AWS Black Belt Online Seminar

【AWS Black Belt Online Seminar】 Elastic Load Balancing (ELB)

アマゾン ウェブ サービス ジャパン株式会社
ソリューションアーキテクト 辻 正史
2016.10.12

amazon web services

AWS Black Belt Online Seminar とは

- AWSJのTechメンバがAWSに関する様々な事を紹介するオンラインセミナーです

【**火曜 12:00~13:00**】
主にAWSのソリューションや
業界カットでの使いどころなどを紹介
(例：IoT、金融業界向け etc.)

【**水曜 18:00~19:00**】
主にAWSサービスの紹介や
アップデートの解説
(例：EC2、RDS、Lambda etc.)

※最新の情報は下記をご確認下さい。
オンラインセミナーのスケジュール&申し込みサイト
- http://aws.amazon.com/jp/about-aws/events/#webinar

内容についての注意点

- 本資料では2016年10月12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います。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.

amazon web services

Agenda

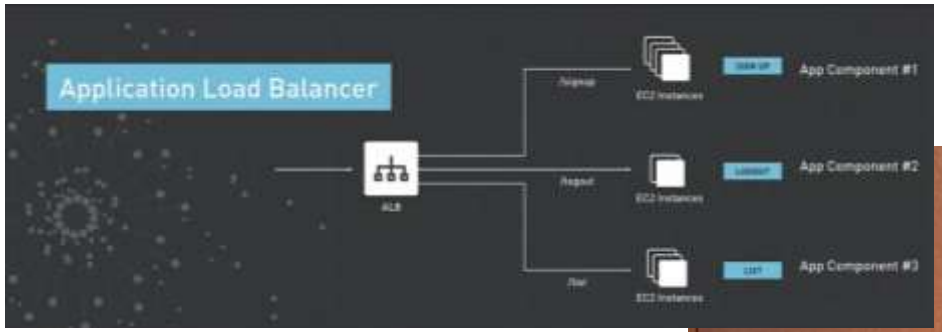
- ALBリリースと従来のELB
- ELBの基本
- 各種機能
  - ELBの機能
  - CLBの機能
  - ALBの機能
- ELBと各種サービスの連携
- ELB負荷試験時のTips
- まとめ

amazon web services

ALBリリースと従来のELB

amazon web services

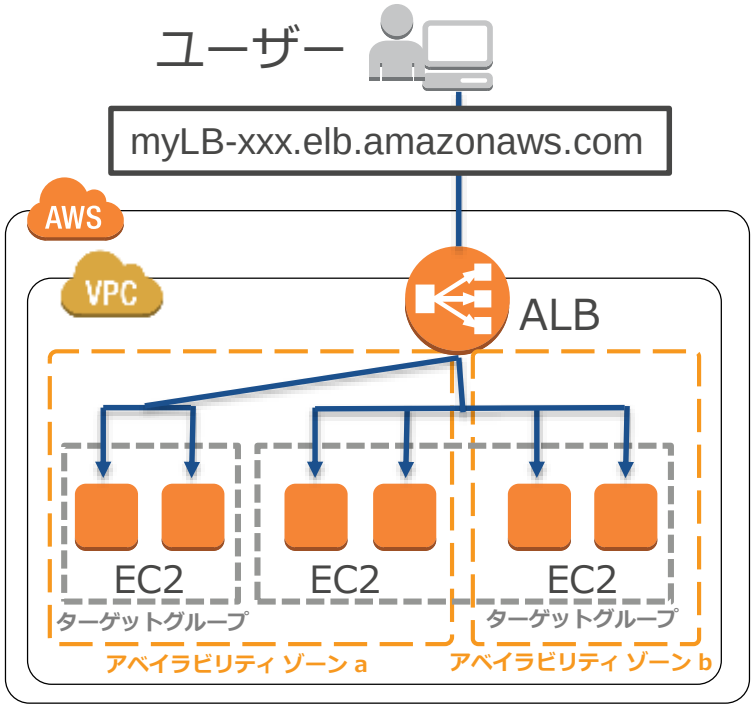
AWS Summit New York 2016 - Keynote
Werner Vogels が “Application Load Balancer” を発表!!



https://www.youtube.com/watch?v=b7yqd7z1RBQ

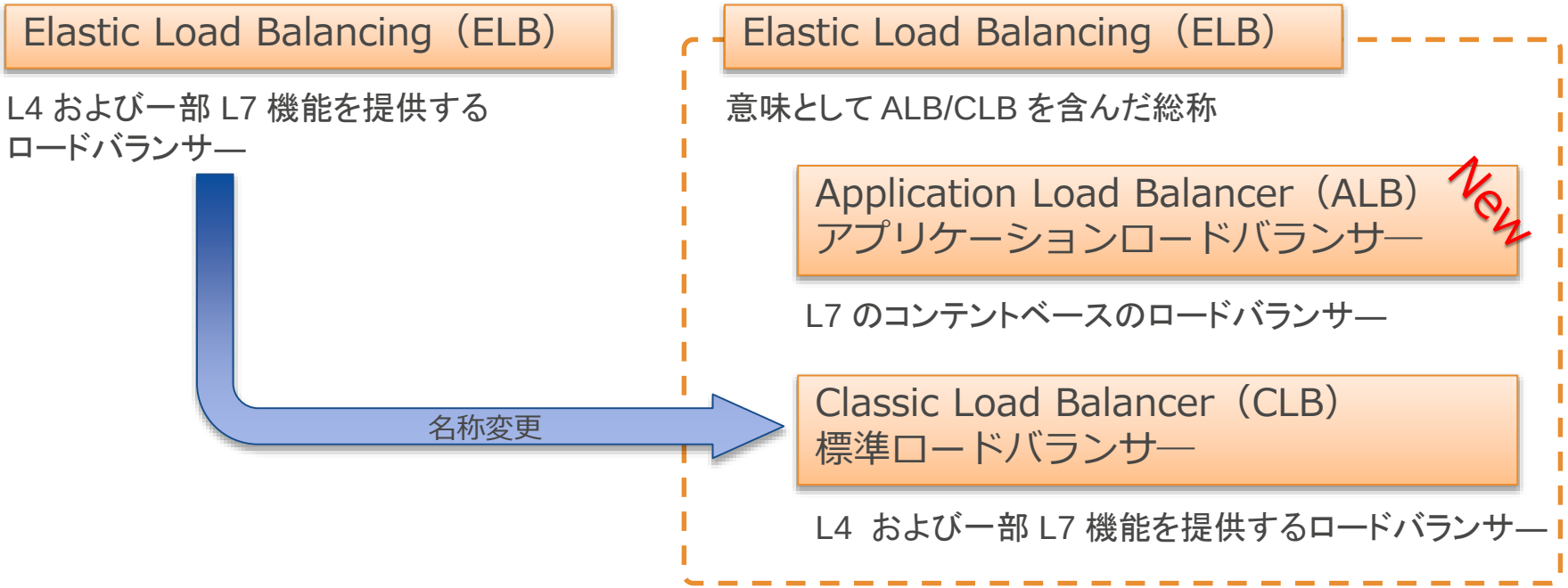
Application Load Balancer (ALB)

レイヤー7のコンテントベースのロードバランサー



- **特徴** (https://aws.amazon.com/elasticloadbalancing/applicationloadbalancer/)
  - レイヤー7のコンテントベースで、ターゲットグループに対してルーティング
  - コンテナベースのアプリケーションのサポート
  - WebSocket と HTTP/2 のサポート
  - 複数のアベイラビリティゾーンに跨って、高レベルの耐障害性を実現
  - ALB自体が自動的にキャパシティを増減
- **価格体系**
  (https://aws.amazon.com/jp/elasticloadbalancing/applicationloadbalancer/pricing/)
  - ALBの起動時間
  - Load Balancer Capacity Units (LCU)の使用量

従来の ELB の位置づけ



amazon web services

ELBの基本

amazon web services

Elaastic Load Balancing (ELB) とは？

～ AWSクラウド上のロードバランシングサービス ～

ELBで実現できるシステム

- **スケーラブル**：複数のEC2インスタンス/ECS Service に負荷分散
- **高い可用性**　：複数のアベイラビリティゾーンにある複数のEC2インスタンス
ECS Service の中から正常なターゲットにのみ振り分け

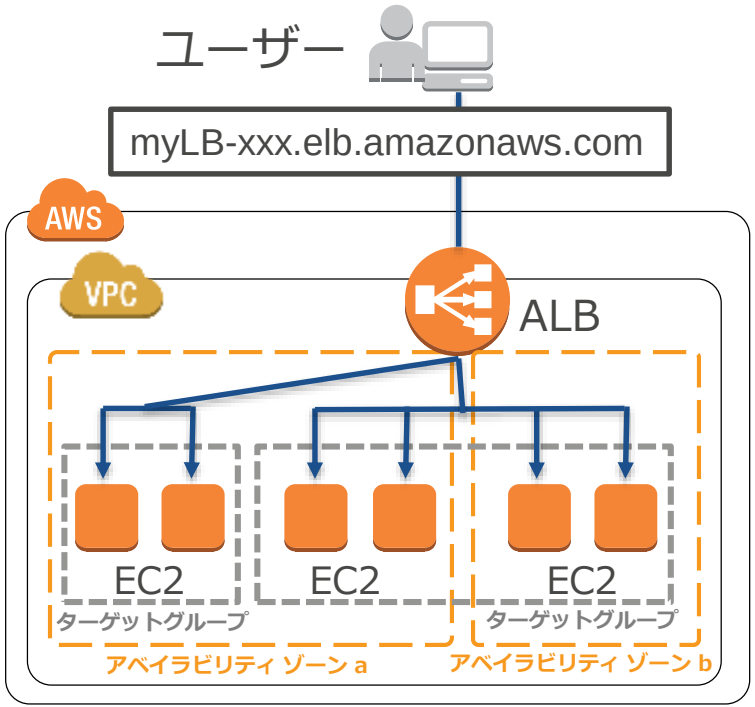
ELB自体の特徴

- **スケーラブル**　：ELB自体も負荷に応じてキャパシティを自動増減
- **安価な従量課金**：従量課金で利用可能
- **運用管理が楽**　：マネージドサービスなので管理が不要
- **豊富な連携機能**：Auto Scaling, Route 53, Cloud Formation… などと連携

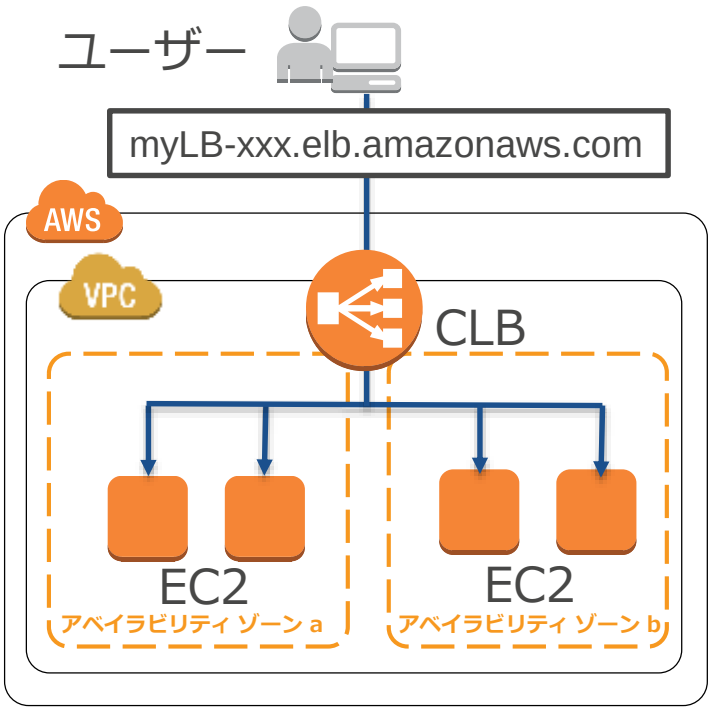
amazon web services

ELBの使い方イメージ

Application Load Balancer ALB



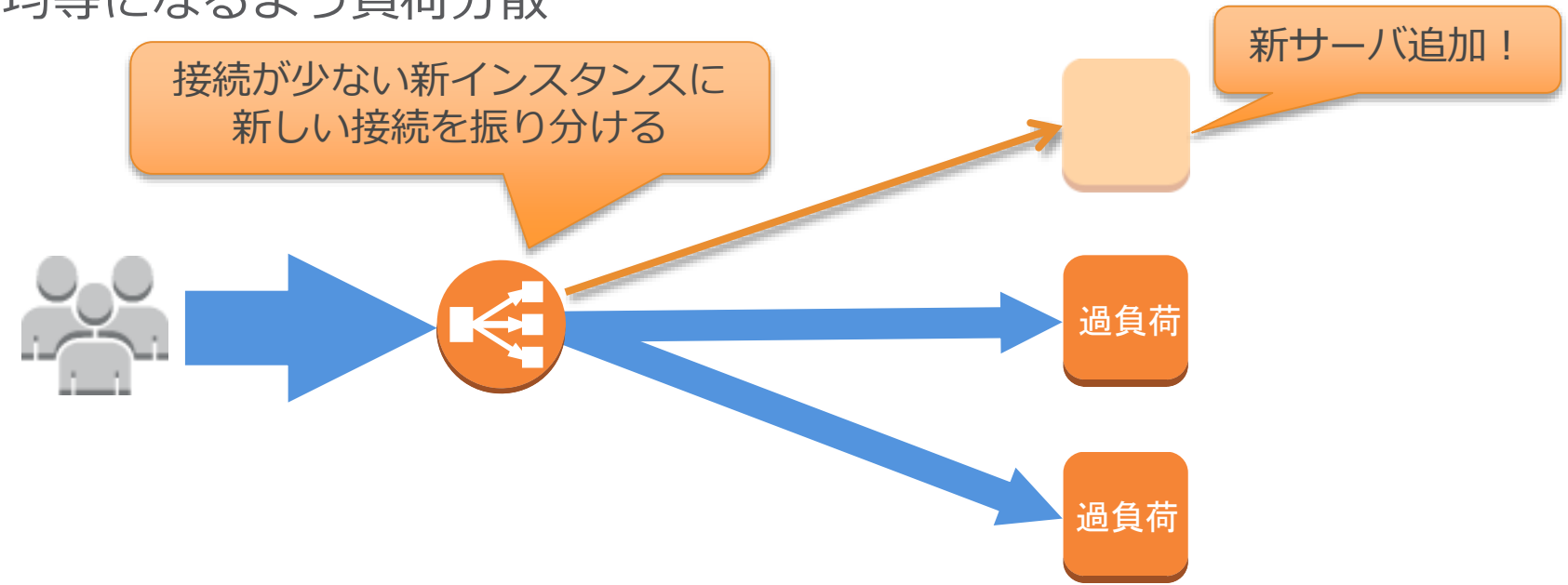
Classic Load Balancer CLB



amazon web services

負荷分散してスケーラブルなシステムを

バックエンドのEC2インスタンスのリクエスト数やコネクション数が均等になるよう負荷分散



ELB自体もスケーラブル

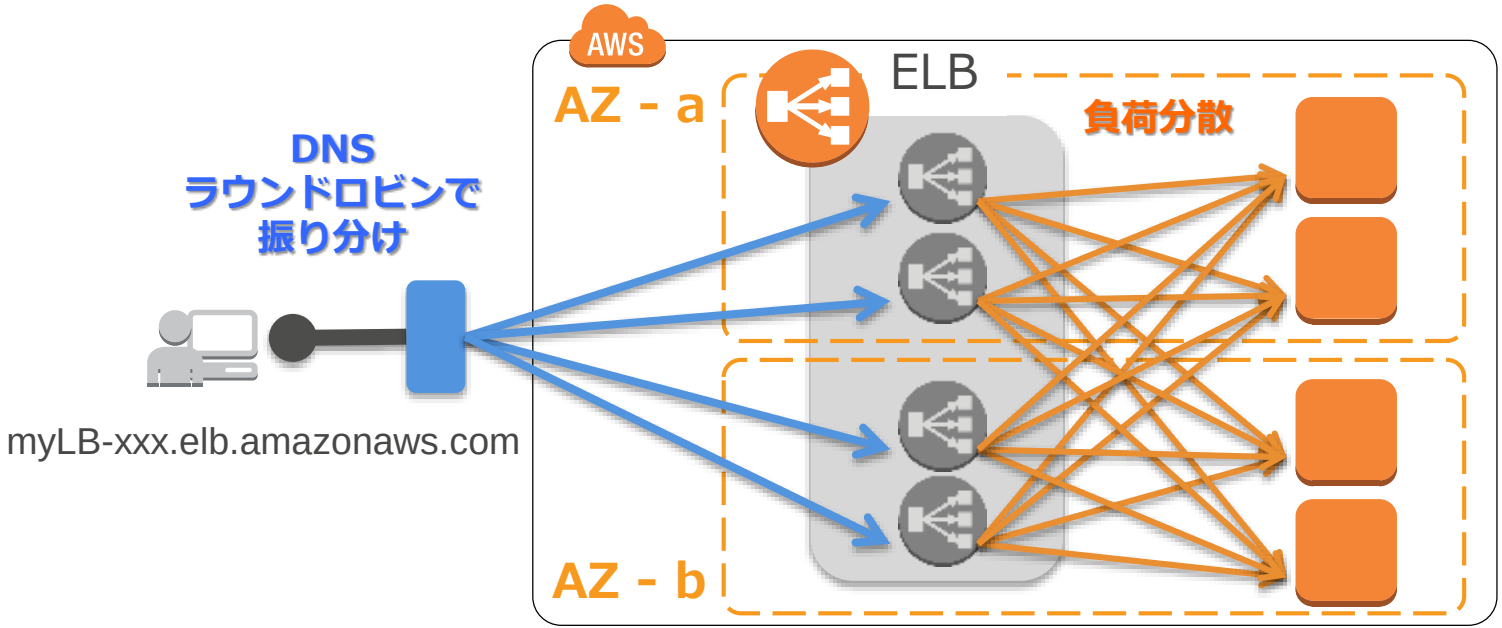
- ELB自体も負荷の増減に応じて自動でスケール
  （キャパシティが自動で増加する）

[注意]
ELBがスケールするときには、IPアドレスが変化します。
ELBへアクセスするときには必ずDNS名で！
DNSに登録することで独自ドメインでのアクセスも可能。

amazon web services

複数アベイラビリティゾーン(AZ)に分散

2段階で負荷分散
1) DNSラウンドロビンで各AZ内のELBに振り分け
2) 負荷が均等になるようにバックエンドのEC2に振り分け



ヘルスチェック

- ELBは指定した設定に基づき、バックエンドのインスタンスのヘルスチェックを行う
  - Pingプロトコル　例：HTTP (200番が返るのを確認)
  - Pingポート　　例：80番
  - Pingパス（HTTP/S利用時）例：/index.html
  - タイムアウト時間　　例：20秒
  - ヘルスチェック間隔　　例：30秒
  - 異常判定までの回数　　例：4回
  - 正常判定までの回数　　例：2回
- 正常との判定が遅いと追加したインスタンスが使えるまでに時間がかかる。逆に異常との判定が厳しすぎても、過負荷時に処理できるインスタンスを減らしてしまうことにも。

安価な従量課金

- 時間当たりの利用料は、複数AZ配置構成でも同じ
- ELBにリザーブドプランはなし
- 処理量の課金は
  - 合計処理量 CLB
  - Load Balancer Capacity Units (LCU) ALB

amazon web services

各種機能

amazon web services

ELBの機能

- ELB利用時のTips
  - 独自ドメイン名で利用
  - クライアントのIPアドレス取得
  - AZとバックエンドキャパシティの関係
  - ELBとバックエンドとのコネクション
  - ELB自身のスケーリング
  - VPCでの利用
  - ELBとSecurity Group
- SSLサポート
- スティッキー セッション
- Connection Draining
- アクセスログのS3保管

amazon web services

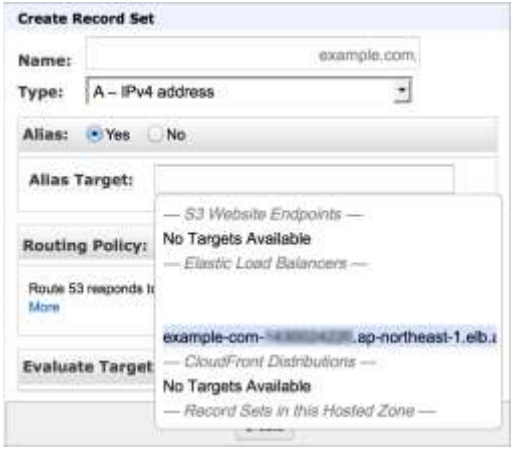
独自ドメイン名で利用

- Route 53以外のDNSを使用する場合はCNAMEで登録

```
www.example.com CNAME myLB-xxxx.ap-northeast-1.elb.amazonaws.com
```

- Route 53を使用する場合は、Route 53エイリアスレコードで登録
  （CNAMEでの登録も可能）

- Zone Apex (www.exapmple.com ではなく example.com を指定) の場合
  → 通常のDNSサーバではCNAME設定不可
  → Route 53のエイリアスレコードを使うことで実現可能



クライアントのIPアドレス取得

バックエンドサーバへの接続は、
ソースIPアドレスがELBのIPアドレスとなる

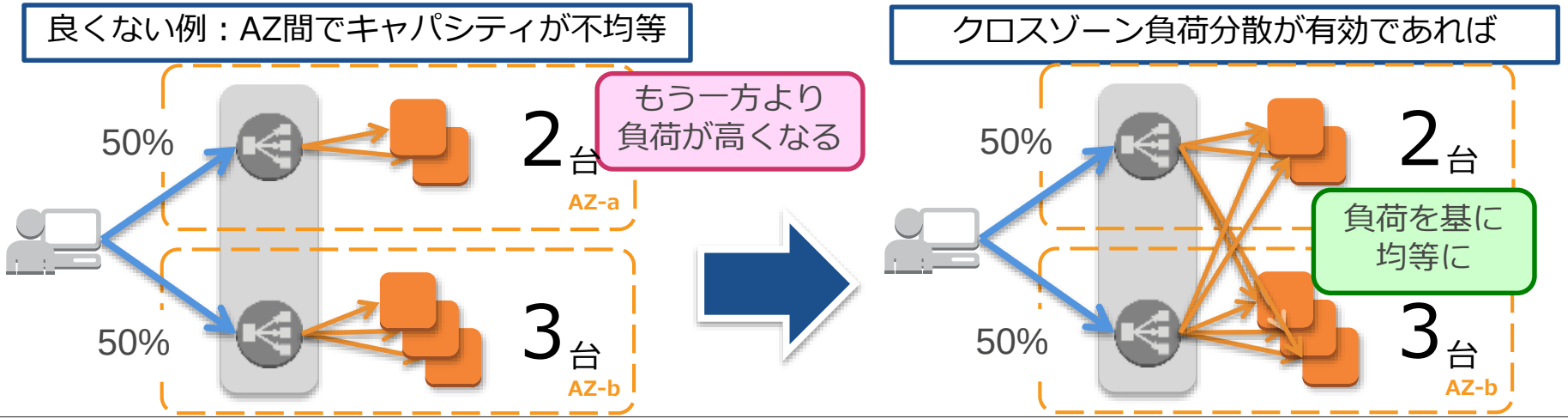
- クライアント⇔ELB, ELB⇔バックエンドの接続はそれぞれ独立しているため
- アクセスログにはELBのIPアドレスが記録される

→ HTTP/HTTPSならHTTPヘッダ上のX-Forwarded-Forで参照可

利用例： - アクセスログにクライアントのIPアドレスを記録
- IPアドレスによるアクセス制限

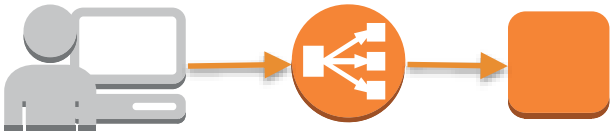
AZとバックエンドキャパシティの関係

- リージョン内の複数AZに負荷分散可能
  - 複数リージョンへの分散にはRoute 53を併用できる
- 各EC2インスタンスのタイプは同じに
- AZごとのEC2インスタンス数はできれば均等に
  - クロスゾーン負荷分散でAZ間を越えて負荷を均等にできる



ELBのコネクションタイムアウト

- 無通信状態が続くとそのコネクションを自動で切断する
  - デフォルトではコネクションタイムアウト値は60秒
- コネクションタイムアウト値を変更可能
  - 1~3600秒の間で自由に設定可能



ELB自体のスケーリング

ELBは負荷に応じて自動でスケールする

- 但し、ELBへの接続・リクエストが瞬間的に急増したために、ELBのスケーリングが間に合わない場合、HTTP 503を返す
  - 新サービス開始
  - TVやメディアによるサービス紹介
  - 負荷テスト　　等

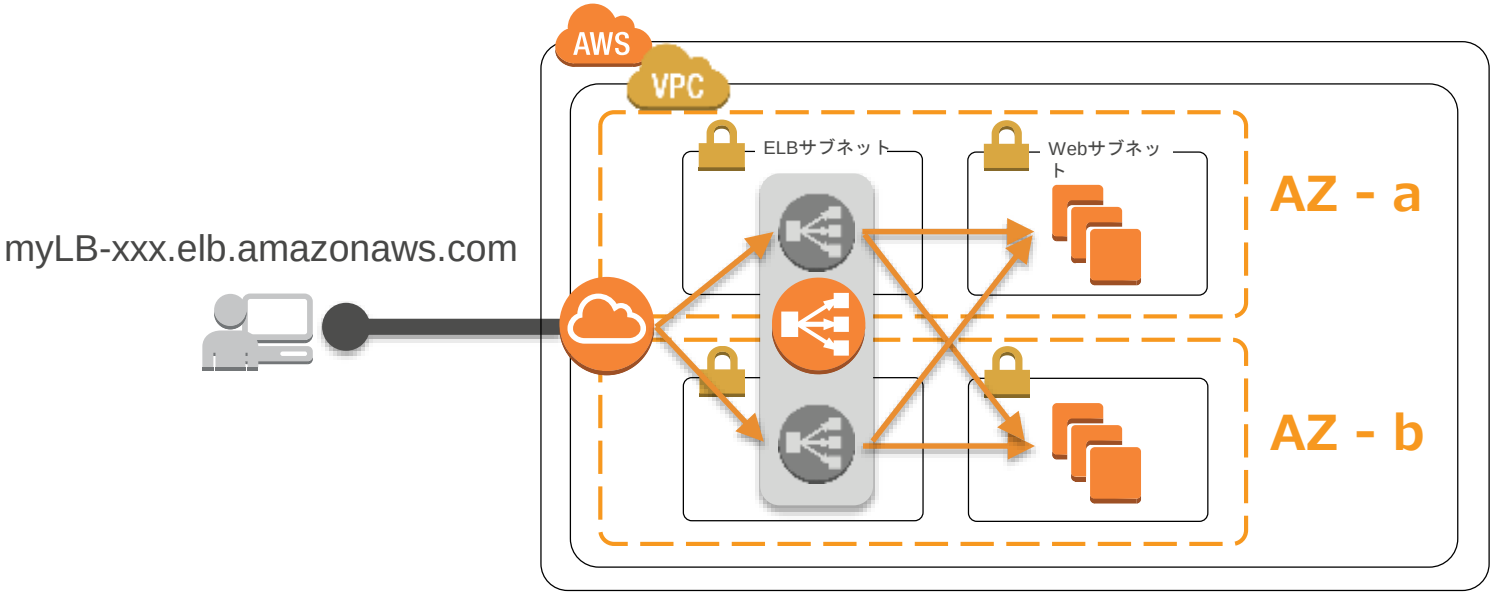
- 回避方法は事前にELBをスケールさせておく
  - Pre-Warming（暖機運転）の申請を行う　※Business/Enterpriseサポート要
  - 負荷を段階的にかける

参照 http://d36cz9buwru1tt.cloudfront.net/jp/documentation/BestPracticesInEvaluatingELB-ja-final.pdf

amazon web services

VPCでの利用

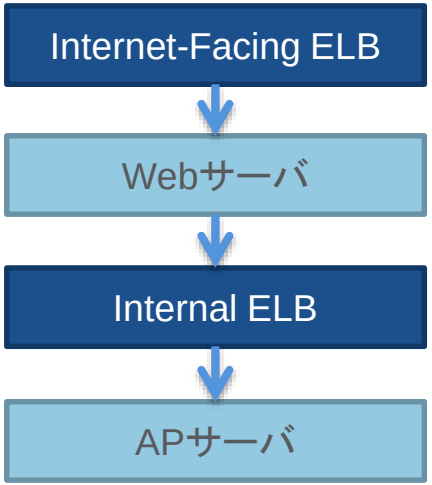
- ELBを配置するサブネットをAZごとに1つ指定
  サブネットは最小 /27 CIDRブロックで、 8 個以上の空きIPが必要



Internet-Facing ELB / Internal ELB

- Internet-Facing ELB
  インターネットからアクセスできるELB

- Internal ELB
  VPC内やオンプレミス環境からのみ
  アクセスできればよいELB
  プライベートサブネットにも配置できる

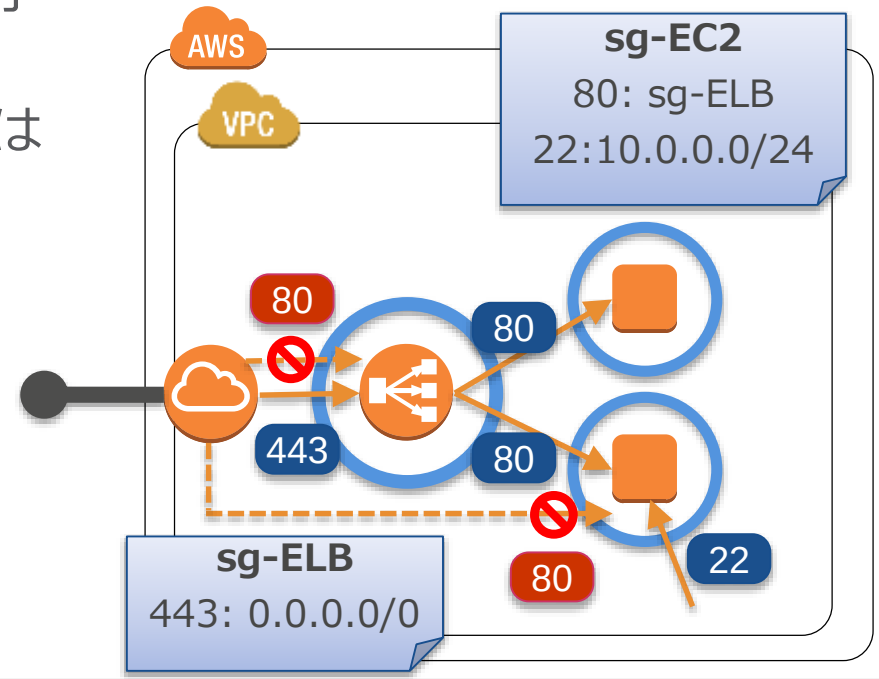


※どちらのELBもDNSレコードはパブリックで解決可能

amazon web services

ELBとSecurity Group

- ELBに任意のSecurity Groupを指定可能
- ICMP Echo Request/Replyを許可すれば、ELBがpingにも応答
- バックエンドのEC2インスタンスはELBからのみリクエストを受け付ける設定を推奨



ELBの機能

- ELB利用時のTips
  - 独自ドメイン名で利用
  - クライアントのIPアドレス取得
  - AZとバックエンドキャパシティの関係
  - ELBとバックエンドとのコネクション
  - ELB自身のスケーリング
  - VPCでの利用
  - ELBとSecurity Group
- SSLサポート
- スティッキー セッション
- Connection Draining
- アクセスログのS3保管

amazon web services

SSLサポート

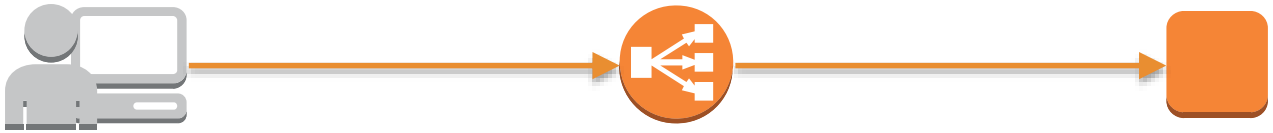
ELBでSSL Terminationできる

a) ELBでSSL Terminationし、バックエンドとはSSLなし
   バックエンドのEC2インスタンスでSSL処理せずに済むため
   負荷をオフロードできる。

b) ELBでSSL Terminationし、バックエンドとは別途SSL

c) SSLをバイパスしてバックエンドにTCPで送信 CLB
   クライアント証明書認証などを利用するためにはTCPとして扱う

a)	HTTPS	HTTP
b)	HTTPS / SSL	HTTPS / SSL
c)	TCP	TCP

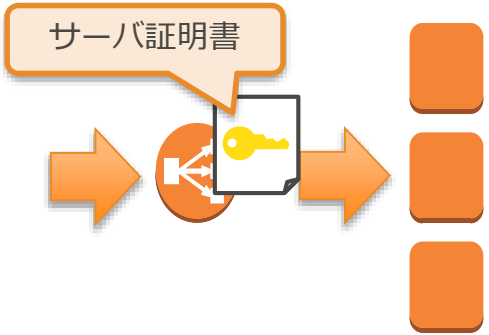


amazon web services

HTTPS/SSL利用時のサーバ証明書

ELBにサーバ証明書をアップロード

- HTTPS/SSL利用にはサーバ証明書のアップロードが必須
- AWS Certificate Manager との統合
- 複数ホスト名には別名・ワイルドカードか複数ELBで対応（SNI未対応）
- バックエンドとの通信にSSLを用いないなら証明書の管理が容易
- マネージメントコンソール or CLI or IAM APIで設定



SSL証明書のライセンスに関しては、サーバ単位/ドメイン単位で発行などそれぞれ異なるので発行元に問い合わせの事

amazon web services

SSLのセキュリティ強化

- TLS 1.1, 1.2のサポート
- Perfect Forward Secrecy (PFS) のサポート
- Server Order Preference
- 新しい定義済みのセキュリティポリシー

  新しく作ったELBではELBSecurity-Policy-2015-03がデフォルト

  → 既存のELBには互換性確認の上 ELBSecurity-Policy-2015-03 の適用を

ELBの機能

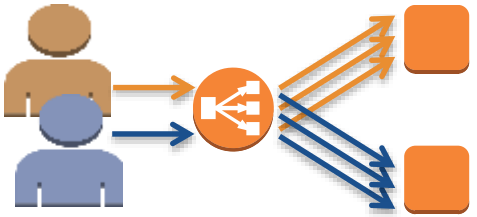
- ELB利用時のTips
  - 独自ドメイン名で利用
  - クライアントのIPアドレス取得
  - AZとバックエンドキャパシティの関係
  - ELBとバックエンドとのコネクション
  - ELB自身のスケーリング
  - VPCでの利用
  - ELBとSecurity Group
- SSLサポート
- スティッキー セッション
- Connection Draining
- アクセスログのS3保管

amazon web services

スティッキー セッション (stickness)

同じユーザから来たリクエストを全て同じEC2インスタンスに送信

- デフォルトで無効、利用するためには有効に
- アプリケーションでのセッション情報、一時ファイルなどを EC2インスタンスが保持する構成の場合に必要
- HTTP/HTTPSでのみ利用可能
- ELBは独自のセッションCookieを挿入



EC2インスタンスの増減を柔軟にできるように、セッション情報などは別のDBサーバやキャッシュサーバに持たせるのが望ましい。
この場合スティッキー セッションは不要。

amazon web services

スティッキー セッションの有効期間

有効期間の制御方法は以下２パターン

- Application Generated Cookie Stickiness（アプリケーション制御）
  アプリケーションが作成したCookieにあわせる
  - アプリケーションが作成するCookie名を指定
  - Cookieの有効パスもあわせる

CLB

- Load Balancer Generated Cookie Stickiness（期間ベース）
  セッション開始からの有効期間を指定してELBで制御
  - 秒単位で指定
  - 無期限にする事も可（無期限でもブラウザを閉じれば終了）

amazon web services

ELBの機能

- ELB利用時のTips
  - 独自ドメイン名で利用
  - クライアントのIPアドレス取得
  - AZとバックエンドキャパシティの関係
  - ELBとバックエンドとのコネクション
  - ELB自身のスケーリング
  - VPCでの利用
  - ELBとSecurity Group
- SSLサポート
- スティッキー セッション
- Connection Draining
- アクセスログのS3保管

amazon web services

Connection Draining

バックエンドのEC2インスタンスをELBから登録解除したり、ヘルスチェックが失敗した時に、新規割り振りは中止して、処理中のリクエストは終わるまで一定期間待つ

- 新しく作成したELBではデフォルトで有効、タイムアウト 300秒
- タイムアウト最大 3600秒
- Connection Draining 動作中
  - Management Consoleではインスタンスの表示が消える CLB
  - ターゲットグループの状態が draining に変化 ALB
  - API/SDK/CLIでは状況がわかる

amazon web services

登録解除時のAWS CLIでの表示

```
$ aws elb describe-instance-health --load-balancer-name (ELB Name)
{
    "InstanceStates": [
        {
            "InstanceId": "i-XXXXXXXX",
            "ReasonCode": "N/A",
            "State": "InService",
            "Description": “N/A”
        }
    ]
}
```

	State	Description
正常時	InService	N/A
Connection Draining 動作中	InService	Instance deregistration currently in progress.
全接続終了後 or タイムアウト後	OutOfService	Instance is not currently registered with the LoadBalancer.

amazon web services

ELBの機能

- ELB利用時のTips
  - 独自ドメイン名で利用
  - クライアントのIPアドレス取得
  - AZとバックエンドキャパシティの関係
  - ELBとバックエンドとのコネクション
  - ELB自身のスケーリング
  - VPCでの利用
  - ELBとSecurity Group
- SSLサポート
- スティッキー セッション
- Connection Draining
- アクセスログのS3保管

amazon web services

アクセスログのS3保管

ELBのアクセスログを指定したS3に自動保管

- 簡単にログのS3保管が実現できる
- ログに含まれる項目の例
  - timestamp
  - elb_status_code
  - backend_status_code CLB
    target_status_code ALB
  - received_bytes
  - sent_bytes
  - request
  - user_agent
  - ssl_cipher
  - ssl_protocol

amazon web services

CLBの機能

- TCP/SSL プロトコルによる負荷分散
- バックエンドインスタンスのサーバ証明書認証

amazon web services

TCP/SSL プロトコルによる負荷分散

CLB

CLB は 、HTTP/HTTPS に加えて、TCP/SSL（セキュアTCP）による負荷分散が可能

- TCPのポートは 1～65535
- Proxy Protocol による発信元IPアドレスの識別

amazon web services

CLBの機能

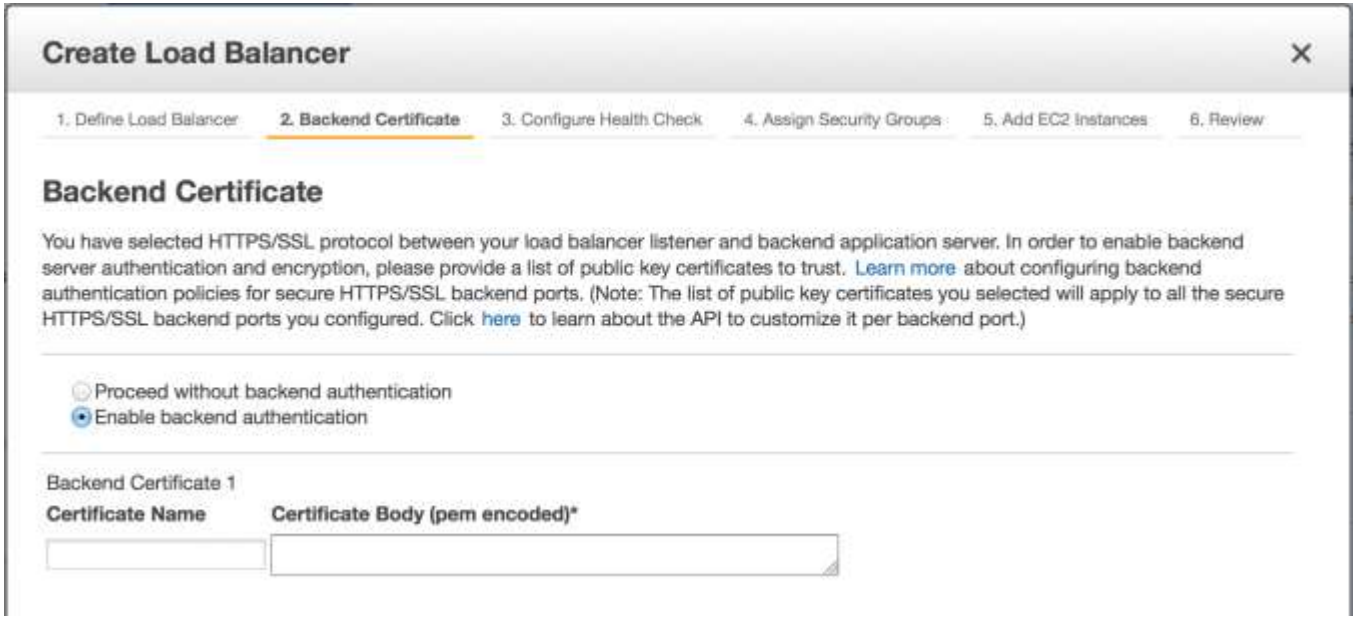
- TCP/SSL プロトコルによる負荷分散
- バックエンドインスタンスのサーバ証明書認証

amazon web services

バックエンドインスタンスのサーバ証明書認証

CLB

ELBとバックエンドのEC2インスタンス間でHTTPS/SSL使用時に、
特定の証明書を使用しているかどうかバックエンドを認証



amazon web services

ALBの機能

- コンテントベースルーティング
- 複数ポート対応と動的ポートマッピング
- その他のアップデート

amazon web services

ALBの機能

- コンテントベースルーティング
- 複数ポート対応と動的ポートマッピング
- その他のアップデート

amazon web services

CLB によるコンテントベースルーティング

CLB

CLB のバックエンドインスタンスは、全て同一の機能を持ったインスタンスが必要であり、異なる機能に対してコンテントベースルーティングは出来ない

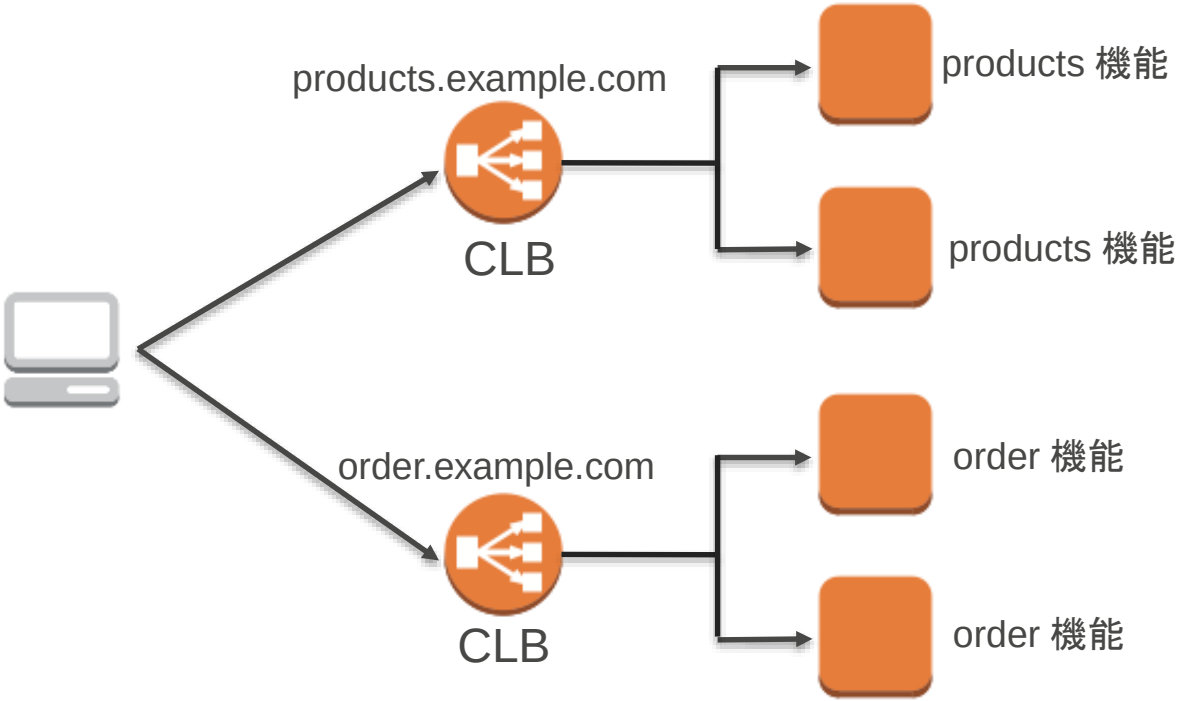
代替策として主に次の方式が利用されている

- サブドメインによるセグメンテーション
- Nginx によるセグメンテーション

amazon web services

サブドメインによるセグメンテーション方式

CLB

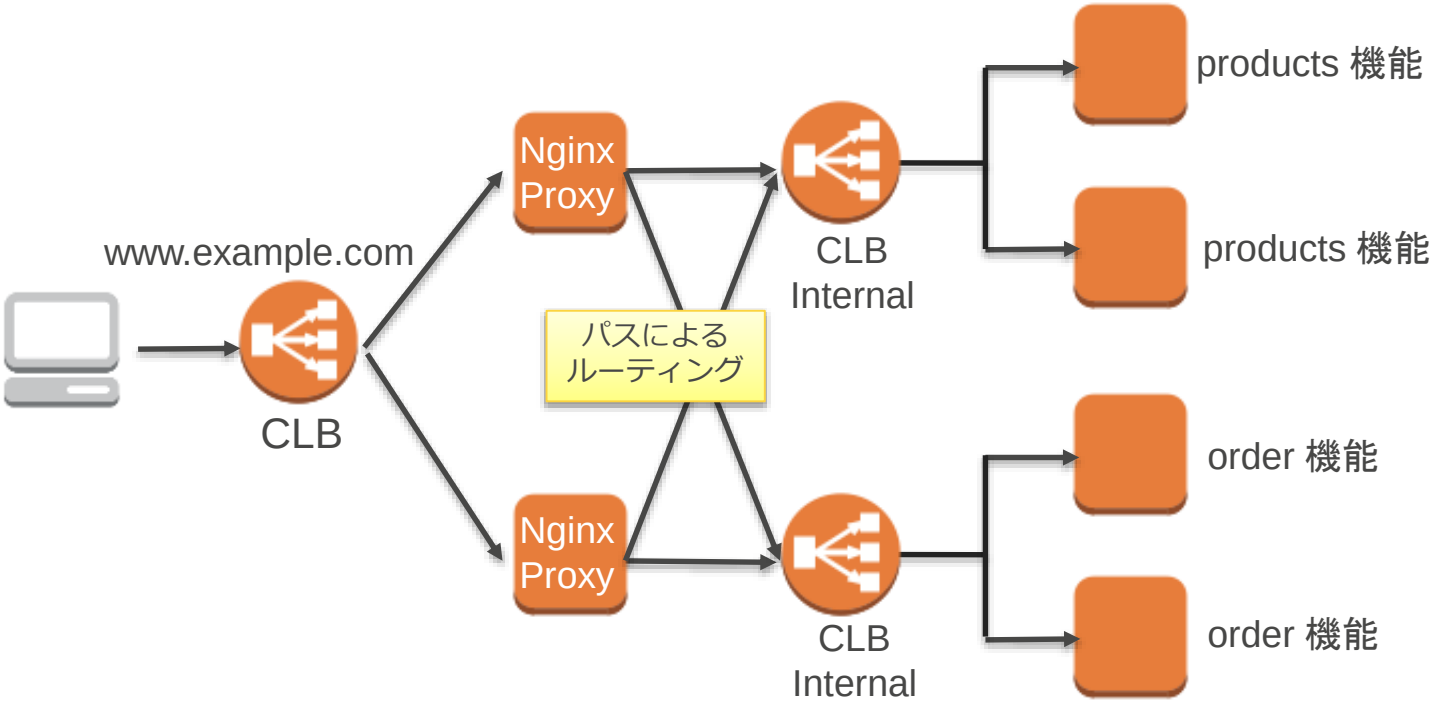


アプリケーションを複数のサブ機能（サービス）に分割する場合、従来は複数の ELB を利用しサブドメインで Endpoint の分割を行う等が必要

amazon web services

Nginx によるセグメンテーション方式

CLB



amazon web services

ALB によるコンテントベースルーティング

ALB

- 単一のロードバランサーで異なるアプリケーションへリクエストをルーティング可能
- EC2インスタンスはターゲットグループに登録され、ルールに従いリクエストをターゲットグループにルーティング
- ターゲットグループ内で負荷分散

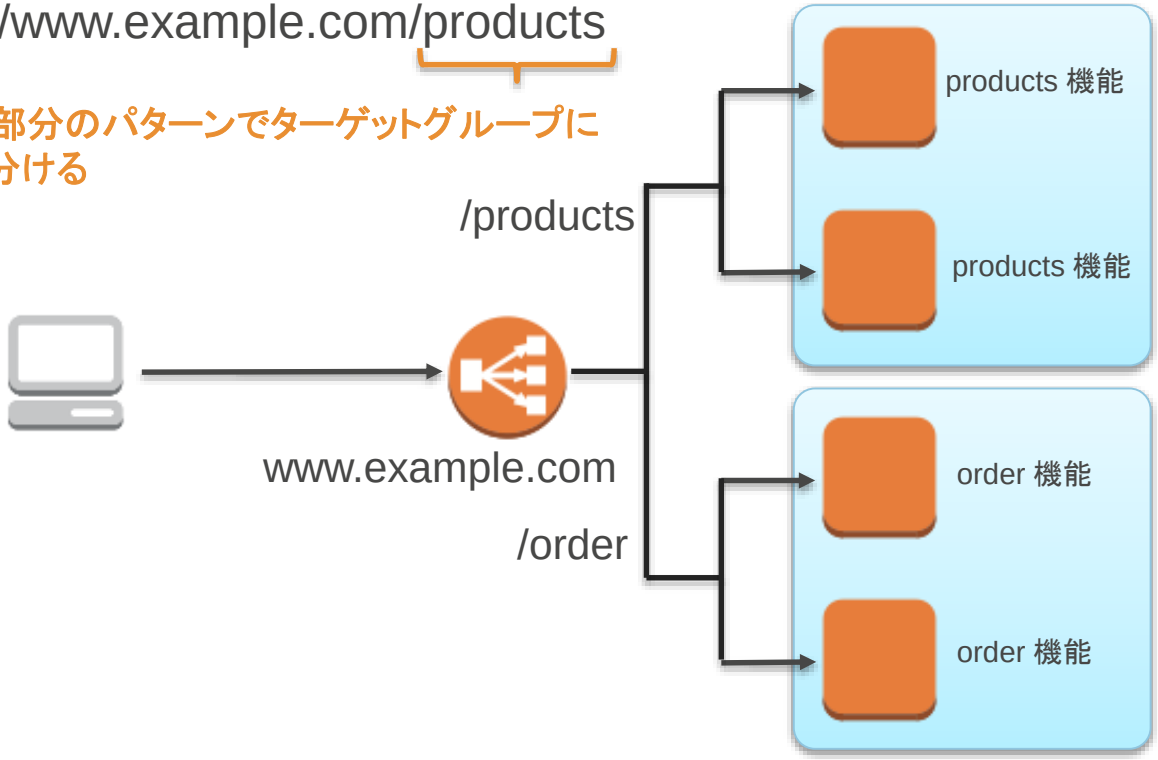
amazon web services

ALB によるコンテントベースルーティング

ALB

http://www.example.com/products

パス部分のパターンでターゲットグループに振り分ける



ALB を利用することで、一つの ALB の背後に複数の機能（サービス）を提供することが可能

ターゲットグループ

amazon web services

ALBの機能

- コンテントベースルーティング
- 複数ポート対応と動的ポートマッピング
- その他のアップデート

amazon web services

コンテナ化されたアプリケーションの制限

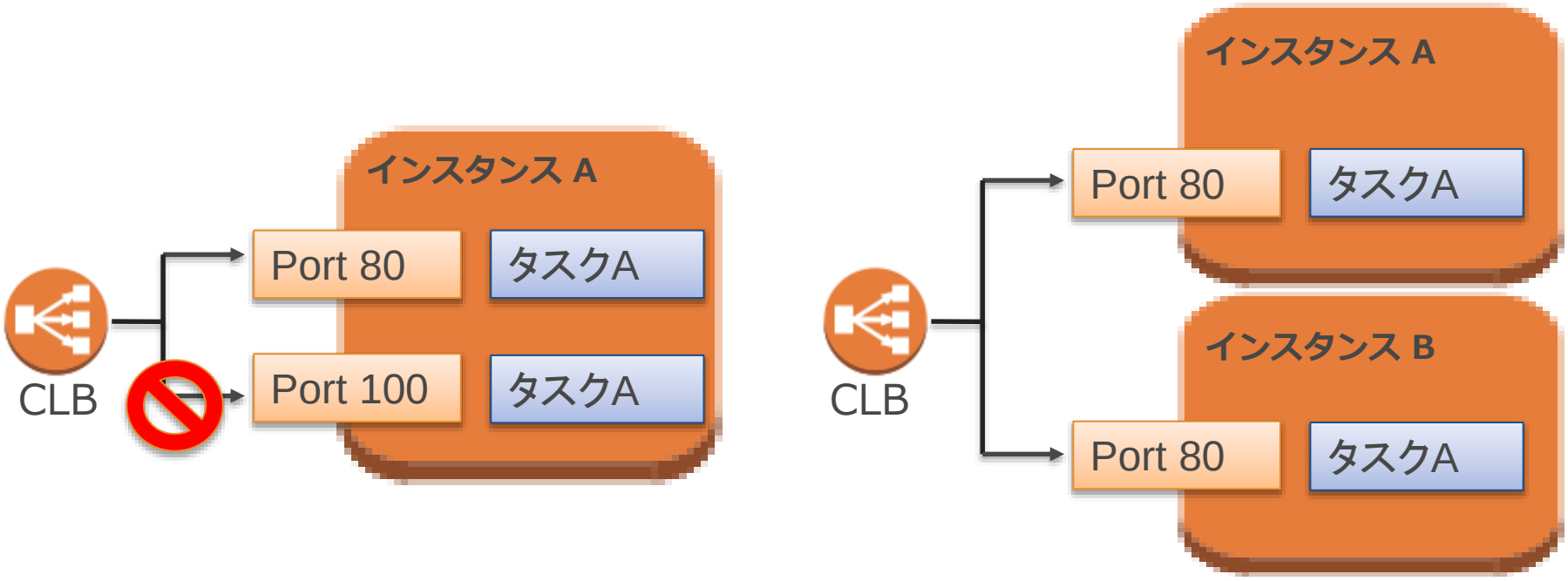
CLB

- コンテナ化されたアプリケーションは、1つのサーバー上で、アプリケーションごとに特定のポートを使ってサービスを提供
- ポートを分けることで、1つのEC2インスタンス上で、同じアプリケーションを複数実行
- これまでのELBでは、コンテナ化されたアプリの利用に制限があった
  - リスナーポートとアプリケーションが1対1でマッピング
  - 個々のアプリケーションが使うポートを管理する必要性
  - EC2インスタンス毎に1タスクにしか負荷分散できず、ECS クラスタの利用効率性が低い

amazon web services

コンテナ化されたアプリケーションの制限

CLB



amazon web services

複数ポートと動的ポートマッピング

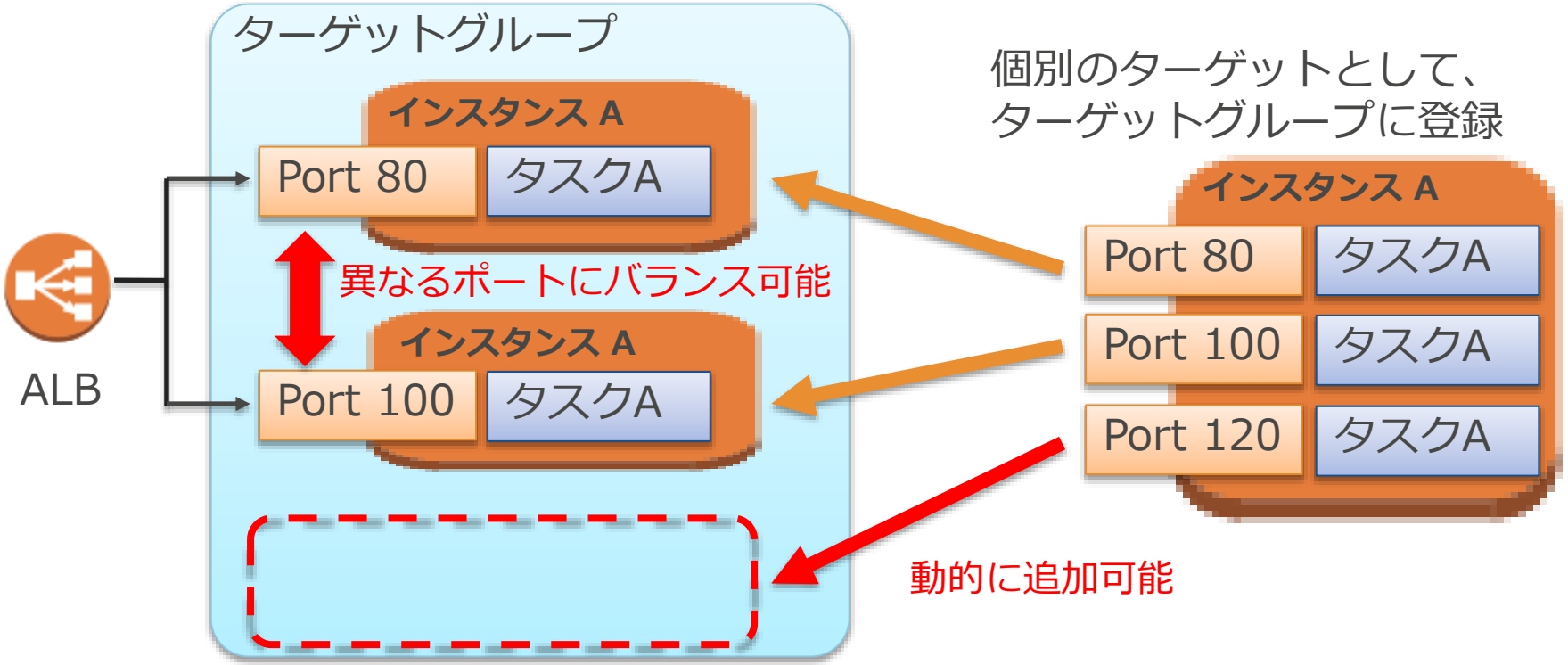
ALB

- EC2インスタンスをターゲットグループに割り当てる際、複数のポートを個別のターゲットとして登録することが可能
- 1つのEC2インスタンスに対して、複数ポートで負荷分散が可能
- ECSのServiceは、タスクがEC2上でスケジュールされる時点で未使用のポートを選択
- ECSのServiceは、自動的にそのポートでタスクをロードバランサに登録

amazon web services

複数ポートと動的ポートマッピング

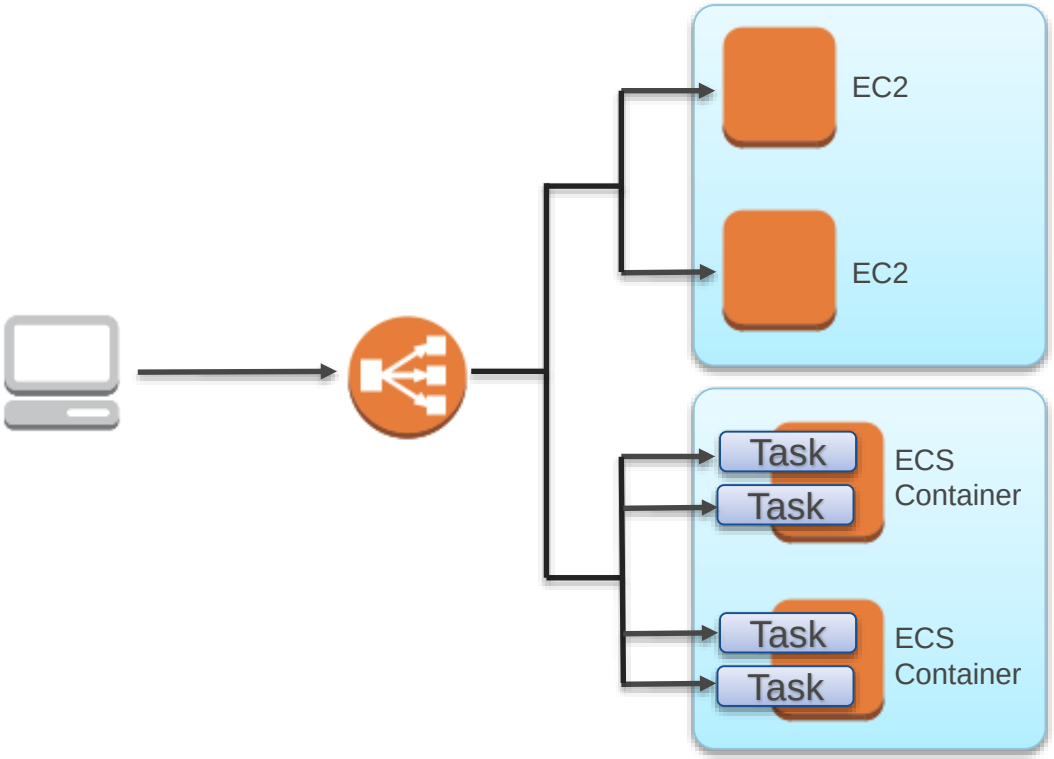
ALB



amazon web services

ALB のコンテナベースアプリケーション対応

ALB



ターゲットグループはECSコンテナをサポート。
ロードバランサーのバックエンドで、EC2とECSコンテナの混在も可能

ターゲットグループ

amazon web services

ALBの機能

- コンテントベースルーティング
- 複数ポート対応と動的ポートマッピング
- その他のアップデート

amazon web services

プロトコルの追加サポートとメトリクスの改善

ALB

- **Websocket**

  WebSocket は株価やスポーツの得点などページ内の動的なデータを配信するのに利用可能

- **HTTP/2**

  HTTP 1.1プロトコルから多数の改善が行われ、1つのコネクションで上で複数のリクエストをサポート

- **メトリクスの改善**

  ヘルスチェックをポートベースで実施し、許容するHTTPのレスポンスのレンジを指定でき、詳細なエラーコードも含まれる

amazon web services

ALB のリソース

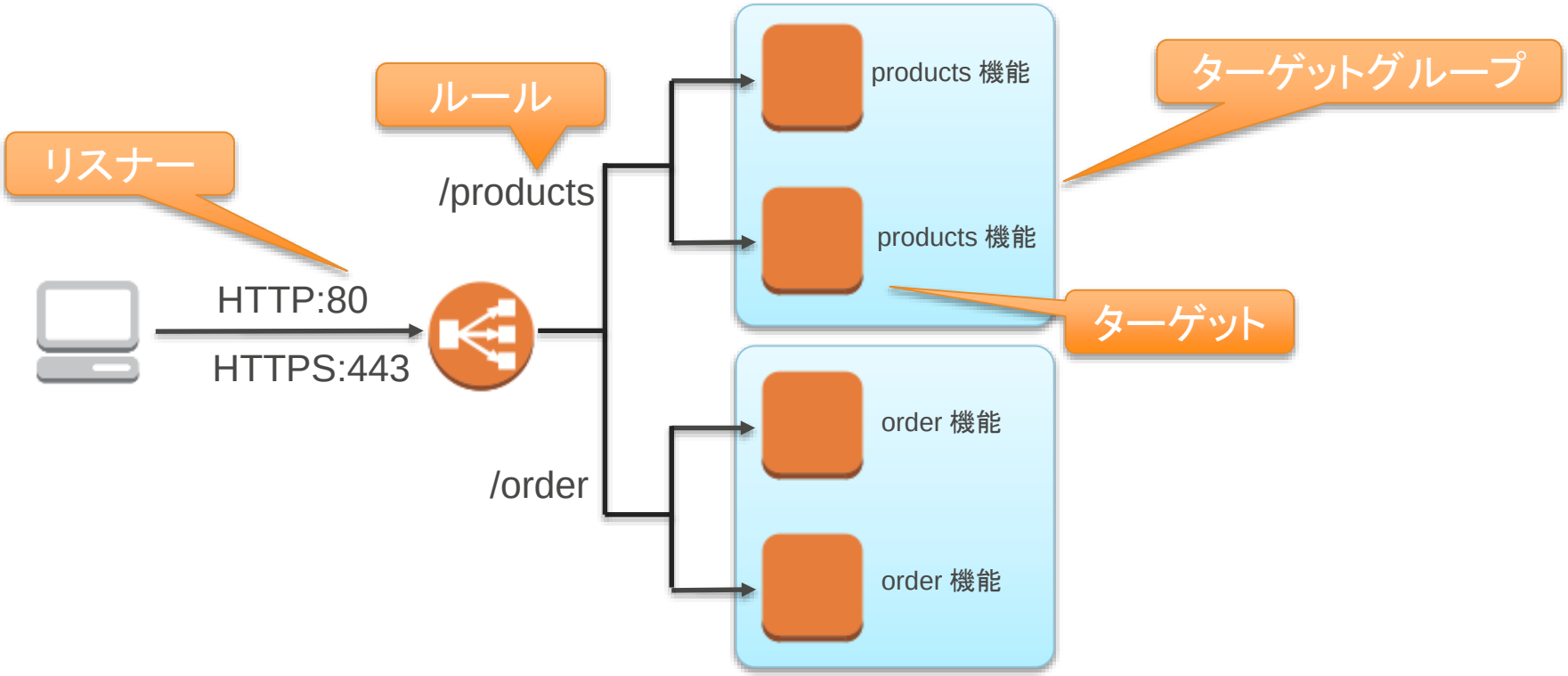
ALB

- **ロードバランサー** – ロードバランサーを表す最上位のリソース (CLBでは唯一のリソース)
- **リスナー** – LB側の他の接続設定と同様に、ロードバランサでListenするポートとプロトコルを含む
- **ターゲットグループ** – EC2インスタンスなどのターゲットの集合。インスタンス側の設定として、インスタンスで公開するポート、プロトコル、設定を含む
- **ターゲット** – ロードバランサーがトラフィックを転送するリソースやエンドポイント
- **ルール** – リクエストがどのように転送されるかを条件とアクションで定義。パスベースの条件で、リクエストを転送するアクションがサポートされる

amazon web services

ALB のリソース

ALB



amazon web services

料金

ALB

ALB の使用時間と、Load Balancer Capacity Units（LCU）の使用量で課金

- $0.0225/時間
- $0.008/LCU/時間

時間単価はCLBより 10% 安価

amazon web services

Load Balancer Capacity Units（LCU）

ALB

以下の 3 つのディメンションを測定し、使用量が最も高いディメンションのみ請求

- **新規接続数**: 1 秒あたりの新しく確立された接続数
- **アクティブ接続数**: 1 分あたりのアクティブ接続数
- **帯域幅**: ロードバランサーで処理されたトラフィック量（Mbps）

1 LCU には次のものが含まれる

- 2 KBの証明書の場合：1 秒あたり最大 25 個の新規接続
  4 KBの証明書の場合：1 秒あたり最大 5 個の新規接続
- 1 分あたり最大 3000 個のアクティブ接続
- 最大 2.22 Mbps の帯域幅

amazon web services

移行ツール

ALB

既存のクラシックロードバランサと同じ設定で、
新しいアプリケーションロードバランサを作成可能

Classic load balancer to Application load balancer copy utility

https://github.com/aws/elastic-load-balancing-tools

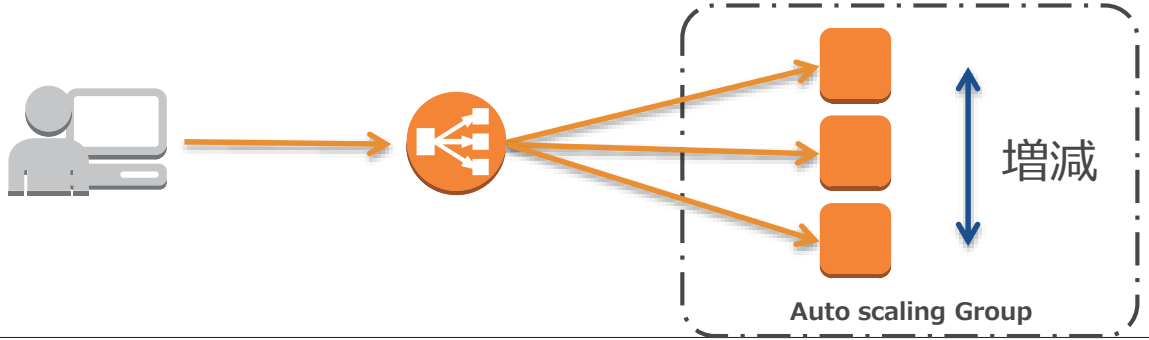
amazon web services

ELBと各種サービスの連携

amazon web services

Auto Scalingとの連携

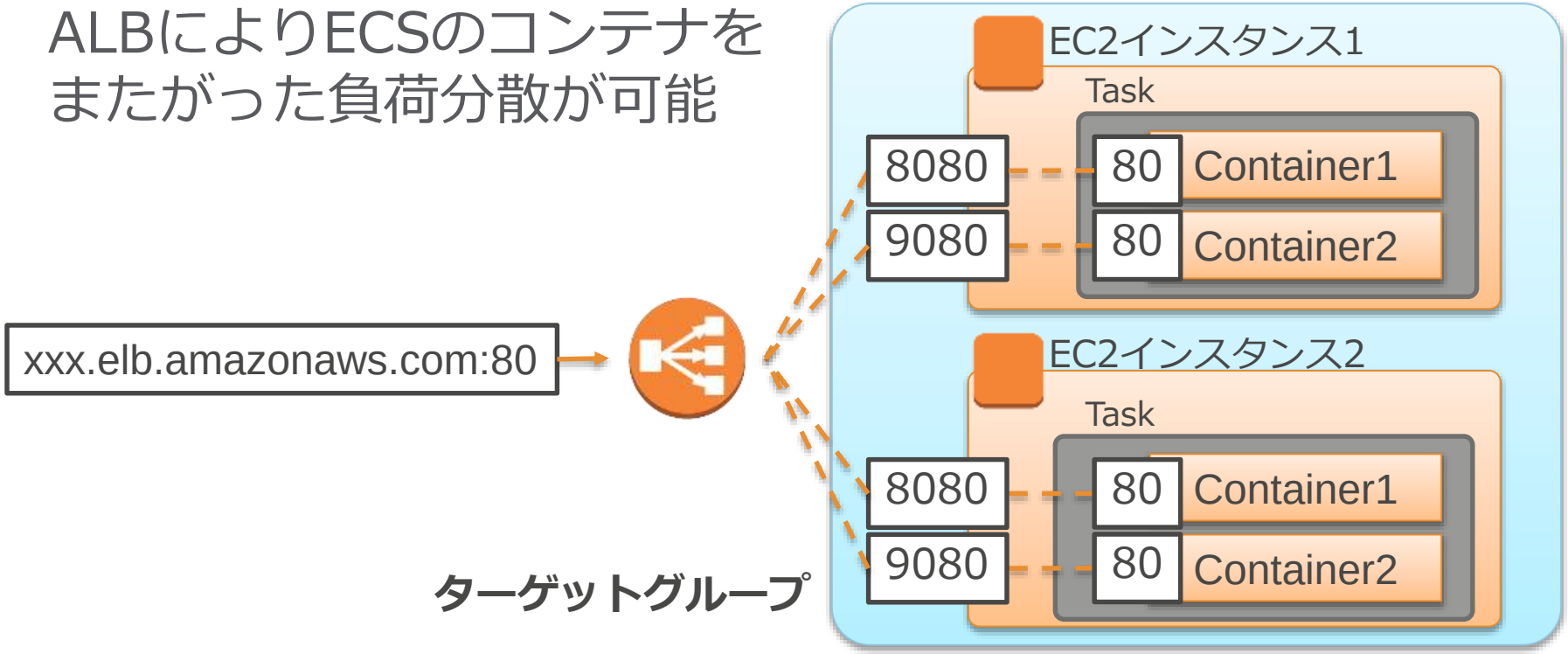
- Auto Scalingによるインスタンス増減時にELBへの追加・削除が可能
- ELBのヘルスチェックの結果をAuto Scalingに反映可能
- インスタンス削減時は、Connection Drainingでの処理中の接続を待つ
- 利用例
  - 一定間隔でレスポンスをチェックし、遅延が増加したらインスタンスを自動追加
  - ELBのヘルスチェックが成功したEC2インスタンスを常にX台以上



EC2 Container Service(ECS)のALB対応

ALB

ALBによりECSのコンテナを
またがった負荷分散が可能



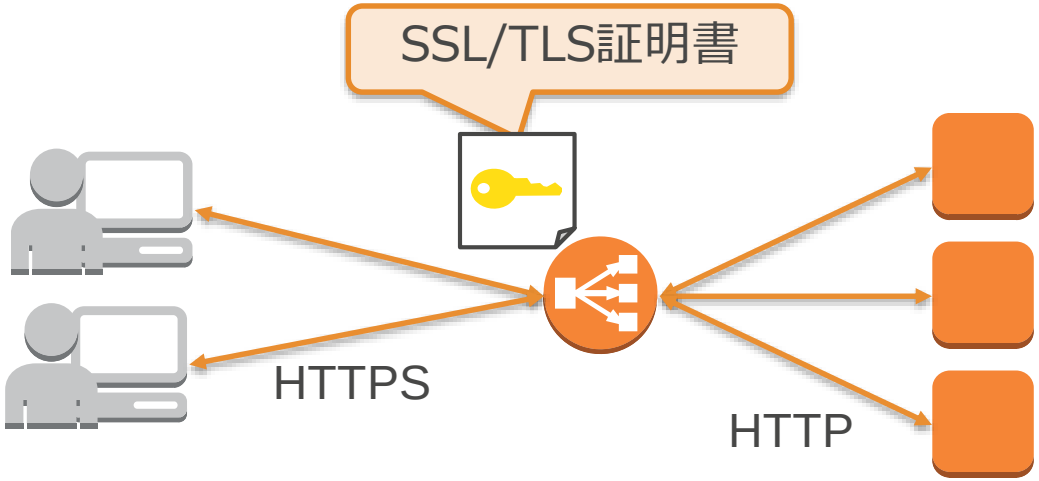
 参照 https://aws.amazon.com/jp/ecs/

amazon web services

AWS Certificate Manager との統合

AWS Certificate Manager(ACM)を使用して、証明書のリクエストと管理を容易に実施

- 無料で利用可能（ELB、Amazon CloudFront に対し）
- ELB に対する証明書の設定を数クリックで完了



参照 https://aws.amazon.com/jp/certificate-manager/

CloudWatchとの連携

CloudWatchによりELBの以下のメトリクスを1分単位で監視可能

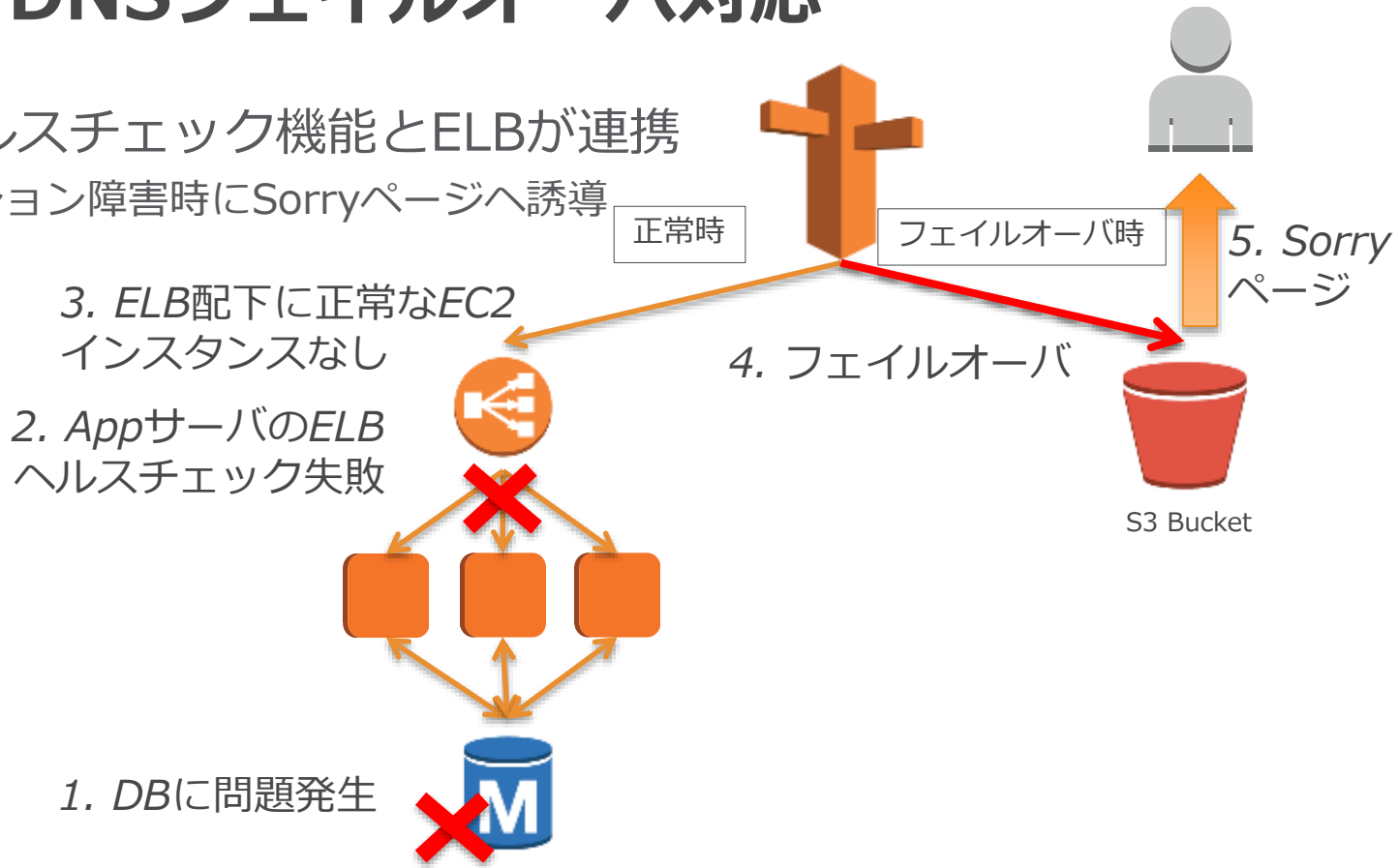
- 正常なバックエンドのホスト数 (HealthyHostCount)
- 異常なバックエンドのホスト数 (UnHealthyHostCount)
- リクエスト数 (RequestCount)
- 遅延時間 (Lantency)
- ELBが返した4xx,5xxのレスポンス数 (HTTPCode_ELB_4xx)
- バックエンドが返した2xx,3xx,4xx,5xxレスポンス数 (HTTPCode_Backend_2xxx)
- バックエンドへの接続エラー回数 (BackendConnectionError)
- バックエンドへの送信保留中の件数 (SurgeQueueLength)
- キュー溢れのため拒否した件数 (SpilloverCount)

amazon web services

Route 53 DNSフェイルオーバ対応

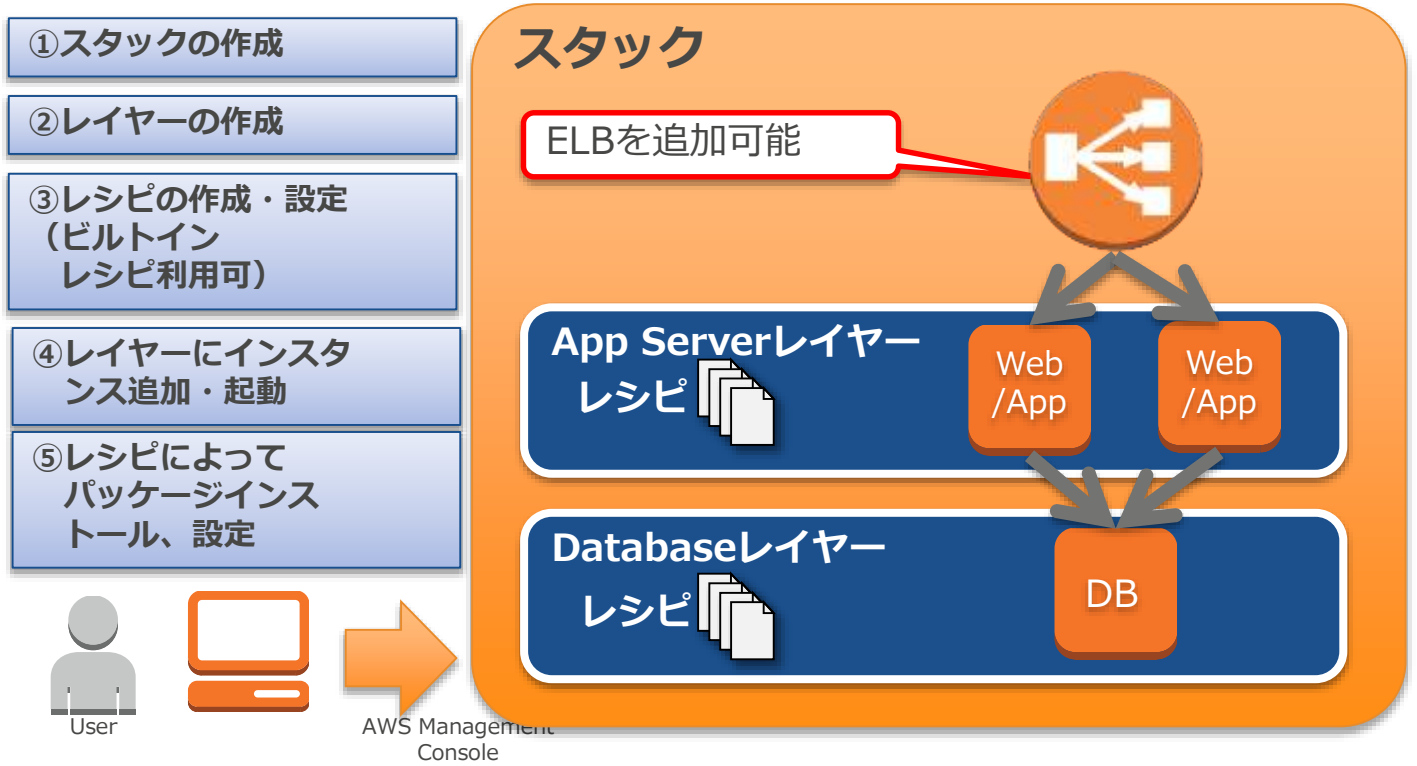
Route 53のヘルスチェック機能とELBが連携

例：アプリケーション障害時にSorryページへ誘導



OpsWorksのELB対応

OpsWorksでELBをレイヤーにアタッチして、負荷分散が可能



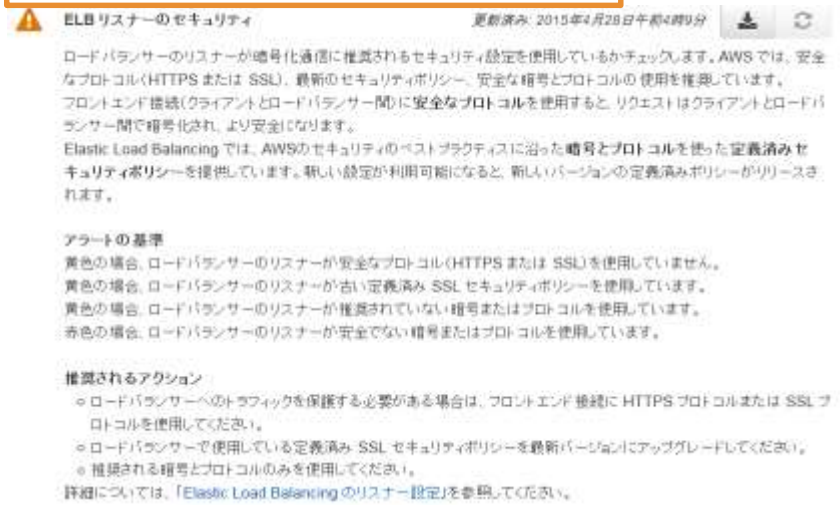
参照 http://docs.aws.amazon.com/ja_jp/opsworks/latest/userguide/layers-elb.html

amazon web services

AWS Trusted AdvisorによるELBのチェック

- ELBのセキュリティとフォールトトレランスをチェック
  - ELBリスナーのセキュリティ
  - ELBセキュリティグループ
  - ELBクロスゾーン負荷分散
  - ELB Connection Draining

ELBリスナーのセキュリティの例



参照 http://aws.typepad.com/aws_japan/2015/03/new-elastic-load-balancer-checks-for-aws-trusted-advisor.html

amazon web services

ELB負荷テストに関するTips

amazon web services

ELBを負荷テストする必要性について

ELBのいくつかの特長がテストシナリオに影響を与える可能性がある

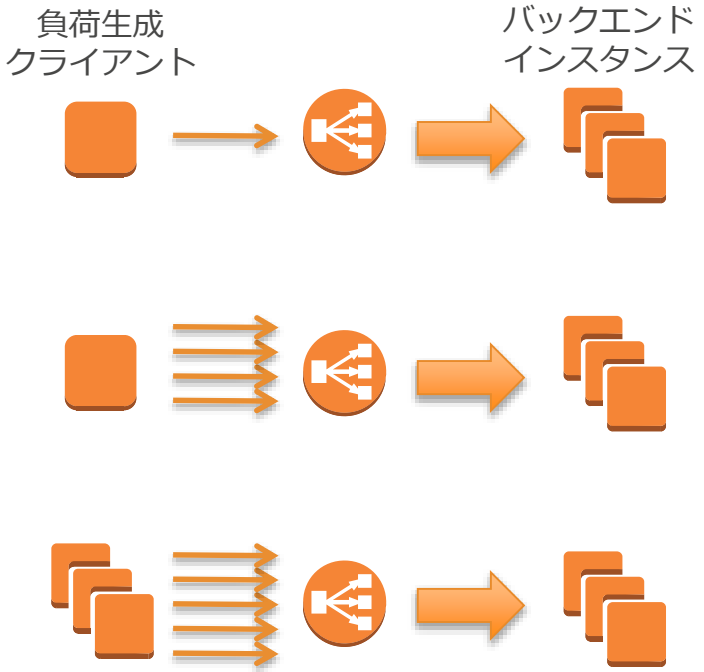
- ELB のスケーリング
- ELB の初期キャパシティ
- アイドル時のコネクションタイムアウト
- バックエンドインスタンスのヘルスチェック
- Sticky セッション　等

利用状況に合わせたシナリオでテストが必要

amazon web services

ELBの負荷テスト方法の種類

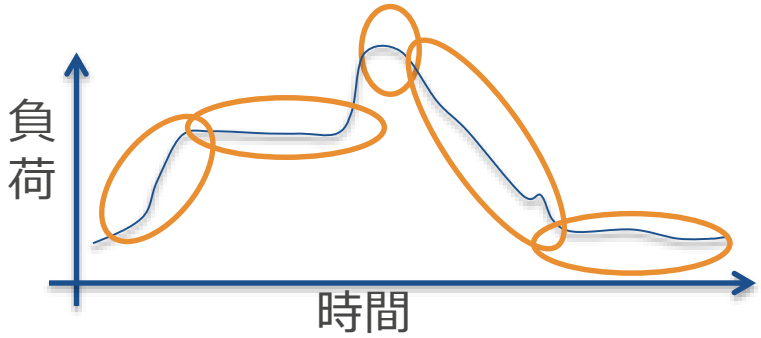
- シングルクライアントテスト
  - 例：Apache Bench(ab)
- マルチクライアントテスト
  - 例：curl-loader
    (都度DNS解決を行うツールが望ましい)
- 分散テスト
  - 例：Fabricフレームワーク、
    BeesWithMachineGuns



クライアントの負荷が足りない場合はクライアント数を増やす等で対応可

推奨テストアプローチ

- 想定する最大負荷のテスト
- 通常のトラフィック時のテスト
  - トラフィックの多い時
  - トラフィックの少ない時
  - トラフィックの傾向に変化がある時（朝や昼の時間帯など）
- 短い時間でトラフィックが大きく変化する場合のテスト



ELB以外にも負荷生成クライアント、バックエンドEC2インスタンスも監視すべき

- アプリケーション内部の動作も要確認
- どこかボトルネックになっているか把握しておく

amazon web services

負荷テストの注意事項

- ELBの初期スケールに注意
  - スケールするまでに、HTTP 503レスポンスを返す期間があり得る
  - 回避策：
    - ELBの暖気運転（ Pre-Warming）申請をする
    - 5分間隔で50%以上のトラフィック増加をしないよう負荷テストを設定
- DNSクエリの仕方に注意
  - テストクライアント側で少なくとも1分に1回DNSの再解決をする
- スティッキーセッション利用時の割り振り方
  - 同じCookieでリクエストを続けた場合などは振り分けに偏りが発生
- バックエンドインスタンスのアイドルタイムアウト
  - ELBのタイムアウト値以上に設定しないとELBが誤って不健全なホストと見なす可能性あり

詳細 http://d36cz9buwru1tt.cloudfront.net/jp/documentation/BestPracticesInEvaluatingELB-ja-final.pdf

amazon web services

まとめ

amazon web services

CLBとALBの機能比較

機能	Classic Load Balancer	Application Load Balancer
プロトコル	HTTP、HTTPS、TCP、SSL	HTTP、HTTPS
プラットフォーム	EC2-Classic、EC2-VPC	EC2-VPC
スティッキーセッション (Cookie)	✓	ロードバランサーが生成したCookie
バックエンドサーバー認証	✓	
バックエンドサーバー暗号化	✓	✓
アイドル接続のタイムアウト	✓	✓
コネクションドレイニング	✓	✓
クロスゾーン負荷分散 †	✓	常に有効
ヘルスチェックする間隔を秒で入力します † †	✓	向上
CloudWatch メトリックス	✓	向上
アクセスログ	✓	向上
パスベースのルーティング		✓
単一のインスタンスで複数のポートにルーティング		✓
HTTP/2 サポート		✓
Websockets サポート		✓
ロードバランサーの削除の保護		✓

† ALBではクロスゾーン負荷分散は常に有効
† † ALBでは、ヘルスチェックでHealthyとみなすHTTPレスポンスコードを設定可能。ALBではヘルスチェック失敗の情報提供を改善

amazon web services

まとめ

～ELBはAWSが提供するロードバランシングサービス～

- 運用管理コストを抑えながら
  スケーラブルで高可用なインフラを構築可能
- **L7 の負荷分散は ALB を利用**
- 各種サービスとの連携もスムーズ＆随時拡充
- 負荷試験時はその特性を理解した上で実施
- 急激な負荷の増大が想定される場合には、
  サポート加入の上で暖機申請（Pre-Warming）

amazon web services

参考資料

- Elastic Load Balancing ユーザーガイド
  http://docs.aws.amazon.com/ja_jp/elasticloadbalancing/latest/userguide/what-is-load-balancing.html
- Elastic Load Balancing アプリケーションロードバランサー
  http://docs.aws.amazon.com/elasticloadbalancing/latest/application/
- Elastic Load Balancing クラシックロードバランサー
  http://docs.aws.amazon.com/elasticloadbalancing/latest/classic/
- FAQ
  https://aws.amazon.com/jp/elasticloadbalancing/applicationloadbalancer/faqs/
  https://aws.amazon.com/jp/elasticloadbalancing/classicloadbalancer/faqs/
- 料金
  https://aws.amazon.com/jp/elasticloadbalancing/classicloadbalancer/pricing/
  https://aws.amazon.com/jp/elasticloadbalancing/applicationloadbalancer/pricing/

amazon web services

Q&A

amazon web services

オンラインセミナー資料の配置場所

- AWS クラウドサービス活用資料集
  - http://aws.amazon.com/jp/aws-jp-introduction/

- AWS Solutions Architect ブログ
  - 最新の情報、セミナー中のQ&A等が掲載されています
  - http://aws.typepad.com/sajp/

amazon web services

公式Twitter/Facebook
AWSの最新情報をお届けします

@awscloud_jp

もしくは
http://on.fb.me/1vR8yWm

最新技術情報、イベント情報、お役立ち情報、
お得なキャンペーン情報などを日々更新しています！

AWSの導入、お問い合わせのご相談

AWSクラウド導入に関するご質問、お見積り、資料請求を
ご希望のお客様は以下のリンクよりお気軽にご相談ください
https://aws.amazon.com/jp/contact-us/aws-sales/

※「AWS　問い合わせ」で検索してください

ご参加ありがとうございました

amazon web services


amazon
web services

Appendix

ELB Updates

- 2016.8　Application Load Balancer のリリース
- 2016.1　AWS Certificate Manager のサポート
- 2015.12　停止および再起動したバックエンドインスタンスの自動再登録をサポート
- 2015.9　全ポートに対する負荷分散とアクセスログへのフィールドの追加
- 2015.4　EC2 Container ServiceのELB対応
- 2015.3　AWS Trusted AdvisorによるELBチェック項目の追加
- 2014.8　ELBのタグ対応
- 2014.7　ELBのコネクションタイムアウト変更
- 2014.4　CloudTrail対象に追加
- 2014.3　接続のストリーミング
- 2014.3　アクセスログのS3保管
- 2014.2　SSLサポート強化
  （Perfect Forward Secrecy(PFS)、Server Order Preference、ELBSecurityPolicy-2014-01ポリシー追加）
- 2013.11　クロスゾーン負荷分散
- 2013.10　CloudWatchのメトリック追加（BackendConnectionError、SurgeQueueLength、SpilloverCount）
- 2014.7　Proxy Protocol
- 2013.5　全てのHTTPメソッド対応
- 2013.5　Route 53のフェールオーバ連携
- 2012.6　Internal ELB