

ログイン

2023年7月18日 13:27

<https://d-956703c220.awsapps.com/start#/>

IAMユーザー

608728620263

osuke_oyaizu

d2^KMU8P@s{E\$b

[TOROハンズオン環境\(学習用環境\)利用申請 – 一般ユーザー向け – Jira Service Management \(atlassian.net\)](#)

osuke_oyaizu

osuke_oyaizu@mail.toyota.co.jp

Gakuen844#000

資料

2023年8月9日 11:27

[https://toyotajp-my.sharepoint.com/:o/g/personal/1464469
tmc_twfr/toyota_co_jp/EqPbT9-pCGhMs5to0eJc78cBqwdYM2fjr7inPFGKzWJy_g?e=682oM5](https://toyotajp-my.sharepoint.com/:o/g/personal/1464469tmc_twfr/toyota_co_jp/EqPbT9-pCGhMs5to0eJc78cBqwdYM2fjr7inPFGKzWJy_g?e=682oM5)

セクション1



2023年7月18日 13:44

本講座のテキスト一覧

- ・システムをインフラから構築できるようになろう
- ・AWSってなに？
- ・インフラってなに？

メリット

- ・自分でサービスを作れる
- ・課題に対してシステム全体で対応できるようになる

どのように？

[サーバー]

- ・どのようなサーバーが必要か
- ・サーバーを設置
- ・サーバーのOSをインストールし、各種設定
- ・必要なソフトウェアをインストールして設定

[ネットワーク]

- ・構築したサーバーをネットワークに接続する。

ネットワークで使用するIPアドレスの範囲を決める

サーバーにIPアドレスを割り当てる

ドメイン名とIPアドレスの対応を割り当てる

AWSとは

2023年7月18日 13:28

それぞれアカウントが分離している

AWS(Amazon Web Service)
世界最大のクラウドサービス

特徴

- ・サービスが豊富
- ・リソースが柔軟
- ・従量課金



用語説明

2023年7月18日 16:47

インフラとは？

サーバーやネットワーク

サーバーとは？

サービスを提供するコンピューター

ネットワークとは？

複数のコンピューターをつないで、データを送受信できるようにするもの。

クラウドとは？

ネットワークを利用してコンピュータリソースを利用する形態のこと

オンプレミス

- ・インフラを自前で用意して自社で所有・管理
- ・利点は自由度が高い
- ・欠点は初期コスト、サーバーの増減がしにくい

クラウド(AWS)

- ・インフラをネットワーク経由で使用・管理
- ・利点は初期コストが少ない、サーバーの増減が容易
- ・欠点は費用の予測がつきづらい

セクション2

2023年7月18日 16:59

- ・アカウント作成
- ・料金アラートを設定
- ・IAMで作業用ユーザーを作成
- ・CloudTrailで操作ログを記録

アカウント作成

2023年7月18日 17:01

- ・ クレカ
- ・ 通話可能な携帯電話
- ・ メールアドレス

料金アラート

2023年7月18日 17:04

請求ダッシュボードでアラートを受け取る
CloudWatchで料金アラートを設定する

IAMユーザー作成

2023年7月18日 17:08

個別にユーザーを作成してそれぞれに必要な権限をつける

操作ログ記録

2023年7月18日 17:13

CloudTrail

デフォルトで有効だが、保存期間は90日のみ

S3だと永久保存

セクション3

2023年7月18日 17:17

- ・システムの全体像
- ・AWSのネットワークの概念
- ・ネットワークのIPアドレスを決める
- ・VPCを作成する
- ・サブネットを作成
- ・ルーティングを設定
- ・考慮ポイント

用語

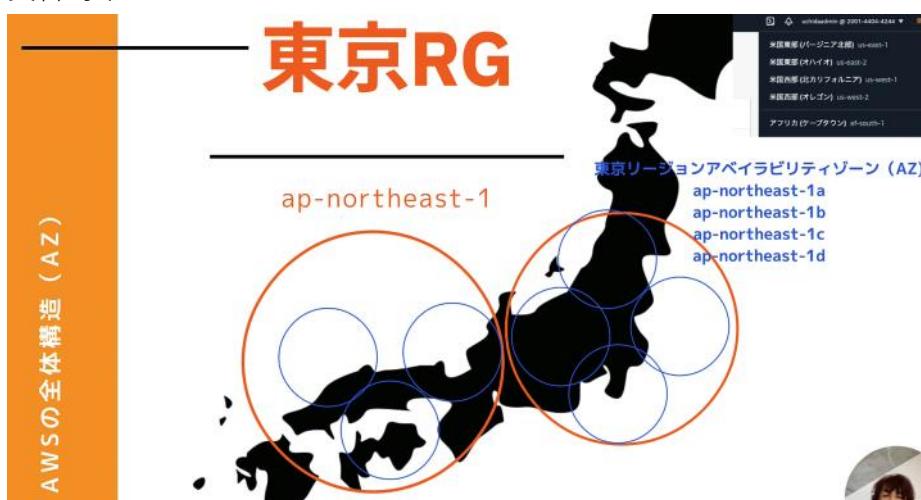
2023年7月18日 17:19

リージョン

AWSの各サービスが提供されている地域のこと
日本で使用する際は東京を選択する
サービスを提供する大きな括り
東京では最新の機能は使えない可能性がある



アベイラビリティゾーン
独立したデータセンター
地理的に分離されたところ（場所）
災害対策



VPC
AWS上に仮想ネットワークを作成できるサービス

サブネット

VPCを細かく区切ったネットワーク

IPアドレス決定

2023年7月18日 17:24

IPアドレス

ネットワーク上の機器を識別するためのインターネット上の住所

パブリックIPアドレス

- ・インターネットに接続する際に使用するIPアドレス
- ・重複しないようにICANNという団体が管理している
- ・プロバイダーやサーバー事業者から貸し出される



VPCの範囲がパブリックもプライベートも含む

プライベートIPアドレス

- ・インターネットで使用されないIPアドレス

.

下記範囲内のアドレスを自由に
使用することができる

10.0.0.0 ~ 10.255.255.255

172.16.0.0 ~ 172.31.255.255

192.168.0.0 ~ 192.168.255.255

- ・社内LANの構成やネットワークの実験時はプライベートIPを使用

範囲の表記法

①CIDR表記

192.168.128.0/24

②サブネットマスク表記

192.168.128.0/255.255.255.0

VPC作成

2023年7月18日 17:42

サブネット作成

2023年7月18日 17:55

パブリックにWebサーバーを設置

プライベートにDBサーバーを設置



パブリックとプライベートの違い

外部からアクセスできるかできないか

ルーティング

2023年7月18日 18:01

デフォルトルートの宛先をInternet gatewayに設定する
Internet gatewayでNATする

- Internet gatewayが必要
- デフォルトゲートウェイの設定が必要(Internet gatewayへの)

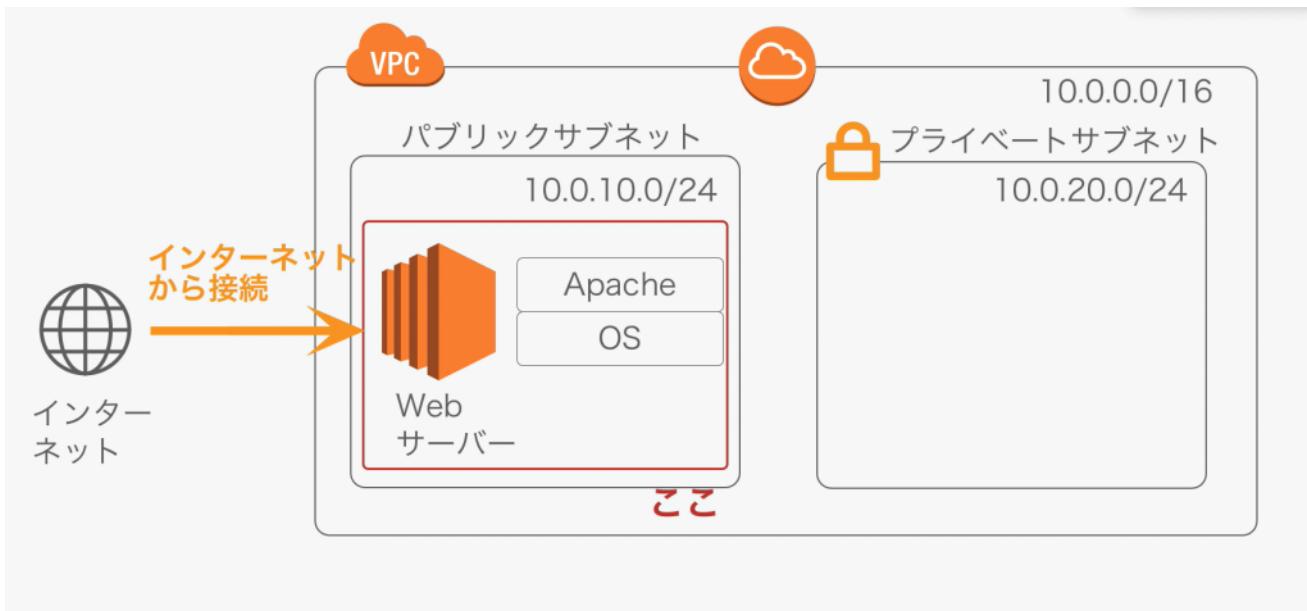
考慮するポイント

2023年7月18日 18:14

- ・プライベートIPアドレス範囲から指定する
- ・作成後は変更できないので、大きめに設定する
大きさは/28から/16 /16が推奨
- ・オンプレミスや他VPCのレンジと重複しないように気をつける
相互接続する可能性がある場合は、重複しないように設計
- ・異なるシステムの場合はアカウントを分ける
- ・

セクション4

2023年7月19日 8:43



24 25とばす

EC2

2023年7月19日 8:45

EC2(Elastic Compute Cloud)
AWSクラウド上の仮想サーバー

インスタンス
EC2から立てられたサーバー

AMI(Amazon Machine Image)
インスタンス起動に必要な情報が入ったOSのイメージ
サーバーのテンプレート

インスタンスタイプ



<https://techblog.forgevision.com/entry/aws-ec2-instance-bgr>

- **EC2インスタンスタイプの種類**

EC2インスタンスタイプは非常に多くの種類があり、それぞれ以下図のような書式にて表記されています。



ではそのインスタンスタイプの見方について4つに分け整理していきます。

①インスタンスファミリー

②インスタンス世代

③追加機能

④インスタンスサイズ

EBS

- ・高い可用性と耐久性をもつストレージ
- ・ほかのインスタンスに付け替え可能
- ・EC2インスタンスをStop/TerminateしてもEBSは保持可能
- ・Snapshotを取得しS3に保存可能
- ・EBSの費用が別途発生
- ・OSやDBなど永続性と耐久性が必要なデータを置く

インスタンスストア

- ・インスタンス専用の一時的なストレージ
- ・ほかのインスタンスに付け替えることができない
- ・EC2インスタンスをStop/Terminateするとクリア
- ・追加費用なし（無料）
- ・なくなってはいけないデータは置かない
- ・一時ファイル、キャッシュなど、失われても問題がないデータを置く

SaaS

2023年7月19日 10:03

OSを選択してその後がユーザー

各サービスの提供範囲



インスタンス作成

2023年7月19日 11:09

注意点

22. は動画を基に操作。ただしインスタンスを起動させないこと！ここは、補足説明しますので、聞いて下さい。
右側のコース内容にあるリソースから、現行の画面での設定方法が書いてるので、そちらを参考にする
EC2作成時、以下の内容を指定する
- ・AMIはAmazon Linux 2023（デフォルト）
 - ・インスタンスタイプ：t3.micro
 - ・プライベートアドレスは指定不要
 - ・ボリュームタイプは汎用SSD(gp2)を指定する。絶対にプロビジョンドにしない！（高額請求に繋がる）
 - ・セキュリティグループは既存のセキュリティグループを指定。絶対にSSHを0.0.0.0から許可しない！（セキュリティ違反）
 - ・高度な詳細→クレジット仕様：スタンダード（動画のT2/T3無制限の話）、メタデータのバージョン：V2のみ
 - ・IPアドレス自動割り振りをオンにする

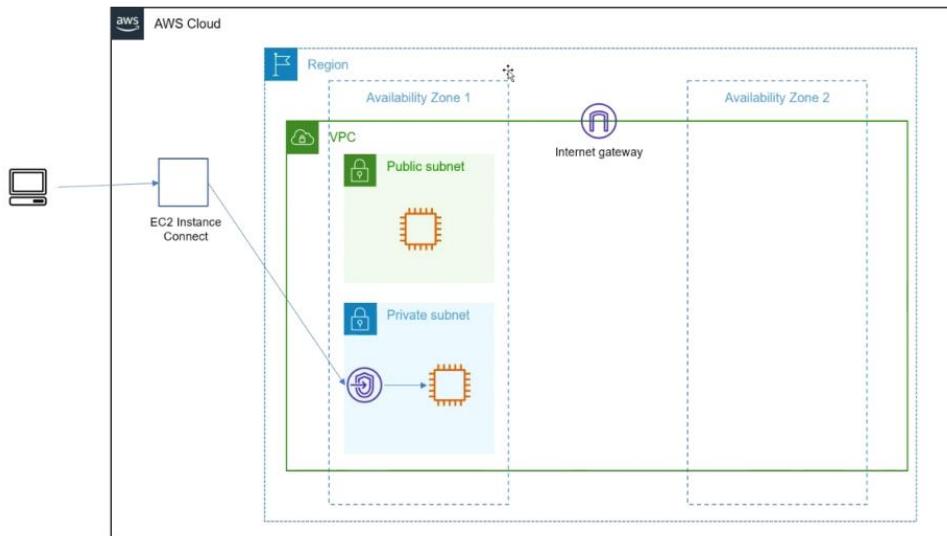
エンドポイント作成

2023年7月19日 11:11

VPC → エンドポイント

https://toyotajp.sharepoint.com/sites/msteams_c3fcad/layouts/15/stream.aspx?id=%2Fsites%2Fmsteams%5Fc3fcad%2FShared%20Documents%2F02%5F%E8%82%B2%E6%88%90%E8%A8%88%E7%94%BB%2FRecordings%2FEC%EF%BC%92%E8%A3%9C%E8%B6%B3%2D20230719%5F100449%2D%E4%BC%9A%E8%AD%B0%E3%81%AE%E9%8C%B2%E9%9F%B3%2Emp4

エンドポイントとは



エンドポイントを作成すればPrivate subnetに接続できる
SSH接続するために設置する

サービスカテゴリ：AWSのサービス

VPC：自分の

セキュリティグループ：EC2と同じ

サブネット：配置するサブネットを選択

SSH

2023年7月19日 11:08

サーバーと自分の目の前のパソコンをセキュアにつなぐサービスのこと
通信内容が暗号化された遠隔ログインシステム

公開鍵認証

- ・サーバーへのログイン時に認証を行う仕組み
- ・ユーザー名とパスワードを使用した認証と比べてよりセキュア
- ・公開鍵暗号（秘密鍵と公開鍵）を用いて認証を行う
- ・公開鍵はサーバーが保有。秘密鍵を持っているユーザーだけログイン可能

公開鍵暗号のイメージは南京錠



南京錠と、開けられる鍵がセットになっている



南京錠は誰でもロックできる



南京錠を解除するには鍵が必要

南京錠でロックされたメッセージは鍵でのみ解除できる

- ソフィーがハリーにだけ見れるメッセージを送りたいとする
- 南京錠はハリーがみんなに配る（一般に公開する）
- 鍵はハリーだけが持っている

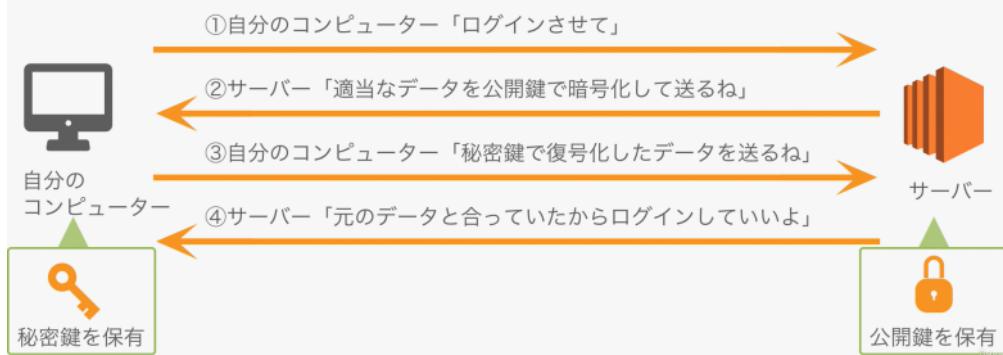


公開鍵を用いて暗号化したメッセージは秘密鍵で復号化



公開鍵暗号を用いて認証を行うのが公開鍵認証

AWSでは、EC2にログインする際に、公開鍵暗号の仕組みを用いて認証を行うことで、サーバーの持ち主だけがログインできるようにしている



ポート番号

2023年7月19日 11:30

プログラムのアドレス

同一コンピューター内で通信を行うプログラムを識別するときに利用される

決め方

標準

- ・代表的なプログラムが使うポート番号 → あらかじめ決められている
- ・ウェルノウンポート番号と呼ばれる 0～1023までのいづれか
- ・(例)SSH : 22番 SMTP : 25番 HTTP : 80番 HTTPS : 443番
- ・接続元が接続先のポート番号を省略したときは、ウェルノウンポート番号が使用される

動的

- ・サーバーはポート番号が決まっている必要があるが、接続元は決まっていなくてもいい
- ・クライアントのポート番号がOSが他のポート番号とかぶらないように、ランダムに決める
- ・動的に割り当てる番号は49142～65535までのいづれかの整数値をとる

接続

2023年7月19日 11:46

sudo lsof -i -n -P

(LISTEN) → 他のコンピューターから待ち受けているポート

(ESTABLISHED) → 現在の他のコンピューターと接続中のポート

apacheインストール

sudo yum -y install httpd

サービスをスタート

sudo systemctl start httpd.service

確認

sudo systemctl status httpd.service

→active(running)になっていることを確認する

全てのプロセスをCPUやメモリの使用率つきで表示

ps -axu

ps -axu | grep httpd

自動起動設定

sudo systemctl enable httpd.service

確認

sudo systemctl is-enabled httpd.service

→enabled

ファイアウォール

2023年7月19日 13:03

「通してよい通信だけを通して、それ以外は通さない」機能の名称

AWSではセキュリティグループがファイアウォールの役割を担っている

インバウンド

サーバーへ入ってくる通信

アウトバウンド

サーバーから出していく通信

以後ルート権限

```
sudo su -
```

https有効

```
yum install mod_ssl
```

証明書作成

```
cd /etc/httpd
```

```
make-dummy-cert server.pem
```

ファイルに書きこむ

```
vi conf.d/ssl.conf
```

```
→SSLCertificateFile /etc/httpd/server.pem
```

```
→SSLCertificateKeyFile /etc/httpd/server.pem
```

セキュリティグループ作成

443を許可

リスタート

```
systemctl restart httpd.service
```

確認

```
curl -k https://localhost
```

ブラウザ確認

route 53で追加したAレコードでアクセスする

Elastic IPアドレス

2023年7月19日 13:48

- ・インターネット経由でアクセス可能な固定グローバルIPアドレスを取得でき、インスタンスに付与できるサービス
- ・そのインスタンスを削除するまでは、ずっとそのIPアドレスを使用できる
- ・EC2インスタンスに関連付けられていて、そのインスタンスが起動中であれば無料。そうでないと課金される

IPを固定 予約できる

セクション5

2023年7月19日 14:04

- ・ ドメインについて
- ・ DNSについて
- ・ ドメインを購入
- ・ Route 53について学ぶ
- ・ Route 53でDNSを設定する

ドメイン

2023年7月19日 14:10

ドメイン名はどのような構造になっているの？

ドメイン名はピリオドで区切られた構造をしている



ドメインは誰が管理しているの？

ドメイン名全体はICANNが管理していて、トップレベルドメインごとにレジストリが管理。販売はレジストラとリセラが行う。



DNS

2023年7月19日 14:16

ネームサーバー

ドメイン名とそれに紐づくIPアドレスが登録されているサーバー

階層ごとにネームサーバーが配置され、そのネームサーバーが配置された階層のドメインに関する情報を管理

フルリゾルバ

「どのドメインに紐づくIPアドレスを教えて」と問い合わせると、色々なネームサーバーに聞いてIPアドレスを教えてくれるサーバー

リソースレコードのタイプ	内容
Aレコード	ドメインに紐づくIPアドレス
NSレコード	ドメインのゾーンを管理するネームサーバー
MXレコード	ドメインに紐づくメール受信サーバー
CNAME	ドメインの別名でリソースレコードの参照先
SOA	ドメインのゾーンの管理情報

Route 53

2023年7月19日 15:43

AWSのDNSサービス。ネームサービスの役割を果たす

特徴

- ・高可用性。SLA100%
- ・高速。エッジロケーションの中で最も近いロケーションから応答を返す
- ・フルマネージドサービス。DNSサーバー設計・構築・維持管理が不要

重要概念

- ・ホストゾーン

DNSのリソースレコードの集合

- ・レコードセット

リソースレコードのこと

- ・ルーティングポリシー

Route 53がRecord Setに対してどのようにルーティングを行うかを決める

- ・ヘルスチェック

サーバの稼働状況をチェック

ルーティングポリシー

2023年7月19日 15:48

シンプル

- レコードセットで事前に設定された値に基づいて、ドメインへの問い合わせに応答する
- 最初はこちらを使用することが多い

加重

- 複数エンドポイント毎に設定された重みづけに基づいて、ドメインへの問い合わせに応答する
- 提供リソースに差がある場合や、ABテスト時に使用

レイテンシー

- リージョン間の遅延が少ない方のリソースヘルーティングする
- マルチリージョンにリソースが存在する場合に使用

位置情報

- クライアントの位置情報に基づいて、ドメインへの問い合わせに応答する
- コンテンツのローカライズや、地域限定配信時に使用

フェイルオーバー

- ヘルスチェックの結果に基づいて、利用可能なリソースヘルーティングする
- 障害発生時にSorryサーバーに簡単に切り替えられる

セクション6

2023年7月19日 16:22

- ・RDSについて学ぶ
- ・プライベートサブネットを作成
- ・RDSを設置
- ・WebサーバーからRDSに接続

RDS

2023年7月19日 16:25

特徴

- ・可用性の向上
- ・パフォーマンスの向上
- リードプリカを簡単に構築
- ・運用不可の軽減

自動的なバックアップ

一日一回バックアップを自動取得（スナップショット）

スナップショットを元にDBインスタンスを作成（リストア）

自動的なソフトウェアメンテナンス

メンテナンスウィンドウで指定した曜日・時間帯にアップデートを自動実施

監視

各種メトリクスを60秒間隔で取得・確認可能

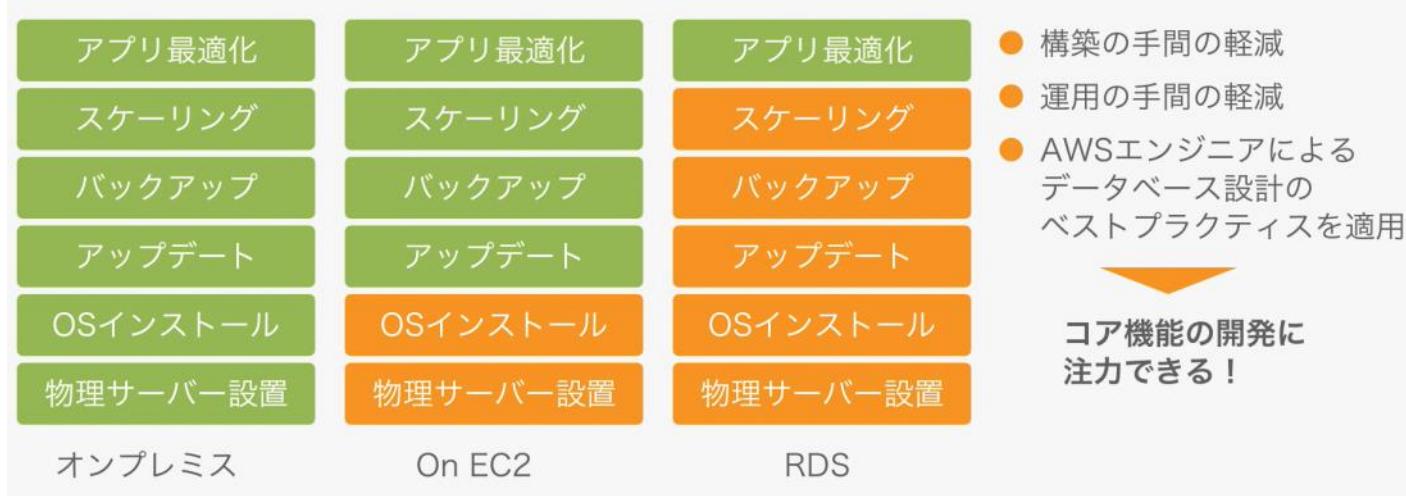
PaaSの機能

レプリケーション

リレーションナルデータベース → データベース

フルマネージド → AWSが運用管理までやってくれる

フルマネージドなリレーションナルデータベースのサービス



- 利用可能なエンジン

- MySQL
- PostgreSQL
- Oracle
- Microsoft SQL Server
- Amazon Aurora
- MariaDB

- 各種設定グループ

- DBパラメータグループ： DB設定値を制御
- DBオプショングループ： RDSへの機能追加を制御
- DBサブネットグループ： RDSを起動させるサブネットを制御

作成手順

2023年7月19日 17:01

①プライベートサブネットの作成 vpcのサブネットから

②RDSの作成準備

セキュリティグループの作成 (EC2) → ソースにはwebサーバーのセキュリティグループを選択
(IPアドレスだと動的に変化するので)

③DBサブネットグループをの作成(RDS)

①～③ 作成準備

④RDS作成 RDS→データベース

エンドポイント名(DBから確認)

⑤接続

```
mysql -h osuke-oyaizu-web.cnkcerumhjjp.ap-northeast-1.rds.amazonaws.com -u admin -p
```

MySQLをインストールする

```
# dnf -y localinstall https://dev.mysql.com/get/mysql80-community-release-el9-1.noarch.rpm
```

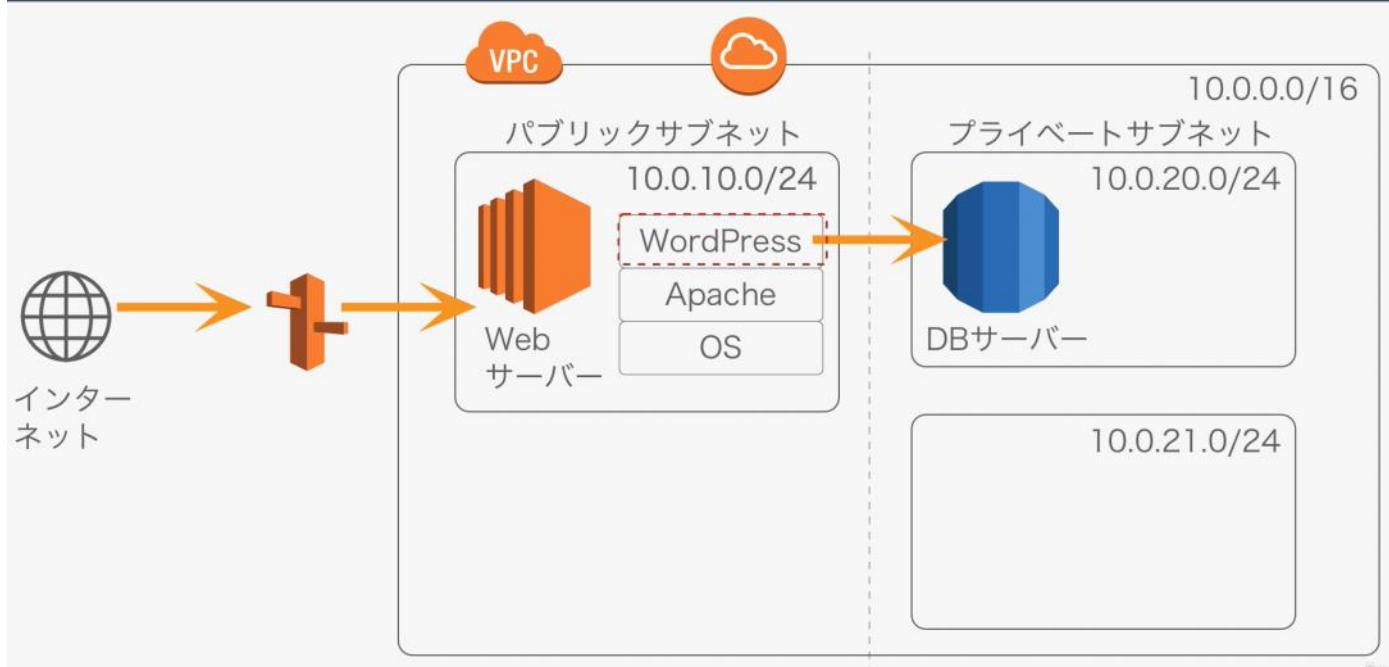
```
# dnf -y install mysql mysql-community-client
```

セクション7

2023年7月20日 10:32

- ・WordPress用のデータベースを作成
- ・WordPressをインストール
- ・WordPressを設定
- ・なぜWordPressのサイトが表示される？
- ・TCP/IPについて学ぶ
- ・HTTPについて学ぶ
- ・TCPとUDPについて学ぶ
- ・IPについて学ぶ

このセクションで構築するもの



作業内容

2023年7月20日 10:35

①WordPress用のデータベース作成 (

データベース作成

ユーザー作成

ユーザーに権限付与

②WordPressインストール

③WordPressの設定

mysqlコマンド

2023年7月20日 10:43

接続

```
mysql -h osuke-oyaizu-web.cnkcerumhjjp.ap-northeast-1.rds.amazonaws.com -u  
admin -p  
Passw0rd
```

データベース作成

```
CREATE DATABASE osuke_oyaizu DEFAULT CHARACTER SET utf8 COLLATE utf8  
_general_ci;
```

※DEFAULT CHARACTER SET デフォルト文字コード

※COLLATE 文字列の比較方法 utf8:文字列 general:多言語 ci:大文字と小文字を区別
しない

表示

```
show databases;
```

ユーザー作成

```
create user 'osuke_oyaizu'@'%' identified by 'password';
```

※@以降は接続元のホスト %:どこでも

権限付与

```
grant all on osuke_oyaizu.* to 'osuke_oyaizu'@'%';
```

※all:すべての権限付与

※<DB>.*データベースのテーブルを全て

※to 'osuke_oyaizu'@'%':ユーザーに付与

設定反映

```
flush privileges;
```

ユーザー確認

```
select user , host from mysql.user;
```

user	host
admin	%
osuke_oyaizu	%
mysql.infoschema	localhost
mysql.session	localhost
mysql.sys	localhost
rdsadmin	localhost

ユーザーログイン

```
mysql -h osuke-oyaizu-web.cnkcerumhjjp.ap-northeast-1.rds.amazonaws.com -u  
osuke_oyaizu -p  
password ← 作成した際に設定したパスワード
```

WordPress

2023年7月20日 14:22

WordPressインストール方法

```
dnf install wget php-mysqlnd httpd php-fpm php-mysqli mariadb105-server php-json  
php php-devel -y  
wget https://wordpress.org/latest.tar.gz ※wgetコマンド：urlを指定してファイルをダウンロードする  
tar -xzf latest.tar.gz ← 解凍コマンド  
[ec2-user@ip-10-0-10-11 ~]$ ls  
latest.tar.gz wordpress
```

apachが見える場所に移動させる

```
cd wordpress/  
sudo cp -r * /var/www/html/
```

```
sudo chown apach:apache /var/www/html/ -R
```

※apache:apache 所有者:所有グループをapacheに変更

※-R /var/www/html/以下をすべてapacheに変更する

```
sudo systemctl status httpd.service
```

※サービスの状態確認 active(running)←起動中

```
sudo systemctl restart httpd.service
```

インスタンスを再起動させておく

ブラウザから

<https://<ドメイン名>>

WordPress設定画面が出てくる

以下に、データベース接続の詳細を入力する必要があります。これらについて不明な点がある場合は、ホストにお問い合わせください。

データベース名	<input type="text" value="osuke_oyaizu"/>	ワードプレスで使用するデータベースの名前。
ユーザー名	<input type="text" value="osuke_oyaizu"/>	データベースのユーザー名。
パスワード	<input type="text" value="password"/>	データベースのパスワード。
データベース・ホスト	<input type="text" value="osuke-oyaizu-web.cnkcerumhjj"/>	うまくいかない場合は、ウェブホストからこの情報を取得できるはずです。localhost
テーブル接頭辞	<input type="text" value="wp_"/>	1つのデータベースで複数のWordPressインストールを実行する場合は、これを変更してください。
<input type="button" value="送信"/>		

※データベース・ホストは「RDS」→「データベース」→「エンドポイント」をコピーして貼り付け

ログインパスワード

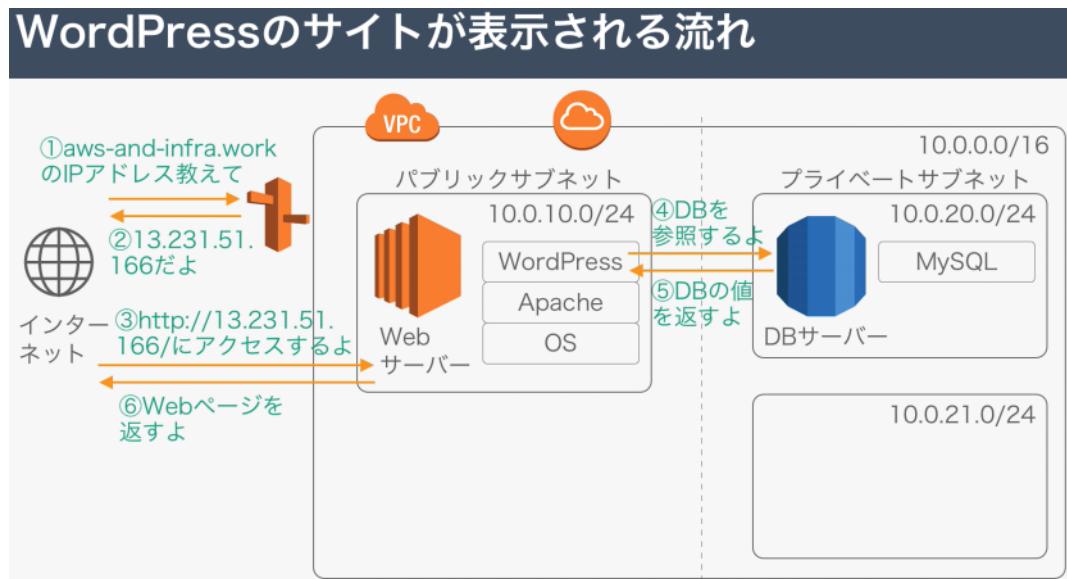
gKoK0LbJ\$IzXPw8RkP

hJjL2iv19msEpEoyKG

Q(Chq#z7at*o6s^Nck

仕組み

2023年7月20日 16:01



TCP/IP

2023年7月20日 16:11

プロトコルとは、コンピューター同士がネットワークを利用して通信するため決められた約束ごと

HTTP
TCP
UDP
IP
SMTP
IPX

[プロトコルが必要な理由]

メーカーやOSが違うコンピューター同士が通信するためには、同じ仕様でやりとりする必要がある
→プロトコルがある、同じプロトコルを使用するという同意があるからこそ、様々なコンピューター同士
が通信できている

[TCP/IP]

TCP/IPとは、TCP・IPを中心として、**インターネットを構築する上で必要なプロトコル群の総称**。インターネットを運用するために開発された



TCP/IPの階層モデル

TCP/IPの階層モデルは、インターネットでコンピュータ同士が通信する一連の処理を、4階層で表現したもの。通信に必要な機能全体を整理している

役割	プロトコル例
アプリケーション層	アプリケーション同士が会話する
トランSPORT層	データの転送を制御する
ネットワーク層	IPアドレスを管理し、経路選択する
ネットワークインターフェース層	直接接続された機器同士で通信する

アプリケーション層

一つ下のトランsport層にコネクションの確立を指示する→メールのデータをトランsport層に送信
トランsport層

コネクションを確立する

通信が終わったらコネクションを切断する（アプリケーションの指示）

どのアプリケーションと通信するか指定する

データが届いてい場合は再送する

データにヘッダーをつけてネットワーク層へ

ネットワーク層

AからBまでパケットを届ける

ヘッダーには受信側IPアドレス情報などに入る

ルーティング情報をもとに次にデータを渡すルーターやコンピューターを決定

ネットワークインターフェース層

NICを動かすためのデバイスドライバ

NIC:LANを使うための部品

デバイスドライバ:NICなどをOSが利用できるようにするソフトウェア

まとめ

- インターネットではTCP/IPプロトコルに基づいて通信が行われている
- TCP/IPでは、インターネットでコンピュータ同士が通信する一連の処理を、4階層で表現する

役割	プロトコル例
アプリケーション層	アプリケーション同士が会話する
トランsport層	データの転送を制御する
ネットワーク層	IPアドレスを管理し、経路選択する
ネットワークインターフェース層	直接接続された機器同士で通信する

HTTP

2023年7月20日 17:10

インターネットでHTMLなどのコンテンツの送受信に用いられる通信の約束ごと
クライアントがHTTPリクエストを送り、それに対してサーバーがHTTPレスポンスを返す

HTTPリクエスト

HTTPリクエストの中身

リクエストライン、ヘッダー、ボディから構成される



GET コンテンツを取得（閲覧）

POST データを送信・作成（投稿・保存）

/ → uri

urlからドメインをぬいたパス

HTTP/1.1

バージョン

Host

要求したいドメイン名

User-Agent

クライアントのアプリケーションのタイプ

Accept-Encoding

コンテンツをどの圧縮アルゴリズムならクライアントが理解できるか

HTTPレスポンス

ステータスライン	HTTP/1.1 200 OK
ヘッダー	Date: Fri, 28 Jun 2019 01:09:23 GMT Content-Type: text/html; charset=UTF-8 Cache-Control: max-age=604800 Last-Modified: Fri, 09 Aug 2013 23:54:35 GMT
ボディ	<!doctype html> <html> ... </html>

200
ステータスコード 結果

Date
いつコンテンツが生成されたか

Content-Type
返したボディがどのコンテンツタイプか

TCP/UDP

2023年7月20日 17:34

トランスポート層のプロトコル

代表的なプロトコルがTCPとUDP。通信の特性により使い分ける

TCP

- Transmission Control Protocol
- 信頼性のある通信を提供
- 信頼性を保つために、送信するパケットの順序制御や再送制御を行う
- 信頼性のある通信を実現する必要がある場合に使用する

UDP

- User Datagram Protocol
- 信頼性のない通信
- 送信するだけで、パケットが届いたかは保証しない
- 高速性やリアルタイム性を重視する通信で使用する

TCP

TCPは通信を制御するプロトコル。データの到達確認や、コネクション管理を行う

データの到達確認

- 送信したデータが届いたかを確認する
- 届いていなければ再送する
- 確認応答とシーケンス番号を使用することで、再送制御などを行う



コンピュータA

①データ「1~1000」

②確認応答「次は1001」



コンピュータB

③データ「1001~2000」

④確認応答「次は2001」

コネクション管理

- 通信相手との間で通信を始める準備をしてから通信を行う
- コネクション指向の通信を提供する



コンピュータA

①SYN (コネクション確立要求)

②SCK (SYNに対する確立応答)
SYN (コネクション確立要求)

③ACK (SYNに対する確認応答)



コンピュータB

- コネクション確立完了
- データを転送

④FIN (コネクション切断要求)

⑤ACK (FINに対する確認応答)。FIN (切断要求)

⑥ACK (FINに対する確認応答)

UDP

送信元ポート番号	宛先ポート番号	パケット長	チェックサム
データ			

TCP

送信元ポート番号	宛先ポート番号	シーケンス番号	チェックサム
データオフセット	予約	コントロールフラグ	ウィンドウサイズ
確認応答番号	緊急ポインタ	オプション	パディング
データ			

● トランスポート層は、アプリケーション間のコネクションの確立・切断を担う

● どのアプリケーションと通信かするかを指定するのがポート番号

● 代表的なプロトコルがTCPとUDP

TCP

- 信頼性のある通信を提供
- 信頼性を保つために、送信するパケットの順序制御や再送制御を行う
- 信頼性のある通信を実現する必要がある場合に使用する

UDP

- 信頼性のない通信を提供
- アプリケーションから送信要求のあったデータをそのままネットワークに流す
- 動画や電話など、即時性が必要な通信に向いている

ヘッダーのフォーマット

送信元IPアドレス	宛先IPアドレス	バージョン	ヘッダ長
サービスタイプ	パケット長	識別子	フラグ
フラグメントオフセット	生存時間	プロトコル	ヘッダチェックサム
オプション	パディング		
データ			

- ネットワーク層の役割は、最終的な宛先のコンピュータにパケットを届けること
- IPの役割は、IPアドレス、終点コンピュータまでのパケット配達（ルーティング）、パケットの分割・再構築処理の3つ
 - IPアドレス：ネットワーク上で、通信を行う宛先を識別するのに使われる
 - ルーティング：宛先IPアドレスのコンピュータまでパケットを届ける
 - パケットの分割・再構築処理：各ネットワークインターフェースの最大転送単位より小さくなるようにパケットを分割して送信し、終点コンピュータで再構築する
- IPヘッダーに、送信元IPアドレスと宛先IPアドレスが含まれている

HTTPとTCP違い

2023年11月22日 11:50

TCPの中にHTTPやNFSが存在する

例えばヘルスチェックでHTTP5000にするとアプリケーションに
アクセスできること + アプリケーションが正常に動作していることをチェックできる

TCPの場合アクセスできるかどうかしか確認ができないのでアプリケーションが動作してい
なくともHealthyが返ってくる

まとめ

2023年7月20日 17:47

基礎編（構築編）

セクション	1	2	3	4	5	6	7
AWS	概要	初期設定					
インフラ	概要	-	IPアドレス ルーティング	SSH 公開鍵認証 ポート番号 ファイアウォール	ドメイン DNS	-	HTTP TCP UDP IP

セクション8

2023年7月20日 17:48

- インフラ設計における重要なポイント
- このセクションで目指す状態
- S3について学ぼう
- S3のバケットを作成しよう
- WordPressの画像をS3にアップロードしよう
- CloudFrontによる高速化について学ぼう
- CloudFrontを設定して高速化しよう

インフラ設計における重要な観点

観点	内容	具体的な指標例
可用性	サービスを継続的に利用できるか	稼働率、目標復旧時間、災害対策
性能・拡張性	システムの性能が十分で、将来的においても拡張しやすいか	性能目標、拡張性
運用・保守性	運用と保守がしやすいか	運用時間、バックアップ、運用監視、メンテナンス
セキュリティ	情報が安全に守られているか	資産の公開範囲、ガイドライン、情報漏えい対策
移行性	現行システムを他のシステムに移行しやすくなっているか	移行方式の規定、設備・データ、移行スケジュール

オンライン上のファイルの保存場所

画像の保存場所をWebサーバーではなくS3にする理由

- Webサーバーのストレージが画像で一杯になるのを防ぐ
- HTMLへのアクセスと画像へのアクセスを分けることで負荷分散する
- サーバーの台数を増やすしやすくする
 - Webサーバー上に画像が保存されていると、Webサーバーの台数を増やした時に、画像を同期する必要があり、スケールアウトが難しい
 - 画像の保存場所は分離されていたほうがWebサーバーの台数を簡単に増やすことができる
- コンテンツ配信サービスから配信することで、画像配信を高速化できる

S3は、安価で耐久性の高いAWSのクラウドストレージサービス

特徴

- 0.023USD/GB・月と、安価。1GB約3円/月
- 99.99999999%の高い耐久性
- 容量無制限。1ファイル最大5TBまで
- バケットやオブジェクトに対してアクセス制限を設定できる

重要概念

- バケット
 - オブジェクトの保存場所。名前はグローバルでユニークな必要あり
- オブジェクト
 - データ本体。S3に格納されるファイルで、URLが付与される
 - バケット内オブジェクト数は無制限
- キー
 - オブジェクトの格納URLパス

S3のよくある利用シーン

- 静的コンテンツの配信
 - img画像はS3から配信する
- バッチ連携用のファイル置き場
 - S3にファイルを置いて、バッチでそのファイルを参照して処理を行う
- ログなどの出力先
 - 定期的にS3にログを送る
- 静的ウェブホスティング
 - 静的なウェブサイト（ランディングページなど）をS3から公開する

EBSとの違い

EBS→確保した分だけ課金

S3→使用分だけ課金

ストレージ

<https://ops.jig-saw.com/tech-cate/aws-storage>

バケット作成

2023年7月20日 18:03

注意点

パブリックアクセスは有効にしない
名前が世界で一意でないと作成できない

Offload Media

2023年7月21日 10:07

- ・Offload Mediaでは「My server is on Amazon Web Services and I'd like to use IAM Roles」を使用
※EC2にIAMロールを割り当てる必要がある(ロール名:Lab1_WordPressOnEC2)
- ・viエディタで良い
- ・まだ画像は見れない、後ほどCloudFrontが必要。理由はブロックパブリックアクセスが有効のため
[EC2 インスタンスで稼働する WordPress サイトの画像を WP Offload Media Lite プラグインを使って S3 バケットに保存するための IAM ロールの設定 - サーバーワークスエンジニアブログ \(serverworks.co.jp\)](#)

CloudFront

2023年7月21日 10:06

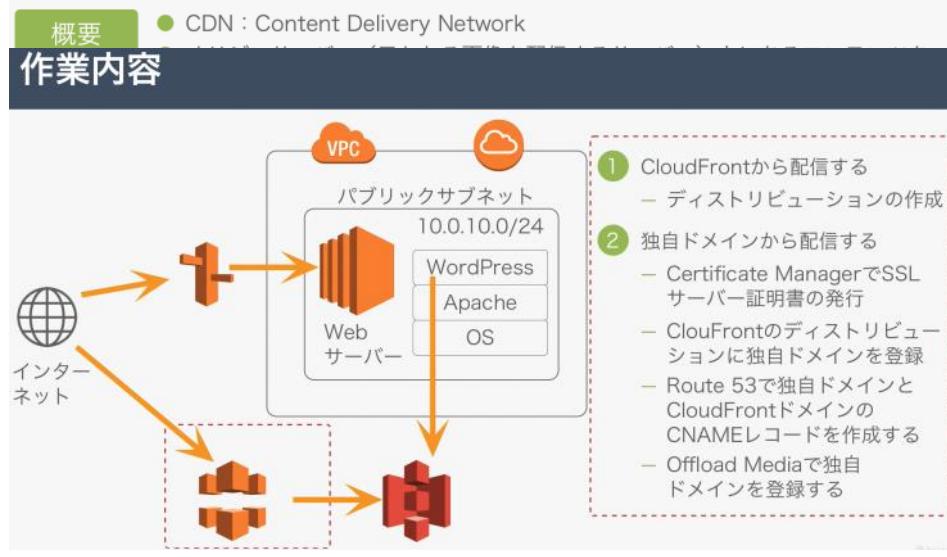
CloudFrontを通してS3にアクセスさせればS3でパブリックアクセス許可しなくてもいい

CDNユーザーから近いところから配信キャッシュを

今回は画像をブラウザに表示させるために「ブラウザ」→「CloudFront」→「S3バケット」
という順番で画像を取りに行く

CloudFrontのURLは証明書はAWS側が発行してくれる

CloudFrontは、高速にコンテンツを配信するサービス (CDNのサービス)



ステップ2

- ・まずOffload Mediaで独自ドメインを登録 → 画像のurlが独自ドメイン（独自ドメインが画像のS3のurlと紐づいていない）
- ・route53でCNAMEで独自ドメインにアクセス→CloudFrontのドメインにアクセス
- ・独自ドメインにアクセスがあったら、CLOUDFRONTのドメイン名にルーティングするように
- ・ルーティングされた際に、独自ドメインとCLOUDFRONTのドメイン名の紐付けをCLOUDFRONTが認識できるようにCLOUDFRONTのディストリビューションでも、独自ドメインを代替ドメインとして登録する。
- ・代替ドメインを登録する際にSSLサーバー証明書が必要になってくるので、そちらをCertificateマネージャーから発行する

手順(証明書など)

2023年7月21日 15:02

AWS Certificate Manager (ACM) の機能

- ▶ (例) CloudFront ディストリビューションに証明書を設定



ディストリビューション作成

- ・ブロックパブリックアクセスが有効な状態では、追加の設定が必要。CloudFrontからS3へのアクセスを許可する設定。

[WP Offload Media プラグインで WordPress サイトを S3 バケットにオフロードするベストプラク](#)

[ティス設定 - TechHarmony \(usize-tech.com\)](#)

→「CloudFrontにS3 Originを追加(OAC設定)」を参照

- ・ビューアプロトコルは HTTPS only

コードに認証情報を直に書くのはタブー

ACM証明書

- ・無料
- ・ALBを挟まないと使えない、AWSのサービスにのみ使える

証明書発行

443番を使うために証明書がいる

CNAME cloudfont.lab2.ahaws.toyota-bibliotheca.com

※名前解決ができたら発行するので信頼できる

証明書

*.lab2.ahaws.toyota-bibliotheca.com

lab2.ahaws.toyota-bibliotheca.com

CNAMEレコードをRoute 53に追加

↓検証終了後

代替ドメイン名 (CNAME) - オプション
このディストリビューションで提供されるファイルの URL で使用するカスタムドメイン名を追加します。

削除項目を追加

① 代替ドメイン名のリストを追加するには、[一括エディタ](#) を使用します。

カスタム SSL 証明書 - オプション

AWS Certificate Manager から証明書を関連付けます。証明書は、米国東部 (バージニア北部) リージョン (us-east-1) にある必要があります。



② [*.lab2.ahaws.toyota-bibliotheca.com](#) [証明書をリクエスト]

ディストリビューションドメイン名をRoute53でCNAMEで追加

レコード名 [情報](#)

レコードタイプ [情報](#)

CNAME - 別のドメイン

ルートドメインのレコードを作成するには、空白のままにします。

エイリアス

値 [情報](#)

WordPressの設定

「設定」 → 「Edit」から

Delivery

1. Select Delivery Provider

Amazon CloudFront

Fast Private Media Supported with [upgrade](#)



Cloudflare

Fast No Private Media



StackPath

Fast No Private Media



Amazon S3

Slow Private Media Supported



Other

Fast No Private Media



The screenshot shows the 'Delivery Settings' tab selected in the navigation bar. On the left, a sidebar lists various settings categories: メディア, 固定ページ, コメント, 外観, プラグイン (with a red notification badge), ユーザー, ツール, and 設定. Under '設定', there are links for 一般, 投稿設定, 表示設定, ディスカッション, メディア, パーマリンク, プライバシー, and AWS Offload Media.

The main content area displays delivery provider information:

- Amazon CloudFront** (CloudFront Distributions)
Edit
- A green checkmark icon indicates the delivery provider is successfully connected and serving offloaded media. A button labeled 'Check again' with a timestamp '6分 ago' is shown.

Two toggle switches are present:

- Deliver Offloaded Media**: Describes serving offloaded media files by rewriting local URLs to point to Amazon CloudFront. A link 'How URL rewriting works' is provided.
- Use Custom Domain Name (CNAME)**: Describes serving media from a custom domain pointed to Amazon CloudFront. A link 'How to set a custom domain name' is provided.

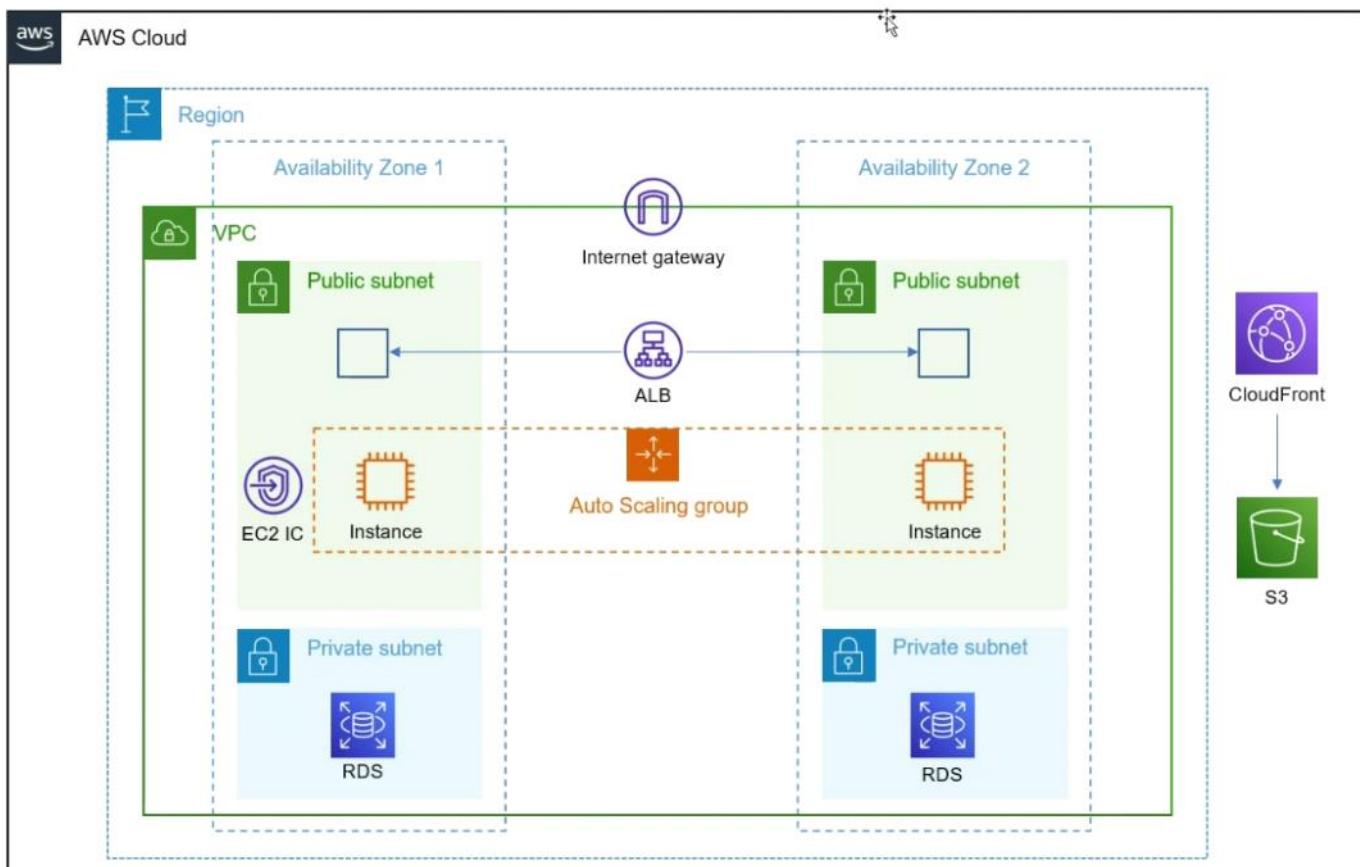
まとめ

2023年7月21日 15:42

セクション9

2023年7月21日 15:47

- このセクションで目指す状態
- 稼働率を上げる方法を学ぼう
- サーバー構成のベストプラクティスを学ぼう
- ELBについて学ぼう
- AMIからEC2を起動しよう
- ELBで負荷分散しよう
- ELBを運用する際のポイント



稼働率

2023年7月21日 15:56

① 要素単体の稼働率を高くする

② 要素を組み合わせて、全体の稼働率を高くする

③ 負荷を適切なプロビジョニングで回避する

要素を組み合わせることで、サービスの構成を冗長化しよう

冗長化構成

- └ Active-Active : 冗長化した両方が利用可能
- └ Active-Standby : 冗長化した片方は利用不可能
 - └ Hot Standby : スタンバイ側は普段起動しすぐに利用可能
 - └ Warm Standby : スタンバイ側は普段起動しているが、利用するのに準備が必要
 - └ Cold Standby : スタンバイ側は普段停止している

アクセス数などを予測し適切にリソースを準備する（プロビジョニング）ことで、負荷をさばけるようにしよう

スケールアップ

- 個々の要素の性能を向上させる
- ある程度の規模まではスケールアップがコストパフォーマンスがよいが、一定範囲を超えると悪くなる

スケールアウト

- 個々の要素の数を増やす
- ある程度の規模を超えそうであれば、スケールアウトで対応する
- 最低限用意しておくべきがN+1構成、安心なのはN+2構成

● 稼働率を高くするためには、障害発生間隔を長くするか、平均復旧時間を短くする。そのために、冗長化を行うというのが基本的な考え方

● 稼働率を上げる具体的な方法

- └ 要素単体の稼働率を高くする
- └ 要素を組み合わせて、全体の稼働率を高くする
- └ 負荷を適切なプロビジョニングで回避する

ベストプラクティス

2023年7月21日 16:03

- サーバー構成のベストプラクティスを学ぼう
- パターン：Webサーバー×1、DBサーバー×1 構成
 - 1台でサーバースペックが足りなくなったら、DBを別のサーバーに切り出す
- パターン：Webサーバー×2、DBサーバー×1 構成
 - Web側の性能が足りない時に、Webサーバーを複数台使うことで、Webの冗長化と負荷分散を行う
- パターン：Webサーバー×2、DBサーバー×2 構成
 - DBをマスタースレーブ方式にすることで、DBの冗長化を行う

ELB

2023年7月21日 16:06

Webの場合をALBを使用する

存在するだけでお金がかかる

WAFでセキュリティ強化

ACM証明書を使用できる

EC2インスタンスにデータを保持する構成では使用できない

インターネットから来たものに対してのみ分散する役割がある。

ターゲットグループに割り当てたものに対して動作する

Webアクセスが来た時の動き

Internet gateway → ALB → EC2インスタンス → ALB

ELBは、AWSクラウド上のロードバランサー

概要

- 複数のEC2インスタンスに負荷分散する
- 複数のアベイラビリティゾーンにある複数のEC2インスタンスの中から正常なターゲットにのみ振り分ける（ヘルスチェック）

特徴

- **スケーラブル**： ELB自体も負荷に応じて自動でスケールアウト・スケールインする
- **アベイラビリティゾーンをまたがる構成**： ELBを利用する場合、一つのリージョンを選び、そのリージョン内のアベイラビリティゾーンはまたがるように構成できる
- **名前解決**： ELBにはDNS名が割り当てられる。ELBへの接続ポイントへのアクセスにはDNSを使用する
- **安価な従量課金**： 従量課金で利用可能
- **マネージドサービス**： 運用が楽

- ロードバランサーは、各サーバーにアクセスを振り分け、負荷を分散する装置
- ELBは、AWSクラウド上のロードバランサー
- 機能
 - 複数のEC2インスタンスに負荷分散する
 - 複数のアベイラビリティゾーンにある複数のEC2インスタンスの中から正常なターゲットにのみ振り分ける
- 特徴
 - ELB自体もスケーラブル
 - アベイラビリティゾーンをまたがる構成
 - ELBへの接続ポイントへのアクセスにはDNSを使用する

インスタンスを作成

2023年7月21日 16:51

AMIでセットアップ

1aのインスタンスからイメージを作成し、「イメージ」→「AMI」から1cのインスタンスを作成

AMIはS3上に保持されるのでその分の料金がかかる

メリット) 高速

デメリット) コストがかかる(S3使用)

ユーザーデータを使用

ユーザーデータ - optional 情報

ユーザーデータを含むファイルをアップロードするか、フィールドに入力します。

```
#!/bin/bash
yum update -y
dnf install wget php-mysqlnd httpd php-fpm php-mysqli php-json php php-devel
php-gd -y

aws s3 cp s3://wordpress20230720lab1/wordpress.tar.gz /tmp
tar xzf /tmp/wordpress.tar.gz -C /var/www/html
chown apache. -R /var/www/html

systemctl start httpd.service
systemctl start php-fpm.service
```

I

メリット) コストが安い

デメリット) サービス提供までの遅延が発生する

起動だけ

AWS準備AMI > 自作AMI

- 汎用性を持たせるために、初回起動時に処理が走る

ELB作成

2023年7月21日 16:52

「EC2」→「ロードバランサー」

クライアント→アプリケーションロードバランサー→webサーバー→アプリケーションロードバランサー→クライアント

ネットワークロードバランサー（レスポンスの遅延が少ない）

クライアント→ネットワークロードバランサー→サーバー→クライアント

リバプロまではHTTPSで問い合わせていて、ブラウザが80に設定してある。

WordPressは80で返そうとするのでエラーが起きてしまう

→HTTPSのアクセスなのにHTTPでアクセスされているのでブラウザがはじく

インスタンスにELBに適応しているHTTPSのセキュリティグループがソースの80番許可のセキュリティグループを追加

名前	セキュリティグループ...	ポート範囲	プロトコル	ソース
-	sgr-060b290852331eab7	443	TCP	pl-06a3ac4b9417d8ca5
-	sgr-0c09e13f8539b1f4d	80	TCP	sg-08786e07fb0abda8b リンク
-	sgr-02c0ba838edc60ca4	すべて	すべて	sg-0e39a25c3822e3482 リンク

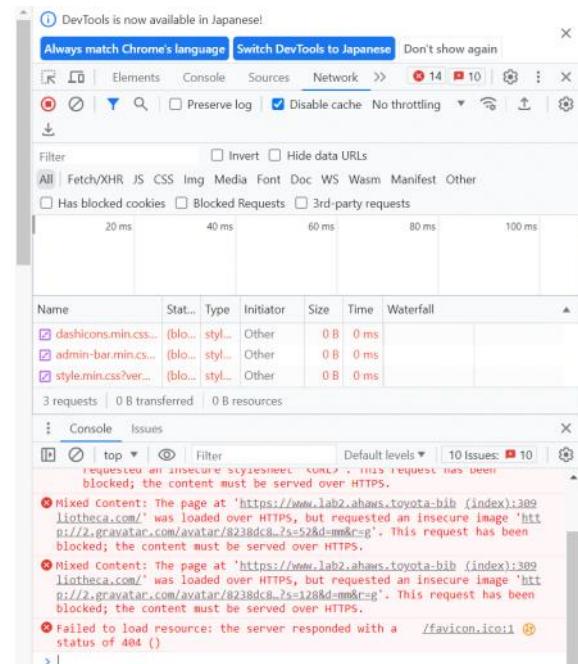
<https://qiita.com/Uryy/items/8cb7ab69b94ebbc5c46c>

インスタンスでHTTPSできたらHTTPSで返すという設定が必要になる

wp-config.php

```
if($_SERVER['HTTP_X_FORWARDED_PROTO'] == 'https') {  
    $_SERVER['HTTPS'] = 'on';  
    $_ENV['HTTPS'] = 'on';  
}
```

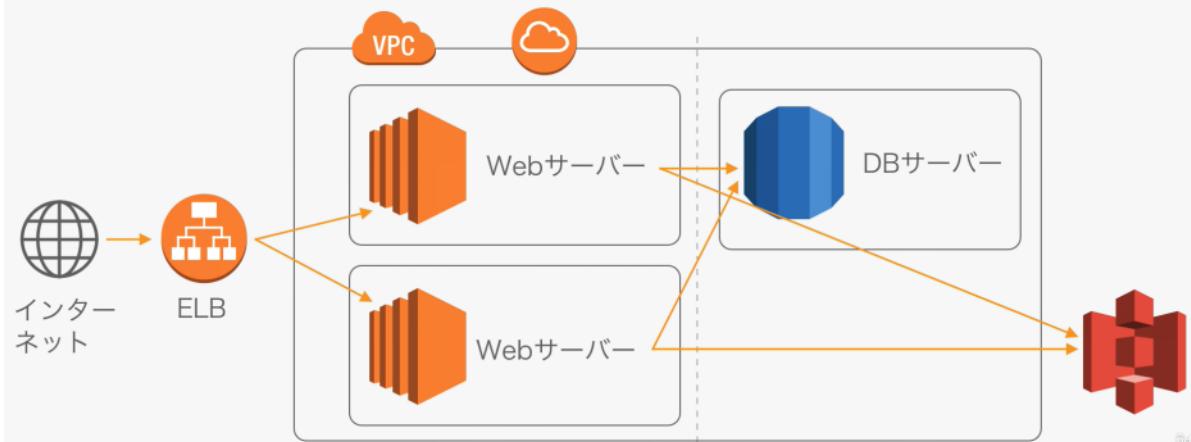
- [WordPressについて](#)
- [WordPress.org](#)
- [ドキュメンテーション](#)
- [サポート](#)
- [フィードバック](#)
- [ゼロから実践するAWS](#)
 - [ダッシュボード](#)
 - [テーマ](#)
- [サイトを編集](#)
 - [1](#)
 - [0](#)
- [新規](#)
 - [投稿](#)
 - [メディア](#)
 - [固定ページ](#)
 - [ユーザー](#)
- 検索
- こんにちは、adminさん
 - [admin](#)
 - [プロフィールを編集](#)
 - [ログアウト](#)



ELB作成ポイント

2023年7月24日 10:25

- ① サーバーをアベイラビリティゾーンをまたがって配置する
- ② Webサーバーをステートレスに構築する



- ①片方が災害などで故障してももう片方が動作する
- ②サーバーAの編集を即座にサーバーBに反映させる

負荷対策

2023年7月24日 17:13

スケールアップ[°]

サーバーの性能を上げる

スケールアウト（推奨）

サーバーの数を増やす

セクション10

2023年7月24日 10:31

セクション：【RDS】DBレイヤを冗長化しよう

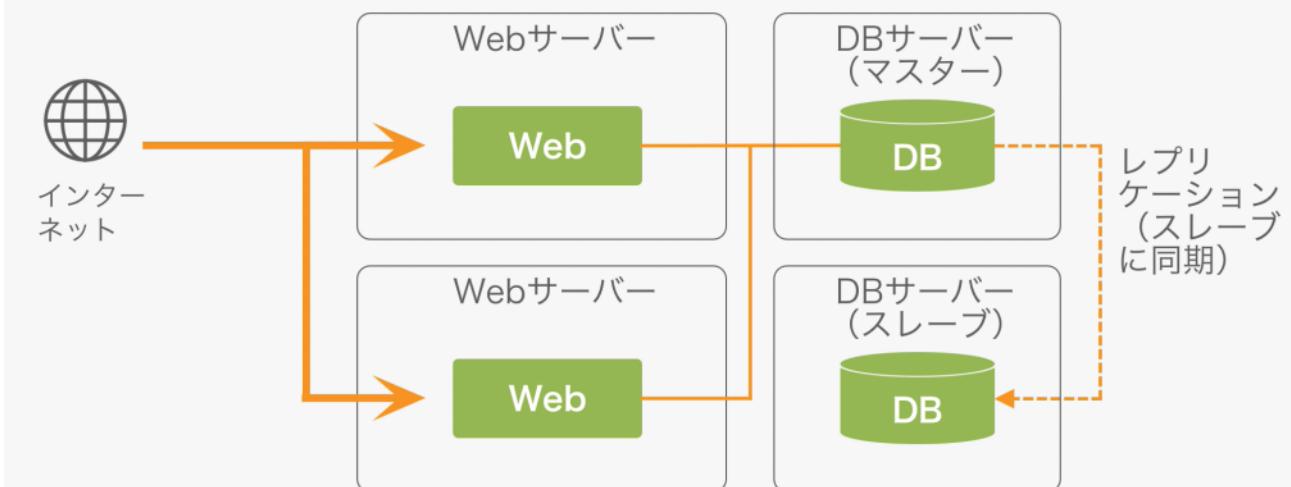
- このセクションで目指す状態
- マスタースレーブ構成を作ろう

このセクションで取り扱うインフラ設計観点

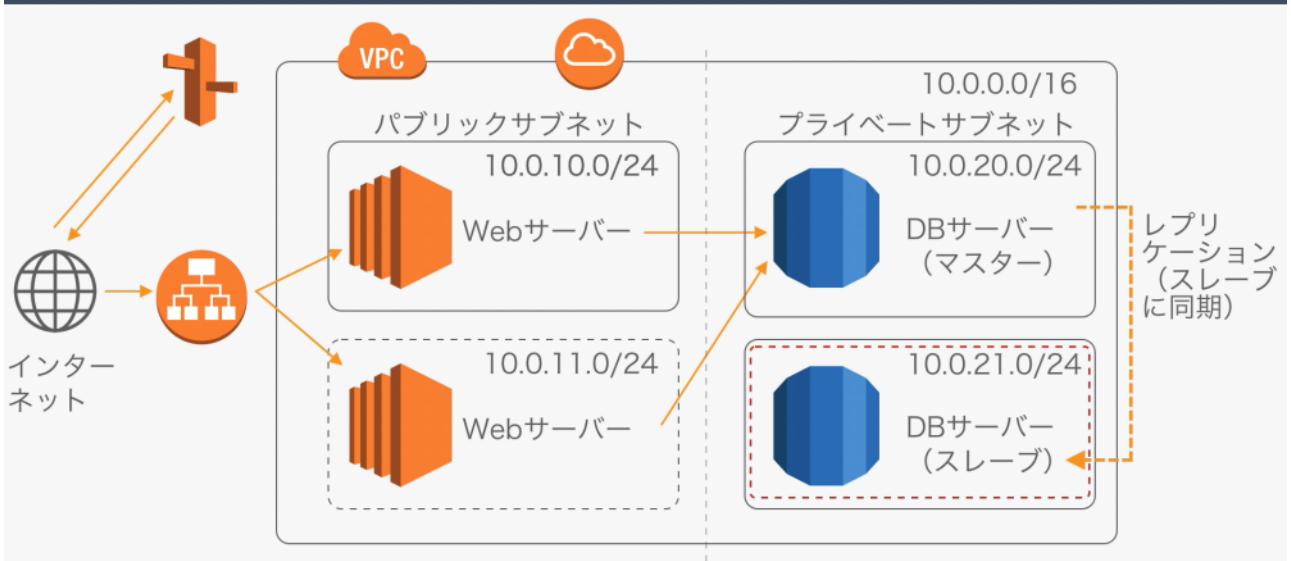
観点	内容	具体的指標例
可用性	サービスを継続的に利用できるか	稼働率、目標復旧時間、災害対策
性能・拡張性	システムの性能が十分で、将来的においても拡張しやすいか	このセクションで扱う内容 性能目標、拡張性
運用・保守性	運用と保守がしやすいか	運用時間、バックアップ、運用監視、メンテナンス
セキュリティ	情報が安全に守られているか	資産の公開範囲、ガイドライン、情報漏えい対策
移行性	現行システムを他のシステムに移行しやすくなっているか	移行方式の規定、設備・データ、移行スケジュール

パターン：Webサーバー×2、DBサーバー×2 構成

DBをマスタースレーブ方式にすることで、DBの冗長化を行う
(Webの冗長化と負荷分散、DBの冗長化ができる)



このセクションで目指す状態



マスタースレーブ構成

2023年7月24日 10:35

マルチAZ

複数のアベイラビリティゾーンにまたがってRDSインスタンスのマスターとスレーブを作成し、マスター故障時はスレーブにマスターからスレーブに対して自動的に同期してくれる

入れ替わった場合

IPアドレスは変更するがエンドポイントは変わらないので特に設定することはない

RDSの設定からマルチAZをありにするだけ

セクション11

2023年7月24日 10:44

セクション：【CloudWatch】システムを監視をしよう

- このセクションで目指す状態
- システム監視について学ぼう
- CloudWatchについて学ぼう
- CloudWatchの設定をしよう
- CloudWatchのアラートを確認しよう

このセクションで取り扱うインフラ設計観点

観点	内容	具体的指標例
可用性	サービスを継続的に利用できるか	稼働率、目標復旧時間、災害対策
性能・拡張性	システムの性能が十分で、将来的においても拡張しやすいか	性能目標、拡張性 このセクションで扱う内容
運用・保守性	運用と保守がしやすいか	運用時間、バックアップ、運用監視、メンテナンス
セキュリティ	情報が安全に守られているか	資産の公開範囲、ガイドライン、情報漏えい対策
移行性	現行システムを他のシステムに移行しやすくなっているか	移行方式の規定、設備・データ、移行スケジュール

システム監視

2023年7月24日 10:45

システム監視は、システムを正常な状態に保てるよう、稼働状況やリソースを監視すること

目的

- すぐに障害発生を確認できるようにする
- 復旧にすぐに取りかかれるようにする

中身

- ① 「正常な状態」を監視項目+正常な結果の形で定義する
- ② 「正常な状態」でなくなった際の対応方法を監視項目ごとに定義する
- ③ 「正常な状態」であることを継続的に確認する
- ④ 「正常な状態」でなくなった場合には通知が来るようにし、すぐ「正常な状態」に復旧させる

死活監視とメトリクス監視の二種類が大きくある

死活監視

- 正常にシステムが動作しているかを確認

メトリクス監視

- パフォーマンスを定量的に確認
- 指標を決め、指標が閾値以上・以下となっているかを把握

- ① システムや利用状況は変わるので、足りない監視を都度足していく
 - 項目が多すぎると、監視疲れする
 - システムも利用状況も変わるので、都度監視項目を調整すればOK
- ② 最初は基本的な要素でOK
 - CPU, Memory, Disk, Network の使用率・枯渇
 - これらの情報を確認できれば、障害発生時に何時が起点なのかを把握できる

CloudWatch

2023年7月24日 13:08

CloudWatchってなに？

CloudWatchは、AWSサービスの監視やモニタリングができる監視サービス

概要

- AWSサービスのメトリクス（リソースの状況）を監視する
- メトリクスに対して閾値を登録し、その条件を満たしたら通知する（アラーム発生）



Amazon SNSは、通知サービス



- CloudWatchは、AWSサービスの監視やモニタリングができる監視サービス
 - AWSサービスのメトリクス（リソースの状況）を監視する
 - メトリクスに対して閾値を登録し、閾値を超えたたら通知する（アラーム発生）
- Amazon SNSは、通知サービス
 - Topicを作成することで、Publisherがメッセージを送信し、Subscriberが通知を受信するための通信チャネルとして機能する

作業

2023年7月24日 13:09

①CloudWatchのアラームを作成

「CloudWatch」→「すべてのアラーム」→「アラームの作成」→監視したいものを選択する
例)

「EC2」→「CPUUtilization」→条件などを設定する

確認

"yes > /dev/null &"コマンドをEC2上で実行

yes : ターミナル上で標準出力で永遠に y という文字を出力する

> : yesコマンドを左に転送

/dev/null : 画面に表示されない

& : バックグラウンドで実行する

yesコマンド解除

ps aux | grep yes → 表示

kill -9 プロセスID

まとめ

2023年7月24日 13:40

- CloudWatchを設定し、監視ができるようになった
 - CloudWatchの設定
 - Amazon SNSとの連携
- やってみよう
 - CloudWatchでCPU使用率以外のメトリクスでアラームを作成してみよう

セクション12

2023年7月24日 13:44

権限がないので作成はできない

● このセクションで目指す状態

● IAMについて学ぼう

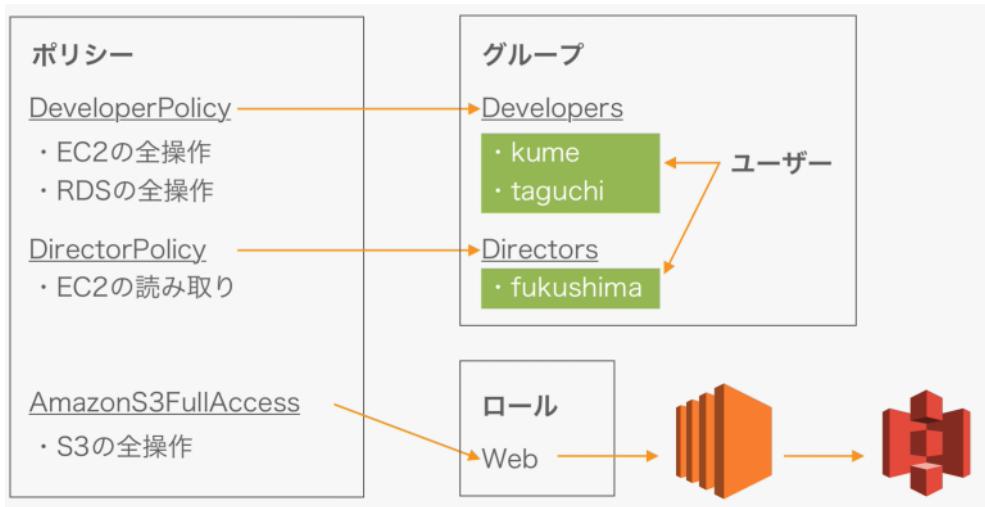
● IAMポリシーを作成しよう

● IAMグループとユーザーを作成しよう

● IAMロールを作成しよう

● IAMのベストプラクティスを学ぼう

観点	内容	具体的指標例
可用性	サービスを継続的に利用できるか	稼働率、目標復旧時間、災害対策
性能・拡張性	システムの性能が十分で、将来的においても拡張しやすいか	性能目標、拡張性
運用・保守性	運用と保守がしやすいか このセクションで扱う内容	運用時間、バックアップ、運用監視、メンテナンス
セキュリティ	情報が安全に守られているか	資産の公開範囲、ガイドライン、情報漏えい対策
移行性	現行システムを他のシステムに移行しやすくなっているか	移行方式の規定、設備・データ、移行スケジュール



IAM

2023年7月24日 13:51

IAMは、AWSのサービスを利用するユーザー権限を管理するサービス

概要

- AWSリソースをセキュアに操作するために、認証・認可の仕組みを提供する
- 各AWSリソースに対して別々のアクセス権限をユーザー毎に付与できる
- AWS IAM自体の利用は無料

用語

- ポリシー
 - アクセス許可の定義。「どのAWSサービスの」「どのリソースに対して」「どんな操作を」「許可する(許可しない)」を定義
- ユーザー
 - 個々のアカウントのユーザー
- グループ
 - IAMユーザーの集合。複数のユーザーにアクセス許可を付与する作業を簡素化
- ロール
 - 一時的にアクセスを許可したアカウントを発行できる。EC2やLambdaなどのAWSリソースに権限を付与するために使用

用語

2023年7月24日 13:53

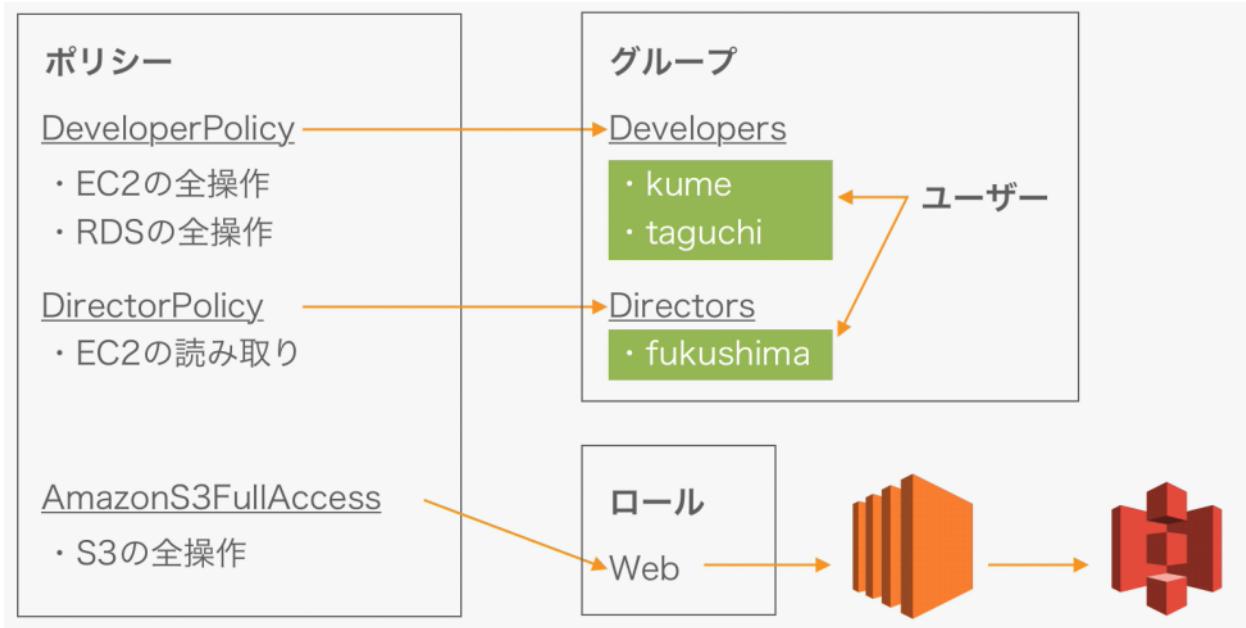
用語	意味
ポリシー	アクセス許可の定義「どのAWSサービスの」「どのリソースに対して」「どんな操作を」「許可する（許可しない）」を定義
ユーザー	個々のアカウントのユーザー
グループ	IAMユーザーの集合。複数のユーザーにアクセス許可を付与する作業を簡素化
ロール	一時的にアクセスを許可したアカウントを発行できる。EC2LambdaなどのAWSリソースに権限を付与するために使用

→ [STS](#)という一時的なアクセスキーとシークレットアクセスキーを発行してEC2からS3へのアクセスを可能にしている
IAMユーザー作成時はアクセスキーとシークレットキーを発行してくれる→恒久的に持つ

- IAMは、AWSのサービスを利用するユーザー権限を管理するサービス
- ポリシーは、アクセス許可の定義
- ユーザーは、個々のアカウントのユーザー
- グループは、IAMユーザーの集合
- ロールは、一時的にアクセスを許可したアカウントを発行

関係

2023年7月24日 14:15



ポリシー作成

2023年7月24日 14:27

「IAM」 → 「ポリシー」 → 「作成」

開発者用ポリシー(EC2とRDFの全操作ができるポリシー)

「アクション」 EC2選択後 「すべてのアクション」 「すべてのリソース」

RDSも同じ

ディレクターポリシー(EC2の読みとりのみ) ※インスタンス作成できない

「アクション」 ←読み込み 「すべてのリソース」

IAMグループ・ユーザー

2023年7月24日 14:32

「IAM」 → 「グループ」

- ・ポリシー選択 → 作成完了(それぞれ作成する)

「IAM」 → 「ユーザー」 → 「ユーザーを追加」

- ・グループに所属させるユーザー選択
- ・AWSマネジメントコンソールへのアクセスを許可
- ・パスワード設定
- ・グループ選択

IAM ロール作成

2023年7月24日 14:45

「IAM」→「ロール」→「ロールを作成」
EC2からS3を操作したいので
ロールを使用するサービスでEC2を選択
AmazonS3FullAccess ← S3に対して全ての権限をもつ
EC2インスタンスに対してロールをアタッチする

IAMのベストプラクティス

2023年7月24日 14:58

- ・個々人にIAMユーザーを作成する（ルートユーザーだと権限が強すぎる、誰がどのような操作をしたのかが分かる）
- ・ユーザーをグループに所属させ、グループに権限を割り当てる（一括でポリシーを操作できる）
- ・権限は最小限にする
- ・EC2インスタンスから実行することアプリケーションには、ロールを使用する（IAMユーザーの認証情報は恒久的なので一度キーが流出してしまうとOUT ロールは一時的なキーを使用するのでよりセキュア）
- ・定期的に不要な認証情報を削除する（使用しなくなったユーザーなどは削除する）

振り返りと今後の学習

2023年7月24日 15:05

前半（構築編）

- ・VPC設置
- ・VPC内にEC2を設置
- ・Route53でドメインでアクセスできるように
- ・RDS（データベース）設置
- ・EC2内にWordPressをインストールしてRDSと接続

振り返り：基礎編（構築編）

セクション	1	2	3	4	5	6	7
AWS	概要	初期設定					
インフラ	概要	-	IPアドレス ルーティング	SSH 公開鍵認証 ポート番号 ファイアウォール	ドメイン DNS	-	HTTP TCP UDP IP

インフラ面（ネットワークの基礎を支える概念）

- ・IP
- ・ルーティング
- ・SSH公開鍵認証
- ・ポート番号
- ・ファイアウォール
- ・ドメイン
- ・DND
- ・HTTP
- ・TCP/IP UDP

後半（発展編）

- ・S3とCloudfrontで画像配信、キャッシュで高速化
- ・ALBを用いてWebレイヤーを冗長化
- ・RDS、マルチAZを用いてDBレイヤーを冗長化
- ・CloudWatchを用いて監視
- ・IAMを用いてユーザーの権限管理

振り返り：発展編（運用編）

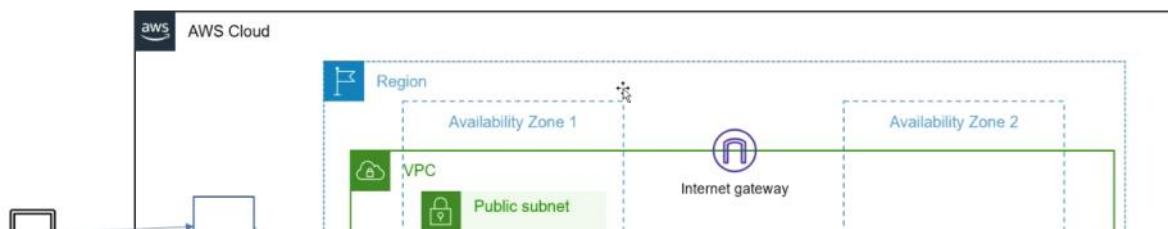
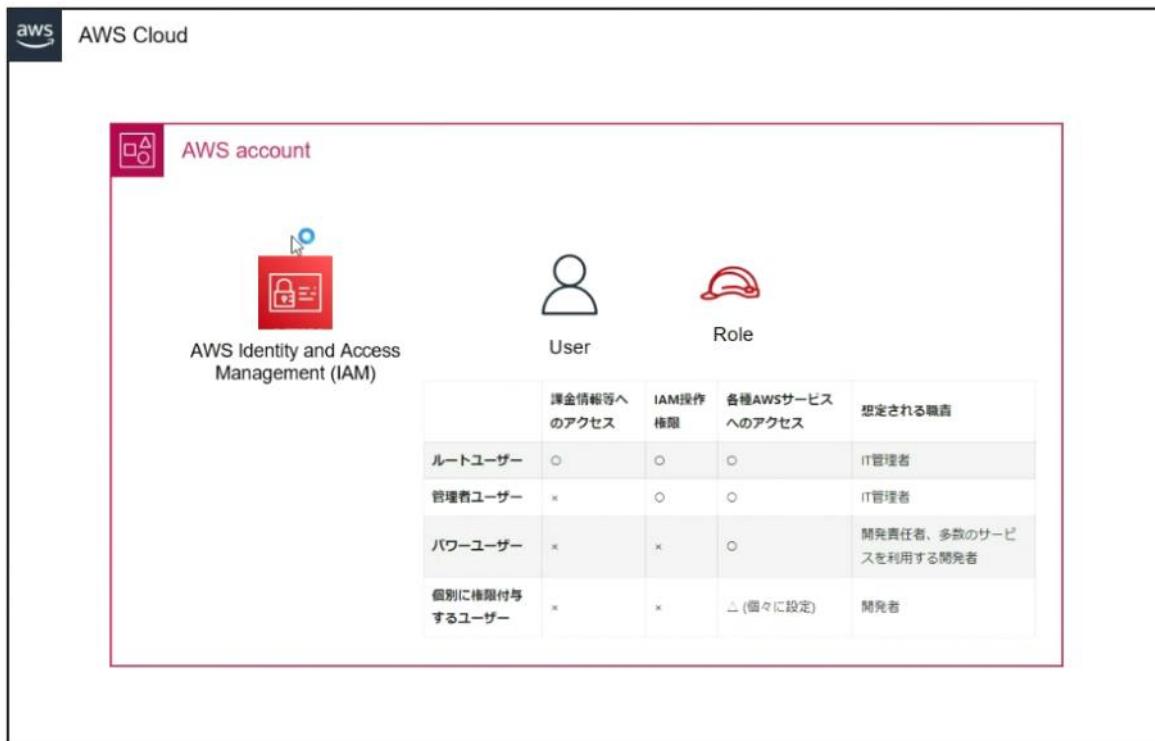
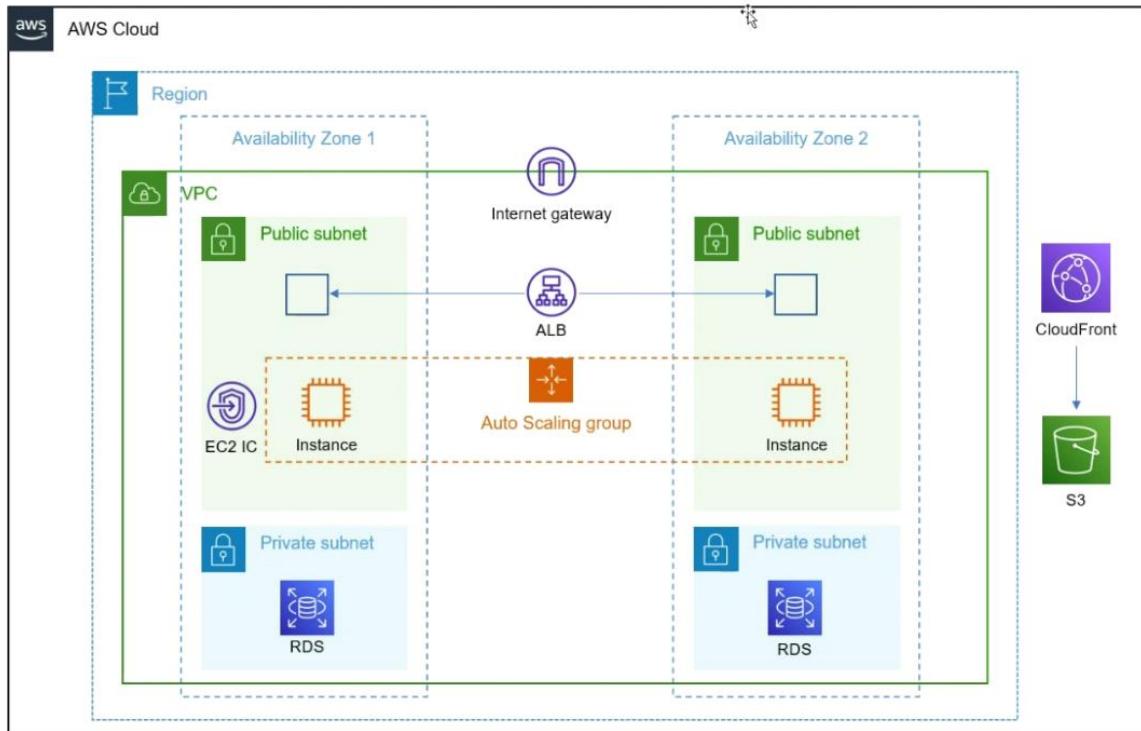
セクション	8	9	10	11	12
AWS					
インフラ	画像配信キャッシュ	Webレイヤ冗長化	DBレイヤ冗長化	監視	権限管理

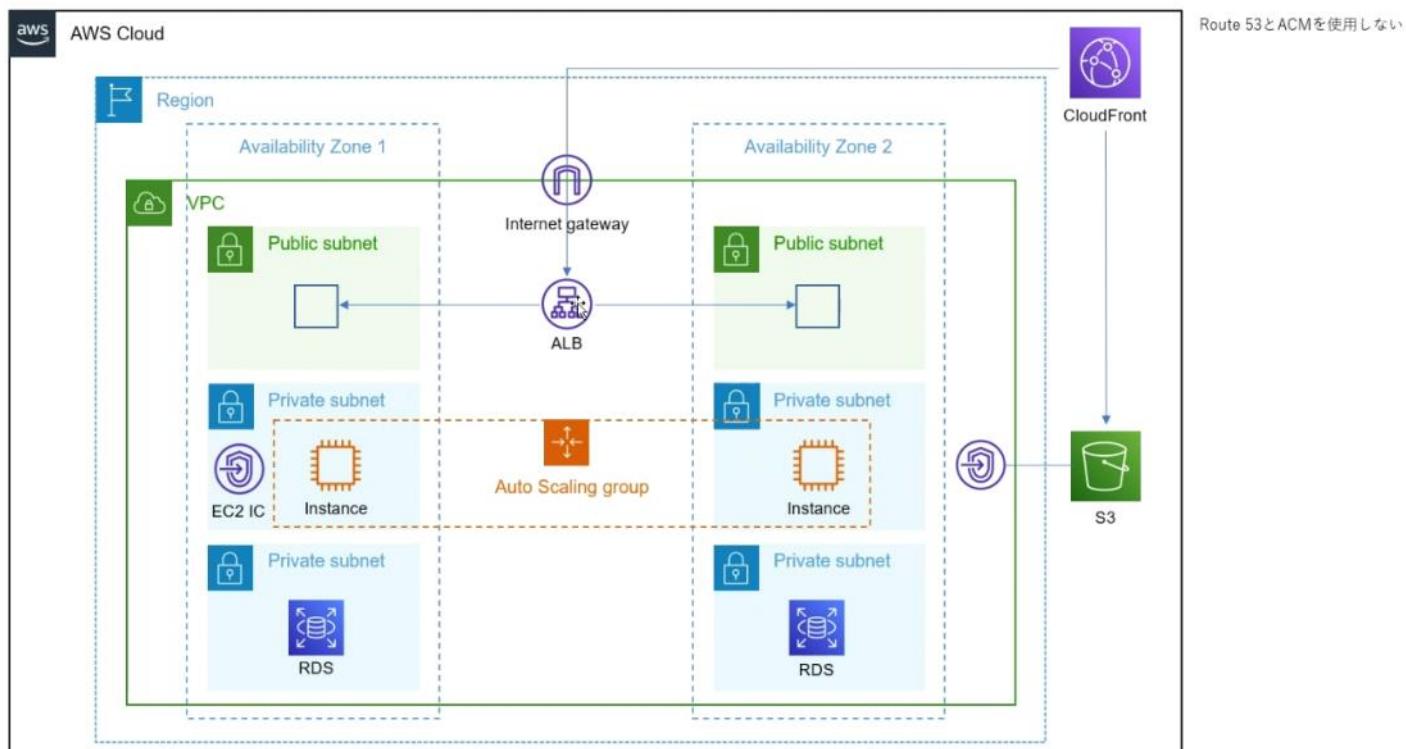
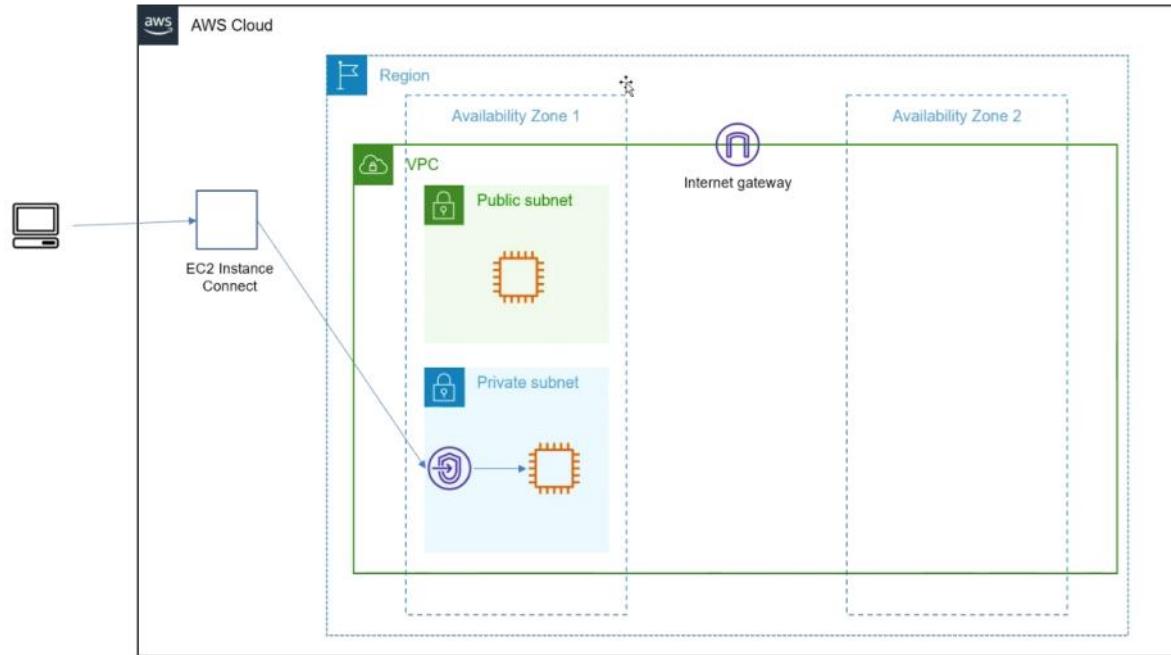
今後の学習

- ・手を動かして作成する
- ・実務、個人プロダクトでAWSを使用する
 - 実践で使ってみるのが一番
 - 動画で扱っていないサービスも使用
- ・AWS認定ソリューションアーキテクト-アソシエイトの受験もおすすめ



2023年7月24日 15:39





復習

VPC作成

サブネット作成

インターネットゲートウェイ作成

インターネットゲートウェイをVPCにアタッチ

ルートテーブルの作成

ルートの編集

(パブリック用)

The screenshot shows the AWS VPC Route Table editor. At the top, there are navigation links: VPC > ルートテーブル > rtb-0ddec9c6182dd3c7f > ルートを編集. Below this, the title ルートを編集 is displayed. The main area contains a table with four columns: 送信先 (Destination), ターゲット (Target), ステータス (Status), and 伝播済み (Propagated). There are two rows in the table:

送信先	ターゲット	ステータス	伝播済み
10.0.0.0/16	local	アクティブ	いいえ
Q 0.0.0.0/0	インターネットゲートウェイ	アクティブ	いいえ

Below the table, there are buttons for ルートを追加 (Add Route), キャンセル (Cancel), プレビュー (Preview), and 変更を保存 (Save Changes).

ALBをCloudFront経由のみアクセス可能にする方法

オリジンドメインにALBのDNS名を指定

プロトコルをHTTPのみで設定する(CloudFrontとALB間の通信は80番)

ACMを指定しない場合はAWSが用意した賞名書を使用するのでHTTPSの通信が可能

オリジンパスはブランク

セキュリティグループの設定

EC2

インバウンドルール

ALBのSGからのHTTP許可

アウトバウンド

全許可(セッションマネージャー)

RDS

インバウンドルール

EC2のSGからの3306許可

ローテーション用のLambdaのSGからの3306許可

ALB

インバウンドルール

CloudFrontのプレフィックスリストからの80番許可