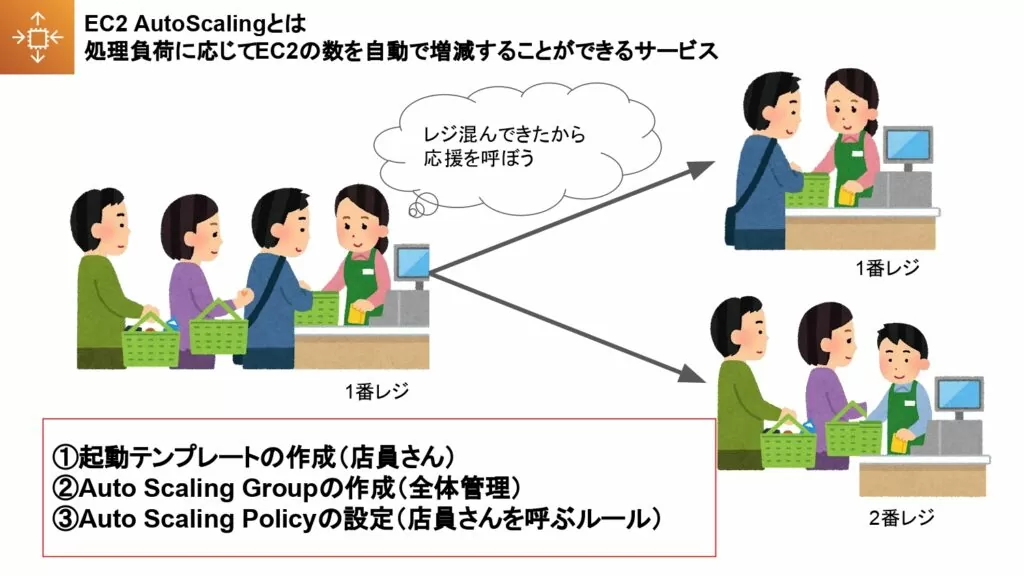
EC2 AutoSclingとは端的にいうと処理負荷に応じて

自動でインスタンスを増減することができるサービス

分かりやすく画像で示すとこのような感じになる



**AutoScalingの特徴**

例えばEC2が1台だけのシステムで導入時から長い年月の間に利用者が増えて

業務時間帯に処理しきれない状態になったとして

現状でもこのように支障が出ていて、このまま放置するとさらに利用者が増えてどんどん処理が遅くなってしまい、業務に多大な影響が出てしまう状況。

**要点をまとめると**

・EC2が1台だけで現在利用者が増えて、処理がしきれなくなっている

・このまま放置しても利用者は更に増えていく見込み

・最終的に業務に多大な影響を及ぼしてしまう可能性が高い

そこでAutoScaling機能を使用すると、EC2の負荷に応じて自動的にスケールアップすることができるため、1台で処理しきれなかった処理を2台、3台と複数台のサーバーで分散処理をすることにより、急激に負荷が掛かったとしても1台あたりの負荷を軽減させることができる

**要点をまとめると**

・EC2の負荷に応じて自動的に分散させる機能

・分散させるので1台あたりの負荷が軽くなる

**疑問点**

ここで思う疑問は最初からEC2の台数を増やしてしまえばいいじゃないか、という疑問

**疑問点を調べたまとめ**

・常時必要以上の台数で運用すると利用者のいない時間帯はリソースが無駄になってしまう

・リソースが無駄になるということは費用が高くなってしまう

・AutoScaling機能を使用するとEC2負荷が低くなった場合に自動的にスケールダウンしてくれる、必要な時にスケールアップし、必要でないときはスケールダウンをしてくれるので

必要最低限のリソースに無駄がない運用ができるのが大きな特徴。

**AutoScalingの料金**

・AutoScaling自体の料金は無料、アプリケーションを実行するのに必要なリソースの料金とAmazon CloudWatchモニタリングには料金が掛かる。

どちらも使用量に応じて価格が変動する従量課金制なので過度な支出を抑えられる。

**AutoScalingのメリット**

・突発的な大量アクセスにも耐えれるシステムを構築できる（単体ではむり

・アクセスが少ないときは使用するサーバー台数が自動的に減らせて無駄なコストがかからない

・運用リソースや機器購入などの大幅なコストダウンが可能

**AutoScalingの判定基準**

AutoScalingは事前に実行するための判定基準を定める必要がある

・サーバーのCPUやメモリが高使用率な状態がどれだけ継続したか

・ネットワークを通してサーバーにどれだけ送られてきたか

などが判定基準となる

利用するサーバーの台数、使用時間によってクラウドサービスの料金が決まる為、判定基準の最適化が重要になる。

**突発的なアクセス増加には即時対応はできない**

・一般的なオンラインサービスを展開していると、予測もつかない急激なトラフィック増加でアプリの処理が追いつかなくなる時がある。

要は炎上したSNSの書き込みや人気バンドのチケット予約の予約サイトのアクセス集中など、まさに”重い”って意味

一見オートスケールにしておけばユーザーが増えるたびにリソースを増やして対応できそうだけども、オートスケールだけでは突発的なアクセス増加などに即時に（例えば数秒）対応には困難。

なぜならインスタンスを起動させてアプリを動作させるには数分のリードタイムが必要だから。

急激なアクセス増加に備えるなら、高速コンテンツ配信ネットワーク（ＣＤＮ）サービスであるAmazon CloudFrontを導入したり

処理性能の上限を設けるためにAmazon API Gatewayを利用したりするなど別サービスで対応したりする

**ELBとAutoScalingの違い**

**ELB**

・ロードバランサーの役割を果たし、トラフィックを複数のバックエンドサーバーに分散する

・クライアントからリクエストを受け取り、バックエンドサーバーに渡すための負荷分散アルゴリズムを使用する

・ヘルスチェックを行い、正常である、バックエンドサーバーにのみトラフィックをルーティングする

**AutoScaling**

・アプリの需要に応じてインスタンスの数を自動的に増減（スケーリング）する

・インスタンスの起動を終了を制御し、可用性やパフォーマンスの確保を支援する

・バックエンドのＥＣ２インスタンス、ＲＤＳデータベースなど様々なＡＷＳリソースに対しAuto Scalingを構成できる

用語集

スケールアップ　サーバーのCPUやメモリーなどをスペックアップして処理性能を高める方法

スケールアウト　サーバーの台数を増やして性能を高める方法

スケールイン　システムを構成するサーバーの台数を減らす事

スケールダウン　サーバーのCPUやメモリーなどをスペックダウンして最適化を行う方法

**高速コンテンツ配信ネットワーク（ＣＤＮ）**

Webコンテンツやその他のデジタルコンテンツを効率的に配信するための分散型のインフラストラクチャ

CDNは世界中に配置された複数のサーバーを使用して、ユーザーに近い場所からコンテンツを提供することにより、高速で信頼性の高いコンテンツ配信を実現する

**Amazon CloudFront**

グローバルなCDNサービス、画像やビデオ、静的ファイル、動画コンテンツなどのWebコンテンツを低遅延でユーザーに配信する

世界中に分散したエッジロケーションのネットワークを活用し、コンテンツをユーザーに最も近いエッジロケーションからキャッシュして提供する、これにより遅延を減らしている

**Amazon API Gateway**

Application Programming Interface）の作成、管理、公開を簡単に行うためのサービス。

バックエンドの様々なリソース、Lamda関数、EC2、などとシームレスに統合できる

リクエストとレスポンスのデータを変換するための機能を提供する、リクエストや

レスポンスのパラメータの追加、削除、変更などを行う事ができ

**シームレス**

何かが滑らかで連続的な状態や経験を提供すること