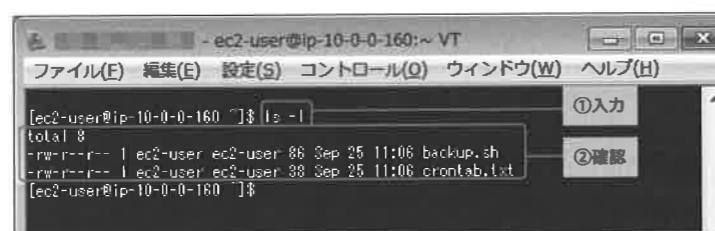


図19 アップロードされたファイルを確認

6 アップロードしたファイルを指定して設定する

最後にリスト4のようにcrontabコマンドでアップロードしたファイルを指定して設定を行います(図20①)。

リスト4 crontabコマンド

```
crontab crontab.txt
```

リスト5のオプションで設定された内容を確認できます(図20②)。

リスト5 オプション

```
crontab -l
```

図20③で表示されているcrontabの内容は、テスト確認のために4分おきに実行されるように指定しています。スナップショットの自動作成を止めるには、「crontab -r」と入力して設定を削除します。

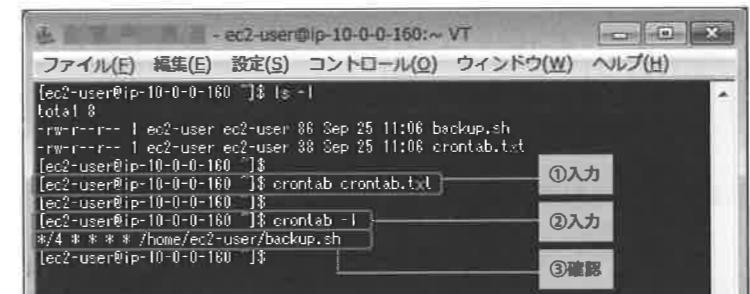


図20 crontabで設定する

注意 crontabコマンド

crontabコマンドを入力して図21のエラーが表示された場合、改行がcrontab.txtに入っていない可能性があります。

```
[ec2-user@ip-172-31-3-240 ~]$ crontab crontab.txt
"crontab.txt":1: premature EOF
```

図21 crontabコマンドのエラー例

7 EC2マネージメントコンソールから確認する

EC2マネージメントコンソールからも確認できます。EC2マネージメントコンソールを開き、左サイドのメニューから「スナップショット」をクリックすると(図22①)、4分おきにスナップショットが作成されているのが確認できます②(4分おきのスナップショットの作成は、テストとして行っている)。



図22 スナップショット自動作成結果

10.2 EC2サーバのインスタンスを自動停止する

 CloudWatchを利用してEC2サーバのインスタンスを停止する方法を解説します。

10-2-1 CPU使用率を監視して、停止を判断する

EC2サーバをテストや開発用に利用していく、使用していない時間、停止しても問題ない時間がある場合に、インスタンスを停止することにより、課金時間を減らすことができます。

CloudWatchを利用してCPU使用率を監視します。一定の時間使用率が低ければ停止するように設定します。ここでは、第6章で作成したMovable TypeのEC2サーバを使って解説します(図1)。

10

処理を自動化して運用費用を削減する

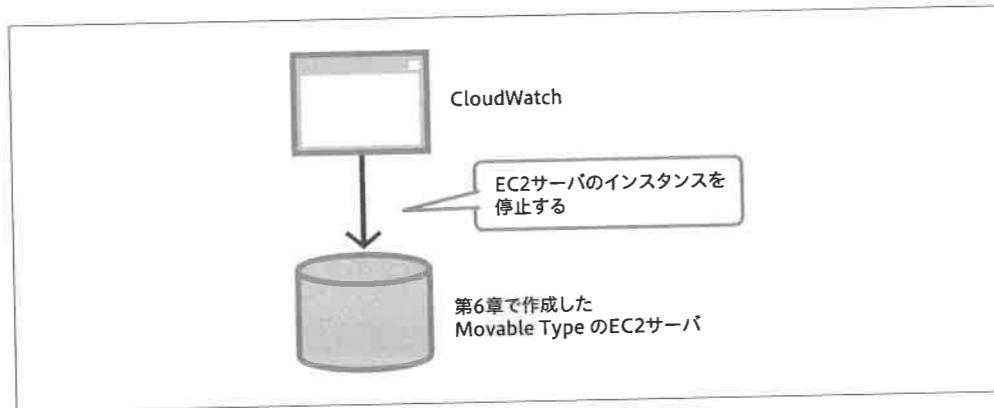


図1 利用する環境

1 EC2インスタンスを選択する

EC2マネージメントコンソールを開き、左サイドのメニューから「インスタンス」を選択します(図2①)。設定したいインスタンスを選択して②、「説明」をクリックして③、インスタンスの詳細情報を一覧で表示します④。

2 EC2サーバのインスタンスを自動停止する

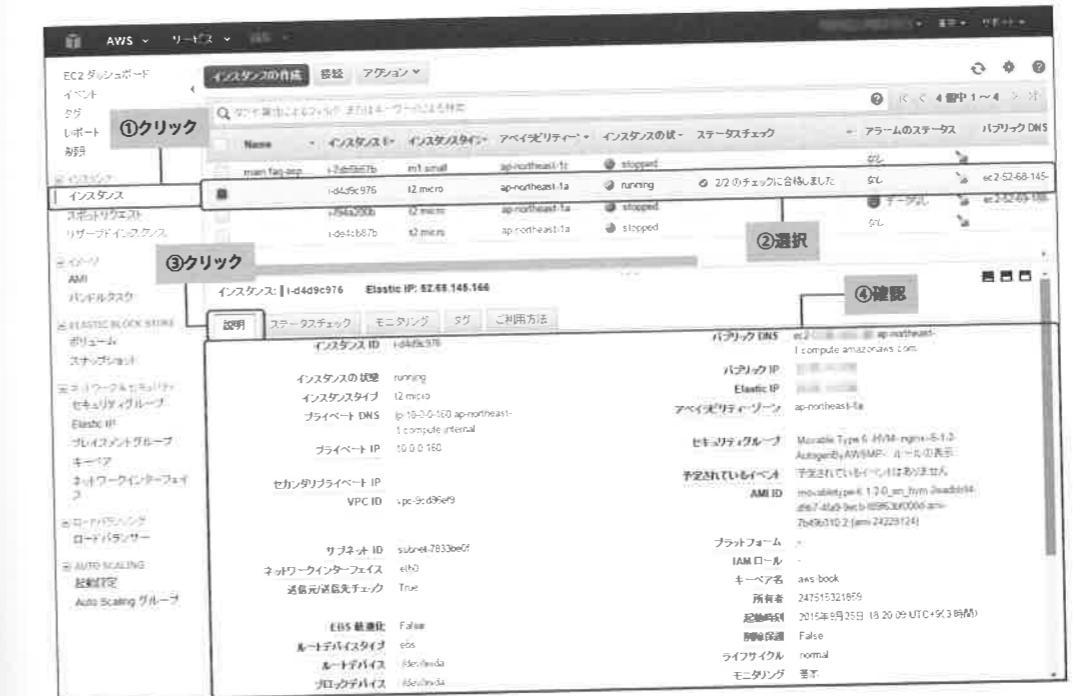


図2 EC2インスタンスを選択

2 「アラームの作成」をクリックする

インスタンスを自動停止させるために、CloudWatchのアラームに監視をさせます。「モニタリング」をクリックして(図3①)、「アラームの作成」をクリックします②。



図3 アラームの作成

2

EC2サーバのインスタンスを自動停止する

3 アラームを作成する

ここでは、CPU使用率の平均が30分間連続で1%を下回っていた場合、指定先に通知を行い、インスタンスを停止させる設定を行います。表1のように入力を行い(図4①~⑥)、最後に「アラームの作成」をクリックします⑦。「IAM ロールの作成EC2ActionAccess…」というメッセージが表示されるのでチェックを入れて、再度「アラームの作成」をクリックします⑧。

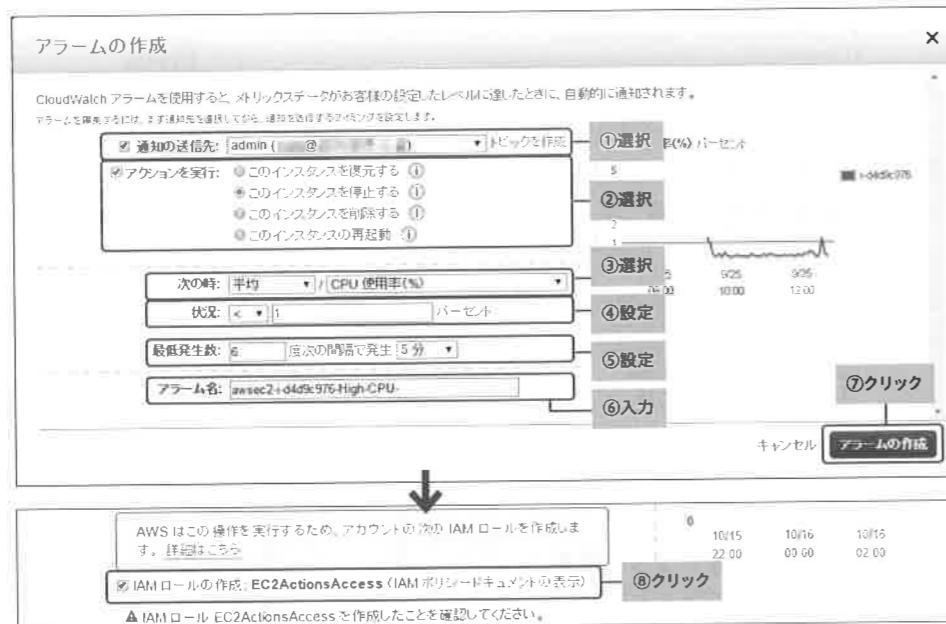


図4 アラーム設定情報の入力

処理を自動化して運用費用を削減する

表1 アラームの設定

| 項目名 | 設定内容 | 備考 |
|-----------|----------------------------|-----------------|
| ①通知の送信先 | チェックを入れる。 admin | 第9章で作成したトピックを指定 |
| ②アクションを実行 | チェックを入れる。 このインスタンスを停止する | — |
| ③次の時 | 平均 CPU 使用率(%) | — |
| ④状況 | < 1 | — |
| ⑤最低発生数 | 6度 次の間隔で発生:5分 | — |
| ⑥アラーム名 | 任意 | — |

MEMO / 条件設定について

ここでは、CPU使用率1%以下に設定しましたが、この値は実稼働されているサーバ環境をモニターして、適切な値に調整してください。

4 アラームへのリンクをクリックする

アラームが作成されると、アラームへのリンクが表示されます。そのリンクをクリックします(図5)。



図5 アラームのリンクを確認

5 アラームの表示を確認する

別ウィンドウにて、SNSマネジメントコンソールのアラームが表示されます。図6の状態は緑色で「OK」と表示されているのがわかります。



図6 アラーム状態は「OK」になっている

6 アラームが発生するか確認する

設定した条件(ここではCPU使用率の平均が30分間連続で1%を下回っていた場合)になつたとき、アラームの状態は赤色で「アラーム」と表示されます(図7)。なお状況によってすぐにアラームになることもあります。



図7 状態がアラームに変化

7 状態変化を知らせるメールを確認する

同時に図8のような状態が変化したことを伝えるメールも届きます。なお、アラーム設定時に、しきい値条件を満たしている場合には、一度インスタンスを再起動してからテストを行ってください。

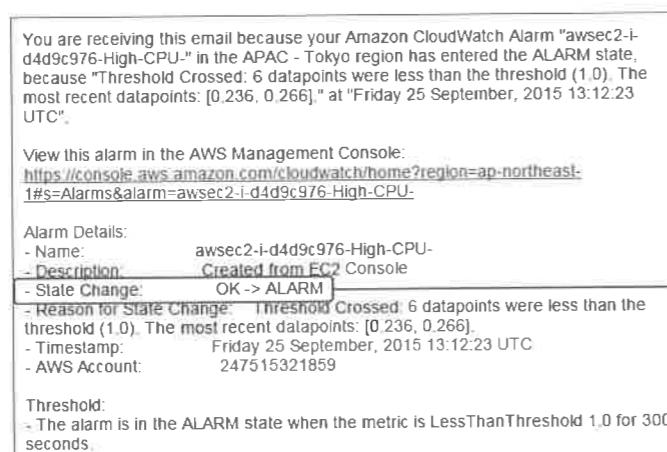


図8 アラームによるメール(状態：「OK」→「ALARM」)

8 EC2マネージメントコンソールで確認する

EC2マネージメントコンソールを開き、左サイドのメニューから「インスタンス」をクリックして(図9①)、一覧より、該当インスタンスが停止状態になっていることが確認できます②。



図9 インスタンスが自動停止



10 3 EC2からS3に自動転送する

ここでは、EC2からS3にcrontabを使って、定期的に自動転送する設定について解説します。

10-3-1 EC2からS3への自動転送するように設定する

第6章では、AWS CLIによるコマンドを入力して、EC2上に作成したファイルをS3に転送しましたが、毎回この操作をするのは面倒です。そこで、この処理を5分に1回自動的に実行させるようにします。自動処理設定はcrontabで、転送処理はシェルスクリプトで記述します。

10-3-2 転送するファイルを準備する

1 シェルスクリプトを作成する

第6章で使用したS3との同期をとるコマンドを使って、シェルスクリプトを「copy.sh」というファイル名で作成します(リスト1)。

リスト1 S3との同期をとるコマンド

```
#!/bin/sh
/usr/bin/aws s3 sync /data/file/static s3://www.aws-book.com/
```

2 crontab用のファイルを作成する

5分おきにcopy.shを実行するように、crontabを記述します。ファイル名は「crontab10.txt」です(リスト2)。

リスト2 5分おきにcopy.shを実行するコマンド

```
*/5 * * * * /bin/sh /home/ec2-user/copy.sh
```

10-3-3 ファイルをアップロードする

作成した2つのファイルをEC2サーバへアップロードします。

1 SFTPで接続する

ソフトウェア「Cyberduck」を起動し、「SFTP(SSHによる暗号化FTP)」で接続を行います(図17①)。ServerにEC2サーバのIPアドレス②、ユーザー名に「ec2-user」③と入力して鍵を指定し④、「Connect」をクリックします⑤。

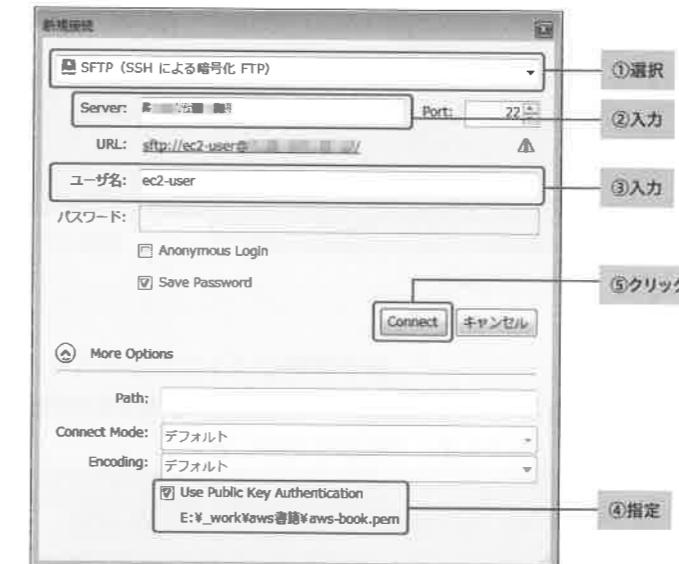


図1 SFTPにて接続

2 作成した2つのファイルをアップロードする

接続後、/home/ec2-userを指定して(図2①)、「アップロード」をクリックし②、作成した2つのファイルをアップロードします③。なお「10-1 EC2のスナップショットを自動作成する」でアップロードしたファイルはここでは事前に削除しています。

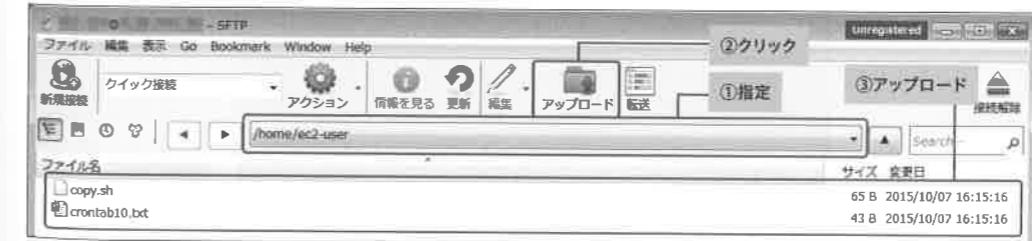


図2 Cyberduckよりファイルをアップロード

10-3-4 アップロードしたファイルを確認する

EC2にSSH接続して、crontabコマンドを使い、アップロードしたファイルを確認します。

1 EC2にSSH接続を行う

ソフトウェア「TeraTerm」を起動して(図3①)、「OK」をクリックし②、EC2サーバに接続します。

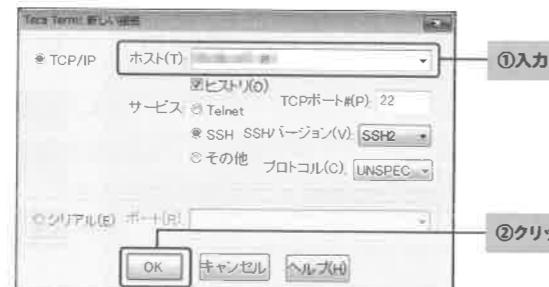


図3 EC2サーバに接続

2 SSH認証を行う

ユーザ名を「ec2-user」として(図4①)、パスワードはなしで、鍵(第5章で作成したキーペア)を指定して②③、「OK」をクリックし④、SSH認証を行います。

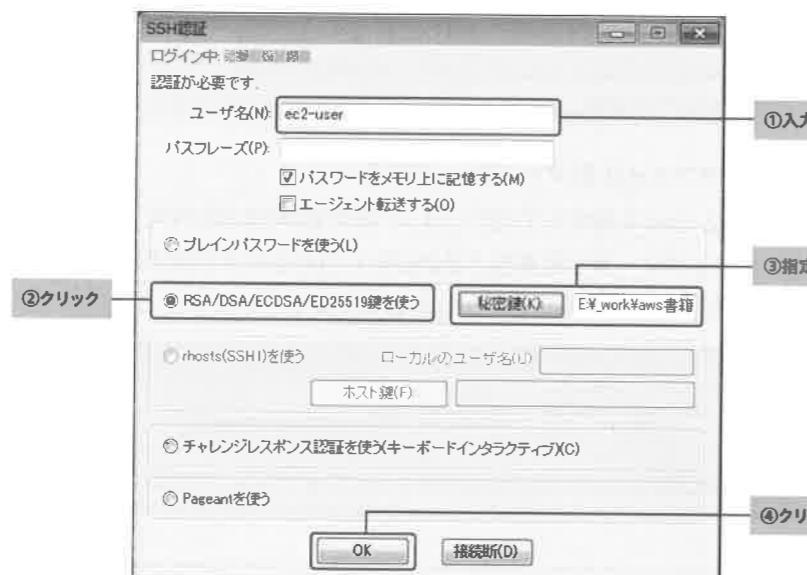


図4 SSH認証を行う

3 ログイン画面を確認する

ログインできると、図5のような画面が表示されます。

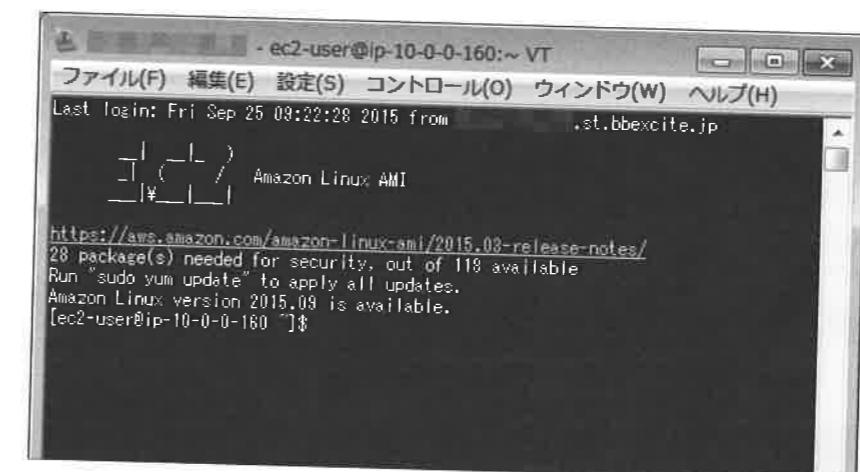


図5 ログイン画面

4 アップロードされたファイルを確認する

アップロードされたファイルをlsコマンドで確認します(図6①②)。



図6 アップロードされたファイルを確認

5 crontabコマンドで設定内容を確認する

リスト1のcrontabコマンドを入力します。

リスト1 crontabコマンド

```
crontab crontab10.txt
```

リスト2のオプションで設定された内容を確認できます(図7)。

リスト2 オプション

```
crontab -l
```

```
- ec2-user@ip-10-0-0-160:~ VT
[ec2-user@ip-10-0-0-160 ~]$ crontab crontab10.txt
[ec2-user@ip-10-0-0-160 ~]$ crontab -l
* * * * * /bin/sh /home/ec2-user/copy.sh
[ec2-user@ip-10-0-0-160 ~]$
```

図7 crontabコマンドを入力する

10-3-5 ファイルの転送を確認する

Movable Typeにログインして転送したファイルを確認します。

1 Movable Typeにログインする

次のURLにアクセスして、Movable Typeにログインします。

- ・アクセスするサイト

URL <http://mt.aws-book.com/mt/admin>

2 記事を追加して更新する

該当するWebサイトの「記事を作成」をクリックして、Webサイトに記事を追加して(図8)①、「公開」をクリックします②。



図8 記事を追加する

3 ページを確認する

まだ反映されていない状態のページを確認しておきます(図9)。

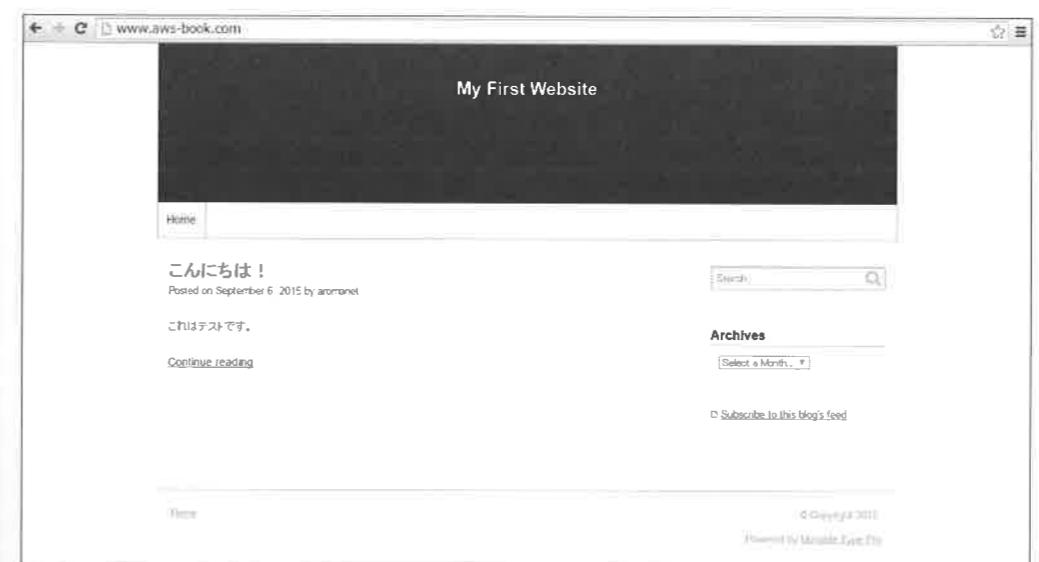


図9 記事を追加する

4 再度アクセスして転送されていることを確認する

5分以上経過して再度アクセス(リロード)して、転送されていることを確認します(図10)。



図10 転送されていることを確認する



負荷対策を行い 機会損失を減らす

インスタンスのスケールアップ、CloudFrontを使った分散配信、ロードバランサー活用といった負荷対策方法について解説します。

- 1 負荷対策を検討する
- 2 インスタンスをスケールアップする
- 3 CloudFrontを利用する
- 4 ELBでトラフィックを分散する



11.1 負荷対策を検討する

↑ 負荷対策を行うにあたって検討すべきこと、モニタリングの活用について解説します。

11-1-1 問題点を把握する

例えば、Webサイトにアクセスして、「何やらレスポンス(表示されるのが遅いな)と感じる場合、きっとどこかにボトルネックがあるはずです。その部分にかかっている負荷を取り除いてあげれば、この問題は解決します。まず「問題点は何か」を検討していくことから始めましょう。

しかし、そのような時間的な余裕があまりないケースもあるかと思いますので、とりあえず、問題解決優先でインスタンスのクラスをスケールアップしておいて、後からゆっくりとログを調べてもよいでしょう。

11

11-1-2 モニタリングとは

負荷対策を行い機会損失を減らす

EC2インスタンスの基本モニタリング情報は5分おきに2週間分のデータを無料で記録しています。この情報を分析して、問題点はCPUへの負荷なのか、メモリ不足なのか、ディスクのI/Oなのか、ネットワークのパフォーマンスなのかを判断しましょう。

詳細モニタリング(図1)では、追加料金が必要となります。より詳しい情報を取得することができます。問題分析期間中だけでも、詳細モニタリングオプションを有効にして、より詳細な情報を取得し、的確なジャッジができるようにしたいものです。

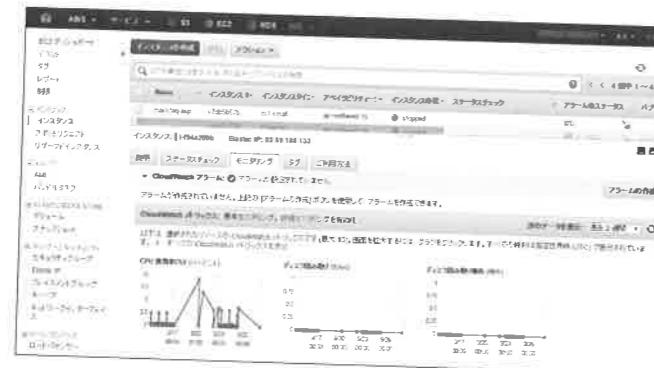


図1 モニタリングのサンプル



11.2 インスタンスをスケールアップする

↑ EC2やRDSの処理能力をアップさせるために最も簡単な方法であるスケールアップ(インスタンスタイプの変更)について解説します。

2

インスタンスをスケールアップする

POINT / 料金について

インスタンスタイプによって料金が異なります。必要以上のリソースを選択せず、少しづつ様子を見ながらスケールアップするといいでしょう。タイプの変更は何度でもできます。

11-2-1 EC2のインスタンスをスケールアップする

EC2は図1のような「一般的な目的」(通常利用の意味)インスタンスクラスが用意されていますので、現在利用しているタイプより上位のものがあれば簡単にスケールアップすることができます。

これ以外に、「コンピューティング最適化」、「GPUインスタンス」、「メモリの最適化」、「ストレージの最適化」のタイプもありますので、目的に応じて選定してください。

| Amazon ウェブ サービス 監視サービス | | | | | |
|-----------------------------------|--|--|--|--|--|
| Amazon EC2 | | | | | |
| 数値の詳細 | | | | | |
| インスタンス | | | | | |
| リージョン | | | | | |
| AWS Management Portal for vCenter | | | | | |
| AWS Lambda | | | | | |
| Amazon EC2 サービスレベルアグリメント | | | | | |
| Amazon EC2 リザーブドインスタンス | | | | | |
| Amazon EC2 ハードウェア導入インスタンス | | | | | |
| Amazon EC2 スポットインスタンス | | | | | |
| Amazon EC2 Windows インスタンス | | | | | |
| Amazon EC2 ワークマシン | | | | | |
| Amazon EC2 リザーブドインスタンス | | | | | |
| Amazon EC2 ハードウェア導入インスタンス | | | | | |
| Amazon EC2 スポットインスタンス | | | | | |
| Amazon EC2 Windows インスタンス | | | | | |
| Amazon EC2 ワークマシン | | | | | |
| Amazon EC2 リザーブドインスタンス | | | | | |
| Amazon EC2 ハードウェア導入インスタンス | | | | | |
| Amazon EC2 スポットインスタンス | | | | | |
| Amazon EC2 Windows インスタンス | | | | | |
| Amazon EC2 ワークマシン | | | | | |
| Amazon EC2 リザーブドインスタンス | | | | | |
| Amazon EC2 ハードウェア導入インスタンス | | | | | |
| Amazon EC2 スポットインスタンス | | | | | |
| Amazon EC2 Windows インスタンス | | | | | |
| Amazon EC2 ワークマシン | | | | | |
| Amazon EC2 リザーブドインスタンス | | | | | |
| Amazon EC2 ハードウェア導入インスタンス | | | | | |
| Amazon EC2 スポットインスタンス | | | | | |
| Amazon EC2 Windows インスタンス | | | | | |
| Amazon EC2 ワークマシン | | | | | |
| Amazon EC2 リザーブドインスタンス | | | | | |
| Amazon EC2 ハードウェア導入インスタンス | | | | | |
| Amazon EC2 スポットインスタンス | | | | | |
| Amazon EC2 Windows インスタンス | | | | | |
| Amazon EC2 ワークマシン | | | | | |
| Amazon EC2 リザーブドインスタンス | | | | | |
| Amazon EC2 ハードウェア導入インスタンス | | | | | |
| Amazon EC2 スポットインスタンス | | | | | |
| Amazon EC2 Windows インスタンス | | | | | |
| Amazon EC2 ワークマシン | | | | | |
| Amazon EC2 リザーブドインスタンス | | | | | |
| Amazon EC2 ハードウェア導入インスタンス | | | | | |
| Amazon EC2 スポットインスタンス | | | | | |
| Amazon EC2 Windows インスタンス | | | | | |
| Amazon EC2 ワークマシン | | | | | |
| Amazon EC2 リザーブドインスタンス | | | | | |
| Amazon EC2 ハードウェア導入インスタンス | | | | | |
| Amazon EC2 スポットインスタンス | | | | | |
| Amazon EC2 Windows インスタンス | | | | | |
| Amazon EC2 ワークマシン | | | | | |
| Amazon EC2 リザーブドインスタンス | | | | | |
| Amazon EC2 ハードウェア導入インスタンス | | | | | |
| Amazon EC2 スポットインスタンス | | | | | |
| Amazon EC2 Windows インスタンス | | | | | |
| Amazon EC2 ワークマシン | | | | | |
| Amazon EC2 リザーブドインスタンス | | | | | |
| Amazon EC2 ハードウェア導入インスタンス | | | | | |
| Amazon EC2 スポットインスタンス | | | | | |
| Amazon EC2 Windows インスタンス | | | | | |
| Amazon EC2 ワークマシン | | | | | |
| Amazon EC2 リザーブドインスタンス | | | | | |
| Amazon EC2 ハードウェア導入インスタンス | | | | | |
| Amazon EC2 スポットインスタンス | | | | | |
| Amazon EC2 Windows インスタンス | | | | | |
| Amazon EC2 ワークマシン | | | | | |
| Amazon EC2 リザーブドインスタンス | | | | | |
| Amazon EC2 ハードウェア導入インスタンス | | | | | |
| Amazon EC2 スポットインスタンス | | | | | |
| Amazon EC2 Windows インスタンス | | | | | |
| Amazon EC2 ワークマシン | | | | | |
| Amazon EC2 リザーブドインスタンス | | | | | |
| Amazon EC2 ハードウェア導入インスタンス | | | | | |
| Amazon EC2 スポットインスタンス | | | | | |
| Amazon EC2 Windows インスタンス | | | | | |
| Amazon EC2 ワークマシン | | | | | |
| Amazon EC2 リザーブドインスタンス | | | | | |
| Amazon EC2 ハードウェア導入インスタンス | | | | | |
| Amazon EC2 スポットインスタンス | | | | | |
| Amazon EC2 Windows インスタンス | | | | | |
| Amazon EC2 ワークマシン | | | | | |
| Amazon EC2 リザーブドインスタンス | | | | | |
| Amazon EC2 ハードウェア導入インスタンス | | | | | |
| Amazon EC2 スポットインスタンス | | | | | |
| Amazon EC2 Windows インスタンス | | | | | |
| | | | | | |

ここでは、表1のように仮想CPU数、メモリ量を2倍に変更する手順を解説します。

表1 仮想CPU数、メモリ量を2倍に変更

| | タイプ | CPU | メモリ | 料金(1時間) |
|-----|-----------|-----|-----|---------|
| 変更前 | t2.micro | 1 | 1GB | \$0.02 |
| 変更後 | t2.medium | 2 | 4GB | \$0.08 |

※料金は東京リージョンのオンデマンドインスタンス(2015年9月時点)

1 EC2インスタンス一覧を表示する

EC2マネジメントコンソールを開き、左サイドのメニューから「インスタンス」を選びます(図2①)。スケールアップしたいEC2インスタンスを選択します②。選択したインスタンスのインスタンスタイプが「t2.micro」になっていることが確認できます。

MEMO / インスタンスについて

インスタンスタイプの変更を行うためには、いったんEC2インスタンスを停止する必要があります。起動中は選択することができません。

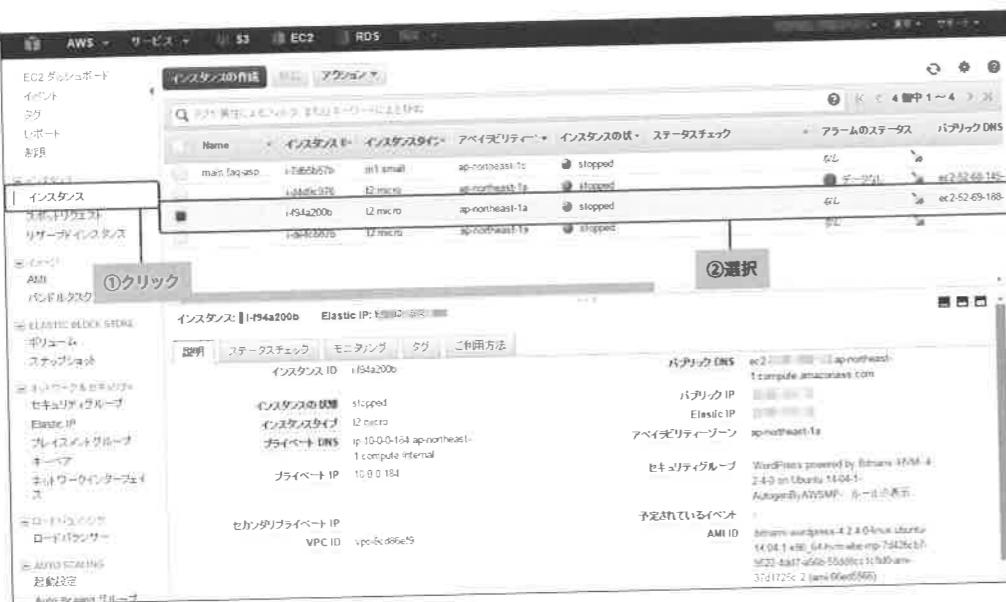


図2 EC2インスタンスの一覧

2 「インスタンスタイプの変更」を選択する

選択されたインスタンスのタイプを変更するには、一覧上にある「アクション」をクリックして(図3①)、「インスタンスの設定」②→「インスタンスタイプの変更」③を選択します。



図3 インスタンスタイプの変更

3 インスタンスタイプを選択する

ポップアップ画面が表示され、インスタンスタイプを自由に選ぶことができます。ここでは、「t2.medium」を選択し(図4①)、「適用」をクリックします②。

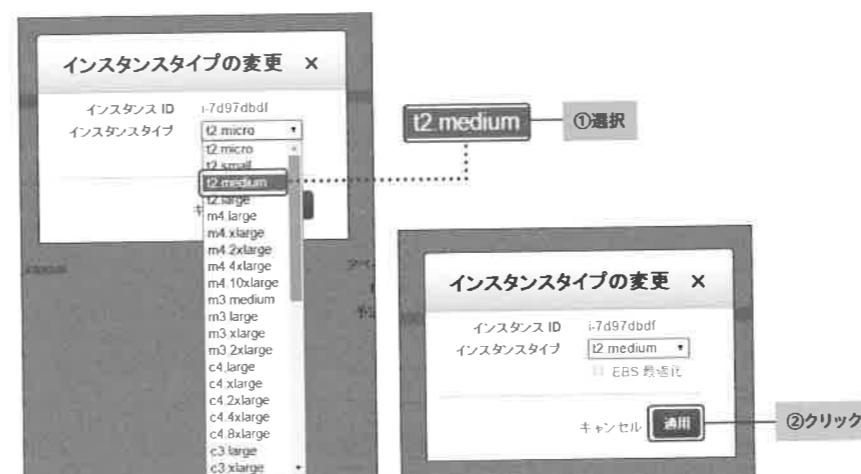


図4 インスタンスタイプの選択

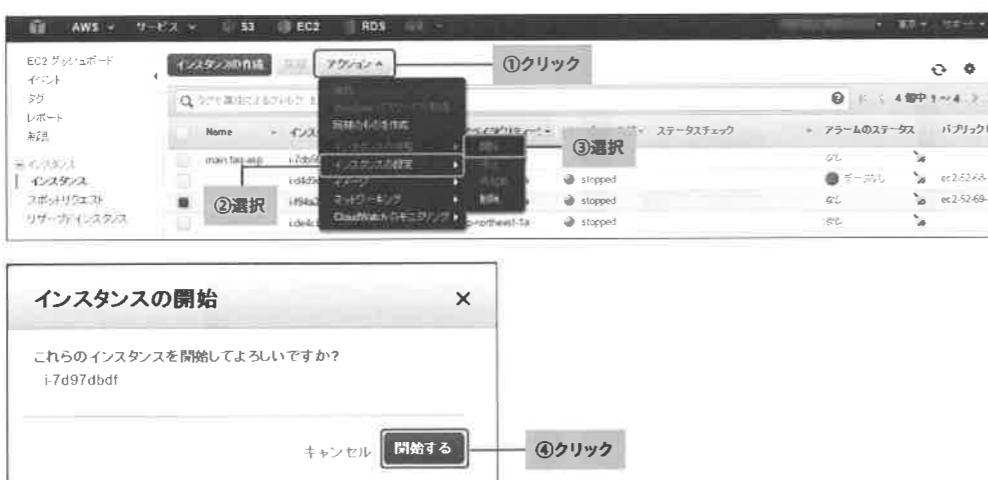
4 インスタンスタイプが変更されたことを確認する

インスタンスの一覧に戻ると、該当インスタンスのインスタンスタイプが、「t2.medium」に変更されたことが確認できます(図5)。



5 インスタンスを開始する

これだけでスケールアップの作業は終了です。最後にインスタンスを開始して問題なく起動されることを確認しておきましょう。「アクション」をクリックして①、「インスタンスの状態」②→「開始」③を選択します。「開始する」をクリックします④。



11-2-2 RDS のインスタンスをスケールアップする

RDSは図7のようなインスタンスクラスが用意されていますので、現在利用しているタイプより上位のものがあれば簡単にスケールアップすることができます。

| インスタンスタイプ | VCPU | メモリ (GiB) | PIOPS 用に最適化 | ネットワークパフォーマンス |
|---|------|-----------|-------------|---------------|
| db.t2.medium | 1 | 2.75 | — | 中 |
| db.t2.large | 2 | 7.5 | — | 中 |
| db.t2.xlarge | 4 | 15 | はい | 高 |
| db.t2.2xlarge | 8 | 30 | はい | 高 |
| Amazon RDS サービスレベルマグリット | | | | |
| 標準 | | | | |
| Amazon RDS for Aurora | | | | |
| Amazon RDS for MySQL | | | | |
| Amazon RDS for PostgreSQL | | | | |
| Amazon RDS for Oracle | | | | |
| Amazon RDS for SQL Server | | | | |
| マネジメントコンソール | | | | |
| リリースノート | | | | |
| コミュニティフォーラム | | | | |
| ドキュメント | | | | |
| また、AWS アカウントを作成しない限り、1 年間の無料料金をご利用いただけます。 | | | | |
| まずは無料で始める | | | | |
| Amazon MySQL Oracle SQL Server PostgreSQL | | | | |
| オンデマンドのインスタンス料金 | | | | |

図7 RDSインスタンスクラスの一覧

ここでは、表2のように仮想CPU数、メモリ量を2倍に変更する手順を解説します。

表2 仮想CPU数、メモリ量を2倍に変更

| | クラス | CPU | メモリ | 料金(1時間) |
|-----|--------------|-----|-----|---------|
| 変更前 | db.t2.micro | 1 | 1GB | \$0.02 |
| 変更後 | db.t2.medium | 2 | 4GB | \$0.08 |

※料金は東京リージョンのオンデマンドインスタンス(2015年9月現在)

1 RDSインスタンス一覧を表示する

RDSマネージメントコンソールを開き、左サイドのメニューから「インスタンス」を選びます(図8①)。スケールアップしたいRDSインスタンスを選択します②。

選択したインスタンスのインスタンスタイプが「db.t2.micro」になっていることが確認できます③。

注意 変更中はデータベースにアクセスできなくなる

RDSインスタンスは、EC2インスタンスのように停止することができませんので、その必要はありませんが、ステータスが「変更中」の間は、データベースを利用することができますので、運用していないメンテナンス時間などに行うようにしてください。



図8 RDSインスタンスの一覧

2 「変更」を選択する

一覧上部の「インスタンスの操作」をクリックして(図9①)、「変更」を選びます②。



図9 インスタンスの操作：変更

3 DBインスタンスのクラスを変更する

DBインスタンスのクラスを「db.t2.micro」から「db.t2.medium」に変更します(図10)。

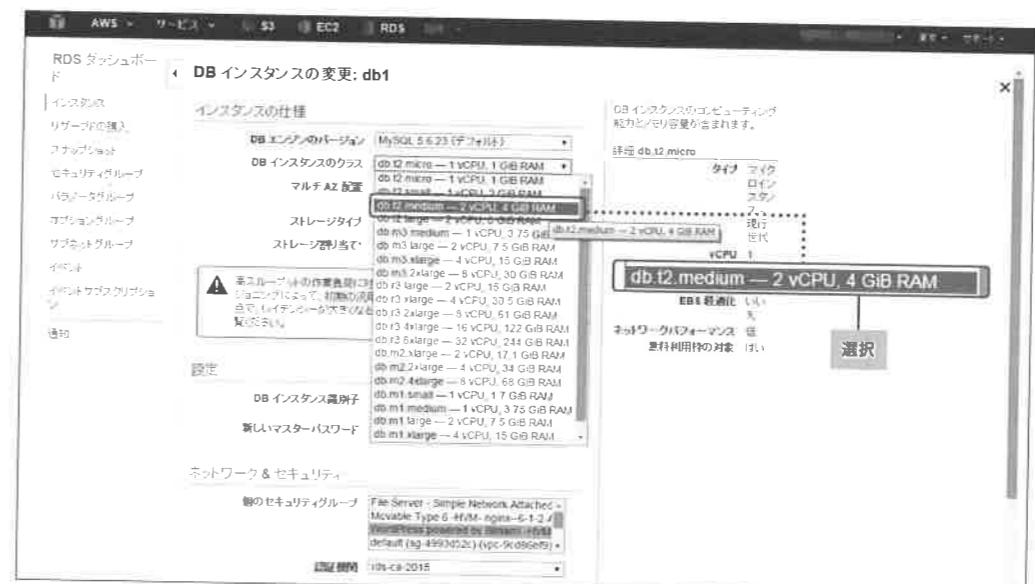


図10 DBインスタンスのクラスを変更する

4 DBインスタンスのクラスの変更を適用する

下までスクロールして、「すぐに適用」にチェックを入れ(図11①)、「次へ」をクリックします②。

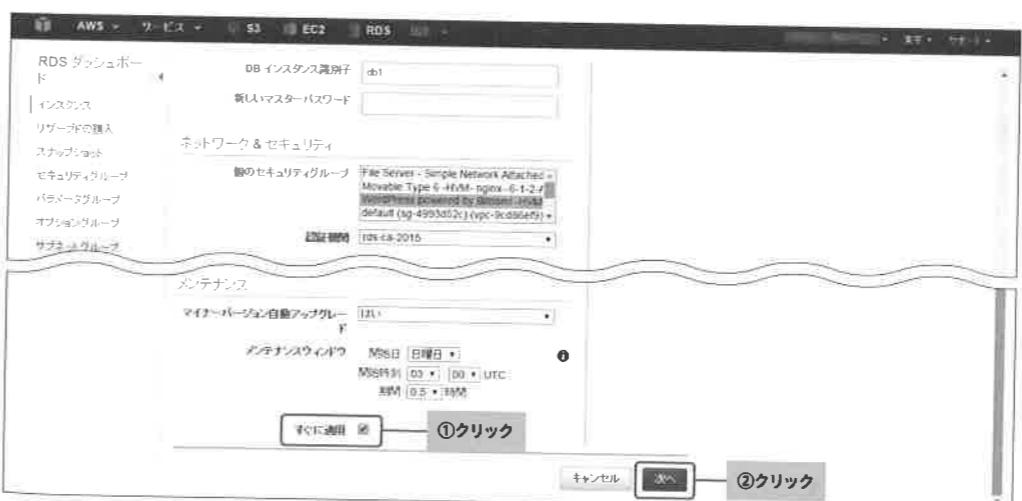


図11 もう一度適用をクリックする

5 DBインスタンスのクラスの変更を適用する

変更内容が表示されるので確認して(図12①)、「DBインスタンスの変更」をクリックします②。

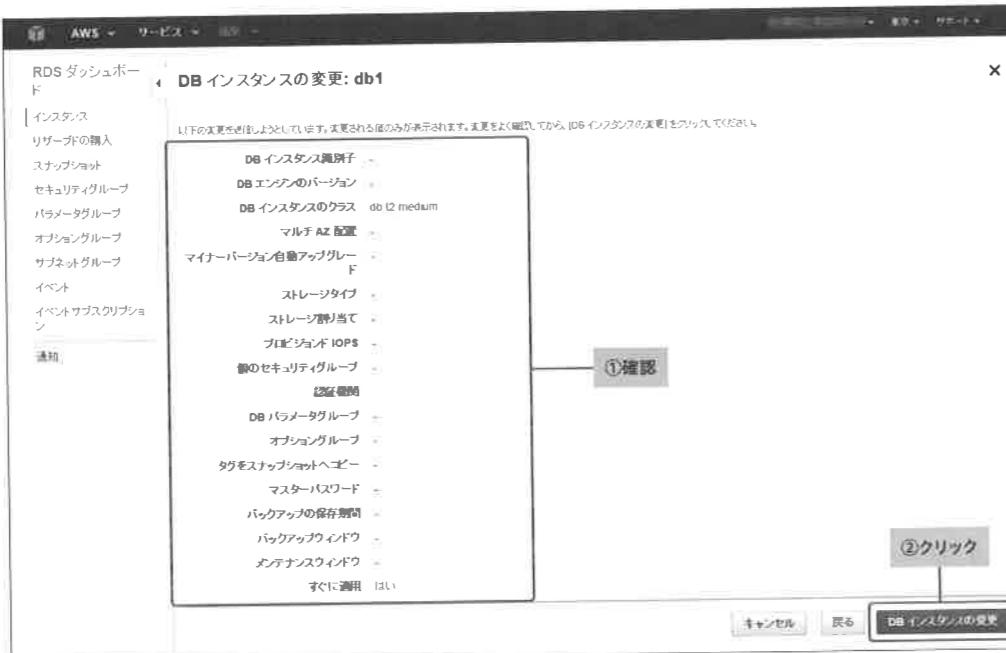


図12 変更の確認

6 変更処理の終了を確認する

RDSインスタンスの一覧に戻ります。ステータスは「変更中」になっています(図13)。

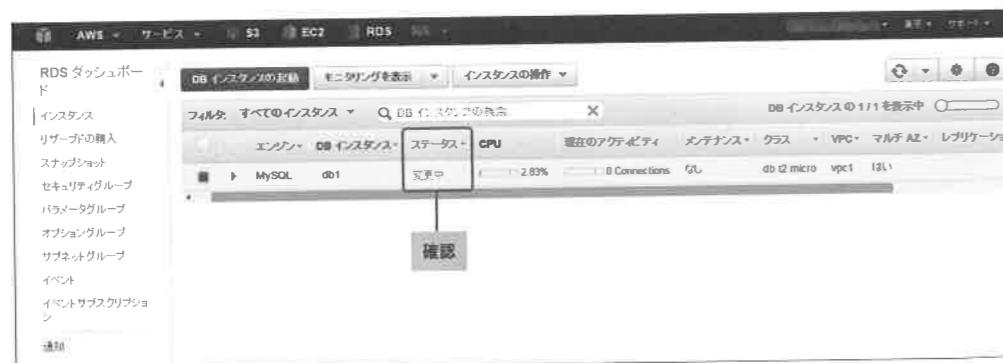


図13 ステータスは「変更中」になっている

7 ステータスが「利用可能」に変わることを確認する

しばらくすると(時間が結構かかる場合もある)、ステータスは「利用可能」に変わり(図14)、データベースにアクセスできるようになります。



図14 データベースが利用可能になる

11 3 CloudFrontを利用する

 CloudFrontを利用したコンテンツ配信のしくみを利用して、エッジロケーションにキャッシングさせることにより、Webサーバへの負担を軽減する方法を解説します(図1)。

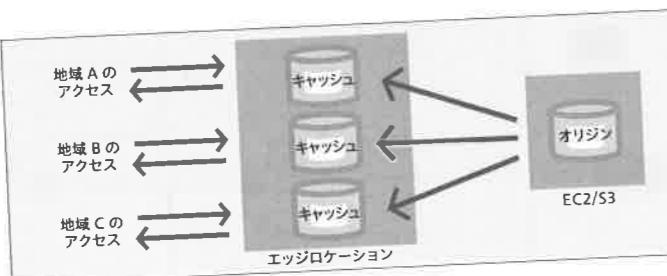


図1 CloudFrontのキャッシングによる負荷分散

1 CloudFrontマネージメントコンソールを開く

CloudFrontマネージメントコンソールを開き、「Create Distribution」をクリックします(図2)。

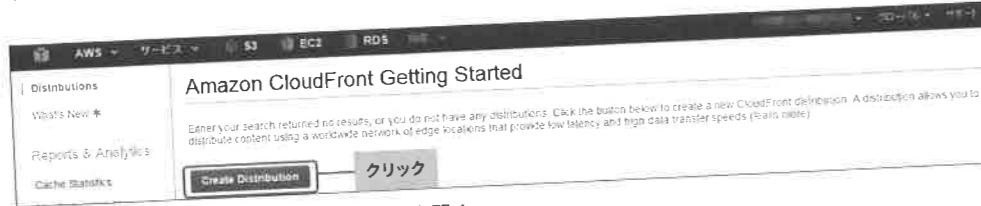


図2 CloudFrontマネージメントコンソールを開く

2 「Get started」をクリックする

Webディストリビューションの「Get started」をクリックします(図3)。

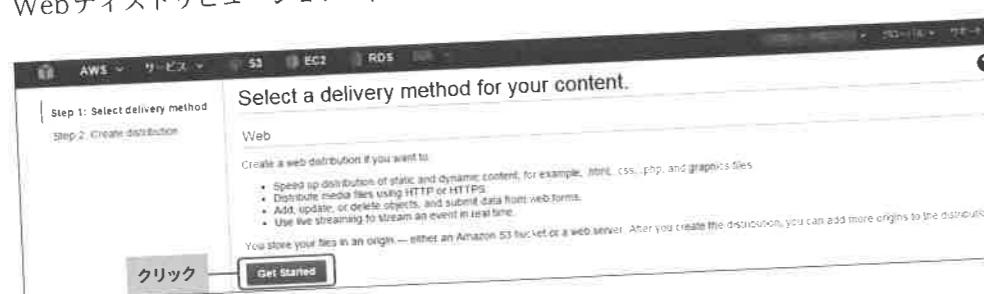


図3 「Get started」をクリック

3 ディストリビューションを作成する

オリジン、キャッシング、ディストリビューションの内容をそれぞれ表1～3のとおり設定します(図4～6)。最後に「Create Distribution」をクリックします(図6⑪)。

表1 オリジンの設定

| 項目名 | 設定内容 | 備考 |
|-------------------------|------------------------|--------------|
| ①Origin Domain Name | wp.aws-book.com | — |
| ②Origin Path | — | — |
| ③Origin ID | Custom-wp.aws-book.com | ①を入力すると変更される |
| ④Origin Protocol Policy | Match Viewer | — |
| ⑤HTTP Port | 80 | — |
| ⑥HTTPS Port | 443 | — |

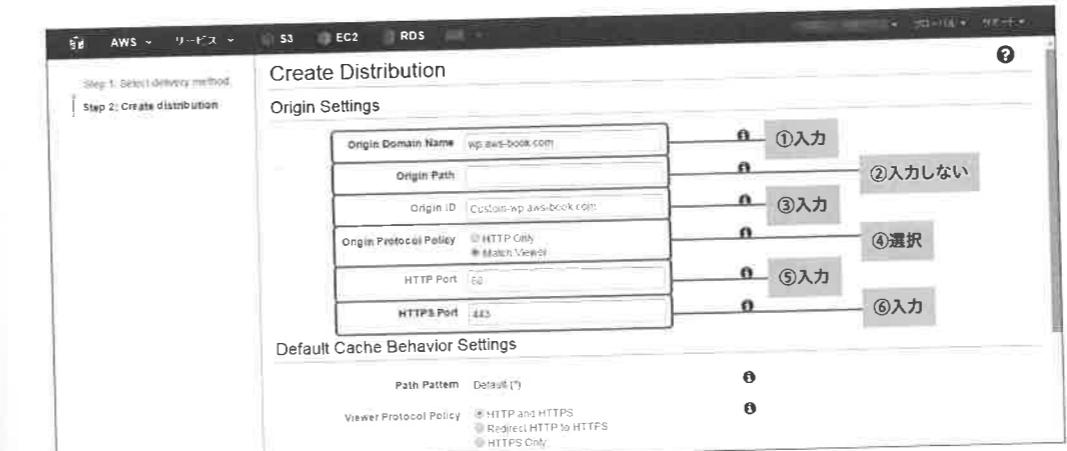


図4 オリジンの設定

表2 キャッシュの設定

| 項目名 | 設定内容 | 備考 |
|-------------------------|--|----|
| ①Path Pattern | * | — |
| ②Viewer Protocol Policy | HTTP and HTTPS | — |
| ③Allowed HTTP Methods | GET, HEAD, OPTIONS, PUT, POST, PATCH, DELETE | — |
| ④Cached HTTP Methods | チェックを外す | — |

(続き)

| 項目名 | 設定内容 | 備考 |
|-------------------------|--------------------------|----|
| ⑤Forward Headers | None | — |
| ⑥Object Caching | Use Origin Cache Headers | — |
| ⑦Minimum TTL | 0 | — |
| ⑧Maximum TTL | 31536000 | — |
| ⑨Default TTL | 86400 | — |
| ⑩Forward Cookies | All | — |
| ⑪Forward Query Strings | Yes | — |
| ⑫Smooth Streaming | No | — |
| ⑬Restrict Viewer Access | No | — |

Step 1: Select delivery method
Step 2: Create distribution

Default Cache Behavior Settings

- Path Pattern: Default (*) ①確認
- Viewer Protocol Policy: *HTTP and HTTPS ②選択
- Allowed HTTP Methods: GET, HEAD ③選択
- Cached HTTP Methods: GET, HEAD (Cached by default) ④選択
- Forward Headers: None (Improves Caching) ⑤選択
- Object Caching: Use Origin Cache Headers ⑥選択
- Minimum TTL: 0 ⑦入力
- Maximum TTL: 31536000 ⑧入力
- Default TTL: 86400 ⑨入力
- Forward Cookies: All ⑩選択
- Forward Query Strings: Yes ⑪選択
- Smooth Streaming: No ⑫選択
- Restrict Viewer Access: No ⑬選択

Distribution Settings

- Price Class: Use Only US, Europe and Asia ⑯

図5 キャッシュの設定

負荷対策を行い機会損失を減らす

表3 ディストリビューションの設定

| 項目名 | 設定内容 | 備考 |
|----------------------------------|--------------------------------|-----------------|
| ①Price Class | Use Only US, Europe and Asia | — |
| ②Alternate Domain Names (CNAMEs) | cdn.aws-book.com | CloudFront動作確認用 |
| ③SSL Certificate | Default CloudFront Certificate | — |
| ④Default Root Object | — | — |
| ⑤Logging | Off | — |
| ⑥Bucket for Logs | — | — |
| ⑦Log Prefix | — | — |
| ⑧Cookie Logging | Off | — |
| ⑨Comment | — | — |
| ⑩Distribution State | Enabled | — |

Step 1: Select delivery method
Step 2: Create distribution

(Use Signed URLs or Signed Cookies)

Distribution Settings

- Price Class: Use Only US, Europe and Asia ①選択
- Alternate Domain Names (CNAMEs): cdn.aws-book.com ②入力
- SSL Certificate: * Default CloudFront Certificate (* cloudfront.net) ③入力しない
- Custom SSL Certificate (Optional): No certificates available ④入力しない
- Default Root Object: — ⑤選択
- Logging: Off ⑥入力しない
- Bucket for Logs: — ⑦入力しない
- Log Prefix: — ⑧選択
- Cookie Logging: Off ⑨入力しない
- Distribution State: Enabled ⑩選択

Create Distribution ⑪クリック

図6 ディストリビューション設定

4 ステータスを確認する

Statusには「InProgress」と表示され、作成処理が開始されます(図7)。

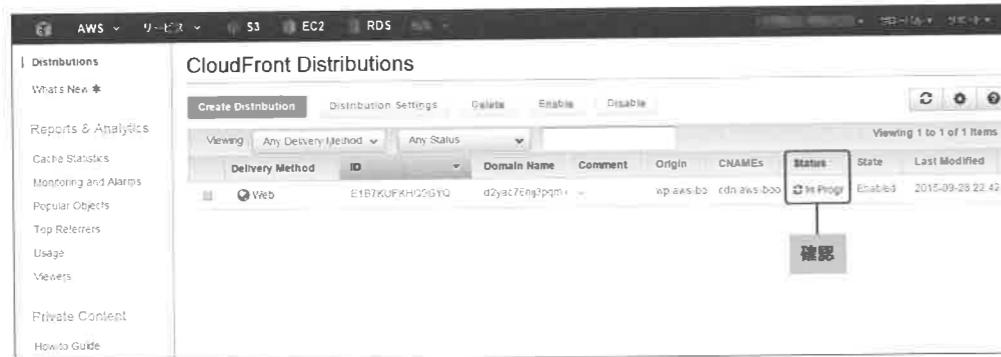


図7 作成処理Status「InProgress」

5 ステータスが変更されたことを確認する

しばらくすると(結構時間がかかる)、Statusが「Deployed」に変わり(図8①)、作成処理完了です。作成したディストリビューションを選択して②、「Distribution Settings」をクリックします③。

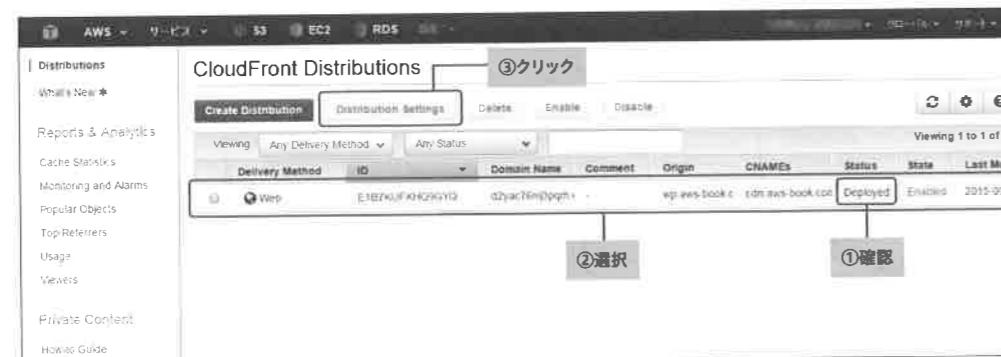


図8 作成処理Status「Deployed」

6 Domain Name を確認する

「General」をクリックして(図8①)、「Domain Name」の内容を確認します②。

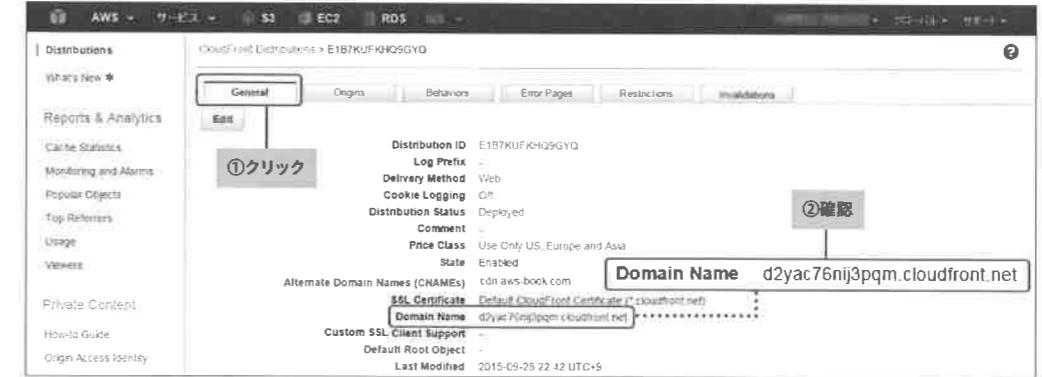


図9 「Domain Name」の確認

7 「Hosted Zones」をクリックする

Route 53マネージメントコンソールを開き、左サイドのメニューから「Hosted Zones」をクリックすると(図10)、登録されているドメイン一覧が表示されます。該当ドメインを選択すると、設定されているDNSレコードが表示されます。この手順以降では該当ドメインのDNSレコードを追加します。

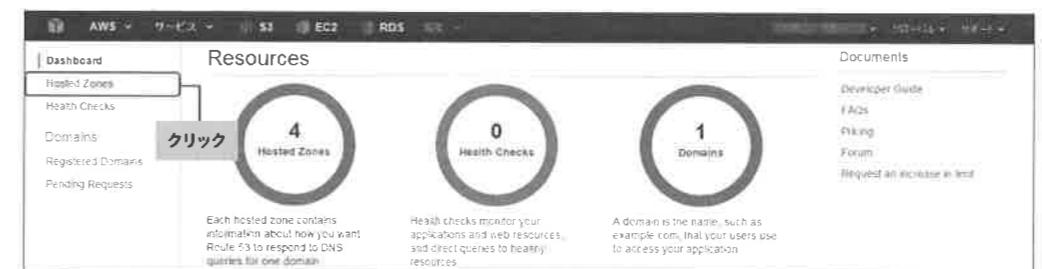


図10 Route 53マネージメントコンソール

8 CloudFrontのDomain NameをCNAMEで追加する

手順⑥で調べたCloudFrontのDomain NameをCNAMEで追加します。「Create Record Set」をクリックして(図11①)、表4を参考にCNAMEレコードを作成します②～⑤。

表4 CNAMEレコード

| 項目 | 内容 |
|--------|-------------------------------|
| ②Name | cdn |
| ③Type | CNAME |
| ④Value | d2yac76nij3pqm.cloudfront.net |