システムステータスのチェック

インスタンスが実行されているAWSシステムを監視する、

これらのチェックでは修復にはAWSの関与が必要なインスタンスの根本的な問題が検出される

システムステータスチェックが失敗した場合AWSが問題を解決するのを待つか、自分で解決できるかを選択できる

システムステータスチェックの失敗の原因となる問題の例↓

・NW接続の損失

・システム電源の損失

・物理ホストのソフトウェアの問題

・ネットワーク到達可能性に影響する、物理ホスト上のハードウェアの問題

質問したいこと

・リージョンとAZについて

システム設計をするうえで、高可用性を実現するためにはマルチＡＺ構成が推奨されてて

災害復旧（ＤＲ）サイトを構築する場合は、複数のリージョンを使うマルチリージョンでシステムを構築

この場合、AZとリージョンもマルチで構築する場合があるのか？

・そもそもスナップショットとは　（　つまりログってこと？（質問

自己解決済み　4/10

・エンドポイントの意味　✔

・Nitroベースとは（ぐぐる　▲（AWS用語サービスの一つ？

・IOPS（ぐぐる　✔

・バックエンド　✔

・でタッチ、アタッチ　✔

・インメモリデータベース（ぐぐる、上のやつも

・セグメント

・デプロイ、プレイスメント違い

AWSでEC2を立てて、ターミナルでローカルからSSH接続する方法

大まかな手順

１EC2インスタンス作成

２アクセス鍵作成

３ローカルからSSH接続

詳しい記事

<https://qiita.com/takuma-jpn/items/b2c04b7a271a4472a900>

１　インターネットへの接続

ゴール

下記内容を学び、インターネットにアクセスして

アプリケーションを使用するまでの仕組みをイメージできるようにする

・インターネットにアクセスする仕組み

・インターネット上のアプリを利用する仕組み

・ファイアウォール

・ルーティング

IPアドレス

・グローバルIPアドレス（パブリックIPアドレス）

世界に一つだけ

インターネット上でデータ送受信ができるＩＰアドレス

・ローカルＩＰアドレス（プライベートＩＰアドレス）

自宅や会社など特定のネットワーク上で利用

ローカルネットワーク上でデータの送受信ができるＩＰアドレス

・ローカルＩＰアドレスとインターネットへの通信

ローカルＩＰアドレスはルータなどを通して

グローバルＩＰアドレスを取得することでインターネットへアクセス可能

サーバー

・サービスを提供するコンピューター

・基本的には私たちが利用するPCと同じクライアント

・サーバーにアクセスしてサービスを利用するコンピューター

例）サイトにブラウザでアクセスする

自分のPCがクライアント、サイトを動かしているコンピューターがサーバー

要点

・IPアドレスには「グローバル」と「ローカル」の２種類がある

・インターネットを利用するときは「グローバルIPアドレス」が必要

・サーバーはサービスを提供するコンピューター

クライアントからサーバーへの通信

・HTTPリクエスト

Hyper Text Transfer Protocol　※ローカル通信のときに利用

・HTTPSリクエスト

HTTPの暗号化通信（Secure）※インターネット通信のときに利用

・ポート番号

プロトコルに割り振られる番号 ※デフォルトの場合省略される

？ローカル通信とは、ローカル通信とインターネット通信の違い

ポート番号とは

プロトコル

クライアントがサーバーにアクセスするときの方式

クライアント側とサーバー側で合わせる必要がある

コンピューターにおけるプロトコルとは通信規約を意味する

同じ言語でない（同じプロトコルでない）と意思（処理）が正しく

伝わらない

※HTTP ICMP SSH　RDP

？ICMP SSH RDP

・ファイアウォール

不正アクセスをブロックし、安全にインターネットを

利用するための仕組み

具体的にポートやプロトコル、発信元IPアドレスなどを指定して

通信制御

・ポート番号

プロトコルに割り振られる番号

一般的な値とカスタマイズできる値がある

※ＨＴＴＰプロトコル：８０

ＨＴＴＰＳプロトコル：４４３など

・ルーティング

目的地までのルートを案内する仕組み

目的地、中継地点を指定する

・ルーター

ネットワークの中継機器

データを見て行先を割り振る

要点

インバウンド制御

・ファイアウォールで不都合なアクセスはブロック

・ポート（プロトコル）で指定する※プロトコルは言語と一緒

アウトバンド制御

・目的に応じてルーティングを指定する

？インバウンド、アウトバウンド

まとめ

・IPアドレスにはグローバルとローカルの二種類がある

・インターネットを利用する時はグローバルIPアドレスが必要

・サーバーはサービスを提供するコンピューター

・プロトコルは言語のイメージ

・インバウンド制御

ファイアウォールで不都合なアクセスはブロック

ポート（プロトコル）で指定する

・アウトバンド制御

目的に応じてルーティングを設定する

２ AWS上でのサーバー構築

ゴール

前章の内容をAWS上で構築するにはどうしたらいいか

ざっくりとしたイメージを持つ

・ＶＰＣ

・ＥＣ２インスタンス

・SecurityGroup

・RouteTable

AWSでネットワーク空間をつくる

・VPC

プライベートなクラウド環境

リージョンを指定

プライベートIPアドレスを範囲指定で利用

Subnetの割当が可能

・Subnet

VPCを用途ごとに区切ったネットワーク群

ＡＺを指定

・リージョン

物理的なロケーション

世界各国にある

・ＡＺ

リージョン内での物理的なロケーション

複数ＡＺにまたがってサーバーを起動する

ことで冗長性、耐障害性の向上

ＡＷＳ上でサーバーを起動する

・ＥＣ２インスタンス

サブネットを指定して起動

ＡＭＩ、インスタンスタイプを指定

・ＡＭＩ（マシンイメージ）

Windoes、Linuxなどの種類がある

AWS提供のものや事前設定が可能

・インスタンスタイプ

マシンタイプ、スペック

スペックがいいものほど料金が高い

・FW（SecurityGroup）を設定する

・SecurityGroup

ホワイトリスト方式

IPCIDR、プロトコル、ポート

ステートフル

・ホワイトリスト

許可するものを記載する

・ブラックリスト

許可しないものを記載する

・CIDR

範囲指定、オクテット

※10.0.0.0/32 -> 10.0.0.0

10.0.0.0/31 -> 10.0.0.0. 10.0.0.1

・ステートフル

行きの通信（インバウンド）で許可されているものは

戻りの通信（アウトバウンド）でも自動的に許可される

・ルーティング（RouteTable）を設定する

・RouteTable

送信先、ターゲット

ターゲットはCIDR形式

ロンゲストマッチ

・ロンゲストマッチ

厳密にしているCIDRが優先

※送信先が10.0.0.0/32 igw-xxxx →〇こちら

10.0.0.0/31 vgw-xxxx→×

まとめ

・サーバーの起動

VPC、Subnetの準備

EC2の起動

インバウンド制御

SecurityGroupを作成

ホワイトリスト形式で許可

・アウトバウンド制御

ルートテーブルにて行き先を設定

ロンゲストマッチ