



このコンテンツは公開から3年以上経過しており内容が古い可能性があります  
最新情報については[サービス別資料](#)もしくはサービスのドキュメントをご確認ください

# [AWS Black Belt Online Seminar]

## Amazon EC2 Auto Scaling & AWS Auto Scaling

サービスカットシリーズ  
ソリューションアーキテクト  
滝口 開資  
2019-10-02

AWS 公式 Webinar  
<https://amzn.to/JPWebinar>



過去資料  
<https://amzn.to/JPArchive>



# 自己紹介

滝口 開資 (たきぐち はるよし)

ソリューションアーキテクト - EC2スポットインスタンススペシャリスト  
日本市場でのEC2スポットインスタンス技術担当

好きなAWSサービス

- Amazon EC2 Auto Scaling
- AWS Auto Scaling
- AWSサポート



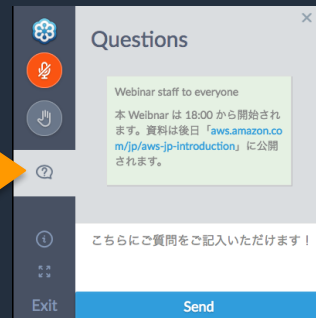
# AWS Black Belt Online Seminar とは

「サービス別」「ソリューション別」「業種別」のそれぞれのテーマに分かれて、アマゾンウェブサービス ジャパン株式会社が主催するオンラインセミナーシリーズです。

質問を投げることができます！

- 書き込んだ質問は、主催者にしか見えません
- 今後のロードマップに関するご質問は  
お答えできませんのでご了承ください

- ① 吹き出しをクリック
- ② 質問を入力
- ③ Sendをクリック



Twitter ハッシュタグは以下をご利用ください  
#awsblackbelt

# 内容についての注意点

- 本資料では2018年x月x日時点のサービス内容および価格についてご説明しています。最新の情報はAWS公式ウェブサイト(<http://aws.amazon.com>)にてご確認ください。
- 資料作成には十分注意しておりますが、資料内の価格とAWS公式ウェブサイト記載の価格に相違があった場合、AWS公式ウェブサイトの価格を優先とさせていただきます。
- 価格は税抜表記となっております。日本居住者のお客様が東京リージョンを使用する場合、別途消費税をご請求させていただきます。
- AWS does not offer binding price quotes. AWS pricing is publicly available and is subject to change in accordance with the AWS Customer Agreement available at <http://aws.amazon.com/agreement/>. Any pricing information included in this document is provided only as an estimate of usage charges for AWS services based on certain information that you have provided. Monthly charges will be based on your actual use of AWS services, and may vary from the estimates provided.

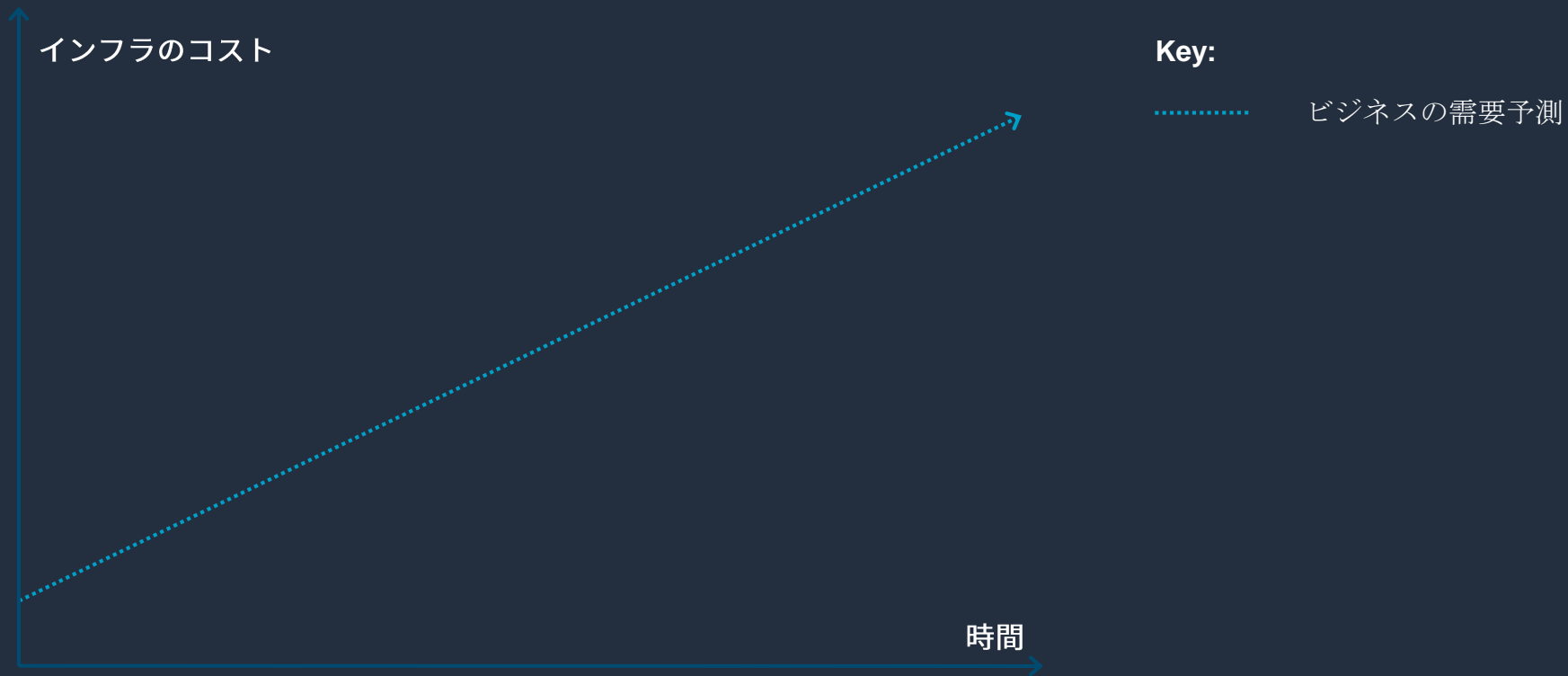
# 本日のアジェンダ

- Auto Scalingサービスのコンセプト
- Auto Scalingの基礎知識
- 主要機能：スケーリングの整理
- Auto Scalingを使ってみる
- こんなときどうする？ - 各種機能の紹介
- まとめ・参考資料

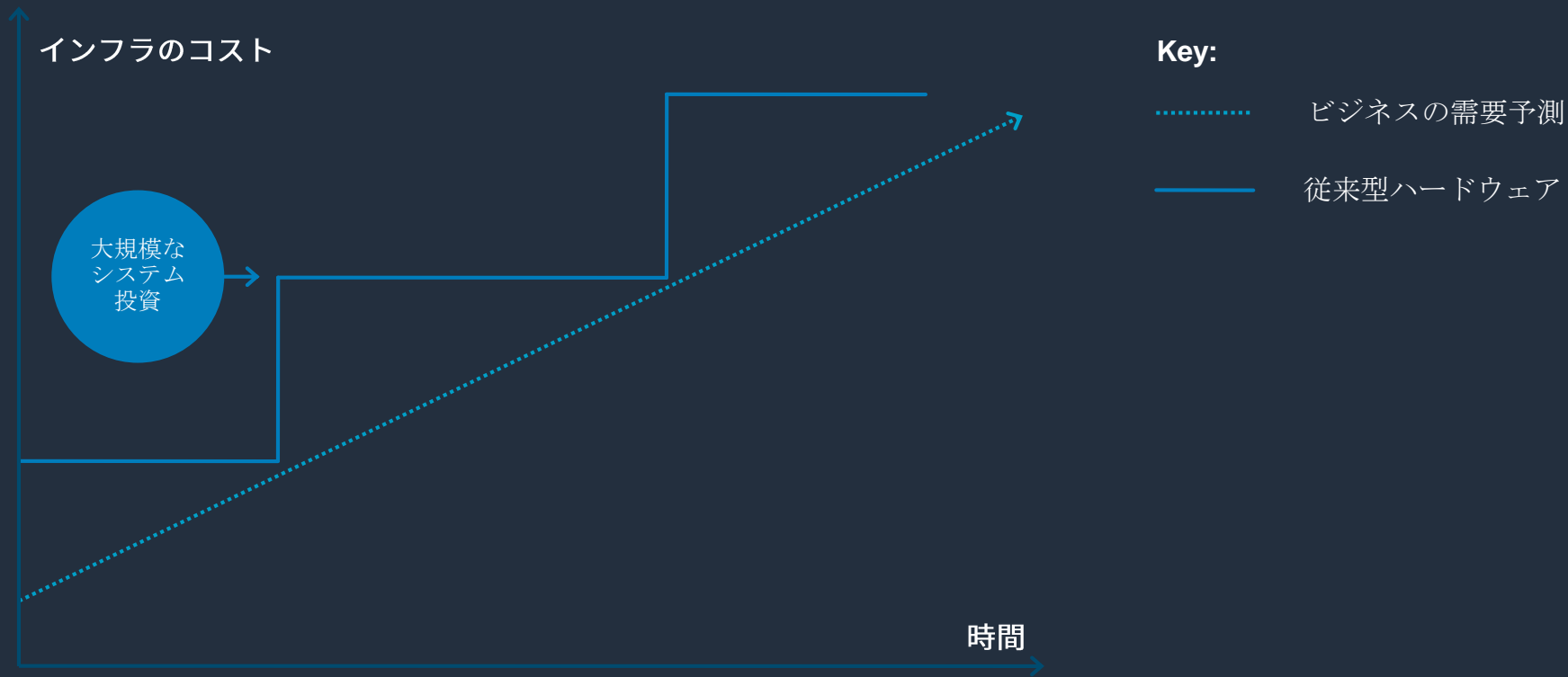
# 本日のアジェンダ

- Auto Scalingサービスのコンセプト
- Auto Scalingの基礎知識
- 主要機能：スケーリングの整理
- Auto Scalingを使ってみる
- こんなときどうする？ - 各種機能の紹介
- まとめ・参考資料

# ビジネス需要に応じたキャパシティ準備

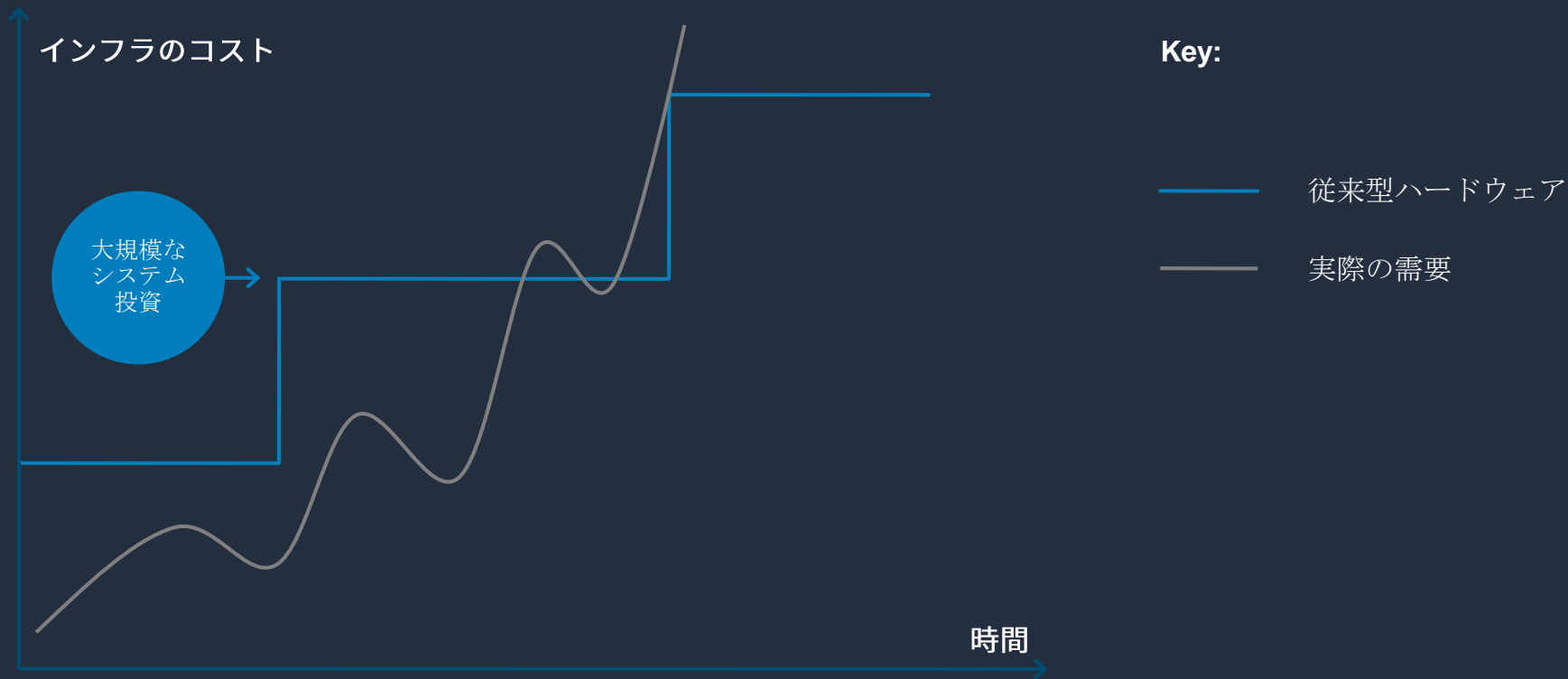


# ビジネス需要に応じたキャパシティ準備

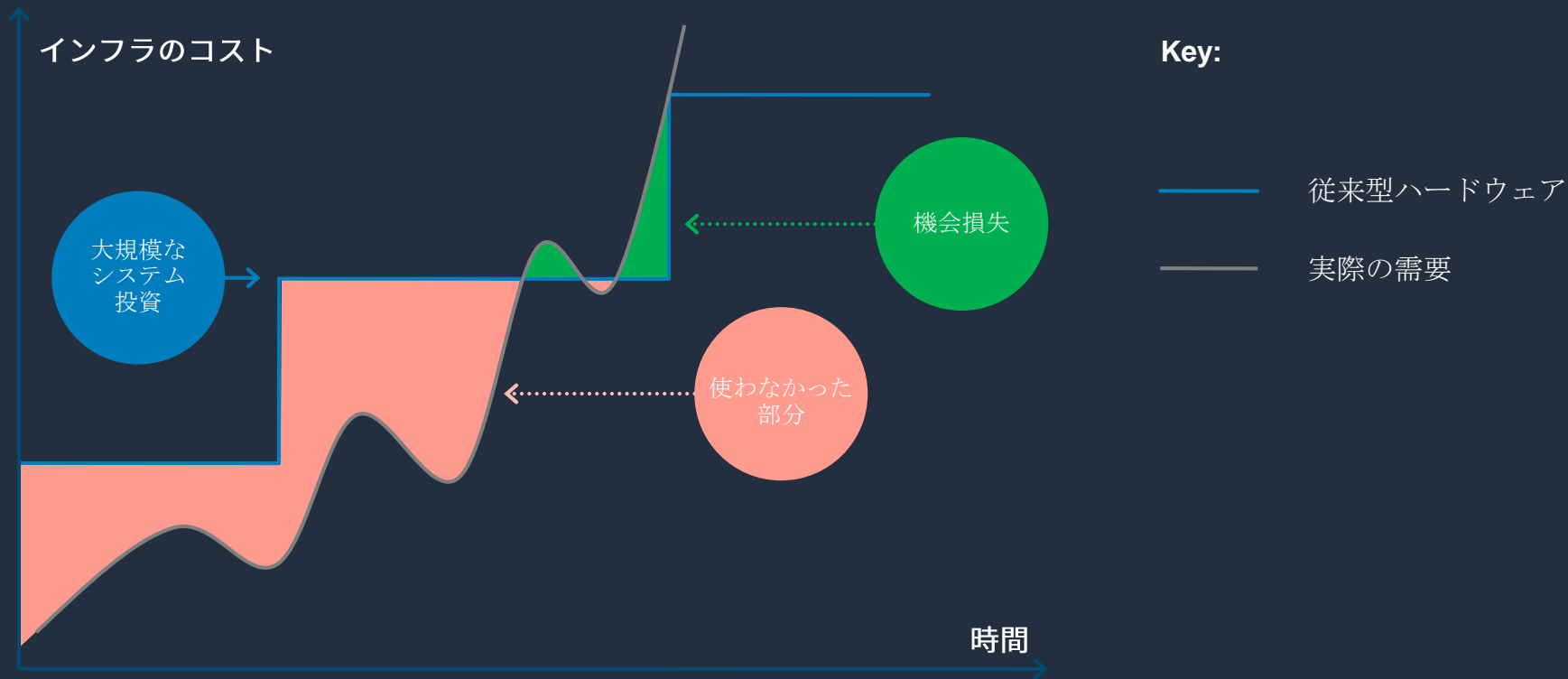




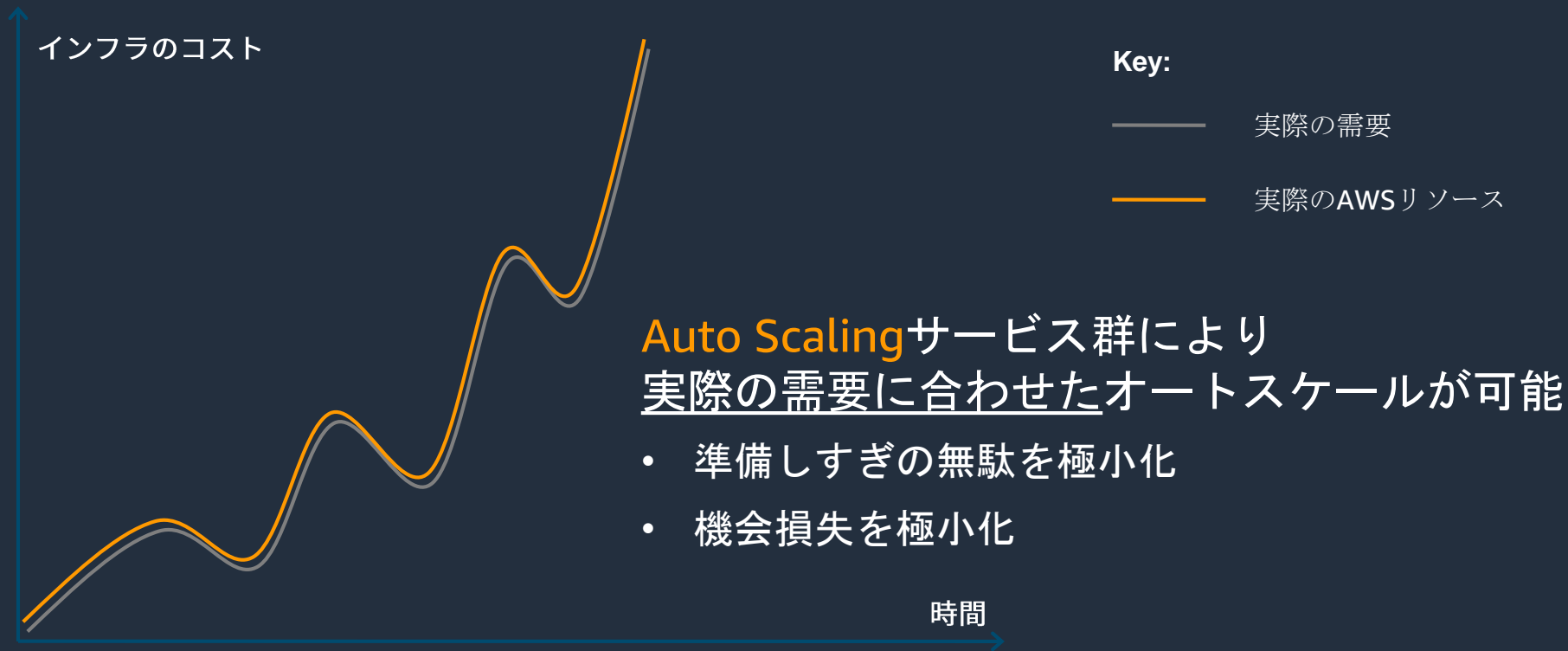
# ビジネス需要に応じたキャパシティ準備



# ビジネス需要に応じたキャパシティ準備



# ビジネス需要に応じたキャパシティ準備



# 本日のアジェンダ

- Auto Scalingサービスのコンセプト
- **Auto Scalingの基礎知識**
- 主要機能：スケーリングの整理
- Auto Scalingを使ってみる
- こんなときどうする？ - 各種機能の紹介
- まとめ・参考資料

# Auto Scalingの基礎知識

- 動作原理 - 希望する容量 (Desired Capacity, 以下「希望容量」) を目標に
- インスタンスの分散
- 均質性 - 「名前をつけてかわいがない」

# Auto Scalingの基礎知識

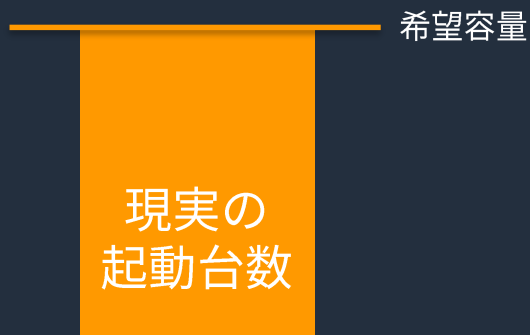
- 動作原理 - 希望する容量 (Desired Capacity, 以下「希望容量」) を目標に
- インスタンスの分散
- 均質性 - 「名前をつけてかわいがない」

# 動作原理

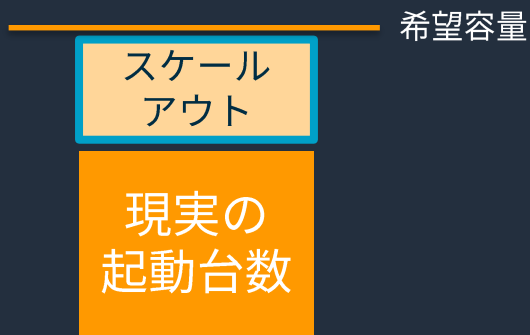
Auto Scalingは、

1. 希望容量と現実の起動台数との差を監視し、
2. 常に希望容量に合致するようにリソース(EC2インスタンスなど)を増減する

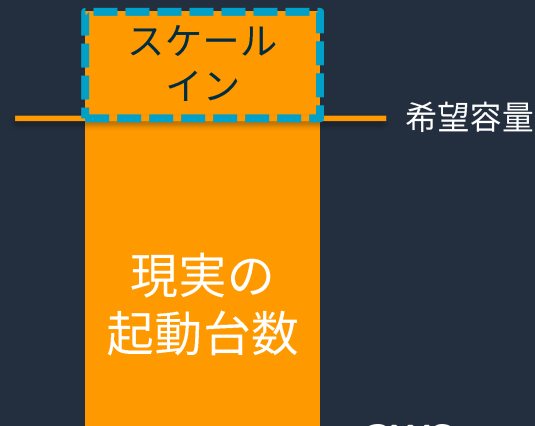
## 1) 静観



## 2) スケールアウト



## 3) スケールイン



# 希望容量の使われ方

- サイズの維持
- 手動スケーリング
- 自動スケーリング



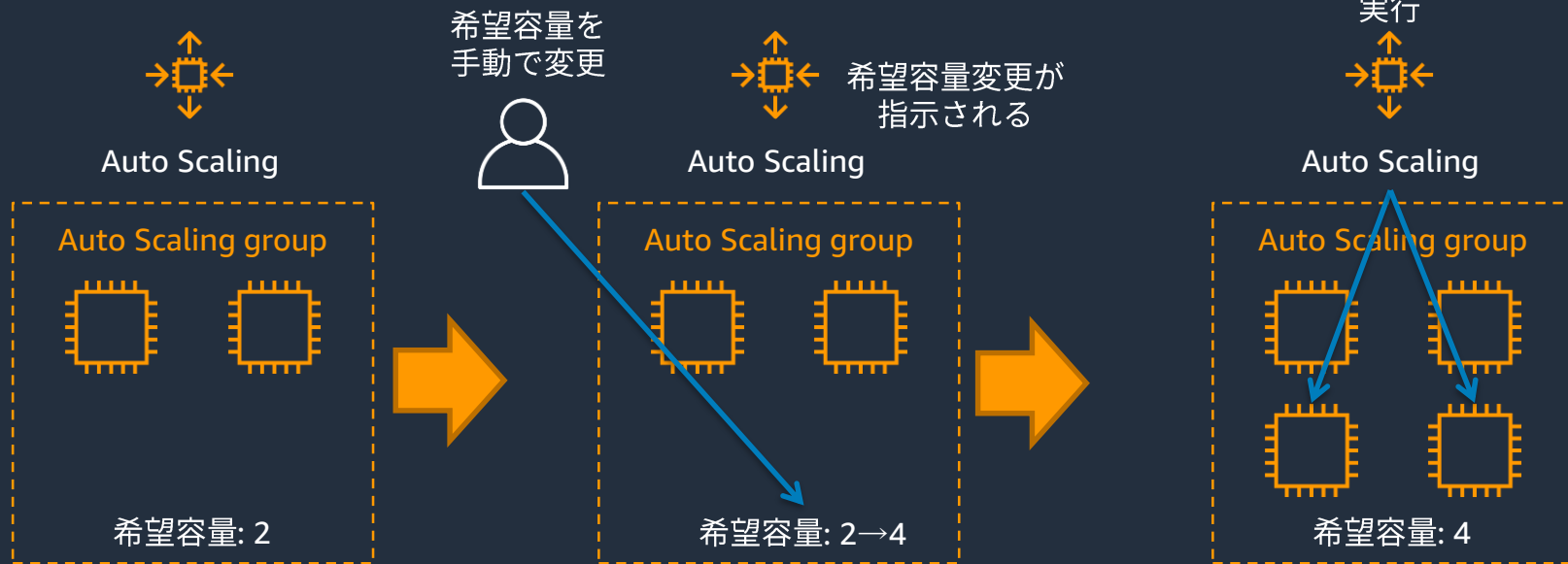
# 希望容量の使われ方

- サイズの維持
  - 希望容量は固定
  - 現実の台数が減るとその差分を検知して1台追加する
  - 一番シンプルな使い方



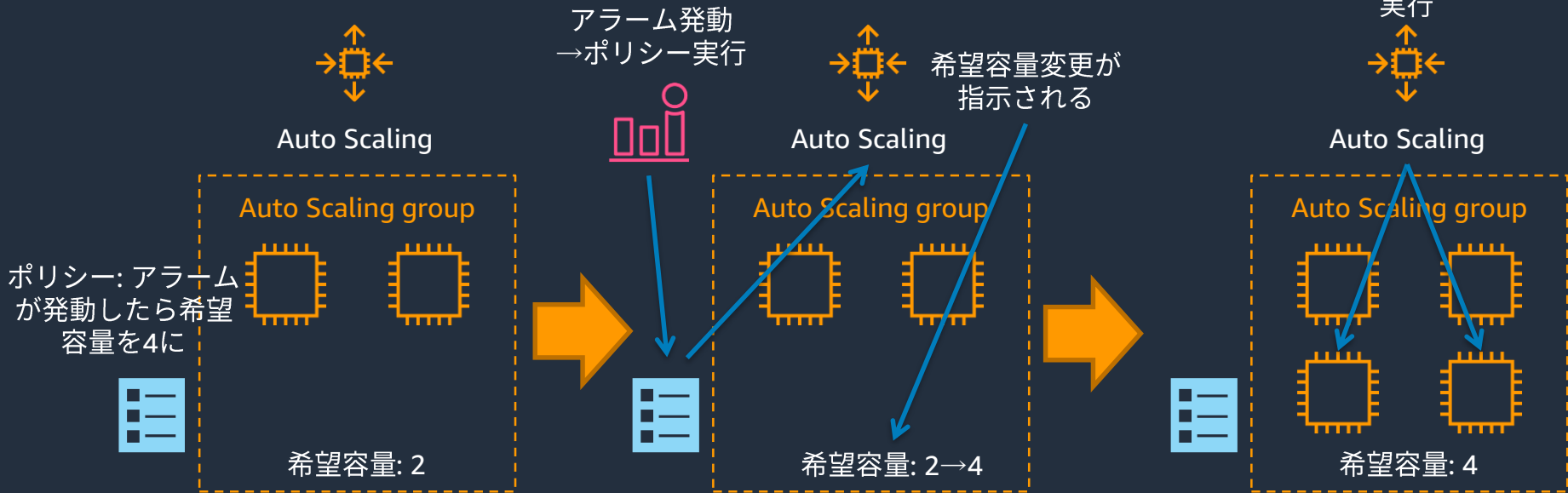
# 希望容量の使われ方

- 手動スケーリング
  - 希望容量を手動で変更する
  - これに追従してAuto Scalingサービスが台数を変化させる
  - 「サイズの維持」も引き続き行われる



# 希望容量の使われ方

- 自動スケーリング
  - 様々な条件に応じて希望容量が動的に変化する
  - これに追従してAuto Scalingサービスが現実の台数を変化させる
  - この設定方法として様々なスケーリングポリシーが提供されている

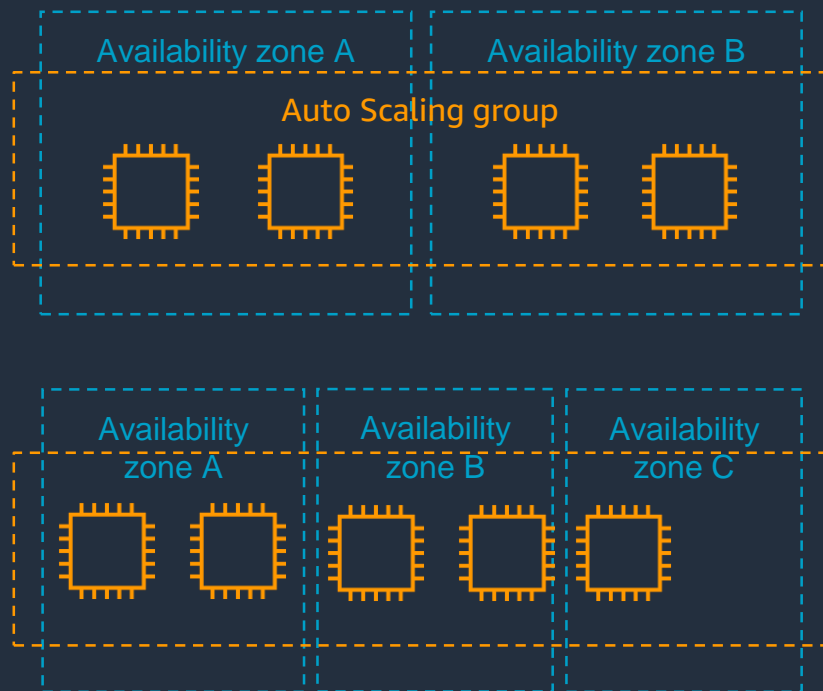


# Auto Scalingの基礎知識

- 動作原理 - 希望する容量 (Desired Capacity, 以下「希望容量」) を目標に
- インスタンスの分散
- 均質性 - 「名前をつけてかわいがない」

# インスタンスの分散

- 使用できるアベイラビリティゾーンの間で、均等にインスタンスを配置しようとする
  - スケールアウトするとき：インスタンス数が最も少ないアベイラビリティゾーンに新規起動
    - これに失敗する場合、別のアベイラビリティゾーンを選択



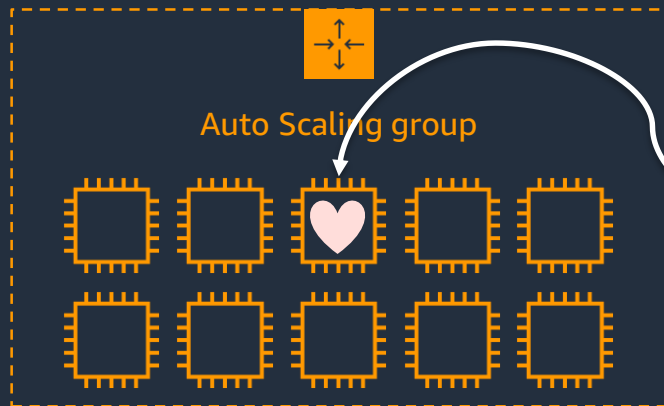
EC2 Auto Scalingはスケール動作時に「インスタンスの分散」を最も重視する

# Auto Scalingの基礎知識

- 動作原理 - 希望する容量 (Desired Capacity, 以下「希望容量」) を目標に
- インスタンスの分散
- 均質性 - 「名前をつけてかわいがない」

# 均質性 - 「名前をつけてかわいがない」

- Auto Scalingグループ内のインスタンスは原則として全て均一で、同一の価値を持つ
- 名前をつけるのはバッドプラクティス。スケールインはいつでも発生するものとして、置き換え可能にしておくのが重要



特定インスタンスを  
かわいがない

# Auto Scalingの世界の整理



EC2 Auto  
Scaling

EC2インスタンス





# Auto Scalingの世界の整理

## EC2 Auto Scaling

EC2インスタンス

## Application Auto Scaling

ECSクラスター、スポットフリート、  
EMRクラスター、AppStream 2.0フリート、  
DynamoDBテーブル、Auroraレプリカ、  
SageMakerエンドポイントバリエーション、  
カスタムリソース

# Auto Scalingの世界の整理

## AWS Auto Scaling

様々なリソース

スケーリングプラン

(動的スケーリング+予測スケーリング)

### EC2 Auto Scaling

EC2インスタンス

### Application Auto Scaling

ECSクラスター、スポットフリート、  
EMRクラスター、AppStream 2.0フリート、  
DynamoDBテーブル、Auroraレプリカ、  
SageMakerエンドポイントバリエーション、  
カスタムリソース

# Auto Scalingの世界の整理

## AWS Auto Scaling

様々なリソース

スケーリングプラン

(動的スケーリング+予測スケーリング)

予測ス  
ケーリン  
グの管理  
(EC2のみ)

## EC2 Auto Scaling

EC2インスタンス

EC2の  
管理

## Application Auto Scaling

ECSクラスター、スポットフリート、  
EMRクラスター、AppStream 2.0フリート、  
DynamoDBテーブル、Auroraレプリ  
カ、SageMakerエンドポイントバリアン  
ト、カスタムリソース

その他  
リソース  
の管理

# 本日のアジェンダ

- Auto Scalingサービスのコンセプト
- Auto Scalingの基礎知識
- **主要機能：スケーリングの整理**
- Auto Scalingを使ってみる
- こんなときどうする？ - 各種機能の紹介
- まとめ・参考資料

# 主要機能：スケーリングの整理

- 動的なスケーリング
  - 簡易スケーリング
  - ステップスケーリング
  - ターゲット追跡スケーリング
- 予測スケーリング
- スケジュールスケーリング
- スケーリングオプションの選択指針

# 動的なスケーリング – 簡易スケーリング

- EC2 Auto Scalingのみ
- 1つのメトリクスに対して1種類だけのスケーリング調整値を指定
  - 例: CPUUtilizationが50%になったら1台追加

mysimplescalingpolicy 操作 ▾

---

**ポリシータイプ:** 簡易スケーリング

**次の場合にポリシーを実行:** awsec2-myalbtestasg-CPU-  
アラームしきい値を超えました: CPUUtilization >= 50 (連続する 300 秒 x 1)  
(メトリクスのディメンション) AutoScalingGroupName = myalbtestasg

**アクションを実行:** 追加 1 インスタンス

**その後待機:** 300 次のスケーリング動作までの秒数

- 現在は非推奨であり、ステップスケーリングを推奨[1]
  - 「スケーリング調整が1つの場合でも、簡易スケーリングポリシーではなくステップスケーリングポリシーを使用することをお勧めします。」
  - 「ほとんどの場合、ステップスケーリングポリシーは簡易スケーリングポリシーよりも適しています。」

[1] [https://docs.aws.amazon.com/ja\\_jp/autoscaling/ec2/userguide/as-scaling-simple-step.html](https://docs.aws.amazon.com/ja_jp/autoscaling/ec2/userguide/as-scaling-simple-step.html)

# 動的なスケーリング – ステップスケーリング (1)

- EC2 Auto Scaling, Application Auto Scaling
- 1つのメトリクスに対して複数のスケーリング調整値を指定可能
- きめ細やかな設定が可能

mysteps-scaling-policy 操作 ▾

---

**ポリシータイプ:** ステップスケーリング

**次の場合にポリシーを実行:** awsec2-myalbtestasg-CPU-  
アラームしきい値を超えました: CPUUtilization >= 50 (連続する 300 秒 x 1)  
(メトリクスのディメンション) AutoScalingGroupName = myalbtestasg

**アクションを実行:** 追加 1 インスタンス 次の条件の場合 50 <= CPUUtilization < 60  
追加 2 インスタンス 次の条件の場合 60 <= CPUUtilization < 70  
追加 3 インスタンス 次の条件の場合 70 <= CPUUtilization < 80  
追加 4 インスタンス 次の条件の場合 80 <= CPUUtilization < +無限大

**インスタンスは:** 300 秒のウォームアップが各ステップ後に必要です

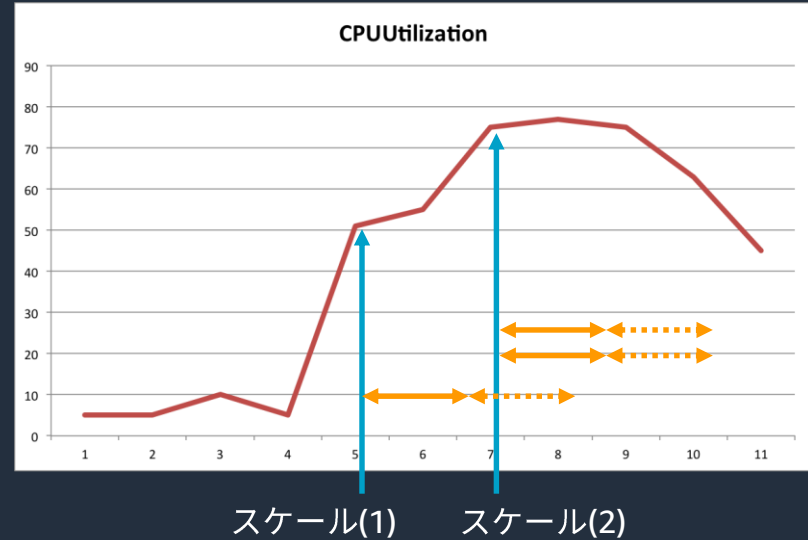
[https://docs.aws.amazon.com/ja\\_jp/autoscaling/ec2/userguide/as-scaling-simple-step.html](https://docs.aws.amazon.com/ja_jp/autoscaling/ec2/userguide/as-scaling-simple-step.html)

[https://docs.aws.amazon.com/ja\\_jp/autoscaling/application/userguide/application-auto-scaling-step-scaling-policies.html](https://docs.aws.amazon.com/ja_jp/autoscaling/application/userguide/application-auto-scaling-step-scaling-policies.html)

# 動的なスケーリング – ステップスケーリング (2)

- ウォームアップ期間：新しいインスタンスがサービス開始できるようになるまでに何秒を要するかを設定する値
  - スケール(1)のウォームアップ期間中に次のアラームが来てスケール(2)が開始される。このとき、3台追加ではなく、「今1台追加中」とみなし、差し引き2台を追加する
  - これにより追加しすぎ問題を解決できる
- スケールアウトのタイミングで、一つ前のスケールアウトが進行中かどうかを判断してくれる、と考えても良い
- デフォルト値は300秒

←→ インスタンス起動  
←...→ ウォームアップ



ステップスケーリングポリシー定義

1台追加:  $50 \leq \text{CPUUtil} < 60$

2台追加:  $60 \leq \text{CPUUtil} < 70$

3台追加:  $70 \leq \text{CPUUtil} < 80$



# 動的なスケーリング – ターゲット追跡スケーリング (1/3)

- EC2 Auto Scaling, Application Auto Scaling
- 1つのメトリクスに対し、単に目標値を指定するのみで良い
  - CPUUtilizationを40%に維持して欲しい。ただこれだけ

AutoScaling-albtest1-58558132-caa3-4ee0-a475-a340c4dcf26d

操作 ▾

**ポリシータイプ:** ターゲットの追跡スケーリング

**次の場合にポリシーを  
実行:** CPU の平均使用率 を 40 に維持するために必要な場合

**アクションを実行:** 必要に応じてインスタンスを追加または削除

**インスタンスは:** 300 スケーリング後にウォームアップする秒数

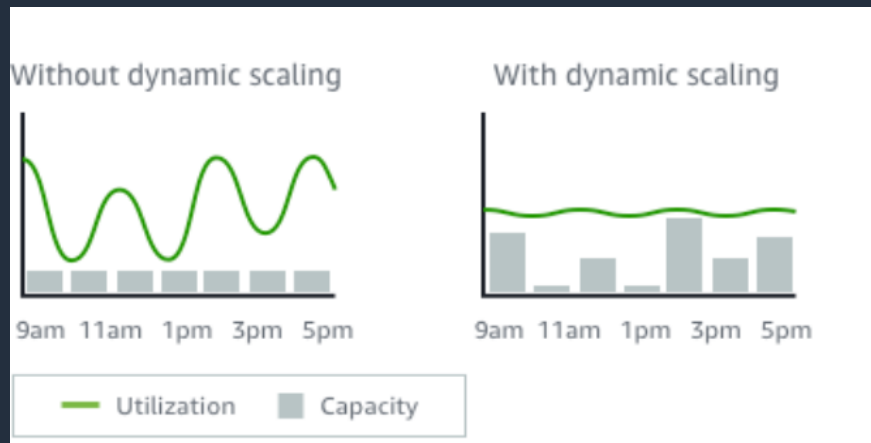
**スケールインの無効化** いいえ

[https://docs.aws.amazon.com/ja\\_jp/autoscaling/ec2/userguide/as-scaling-target-tracking.html](https://docs.aws.amazon.com/ja_jp/autoscaling/ec2/userguide/as-scaling-target-tracking.html)

[https://docs.aws.amazon.com/ja\\_jp/autoscaling/application/userguide/application-auto-scaling-target-tracking.html](https://docs.aws.amazon.com/ja_jp/autoscaling/application/userguide/application-auto-scaling-target-tracking.html)

# 動的なスケーリング – ターゲット追跡スケーリング (2/3)

- 目標値を満たすように自動的にリソースが調整される
  - 何も設定しない場合、キャパシティ (灰色) が一定のため負荷 (緑色) が変動する
  - ターゲット追跡スケーリングを設定すると、負荷に応じてキャパシティが増減する。その結果、負荷が一定の値におさまる



# 動的なスケーリング – ターゲット追跡スケーリング (3/3)

- スケールアウト用・スケールイン用の2本のアラームが自動的に作成される
  - TargetTracking-xxx-AlarmLow-UUID：スケールイン条件
  - TargetTracking-xxx-AlarmHigh-UUID：スケールアウト条件
- Highは敏感(3分など)、Lowはゆっくり(15分など)
- 基本的に、これらのアラームがアラーム状態になったときスケール



CloudWatch > アラーム

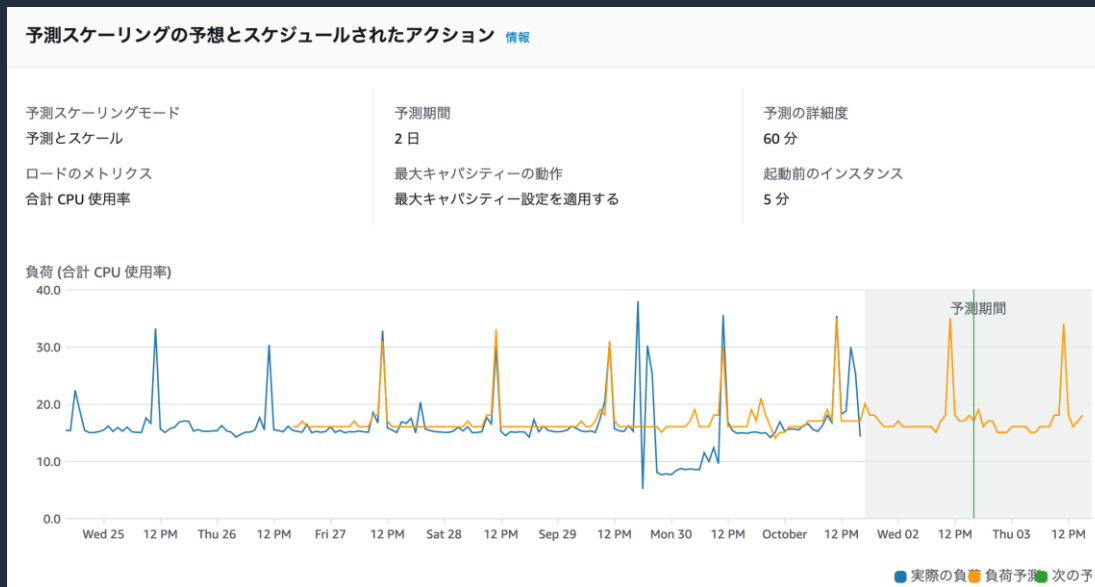
アラーム (11)  Auto Scaling アラームを非表示   アクション

🔍 TargetTracking-myalbtestasg  任意の状態  1

<input type="checkbox"/>	名前	状態	条件	アクション
<input type="checkbox"/>	TargetTracking-myalbtestasg-AlarmLow-1f2d7313-cfe1-476c-b70a-7c62fdfa6f98	🚨 アラーム状態	15 分内の15データポイントのCPUUtilization < 28	-
<input type="checkbox"/>	TargetTracking-myalbtestasg-AlarmHigh-1f287b9a-c798-477f-ad97-9185565127e3	✅ OK	3 分内の3データポイントのCPUUtilization > 40	-

# 予測スケーリング (1/3)

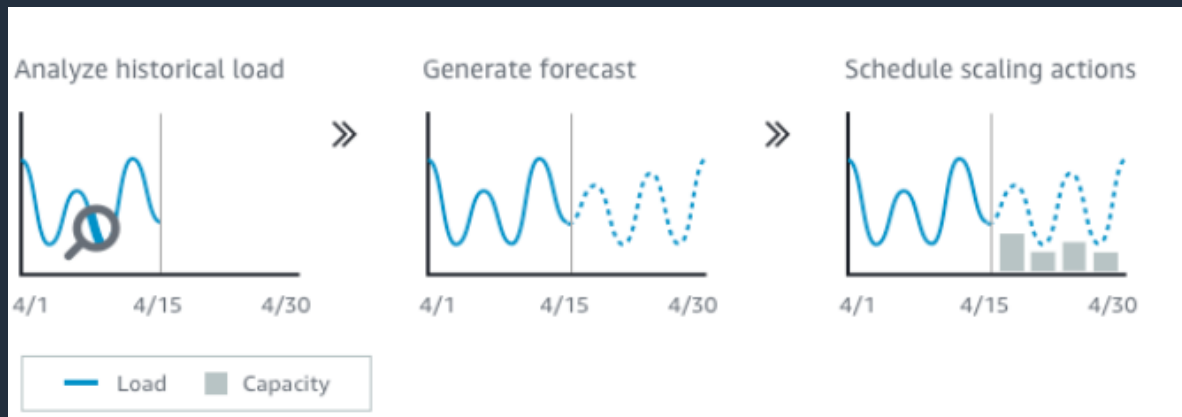
- EC2 Auto Scalingのみ (2019-10現在)
- 2週間分のメトリクスを分析し、次の2日の今後の需要を予測  
使用可能なメトリクス：CPUUtilization, NetworkIn, NetworkOut, および任意のメトリクス



[https://docs.aws.amazon.com/ja\\_jp/autoscaling/plans/userguide/how-it-works.html](https://docs.aws.amazon.com/ja_jp/autoscaling/plans/userguide/how-it-works.html)

## 予測スケーリング (2/3)

- 24時間ごとに、次の48時間の予測値を作成し、キャパシティの増減をスケジュールする



- 予測の基準時刻は毎時0分
  - 「インスタンスの事前起動」設定により、スケール動作を前もって実行させることができる
  - デフォルトは5分(300秒)前
    - 午前10時に負荷が予測されている場合、対応するスケールアウトは午前9時55分に実行される

# 予測スケーリング (3/3)

- 考慮点とベストプラクティス
  - AWS Auto Scaling マネジメントコンソールを使う
  - いきなり使い始めず、「予測のみ」モードでどのような予測値が評価されるかを確認できる
  - ASGの作成後、24時間待つ。予測の開始には最低24時間分のデータポイントが必要

# スケジュールスケーリング (1/2)

- EC2 Auto Scaling, Application Auto Scaling
- 一度限り、もしくは定期的なスケジュールを指定可能



The screenshot displays the AWS Auto Scaling console interface. At the top, the title is 'Auto Scaling グループ: myalbttestasg'. Below the title are several tabs: '詳細', 'アクティビティ履歴', 'スケーリングポリシー', 'インスタンス', 'モニタリング', '通知', 'タグ', and 'スケジュールされたアクション' (which is highlighted). Under the 'スケジュールされたアクション' tab, there are buttons for '予定アクションの作成' and '操作'. A search filter is present with the text 'Filter: 予定アクションのフィルター...'. Below the filter, a pagination indicator shows '1 から 25/ 37 の スケジュールされたアクション'. A table lists the scheduled actions with columns for '名前', '開始時刻', '終了時刻', '繰り返し', '希望するキャパシティ', '最小', and '最大'. One action is listed: 'mytestsched' with a start time of '2019 October 1 22:55:00 UTC+9', a capacity of '1', and a maximum capacity of '5'.

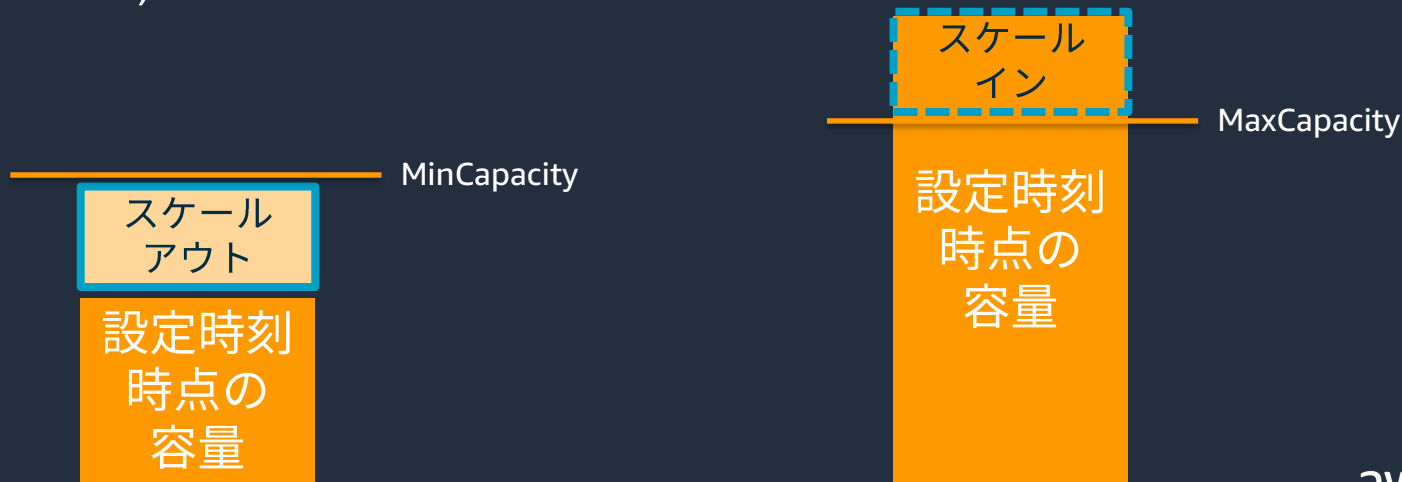
<input type="checkbox"/>	名前	開始時刻	終了時刻	繰り返し	希望するキャパシティ	最小	最大
<input type="checkbox"/>	mytestsched	2019 October 1 22:55:00 UTC+9			1	1	5

[https://docs.aws.amazon.com/ja\\_jp/autoscaling/ec2/userguide/schedule\\_time.html](https://docs.aws.amazon.com/ja_jp/autoscaling/ec2/userguide/schedule_time.html)

[https://docs.aws.amazon.com/ja\\_jp/autoscaling/application/userguide/application-auto-scaling-scheduled-scaling.html](https://docs.aws.amazon.com/ja_jp/autoscaling/application/userguide/application-auto-scaling-scheduled-scaling.html)

## スケジュールスケーリング (2/2)

- MinCapacity(最小キャパシティ)とMaxCapacity(最大キャパシティ)のいずれか、あるいは両方を指定可能
  - 設定時刻時点の容量がMinCapacityに満たない→MinCapacityまでスケールアウト
  - 設定時刻時点の容量がMaxCapacityを超している→MaxCapacityまでスケールイン
- (EC2 ASのみ) MinCapacity, MaxCapacity, DesiredCapacity(希望キャパシティ)を指定可能

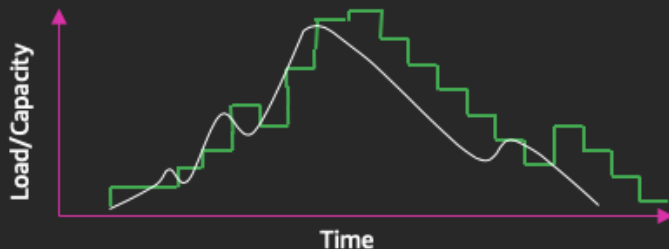




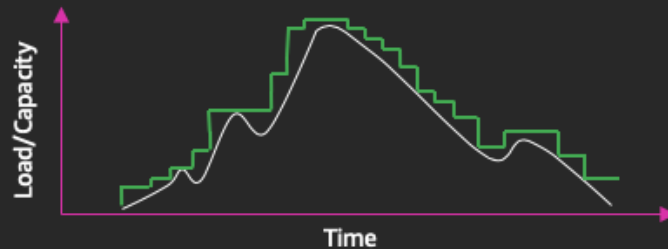
# スケーリングオプションの選択指針 (1/2)

EC2でのお勧めオプション：予測スケーリングを使い、同時にターゲット追跡スケーリングも有効にする

- 1) 大まかなキャパシティ増減は予測スケーリングに任せ、前もってスケールしておく
- 2) 実際の負荷に対して不足した分をターゲット追跡で補充する



予測スケーリングによる  
キャパシティの事前準備



予測スケーリング+ターゲット追跡スケーリング  
によるキャパシティ準備

準備されたキャパシティ量

実際の負荷

# スケーリングオプションの選択指針 (2/2)

ステップスケーリングおよびスケジュールスケーリングを使う

- 個々の条件下でのスケーリングを細かく制御したいときの考え方として引き続き有効
- EC2以外のリソースについてはこちらを選択
- キャパシティ設計時に、個別のパターンを下から積み上げて設定していく場合に向いている

# 本日のアジェンダ

- Auto Scalingサービスのコンセプト
- Auto Scalingの基礎知識
- 主要機能：スケーリングの整理
- Auto Scalingを使ってみる
- こんなときどうする？ - 各種機能の紹介
- まとめ・参考資料

# Auto Scalingを試してみる

- EC2 Auto Scalingを試してみる
- Application Auto Scalingを試してみる
- AWS Auto Scaling – 予測スケーリングを試してみる

# Auto Scalingを試してみる

- EC2 Auto Scalingを試してみる
- Application Auto Scalingを試してみる
- AWS Auto Scaling – 予測スケーリングを試してみる

# EC2 Auto Scalingの新機能 – ミックスインスタンスグループ

- オンデマンドインスタンスとスポットインスタンスをひとつのAuto Scalingグループで管理
  - (オンデマンド:スポット) = (9:1)といった指定ができる
- インスタンスタイプを複数指定できる
- インスタンスタイプを分散できる



# EC2 Auto Scaling Groupの作成 (1)

キーペア  
ネットワークインターフェイス

ロードバランシング  
ロードバランサー  
ターゲットグループ

AUTO SCALING  
起動設定

**Auto Scaling グループ**

SYSTEMS MANAGER SERVICES  
コマンドの実行  
ステートマネージャー

**i** 起動テンプレートの提供が開始されました。  
EC2 Auto Scaling コンソールで、EC2 起動テンプレートのフルサポートが開始されました。新しい Auto Scaling グループには起動テンプレートを使用することをお勧めします。起動テンプレートにより、Amazon EC2 の最新機能を活用することができます。Auto Scaling グループを作成して開始するか、[詳細はこちら](#)を参照してください。

## Auto Scaling へようこそ

Auto Scaling を使用すると、Amazon EC2 キャパシティの自動的な管理、適切な数のアプリケーションインスタンスの維持、インスタンスの正常なグループの運用、必要に応じたスケーリングを行うことができます。  
[詳細はこちら](#)

**Auto Scaling グループの作成**

注意: 別のリージョンで Auto Scaling グループを作成するには、ナビゲーションバーからリージョンを選択します。

## 追加情報

- 入門ガイド
- ドキュメント
- すべての EC2 リソース
- フォーラム
- 料金
- お問い合わせ

# EC2 Auto Scaling Groupの作成 (2)

## Auto Scaling グループの作成

キャンセルして終了

このウィザードを終了して Auto Scaling グループを作成します。最初に、起動設定または起動テンプレートを選択して、インスタンスの起動に Auto Scaling グループが使用するパラメータを指定します。

起動設定

必要な Amazon EC2 の機能をサポートしている場合は、引き続き起動設定を使用できます。[詳細はこちら](#)

起動テンプレート **新規**

起動テンプレートにより、1つの種類のインスタンスを起動するか、複数のインスタンスタイプと購入オプションの組み合わせを起動するかのオプションを利用できます。起動テンプレートには Amazon EC2 の最新機能が含まれていて、更新とバージョンアップができます。[詳細はこちら](#)  
[新しい起動テンプレートの作成](#)

ミックスインスタンスグループ機能を使うには  
「起動テンプレート」を用いる必要がある



# EC2 Auto Scaling Groupの作成 (3)

1. Auto Scaling グループの詳細設定   2. スケーリングポリシーの設定   3. 通知の設定   4. タグを設定   5. 確認

## Auto Scaling グループの作成

グループ名 ⓘ

起動テンプレート ⓘ lt-0757b443c586fc262

起動テンプレートのバージョン ⓘ 1 (デフォルト)  新しい起動テンプレートの作成

起動テンプレートの説明 ⓘ -

フリートの構築

- 起動テンプレートに従う
- 購入オプションとインスタンスを組み合わせる

起動テンプレートにより、インスタンスタイプと購入オプション (オンデマンドまたはスポット) が決まります。  
オンデマンドインスタンスとスポットインスタンスの組み合わせ、および複数のインスタンスタイプを選択します。スポットインスタンスは、利用できる最も安い料金で自動的に起動されます。

グループサイズ ⓘ 開始時  インスタンス

「購入オプションとインスタンスを組み合わせる」を選択

# EC2 Auto Scaling Groupの作成 (4)

1. Auto Scaling グループの詳細設定

2. スケーリングポリシーの設定

3. 通知の設定

4. タグを設定

5. 確認

## Auto Scaling グループの作成

グループ名 ⓘ

起動テンプレート ⓘ

lt-0757b443c586fc262

起動テンプレートのバージョン ⓘ

1 (デフォルト)



新しい起動テンプレートの作成

起動テンプレートの説明 ⓘ

-

フリートの構築

起動テンプレートに従う

起動テンプレートにより、インスタンスタイプと購入オプション (オンデマンドまたはスポット) が決まります。

購入オプションとインスタンスを組み合わせる

オンデマンドインスタンスとスポットインスタンスの組み合わせ、および複数のインスタンスタイプを選択します。スポットインスタンスは、利用できる最も安い料金で自動的に起動されます。

インスタンスタイプ

許容できるインスタンスタイプをフリートに追加します。順序を変更し、オンデマンドインスタンスの起動の優先度を設定します。この順序によるスポットインスタンスへの影響はありません。

インスタンスタイプの選択

最低2つのインスタンスタイプを追加してください

インスタンスタイプの追加

インスタンスの分散 ⓘ

次のデフォルト設定を使用し、すぐに開始します。

# EC2 Auto Scaling Groupの作成 (5)

1. Auto Scaling グループの詳細設定

2. スケーリングポリシーの設定

3. 通知の設定

4. タグを設定

5. 確認

## Auto Scaling グループの作成

グループ名 ⓘ

起動テンプレート ⓘ

lt-0757b443c586fc262

起動テンプレートのバージョン ⓘ

1 (デフォルト)



新しい起動テンプレートの作成

起動テンプレートの説明 ⓘ

-

フリートの構築

起動テンプレートに従う

起動テンプレートにより、インスタンスタイプと購入オプション (オンデマンドまたはスポット) が決まります。

購入オプションとインスタンスを組み合わせる

オンデマンドインスタンスとスポットインスタンスの組み合わせ、および複数のインスタンスタイプを選択します。スポットインスタンスは、利用できる最も安い料金で自動的に起動されます。

インスタンスタイプ

許容できるインスタンスタイプをフリートに追加します。順序を変更し、オンデマンドインスタンスの起動の優先度を設定します。この順序によるスポットインスタンスへの影響はありません。

インスタンスタイプの選択

最低2つのインスタンスタイプを追加してください

インスタンスタイプの追加

インスタンスの分散 ⓘ

次のデフォルト設定を使用し、すぐに開始します。

# EC2 Auto Scaling Groupの作成 (6)

## フリートの構築

起動テンプレートに従う

起動テンプレートにより、インスタンスタイプと購入オプション (オンデマンドまたはスポット) が決まります。

購入オプションとインスタンスを組み合わせる

オンデマンドインスタンスとスポットインスタンスの組み合わせ、および複数のインスタンスタイプを選択します。スポットインスタンスは、利用できる最も安い料金で自動的に起動されます。

## インスタンスタイプ

許容できるインスタンスタイプをフリートに追加します。順序を変更し、オンデマンドインスタンスの起動の優先度を設定します。この順序によるスポットインスタンスへの影響はありません。

m4.large (2vCPU、8GiB)

c4.large (2vCPU、3.75GiB)

インスタンスタイプの追加

- 要件に合うインスタンスタイプを複数選択する
- 起動テンプレートに指定しておくことも可能

# EC2 Auto Scaling Groupの作成 (7)

## インスタンスの分散 ⓘ

次のデフォルト設定を使用し、すぐに開始します。

- 上記の優先度に基づき、オンデマンドインスタンスを起動します。
- アベイラビリティゾーンごとに 2 つの最低価格インスタンスタイプ間でスポットインスタンスを多様化します。
- 各インスタンスタイプの最大スポット料金を、オンデマンド料金と同じに設定します。
- 70% のオンデマンドインスタンスと 30% のスポットインスタンスを組み合わせで維持します。

## グループサイズ ⓘ

開始時  インスタンス

「インスタンスの分散」のチェックを外す

# EC2 Auto Scaling Groupの作成 (9)

インスタンスの分散 ⓘ	<input type="checkbox"/> 次のデフォルト設定を使用し、すぐに開始します。
オンデマンドの割り当て戦略 ⓘ	優先順位付け
最大スポット料金 ⓘ	<input checked="" type="radio"/> デフォルトを使用 (推奨) デフォルトでは現在のスポット料金が使用されますが、オンデマンド価格に上限が設定されます。 <input type="radio"/> 上限価格を設定 (1 インスタンス/時間あたり)
スポットの配分戦略 ⓘ	スポットインスタンスを <input type="text" value="2"/> アベイラビリティゾーンごとに最も価格の安いインスタンスタイプ間で多様化する
オプションのオンデマンドベース ⓘ	最初のインスタンスを <input type="text" value="0"/> オンデマンドとして指定します
ベースを超えるオンデマンド割合 ⓘ	<input type="text" value="70"/> % オンデマンドおよび 30% スポット
グループサイズ ⓘ	開始時 <input type="text" value="1"/> インスタンス

台数の考え方について次のスライドで説明

# EC2 Auto Scaling Groupの作成 (10)

インスタンスの分散 ⓘ	<input type="checkbox"/> 次のデフォルト設定を使用し、すぐに開始します。
オンデマンドの割り当て戦略 ⓘ	優先順位付け
最大スポット料金 ⓘ	<input checked="" type="radio"/> デフォルトを使用 (推奨) デフォルトでは現在のスポット料金が使用されますが、オンデマンド価格に上限が設定されます。 <input type="radio"/> 上限価格を設定 (1 インスタンス/時間あたり)
スポットの配分戦略 ⓘ	スポットインスタンスを <input type="text" value="2"/> アベイラビリティゾーンごとに最も価格の安いインスタンスタイプ間で多様化する
オプションのオンデマンドベース ⓘ	最初のインスタンスを <input type="text" value="2"/> オンデマンドとして指定します
ベースを超えるオンデマンド割合 ⓘ	<input type="text" value="70"/> % オンデマンドおよび 30% スポット
グループサイズ ⓘ	開始時 <input type="text" value="12"/> インスタンス

台数の考え方の例

- 「グループサイズ」: 12

グループサイズ = 12

# EC2 Auto Scaling Groupの作成 (10)

インスタンスの分散 ⓘ	<input type="checkbox"/> 次のデフォルト設定を使用し、すぐに開始します。
オンデマンドの割り当て戦略 ⓘ	優先順位付け
最大スポット料金 ⓘ	<input checked="" type="radio"/> デフォルトを使用 (推奨) デフォルトでは現在のスポット料金が使用されますが、オンデマンド価格に上限が設定されます。 <input type="radio"/> 上限価格を設定 (1 インスタンス/時間あたり)
スポットの配分戦略 ⓘ	スポットインスタンスを <input type="text" value="2"/> アベイラビリティゾーンごとに最も価格の安いインスタンスタイプ間で多様化する
オプションのオンデマンドベース ⓘ	最初のインスタンスを <input type="text" value="2"/> オンデマンドとして指定します
ベースを超えるオンデマンド割合 ⓘ	<input type="text" value="70"/> % オンデマンドおよび 30% スポット
グループサイズ ⓘ	開始時 <input type="text" value="12"/> インスタンス

## 台数の考え方の例

- 「グループサイズ」: 12
- 「オプションのオンデマンドベース」: 2

グループサイズ = 12

オンデマンド = 2



# EC2 Auto Scaling Groupの作成 (10)

インスタンスの分散 ⓘ	<input type="checkbox"/> 次のデフォルト設定を使用し、すぐに開始します。
オンデマンドの割り当て戦略 ⓘ	優先順位付け
最大スポット料金 ⓘ	<input checked="" type="radio"/> デフォルトを使用 (推奨) デフォルトでは現在のスポット料金が使用されますが、オンデマンド価格に上限が設定されます。 <input type="radio"/> 上限価格を設定 (1 インスタンス/時間あたり)
スポットの配分戦略 ⓘ	スポットインスタンスを <input type="text" value="2"/> アベイラビリティゾーンごとに最も価格の安いインスタンスタイプ間で多様化する
オプションのオンデマンドベース ⓘ	最初のインスタンスを <input type="text" value="2"/> オンデマンドとして指定します
ベースを超えるオンデマンド割合 ⓘ	<input type="text" value="70"/> % オンデマンドおよび 30% スポット
グループサイズ ⓘ	開始時 <input type="text" value="12"/> インスタンス

## 台数の考え方の例

- 「グループサイズ」: 12
- 「オプションのオンデマンドベース」: 2
- 「ベースを超えるオンデマンド割合」: 70:30

グループサイズ = 12

オンデマンド = 2

オンデマンド = 7

スポット = 3

# EC2 Auto Scaling Groupの作成 (10)

インスタンスの分散 ⓘ	<input type="checkbox"/> 次のデフォルト設定を使用し、すぐに開始します。
オンデマンドの割り当て戦略 ⓘ	優先順位付け
最大スポット料金 ⓘ	<input checked="" type="radio"/> デフォルトを使用 (推奨) デフォルトでは現在のスポット料金が使用されますが、オンデマンド価格に上限が設定されます。 <input type="radio"/> 上限価格を設定 (1 インスタンス/時間あたり)
スポットの配分戦略 ⓘ	スポットインスタンスを <input type="text" value="2"/> アベイラビリティゾーンごとに最も価格の安いインスタンスタイプ間で多様化する
オプションのオンデマンドベース ⓘ	最初のインスタンスを <input type="text" value="2"/> オンデマンドとして指定します
ベースを超えるオンデマンド割合 ⓘ	<input type="text" value="70"/> % オンデマンドおよび 30% スポット
グループサイズ ⓘ	開始時 <input type="text" value="12"/> インスタンス

## 台数の考え方の例

- 「グループサイズ」: 12
- 「オプションのオンデマンドベース」: 2
- 「ベースを超えるオンデマンド割合」: 70:30

## 結果

オンデマンド 9台  
スポット 3台

グループサイズ = 12

オンデマンド = 2

オンデマンド = 7

スポット = 3

2+7 = 9台

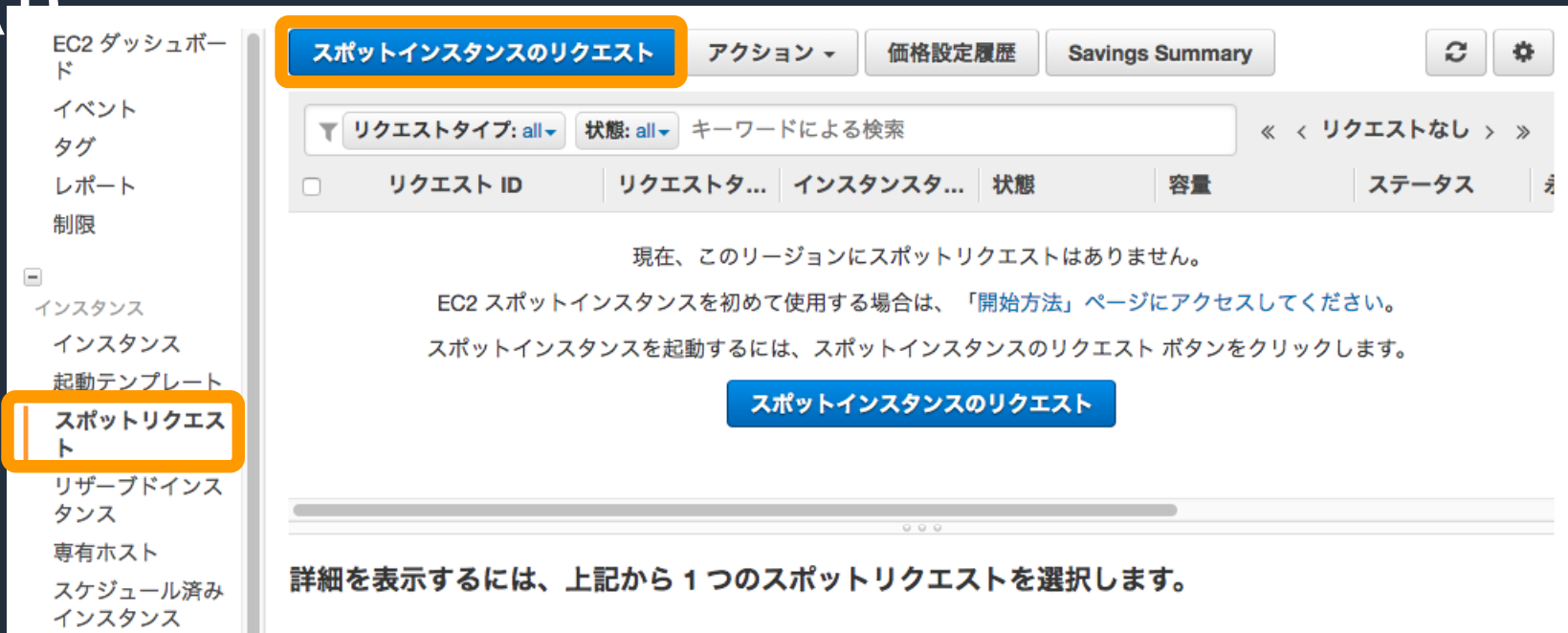
3台

# Auto Scalingを試してみる

- EC2 Auto Scalingを試してみる
- **Application Auto Scaling**を試してみる
- AWS Auto Scaling – 予測スケーリングを試してみる

# スポットフリートでのApplication Auto Scaling の活用

(1)



EC2 ダッシュボード

イベント

タグ

レポート

制限

インスタンス

インスタンス

起動テンプレート

**スポットリクエスト**

リザーブインスタンス

専用ホスト

スケジュール済みインスタンス

**スポットインスタンスのリクエスト**    アクション ▾    価格設定履歴    Savings Summary    ↻    ⚙

リクエストタイプ: all ▾    状態: all ▾    キーワードによる検索    << < リクエストなし > >>

リクエスト ID	リクエストタ...	インスタンス...	状態	容量	ステータス
----------	-----------	-----------	----	----	-------

現在、このリージョンにスポットリクエストはありません。

EC2 スポットインスタンスを初めて使用する場合は、「開始方法」ページにアクセスしてください。

スポットインスタンスを起動するには、スポットインスタンスのリクエスト ボタンをクリックします。

**スポットインスタンスのリクエスト**

詳細を表示するには、上記から1つのスポットリクエストを選択します。

「スポットリクエスト」 → 「スポットインスタンスのリクエスト」

詳細は以下の「EC2スポットインスタンスのすべて」資料をご参照ください

<https://aws.amazon.com/jp/summits/tokyo-osaka-2019-report/>

# スポットフリートでのApplication Auto Scaling の活用

必要な容量をお知らせください

起動するターゲット容量 (インスタンス数または vCPU 数) を設定します。起動テンプレートを指定した場合、ターゲット容量の一部をオンデマンドとして割り当てることができます。オンデマンドインスタンスの数は常に保持されますが、スポットインスタンスはスケールできます。

合計ターゲット容量 ⓘ

1

インスタンス

オプションのオンデマンド部分 [詳細はこちら](#)

ら ↗

0

インスタンス

起動テンプレートを指定するリクエストのみがオンデマンドの対象です

ターゲット容量を維持する

中断動作 ⓘ

終了

- 作成時の注意点：「ターゲット容量を維持する」にチェックを入れる(maintainモードを指定する)  
[https://docs.aws.amazon.com/ja\\_jp/AWSEC2/latest/UserGuide/spot-fleet-target-tracking.html](https://docs.aws.amazon.com/ja_jp/AWSEC2/latest/UserGuide/spot-fleet-target-tracking.html)
  - 「スポットフリート リクエストには、タイプが maintain のリクエストが必要です。」
- 中断などで指定容量を下回った場合にスポットフリートが自動で新しいインスタンスを起動する

# スポットフリートでのApplication Auto Scaling の活用

(3)

The screenshot shows the AWS Management Console interface for a Spot Instance Fleet. At the top, there are navigation tabs: 'スポットインスタンスのリクエスト' (Spot Instance Requests), 'アクション' (Actions), '価格設定履歴' (Price History), and 'Savings Summary'. Below these are search filters for request type, status, and a keyword search box. A table lists the fleet details, with one entry highlighted in blue. Below the table, the specific request ID is shown, followed by a set of tabs: '説明' (Description), 'インスタンス' (Instances), '履歴' (History), '削減額' (Savings), 'Auto Scaling', and 'スケジュールに基づくスケールリング' (Scaling based on schedule). The 'Auto Scaling' tab is highlighted with an orange box. Below the tabs, a message states that Auto Scaling is not configured for this fleet, and a '設定' (Configure) button is highlighted with an orange box. At the bottom, a note explains that CloudWatch alarms can be used to automatically adjust the target capacity within a specified range.

スポットインスタンスのリクエスト    アクション    価格設定履歴    Savings Summary

リクエストタイプ: all    状態: all    キーワードによる検索    << < 1 リクエスト中 1 から 1 を表示 > >>

リクエスト ID	リクエストタ...	インスタンスタ...	状態	容量	ステータス	永続性	作成日	
<input checked="" type="checkbox"/>	▶ sfr-d8948be1-dc6b...	fleet	t3.medium,t2.m...	active	3 of 3	fulfilled	maintain	a day

リクエスト ID: sfr-d8948be1-dc6b-47d6-b1e7-b20212525f78

説明    インスタンス    履歴    削減額    **Auto Scaling**    スケジュールに基づくスケールリング

このフリートに対して Auto Scaling は設定されていません

**設定**

CloudWatch アラームに応じてフリートのターゲット容量を指定範囲内で自動的に調整します。

“Auto Scaling”タブ → 「設定」

# スポットフリートでのApplication Auto Scaling の活用 (4)

リクエスト ID: sfr-d8948be1-dc6b-47d6-b1e7-b20212525f78

説明 インスタンス 履歴 削減額 **Auto Scaling** スケジュールに基づくスケーリング

スケーリング容量の範囲 0 および 30 インスタンス

フリート Auto Scaling 用の IAM ロール [aws-ec2-spot-fleet-autoscale-role](#)

スケーリングポリシー

ポリシー名 Scale fleet to target

ターゲットメトリクス CPU の平均使用率

ターゲット値 50

クールダウン期間 300 スケーリングアクション間の秒数

スケールインの無効化

ステップまたはシンプルなスケーリングポリシーを使用したスポットフリートのスケーリング

保存 キャンセル

- ターゲット追跡スケーリングポリシーの目標値を設定する
- もしくはステップスケーリングポリシーを選択することもできる

# Application Auto Scaling の設定ポイント

- マネジメントコンソールを使う場合と CLI(API, SDK)を使う場合の違い
  - マネジメントコンソールの場合  
各サービスのマネジメントコンソールから設定する
  - CLI(API, SDK)の場合  
使用するAPIはすべてApplication Auto ScalingサービスのAPIである
  - (CLIの例)  
aws application-autoscaling register-scalable-target ¥  
--service-namespace サービス名 ¥  
--resource-id リソースID ¥  
...



# Auto Scalingを試してみる

- EC2 Auto Scalingを試してみる
- Application Auto Scalingを試してみる
- **AWS Auto Scaling – 予測スケーリングを試してみる**

# 下準備 - EC2 Auto Scalingグループの設定変更

Auto Scaling グループ: myasgforssm

詳細 アクティビティ履歴 スケーリングポリシー インスタンス **モニタリング** 通知 タグ スケジュールされたアクション ラ

Auto Scaling メトリックス: **グループメトリックスコレクションを有効にする** 次のデータを表示: 過去 1 時間

表示: Auto Scaling または EC2

**警告**

以下の Auto Scaling グループでは、グループメトリックスコレクションが有効になっていません: myasgforssm

以下は、選択されたリソースの CloudWatch メトリックスです (最大 10)。画面を拡大するには、グラフをクリックします。すべての時刻は協定世界時 (UTC) で表示されています。 [すべての CloudWatch メトリックスを表示](#)

myasgforssm (有効でない)

最小グループサイズ (カウント)	最大グループサイズ (カウント)	希望するキャパシティ (カウント)
1	1	1
0.75	0.75	0.75

EC2 Auto Scaling マネジメントコンソールから  
「モニタリング」 → 「グループメトリックコレクションを有効にする」

# AWS Auto ScalingによるEC2 Auto Scalingグループの予測スケーリング (1)

管理ツール

## AWS Auto Scaling 複数のリソースを迅速かつ 簡単にスケールできるよう 支援します

AWS Auto Scaling では、アプリケーションの基になるすべてのスケーラブルなリソースをすばやく検出し、組み込みのスケーリング推奨項目を使用して数分でアプリケーションのスケーリングをセットアップできます。

この機能の説明

### スケーリングプランの作成

わずか数ステップでアプリケーションを最適化します

今すぐ始める

### 料金表

AWS Auto Scaling は無料です。

AWS Auto Scaling は、Amazon CloudWatch で有効にすることができますが、追加料金は発生しません。アプリケーションリソースおよび Amazon CloudWatch のサービス料金が適用されます。

# AWS Auto ScalingによるEC2 Auto Scalingグループの予測スケーリング (2)

AWS Auto Scaling > スケーリングプラン > スケーリングプランの作成

ステップ 1  
スケーラブルなリソースの検索

## スケーラブルなリソースの検索

自動検出または手動で、スケーリングプランに追加するリソースを選択します。 [情報](#)

ステップ 2  
スケーリング戦略を指定します。

ステップ 3  
詳細設定の設定 (オプション)

ステップ 4  
確認と作成

メソッドの選択

- CloudFormation スタックによる検索  
AWS CloudFormation によってプロビジョニングされたリソースを検索します。
- タグによる検索  
適用されたタグを使用してリソースを検索します。
- Amazon EC2 Auto Scaling グループの選択  
スケーリング計画に含めるための、Auto Scaling グループを 1 つ以上選択します。

「EC2 Auto Scalingグループの選択」を選択

# AWS Auto ScalingによるEC2 Auto Scalingグループの予測スケールリング (3)

AWS Auto Scaling > スケーリングプラン > スケーリングプランの作成

ステップ 1  
スケラブルなリソースの検索

ステップ 2  
スケール戦略を指定します。

ステップ 3  
詳細設定の設定 (オプション)

ステップ 4  
確認と作成

## スケラブルなリソースの検索

自動検出または手動で、スケールプランに追加するリソースを選択します。 [情報](#)

メソッドの選択

- CloudFormation スタックによる検索  
AWS CloudFormation によってプロビジョニングされたリソースを検索します。
- タグによる検索  
適用されたタグを使用してリソースを検索します。
- Amazon EC2 Auto Scaling グループの選択  
スケール計画に含めるための、Auto Scaling グループを 1 つ以上選択します。

### Auto Scaling グループの選択 [情報](#)

Auto Scaling グループ

Auto Scaling グループを選択します。 ▼

🔍 |

- myalbtestsg
- myasgforssm

キャンセル [次へ](#)

既存のAuto Scalingグループを選択して「次へ」

# AWS Auto ScalingによるEC2 Auto Scalingグループの予測スケーリング (4)

AWS Auto Scaling > スケーリングプラン > スケーリングプランの作成

ステップ 1  
スケーラブルなリソースの検索

## スケーリング戦略を指定します。

スケーリング戦略を使用して、アプリケーションのスケーラブルなリソースを最適化する方法を定義します。 [情報](#)

ステップ 2  
スケーリング戦略を指定します。

### スケーリングプランの詳細

名前

myfirstscalingplan

長さは 1~128 文字にする必要があり、パイプ文字「|」、コロン「:」、およびスラッシュ「/」を含めることはできません。

リソース

1 Auto Scaling グループ 選択されました。

ステップ 3  
詳細設定の設定 (オプション)

### Auto Scaling グループ (1)

1 Auto Scaling グループ のためにスケーリング戦略を指定します。

スケーリングプランに含める

ステップ 4  
確認と作成

スケーリングプラン名を入力

# AWS Auto ScalingによるEC2 Auto Scalingグループの予測スケーリング (5)

## Auto Scaling グループ (1)

1 Auto Scaling グループ のためにスケーリング戦略を指定します。

スケーリングプランに含める

### スケーリング戦略

その戦略では、リソースの拡張に使用するスケーリングメトリックとターゲット値を定義します。

可用性を考えた最適化  
高い可用性を提供し、需要の急増に対応できるキャパシティを確保するため、Auto Scaling グループの平均 CPU 使用率を常に 40% に維持します。

可用性とコストのバランスを取ります  
最適な可用性を提供し、コストを削減するため Auto Scaling グループの平均 CPU 使用率を常に 50% に維持します。

コストに合わせて最適化  
確実にコストを削減するため、Auto Scaling グループの平均 CPU 使用率を常に 70% に維持します。

カスタム  
独自のスケーリングメトリック、ターゲット値、およびその他の設定を選択します。

予測スケーリングの有効化  
必要になる前に、継続的にロードし、積極的にスケジュール設定することで、スケーリング戦略をサポートします。 [情報](#)

動的スケーリングの有効化  
スケーリングメトリックを監視し、必要に応じてキャパシティを増減するため、ターゲット追跡スケーリングポリシーを作成して、スケーリング戦略をサポートします。 [情報](#)

▶ 設定の詳細

- 「予測スケーリングの有効化」にチェックが入っていることを確認
- 「動的スケーリングの有効化」のチェックを外す

# AWS Auto ScalingによるEC2 Auto Scalingグループの予測スケーリング (6)

ステップ 1  
スケーラブルなリソースの検索

ステップ 2  
スケーリング戦略を指定します。

ステップ 3  
詳細設定の設定 (オプション)

ステップ 4  
確認と作成

## 詳細設定の設定 (オプション)

個々のリソースまたは複数のリソースの設定を、同時にカスタマイズします。 [情報](#)

▶ Auto Scaling グループ (1)

Auto Scaling グループでは、カスタム設定が使用されます。 予測スケーリングは有効です。

キャンセル

“Auto Scalingグループ”をクリックして展開



# AWS Auto ScalingによるEC2 Auto Scalingグループの予測スケーリング (7)

詳細設定の設定 (オプション)  
個々のリソースまたは複数のリソースの設定を、同時にカスタマイズします。 [情報](#)

▼ Auto Scaling グループ (1 個中 1 個を選択) 元に戻す  
カスタム設定を指定する 1 つ以上の Auto Scaling グループを選択します。

<input checked="" type="checkbox"/>	リソース ▲	プランに含める	外部のスケーリングポリシーの置き換え	既存のポリシー
<input checked="" type="checkbox"/>	myasgforssm	はい	なし	なし

1 個のリソースが選択されました

- スケーリングプランに含める
  - ▶ 全般設定
  - ▶ 動的スケーリングの設定
  - ▶ 予測スケーリング設定**

キャンセル 戻る 次へ

対象Auto Scalingグループを選択すると詳細設定メニューが展開されるので「予測スケーリング設定」を展開する

# AWS Auto ScalingによるEC2 Auto Scalingグループの予測スケーリング (8)

▼ 予測スケーリング設定

**予測スケーリングモード**  
予測の実行にスケーリングを使用するかどうかを決定します。これはいつでも変更できます。 [情報](#)

予測のみ ▼

予測とスケール

予測のみ

ガーします。 [情報](#)

5 分

**最大キャパシティの動作**  
予測キャパシティが最大キャパシティに近づいたか、それを越えたときに使用するルールを選択します。 [情報](#)

最大キャパシティ設... ▼

**予測期間**  
事前予測する日数。 [情報](#)

2 日

**予測の詳細度**  
予測とキャパシティの計算間隔。 [情報](#)

60 分

**予測頻度**  
予測更新の頻度。 [情報](#)

毎日

キャンセル 戻る **次へ**

「予測のみ」を選択して「次へ」

# AWS Auto ScalingによるEC2 Auto Scalingグループの予測スケーリング (9)

ステップ1  
スケーラブルなリソースの検索

ステップ2  
スケーリング戦略を指定します。

ステップ3  
詳細設定の設定 (オプション)

ステップ4  
確認と作成

## 確認と作成

### スケーリングプランの詳細

名前  
myfirstscalingplan

リソース  
1 Auto Scaling グループ 選択されました。1 個のスケーリングポリシーが作成され、0 個の外部ポリシーが維持されます。

### Auto Scaling グループ

スケーリング戦略	スケーリングメトリクス	ターゲット値
可用性を考えた最適化	CPU の平均使用率	40 %

概要  
お客様のスケーリングプランは、40 % で CPU の平均使用率 メトリクスを保持することで、1 Auto Scaling グループを最適化するように設定されています。

動的スケーリング 動的スケーリングは有効です。40 % で CPU の平均使用率 メトリクスを保持する必要に応じて、インスタンスを追加または削除するため、1 ターゲット追跡スケーリングが適用されます。	予測スケーリング 予測スケーリングは有効です。40 % で CPU の平均使用率 メトリクスを保持するために必要なインスタンスの最小数を維持するため、合計 CPU 使用率 メトリクスの予測に基づいて、スケジュールされたスケーリングアクションが生成されます。
---	---

▶ 詳細

キャンセル 戻る **スケーリングプランの作成**

内容を確認して「スケーリングプランの作成」

# AWS Auto ScalingによるEC2 Auto Scalingグループの予測スケーリング (10)

AWS Auto Scaling > スケーリングプラン

スケーリングプラン (1) [情報](#) 🔄 編集 削除 スケーリングプランの作成

<input type="checkbox"/>	名前	ステータス	スケーリングポリシー	作成時刻
<input type="checkbox"/>	<a href="#">myfirstscalingplan</a>	✔️ Active	1	2019-10-02 01:33:59 UTC+0900

「ステータス」が"Active"になったらスケーリングプラン名をクリック

# AWS Auto ScalingによるEC2 Auto Scalingグループの予測スケーリング (11)

The screenshot shows the AWS Auto Scaling console for a scaling plan named 'myfirstscalingplan'. The plan is in an 'Active' state. Below the plan details, there is a section for 'Auto Scaling グループ (1)' containing one group: 'autoScalingGroup/myasgforssm'. This group is also in an 'Active' state. The console displays the scaling metrics for the group, specifically the 'CPUの平均使用率' (Average CPU Utilization), with a target value of 40%. A line graph shows the '合計 CPU 使用率 (%)' (Total CPU Utilization) over time, with values ranging from 0 to 2.67. The graph shows a fluctuating line that remains below the 40% target. The x-axis of the graph is labeled with times: 14:00, 14:30, 15:00, 15:30, 16:00, and 16:30. The y-axis is labeled '合計 CPU 使用率 (%)' and has markers at 0, 1.33, and 2.67. The 'autoScalingGroup/myasgforssm' link is highlighted with an orange box.

myfirstscalingplan

編集 削除

スケーリングプランの詳細

ステータス  
Active

ステータスの説明  
Scaling plan has been created and applied to all resources.

Auto Scaling グループ (1)

autoScalingGroup/myasgforssm

ステータス	スケーリングメトリクス	ターゲット値
アクティブ	CPUの平均使用率	40 %

1 時間 3 時間 12 時間 1 日 3 日 1 週

ダッシュボードに追加

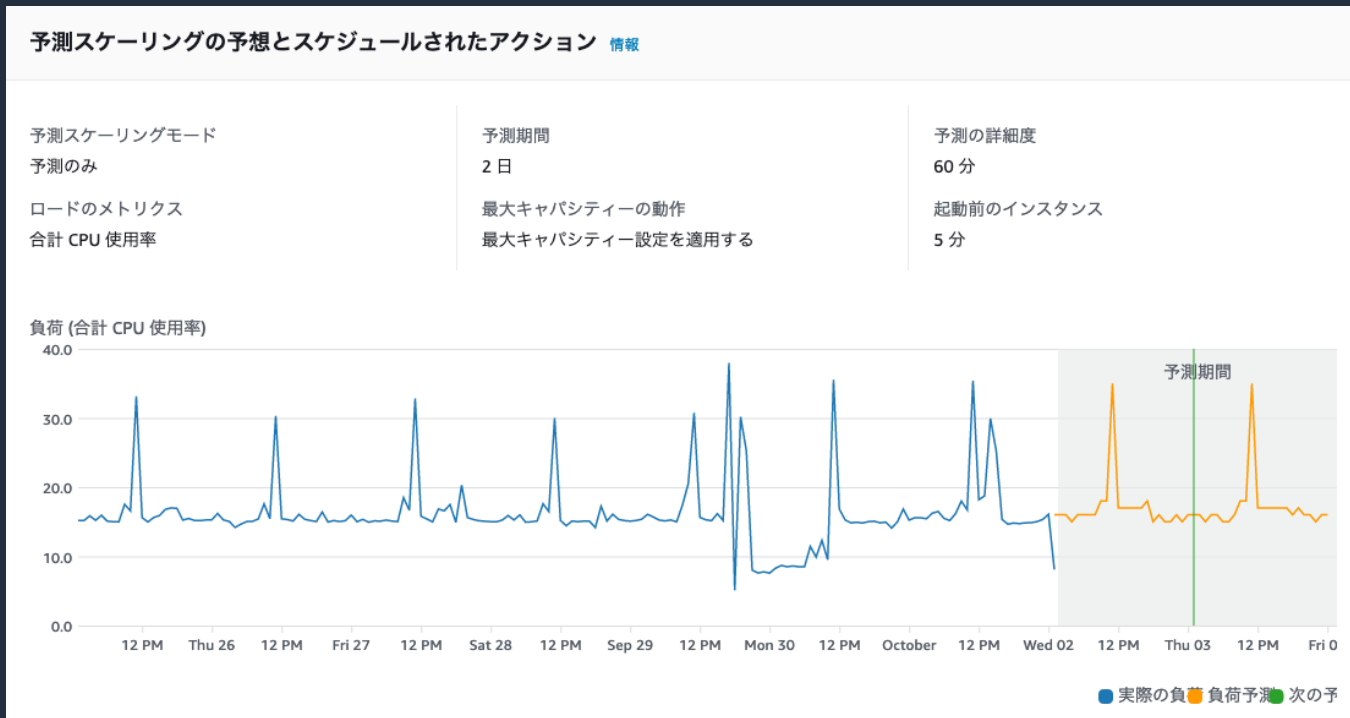
合計 CPU 使用率 (%)

2.67  
1.33  
0

14:00 14:30 15:00 15:30 16:00 16:30

“Auto Scalingグループ”の下の対象Auto Scalingグループリソース名をクリック

# AWS Auto ScalingによるEC2 Auto Scalingグループの予測スケーリング (12)



画面下にスクロールすると、向こう48時間の負荷の予測結果が表示される

# AWS Auto ScalingによるEC2 Auto Scalingグループの予測スケーリング (13)



さらに下にスクロールすると、スケール予定のインスタンス数と  
そのためのスケジュールスケーリング設定の計画を確認できる

# AWS Auto ScalingによるEC2 Auto Scalingグループの予測スケーリング (14)

AWS Auto Scaling > スケーリングプラン > myfirstscalingplan

## myfirstscalingplan

編集

削除

### スケーリングプランの詳細

ステータス

✔ Active

ステータスの説明

Scaling plan has been created and applied to all resources.

- 「予測のみ」モードから「予測とスケーリング」モードへ変更
- 対象スケーリングプランを選択して「編集」



# AWS Auto ScalingによるEC2 Auto Scalingグループの予測スケーリング (15)

AWS Auto Scaling > スケーリングプラン > myfirstscalingplan > 編集

## myfirstscalingplan の編集

▶ Auto Scaling グループ (1)

すべての Auto Scaling グループでは、「可用性を考えた最適化」スケーリング戦略が使用されます。予測スケーリングは有効です。

キャンセル [次へ](#)

“Auto Scalingグループ”をクリックして展開

# AWS Auto ScalingによるEC2 Auto Scalingグループの予測スケーリング (16)

## myfirstscalingplan の編集

▼ Auto Scaling グループ (1 個中 1 個を選択)  
カスタム設定を指定する 1 つ以上の Auto Scaling グループを選択します。

<input checked="" type="checkbox"/>	リソース ▲	プランに含める	外部のスケーリングポリシーの置き換え
<input checked="" type="checkbox"/>	myalbtestasg	はい	なし

1 個のリソースが選択されました

- スケーリングプランに含める
- ▶ 全般設定
- ▶ 動的スケーリングの設定
- ▶ 予測スケーリング設定**

対象Auto Scalingグループを選択すると詳細設定メニューが展開されるので「予測スケーリング設定」を展開する

# AWS Auto ScalingによるEC2 Auto Scalingグループの予測スケーリング (17)

▶ 全般設定

▶ 動的スケーリングの設定

▼ 予測スケーリング設定

予測スケーリングモード  
予測の実行にスケーリングを使用するかどうかを決定します。これはいつでも変更できます。 [情報](#)

予測とスケール ▼

予測とスケール

予測のみ

5 分

最大キャパシティの動作  
予測キャパシティが最大キャパシティに近づいたか、それを超えたときに使用するルールを選択します。 [情報](#)

最大キャパシティ設定を適用する ▼

予測期間  
事前予測する日数。 [情報](#)

2 日

予測の詳細度  
予測とキャパシティの計算間隔。 [情報](#)

60 分

予測頻度  
予測更新の頻度。 [情報](#)

毎日

キャンセル **次へ**

「予測とスケール」を選択して「次へ」

# AWS Auto ScalingによるEC2 Auto Scalingグループの予測スケーリング (18)

AWS Auto Scaling > スケーリングプラン > myfirstscalingplan > 編集

## myfirstscalingplan の編集

### Auto Scaling グループ

スケーリング戦略 可用性を考えた最適化	スケーリングメトリクス CPU の平均使用率	ターゲット値 40 %
------------------------	---------------------------	----------------

**概要**  
動的スケーリングが有効になっている Auto Scaling グループがありません。スケーリングプランで 1 Auto Scaling グループのターゲット追跡スケーリングポリシーを作成できるようにするには、先に動的スケーリングを有効にする必要があります。

<b>動的スケーリング</b> 動的スケーリングは無効です。	<b>予測スケーリング</b> 予測スケーリングは有効です。40% で CPU の平均使用率 メトリクスを保持するために必要なインスタンスの最小数を維持するため、合計 CPU 使用率 メトリクスの予測に基づいて、スケジュールされたスケーリングアクションが生成されます。
-----------------------------------	---

▶ 詳細

キャンセル 戻る **変更の保存**

「変更の保存」

# AWS Auto ScalingによるEC2 Auto Scalingグループの予測スケーリング (19)

Auto Scaling グループ: myalbttestasg

詳細 アクティビティ履歴 スケーリングポリシー インスタンス モニタリング 通知 タグ **スケジュールされたアクション** ライ

予定アクションの作成 操作 ▾

Filter: 🔍 予定アクションのフィルター... ✕

1 から 25 / 46 の スケジュールされたアクション > | <

<input type="checkbox"/>	名前	開始時刻	終了時刻	繰り返し	希望するキャパシティ	最小	最大
<input type="checkbox"/>	AutoScaling-myfirstscalingplan-1-201910011800	2019 October 2 03:00:00 UTC+9				1	5
<input type="checkbox"/>	AutoScaling-myfirstscalingplan-1-201910011900	2019 October 2 04:00:00 UTC+9				1	5
<input type="checkbox"/>	AutoScaling-myfirstscalingplan-1-201910012000	2019 October 2 05:00:00 UTC+9				1	5
<input type="checkbox"/>	AutoScaling-myfirstscalingplan-1-201910012100	2019 October 2 06:00:00 UTC+9				1	5

EC2 Auto Scaling マネジメント コンソール から  
「スケジュールされたアクション」に毎時のアクションが設定されたことを確認

# 本日のアジェンダ

- Auto Scalingサービスのコンセプト
- Auto Scalingの基礎知識
- 主要機能：スケーリングの整理
- Auto Scalingを使ってみる
- こんなときどうする？ - 各種機能の紹介
- まとめ・参考資料

# こんなときどうする？

- (EC2 Auto Scaling) スポットインスタンスを活用したいです
  - →ミックスインスタンスグループを活用してください



# こんなときどうする？

- (EC2 Auto Scaling) 「起動設定」と「起動テンプレート」のどちらを使えば良いか
  - → 「起動テンプレート」を強く推奨します！



# こんなときどうする？

- (EC2 Auto Scaling) 速やかにスケールアウト(スケールイン)してくれません
  - →インスタンスの詳細モニタリングを有効にしてください
- CloudWatch Metricsを1分粒度にする。5分粒度では速やかにスケールできない
- 有料オプションながらAuto Scalingを使用する際のベストプラクティス

[https://docs.aws.amazon.com/ja\\_jp/AWSEC2/latest/UserGuide/using-cloudwatch-new.html](https://docs.aws.amazon.com/ja_jp/AWSEC2/latest/UserGuide/using-cloudwatch-new.html)

# こんなときどうする？

- (EC2 Auto Scaling) 正常に動作しないインスタンスを自動的に置き換えたい
  - →ヘルスチェックを活用します
- 特に指定しない場合、EC2ヘルスチェックが有効になっている
  - 2/2以外のステータスが連続するとAuto Scalingが置き換える
- ELB配下のASGの場合、ELBヘルスチェックを有効にする
  - EC2ヘルスチェックに加え、ELBからのヘルスチェックに応答しない場合の速やかな入れ替えが可能になる

[https://docs.aws.amazon.com/ja\\_jp/autoscaling/ec2/userguide/healthcheck.html](https://docs.aws.amazon.com/ja_jp/autoscaling/ec2/userguide/healthcheck.html)

# こんなときどうする？

- (EC2 Auto Scaling)スケールイン・スケールアウトを繰り返してしまい、いつまでたってもインスタンスが追加されない
  - → 「ヘルスチェックの猶予期間」の設定を見直す
- ヘルスチェックの猶予期間：起動したばかりでヘルスチェックに応答できないインスタンスを保護する期間
  - /index.html などは速やかに返せるようになるが、S3からのコンテンツ配備やDB接続などが整った前提のヘルスチェックパスを指定している場合は準備期間が必要
  - 特にELBヘルスチェックにアプリケーションのパスを採用している場合に有効
- デフォルトは5分(300秒)

[https://docs.aws.amazon.com/ja\\_jp/autoscaling/ec2/userguide/healthcheck.html](https://docs.aws.amazon.com/ja_jp/autoscaling/ec2/userguide/healthcheck.html)

# こんなときどうする？

- (EC2 Auto Scaling) 次にどのインスタンスがスケールイン対象になるか知りたい
  - →デフォルトの終了ポリシー
- おおまかには次の流れで決まる
  1. インスタンスが最も多いアベイラビリティゾーンを選択
  2. (そのアベイラビリティゾーンに候補が複数あるなら) 最も古い起動設定・起動テンプレートから起動されたインスタンスを選択
  3. (複数候補が残っている場合) 次のインスタンス時間に近いものを選択
  4. まだ複数いるならランダム
- カスタマイズも可能

[https://docs.aws.amazon.com/ja\\_jp/autoscaling/ec2/userguide/as-instance-termination.html](https://docs.aws.amazon.com/ja_jp/autoscaling/ec2/userguide/as-instance-termination.html)

# こんなときどうする？

- (EC2 Auto Scaling) 特定のインスタンスをスケールインから保護したい
  - →インスタンスの保護
- ASG単位、もしくはインスタンス単位で設定。スケールインされなくなる
- 次の条件からは保護できないことに注意
  - 手動でのインスタンス削除(Terminate)
  - ヘルスチェックによる置き換え
  - スポットインスタンスの中断
- すべてのインスタンスが終了保護された状態でスケールインイベントが発生した場合、希望容量だけが減少し、スケールイン(インスタンス削除)は行われない

[https://docs.aws.amazon.com/ja\\_jp/autoscaling/ec2/userguide/as-instance-termination.html#instance-protection](https://docs.aws.amazon.com/ja_jp/autoscaling/ec2/userguide/as-instance-termination.html#instance-protection)

# こんなときどうする？

- (EC2 Auto Scaling) 一時的にスケールインやスケールアウトを止めたい
  - →スケーリングプロセスの中断
- 一時的にスケール動作を停止できる
- ASG単位で設定
- 中断できるプロセス一覧：Launch, Terminate, AddToLoadBalancer, AlarmNotification, AZRebalance, HealthCheck, ReplaceUnhealthy, ScheduledActions
- 使いどころ：機能テストなど、一時的にAuto Scalingグループの特定プロセスの動作を止めてテスト条件を整えたい場合
  - LaunchとTerminateの両方のプロセスを中断することで、「何もしない」Auto Scalingグループを作り出せる
- 動作のおかしいインスタンスがあるのでスケールイン・スケールアウトを止めたい
  - →プロセスの中断ではなく次の項目を参照

[https://docs.aws.amazon.com/ja\\_jp/autoscaling/ec2/userguide/as-suspend-resume-processes.html](https://docs.aws.amazon.com/ja_jp/autoscaling/ec2/userguide/as-suspend-resume-processes.html)

# こんなときどうする？

- (EC2 Auto Scaling) このインスタンスをAuto Scalingグループから外したい
  - →スタンバイ、もしくはデタッチ
- スタンバイ(「一時的なインスタンスの削除」)
  - インスタンス単位で設定
  - そのインスタンスはAuto Scalingグループにしながら「スタンバイ」状態に入る
    - 具体的にはそのインスタンスはELBから登録解除され、ヘルスチェック対象から外される。  
そのAuto Scalingグループの希望容量は1つ減少する
  - その間にインスタンスのトラブルシューティングなどを行う
- デタッチ
  - インスタンス単位で設定
  - そのインスタンスはそのAuto Scalingグループのメンバーから外れる
  - スタンバイと実質的な効果は同一。インスタンスはそのままRunning状態で保持される。ただしデタッチの場合、Auto Scalingグループとして与えていたタグも除去される

[https://docs.aws.amazon.com/ja\\_jp/autoscaling/ec2/userguide/as-enter-exit-standby.html](https://docs.aws.amazon.com/ja_jp/autoscaling/ec2/userguide/as-enter-exit-standby.html)

[https://docs.aws.amazon.com/ja\\_jp/autoscaling/ec2/userguide/detach-instance-asg.html](https://docs.aws.amazon.com/ja_jp/autoscaling/ec2/userguide/detach-instance-asg.html)

# こんなときどうする？

- (EC2 Auto Scaling) スケールアウトした後、サービス開始前にインスタンスに準備させたい / スケールインの前にログ退避させたいのでTerminateを少し待つて欲しい
  - →ライフサイクルフック
- ライフサイクルフック：インスタンス起動時・削除時にインスタンスを一時停止し、カスタムアクションを実行できる
- ライフサイクルフックはAuto Scalingグループ単位に設定
- 実際のライフサイクルフックによる待機はインスタンスごと
- 実装例：CloudWatch Eventからライフサイクル通知を受け取り、Lambdaがカスタムアクションを実行する

[https://docs.aws.amazon.com/ja\\_jp/autoscaling/ec2/userguide/lifecycle-hooks.html](https://docs.aws.amazon.com/ja_jp/autoscaling/ec2/userguide/lifecycle-hooks.html)



# こんなときどうする？

- (EC2 Auto Scaling) スケールアウトを素早くしたい
  - →ユーザーデータでのyum updateやyum installなどを用いず、なるべくミドルウェアや必要設定などを済ませた状態のAMIを起動テンプレートに指定する(いわゆるゴールデンイメージ)
  - ただし考慮点があるので次のスライドで説明

# こんなときどうする？

- (EC2 Auto Scaling) WindowsやRed Hat Enterprise Linuxなどの考慮点は？
  - 1時間単位の課金になるため、終了ポリシーはデフォルトでお使いいただくのをお勧めする
  - 起動時間を短縮する際、ゴールデンイメージの起動時間と、標準AMIからの起動+ユーザーデータでセットアップした場合とを比較すると良い
    - 2019年現在、標準AMIはカスタマイズしたAMIより素早く起動できるようにチューニングされている
    - 場合によってはユーザーデータの方が速い可能性も

# 本日のアジェンダ

- Auto Scalingサービスのコンセプト
- Auto Scalingの基礎知識
- 主要機能：スケーリングの整理
- Auto Scalingを使ってみる
- こんなときどうする？ - 各種機能の紹介
- **まとめ・参考資料**

# 本日のまとめ

- Auto Scalingの価値
  - アプリケーションの可用性の維持
    - アベイラビリティゾーン間でのインスタンスの分散、異常なインスタンスの自動置き換え
  - 自動的なキャパシティの増減
    - 動的なスケーリング、予測スケーリング、スケジューリングスケーリング
    - 予測スケーリングとターゲット追跡スケーリングの組み合わせは2019年におススメする推奨セット
  - 様々なユースケースをカバーする機能群
    - コスト最適化のためのミックスインスタンсグループ、ライフサイクルフック
- Auto Scalingを使いこなし、クラウドの世界の本質をぜひ実感してください

# 参考資料

## よくある質問

- よくある質問 - Amazon EC2 Auto Scaling | AWS — <https://aws.amazon.com/jp/ec2/autoscaling/faqs/>
- よくある質問 - AWS Auto Scaling | AWS — <https://aws.amazon.com/jp/autoscaling/faqs/>

## ユーザーガイド

- Amazon EC2 Auto Scaling とは - Amazon EC2 Auto Scaling (日本語) — [https://docs.aws.amazon.com/ja\\_jp/autoscaling/ec2/userguide/what-is-amazon-ec2-auto-scaling.html](https://docs.aws.amazon.com/ja_jp/autoscaling/ec2/userguide/what-is-amazon-ec2-auto-scaling.html)
- Application Auto Scaling とは - Application Auto Scaling — [https://docs.aws.amazon.com/ja\\_jp/autoscaling/application/userguide/what-is-application-auto-scaling.html](https://docs.aws.amazon.com/ja_jp/autoscaling/application/userguide/what-is-application-auto-scaling.html)
  - 各サービスでのApplication Auto Scalingの使い方・考慮点は以下のリンクから
  - ご利用開始にあたって - Application Auto Scaling — [https://docs.aws.amazon.com/ja\\_jp/autoscaling/application/userguide/what-is-application-auto-scaling.html#getting-started](https://docs.aws.amazon.com/ja_jp/autoscaling/application/userguide/what-is-application-auto-scaling.html#getting-started)
- AWS Auto Scaling とは - AWS Auto Scaling — [https://docs.aws.amazon.com/ja\\_jp/autoscaling/plans/userguide/what-is-aws-auto-scaling.html](https://docs.aws.amazon.com/ja_jp/autoscaling/plans/userguide/what-is-aws-auto-scaling.html)

# Q&A

お答えできなかったご質問については

AWS Japan Blog 「<https://aws.amazon.com/jp/blogs/news/>」にて  
後日掲載します。

# AWS の日本語資料の場所「AWS 資料」で検索



日本担当チームへお問い合わせ サポート 日本語 ▾ アカウント ▾

コンソールにサインイン

製品 ソリューション 料金 ドキュメント 学習 パートナー AWS Marketplace その他 🔍

## AWS クラウドサービス活用資料集トップ

アマゾン ウェブ サービス (AWS) は安全なクラウドサービスプラットフォームで、ビジネスのスケールと成長をサポートする処理能力、データベースストレージ、およびその他多種多様な機能を提供します。お客様は必要なサービスを選択し、必要な分だけご利用いただけます。それらを活用するために役立つ日本語資料、動画コンテンツを多数ご提供しております。(本サイトは主に、AWS Webinar で使用した資料およびオンデマンドセミナー情報を掲載しています。)

[AWS Webinar お申込 »](#)

[AWS 初心者向け »](#)

[業種・ソリューション別資料 »](#)

[サービス別資料 »](#)

<https://amzn.to/JPArchive>







# ご視聴ありがとうございました

AWS 公式 Webinar

<https://amzn.to/JPWebinar>



過去資料

<https://amzn.to/JPArchive>

