AWS をはじめよう

mochikoAsTech　著

2018-10-08 版　　mochikoAsTech 発行

はじめに

2018 年 10 月 mochikoAsTech

この本を手に取ってくださったあなた、こんにちは、あるいははじめまして。「AWS をはじめよう」の筆者、mochikoAsTech です。

本著は、前作「DNS をはじめよう」の続きとなっています。「DNS をはじめよう」ではドメインを買って、AWS のアカウントを作り、Route53 をネームサーバとして使うところまでをやってみました。

「AWS をはじめよう」では、そのドメインを使って自分のサイトを作ってみます。AWS でウェブサーバを立てたり、DNS サーバを立てたり、WordPress をインストールしたりして、ブラウザで自分のサイトが見られるところまで一緒に頑張ってやっていきましょう！

え、この本って別の本の続きなの？「DNS をはじめよう」を先に読んだ方がいい？ と思ったあなた、ぜひ「DNS をはじめよう」を先に読んでください。

ケーキのデコレーションはスポンジを焼いてからでないと出来ないように、順を追ってやっていった方が絶対に分かりやすいです。ぜひ、はじめようシリーズ 1 作目の「DNS をはじめよう」を読んでから、「AWS をはじめよう」に帰ってきてください。

想定する読者層

本著は、こんな人に向けて書かれています。

- これからシステムやプログラミングを学ぼうと思っている新人
- ウェブ系で開発や運用をしているアプリケーションエンジニア
- 「インフラがよく分からないこと」にコンプレックスのある人
- 折角ドメインを買ったので自分のブログやポートフォリオサイトを作ってみたい人
- 自分のドメインでメールアドレスを作ってみたい人
- AWS や EC2 や ELB や RDS といった単語に興味のある人
- クラウドってなんだろう？ サーバってなんだろう？ という初心者

本著の特徴

本著では前作「DNS をはじめよう」で買ったドメインを使って、実際に WordPress で自分のサイトを作ってみます。手を動かして AWS でサーバを立てたりロードバランサーの設定をしたりしながら学べるので理解がしやすく、インフラ初心者でも安心して読み進められる内容です。

また実際にありがちなトラブルをとり上げて、

- 上手くいかないときは原因をどう調べたらいいのか？
- 問題をどう解決したらいいのか？
- どうしたら事前に避けられるのか？

を解説するとともに、実際にコマンドを叩いて反復学習するためのドリルもついています。

本著のゴール

本著を読み終わると、あなたはこのような状態になっています。

- WordPress でおしゃれなサイトができあがっている
- 使うも壊すも自由な勉強用の Linux サーバ環境が 1 台手に入る
- 「mochiko@startdns.fun」のような自分のメールアドレスでメールを受信できるようになっている
- 読む前より AWS やサーバや黒い画面が怖くなくなっている

免責事項

本著に記載されている内容は筆者の所属する組織の公式見解ではありません。

また本著はできるだけ正確を期すように努めましたが、筆者が内容を保証するものではありません。よって本著の記載内容に基づいて読者が行った行為、及び読者が被った損害について筆者は何ら責任を負うものではありません。

不正確あるいは誤認と思われる箇所がありましたら、電子版については必要に応じて適宜改訂を行いますので筆者までお知らせいただけますと幸いです。

目次

第 1 章

インフラとサーバってなに？

この章では AWS とはなにか？ そもそもクラウドとは何か？ サーバとは何か？ という基本を学びます。

1.1 AWS を理解するには先ずインフラを知ろう

AWS とはアマゾン ウェブ サービス（Amazon Web Services）の略で、欲しいものをぽちっとな！ すると翌日には届くあのアマゾンがやっているクラウドです。

「AWS はアマゾンがやっているクラウドです」と言われても、「クラウド」が分からないと結局 AWS が何なのかよく分からないままですよね。

クラウドって何なのでしょう？

クラウドだけではありません。よくクラウドと一緒に並んでいるサーバやインフラという言葉がありますが、こちらも何だか分かりますか？ IT 系で働いていても、その辺って「なんか・・・ふんわり・・・なんか雲の向こう側にある・・・ウェブサイト作るための何か・・・？」という程度の認識で、クラウドってなに？ とか、サーバってなに？ と聞かれたときに、ちゃんと説明できる人は意外と少ないのではと思います。

なので、先ずは「AWS はアマゾンがやっているクラウド」という文章の意味が分かるよう、インフラ周りから順を追って学んでいきましょう。

1.2 インフラとは？

インフラという言葉は知っていますか？

はじめて聞いたという人も、「なんとなくは分かるけど、説明してと言われたらうーん・・・」な人も、いま自分が考える「インフラ」についての説明をここに書いてみましょう。いきなり正解を聞かされるより、自分で答えを考えて書き出してみてからの方が、正解を聞いたときにきっと自分の中へより染み渡ってくるはずです。

インフラとは

では答え合わせをしてみましょう。

インフラとは**サーバやネットワークのこと**です。

そもそもインフラこと「Infrastructure」は、直訳すると基盤や下部構造といった意味です。ですので「生活インフラ」と言うと一般的には上下水道や道路、そしてインターネットなど、生活に欠かすことの出来ない社会基盤のことを指します。

そして技術用語としては、インフラはシステムやサービスの基盤となる「設備」のことをいいます。なので、分かりやすく言うと「インフラとはサーバやネットワークのこと」なのです。

これでもう会社で後輩に「インフラってなんですか？」と聞かれても、堂々と「サーバとかネットワークのことだよ」と答えられますね！

でも後輩に、続けて「え、サーバってなんですか？」と聞かれたらどうしましょう？

1.3 サーバとは？

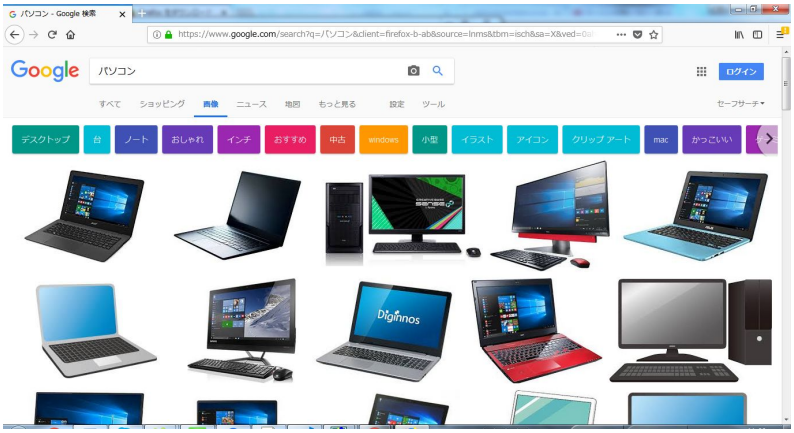
後輩から「サーバってなんですか？」という直球の質問を投げつけられたら、しっかりホームランで打ち返せますか？

しっかり答えられるか不安な人も、自信のある人も、ちょっと頭の中でサーバの姿を思い浮かべてみてください。できればイラストじゃなくて実写でお願いします。

▲図 1.1　サーバの姿を思い浮かべて描いてみよう

思い浮かびますか？ あんまり浮かばないですよね。サーバの姿がくっきり思い描ける人の方が少ないのではないかと思います。

でもこれが「パソコンの姿を思い浮かべてください」なら、きっとすぐに浮かんでくるはずです。(図 1.2)



▲図 1.2　パソコンの姿ならすぐに思い浮かぶ

でもサーバの姿は？ と考えるとなかなか思い浮かびません。
ではサーバの姿をお見せしたいと思います。

1.3.1 サーバの姿を見てみよう

▲図 1.3 サーバの姿（HPE ProLiant DL360）

これは Hewlett Packard Enterprise（ヒューレット・パッカード エンタープライズ）の HPE ProLiant DL360[*1]というラックマウント型のサーバです。DL360 は 15 年以上前から愛されているシリーズ[*2]で、日本でもっとも売れたラックマウント型のサーバと言っても過言ではないかも知れません。定価で 1 台おおよそ 50 万円以上します。

そして本は本棚に収めるように、サーバはサーバラック（図 1.4）[*3]という専用の棚に収めることが多いです。

[*1] HPE ProLiant DL360 https://www.hpe.com/jp/ja/product-catalog/servers/proliant-servers/pip.hpe-proliant-dl360-gen10-server.1010007891.html

[*2] 2018 年 8 月時点で発売されているのは HPE ProLiant DL360 Gen10 です。末尾の「Gen10」は世代（Generation）を表しているので、10 世代目ということです。

[*3] 写真のサーバラックは HPE Advanced G2 シリーズ https://www.hpe.com/jp/ja/product-catalog/servers/server-racks/

▲図 1.4　サーバを収めるためのサーバラック

先ほどの HPE ProLiant DL360 のようなサーバは、ラック（＝棚）にマウントする（＝乗せる）ことができる形状のため「ラックマウント型サーバ」、略してラックサーバと呼ばれています。

ラックマウント型のサーバは 1U（ワンユー）・2U・4U のように厚みが異なり、1U ならこのサーバラックの 1 ユニット（1 段）分、2U なら 2 ユニット（2 段）分を使うことになります。そのためラックマウント型サーバは 1U サーバという名前で呼ばれることもあります。サーバラックは 42U サイズが多く、その名前のとおり 1U サーバを 42 台収めることができます。[*4]

[*4] 但しラックに供給される電源の量や放熱の問題もあるため、実際は 42U サイズのラックにサーバ 42 台をぎちぎちに詰めることは少ないのでは、と思います。

▲図 1.5　タワー型サーバとブレードサーバ

「ラックマウント型サーバ」だけでなく、デスクトップパソコンのような「タワー型サーバ」（図 1.5）[*5]や、シャーシやエンクロージャーと呼ばれる箱の中に何本も差し込んで使う省スペースな「ブレードサーバ」[*6]など、サーバには色々な形があります。

こうしたラックマウント型サーバ、タワー型サーバ、ブレードサーバのように、手で触れる実体があるサーバのことを**物理サーバ**といいます。物理的な実態があるから物理サーバです。

そもそもですが、人がウェブサイトを作る時には土台となるサーバが必ず必要となります。たとえはてなブログで無料のブログを作ったときでも、Ameba Ownd で無料のホームページを作ったときでも、あなた自身はサーバのことなど気にも留めないと思いますが、どこかしらに必ずそのブログやサイトが乗っかっているサーバは存在しています。

ではサーバはいったいどこにいるのでしょう？

1.3.2　サーバはデータセンターにいる

前述のサーバラックや、その中に詰まったラックサーバを実際に見たことはありますか？

どんなウェブサイトも、世界中のどこかにあるサーバの中で稼動しているはずなのですが、インフラエンジニアでなければサーバを見る機会はなかなかないかも知れません。

サーバはほとんどの場合、**データセンター**と呼ばれる場所に設置されています。[*7]

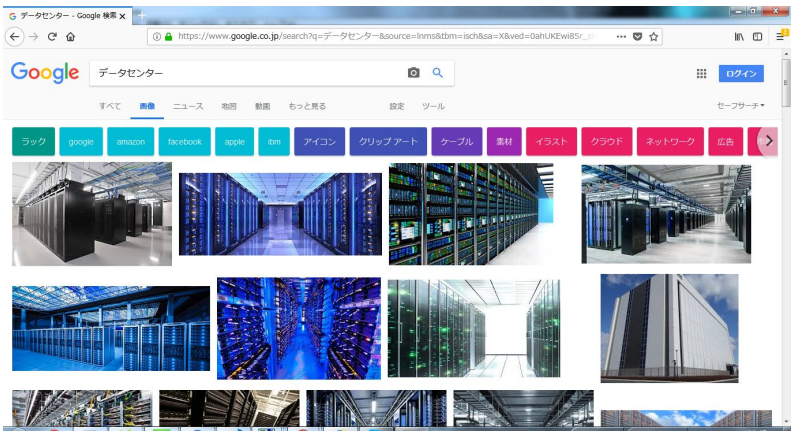
カラオケをするには防音や音響設備の整ったカラオケルームが適しているように、サー

[*5] https://www.hpe.com/jp/ja/product-catalog/servers/proliant-servers/pip.hpe-proliant-ml110-gen10-server.1010192782.html

[*6] https://www.hpe.com/jp/ja/integrated-systems/bladesystem.html

[*7] 企業によっては、オフィス内にサーバルームがあってサーバはそこにいるかも知れません。

バを動かすためのさまざまな設備が整った場所のことをデータセンター、略して DC[*8] といいます。先ほどのラックサーバがたくさん並んでいますね。（図 1.6}）



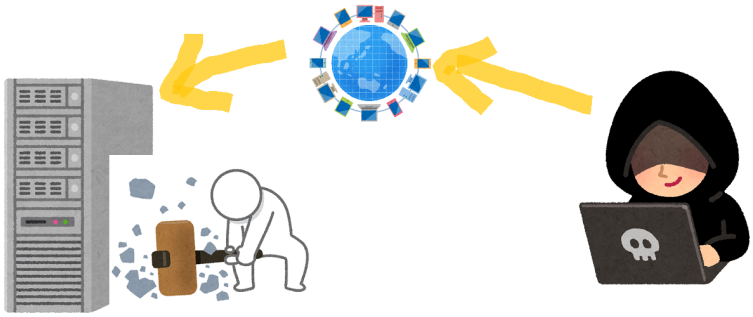
▲図 1.6　データセンターはサーバのための設備が整っている

しかし「サーバを動かすための設備」と言われても、「電源取れればそれでいいんじゃないの？ 専用の建物なんか要る？」と思われるかも知れません。サーバを動かすのに適した設備とはどんなものなのでしょう？

〈サーバに適した設備〉1. 防犯設備

もし悪い誰かが「この企業が気に食わないから商品のウェブサイトを落としてやろう」と思ったとき、あるいは「このネットショップの顧客情報を根こそぎ盗んでやろう」と思ったとき、ネット越しにサイトを攻撃したり侵入したりするだけでなく、そのサイトが動いているサーバのところまで行って物理的に破壊したり、ハードディスクを引っこ抜いて盗んだり、という手段があります。

[*8] データセンター（Data Center）の頭文字を取って DC ですが、その中でもインターネット用途向けのデータセンターはインターネットデータセンター、略して iDC と呼ばれたりします。「逆にインターネット用途以外のデータセンターってなに？」となりますが、メインフレーム（インターネット以前の時代の大型コンピュータ）向けということのようです。



▲図 1.7　攻撃や窃盗はインターネット越しでも物理的にでもできる

データセンターは後者の「物理的な攻撃や侵入」からサーバを守るための設備を整えています。

堅牢さはデータセンターによって異なりますが、たとえば次のような防犯対策が取られています。

- 所在地を一般に公開しない[*9]
- 建物自体に侵入経路となる窓がない
- 事前予約をした上で顔写真つきの身分証を提示しないと建物に入れない
- エントランスで空港と同じような手荷物チェックや金属探知機チェック、静脈認証がある
- 上着や荷物、携帯電話、カメラなどは持ち込み禁止
- 借りているサーバラックがある階にしかエレベータが止まらない
- サーバルームへの入退室は監視カメラと静脈認証で記録
- 入るときと出るときで体重が違うと出られない

また「うちは弱小サイトだから誰かのうらみも買わないし、盗まれるような個人情報もないよ」という場合でも、防犯だけでなく天災や熱の対策も必要です。

[*9] 2018 年 7 月、都内で建築現場の火災が発生した際に「この建物は AWS のデータセンターとして建築していた可能性が高い」というニュースが出ており、断定はしていなかったものの「それは報道していいの？ 周知の事実になってしまったら、もう一度同じ場所に立てるの無理なのでは・・・？」とちょっと気になりました。

〈サーバに適した設備〉2. 地震・火事対策設備

ウェブサイトはサーバ上で稼動しているため、サーバが止まればもちろんサイトも見られなくなってしまいます。[*10]

もし地震などの天災があったときにも絶対にサーバを止めないため、データーセンターの建物は耐震構造になっていることはもちろん、次のような電力供給対策もされています。

- 変電所から電力を受ける受電設備は複数用意して冗長化している
- もし停電があっても電力が途絶えないよう、電力は複数の変電所から引いている
- 両方の変電所が止まって完全に電力が途絶えたら UPS（無停電電源装置）が自動稼働
- さらに数十秒以内に自家発電機が稼動し、最低数日間は追加の燃料給油なしで稼動可能
- 燃料（重油やジェット燃料）は販売元業者と有事の優先供給の契約をしており、供給が続く限りは自家発電で稼動し続けることが可能

電源がなくなればパソコンも落ちてしまうように、サーバも、その上で稼動しているウェブサイトも、電力が供給されなければ落ちてしまいます。仮にサーバを 2 台用意して「1 台壊れても、もう 1 台でサイトは見られる！ 大丈夫！」と安心していても、データセンターの電力そのものが止まってしまえば、どちらのサーバも電源が切れてサイトは見られなくなります。電気は使えて当たり前と思いがちですが、311 の輪番停電のように「当たり前が崩れたとき」にも、いつもどおり稼動できる環境がデータセンターには求められているのです。

さらに万が一火事が起きても、サーバにじゃんじゃん水をかけて消火するわけにはいきません。火は酸素をエネルギーにして燃えるので、多くのデータセンターでは酸素以外のガスで部屋を満たして消火するガス消火設備を備えています。

〈サーバに適した設備〉3. 空調設備

そして一生懸命稼動しているサーバはとても熱くなります。皆さんのパソコンにもファンなどの冷却機構が付いていて、使っていると熱を冷まそうとしますよね。サーバも同じで、たくさんのサーバが詰まったサーバラックの裏側には、熱い空気がいっぱい吐き出されてきます。

[*10] 冗談のようですが「こちらのサーバを停止して削除しますね」「はい、使ってないのでいいです」という会話をした後で、実際にサーバを削除したら「サイトが見られなくなったんですけど！」という連絡が来た、という話も聞きます。サーバがなければサイトは見られない、というのは決して万人にとって当たり前のことではないのです。

暑い部屋ではサーバが故障したり落ちてしまったりする[*11]ため、データセンター内のサーバルームの空調はとても強く、人間が過ごすにはちょっとつらい寒さです。

このように防犯、地震・火事対策、空調といった設備が整ったデータセンターで、サーバは日々元気に稼動しているのです。

1.3.3 物理サーバと仮想サーバ

部屋を借りるとき、「1DK の部屋なら 12 万、2LDK の部屋なら 20 万」のように広いほうが家賃は高くなります。データセンターを借りるときもまったく同じで「1/2 ラックなら 12 万、1 ラックまるごとなら 20 万」のように、ラックサイズによって月額の費用が変わってきます。

そして 2LDK の部屋に 1 人で住むと、1 人で 20 万負担しなければなりませんが、シェアハウスにして 10 人で住めば 1 人当たりの家賃コストは 2 万円に下がります。

サーバラックも同様に、42U のラックに 1U サーバを 1 台しか乗せないより、42 台詰め込んだ方が 1 台当たりのコストは下がります。

2000 年代前半、インターネットが盛り上がってきてサーバがどんどん必要になってきた頃、42U のラックになんとかもっとたくさんのサーバを詰め込めないか？ と試行錯誤した結果、前述の省スペースなブレードサーバや、1U の半分サイズで奥と手前に 2 台収納できる 1U ハーフサイズのサーバなどが台頭してきました。

しかし 1 つのラックに割り当てられた電源の量には上限があるため、42U にぎちぎちに詰め込むと今度は電源が足りなくなってしまいます。[*12]データセンターによっては、電源容量を増やせるオプションを提供しているところもありますが、それはそれで「10A 増やしたら 3 万円」のように月額費用に跳ね返ってきます。

ラックのスペースは決まっている、使える電源の容量も決まっている。でもそこに置けるサーバの台数を増やしたい！ そこで「物理サーバのサイズを小さくする」とは別のアプローチで生まれてきたのが**仮想サーバ**です。

物理サーバが一軒家だとすれば、仮想サーバはマンション（図 1.8）です。

[*11] アニメ映画のサマーウォーズで、冷却用の氷が部屋からなくなったことでスーパーコンピュータが熱暴走し、主人公が大事なゲームに負けてしまうシーンを思い出してください。

[*12] ぎちぎちに詰め込んでなんとか稼動していたものの、ある日データセンターで停電が起きて全台停止。電源はすぐに復旧して自動で全台いっせいに起動しようとしたが、サーバは起動時がいちばん電源を食うため、ラックのブレーカーが落ちて再度全台停止。いっせいに起動しようとしてはブレーカーが落ちて全台停止、をずっと繰り返していた・・・という怪談を聞いたことがあります。本当に怖い話ですね。

▲図 1.8　物理サーバは一軒家、仮想サーバはマンション

一軒家には 1 世帯しか住めません[*13]}が、マンションにすれば土地のサイズは同じままで 10 世帯住むことができます。1 台の物理サーバをそのまま使うのではなく、物理サーバ上に何台もの仮想サーバを作ることで、サーバラックのサイズはそのままでサーバ台数を増やすことができたのです。

このときマンションの建物にあたる物理サーバを**ホストサーバ**、101 号室や 201 号室のような各部屋にあたる仮想サーバを**ゲストサーバ**と呼んだりします。

物理的な実体があるのが物理サーバですが、その逆で手で触れる物理的な実態がないのが仮想サーバです。手で触れられるのはあくまでホスト OS のサーバであり、ゲスト OS のサーバはその中で仮想的にしか存在しないため、手で触ることはできません。

同じ広さのラックスペースに、いままでよりたくさんのサーバが詰め込めるなんて仮想サーバ素晴らしい！ と思いますが、一軒家よりマンションの方が建築コストが高いのと同じで、仮想サーバを立てるにはホストサーバとなる物理サーバのスペックも高くなければならないため、初期投資額がぐっと高くなります。

データセンターで借りるラックスペース代も高いし、物理サーバだって何十万もする、スペースを切り詰めるために仮想サーバにしたいけどホストサーバとなる物理サーバはスペックが高いのでさらに高額、となると中小企業が自社のサーバを持つのはなかなか大変です。

そこで資本力のある会社が大きなホストサーバをたくさん立てて、その上の仮想サーバ（ゲストサーバ）を他の人に貸すような仕組みが生まれました。

お分かりでしょうか？ 勘のいい方はもうお気づきのことと思いますが、ここでようやく「クラウドとは何か？」という話と繋がってきます。

[*13] 一軒家で 2 世帯同居だってあるでしょ！ みたいな突っ込みは心にしまってください。

1.3.4 オンプレミスとクラウド

昔々は、企業が「そろそろ自社のウェブサイト作りたいなぁ・・・サーバが欲しい！」と思ったら、「どのメーカーのサーバにしよう？ EHP かな？ それとも IBM かな？ DELL がいいかな？」と各社の見積もりをとって、値引き交渉をして、それでも数十万から数百万するので社内の稟議を通してやっと購入。購入してもすぐ届くわけではなく、数週間待ってやっと届きます。そして届いたら段ボールから出して、データセンターもしくは自社のサーバルームのサーバラックまで持って行って、がっちゃんこと設置したら今度は同じく自前のネットワーク機器から LAN ケーブルを繋いで電源も繋いで、OS のインストールディスクを用意したらサーバに OS をインストールして・・・以下省略しますが、要は自分でサーバを買って、何もかも自分で用意しないといけませんでした。

そのため* 初期投資のサーバ代が高い* サーバを置くのに適した場所も必要* 「欲しい！」と思ってから使い始めるまでに時間がかかるという状況でした。このようにインフラを自前で用意して、自社で所有・管理するのがいわゆる**オンプレミス**です。

これに対してクラウドは、オンプレミスと違ってサーバを買うのではなく、サービスとして「使う」だけです。

クラウドなら「自社のウェブサイト作りたいなぁ」と思ったら、サイト上で使いたいサーバのスペックを選んでぽちっとなするだけで、すぐにサーバを立てることができます。AWS なら課金も 1 秒単位の従量課金なので、たとえばサーバを 5 分使ったら 5 分ぶんの費用しかかかりません。

オンプレミスはサーバを買って使う、クラウドはサービスとして使う、ということです。でもまだちょっとわかりにくいと思うので、お店を例にしてみましょう。

クラウドのメリット

たとえば私が突然ピザ作りに目覚めて、もうインフラエンジニアなんかやってる場合じゃない！ ピザ屋を始めるんだ！[*14]と思い立ったとします。

ピザ屋さんをオープンすべく、土地を買って、そこに店舗となる建物を建てて、電気とガスと水道を通して、床板とか壁紙とか貼って・・・からやると、お金も場所も時間もたくさん必要送ですよね。しかも準備が整ってやっとオープンしたと思ったら、たった 1 か月で資金が足りなくなってお店がつぶれることになったとしても、今度は建物の取り壊しとか土地の処分とか、それはそれでやることがたくさんあります。このように全部自分で買って、自分で所有・管理するオンプレミスだと、「ちょっと気軽にピザ屋さんをやってみよう」はなかなか厳しいことが分かります。

[*14] そしたら「ピザ屋をはじめよう」という本が書けますね。

一方クラウドは、フードコートへの出店に似ています。「ピザ屋をはじめたい！ フードコートの一区画を借りてやってみよう！」という感じです。

これだと建物はもうあって、電気ガス水道ももう準備されています。フードコート内の一区画を契約して使わせてもらうだけなので、すべて自分で準備するオンプレミスと違ってすぐに始められます。しかも数か月やってみて「もうピザ焼くの飽きたー！」と思ったら、その区画を借りるのを止めるだけでいいのです。前述のとおり AWS なら 1 秒単位の従量課金なので「前月の 25 日までに契約終了を申し出る必要がある」や「フードコード内の区画を借りる契約は 1 年単位」といった制限がありません。いつでもやめられます。

とても簡単に出店できるので、私は本来やりたかった「美味しいピザを焼く」「ピザを売る」といった本業だけに注力できます。

さらに、もしピザ屋さんが大繁盛したら、フードコート内で自店の隣の区画も借りて、お店を広くすることも簡単にできるので、初めから広い区画を借りておく必要もありません。

よく「クラウドはスモールスタートに向いてる」と言われますが、その理由はまさにこういうところにあるのです。

クラウドのデメリット

但し、長い目で見るとフードコートにテナント料を払い続ける方が、土地や建物を買うより最終的には高くなるかも知れません。

なので初期投資は少なくて済むのですが、クラウドのいいところは、決して「コストが安くなる」ということではありません。たとえば万が一フードコートの入っている東館が地震で壊れても、すぐ西館に移って営業を再開できる、といった冗長性だったりします。

これ自分でやろうとしたら大変ですよね。

いつ来るか分からない地震に備えて、最初から 2 か所にお店を構えておくとなったら、相当なコストがかかります。サーバでいったら、ただ自社サイト作りたいだけなのに、東京と大阪の 2 か所にデータセンターを用意して両方に 1 台ずつサーバ置いとかないといけない、みたいな感じです。これを勝手にやってくれるクラウドすごい。

ただし、フードコートにもデメリットはあります。

何かトラブルがあって、フードコート全体がお休みになるときは、問答無用でお店もお休みになってしまいます。

つまり使っているクラウドで大規模障害が起きたら、利用者である我々に出来ることはなくて復旧までひたすら待つしかない、ということです。

AWS でも大規模障害は定期的に起きています。去年も豪雨による電源障害でサーバがダウンしたりしてました。こうした障害の際も、AWS が発表してくれることがすべてなので、原因が分かるまで自分で徹底的に調べる、ということができません。

それから通路やトイレ、駐車場といった共有スペースは他店舗と共有しているので、フードコート内で他のお店が混んでくると、駐車場が満杯でピザ屋さんに来たかったお客さんが入れなかったり、人波が自分の店の方まで押し寄せてきたりとマイナスな影響も受けます。

つまり同じクラウドを使っている他のサイトにアクセスが集中すると、回線がひっ迫して、自分のところまでつながりにくくなる、ということですね。

これがクラウドです。

オンプレミスとクラウドのおさらい

いっぱい書いたのでので、もう 1 回おさらいすると、「自社サイト作りたいなぁ」と思ったら自分でサーバを買わないといけないのがオンプレミスで、「自社サイト作りたいなぁ」と思ったらサーバを従量課金ですぐ使えるのがクラウドです。

そしてようやく最初の話に戻ると、AWS は、Amazon がやっているクラウドのことです。

AWS がなんなのか、お分かりいただけましたでしょうか？

1.3.5 パブリッククラウドとプライベートクラウド

同一セグメント内に他の利用者がいなければ、プライベートクラウドか？

1.4 AWS 以外のクラウド

ところでクラウドの便利さは分かりましたが、クラウドは Google の Google Cloud Platform、Microsoft の Azure（アジュール）、さくらインターネットのクラウドや、GMO クラウド、REX 案件で使っている IDCF クラウド、などなど AWS 以外にもたくさんあります。

その中でもなぜ「AWS」がいい！ のでしょうか？

2017 年時点、クラウド市場では AWS がシェア 34% でトップを独走中です。そのため他のクラウドと比べると、シェア No.1 なので対応できる人も多いし、何か困って調べたときに出てくる情報も多い、というのが AWS を選ぶいちばんの理由です。

あとはサポートもしっかりしてるし、儲かった分だけ投資されて、どんどん改良されていくので、細かな使い勝手もいい・・・そんなところですかね。

1.4.1 【ドリル】サンプル

問題

問題問題

- A. Route53
- B. お名前.com

答え _____

解答

正解は B です。

第 2 章

AWS とは？

2.1　AWS は最初の 1 年無料

2.1.1　無料枠に含まれるもの

2.1.2　無料の 1 年が終わる前にすべきこと

第 3 章

AWS でウェブサーバを立てよう

3.1 マネジメントコンソール

3.2 IAM

3.3 請求アラート

3.4 リージョン

3.5 CroudTrail

3.6 SecurityGroup

3.7 VPC

3.8 EC2

3.8.1 SSH の鍵認証

3.8.2 鍵の変換

3.8.3 ElasticIP

3.8.4 Bastion

第 4 章

サーバのバックアップを取っておこう

4.1 AMI

第 5 章

ELB でバランシングやサーバの台数を管理しよう

5.1 ELB

5.2 Auto Scaling

5.2.1 スケーリングに使える

5.2.2 サーバが 1 台死んでも自動で 1 台立ち上がる

第 6 章

DB サーバを立てよう

6.1 RDS

6.2 Amazon Aurora

第 7 章

ネームサーバの設定をしよう

7.1 Route53

付録 A

本当の Git

またしても何を言っているのかわからないと思いますが、「Git 用語だけでアイドルソングを作って架空のアイドルに歌わせたい」そんな気持ちで作詞をしてしまいました。〆切が厳しい時ほど才能が花開いてしまう傾向にある、と言い訳をしたいのですが、本著を書くにあたって最初に着手したのがこの付録だったということは、GitHub でコミットログを見れば一目瞭然です。それでは聞いてください。

A.1　Git - ぎゅっと言えないトゥインクル

いま何してる？　リモートのあなた
ガマンできずに フェッチして
髪型変えたの 知ったの

あなたへの気持ち 切なくて
思わずスタッシュしたまま ずっと埃つもってる

下駄箱に入れた プルリクエスト
変わっていくわたしを ちゃんとプルして抱きしめて

分かれてしまった ふたりのブランチ
コミットログ読めば あの日の気持ちも分かるはず

たくさんのライバル わたしだけをチェリーピックして
きっとわたし あなたのクローン
フォークしたあの日から ずっとあなたを見つめてる

ステージに上がったら もうコミット逃げられない
ためらわないで オリジンにプッシュ

リバートしたって 過去がなくなるわけじゃない
ただ逆の気持ちで 打ち消しただけ

別々に歩んだ ふたりの過去も
リベースすれば ひとつになれるわ

ねぇ いますぐ抱きしめて
ぎゅっと言えない トゥインクル

あとがき

2018 年 10 月
mochikoAsTech

Special Thanks:

- ブロッコリー好きな茶色い猫に捧ぐ

レビュアー

-

参考書

-

著者紹介

mochiko / @mochikoAsTech

Web 制作会社のシステムエンジニア。モバイルサイトの Web アプリケーションエンジニア、SIer とソーシャルゲームの広報を経て、2013 年よりサーバホスティングサービスの構築と運用を担当。現在は再び Web アプリケーションエンジニアとしてシステム開発に従事。「分からない気持ち」に寄り添える技術者になれるように日々奮闘中。

Hikaru Wakamatsu

表紙デザインを担当。本著の名付け親。

Shinya Nagashio

挿絵デザインを担当。

AWS をはじめよう

2018 年 10 月 8 日　技術書典 5 版 v1.0.0
著　者　mochikoAsTech
デザイン　Hikaru Wakamatsu / Shinya Nagashio
発行所　mochikoAsTech
印刷所　日光企画