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。

前作の「DNS をはじめよう」では、お名前.com でドメインを買って、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。そして本著「AWS をはじめよう」では、その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、DB 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？ そっち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、いきなり「AWS をはじめよう」から読み始めると「え、これを先にやっておかなきゃだったの？」「なにこれ、こんなの用意してない」となってさまざまな手戻りが発生します。はじめようシリーズ 1 作目となる「DNS をはじめよう」を読んでから、「AWS をはじめよう」を読み始めることを強くお勧めします。

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。

- AWS がなんなのかよく分かっていない人
- ブログやポートフォリオサイトを独自ドメインで作ってみたい人
- JavaScript や HTML や CSS なら書けるけどサーバは分からなくて苦手という人
- プログラミングの勉強がしたいけど環境構築でつまづいて嫌になってしまった人
- 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

- インフラやサーバになんとなく苦手意識のある人
- AWS、EC2、RDS、ELB、Auto Scaling、IAM、CloudTrail、Route53 などの単語に興味がある人
- 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 の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
- 物理サーバと仮想サーバの違いが説明できるようになっている
- オンプレミスとクラウド、それぞれのメリットデメリットが分か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？

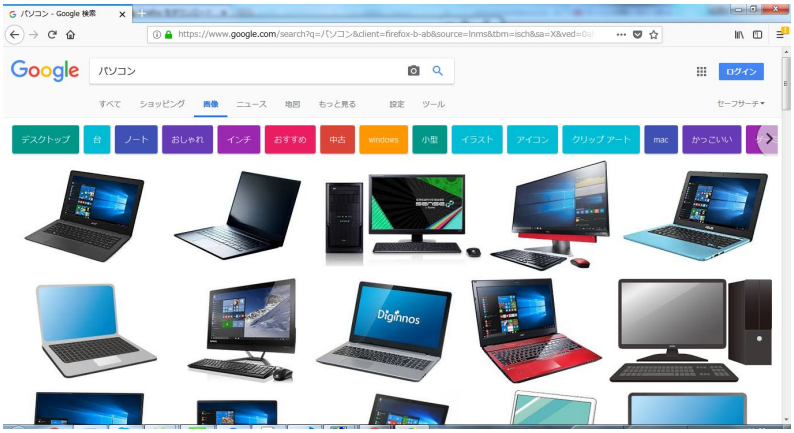
後輩から「サーバってなんですか？」という直球の質問を投げつけられたら、しっかりホームランで打ち返せますか？

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]

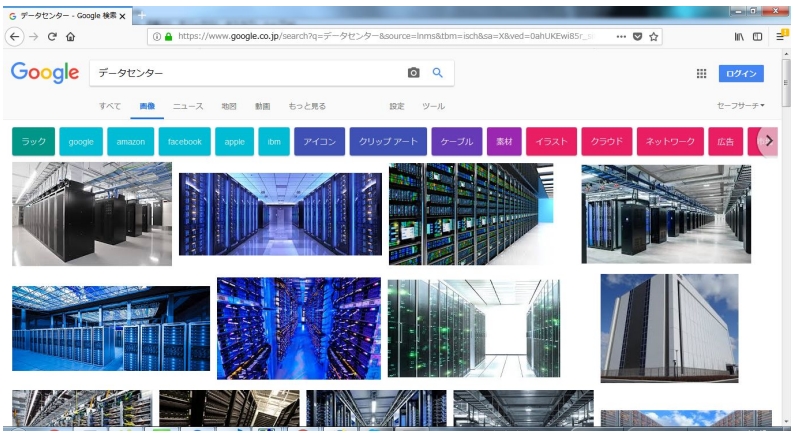
カラオケをするには防音や音響設備の整ったカラオケルームが適しているように、サー

[*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}）



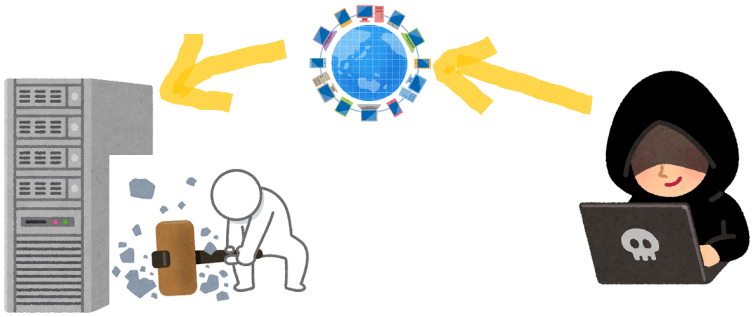
▲図 1.6　データセンターはサーバのための設備が整っている

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[*14]。

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します。スペースを切り詰めるために仮想サーバにしたいと思っても、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・・・となると中小企業やスタートアップ企業が自社で物理サーバや仮想サーバを所有・管理するのはなかなか大変なことです。

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。

[*14] ホスト OS、ゲスト OS という呼び方をすることもあります。

1.4 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・・・だからサーバが必要だ！」と思ったら、「サーバを買う」という選択肢しかありませんでした。

サーバを買うと言っても、お店に行ってぱっと買って持ち帰れる、という訳ではありません。

「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り、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。そして届いたらサーバを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にあるサーバラックのところまで持って行って、がっちゃんこと設置。設置できたら今度は同じく自前のネットワーク機器から LAN ケーブルを繋ぎ、電源も繋ぎます。そして 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「ただ自社のウェブサイトが作りたいだけなのに、サーバを用意するのがすごく大変だった」ということです。

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ないため、* 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・・・だからサーバが必要だ！」と思ったら、ブラウザを開いて、クラウド事業者の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。しかも 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。こんな簡単にサーバを使い始めたりやめたりできるのは、クラウド事業者が物理サーバそのものを提供しているのではなく、大きなホストサーバをたくさん用意しておいて、その上に立てた仮想サーバ（ゲストサーバ）を提供しているからです。

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、ということですね。でもまだちょっとわかりにくいと思うので、お店を例にオンプレミスとクラウドの違いを確認してみましょう。

1.4.1 クラウドのメリットとデメリット

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！[*15]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、床板や壁紙を貼って・・・からやると、お金も場所も時間もたくさん必要です。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や土地の処分など、止めるときは止めるとき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。

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！ だから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用意されています。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わー！」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 なら使い始めるときの初期費用もなく 1 秒単位の従量課金なので「ピザ屋さんもうやめたいけど、この先 30 年のローン支払いが残ってるからやめるにやめられない・・」ということもありません。「前月の 25 日までに契約終了を申し出る必要がある」といった制限すらないので、本当にいつでもやめられます。

クラウドならとても簡単に出店できる（つまりサーバを用意できる）ので、私は本来やりたかった「美味しいピザを焼いて売る」（ウェブサイトを作って自社を宣伝する）という本業に注力できます。

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。つまり、ウェブサイトへのアクセスが増えてきてサーバのスペックが足りなくなったら、後から増強したり好きなだけサーバの台数を増やしたりもできるので、最初から高スペックなサーバを借りておく必要がありません。

クラウドなら初期投資額が少なく、すぐに始められて、すぐにやめられる。よく「クラウドはスモールスタートに向いている」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。

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

[*15] そしたら「AWS をはじめよう」の続編として「ピザ屋をはじめよう」という本が書けますね。

よりも最終的には高くなるかも知れません。前述のとおり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。実際、AWS は他社の共有レンタルサーバや VPS[*16] と比べると高額です。

クラウドのよいところは、たとえばショッピングセンター内でフードコートが入っている南館が地震で崩れてしまっても、すぐに北館に移ってピザ屋の営業を再開できる、といった冗長性です。

この冗長性を自力で確保しようとしたら大変です。ピザ屋さんを常に営業し続けておくために、いつ来るか分からない地震に備えて最初から予備の店舗も確保しておかなければならないとしたら、相当なコストがかかります。オンプレミスのサーバなら、ただ自社サイトを作りたいだけなのに、品川と渋谷の 2 か所でデータセンターを借りて両方に 1 台ずつホストサーバを用意しておき、品川のデータセンターが使えなくなったらその上で動いていたゲストサーバを渋谷のホストサーバに移動させる、というような大仰な話です。これを勝手にやってくれるクラウドはすごいですよね。

ここまでクラウドの良さを色々お話ししてきましたが、もちろんデメリットもあります。

もし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、問答無用でピザ屋さん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、一利用者である私たちにできることはなく復旧までひたすら待つしかない、ということです。AWS でも広範囲にわたる障害は定期的に起きています。たとえば 2016 年には豪雨による電源障害でサーバに接続できなくなる事象が発生[*17]しました。こうした障害の際も、AWS が発表してくれる内容がすべてですので、原因が分かるまで自分で徹底的に調べる、あるいは自力で何とかする、ということはできません。

また通路やトイレ、駐車場といった共有スペースはフードコート内の他店舗（ドーナツ屋さんやラーメン屋さんなど）と共有していますので、フードコート内で他のお店が混んでくると、駐車場が満杯になって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[*18]を使っているウェブサイトにアクセスが集中すると、たとえば回線がひっ迫したりして自分のサイトまで繋がりにくくなる、ということです。

1.4.2 AWS は Amazon がやっているクラウド

たくさんお話ししてきたので、一度おさらいをしましょう。

[*16] Virtual Private Server の略。先ほど出てきた仮想サーバのことだと思ってください。

[*17] Amazon クラウドのシドニーリージョン、豪雨による電源障害で EC2 などに一部障害。現在は復旧 － Publickey `https://www.publickey1.jp/blog/16/amazonec2.html`

[*18] 具体的には、同じホストサーバを使っている他のゲストサーバ上のサイト。あるいは同じインターネット回線を使っている他のサーバ上のサイト、ということです。

企業が「自社のウェブサイト作りたいなぁ・・・だからサーバが必要だ！」と思ったとき、自分でサーバを買って自分で管理しなければいけないのがオンプレミスです。そして「自社のウェブサイト作りたいなぁ・・・だからサーバが必要だ！」と思ったとき、従量課金ですぐ使えて、性能や台数の増減も簡単にでき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 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 Amazon がやっているクラウドです。

AWS がなんなのか、お分かりいただけましたでしょうか？

1.4.3 パブリッククラウドとプライベートクラウド

ところでパブリッククラウドやプライベートクラウド、という言葉は聞いたことがありますか？

AWS のようなクラウドは、パブリッククラウドと呼ばれることもあります。みんなでホストサーバという資源（リソース）を共有して使うので、「公共の」という意味の「パブリック」が付いています。

クラウドが少しずつ使われるようになった頃に「クラウドって便利そうだけど、みんなで共有するのってちょっと抵抗あるな・・・」と思った人たちを安心させるため、「プライベートクラウド」という言葉が生まれました。プライベートクラウドとはいったい何なのでしょう？

たとえばオンプレミスの環境で「高スペックな物理サーバを買ってホストサーバにして、その上でゲストサーバ（仮想サーバ）を立てられるようにしたぞ！ ホストサーバのスペックが足りる限りという制限はあるものの、好きなときに好きなだけゲストサーバを立てたり、増強したりできるのでこれはクラウド！ プライベートなクラウドだ！」と言うこともできます。また「クラウド事業者が提供しているホストサーバを 1 台まるまる占有する契約をしたぞ！ 自社で物理サーバを所有している訳ではないのでこれはクラウドだ。しかも他の人はこのホストサーバ上のゲストサーバを使えないから、プライベートなクラウドだ！」と言うこともできます。定義は曖昧なのですが、このように**みんなで共有せず、自社だけで専有できるクラウドをプライベートクラウドと呼ぶ**ようです。

このプライベートクラウドだと「初期投資額が少ない」「サーバの性能や台数を後から好きなだけ増強できる」といった、クラウド本来のメリットが享受できないように思えますが、これもクラウドなのでしょうか？

このようにクラウドという言葉はとても曖昧です。結局「クラウド」という言葉の定義がはっきりしていないため、その人が言っている「クラウド」という言葉がなにを指して

いるのかは、よくよく聞いてみないと分からない、ということです。[*19]

1.4.4 AWS 以外のクラウド

ところでクラウドは AWS 以外にも Google の Google Cloud Platform[*20]、Microsoft の Azure（アジュール）[*21]、その他にも国内クラウドとしてさくらインターネットがやっているさくらのクラウド[*22]、GMO クラウド[*23]などたくさんあります。

その中でもなぜ AWS なのでしょう？

2018 年時点、クラウド市場では AWS がシェア 33% でトップを独走中[*24]です。そのため他のクラウドと比べて、使ったことがあって対応可能なエンジニアも多いし、何か困ったときに調べて出てくる情報も多い、というのが、私が AWS を選ぶいちばんの理由です。それ以外だと、利益が出た分だけどんどん投資されてサービスが改良されていくため、細かな使い勝手がどんどん良くなっていく[*25]ところもポイントです。

クラウドを選ぶ理由、その中でも AWS を選ぶ理由というのは、普遍的な何かがあるわけではなく、本来は使う人やその上で動かすサイトによって異なるはずです。あなたが動かしたいサイトによっては、AWS ではなく他の VPS の方がいいケースだってもちろんあるはずです。これから使ってみて、あなた自身が AWS の良いところを発見できたらいいですね。

[*19] 実際、オンプレミス環境にある仮想サーバをクラウドサーバと呼んでいるケースも多々あります。

[*20] `https://cloud.google.com/`

[*21] `https://azure.microsoft.com/ja-jp/`

[*22] `https://cloud.sakura.ad.jp/`

[*23] `https://www.gmocloud.com/`

[*24] 2018 年第 1 四半期、クラウドインフラ市場で AWS のシェアは揺るがず 33 %前後、マイクロソフト、Google が追撃、IBM は苦戦中。Synergy Research − Publickey `https://www.publickey1.jp/blog/18/20181aws33googleibmsynergy_research.html`

[*25] 画面や機能もどんどん変わっていくので、この後出てくる設定画面も皆さんが手を動かしてやってみる頃にはキャプチャと違っているかも知れません。AWS のいいところでもあり、マニュアルなどを作って説明する側にとってはつらいところでもあります。

第 2 章

AWS にログインしてみよう

この章では AWS の管理画面にログインして、最初に行うべき設定をします。

2.1 AWS は最初の 1 年無料

AWS を初めて使用する場合、AWS アカウントを作成してから 1 年間は利用料が無料となります。但し、無料利用枠の範囲は決まっており、何をどれだけ使っても無料という訳ではありません。全部無料だと思ってサーバをバカスカ立てると、あとでクレジットカードにしっかり請求が来ますので注意してください。

どのサービスをどれくらい無料で使えるのか？ は「AWS 無料利用枠の詳細（`https://aws.amazon.com/jp/free/`」（図 2.1）で確認してください。

▲図 2.1　AWS 無料利用枠の詳細

なお本著で使用する AWS のサービスは、基本的に無料利用枠の範囲内に収まるようにしています。但し、Route53 というネームサーバのサービスについては無料利用枠の対象外であるため、毎月 50 セント程度かかりますのでその点はご留意ください。

うっかり請求が来ても筆者が代わりに支払うことはできませんので、この後「利用金額が〇円を超えたらメールで知らせる」という請求アラートの設定をしておきましょう。

2.2 AWS のアカウント作成

AWS を使うには、先ず AWS アカウントを作成する必要があります。

AWS アカウントの作成は「DNS をはじめよう」の「第 3 章 AWS のネームサーバ（Route53）を使ってみよう」で済ませていますので、本著でもその AWS アカウントを引き続き使用していきます。

まだ AWS アカウント作っていない！ という人は、先に「DNS をはじめよう」で、* ドメインを買う* AWS のアカウントを作る* ネームサーバとして AWS の Route53 を使うという 3 つのステップを踏んでから、この先へ進むようにしてください。

2.3 マネジメントコンソールにログイン

それでは https://aws.amazon.com/ を開いて、黄色の「コンソールへサインイン」ボタンを押してください。

E メールアドレスとパスワードを入力して、「サインイン」ボタンを押します。

マネジメントコンソールにログインできました。皆さんできましたか？

このマネジメントコンソールが、AWS の管理画面となります。サーバ立てたりも全部ここの画面からできます。

2.4 リージョンの変更

最初に、マネジメントコンソールの右上に表示されているリージョンが「東京」であることを確認してください。

もし東京以外のリージョンだったら、クリックして「アジアパシフィック (東京)」を選択してください。選択後、右上が「東京」に変わったらリージョンの変更は完了です。

AWS はバージニア、カナダ、ロンドン、シンガポール、東京など世界の各地域にデータセンターを持っていて、サーバはそのデータセンターの中にあります。

このリージョンとはその中でどこを使うか？ を指定するものです。物理的に遠ければ、それだけ通信にも時間がかかって応答時間も遅くなるので、基本はこの「東京」リージョンを選びましょう。但しサービスによって、まだ東京リージョンが使えないものがありますので、その場合は「米国東部 (バージニア北部)」を選びましょう。

今回は詳しく説明しませんが、リージョンという地域ごとの区分の下に、さらにアベイラビリティゾーンという区分があります。アベイラビリティゾーンの場所は公開されていませんが、たとえば東京リージョンの下に、新宿アベイラビリティゾーンと渋谷アベイラビリティゾーンがある、というようなイメージです。

この後、EC2 でサーバを立てる際、誤って東京以外のリージョンで立てないように注意して下さい。

2.5 IAM

2.6 請求アラート

2.7 リージョン

2.8 CroudTrail

第 3 章

AWS でウェブサーバを立てよう

3.1 SecurityGroup

3.2 VPC

3.3 EC2

3.3.1 SSH の鍵認証

3.3.2 鍵の変換

3.3.3 ElasticIP

3.3.4 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

第 8 章

AWS をやめたくなったらすること

8.1　無料の 1 年が終わる前にすべきこと

8.1.1　【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。

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。「DNS をはじめよう」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