

AWS でインフラことはじめ

maki 著

2019-09-22 版 ひよこ開発室出版部 発行

はじめに

こんにちは。著者は情報系の知識ゼロで IT 企業に入社しました。入社してからは情報系の資格をとったりしましたが、机上の勉強だけではいまいちピンと来ていませんでした。また、仕事ではオンプレミスかつ大規模なシステムだったため、何もないところから自分が構築を担当するようなことはなかったですし、知識的にも断片的で理解が浅いと感じていました。

そんな時、自分でクラウドの AWS を利用しながらイチからシステムを構築する機会がありました。オンプレミスとクラウドの違いは後ほど説明しますが、コスト的にも、技術的にも気軽にシステムを構築できるところがクラウドのよいところだと思います。実際に自分でネットワークやサーバーの設定をすることで、「知っている」から「自分で構築できる」と思えるようになりました。

本著では、私のように「なんとなく知っているけど自分でインフラを構築したことはない」という方が、AWS で実際に構築することにより「自分で構築できる」そして知識が「身につく」ことを実感していただけると幸いです。

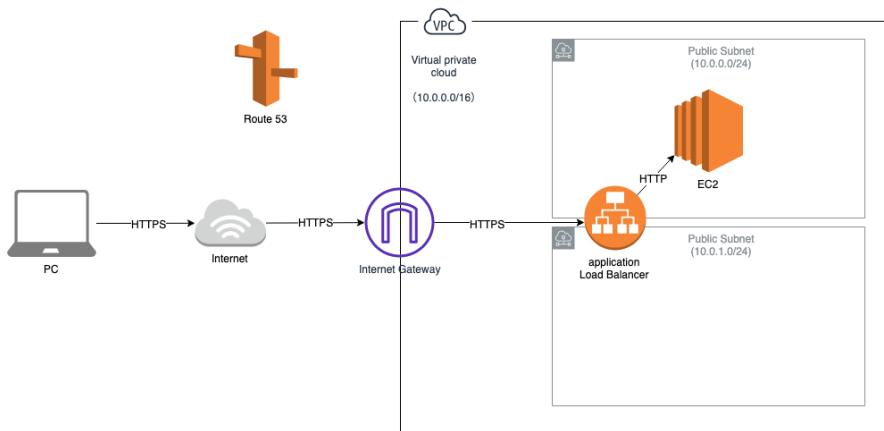
本著は、こんな人に向けて書かれています

- これからシステムについて勉強しようとしている人
- AWS を触ってみたい人
- 自分でインフラ構築したことがない人

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

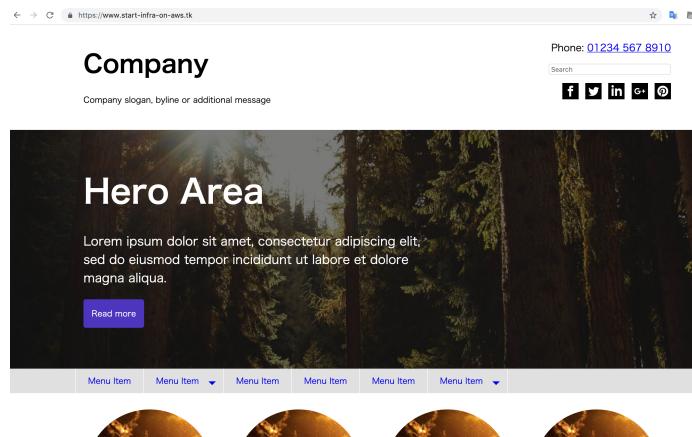
- グローバル IP アドレス、ドメイン、SSH、TLS などのシステムを作る上で必要な基本的な仕組みについて説明でき、設定できる
- VPC、サブネット、EC2 インスタンス、セキュリティグループ、ロードバランサーなどの AWS サービスを使用できる

- 次のようなアーキテクチャを自分で作成することができる



▲図 1 作成するシステムアーキテクチャ

- 次のような HP を作成することができる



▲図 2 完成した HP

免責事項

本書に記載された内容は、情報の提供のみを目的としています。したがって、本書を用いた開発、製作、運用は、必ずご自身の責任と判断によって行ってください。また、技術的な仕組みを説明するにあたってはイメージのしやすさを優先し、説明が不足している部分があるかと思いますが、ご容赦ください。これらの情報による開発、製作、運用の結果について、著者はいかなる責任も負いません。また、本書を用いた環境構築により発生した AWS の利用料金やその他の料金について、著者は負担致しません。

目次

はじめに	2
本著は、こんな人に向けて書かれています	2
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	2
免責事項	4
第 1 章 システムの構築を始める前に	8
1.1 システムをつくる要素	8
1.2 インフラの範囲	9
1.3 システムはどこにあるの？	9
1.4 オンプレミスとクラウドの違い	9
1.4.1 オンプレミスとは？	10
1.4.2 クラウドとは？	10
第 2 章 AWS を使い始める	12
2.1 アカウントを作る	12
2.1.1 アカウント作成に必要なもの	12
2.1.2 アカウントの作成手順	12
2.2 AWS マネジメントコンソールへのログイン	12
2.3 MFA の設定	14
2.3.1 MFA（多要素認証）とは？	14
2.3.2 MFA を設定する	15
2.4 請求アラートの設定	20
2.5 IAM ユーザーの作成	30
2.6 アカウント ID にエイリアスを設定する	35
第 3 章 ネットワークを作成しよう	38
3.1 ネットワークを作成する前に	38

3.1.1	IP アドレスってどんなもの？	38
3.1.2	ネットワーク部とホスト部	39
3.1.3	ネットワークのサブネット化	39
3.1.4	グローバル IP アドレスとプライベート IP アドレス	40
3.2	VPC を作成する	41
3.3	IGW を作成する	44
3.4	IGW を VPC にアタッチする	46
3.5	サブネットを作成する	48
3.5.1	パブリックサブネットとプライベートサブネット	48
3.5.2	パブリックサブネットを作成する	49
3.5.3	アベイラビリティゾーンとは	49
3.6	ルートテーブルの設定	52
第 4 章	EC2 インスタンスを作成しよう	57
4.1	EC2 インスタンスを作成する	57
4.1.1	プロトコルとは？	61
4.1.2	ポート番号とは？	62
4.1.3	セキュリティグループの設定	62
第 5 章	ロードバランサーを作成しよう	66
5.1	ロードバランサーの作成	66
5.2	EC2 インスタンスのセキュリティグループの編集	72
5.3	ターゲットの状態確認	74
第 6 章	SSH で EC2 インスタンスにログインしよう	76
6.1	SSH とは	76
6.1.1	パスワード認証方式	76
6.1.2	公開鍵認証方式	76
6.2	SSH 接続する	77
6.3	MAC の場合	78
6.4	Windows の場合	84
第 7 章	HP を立ち上げよう	92
7.1	python で HTTP サーバーを起動しよう	92
7.2	サンプルのテンプレートファイルを使おう	97

第 8 章	ドメインでアクセスできるようにしよう	102
8.1	ドメインってなに？	102
8.1.1	分野別トップレベルドメイン (gTLD)	102
8.1.2	国コードトップレベルドメイン (ccTLD)	103
8.1.3	サブドメイン	103
8.1.4	なぜドメインにアクセスするとサーバーにアクセスできるの？ .	104
8.2	ドメインを取得する	104
8.3	Route53 にドメインを登録してロードバランサーに紐付けよう	110
8.4	ドメインのネームサーバーを変更する	113
第 9 章	HTTPS でアクセスできるようにしよう	117
9.1	HTTP と HTTPS の違い	117
9.2	SSL/TLS 通信の仕組み	118
9.2.1	暗号スイートの合意	118
9.2.2	デジタル証明書と公開鍵の提示	118
9.2.3	共通鍵の元データ交換・共通鍵の生成	119
9.2.4	暗号化通信開始	119
9.3	デジタル証明書を発行しよう	120
9.4	ロードバランサーで HTTPS の設定をしよう	130

第1章

システムの構築を始める前に

1.1 システムをつくる要素

いきなりですが、システムってなに？と聞かれたら、どんなものを想像しますか。基本的なシステムは、次の要素で構成されています。

- アプリケーション
- ミドルウェア
- OS
- ハードウェア
- ネットワーク



▲図 1.1 基本的なシステム要素

1.2 インフラの範囲

では、システムをつくる要素の中でインフラと呼ばれるのはどこの部分でしょうか？

実は、アプリケーション以外のすべてです。インフラとは、アプリケーションを動かすための基盤/土台という意味なのです。

1.3 システムはどこにあるの？

システムは、データセンターにあります。（小さい規模のシステムの場合は、お家や職場に置いている場合もあります。）どんな会社のデータセンターでも、住所は基本的に秘密です。悪い人に攻撃されたりしたら困るからです。

データセンターはシステム専用のマンションのようなイメージです。多くのシステムを動かすための電源装置や、システムが暑くなりすぎないように冷やす冷却装置などがあり、ラック（棚）に、サーバーや NW 機器が収納されています。



▲図 1.2 データセンター

1.4 オンプレミスとクラウドの違い

システムの種類として、オンプレミスとクラウドがあります。これは、システムをつくる時に、自前ですべて用意するかどうかの違いです。家にたとえると、さら地から一戸建てを立てるのか、マンションの一室を借りるかの違いに似ています。

1.4.1 オンプレミスとは？

クラウドが登場するまではこちらの方法がスタンダードでした。データセンターの契約から、HW や NW 機器の購入・搬入・設置、OS のインストール・・すべて自分たちでやります。すべて自分で購入するため、初期投資が大きな金額になりますが、その代わり物理的に自分専用のシステムがつくれます。



▲図 1.3 オンプレミスのイメージ

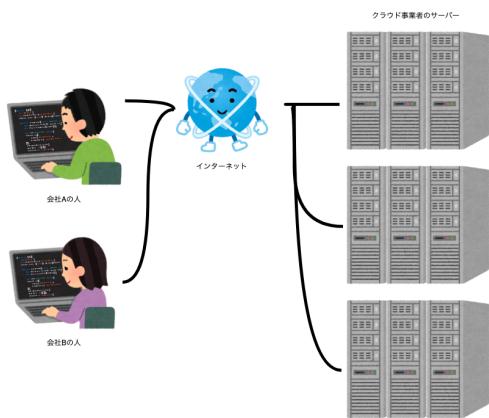
1.4.2 クラウドとは？

クラウド事業者がデータセンターの契約から、HW や NW 機器の購入・搬入・設置、OS のインストールをやってくれます。主なクラウド事業者は、Amazon の AWS (Amazon Web Services)、Microsoft の Microsoft Azure、Google の GCP (Google Cloud Platform) が有名です。それぞれのクラウド事業者によって、提供しているサービス内容には特色がありますので、自分がやりたいことにあったクラウド事業者を選ぶのがよいでしょう。

クラウド利用者は、すでに準備されたサーバーや NW を使いたい分だけ使う代わりに、クラウド事業者に利用料を支払います。最初はスペックの低いシステムを作って、あとでもっとスペックをよくする、ということも可能ですので、手軽に始められます。

クラウドの場合はインターネット経由で自分のサーバーにアクセスします。このため、公開する必要のない場所は自分しかアクセスできないようにするなど、セキュリティの設定が大変重要になります。

1.4 オンプレミスとクラウドの違い



▲図 1.4 クラウドのイメージ

第2章

AWS を使い始める

2.1 アカウントを作る

2.1.1 アカウント作成に必要なもの

- メールアドレス
- クレジットカード

AWSには1年無料のサービスがありますので、基本的にはお金がかかりません。ただし、無料サービス対象外のサービスも使用しますので、毎月数ドル費用が発生します。これについては、後ほどご紹介する請求アラートの設定をしておきましょう。

2.1.2 アカウントの作成手順

公式サイトに分かりやすく説明されておりますので、こちらの手順にしたがってアカウントを作成します。

<https://aws.amazon.com/jp/register-flow/>

無料のベーシックプランを選びましょう。また、アカウントのパスワードは忘れないようにメモしておきましょう。

2.2 AWS マネジメントコンソールへのログイン

アカウントが作成できたら、マネジメントコンソールにログインしましょう。

<https://console.aws.amazon.com/console/home>

2.2 AWS マネジメントコンソールへのログイン



▲図 2.1 AWS マネジメントコンソールログイン画面

ログインすると、AWS マネジメントコンソールのホーム画面が表示されます。右上のアカウント名の右側を見ると、「オハイオ」という文字があります。クリックしてみましょう。

世界の都市名がずらっと出てきました。

これをリージョンと呼びます。地理的に離れた領域のことで、世界に 20箇所（2018 年 12 月現在）あります。各リージョンに AWS のデータセンターがあり、AWS のサーバーが稼働しています。どこでも好きなリージョンを選んで AWS のサービスを利用することができますが、日本で使うシステムをオハイオなどの距離的に遠いリージョンで作成すると、サーバーまでの距離が長くなるため遅延（レイテンシ）が発生しますので、基本的には東京リージョンを選択します。

ただし、東京リージョンではまだ使えないサービスを使いたい場合や、レスポンスが多少遅くても問題ないシステムであれば、コスト節約のために海外のリージョンを利用することもあります。

今回は「東京リージョン」を利用しましょう。



▲図 2.2 リージョンの変更

今の URL^{*1}をお気に入りに登録しておくと、次回からは東京リージョンを選択した状態でログインできます。

2.3 MFA の設定

2.3.1 MFA（多要素認証）とは？

MFA（Multi-Factor Authentication）とは日本語で「多要素認証」のことです。多要素認証とは、あなたのアカウントに不正ログインされないように次の要素のうち 2つ以上を組み合わせて認証することです。

- 知識情報（あなたが知っているもの）
 - パスワード、秘密の質問
- 所持情報（あなたが持っているもの）
 - IC カード、スマートフォン
- 生体情報（あなた自身）
 - 指紋、静脈

今までは、ユーザー名とパスワードだけで AWS のコンソールにログインできてしま

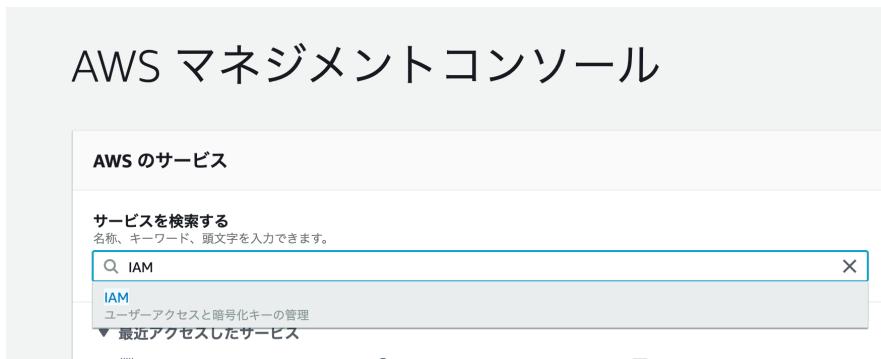
^{*1} <https://ap-northeast-1.console.aws.amazon.com/console/home?region=ap-northeast-1#>

またため、多要素認証にはなっておらず、ちょっと心配です。

AWS では所持情報を使った多要素認証の設定ができます。多要素認証専用のデバイスを使用することができますが、スマートフォンのアプリを使うのが一番楽なので、これから設定していきましょう。

2.3.2 MFA を設定する

MFA の設定は、「IAM」というサービスで設定します。AWS マネジメントコンソールにログイン後、「IAM」を検索しましょう。



▲図 2.3 MFA を選択

IAM のダッシュボードで「ルートアカウントの MFA を有効化」をクリックして開き、「MFA の管理」をクリックします。

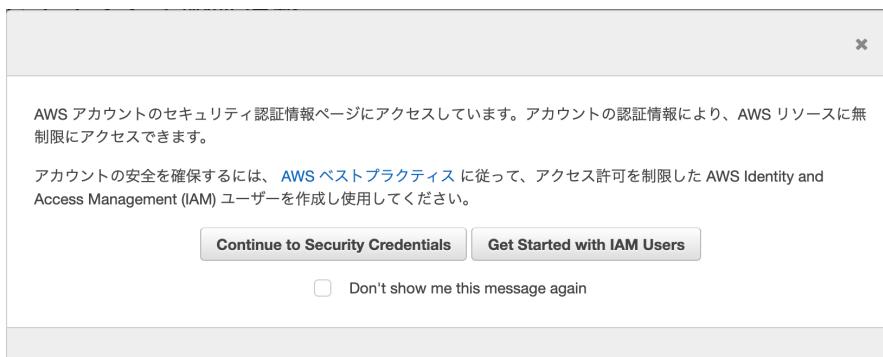
第2章 AWS を使い始める

The screenshot shows the AWS IAM console with the following details:

- Identity and Access Management (IAM)** menu:
 - AWS Account (153190274462)
 - ダッシュボード**
 - グループ
 - ユーザー
 - ロール
 - ポリシー
 - ID プロバイダー
 - アカウント設定
 - 認証情報レポート
 - IAM の検索**
- Identity and Access Management へようこそ** section:
 - IAM ユーザーのサインインリンク: <https://153190274462.sigin.aws.amazon.com/console>
 - | カスタマイズ
- IAM リソース** section:
 - ユーザー: 0
 - グループ: 0
 - ID プロバイダー: 0
- セキュリティステータス** section:
 - ① クリックして閉く
 - ルートアクセスキーの削除
 - ② クリック (highlighted with a red arrow)
 - ルートアカウントの MFA を有効化
 - アカウントのセキュリティを保護するには、AWS ルートアカウントで多要素認証 (MFA) を有効化して、別の保護レイヤーを追加します。詳細はこちら
 - MFA の管理 (highlighted with a red arrow)
 - 個々の IAM ユーザーの作成
 - グループを使用したアクセス許可の割り当て
 - IAM パスワードポリシーの適用
- 注目の機能** section:
 - Introduction to AWS IAM

▲図 2.4 「MFA の管理」をクリック

すると、こんな画面が表示されます。



▲図 2.5 MFA 設定時の確認画面

今ログインしているユーザーはルートアカウントだから最強の権限を持っているのだけど、必要な権限だけもったユーザーを作った方がいいんじゃない？ という確認です。

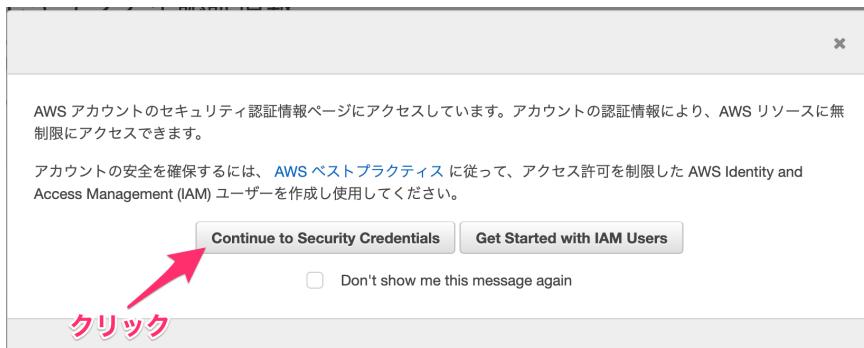
あなたが今ログインに使っているのは一番最初に作成したルートアカウントですが、他にもユーザーを作成することができます。

ルートアカウントは何でもできてしまうので、使うときにも慎重に扱わなければなりません。このため、AWS ではルートアカウントは極力使わず、適切な権限をつけた IAM ユーザー^{*2}を作成して使うことを推奨しています。

^{*2} ルートアカウント以外を IAM ユーザーと言い、ユーザーごとに割り当てる権限を変えることによって、

2.3 MFA の設定

今はそのままルートアカウントで操作しますので、「Continue to Security Credentials」をクリックしましょう。



▲図 2.6 「Continue to Security Credentials」をクリック

MFA デバイスは「仮想 MFA デバイス」を選択し、「続行」をクリックします。

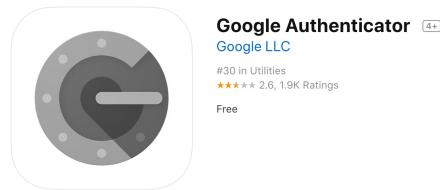


▲図 2.7 「仮想 MFA デバイス」を選択

仮想デバイスの設定画面が出てきますので、手順に従い、設定していきましょう。

スマートフォンに Google の認証アプリ「Google Authenticator」をインストールしてください。

Aさんは新しいサーバーを作ることができるけどBさんにはできない、とユーザーごとにできることを制限することができます。



▲図 2.8 Google Authenticator アプリ

仮想デバイスの設定画面の「QR コードの表示」をクリックします。

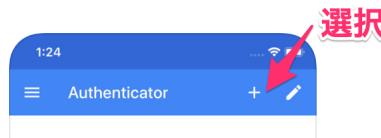


▲図 2.9 「QR コードの表示」をクリック

QR コードが表示されるので、スマートフォンにインストールした Google Authenticator アプリを起動し、右上の「+」ボタンを押して「バーコードをスキャン」を選択し

ます。

すると、カメラが起動するので先ほど表示した QR コードを読み取ります。



▲図 2.10 Google Authenticator アプリで QR コードをスキャン

Google Authenticator アプリに数字が表示されるので、MFA コード 1 に入力し、番号が変わることを待ちます。番号が変わったら、MFA コード 2 に入力して「MFA の割り当て」をクリックします。

もし入力が終わる前に番号が変わってしまった場合、MFA コード 1 から入力し直しましょう。連続して表示された数字を MFA コード 1 と MFA コード 2 に入力する必要があります。



▲図 2.11 QR コードを読み取り、MFA コードを入力する

設定が完了しました。



▲図 2.12 設定が完了

一度、AWS マネジメントコンソールからログアウトして、ログインし直してみましょう。

ログアウトは、一番上のアカウント名をクリックするとメニューが出てくるので「サインアウト」をクリックします。^{*3}



▲図 2.13 ログアウトする

再度ログインしようとすると MFA を求められるので、Google Authenticator アプリで表示されている数字を入力してログインしましょう。

あなたのスマートフォンを使用した多要素認証の設定が完了しましたね。

2.4 請求アラートの設定

クラウドでは、サービスを使った分だけ請求されます。勉強のために色々つくったまま放置していると、いつの間にか何十万円の請求が・・などとならないように、請求アラート

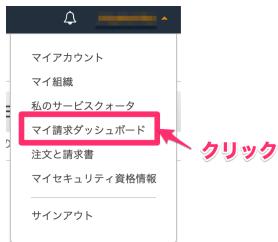
^{*3} 本著では、分かりやすさのためにログイン・ログアウトを使用していますが AWS ではサインイン・サインアウトが使われています。意味はログイン・ログアウトと同じです。

2.4 請求アラートの設定

トを設定しましょう。

ログイン後のアカウント名をクリックし、メニューを表示します。

「マイアカウント請求ダッシュボード」をクリックします。



▲図 2.14 マイ請求ダッシュボードをクリック

請求情報とコスト管理ダッシュボードを表示したら、左の「Billing の設定」をクリックします。



請求情報とコスト管理ダッシュボード

Billing Cost Explorer

AWS 請求とコスト管理の開始方法

- AWS の予算 を使用して、コストと使用状況を管理します
- Cost Explorer を通じてコスト要因と使用傾向を視覚化します
- Athena の組合により、コストと使用状況レポートを使用してコストを詳しく分析する
- 詳細: AWS の最新情報ウェーブページをご覧
リザーフィングインスタンス (RI) がありますか?
- RI 使用率レポート、カラーリッジレポート、および RI 購入レポートに、コストエクスプローラー を通じてアクセスします。

利用料の概要

過去 1 か月間の残高: 2019年9月

今月の初めから今日までのサービス別利用料

次のグラフは各サービスの利用料比率を表しています。

EC2
Route53
CloudWatch
DataTransfer

▲図 2.15 Billing の設定をクリック

設定画面で、次のすべてにチェックを入れます。

- 電子メールで PDF 版請求書を受け取る
- 無料利用枠の使用のアラートの受信（アカウント登録時のメールアドレス以外のメールアドレスで受け取りたい場合は、メールアドレスを入力）
- 請求アラートを受け取る

「設定の保存」をクリックして保存したら、「請求アラートを受け取る」の項目説明にあ

第2章 AWS を使い始める

る「請求アラートを管理する」リンクをクリックします。



▲図 2.16 「請求アラートを管理する」リンクをクリック

CloudWatch のダッシュボードが表示されましたね。CloudWatch は AWS のサービスで、システムの状態やログを監視したり、アラームが設定できたりします。

今回は、AWS からの請求金額が想定以上の金額になった時に、メールで通知してくれるようなアラームを作成していきましょう。

左側のメニューで「請求」をクリックします。

2.4 請求アラートの設定



▲図 2.17 請求をクリック

「アラームの作成」をクリックします。



▲図 2.18 アラームの作成をクリック

「メトリクスの選択」をクリックします。



▲図 2.19 メトリクスの選択をクリック

いくつかのメトリクスが表示されますが、「請求」を選択します。



▲図 2.20 請求をクリック

「概算合計請求額」を選択します。

2.4 請求アラートの設定



▲図 2.21 概算合計請求額をクリック

「USD」を選択して「メトリクスの選択」をクリックします。



▲図 2.22 メトリクスの選択をクリック

「メトリクスと条件の指定画面」が表示されます。下の方を見るとしきい値^{*4}の設定が

*4 境目となる値で、限界値のようなものです。

できます。本著の内容のみ AWS を使用するのであれば、2 ドルで十分ですので「2」と入力します。

「次へ」をクリックします。

条件

しきい値の種類

静的
値をしきい値として使用

異常検出
バンドをしきい値として使用

EstimatedCharges が次の時...

アラーム条件を定義

より大きい
 $> \text{しきい値}$

以上
 $\geq \text{しきい値}$

以下
 $\leq \text{しきい値}$

より低い
 $< \text{しきい値}$

...よりも
しきい値を定義します。

2 USD

数字である必要があります

①2ドルに設定

②クリック

キャンセル 次へ

▲図 2.23 しきい値の設定

アクションの設定では、設定したしきい値（2 ドル）よりも請求金額が大きくなった時に、メールで通知するための設定をします。

「新しいトピックの作成」を選択し、トピック名に「aws-alarm」、自分のメールアドレスを入力し、「トピックの作成」をクリックします。メールは自分が気づきやすいメールアドレスを設定しましょう。

設定したら、「次へ」をクリックします。

2.4 請求アラートの設定



▲図 2.24 トピックの作成

説明の追加画面で、アラーム名に「aws-alarm」、説明に「start-infra-on-aws」と入力したら「次へ」をクリックします。



▲図 2.25 アラームの説明追加

プレビューと作成画面で内容を確認したら、「アラームの作成」をクリックします。

第2章 AWS を使い始める

2

▶ その他の設定

ステップ 2: アクションの設定

Edit

アクション

通知

アラーム状態のとき、「aws-alarm」に通知を送信します

ステップ 3: 説明を追加

Edit

名前と説明

名前

aws-alarm

説明

start-infra-on-aws

キャンセル

戻る

アラームの作成

クリック

▲図 2.26 アラームの作成

アラーム画面で、「一部サブスクリプションが確認待ちの状態です」というバーが出ています。設定したメールアドレスにメールがちゃんと届くのか確認してね、という意味です。

横にある、「SNS のサブスクリプションを表示」をクリックします。

2.4 請求アラートの設定



▲図227 SNSのサブスクリプションを表す

保留中の確認となっているサブスクリプションを選択し、「リクエストの確認」をクリックします。



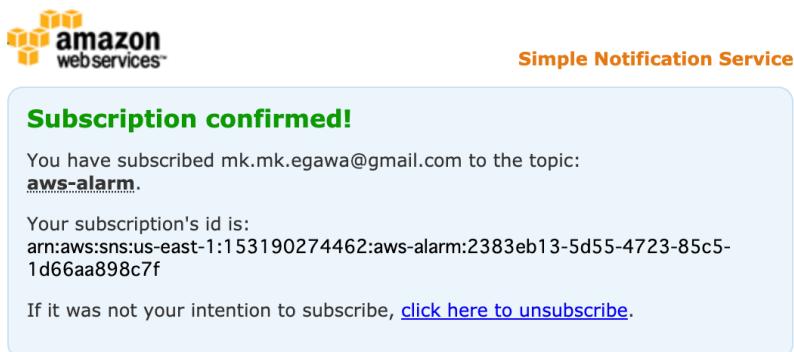
▲図 2.28 リクエストの確認

設定したメールアドレスに AWS からのメールが届いていますので、確認しましょう。メール本文の「Confirm subscription」リンクをクリックします。



▲図 2.29 「Confirm subscription」リンクをクリック

Subscription confirmed!と書かれた画面が表示されましたね。これで確認は完了です。
画面は閉じてしまってOKです。



▲図 2.30 サブスクリプションの確認

サブスクリプションのステータスが「確認済み」になりました。

ID	エンドポイント	ステータス	プロトコル	トピック
2383eb13-5d55-4723-85c5-1d66aa898c7f	mk.mk.egawa@gmail.com	確認済み	EMAIL	aws-alarm

▲図 2.31 サブスクリプションの確認完了

2.5 IAM ユーザーの作成

先ほど説明したように、ルートアカウントを使って作業を続けるのはあまりよくありません。ここで、作業用に IAM ユーザーを作成しましょう。

これも AWS のサービス「IAM」で作業します。

IAM のダッシュボードを開き、左のメニューで「ユーザー」を選択し、「ユーザーを追加」をクリックします。

2.5 IAM ユーザーの作成



▲図 2.32 「ユーザーを追加」をクリック

ユーザー詳細の設定画面で、ユーザー名を入力し、「AWS マネジメントコンソールへのアクセス」をチェックします。コンソールのパスワードは、「カスタムパスワード」を選択して自分のパスワードを設定しましょう。「パスワードのリセットが必要」のチェックは外します。

「次のステップ」をクリックします。

This screenshot shows the 'User Details Settings' step of the IAM User creation wizard. It has the following key elements:

- User Name:** A text input field containing 'testuser' (highlighted with a red box and labeled ①ユーザ名の入力).
- AWS Access Type Selection:** A section titled 'AWS アクセスの種類を選択' with the note: 'これらのユーザーから AWS にアクセスする方法を選択します。アクセスキーと自動生成パスワードは前のステップで提供されています。 詳細はこちら'. It includes two options:
 - アクセスの種類***: A dropdown menu currently set to 'プログラムによるアクセス'.
 - AWS マネジメントコンソールへのアクセス**: A checkbox that is checked (highlighted with a red box and labeled ②チェック).
- Console Password Selection:** A section titled 'コンソールのパスワード*' with three options:
 - 自動生成パスワード**: A radio button that is unselected.
 - カスタムパスワード**: A radio button that is selected (highlighted with a red box and labeled ③「カスタムパスワード」を選択してパスワードを入力).
 - パスワードの表示**: A checkbox that is unselected.
- Password Reset Requirement:** A section titled 'パスワードのリセットが必要' with a checkbox that is unselected.
- Buttons:** At the bottom right are 'キャンセル' (Cancel) and a large pink '次のステップ: アクセス権限' (Next Step: Access Permissions) button (highlighted with a red box and labeled ④クリック).

▲図 2.33 ユーザーの詳細設定

「既存のポリシーを直接アタッチ」を選択します。^{*5}

たくさんのポリシーが出てきましたね。行の左にある▶を少し広げて見ると、サービスごとに何ができるかの一覧が出てきます。本来あれば必要最低限のポリシーを1つずつ選択してアタッチしていくのですが、今回は制限のあまりない「AdministratorAccess」を選択しましょう。

「次のステップ」をクリックします。

The screenshot shows the 'Select the policies you want to attach' step of the IAM policy creation wizard. At the top, there are three buttons: 'Add users to groups' (disabled), 'Copy access permissions from an existing user' (disabled), and 'Attach existing policy directly' (selected). A red box and arrow labeled '①選択' point to the third button. Below this, a table lists various AWS policies. The first row, 'AdministratorAccess', has a checked checkbox in the 'Select' column, indicated by a red circle and arrow labeled '②選択'. The table includes columns for Policy Name, Type, Action, and Description.

ポリシー名	タイプ	次として使用	説明
<input checked="" type="checkbox"/> AdministratorAccess	ジョブ機能	なし	Provides full access to AWS services and re...
<input type="checkbox"/> AlexaForBusinessD...	AWSによる管理	なし	Provide device setup access to AlexaForBu...
<input type="checkbox"/> AlexaForBusinessF...	AWSによる管理	なし	Grants full access to AlexaForBusiness reso...
<input type="checkbox"/> AlexaForBusinessG...	AWSによる管理	なし	Provide gateway execution access to Alexa...
<input type="checkbox"/> AlexaForBusinessR...	AWSによる管理	なし	Provide read only access to AlexaForBusine...
<input type="checkbox"/> AmazonAPIGatewa...	AWSによる管理	なし	Provides full access to create/edit/delete A...
<input type="checkbox"/> AmazonAPIGatewa...	AWSによる管理	なし	Provides full access to invoke APIs in Amaz...
<input type="checkbox"/> AmazonAPIGatewa...	AWSによる管理	なし	Allows API Gateway to push logs to user's ...

▲図 2.34 ポリシーの設定

タグは、自分の好きなキーと値を設定でき、名前や使用目的などを分かりやすくするために使用します。

今回は設定を省略し、そのまま「次のステップ」をクリックします。

^{*5} アタッチとは「つける」という意味です。もし複数のユーザーで AWS を使用する場合は、グループを作成してユーザーをグループに追加します。そして、グループにポリシーをつけます。こうすることで、グループごとに権限を管理することができ、いちいちユーザー一人ずつのポリシーを管理する必要が無くなります。

2.5 IAM ユーザーの作成

ユーザーを追加

タグの追加 (オプション)

IAM タグは、ユーザーに追加できるキーと値のペアです。タグには、E メールアドレスなどのユーザー情報を含めるか、役職などの説明文とすることができます。タグを使用して、このユーザーのアクセスを整理、追跡、制御できます。詳細は[こちら](#)

キー	値 (オプション)
新しいキーを追加	

さらに 50 個のタグを追加できます。

キャンセル 戻る 次のステップ: 確認



▲図 2.35 タグの設定

作成する内容を確認し、「ユーザーの作成」をクリックします。

ユーザーを追加

確認

選択内容を確認します。ユーザーを作成した後で、自動生成パスワードとアクセスキーを確認してダウンロードできます。

ユーザー詳細

ユーザー名	myiamuser01
AWS アクセスの種類	プログラムによるアクセスと AWS マネジメントコンソールへのアクセス
コンソールのパスワードの種類	カスタム
パスワードのリセットが必要	いいえ
アクセス権限の境界	アクセス権限の境界が設定されていません

アクセス権限の概要

次のポリシーは、上記のユーザーにアタッチされます。

タイプ	名前
管理ポリシー	AdministratorAccess

タグ

追加されたタグはありません。

キャンセル 戻る ユーザーの作成



▲図 2.36 確認

ユーザーの作成が完了しました。

第2章 AWS を使い始める



▲図 2.37 IAM ユーザー作成完了

それでは、IAM ユーザーにも MFA を設定しましょう。
IAM の左のメニューから「ユーザー」を選択し、作成したユーザーのユーザー名をクリックします。



▲図 2.38 IAM ユーザーの MFA 設定

概要画面で「認証情報」タブを選択し、「MFA デバイスの割り当て」の右側にある「管理」をクリックします。

2.6 アカウント ID にエイリアスを設定する



▲図 2.39 IAM ユーザーの MFA 設定

ルートアカウントに設定した時と同様に図 2.7 画面が表示されたと思うので、同じ手順で設定しましょう。

2.6 アカウント ID にエイリアスを設定する

IAM ユーザーでマネジメントコンソールにログイン後、アカウント名の隣に数字が表示されています。これは、アカウント ID といって会社や組織のようなものです。このアカウント ID ごとに、ユーザーやシステムの管理を行います。

マネジメントコンソールにログインする時にも必要になるのですが、この数字では覚えづらいですよね。そこで、この数字に分かりやすい別名（エイリアスと言います）をつけることができます。



▲図 2.40 アカウント ID が表示されている

また「IAM」から設定しますので、AWS のサービスから IAM を選択しましょう。IAM の左のメニューで「ダッシュボード」を選択し、IAM ユーザーのサインインリンク

クの右側にある「カスタマイズ」をクリックします。



▲図 2.41 ダッシュボード

アカウントの別名を入力する画面が表示されますので、自分が分かりやすい名前を入力しましょう。（すでに他の人に使われている名前は使用できません。）

「はい、作成する」をクリックします。



▲図 2.42 アカウントの別名を入力

IAM ユーザーのサインインリンクが設定した別名に変更できました。

2.6 アカウント ID にエイリアスを設定する

The screenshot shows the AWS IAM console. On the left, a sidebar lists 'Identity and Access Management (IAM)' with options like 'AWS Account', 'Groups', 'Users', 'Roles', 'Policies', and 'ID Providers'. The 'AWS Account' section shows the account ID '153190274462'. The main area is titled 'Identity and Access Management へようこそ' and displays the URL 'https://start-infra-on-aws.signin.aws.amazon.com/console'. Below the URL, it says 'IAM リソース' and 'エイリアスに変更された' (Alias changed). It also shows 'ユーザー: 1', 'ロール: 4', 'グループ: 0', 'ID プロバイダ: 0', and 'カスタマー管理ポリシー: 0'. A progress bar at the bottom indicates '5 項目中 3 項目が完了しています' (3 items out of 5 completed).

▲図 2.43 変更完了

これで、マネジメントコンソールのログイン画面でアカウントにエイリアスを入力してログインできるようになります。



▲図 2.44 マネジメントコンソールのログイン画面

第3章

ネットワークを作成しよう

3.1 ネットワークを作成する前に

3.1.1 IP アドレスってどんなもの？

たとえば、2台のPCがあったとします。このPC同士が通信するためには、LANケーブルで繋いだだけでは通信できません。PCにはIPアドレスを与える必要があります。手紙を郵便でやり取りするためには住所が必要ですよね。IPアドレスはネットワーク上の住所みたいなものです。

IPアドレスは次のような「.」で区切った $8\text{ビット} \times 4\text{つ} = 32\text{ビット}$ の数字です。¹

▼リスト 3.1 IP アドレス

```
11000000.10101000.00000001.00000000
```

この1と0の羅列では、パッと見て解読するのが難しいですよね。そこで、人間がわかりやすいように10進数に変換するとこんな4つの数字になります。

▼リスト 3.2 IP アドレス (10進数)

```
192.168.1.0
```

だいぶ分かりやすくなりましたね。

¹ 普段私たちが使っている数字は10進数といい、コンピューターの世界ではすべてのデータを2進数(1か0)で表します。1ビットは、2進数で表した時の1桁のことです。

3.1.2 ネットワーク部とホスト部

2台のPC同士が通信するためには、IPアドレスを持っているだけでなく、同じNWに属していないといけません。では、どうやって同じNWか見分けるのでしょうか？

実は、IPアドレスはネットワーク部とホスト部に分かれています。この分け目がどこなのかを表すために、CIDR^{*2}表記という次のようなIPアドレスの書き方があります。

▼リスト 3.3 CIDR 表記

192.168.1.0/24

「/24」がネットワーク部とホスト部の分け目です。これは、左から24ビット目（10進数だと3つめの数字）までがネットワーク部であることを示しています。

ネットワーク部の数字が同じであれば、同じネットワークということになります。次に例を記載します。

- 192.168.1.1/24 と 192.168.1.2/24 は同じネットワークです。
- 192.168.1.1/24 と 192.168.2.1/24 は違うネットワークです。

また、ネットワークに接続できるコンピューターの数には制限があります。いくつかというと、ホスト部の範囲で表せる分です。/24の場合、ホスト部は最後の8ビット（10進数だと1つの数字）になりますね。8ビットは、00000000～11111111の数字が表せますが、10進数で表すと0～255になります。（256個）なお、0はネットワークアドレス、255はブロードキャストアドレスと決まっていてホストには使えないで、254個がホストアドレスとして使えます。

254台のコンピューターが接続できると考えてください。

3.1.3 ネットワークのサブネット化

1つのネットワークを、分割することができます。これをサブネット化と呼びます。サブネット化することで、お互いに接続するための設定をしない限り、サブネット間でアクセスできなくなります。意図しない通信が発生しないことで通信経路が分かりやすくなり

^{*2} Classless Inter-Domain Routing の略で、サイダーと呼びます。CIDRは「アドレスクラスの概念を用いないでIPアドレスの割り当てなどを可能にする仕組み」です。IPアドレスには、もともとIPアドレスの範囲ごとにクラス分けする「アドレスクラス」という考え方があり、ネットワーク部とホスト部の分け目はクラスごとに固定されていました。アドレスクラスに縛られずに分け目を決められるようにしたのがCIDRです。

ますし、セキュリティ面での安全性が高くなります。

たとえば次のネットワークがあったとします。ネットワーク部が左から 16 ビット分 (10 進数で数字 2 つ分) ですね。

▼リスト 3.4 1つのネットワーク

```
10.0.0.0/16
```

ネットワークを分割するためには、ホスト部を犠牲にしてネットワーク部を増やします。ネットワーク部を左から 24 ビット分に (10 進数で数字 3 つ分に) 増やしましょう。^{*3}

▼リスト 3.5 ネットワーク部を増やす

```
10.0.0.0/24
```

次のように、増やした分のネットワーク部でネットワークを分割することができます。これをサブネット化といいます。

▼リスト 3.6 サブネット化してネットワークを分割する

```
10.0.0.0/24・・・サブネット 1  
10.0.1.0/24・・・サブネット 2  
10.0.2.0/24・・・サブネット 3  
・・・
```

3.1.4 グローバル IP アドレスとプライベート IP アドレス

IP アドレスには、2 種類あります。グローバル（パブリック）IP アドレスとプライベート（ローカル）IP アドレスです。

インターネットに接続するための IP アドレスのことはグローバル IP アドレスと呼びます。

一方で、社内のネットワークや後述する VPC などのプライベート空間内でのみ利用される IP アドレスは、プライベート IP アドレスと呼びます。インターネットに接続することはできません。

^{*3} 増やすのは 2 ビット分でも 3 ビット分でも構いません。どれくらいネットワーク部を増やすかは、どれくらいサブネットを分割したいか、どれくらい各サブネットにコンピューターを接続したいかによって決まります。

複数のプライベート IP アドレスがグローバル IP アドレスを共有することで、コンピュータごとに IP アドレスを振り分けるのではなく、ネットワークごとに振り分けられるようになり、より多くのコンピュータがインターネットに接続できるようになっています。

また、IP アドレスの管理組織により、次の IP アドレスは、プライベート IP アドレスとして使うことが定められています。グローバル IP アドレスは、次の IP アドレス以外になります。

- 10.0.0.0～10.255.255.255
- 172.16.0.0～172.31.255.255
- 192.168.0.0～192.168.255.255

3.2 VPC を作成する

それでは、AWS で実際にネットワークを作成していきましょう。VPC というサービスを使います。

VPC (Virtual Private Cloud) とは、その名のとおり、仮想的なプライベートクラウドです。

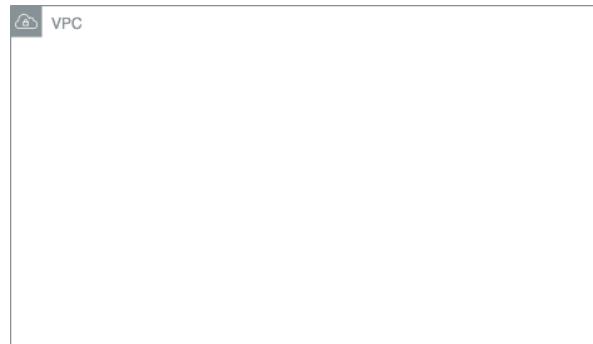
クラウドには、大きく次の 2 種類があります。

- パブリッククラウド・・・不特定多数のユーザーがサーバーを共有する。
- プライベートクラウド・・・特定の企業やユーザー専用のクラウド。その分費用が高くなるが、サーバーも独り占め。

AWS はパブリッククラウドなのですが、VPC を使うことで、まるでプライベートクラウドのように他のネットワークから隔離されたようなシステムをつくることができます。

これから AWS で作るものすべて（サブネットや EC2 インスタンスなど）は、VPC の中に作成します。

第3章 ネットワークを作成しよう



▲図 3.1 VPC

IAM ユーザーでログインしたら、トップページから「VPC」を検索しましょう。



▲図 3.2 VPC を検索する

VPC のダッシュボードが開きましたね。

左のメニューで「VPC」を選択し、「VPC の作成」をクリックします。

3.2 VPC を作成する



▲図 3.3 VPC の作成をクリック

VPC の作成画面です。VPC 名に「start-infra-on-aws」と入力しましょう。

CIDR ブロックとは VPC の中で使える IP アドレスの範囲を決めるものです。VPC はプライベートネットワークですので、プライベート IP アドレスの範囲を使います。今回は「10.0.0.0/16」と入力しましょう。

残りの項目は「IPv6 CIDR ブロックなし」を選択、「テナント」は「デフォルト」のままで、「作成」をクリックします。

▲図 3.4 VPC の作成

作成できましたね。

「閉じる」をクリックします。

第3章 ネットワークを作成しよう



▲図 3.5 VPC の作成完了

作成された VPC が確認できます。

The screenshot shows the AWS VPC dashboard. On the left, there's a sidebar with 'VPC ダッシュボード', 'VPC でフィルタリング:', 'VPC の選択', 'Virtual Private Cloud', 'VPC' (which is selected), 'サブネット', 'ルートテーブル', and 'インターネットゲートウェイ'. The main area is titled 'VPC の作成' and shows a table of VPCs. The table has columns: Name, VPC ID, 状態 (Status), IPv4 CIDR, IPv6 CIDR, DHCP オプションセット, and メインルートテーブル. Two VPCs are listed: 'vpc-04774cb3' and 'vpc-0f0666519afdb6ad7'. A red arrow points to the second VPC, with the text '作成したVPCが表示されている' (The newly created VPC is displayed).

Name	VPC ID	状態	IPv4 CIDR	IPv6 CIDR	DHCP オプションセット	メインルートテーブル
vpc-04774cb3	vpc-04774cb3	available	172.31.0.0/16	-	dopt-77025b10	rtb-74ee6612
start-infra-on-aws	vpc-0f0666519afdb6ad7	available	10.0.0.0/16	-	dopt-77025b10	rtb-0faad24aae497c00e2

▲図 3.6 VPC が作成された

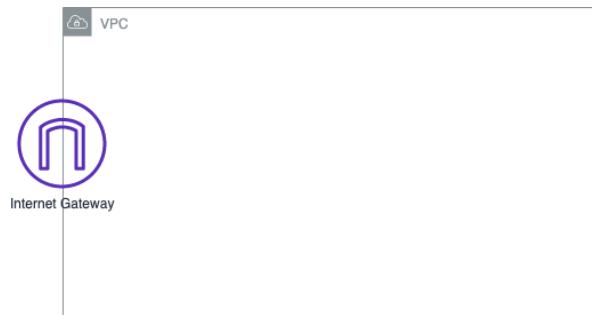
3.3 IGW を作成する

VPC が作成できましたが、このままでは VPC の中のサーバーからインターネットにアクセスできませんし、逆にインターネットから VPC の中のサーバーにアクセスできません。

これを可能にするために IGW (Internet Gateway) が必要です。

IGW は、VPC 内とインターネットをつなぐ玄関のようなものです。具体的にはプライベート IP アドレスとグローバル IP アドレスの紐付けを把握して、出入りする時に変換してくれます。

3.3 IGW を作成する



▲図 3.7 IGW

引き続き VPC ダッシュボードから操作します。左側のメニューで「インターネットゲートウェイ」を選択し、「インターネットゲートウェイの作成」をクリックします。



▲図 3.8 IGW の作成

名前タグに、「start-infra-on-aws」と入力し、「作成」をクリックします。



▲図 3.9 IGW の作成

IGW が作成できました。

インターネットゲートウェイ > インターネットゲートウェイの作成

インターネットゲートウェイの作成



▲図 3.10 IGW の作成完了

3.4 IGW を VPC にアタッチする

IGW は作成しただけだと VPC に紐づいていないため、VPC にアタッチ（紐づけ）する必要があります。



▲図 3.11 IGW が VPC にアタッチされていない

先ほど作成した IGW を選択し、「アクション」から「VPC にアタッチ」をクリックします。

3.4 IGW を VPC にアタッチする



▲図 3.12 「VPC にアタッチ」をクリック

プルダウンから先ほど作成した「start-infra-on-aws」VPC を選択し、「アタッチ」をクリックしましょう。



▲図 3.13 「start-infra-on-aws」VPC を選択

これで、VPC に IGW がアタッチできました。



▲図 3.14 IGW のアタッチ完了

3.5 サブネットを作成する

次に、「3.1.3 ネットワークのサブネット化」で説明したように、VPC 「10.0.0.0/16」 のネットワークをサブネット化しましょう。



▲図 3.15 サブネットの作成

3.5.1 パブリックサブネットとプライベートサブネット

サブネットは次の 2 種類があります。

- パブリックサブネット・・・IGW を経由してインターネットから接続可能。サブネット内からインターネットへのアクセスも可能。
- プライベートサブネット・・・IGW を経由せず、インターネットから接続不可。サブネット内からインターネットへのアクセスは NAT^{*4}を使用することで可能。

^{*4} NAT も IGW と同じくプライベート IP アドレスとグローバル IP アドレスを変換します。NAT は NAT 自身にグローバル IP アドレスを割り当てることで、プライベートサブネット内のサーバーが持つ

3.5.2 パブリックサブネットを作成する

今回は、パブリックサブネット 2つを作成します。引き続き VPC ダッシュボードで操作します。

左側のメニューで「サブネット」を選択し、「サブネットの作成」をクリックします。



▲図 3.16 「サブネットの作成」をクリック

3.5.3 アベイラビリティゾーンとは

サブネットの作成画面で「アベイラビリティゾーン」という見慣れない言葉が出てきましたね。

プライベート IP アドレスを NAT が持つ 1 つのグローバル IP アドレスに変換します。プライベートサブネットからインターネットに出て行くことはできても、インターネットから NAT を通してプライベートサブネットにアクセスすることはできません。

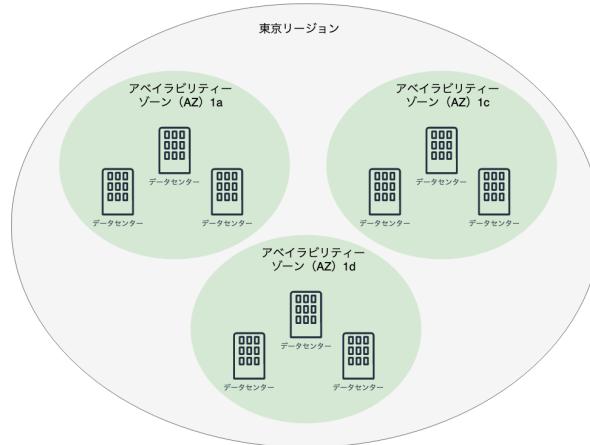
第3章 ネットワークを作成しよう



▲図 3.17 アベイラビリティゾーン

AWS でサブネットを作成すると、アベイラビリティゾーン (AZ) の中に作成されます。アベイラビリティゾーンとは、リージョンをさらに小さく分割した区分で、複数のデータセンターの集合体です。

例えば東京リージョンには現在、3 つの AZ があります。



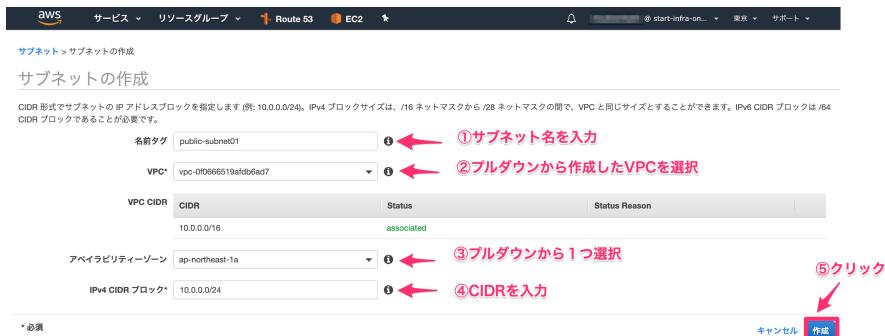
▲図 3.18 アベイラビリティゾーン

サブネットの作成画面で次のように設定し、「作成」をクリックします。

▼表 3.1 サブネット1つめの作成

サブネット名	VPC	アベイラビリティゾーン	IPv4 CIDR ブロック
public-subnet01	start-infra-on-aws の VPC	ap-northeast-1a	10.0.0.0/24

3.5 サブネットを作成する



▲図 3.19 サブネットの作成

サブネットが作成できました。



▲図 3.20 サブネットの作成

もう1つサブネットを作成するため、「サブネットの作成」をクリックし、次のように設定しましょう。

▼表 3.2 サブネット2つめの作成

サブネット名	VPC	アベイラビリティゾーン	IPv4 CIDR ブロック
public-subnet02	start-infra-on-aws の VPC	ap-northeast-1c	10.0.1.0/24



▲図 3.21 サブネットの作成

public-subnet01 と public-subnet02 が作成できましたね。

3.6 ルートテーブルの設定

サブネットを作成しただけでは、すべてプライベートサブネットになっています。これから作成するルートテーブルでサブネットから IGW に通信を中継するように設定することで、パブリックサブネットになります。

引き続き VPC ダッシュボードから操作します。

左のメニューから「ルートテーブル」を選択し、「ルートテーブルの作成」をクリックします。



▲図 3.22 ルートテーブルの作成

3.6 ルートテーブルの設定

ルートテーブル名に「public-subnet-route-table」と入力し、VPCにはプルダウンから作成したVPCを選択します。

「作成」をクリックします。



▲図 3.23 ルートテーブルの作成

「閉じる」をクリックします。



▲図 3.24 ルートテーブルの作成完了

作成したルートテーブルを選択し、「ルート」タブを表示したら、「ルートの編集」をクリックします。

第3章 ネットワークを作成しよう

The screenshot shows the AWS VPC Route Table Management interface. On the left sidebar, under 'Virtual Private Cloud' > 'Route Tables', there is a list of existing route tables: 'public-subnet-route-table' (selected), 'rtb-0faad24ae497c00e2', and 'rtb-74ee6612'. A red arrow points to the 'public-subnet-route-table' entry. Below the list, a pink box highlights the text '作成したルートテーブルが選択されている' (The newly created route table is selected). The main panel shows the 'Route' tab of the 'Route Table: rtb-0e013c3f10ab2c60b' configuration. A red box highlights the 'ルートの編集' (Edit Route) button. The route table configuration table has columns: '送信先' (Destination), 'ターゲット' (Target), and 'ステータス' (Status). One row shows '10.0.0.0/16' as the destination, 'local' as the target, and 'active' as the status.

▲図 3.25 ルートの編集

「ルートの追加」ボタンをクリックしたら、送信先に「0.0.0.0/0」を入力し、ターゲットには「Internet Gateway」を選択します。これは、ルーティングテーブル上に一致する送信先が設定されていない場合は、デフォルトルートの転送先 (IGW) に中継するという意味です。

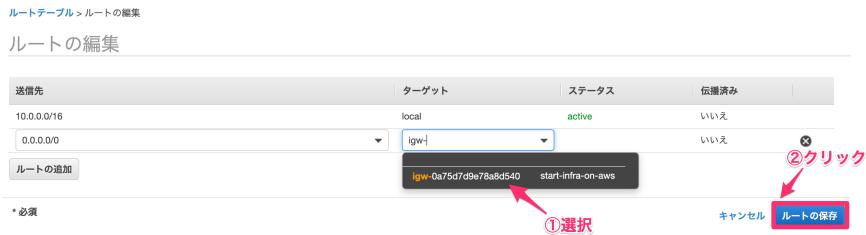
もともと送信先「10.0.0.0/16」は「local」というルートが設定されていましたね。そのため、送信先がVPC内のネットワーク（10.0.0.0/16）以外であれば、すべてIGWに中継され、インターネットに探しにいくことになります。

The screenshot shows the 'Route Table: rtb-0e013c3f10ab2c60b' configuration. In the 'Route' tab, the 'ルートの編集' (Edit Route) button is highlighted with a red box and a red arrow pointing to it. The '送信先' (Destination) field contains '0.0.0.0/0' (①入力). A dropdown menu is open, showing options: 'Egress Only Internet Gateway Instance', 'Internet Gateway' (②選択), 'NAT Gateway', 'Network Interface', 'Peering Connection', 'Transit Gateway', and 'Virtual Private Gateway'. The 'ターゲット' (Target) column shows 'local' and 'active' status. The 'ステータス' (Status) column shows 'いいえ' (No) for the first row and 'いいえ' (No) with a crossed-out icon for the second row.

▲図 3.26 ルートの編集

ターゲットに作成したIGWが出てきますので選択して、「ルートの保存」をクリックします。

3.6 ルートテーブルの設定



▲図 3.27 ルートの保存

ルートが保存できました。



▲図 3.28 ルートの保存完了

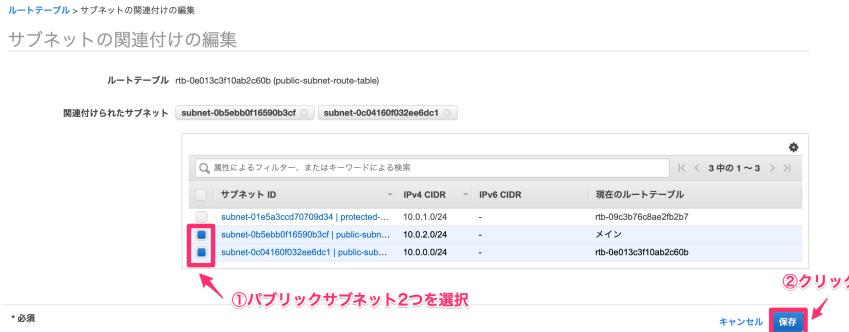
ルートテーブルを選択したまま、「サブネットの関連付け」タブを表示し、「サブネットの関連付けの編集」をクリックします。



▲図 3.29 サブネットの関連付け

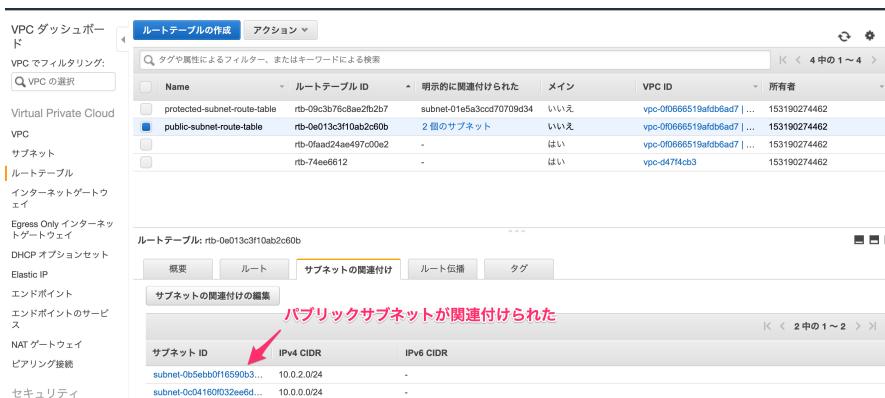
第3章 ネットワークを作成しよう

作成したサブネットが表示されますので、パブリックサブネット2つを選択し、「保存」をクリックします。



▲図 3.30 パブリックサブネットの関連付け

パブリックサブネット2つが関連づけられました。



▲図 3.31 パブリックサブネットの関連付け完了

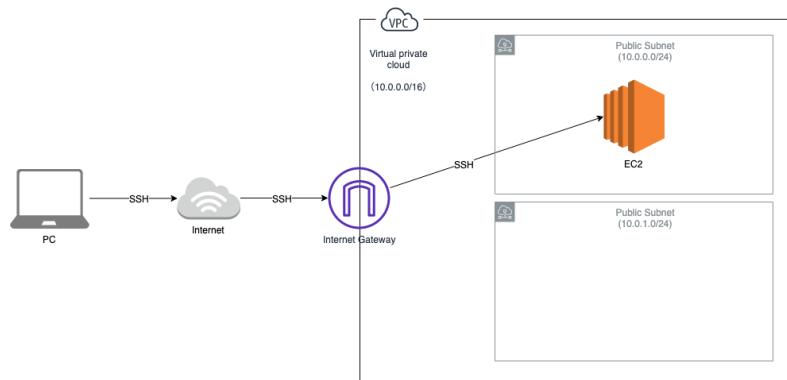
第4章

EC2 インスタンスを作成しよう

4.1 EC2 インスタンスを作成する

EC2 とは Elastic Compute Cloud の略で、アプリケーションを実行するために必要なコンピューターのようなものです。EC2 で作成したコンピューターのことをインスタンスと呼びます。

パブリックサブネットに EC2 インスタンスを作成しましょう。



▲図 4.1 EC2 を作成する

AWS マネジメントコンソールのトップ画面で、「EC2」と検索します。



▲図 4.2 EC2 を検索

EC2 ダッシュボードで、「インスタンスの作成」をクリックします。



▲図 4.3 「インスタンスの作成」をクリック

マシンイメージは 無料利用枠が対象の Amazon Linux 2 を選択します。

4.1 EC2 インスタンスを作成する



▲図 4.4 Amazon Linux 2 を選択

インスタンスタイプも、デフォルトで選択されている無料利用枠の対象（t2.micro）を選択したまま「次の手順」をクリックします。インスタンスタイプとは、インスタンスのスペックを決めるもので、スペックがよければよいほど利用料が高くなります。



▲図 4.5 t2.micro を選択

インスタンスの詳細設定では、ネットワークに先ほど作成した VPC を選択し、サブネットには「public-subnet01」を選択、自動割り当てパブリック IP^{*1}を「有効」にします。

^{*1} このパブリック IP はサーバーを再起動すると変わります。普通はサーバーを再起動しても変わらない IP アドレス（EIP）を割り当てますが、今回は省略します。

第4章 EC2インスタンスを作成しよう

手順3: インスタンスの詳細の設定
要件に合わせてインスタンスを設定します。同じ AMI からの複数インスタンス作成や、より低成本を実現するためのスポットインスタンスのリクエスト、インスタンスへのアクセス管理ロール割り当てなどを行うことができます。

①作成したVPCを選択
②パブリックサブネットを選択
③有効を選択

1. AMI の選択 2. インスタンスタイプの選択 3. インスタンスの設定 4. ストレージの追加 5. タグの追加 6. セキュリティグループの設定 7. 確認

▲図 4.6 インスタンスの詳細設定

その他の設定はデフォルトのまま「次の手順」をクリックします。

削除保護の有効化 ① 設定された削除から保護します
モニタリング ① CloudWatch 詳細モニタリングを有効化
追加料金が適用されます。
チナンサー ① 共有 - 共有ハードウェアインスタンスの実行
専有チナンサーには追加料金が適用されます。
Elastic Inference ① Elastic Inference アクセラレーターを追加
追加料金が適用されます。
T2/T3 無制限 ① 有効化
追加料金が適用される場合があります。

▼ ネットワークインターフェイス ①

デバイス	ネットワークインターフェイス	サブネット	プライマリ IP	セカンダリ IP アドレス	IPv6 IP
eth0	新しいネットワークイフ	subnet-0c04160f	自動的に割り当て	IP の追加	IP の追加

デバイスの追加
高度な詳細

キャンセル 戻る 確認と作成 次の手順: ストレージの追加

▲図 4.7 「次の手順」をクリック

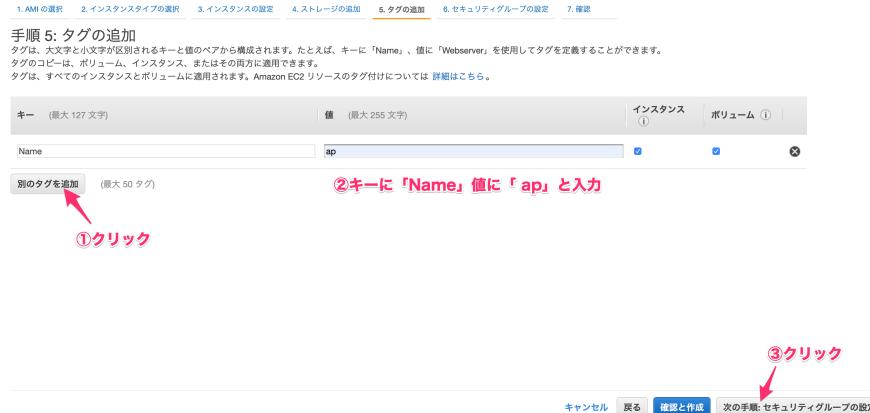
ストレージはデフォルトの設定のままで「次の手順」をクリックします。

4.1 EC2 インスタンスを作成する



▲図 4.8 「次の手順」をクリック

次に、タグの作成です。インスタンスをたくさん作成した時に、どれが何のインスタンスか分からなくならないように使います。キーに「Name」値に「ap」と入力して「次の手順」をクリックします。



▲図 4.9 「次の手順」をクリック

4.1.1 プロトコルとは？

次のセキュリティグループの設定では、インスタンスに対してどんな通信のアクセスを許可するか設定できます。セキュリティグループの設定をする前に、少しプロトコルにつ

いて説明します。プロトコルとは、コンピューター同士が通信するためのルールです。

たとえば HTTP はブラウザでウェブページを表示したりするのに使うプロトコルです。SSH はコマンドでサーバーに接続するためのプロトコルです。

さらに、プロトコルは次の 4 層にわかっています。HTTP や SSH はアプリケーション層のプロトコルで、トранスポート層は TCP というプロトコルを使用しています。

- アプリケーション層 (HTTP、SSH)
- トランスポート層 (TCP)
- インターネット層 (IP)
- ネットワークインターフェイス層 (Ethernet)

4.1.2 ポート番号とは？

IP アドレスを建物の住所にたとえましたが、ポート番号は部屋の番号だと思ってください。プロトコルごとに使用するポート番号が異なります。

また、ポートは 0 から 65535 までありますが、その中でも 0~1023 までを「well-known port」と呼び、あるプロトコルとポートの組み合わせで予約されています。AWS マネジメントコンソールのセキュリティグループの設定で「タイプ」を HTTP にするとポートが自動で 80 になりますし、HTTPS にすると 443 になりますね。これは、使うポートが決まっているプロトコルだから AWS が自動で設定してくれているのです。

4.1.3 セキュリティグループの設定

では、セキュリティグループの設定に戻りましょう。

セキュリティグループ名と説明は「ap」と変更します。

デフォルトでは、どこからでも SSH で接続できるような設定になっています。(SSH 接続については後ほど説明します。) これでは不正にサーバーにアクセスされてしまう可能性があるので、自分の PC からのみアクセスできるようにしましょう。

ソースを自分の PC が使用しているグローバル IP アドレス/32^{*2}に修正します。

^{*2} CIDR 表記で 1 つの IP アドレスのみを表す場合に /32 と表します。

4.1 EC2 インスタンスを作成する



▲図 4.10 セキュリティグループの設定

あれ、私グローバル IP アドレスなんて使ってないけど・・とおもいましたか？ 無意識でもインターネットに接続しているのなら、グローバル IP アドレスを使用しているのです。自宅や会社の wifi がグローバル IP アドレスを持っていて、そのグローバル IP アドレスを使ってインターネットに接続しているはずです。

自分の PC が使用しているグローバル IP アドレスはこちらのサイトにアクセスすれば教えてくれます。

https://www.cman.jp/network/support/go_access.cgi



▲図 4.11 グローバル IP アドレスの確認

ソースを修正できたら、「確認と作成」をクリックします。

第4章 EC2 インスタンスを作成しよう

なお、家の wifi やカフェの wifi などはグローバルアドレスが固定でなく、時間がたつと変わってしまう場合があります。もし EC2 インスタンスに繋がらなくなったり、作成した HP にアクセスできなくなったらグローバル IP アドレスが変わっていないか確認しましょう。変わっていたら、セキュリティグループの修正で IP アドレスを修正しましょう。

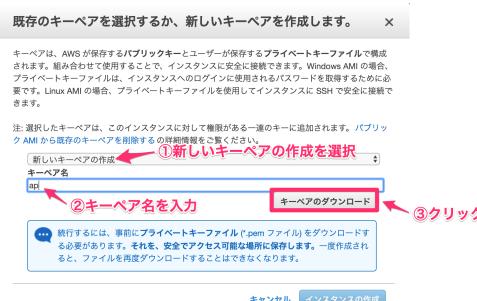
インスタンスの作成内容を確認し、「起動」をクリックします。



▲図 4.12 EC2 の起動

キーペア作成しましょう。キーペアとは、EC2 インスタンスに接続するための鍵です。キーペアについては、後ほど説明します。

「新しいキーペアの作成」を選択し、キーペア名に「ap」と入力し、「キーペアのダウンロード」をクリックします。



▲図 4.13 キーペアのダウンロード

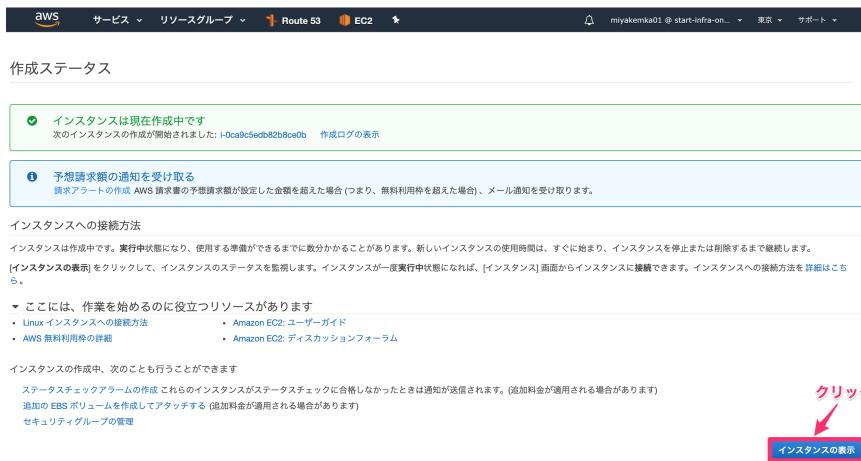
4.1 EC2 インスタンスを作成する

ap.pem ファイルがダウンロードできたら、「インスタンスの作成」をクリックします。



▲図 4.14 インスタンスの作成

EC2 インスタンスの作成が開始されました。



▲図 4.15 EC2 の作成開始

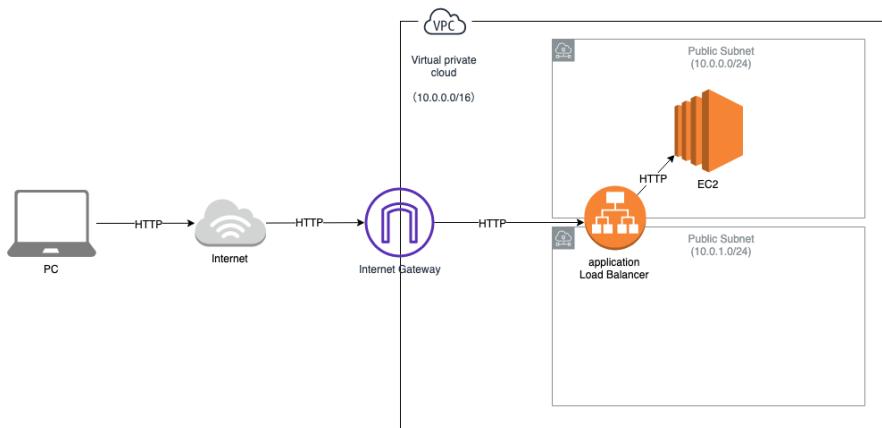
第5章

ロードバランサーを作成しよう

5.1 ロードバランサーの作成

ロードバランサーとは、負荷分散装置ともいいます。その名のとおり、負荷を分散する装置なのですが、他にも後述する HTTPS 通信をすることができたりします。今回はできる限り無料枠の中でシステムを構築するため、EC2 インスタンスは 1 台しか作成していません。このため、負荷分散装置としての役割はないですが、HTTPS の終端装置としてロードバランサーを配置したいと思います。

まずは HTTP の通信を EC2 インスタンスに中継できるようにしましょう。



▲図 5.1 ロードバランサーの作成

ロードバランサーは、EC2 ダッシュボードから操作していきます。左側のメニューで

5.1 ロードバランサーの作成

「ロードバランサー」を選択しましょう。



▲図 5.2 「ロードバランサー」を選択

「ロードバランサーの作成」をクリックします。



▲図 5.3 「ロードバランサーの作成」をクリック

ロードバランサーは、2019年9月現在2種類存在し、扱えるプロトコルが異なります。¹今回はHTTPやHTTPSを扱いたいので、一番左の「Application Load Balancer」(ALB)の「作成」をクリックします。

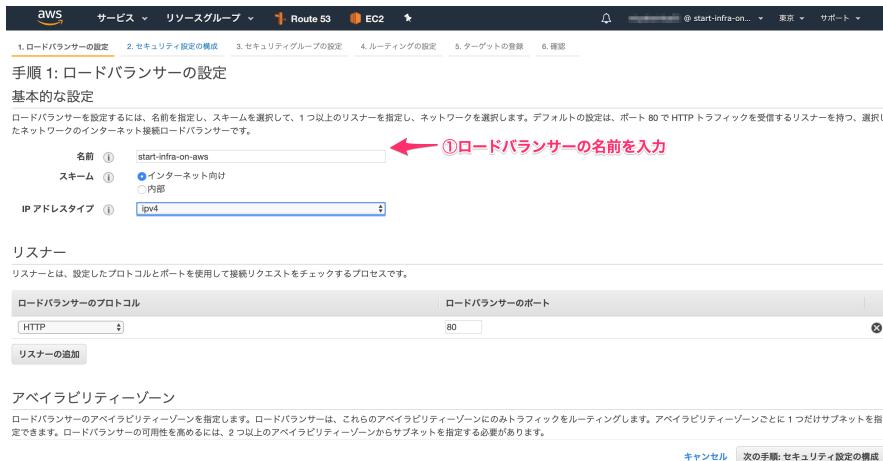
¹ Application Load BalancerはL7のロードバランサー、Network Load BalancerはL4のロードバランサーといわれます。HTTPやHTTPSなどアプリケーション層をL7、TCPなどのトランスポート層をL4と呼ぶためです。L7の方がアプリケーションごとに振り分けができたりして高機能ですが、色々できる分L4のロードバランサーよりも処理自は遅くなります。

第5章 ロードバランサーを作成しよう



▲図 5.4 「Application Load Balancer」を選択

ロードバランサーの名前には「start-infra-on-aws」と入力します。

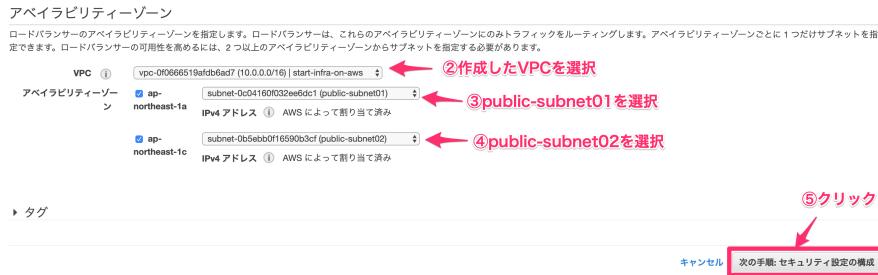


▲図 5.5 「start-infra-on-aws」と入力

VPC は「start-infra-on-aws」を選択します。ロードバランサーは、2つ以上のアベイリティーゾーン (AZ) の使用を前提としていますので、public-subnet01 と public-subnet02 の2つを選択します。

「次の手順」をクリックします。

5.1 ロードバランサーの作成



▲図 5.6 VPC とアベイラビリティーゾーンを選択

https を使用するように警告が出ますが、今はそのままで構いません。

「次の手順」をクリックします。



▲図 5.7 「次の手順」をクリック

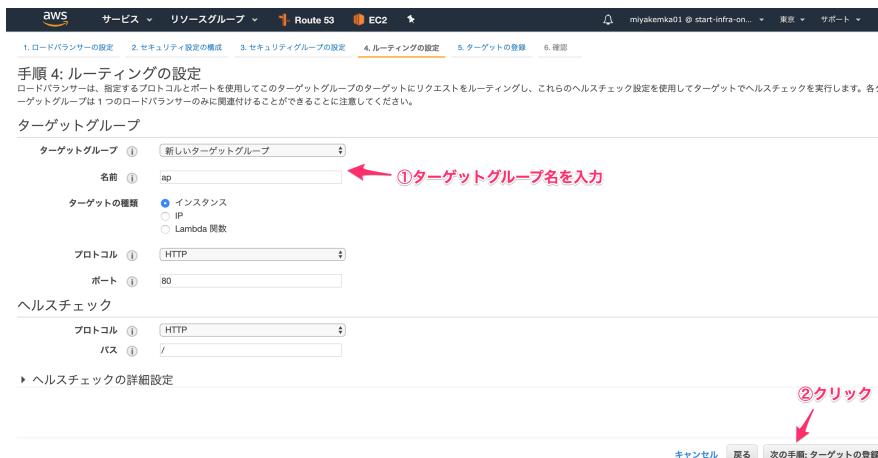
セキュリティグループの設定は、セキュリティグループ名と説明に「alb」と入力します。許可する通信として、タイプ「http」、プロトコル「TCP」ポート「80」、ソースには、やはり自分の PC のみアクセスできるように「自分の PC が使用しているグローバル IP/32」を入力しましょう。

第5章 ロードバランサーを作成しよう



▲図 5.8 セキュリティグループの設定

ルーティングの設定は、ロードバランサーから通信を振り分ける先のターゲットを設定します。「新しいターゲットグループ」を選択し、名前に「ap」と入力します。
他はデフォルト設定のまま、「次の手順」をクリックします。



▲図 5.9 ルーティングの設定

ロードバランサーから通信を振り分ける先のターゲットを登録します。インスタンスの一覧から ap の EC2 インスタンスを選択し、「登録済みに追加」をクリックします。

5.1 ロードバランサーの作成



▲図 5.10 ターゲットの登録

登録済みターゲットに、ap の EC2 インスタンスが登録されました。

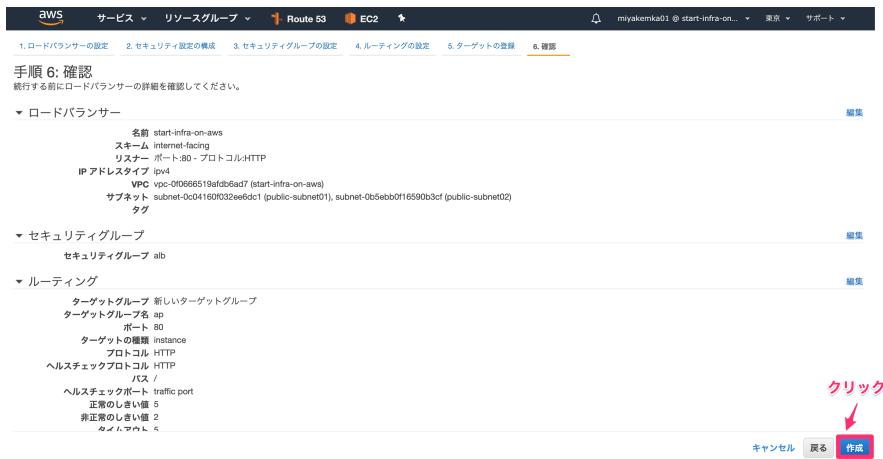
「次の手順」をクリックします。



▲図 5.11 EC2 インスタンスが登録された

設定内容を確認したら、「作成」をクリックします。

第5章 ロードバランサーを作成しよう



▲図 5.12 「作成」をクリック

ロードバランサーが作成できました。



▲図 5.13 ロードバランサーが作成できた

5.2 EC2 インスタンスのセキュリティグループの編集

ロードバランサーを作成しただけでは、ロードバランサーから EC2 インスタンスに接続できません。EC2 インスタンス側で、ロードバランサーからのアクセスを許可する必要があります。

通信のアクセスを許可する設定はどこでしたか？

セキュリティグループでしたね。そのまま EC2 ダッシュボードの左のメニューから「セキュリティグループ」を選択し、ap のセキュリティグループを選択します。

次に、インバウンドタブを表示し、「編集」をクリックしましょう。

5.2 EC2 インスタンスのセキュリティグループの編集

The screenshot shows the AWS Management Console with the EC2 service selected. In the left navigation pane, 'セキュリティグループ' (Security Groups) is highlighted. On the main page, a table lists several security groups. One row for 'ap' is selected, indicated by a blue border and a red arrow pointing to it. The table columns include Name, グループ ID, グループ名, VPC ID, 所有者, and 説明. Below the table, a modal window titled 'セキュリティグループの作成' (Create Security Group) is open. It has tabs for '選択' (Select), 'インバウンド' (Inbound), and 'アウトバウンド' (Outbound). The 'インバウンド' tab is selected, indicated by a red arrow. A sub-modal window titled 'インバウンドのルールの編集' (Edit Inbound Rule) is also open, showing a list of rules. A red arrow points to the '④クリック' (Click) button in this sub-modal.

▲図 5.14 セキュリティグループの編集

「ルールの追加」をクリックし、タイプ「HTTP」、プロトコル「TCP」、ポート「80」、ソースには「alb」のセキュリティグループを選択します。

「保存」をクリックします。

The screenshot shows the 'Edit Inbound Rule' dialog. It has tabs for 'タイプ' (Type), 'プロトコル' (Protocol), 'ポート範囲' (Port Range), and 'ソース' (Source). The 'ソース' tab is selected, showing 'カスタム' (Custom) selected in the dropdown. The source field contains 'alb'. A red arrow points to the '②80ポートを許可' (Allow port 80) button. Below the source field, a note says 'ルールの追加' (Add rule) with a red arrow pointing to it. Another note at the bottom says '③albのセキュリティグループを許可' (Allow security group alb). The '保存' (Save) button is highlighted with a red arrow. A red box highlights the entire 'ソース' section.

▲図 5.15 ALB から HTTP の通信を許可する

インバウンドに、alb のセキュリティグループが追加されました。

これで、ロードバランサーから EC2 インスタンスへのアクセスが可能になりました。

第5章 ロードバランサーを作成しよう

The screenshot shows the AWS Security Groups page. In the left sidebar, under 'セキュリティグループ', there is a section for 'ALB'. In the main content area, a table lists security groups, including 'alb' which has been selected. Below the table, a modal window titled 'セキュリティグループ: sg-0e679263d0ebb5305' shows an 'Inbound' tab with a new rule being added. The rule details are: Type: HTTP, Protocol: TCP, Port Range: 80, Source: sg-0727d741c48884b2e (alb). A red arrow points to the source field.

▲図 5.16 ALB から HTTP の通信が許可された

5.3 ターゲットの状態確認

左のメニューで、ロードバランサーの「ターゲットグループ」を選択しましょう。

The screenshot shows the 'Load Balancer creation' page. In the left sidebar, under 'ロードバランサー', there is a section for 'ターゲットグループ'. A red arrow points to this section. In the main content area, a table lists target groups, including 'start-infra-on-aws' which has been selected. Below the table, a modal window titled 'ロードバランサー: start-infra-on-aws' shows the 'Target Groups' tab selected. The basic settings for the target group are displayed, including the name, ARN, DNS name, status, type, scheme, and IP address type. A red box highlights the 'Target Groups' tab.

▲図 5.17 「ターゲットグループ」を選択

ターゲットタブを表示し、登録済みターゲットを見てみると、ステータスが「unhealthy」になっていますね。

5.3 ターゲットの状態確認

これは、ロードバランサーが EC2 インスタンスにヘルスチェック^{*2}をしても ap インスタンス上で何もレスポンスを返すアプリケーションが動いていないためです。

これからいよいよアプリケーションを起動していきます。

The screenshot shows the AWS CloudWatch Metrics console with the 'Targets' tab selected. A red arrow points to the 'Health Check' status for the target 'ap'. Another red arrow points to the 'unhealthy' status in the table below.

インスタンス ID	名前	ポート	アベイラビリティゾーン	ステータス
i-0ee7de72a5cc4f08b	ap	80	ap-northeast-1a	unhealthy ⓘ

▲図 5.18 ターゲットが unhealthy

^{*2} ALB にはヘルスチェックの設定がされています。デフォルト設定のまま作成しましたが、ALB が 30 秒間隔で EC2 インスタンスへ HTTP アクセスし、200ok が返ってくるようであれば正常 (healthy) とみなされるように設定されています。ターゲットが unhealthy の場合、そのターゲットには通信を振り分けません。

第6章

SSH で EC2 インスタンスにログインしよう

6.1 SSH とは

SSH とは「Secure Shell」の略で、ネットワークに接続されたサーバーなどの機器をネットワークごとに安全に操作するための手段です。通信は暗号化され、不正にサーバーにログインすることができないような仕組みになっています。ログイン方法については次の2種類があります。

6.1.1 パスワード認証方式

接続先にユーザー名とパスワードを設定して、ログイン時にパスワードで認証する方法です。パスワードが分かれば誰でもログインできてしまうため、安全性は公開鍵認証方式よりも劣ります。

6.1.2 公開鍵認証方式

AWS の EC2 インスタンスはデフォルトでこちらの方式のみ有効になっています。接続のための詳細なフローの説明は省略しますが、公開鍵認証方式はよく南京錠にたとえられます。

公開鍵と秘密鍵のペアを作成し、秘密鍵を接続元に、公開鍵を接続先に配置します。こうすることで、秘密鍵を使ってサーバーにログインできるようになります。

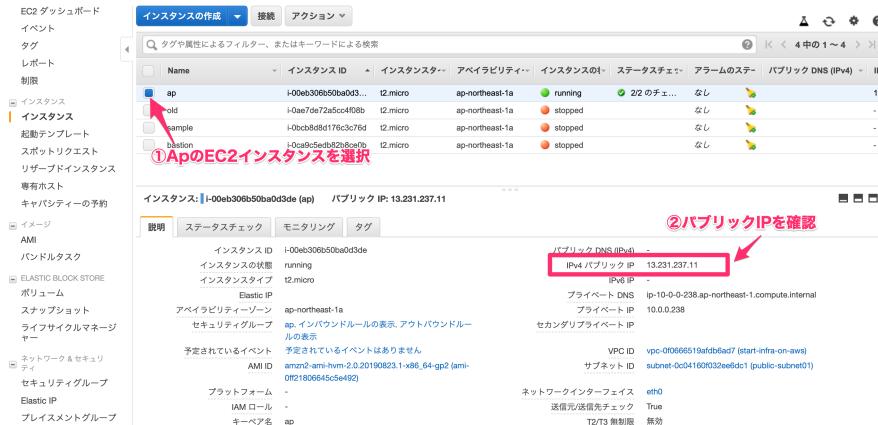


▲図 6.1 公開鍵認証方式

EC2 インスタンスを作成した時に、キーペアを作成して「ap.pem」ファイルをダウンロードしましたよね。実は、ダウンロードした「ap.pem」ファイルが秘密鍵です。公開鍵は、AWS がインスタンス作成時にインスタンスに配置してくれています。

6.2 SSH 接続する

ap の EC2 インスタンスのパブリック IP を確認しておきましょう。



▲図 6.2 パブリック IP の確認

お使いの PC が MAC なのか Windows なのかによって SSH の方法が異なります。自分が使用している PC の手順を確認してください。

6.3 MAC の場合

Finder を表示し、ツールバーの「移動」から「フォルダへ移動」を選択し、「~/ssh/」を入力します。このフォルダに、EC2 インスタンスを作成した際にダウンロードした「ap.pem」を移動します。

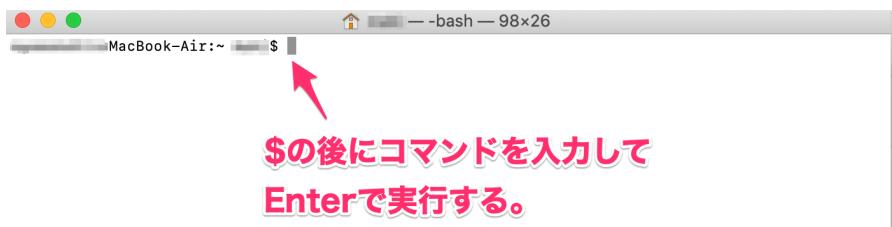
アプリケーションから、ターミナルを開きます。



ターミナル.app

▲図 6.3 ターミナル.app を開く

ターミナルでコマンドを実行するには、\$の後にコマンドを入力し、Enter を押してコマンドを実行します。



▲図 6.4 ターミナルでコマンドを実行する

実行するコマンドは「\$」から書きますので、実際に実行する時には「\$」の後ろから入力してください。先頭に「\$」がついていない場合、それはコマンドの実行結果です。

早速、次のコマンドを実行しましょう。

「/Users/<ユーザー名>/ssh」フォルダに移動し（1行目）*1

vi コマンドで「config」というファイル名のファイルを作成して開きます。（2行目）

▼リスト 6.1 config ファイルの作成

```
1: $ cd ~/.ssh
2: $ vi config
```

次のテキストをコピーします。<ap EC2 インスタンスのパブリック IP アドレス>の中は先ほど確認した ap EC2 インスタンスのパブリック IP アドレスに修正しましょう。

▼リスト 6.2 config ファイルの編集

```
3 行目：接続先の名前
4 行目：接続先の IP アドレス
5 行目：ログインユーザー（デフォルトで存在する ec2-user を使用します）
6 行目：秘密鍵の場所
7 行目：接続ポート番号（SSH は 22 と決まっています）
```

ターミナルに戻り、貼り付けます。貼り付けられない場合、キーボードで「i」を押して編集モードに切り替えてから貼り付けましょう。

▼リスト 6.3 config ファイルの編集

```
1: # start infra on aws
2:
3: Host ap
```

*1 「~」は「/Users/<ユーザー名>」フォルダのことです。ホームディレクトリとも呼ばれます。

第6章 SSHでEC2インスタンスにログインしよう

```
4: HostName <ap EC2インスタンスのパブリックIPアドレス>
5: User ec2-user
6: IdentityFile ~/.ssh/ap.pem
7: Port 22
```



```
Host ap
  HostName 13.231.237.11
  User ec2-user
  IdentityFile ~/.ssh/ap.pem
  Port 22

Host aws
  HostName 13.231.237.11
  User ec2-user
  IdentityFile ~/.ssh/aws.pem
  Port 22

-- INSERT --
```

キーボードで「i」を押して
編集モードにする

▲図 6.5 ターミナルでコマンドを実行する

貼り付けられたら、キーボードで「esc」を押して、コマンドモードに切り替えます。
次のコマンドでファイルを上書き保存します。

▼リスト 6.4 config ファイルの保存

```
1: :wq
```

```
# start infra on aws
Host ap
  Hostname 13.231.237.11
  User ec2-user
  IdentityFile ~/.ssh/ap.pem
  Port 22

```
「esc」キーを押してコマンドモードに切り替え

「:wq」で上書き保存する
```
```

▲図 6.6 ターミナルでコマンドを実行する

ファイルが閉じたら、次のコマンドで ap EC2 インスタンスにログインしてみましょう。

▼リスト 6.5 EC2 インスタンスにログイン

```
1: $ ssh ap
```

こんなメッセージが表示されました。初回の ssh 接続時には、接続してもよいのか確認されます。

yes を入力して「Enter」を押しましょう。

▼リスト 6.6 確認メッセージが表示される

```
1: The authenticity of host '52.196.89.55 (52.196.89.55)' can't be established.  
2: ECDSA key fingerprint is SHA256:+xXt86U/2APwsGsb7Ff84mbV5aC6q4JEhKuB5/xZqg.  
3: Are you sure you want to continue connecting (yes/no)? yes
```

今度は別のメッセージが出ました。

キーペアのファイルのパーミッションがオープンすぎる、というメッセージです。

▼リスト 6.7 警告が表示される

```
8: ec2-user@52.196.89.55: Permission denied (publickey, gssapi-keyex, gssapi-with-mic).
```

普段はあまり意識することがないかもしれません、ファイルやディレクトリにはパーミッションと呼ばれる、アクセス権限が設定されています。

パーミッションは、次の単位でそれぞれの権限を設定します。

- 自分（ログインしているユーザー）
- グループメンバ（自分と同じグループの人）
- 他人（他のグループの人）

権限は、次の3つがあります。

- 読み取り権限（読んでOK？）
- 書き込み権限（書いてOK？）
- 実行権限（動かしてOK？）

また、権限は記号と数字の2パターンで表すことができます。

▼表6.1 パーミッション

権限	記号	数字
読むことができる（Readable）	r	4
書くことができる（Writable）	w	2
実行することができる（eXecutable）	x	1
なにもできない	-	0

次のコマンドを打ってみましょう。

▼リスト6.8 ファイルの情報を表示する

```
1: $ ls -l ap.pem
```

こんな結果が表示されました。

▼リスト6.9 ファイルの情報が表示された

```
1: -rw-r--r--@ 1 hogeuser staff 1692 9 1 18:26 ap.pem
```

「-rw-r--r--」の部分がパーミッションです。

左から 2 文字目から 3 文字ずつ、自分 (rw-)、グループメンバ (r--)、他人 (r--) の順番でパーミッションが表されています。

これは自分は読み書き可能、グループメンバーと他人も読み取りは可能というパーミッションです。

数字だと、3 文字ずつ数字に直して足し算し、644 と表します。

それでは、パーミッションを自分は読み書き可能、グループメンバーと他人は何もできないようにパーミッションを変更しましょう。

▼リスト 6.10 パーミッションの変更

```
1: $ chmod 600 ap.pem
```

パーミッションが、自分 (rw-)、グループメンバ (---)、他人 (---) になっていることを確認します。

▼リスト 6.11 パーミッションの確認

```
1: $ ls -l ap.pem
2: -rw-----@ 1 hoge  staff  1692  9  1 18:26 ap.pem
```

パーミッションが変更できましたね。

それでは、もう一度 ap EC2 インスタンスにログインしましょう。

▼リスト 6.12 ap EC2 インスタンスにログイン

```
1: $ ssh ap
2:
3:      _ _|_ _|_
4:      _ | (   _ ) /   Amazon Linux 2 AMI
5:      _ _| \_ _| _ |
6:
7: https://aws.amazon.com/amazon-linux-2/
8: 4 package(s) needed for security, out of 12 available
9: Run "sudo yum update" to apply all updates.
```

無事にログインできましたね。

サーバーからログアウトする場合は次のコマンドです。

▼リスト 6.13 サーバーからログアウトする場合

```
1: $ exit
```

6.4 Windowsの場合

Windows PCの場合は、SSH接続するためTera Termというアプリをダウンロードします。

次のサイトにアクセスし、「teraterm-x.xxx.exe」をクリックします。(広告のボタンが目立ちますが、間違ってクリックしないようにしましょう。)

<https://ja.osdn.net/projects/ttssh2/releases/>



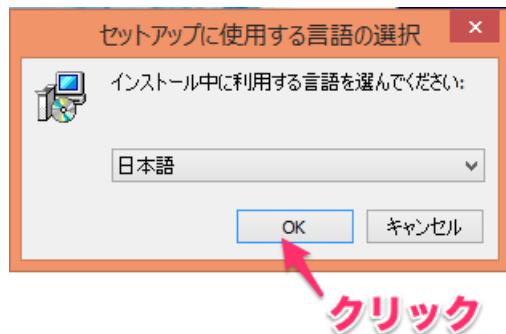
▲図 6.7 teraterm のダウンロード

ダウンロードしたexeファイルをダブルクリックします。



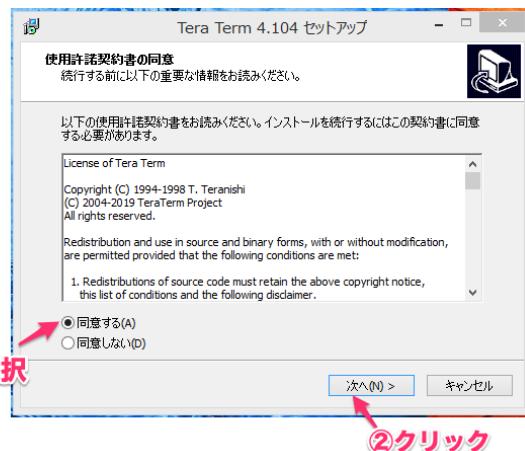
▲図 6.8 exeファイルをダブルクリック

インストール言語は「日本語」のままで「OK」をクリックします。



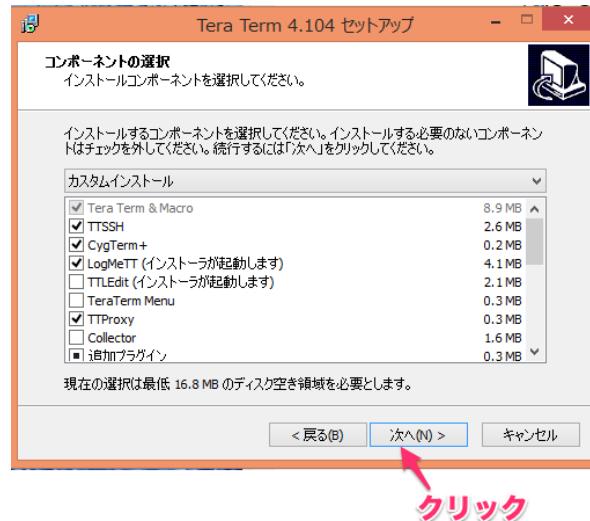
▲図 6.9 「OK」をクリック

「同意する」を選択し、「次へ」をクリックします。



▲図 6.10 「次へ」をクリック

コンポーネントもそのまで「次へ」をクリックします。



▲図 6.11 「次へ」をクリック

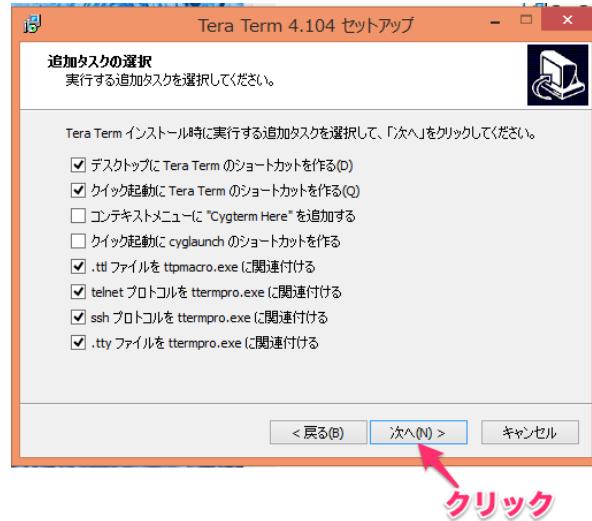
ユーザーインターフェースの言語も「日本語」のまま「次へ」をクリックします。



▲図 6.12 「次へ」をクリック

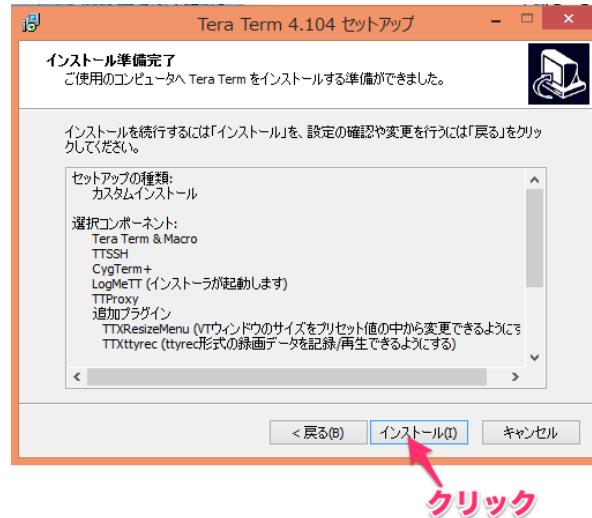
追加タスクもそのまで「次へ」をクリックします。

6.4 Windowsの場合



▲図 6.13 「次へ」をクリック

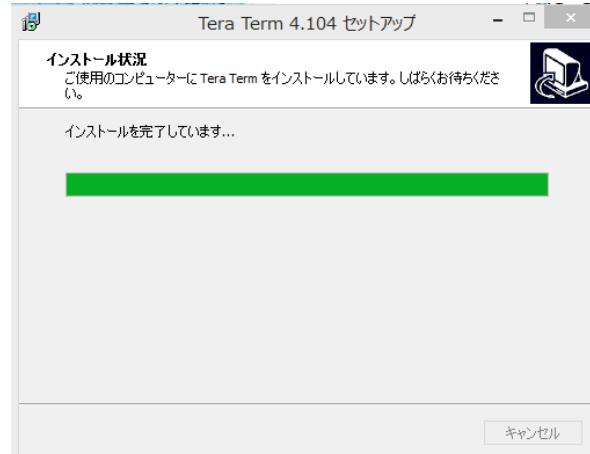
「インストール」をクリックします。



▲図 6.14 「インストール」をクリック

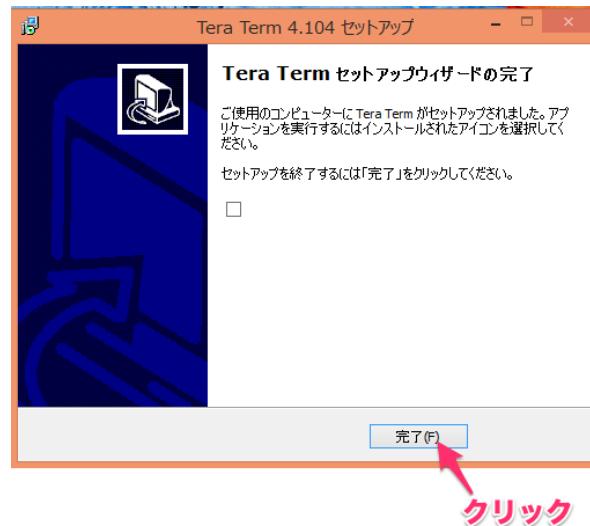
インストールが完了するのを待ちます。もし、「LogMeTT」のインストール画面が表示

されたらついでにインストールするか、そちらは「Cancel」で閉じても問題ありません。



▲図 6.15 インストール中

セットアップが完了しました。



▲図 6.16 セットアップが完了

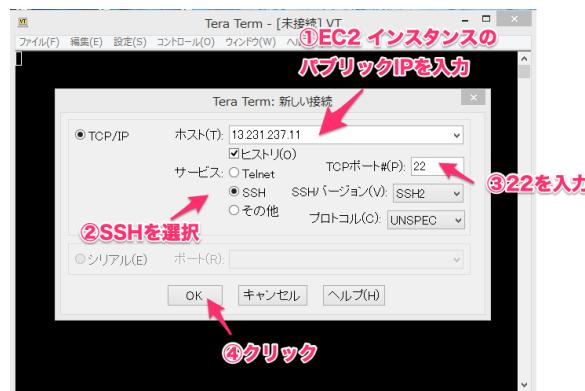
デスクトップに Tera Term のアイコンができていますので、ダブルクリックしま

しよう。



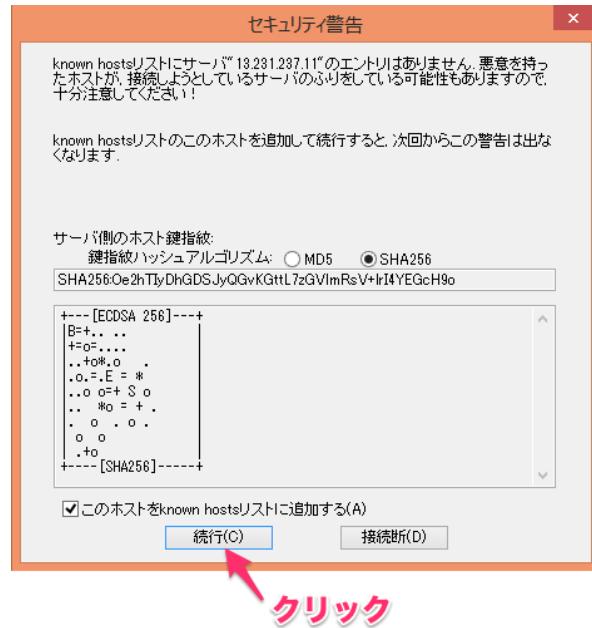
▲図 6.17 Tera Term のアイコンをダブルクリック

ホストに ap EC2 インスタンスのパブリック IP を入力し、TCP ポートは「22」サービスは「SSH」を選択し「OK」をクリックしましょう。



▲図 6.18 EC2 インスタンスに接続

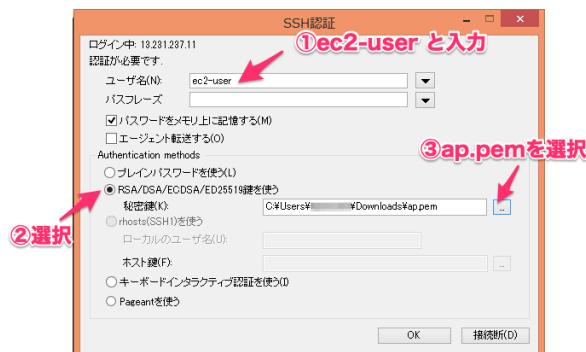
はじめて接続するサーバーには警告が表示されますので、「続行」をクリックします。



▲図 6.19 「続行」をクリック

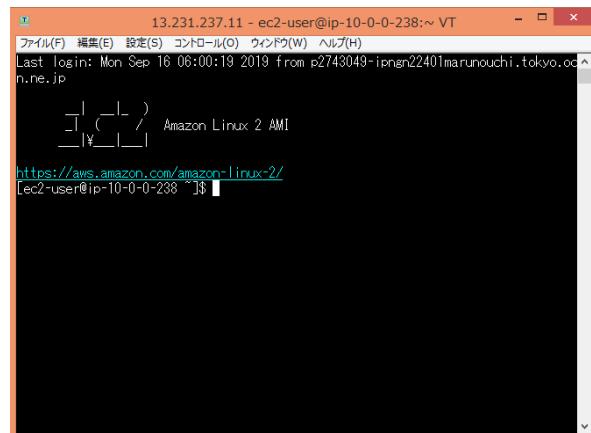
ユーザー名に「ec2-user」と入力、鍵を使う方を選択し、秘密鍵にダウンロードした「ap.pem」を選択しましょう。

「OK」をクリックします。



▲図 6.20 「OK」をクリック

無事にログインできましたね。



▲図 6.21 ログインできた

サーバーからログアウトする場合は次のコマンドです。

▼リスト 6.14 サーバーからログアウトする場合

```
1: $ exit
```

第7章

HP を立ち上げよう

7.1 python で HTTP サーバーを起動しよう

ap EC2 インスタンスにログインしていますね。Amazon Linux 2 のマシンイメージを使用している場合、python というプログラミング言語が最初から使えるようになっています。

python は楽に HTTP サーバーを起動できるコマンドがあるため、今回は python を使ってみましょう。

ターミナルで、次のコマンドを打って python が入っていることを確認します。

▼リスト 7.1 python が入っていることを確認

```
1: $ python -V  
2: Python 2.7.14
```

公開用の「/var/www/html」ディレクトリを作成し（1行目）、
「/var/www/html」ディレクトリに移動します。（2行目）

▼リスト 7.2 公開ディレクトリの作成とディレクトリ移動

```
1: $ sudo mkdir -p /var/www/html  
2: $ cd /var/www/html
```

では早速ですが、python で HTTP サーバーをポート 80 で立ち上げてみましょう。
サーバーを起動するディレクトリが公開ディレクトリ（ブラウザからアクセスできるディレクトリ）になります。

▼リスト 7.3 HTTP サーバーをポート 80 で立ち上げる

```
1: $ sudo python -m SimpleHTTPServer 80
2: Serving HTTP on 0.0.0.0 port 80 ...
```

ターミナルをそのまま放っておくと、次のようなログが 30 秒間隔くらいで出てきますね。これは、ロードバランサーからのヘルスチェックに HTTP サーバーが 200 で応答しているからです。^{*1}

また、一番左の IP アドレスがロードバランサーの IP アドレスで、2 種類あるのは、ロードバランサーが実は 2 台あるからです。^{*2}

▼リスト 7.4 ヘルスチェックに応答している

```
1: 10.0.0.39 -- [16/Sep/2019 04:42:04] "GET / HTTP/1.1" 200 -
2: 10.0.2.198 -- [16/Sep/2019 04:42:29] "GET / HTTP/1.1" 200 -
```

ブラウザから AWS マネジメントコンソールを開き、EC2 ダッシュボードを表示します。

左のメニューで「ターゲットグループ」を選択し、ターゲットタブを表示します。ターゲットグループが healthy になっていることを確認しましょう。

これは、EC2 インスタンスの 80 ポートで HTTP サーバーが起動し、ロードバランサーからのヘルスチェックに 200 で応答しているからです。

^{*1} 200 というのは、HTTP ステータスコードで OK という意味です。ステータスコードには 100 番台から 500 番台まで多くの種類があり、番号ごとに意味が決まっています。

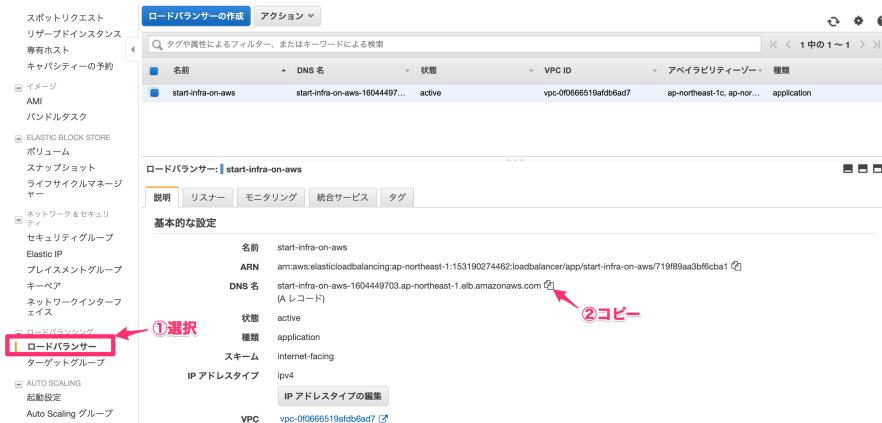
^{*2} ロードバランサーを作成する時に、アベイラビリティゾーンを 2 つ選択しましたよね。1 つのアベイラビリティゾーンが地震などで使えなくなってしまっても、もう片方のアベイラビリティゾーンにあるロードバランサーが稼働してくれます。このように、1 台使えなくなても大丈夫なように複数台準備することを「冗長化」と呼びます。

第7章 HPを立ち上げよう



▲図 7.1 ターゲットグループが healthy になっている

今度はEC2ダッシュボード左のメニューでロードバランサーを選択します。作成したロードバランサーを選択し、説明タブのDNS名をコピーしましょう。



▲図 7.2 ロードバランサーのDNS名をコピー

ブラウザを開いて、次のURLにアクセスしてみましょう。

<http://<ロードバランサーのDNS名>>

Directory listing for /

▲図 7.3 白い画面が表示された

何も置いてないので、「Directory_listing_for / 」と書かれた白い画面が表示されました。

python でサーバーを起動していたターミナルに戻って、キーボードで「Control」+「c」を押下し、サーバーを停止します。

「/var/www/html」ディレクトリ配下に、「index.html」ファイルを作成しましょう。次のコマンドを実行します。

▼リスト 7.5 index.html ファイルを作成

```
1: $ sudo vi index.html
```

キーボードで「i」を押して編集モードに変更し、次のテキストを記入します。

▼リスト 7.6 index.html ファイルを編集

```
1: hello world !
```

キーボードで「esc」を押下し、コマンドモードに切り替えます。次のコマンドを実行し、ファイルを上書き保存しましょう。

▼リスト 7.7 index.html ファイルを保存

```
1: :wq
```

ファイルが保存できたら、次のコマンドを実行してもう一度 HTTP サーバを立ち上げましょう。

▼リスト 7.8 HTTP サーバを起動

```
1: $ sudo python -m SimpleHTTPServer 80
2: Serving HTTP on 0.0.0.0 port 80 ...
```

ブラウザを開いて、また先ほどの画面を表示します。更新されない場合は、ブラウザの更新ボタンを押して画面を更新しましょう。

<http://<ロードバランサーの DNS 名>>



▲図 7.4 hello world !が表示された

hello world !が表示できましたね。

次に、HTTP サーバーをバックグラウンドで動かしましょう。今はターミナルで HTTP サーバーを起動すると、コマンドが入力できない状態で止まってしまい、少し不便です。そこで、HTTP サーバーを起動している間もターミナルがコマンド実行できる状態にしましょう。

キーボードで「Control」+「c」を押下し、サーバーを停止します。

次のコマンドで HTTP サーバーを起動します。なお、起動時には 5 行の数字（プロセス ID）が表示されます。

▼リスト 7.9 HTTP サーバーをバックグラウンドで起動

```
1: $ sudo nohup python -m SimpleHTTPServer 80 &
2: [1] <プロセス ID が出力される>
3: nohup: 入力を無視し、出力を `nohup.out` に追記します
```

Enter を押すと、またコマンドが入力できるようになります。HTTP サーバーを停止する場合、次のコマンドを実行します。

▼リスト 7.10 HTTP サーバーを停止

```
1: $ sudo kill <起動に表示されたプロセス ID>
2: [1]+ 終了                  sudo nohup python -m SimpleHTTPServer 80
```

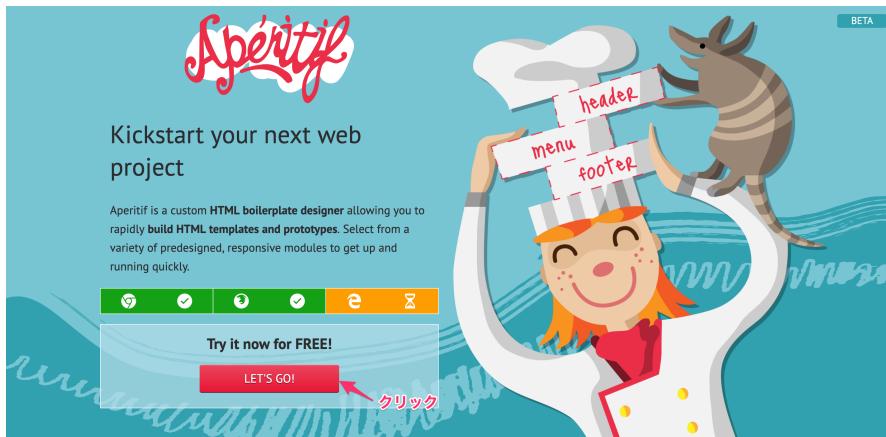
HTTP サーバーは停止しておきましょう。

7.2 サンプルのテンプレートファイルを使おう

今のままだと、hello world しか表示されないため、少し物寂しいですよね。そこで、無料で使えるテンプレートを使用して、HP ぼくしてみましょう。

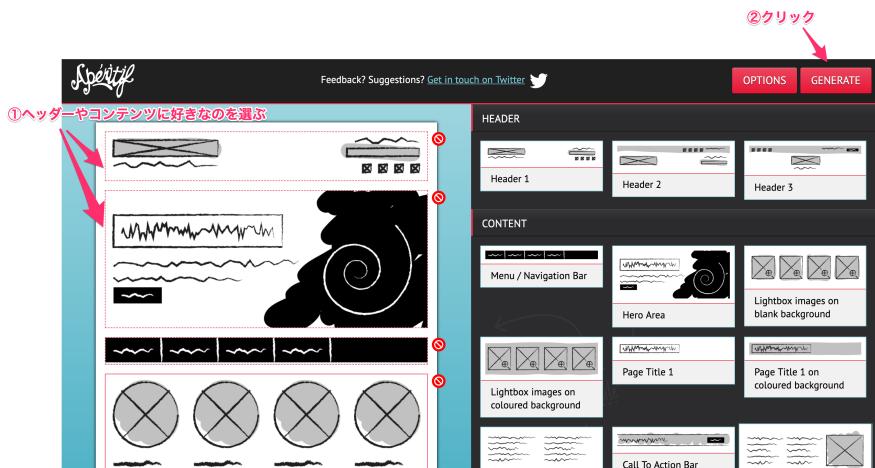
次のサイトにアクセスします

<https://aperitif.io/>



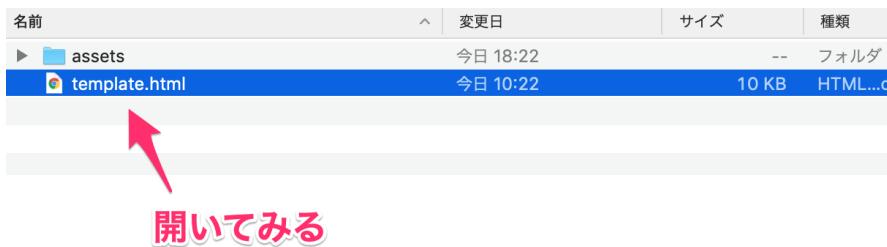
▲図 7.5 無料で使えるテンプレート作成サイト

好きなようにヘッダー・コンテンツなどを選び、「Generate」をクリックします。

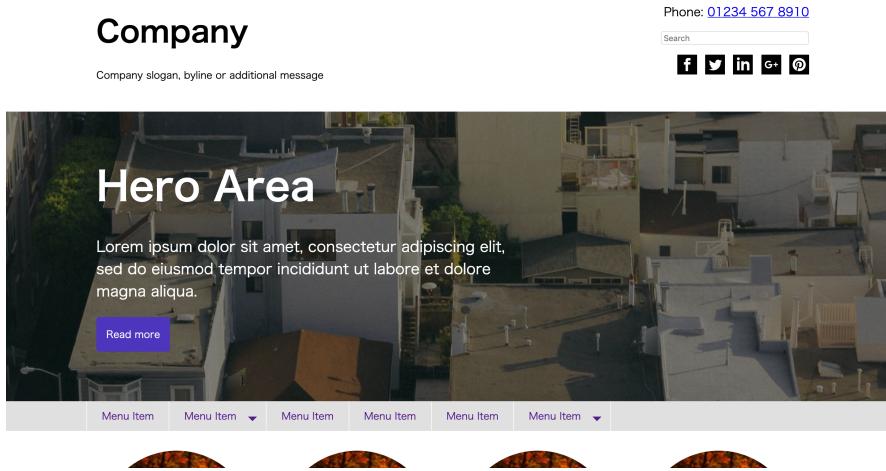


▲図 7.6 好きなコンテンツを選んで「Generate」をクリック

ダウンロードした zip ファイルを解凍し、中に入っている「template.html」をクリックして表示してみましょう。いい感じでしたか？



▲図 7.7 テンプレートファイルをダウンロード



▲図 7.8 template.html を確認

もし修正したければ、またサイトに戻って Generate しなおしてください。
いい感じのテンプレートファイルができたら、ダウンロードしたフォルダごと EC2 インスタンスにアップロードします。
新しいターミナルを開きましょう。次のコマンドで、自分の PC から ap EC2 インスタンスの「/home/ec2-user/」フォルダにテンプレートフォルダをアップロードします。
「/Users/<ユーザー名>/Downloads/」は、自分がテンプレートをダウンロードしたフォルダの場所に合わせて修正してください。

▼リスト 7.11 テンプレートフォルダをアップロード

```
1: $ scp -P 22 -r /Users/<ユーザー名>/Downloads/file ap:/home/ec2-user/
2: template.html                                         100%  9511    132.6KB/s   00:00
3: style.min.css                                         100%   19KB  274.6KB/s   00:00
4: modules.css                                          100%   28KB  596.6KB/s   00:00
5: index.js                                              100%  9662    313.9KB/s   00:00
```

アップロードできたら、ap EC2 インスタンスにログインします。

▼リスト 7.12 EC2 インスタンスにログイン

```
1: $ ssh ap
2: Last login: Thu Sep  5 07:47:43 2019 from kd106181068205.au-net.ne.jp
3:
4:      _\|_ _\|_ )
5:      _\|_ ( _\|_ ) Amazon Linux 2 AMI
```

```
6:      _\_\_|
7:
8: https://aws.amazon.com/amazon-linux-2/
9: 4 package(s) needed for security, out of 12 available
10: Run "sudo yum update" to apply all updates.
```

次のコマンドを実行し、テンプレートフォルダがアップロードされているか確認しましょう。

▼リスト 7.13 テンプレートフォルダがアップロードされているか確認

```
1: $ ls -l
2: 合計 0
3: drwx----- 3 ec2-user ec2-user 41  9月  5 09:36 file
```

アップロードされていましたね。テンプレートを公開ディレクトリ「/var/www/html/」に移動します。

▼リスト 7.14 テンプレートを移動

```
1: $ sudo mv file /var/www/html/
```

HTTP サーバーが表示する画面を先ほど作成した「hello world!」から、テンプレートに変更しましょう。

まず、公開ディレクトリに移動します。(1 行目)

先ほど作成した「hello world!」と記入した index.html を削除します。(2 行目)

file ディレクトリに移動します。(3 行目)

テンプレートのファイル名を template.html から index.html に変更します。(4 行目)

file ディレクトリ配下でサーバーを起動します。(5 行目)

▼リスト 7.15 index.html をテンプレートに変更

```
1: $ cd /var/www/html/
2: $ sudo rm index.html
3: $ cd file
4: $ mv template.html index.html
5: $ sudo nohup python -m SimpleHTTPServer 80 &
```

ブラウザで次の URL にアクセスし、ブラウザの画面を更新しましょう。

http://<ロードバランサーの DNS 名>

今度は hello world ! ではなくテンプレートで作成したホームページが表示されましたね。

7.2 サンプルのテンプレートファイルを使おう

テンプレート用の html ファイルの中身を修正すれば、自分のサイト用に修正することができますが、今回はインフラ中心で進めるため、サイトの中身のことはスルーします。

第8章

ドメインでアクセスできるようにしよう

8.1 ドメインってなに？

Google の URL を見てみましょう。

<https://www.google.com/>

google.com がドメインです。「.」で区切られた一番右側の部分「com」をトップレベルドメインと呼びます。

トップレベルドメインは何でもよいわけではなく、次のように決まっています。

8.1.1 分野別トップレベルドメイン (gTLD)

利用者の居住国に関係なく誰でも取得できるドメインです。具体的には次のような種類があります。

▼表 8.1 分野別トップレベルドメインの種類

ドメインの種類	特徴
com	企業や商用サービスを表すドメイン
net	主にネットワークサービスの提供者を表すドメイン
org	主に非営利団体を表すドメイン
biz	主にビジネスを表すドメイン
info	主に情報の提供者を表すドメイン

8.1.2 国コードトップレベルドメイン (ccTLD)

国ごと・地域ごとに割り当てられたドメインです。日本であれば「jp」が該当します。また、「jp」ドメインの中でも、トップレベルドメインの次の文字列（セカンドレベルドメイン）に指定の文字列が入り、その文字列ごとに取得可能な組織が限定されるドメインを属性型 JP ドメイン名と呼び、次のような種類に分類されます。ネットワークサービスを示す「ne.jp」を除き 1 組織 1 つしか取得できない決まりとなっています。

▼表 8.2 属性型 JP ドメインの種類

ドメインの種類	特徴
co.jp	日本国内で登記を行っている会社・企業が登録可能（例：株式会社・有限会社など）
or.jp	特定の法人組織が登録可能（例：財団法人、社団法人など）
ne.jp	ネットワークサービスごとに登録可能
ac.jp	学校教育法などの規定による学校が登録可能（例：学校法人など）
go.jp	日本国の政府機関、各省庁が管轄する研究所、特殊法人（特殊会社を除く）が登録可能

ヤフーの URL を見てみましょう。トップレベルドメインを見れば、日本国内の企業であることが分かります。

<https://www.yahoo.co.jp/>

8.1.3 サブドメイン

もう一度 yahoo の URL をみてみましょう。

<https://www.yahoo.co.jp/>

ドメインの手前に「www.」とあります。これはサブドメインといい、ドメインをさらに分割して利用できるドメインのことを指します。1 つのドメインを目的別や用途別に分けて利用する際に使用されます。yahoo の例が分かりやすいかと思います。

▼表 8.3 サブドメインの例

サービス名	URL	ドメイン名
Yahoo! JAPAN	https://www.yahoo.co.jp/	yahoo.co.jp
Yahoo!ショッピング	https://shopping.yahoo.co.jp/	shopping.yahoo.co.jp
ヤフオク	https://auctions.yahoo.co.jp/	auctions.yahoo.co.jp
Yahoo!天気	https://weather.yahoo.co.jp/	weather.yahoo.co.jp
Yahoo!ニュース	https://news.yahoo.co.jp/	news.yahoo.co.jp

8.1.4 なぜドメインにアクセスするとサーバーにアクセスできるの？

ネットワークに接続した機器同士が通信するために必要なものは何でしたか？ IP アドレスがネットワーク上の住所だと説明しましたね。

では、なぜドメインで目当てのサーバーにアクセスできるのでしょうか？

それは、このドメインはこの IP アドレス、という風にドメインとサーバーの IP アドレスが紐づけられているからです。この紐付けをしているのが「DNS サーバー」です。

8.2 ドメインを取得する

同じドメインをもった Web ページが複数存在したらどの Web ページを表示すればよいのか分からなくて困りますよね。同じドメインは世界中で登録できないように、「レジストリ」という機関で厳密に管理されています。レジストリと契約してドメインの販売を担当しているのが、「レジストラ」やその代理店です。

どこのレジストラから購入するか、どんなドメインを購入するかによって値段や安全性が異なります。たとえば、「.jp」は日本でしか取得できないため信頼性が高い分、値段も高くなります。

今回は勉強のためのドメインなので、freenom というレジストラから無料のドメインを取得することにします。

次のサイトにアクセスします。

<https://www.freenom.com/ja/index.html>

トップページで、取得したいドメイン名を入力します。

8.2 ドメインを取得する



▲図 8.1 取得したいドメイン名を入力

一覧の中から「.tk」^{*1}を選択し、「チェックアウト」をクリックしましょう。

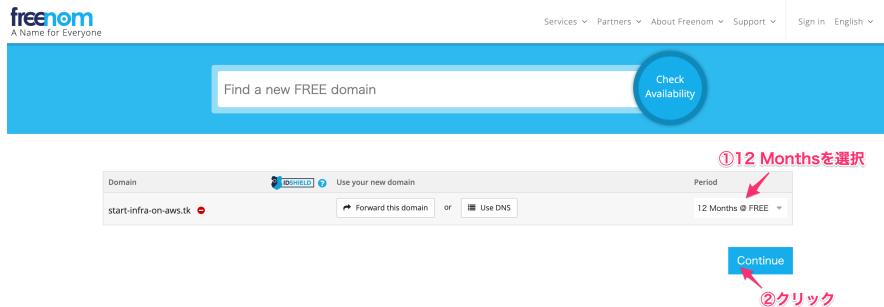


▲図 8.2 「チェックアウト」をクリック

*1 .tk は国別コードトップレベルドメイン (ccTLD) のひとつで、オーストララシアにあるニュージーランドの領土であるトケラウに割り当てられています。登録してから 1 年間は無料で利用できるのですが、アクセスがなく放置されたドメインは無効化され、広告を表示することによって、利益を得ている仕組みです。90 日間に 25 回のアクセスがないと自動的に広告に置き換わります。今回はあくまで勉強用なので、こちらのドメインを使用します。

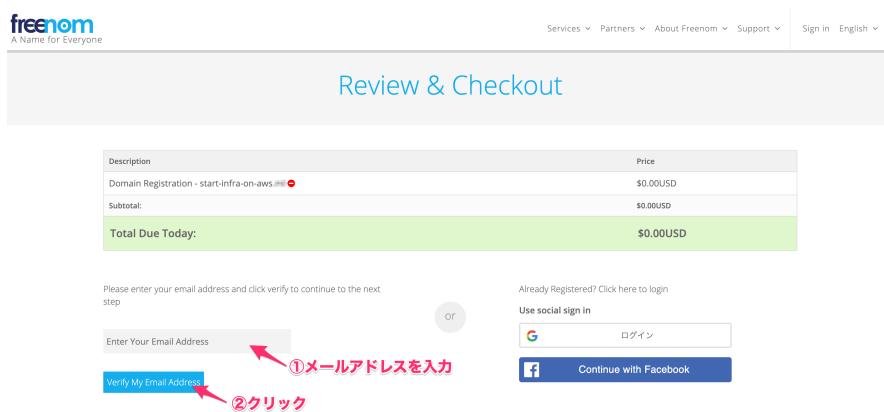
第8章 ドメインでアクセスできるようにしよう

Period はプルダウンから無料で使用できる最長の「12 Months」を選択し、「Continue」をクリックします。



▲図 8.3 「Continue」をクリック

メールアドレスを入力し、「Verify My Email Address」をクリックします。



▲図 8.4 メールアドレスの確認

確認メールが送信されたため、メールを確認しましょう。

8.2 ドメインを取得する

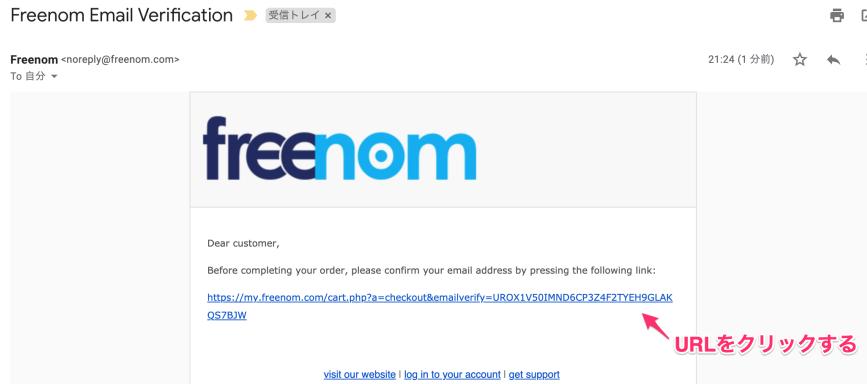
The screenshot shows the 'Review & Checkout' section of the freenom website. At the top, there's a navigation bar with links for Services, Partners, About freenom, Support, Sign in, and English. Below the navigation is a summary table:

Description	Price
Domain Registration - start-infra-on-aws.tk	\$0.00USD
Subtotal:	\$0.00USD
Total Due Today:	\$0.00USD

Below the table, a message states: "Verification link sent to your email (mailto:...). The link is valid for only 24 hours. Go to your email inbox and click on the link." A red arrow points to this message. To the right of the message, the text "メールが送信されたため、メールを確認する" (Email was sent, check the email) is written in pink.

▲図 8.5 確認メールが送信された

freenom から送信されたメールを開き、URL をクリックします。



▲図 8.6 メールの URL をクリック

ドメイン取得者の情報を入力する必要があります。会社でなく個人で取得する場合、住所などは個人情報になりますので、自分で判断して入力してください。^{*2}

^{*2} レジストリやレジストラがドメイン名の登録者などに関する情報を、whois というサービスでインターネットユーザーが誰でも参照できるサービスとして提供しています。これにはドメイン名登録者の名前および住所も含まれ、ネットワークの安定的運用を実施する上で、何か問題が起きた時に連絡する目的で公開されます。今回 freenom で入手したドメインは freenom から無料で貸与されている扱いなので、freenom の情報 (TK ドメインだと Teletok 社の情報) が公開されています。取得時に入力した個人情報は公開されないので安心です。

第8章 ドメインでアクセスできるようにしよう

The screenshot shows the 'Review & Checkout' page of the freenom website. At the top, there's a navigation bar with links for Services, Partners, About Freenom, Support, Sign in, and English. The main title 'Review & Checkout' is centered above a table.

Description	Price
Domain Registration - start-infra-on-aws.tk	\$0.00USD
Subtotal:	\$0.00USD
Total Due Today:	\$0.00USD

Below the table, there's a section titled 'Your Details' with input fields for First Name, Last Name, Company Name, Address 1, and Zip Code. To the right of these fields is a reCAPTCHA verification box with the text 'reCAPTCHA で保護されています' and 'プライバシーと利用規約'.

▲図 8.7 ドメイン取得者の情報を入力

「I have read and agree to the Terms & Conditions」にチェックを入れたら「Complete Order」をクリックします。

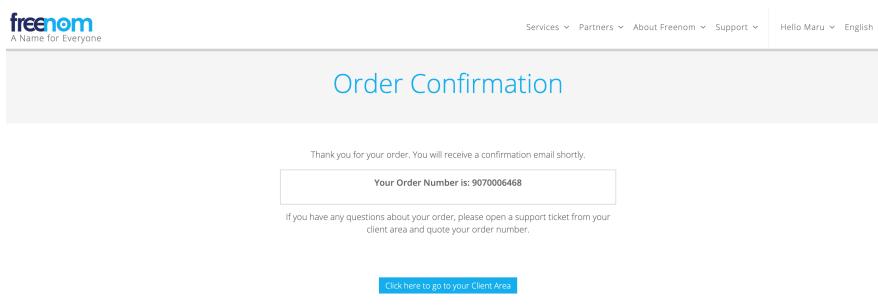
8.2 ドメインを取得する

First Name	Maru
Last Name	[REDACTED]
Company Name	[REDACTED]
Address 1	Oizumi
Zip Code	[REDACTED]
City	Nasima
Country	[REDACTED]
State/Region	[REDACTED]
Phone Number	+81
Email Address	maru.maru@gmail.com
<input type="button" value="Change"/>	
Password	*****
Confirm Password	*****

Tax may be charged depending upon the state and country selections you make. Click to recalculate after making ②クリック your choices. ①チェック I have read and agree to the Terms & Conditions ②

▲図 8.8 「Complete Order」をクリック

ドメイン取得の申し込みが完了しました。



▲図 8.9 ドメイン取得の申し込みが完了

8.3 Route53 にドメインを登録してロードバランサーに紐付けよう

Route53 は、AWS が提供する DNS サーバーです。

AWS マネジメントコンソールで「Route53」を検索しましょう。



▲図 8.10 「Route53」を検索

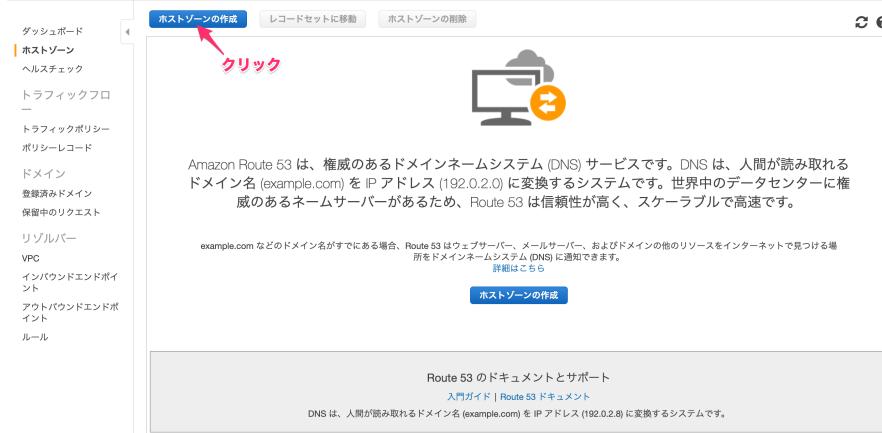
「DNS 管理」にある「今すぐ始める」をクリックしましょう。



▲図 8.11 DNS 管理を今すぐ始める

Route53 のダッシュボードで「ホストゾーンの作成」をクリックします。

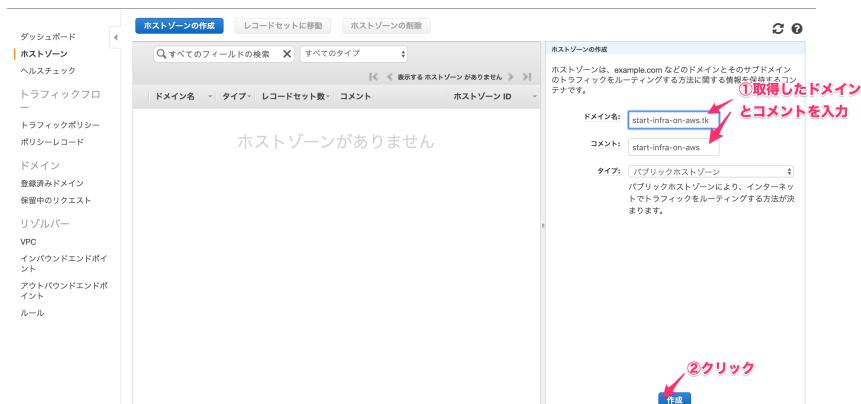
8.3 Route53 にドメインを登録してロードバランサーに紐付けよう



▲図 8.12 「ホストゾーンの作成」をクリック

右側の「ホストゾーンの作成」で、ドメイン名に取得したドメイン名を、コメントに「start-infra-on-aws」と入力しましょう。

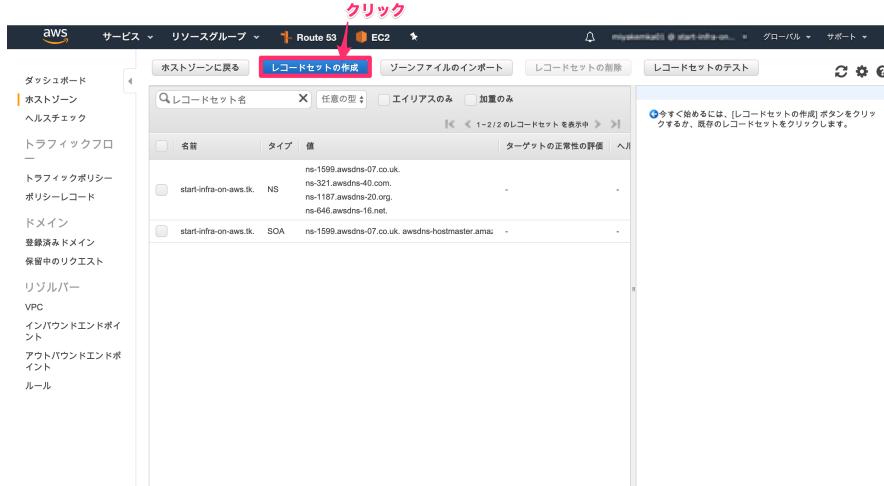
「作成」をクリックします。



▲図 8.13 「作成」をクリック

「レコードセットの作成」をクリックします。

第8章 ドメインでアクセスできるようにしよう



▲図 8.14 「レコードセットの作成」をクリック

名前に「www」を入力し、エイリアスは「はい」を選択、エイリアス先に「start-infra-on-aws」という名前のロードバランサーを選択します。

AWS のアカウント ID のエイリアスを設定した時にも説明しましたが、エイリアスとは、別名のことでしたね。つまり、「www.<取得したドメイン名>」の別名が「start-infra-on-aws」のロードバランサーになります。「www.<取得したドメイン名>」にアクセスすると、「start-infra-on-aws」のロードバランサーにアクセスします。

「作成」をクリックします。



▲図 8.15 エイリアスを作成

8.4 ドメインのネームサーバーを変更する

エイリアスが作成されました。

The screenshot shows the AWS Route 53 console. On the left, there's a navigation pane with options like 'ダッシュボード', 'ホストゾーン', 'ヘルスチェック', etc. The main area is titled 'レコードセットの作成' (Create Record Set) and shows a table of records. A red arrow points to the last row, which is 'www.start-infra-on-aws.tk.' of type 'A' pointing to 'ALIAS dualstack.start-infra-on-aws-1604449703.ap...'. A red box highlights this row with the text 'エイリアスが作成された' (Alias created).

名前	タイプ	値	ターゲットの正常性の評価
start-infra-on-aws.tk.	NS	ns-1599.awsdns-07.co.uk. ns-321.awsdns-40.com. ns-1187.awsdns-20.org. ns-446.awsdns-16.net.	-
start-infra-on-aws.tk.	SOA	ns-1599.awsdns-07.co.uk. awsdns-hostmaster.amazon. ns-1599.awsdns-07.co.uk. awsdns-hostmaster.amazon. ns-1599.awsdns-07.co.uk. awsdns-hostmaster.amazon.	-
www.start-infra-on-aws.tk.	A	ALIAS dualstack.start-infra-on-aws-1604449703.ap...	いいえ

▲図 8.16 エイリアスが作成された

取得したドメインに置き換えて、次の URL にアクセスしてみましょう。

<http://www.<取得したドメイン>>

残念、何も表示されませんね。

8.4 ドメインのネームサーバーを変更する

ドメイン名と IP アドレスを紐付ける情報を、どの DNS サーバーに登録するかを指定するのが「ネームサーバー情報」です。先ほど AWS の Route53 にドメインとエイリアスを登録しましたが、今のままだとネームサーバーが freenom のネームサーバーになっており、先ほどのドメインにアクセスすると AWS の Route53 ではなく freenom のネームサーバーに聞きに行ってしまいます。

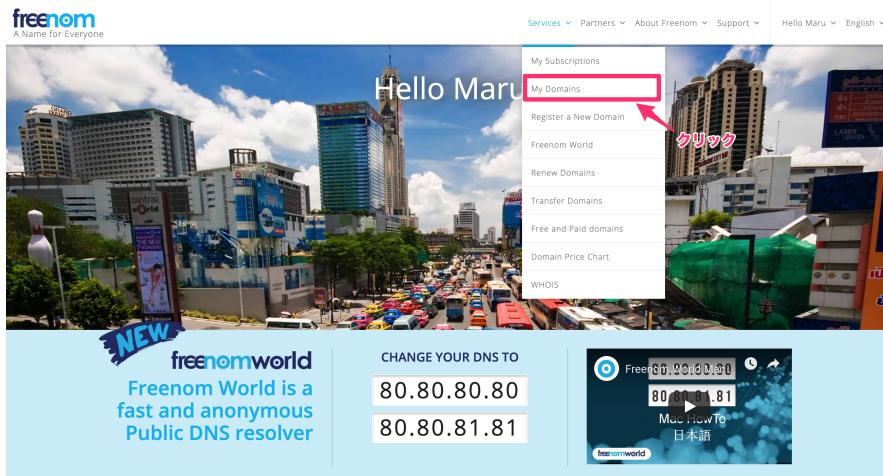
ネームサーバーを AWS の Route53 に変更する必要がありますね。

freenom にログインします。

<https://www.freenom.com/ja/index.html>

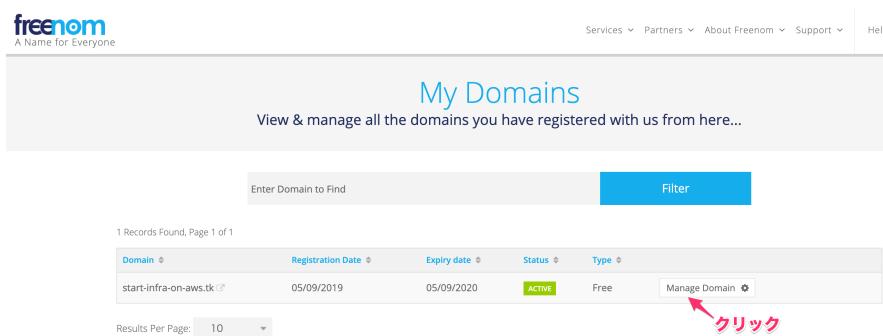
ログイン後、「Services」から「My Domains」を選択します。

第8章 ドメインでアクセスできるようにしよう



▲図 8.17 「My Domains」を選択

取得したドメインが表示されたら、「Manage Domain」をクリックします。



▲図 8.18 「Manage Domain」をクリック

「Management Tools」 タブを表示し、「Nameservers」を選択します。

8.4 ドメインのネームサーバーを変更する

The screenshot shows the freenom domain management interface for the domain 'start-infra-on-aws.tk'. The top navigation bar includes 'Services', 'Partners', and 'At'. Below the navigation is a sub-menu for 'Management Tools' with options like 'Information', 'Upgrade', 'Management Tools', and 'Manage Freenom DNS'. A red arrow labeled ① 選択 points to the 'Management Tools' dropdown. Another red arrow labeled ② 選択 points to the 'Nameservers' option in the dropdown menu. The main content area is titled 'Managing start-infra-on-aws.tk' and contains sections for 'Information', 'Nameservers', 'Register glue records', 'URL Forwarding', 'Dot TK Apps', 'Dot TK Ambassadors', and 'Cancel domain'. The 'Nameservers' section is currently selected.

▲図 8.19 「Nameservers」を選択

AWS マネジメントコンソールの Route53 ダッシュボードを表示します。
タイプ「NS」と表示された列があり、4 つネームサーバーが登録されています。

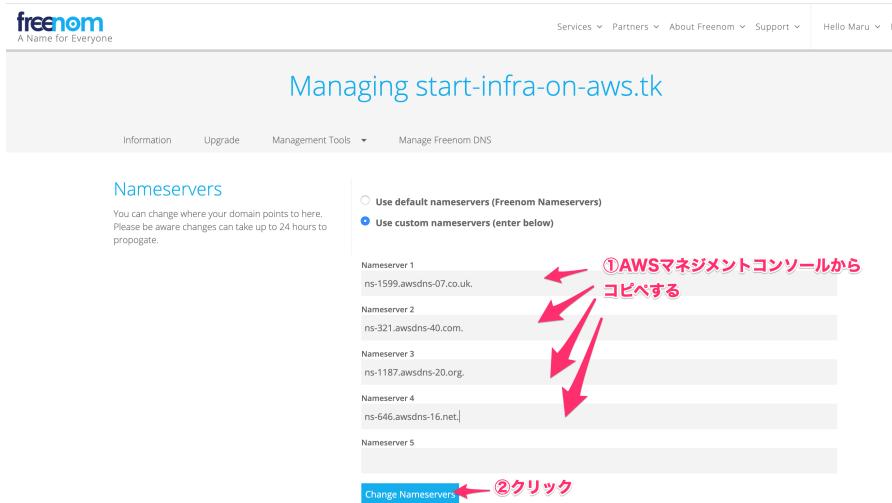
The screenshot shows the AWS Route53 Record Set configuration screen for the domain 'start-infra-on-aws.tk'. The left sidebar lists services like Hosted Zones, Lambda, VPC, and CloudFront. The main panel shows a table of record sets:

名前	タイプ	値
start-infra-on-aws.tk.	NS	ns-1599.awsdns-07.co.uk. ns-321.awsdns-40.com. ns-1187.awsdns-20.org. ns-648.awsdns-16.net.
start-infra-on-aws.tk.	SOA	ns-1599.awsdns-07.co.uk. awsdns-hostmaster.awsdns-07.co.uk.
www.start-infra-on-aws.tk.	A	ALIAS dualstack.start-infra-on-aws-1604449703.ap-southeast-2.amazonaws.com.

A red box highlights the 'ns-1599.awsdns-07.co.uk.' value under the first NS record. A pink annotation with a red arrow points to this value, stating '1つずつコピペして freenom のサイトのNameserver1～4に入力していく' (Copy and paste one by one into the Nameserver 1~4 input fields on the freenom site).

▲図 8.20 Route53 のレコードセットを確認

freenom のネームサーバーの設定ページで「Use custom nameservers (enter below)」を選択し、Route53 に登録されている 4 つのネームサーバーをコピペします。
4 つともすべて貼り付けたら「Change Nameservers」をクリックします。



▲図 8.21 ネームサーバーを Route53 に変更

今度こそ、取得したドメインでアクセスしてみましょう。

<http://www.<取得したドメイン>>

無事、HP が表示されましたね。

第9章

HTTPS でアクセスできるようにしよう

9.1 HTTP と HTTPS の違い

何度か説明しましたが、HTTP や HTTPS は通信プロトコルでしたね。

Internet Explorer や Google Chrome 等のブラウザを使用して Web サイトにアクセスし、閲覧したいページや画像、動画などをサーバに要求して、内容に応じてサーバがレスポンスを返します。レスポンスを受け取ったブラウザは画面上にページや画像、動画などを表示します。

HTTP は、この一連の流れを暗号化されていない状態で行います。

先ほど作成した HP を見てみましょう。ブラウザに Google chrom を使用している場合、URL の左側に「保護されていない通信」と出ていますよね。



▲図 9.1 「保護されていない通信」と表示されている

暗号化されていない状態だと、通信経路で不正に盗聴される可能性があります。個人情報を不正に取得されるかもしれません。これを防ぐのが HTTPS です。

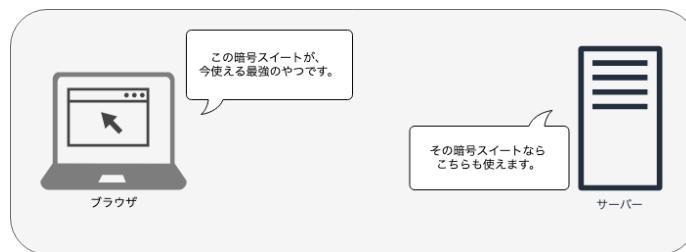
HTTPS は SSL/TLS を利用した HTTP 通信です。SSL/TLS は通信を暗号化するので、安全に情報のやりとりを行うことができます。万が一通信経路から情報が抜き取られたとしても、暗号化されているので中身を解読するのは困難です。

9.2 SSL/TLS 通信の仕組み

SSL/TLS 通信の流れを見ていきましょう。

9.2.1 暗号スイートの合意

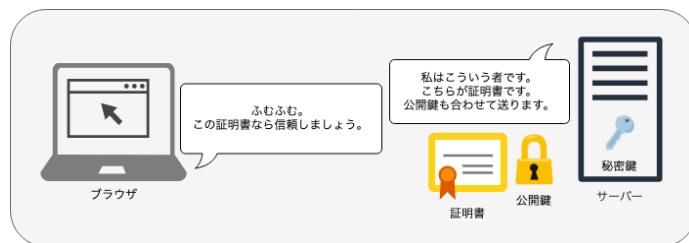
せっかく暗号化するのであれば、不正に抜き取られても解読が難しい安全な暗号技術を使いたいですよね。SSL/TLS 上ではいくつかの暗号化技術が使われるのですが、使用される暗号技術の組のことを暗号スイートと呼びます。まずは、ブラウザとサーバ間の両者が使用できる一番安全な SSL/TLS 暗号化技術の組み合わせ（暗号スイート）を決めます。



▲図 9.2 暗号スイートの合意

9.2.2 デジタル証明書と公開鍵の提示

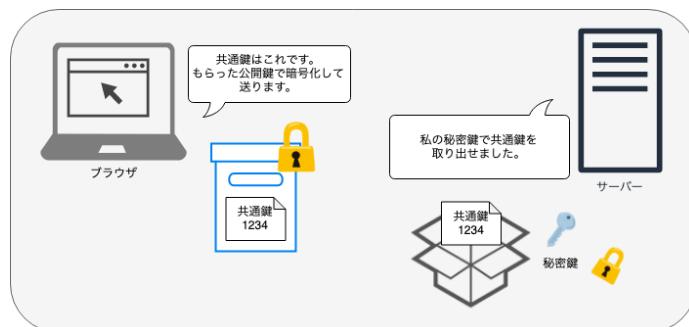
サーバはブラウザへデジタル証明書（SSL サーバ証明書ともいいます）と公開鍵を送信します。デジタル証明書は認証局（CA:Certificate Authority）が発行するもので、ブラウザは通信相手が送ってきた証明書が信頼できる認証局のものであることを確認します。（ブラウザには主要な認証局の証明書がプリインストールされているので、それを使って確認します。）



▲図 9.3 デジタル証明書と公開鍵の提示

9.2.3 共通鍵の元データ交換・共通鍵の生成

ブラウザは共通鍵をサーバからもらった公開鍵で暗号化してサーバへ送信します。



▲図 9.4 共通鍵の元データ交換・共通鍵の生成

9.2.4 暗号化通信開始

以降、共通鍵を使ったデータの SSL 暗号化通信が開始されます。



▲図 9.5 暗号化通信開始

このブラウザとの一連のやり取りをロードバランサーでできるため、これから設定していきましょう。

9.3 デジタル証明書を発行しよう

HTTPS 通信に必要なデジタル証明書を発行しましょう。

AWS の Certificate Manager (ACM) で無料でデジタル証明書を発行することができます。

AWS マネジメントコンソールで Certificate Management を検索しましょう。



▲図 9.6 Certificate Management を検索

「証明書のプロビジョニング」の「今すぐ始める」をクリックします。

9.3 デジタル証明書を発行しよう



▲図 9.7 証明書のプロビジョニングを今すぐ始める

「パブリック証明書のリクエスト」を選択し、「証明書のリクエスト」をクリックします。



▲図 9.8 証明書のリクエスト

取得したドメインと、取得したドメインの手前に「*.」をつけたドメインを登録します。著者の場合は、「start-infra-on-aws.tk」と「*.start-infra-on-aws.tk」を登録します。

*（アスタリスク）はワイルドカードで「www.start-infra-on-aws.tk」や「hoge.start-infra-on-aws.tk」など、すべてのサブドメインが対象となります。

「次へ」をクリックします。

第9章 HTTPSでアクセスできるようにしよう



▲図9.9 ドメインを登録

検証方法は「DNSの検証」を選択し、「確認」をクリックします。



▲図9.10 DNSの検証を選択

証明書のリクエスト内容を確認し、「確定とリクエスト」をクリックします。

9.3 デジタル証明書を発行しよう



▲図 9.11 「確定とリクエスト」をクリック

証明書のリクエストが完了しましたが、まだ検証保留中の状態です。

AWS が証明書を発行する際には、リクエストしたドメイン（著者の場合は start-infra-on-aws.tk）の所有者が本当に証明書をリクエストした人なのか？を検証する必要があります。

先ほど、検証方法は「DNS の検証」を選択しましたね。

Route53 に登録した start-infra-on-aws.tk に対して、CNAME レコード（ドメインの別名）を作成することで、著者が本当に「start-infra-on-aws.tk」のドメインを所有していることを証明できます。

ドメイン名をクリックして、詳細レコードを表示しましょう。



▲図 9.12 詳細レコードを表示

第9章 HTTPSでアクセスできるようにしよう

「Route53でのレコード作成」をクリックします。なお、「Route53でのレコード作成」ボタンが存在しない場合のフローは後ほど説明します。



▲図 9.13 「Route53でのレコード作成」をクリック

作成される DNS レコード内容が表示されるため、「作成」をクリックします。



▲図 9.14 「作成」をクリック

サブドメインの方も同じく「Route53でのレコード作成」を実施しましょう。

9.3 デジタル証明書を発行しよう

証明書のリクエスト

ステップ1: ドメイン名の追加
ステップ2: 検証方法の選択
ステップ3: 検証とリクエスト
ステップ4: 検証

❶ **進行中のリクエスト**
証明書のリクエストが作成されました。その状況は検証保留中になっています。証明書の検証と承認を完了するには、追加のアクションが必要です。

検証

以下に示すドメインごとに、DNS 設定で CNAME レコードを作成します。AWS Certificate Manager (ACM) が証明書を発行する前にこのステップを完了する必要がありますが、ここでは「続行」をクリックしてこのステップをスキップすることができます。後でこのステップに戻るには、ACM コンソールで証明書リクエストを開きます。

ドメイン	検証状態
start-infra-on-aws.tk	検証保留中
*.start-infra-on-aws.tk	検証保留中

ドメインの DNS 設定に次の CNAME レコードを追加します。CNAME レコードを追加する手順は、お使いの DNS サービスプロバイダによって異なります。[詳細はこちちら](#)。

名前	タイプ	値
_d5abffba089b4de35653449357fea245.start-infra-on-aws.tk.	CNAME	_7cece2d7ae288713afe547db7b849e.oprtlswtu.acm-validations.aws

注意: DNS 設定を変更すると、ACM は、DNS レコードが存在する限り、このドメイン名用の証明書を発行できるようになります。アクセス許可は、レコードを削除することでいつでも取り消すことができます。[詳細はこちちら](#)。

Route 53 でのレコードの作成 Amazon Route 53 DNS のお客様 ACM はお客様の DNS 設定を更新できます。[詳細はこちちら](#).

クリック

▲図 9.15 Route53 でのレコード作成を実施

DNS レコードの作成が完了しましたね。

ドメインの DNS 設定に次の CNAME レコードを追加します。CNAME レコードを追加する手順は、お使いの DNS サービスプロバイダによって異なります。[詳細はこちちら](#)。

名前	タイプ	値
_d5abffba089b4de35653449357fea245.start-infra-on-aws.tk.	CNAME	_7cece2d7ae288713afe547db7b849e.oprtlswtu.acm-validations.aws

注意: DNS 設定を変更すると、ACM は、DNS レコードが存在する限り、このドメイン名用の証明書を発行できるようになります。アクセス許可は、レコードを削除することでいつでも取り消すことができます。[詳細はこちちら](#)。

Route 53 でのレコードの作成 Amazon Route 53 DNS のお客様 ACM はお客様の DNS 設定を更新できます。[詳細はこちちら](#).

成功
DNS レコードは Route 53 ホストゾーンに書き込まれました。変更が反映され、AWS がドメインを検証するまでに最大で 30 分かかる場合があります。

DNSレコードの作成が成功した

***.start-infra-on-aws.tk** 検証保留中

Route 53 でのレコードの作成 Amazon Route 53 DNS のお客様 ACM はお客様の DNS 設定を更新できます。[詳細はこちちら](#).

▲DNS 設定をファイルにエクスポート すべての CNAME レコードをファイルにエクスポート

▲図 9.16 DNS レコードの作成が完了

「続行」をクリックします。

第9章 HTTPSでアクセスできるようにしよう

The screenshot shows the 'DNS Settings' section of the AWS Certificate Manager. It displays a table of CNAME records for the domain `*.start-infra-on-aws.tk`. A green success message box at the top indicates that the DNS record was successfully created in Route 53. Another message box below it says 'Route 53 でのレコードの作成' (Record creation in Route 53) and 'Amazon Route 53 DNS のお客様 ACM はお客様の DNS 設定を更新できます' (Customer ACM updates your customer's DNS settings). A pink arrow points to the blue 'Continue' button at the bottom right.

▲図 9.17 「続行」をクリック

証明書一覧を確認すると、状況が「発行済み」となっています。証明書が発行できました。

The screenshot shows the 'Certificates' section of the AWS Certificate Manager. It lists two certificates: `start-infra-on-aws.tk` and `*.start-infra-on-aws.tk`, both of which are marked as 'Issued' (発行済み). A pink arrow points to the status column for the first certificate. Below the list, there is a detailed view of the certificate for `start-infra-on-aws.tk`, showing its status as 'Issued' (成功), the request date (2019-09-07T01:46:20 UTC), and the expiration date (2019-09-07T01:48:11 UTC).

▲図 9.18 証明書が「発行済み」となった

9.3 デジタル証明書を発行しよう

ステップ4：検証の画面で「Route53でのレコード作成」ボタンが表示されていない場合、自力でDNSレコード追加する必要があります。

証明書のリクエスト

ステップ1: ドメイン名の追加
ステップ2: 検証方法の選択
ステップ3: 検証とリクエスト
ステップ4: 検証

① 進行中のリクエスト
証明書のリクエストが作成されました。その状況は検証保留中になっています。証明書の検証と承認を完了するには、追加のアクションが必要です。

検証状態
ドメイン 検証保留中
hoge.tk

ドメインのDNS設定に次のCNAMEレコードを追加します。CNAMEレコードを追加する手順は、お使いのDNSサービスプロバイダによって異なります。[詳細はこちる。](#)

名前	タイプ	値
_d55247aa17c44f915ba0cd576040c4ad.hoge.tk.	CNAME	_1d2ac06e77cc25edd7cf00405d4d09c.olprtswtu.acm-validations.aws.

注意: DNS設定を変更すると、ACMは、DNSレコードが存在する限り、このドメイン名用の証明書を発行できるようになります。アクセス許可は、レコードを削除することでいつでも取り消すことができます。[詳細はこちる。](#)

[DNS設定をファイルにエクスポート](#) すべてのCNAMEレコードをファイルにエクスポート

「Route53でのレコードの作成」ボタンが表示されない場合

続行



▲図 9.19 「Route53でのレコードの作成」ボタンが表示されていない場合

まずは、「DNS設定をファイルにエクスポート」リンクをクリックしましょう。

証明書のリクエスト

ステップ1: ドメイン名の追加
ステップ2: 検証方法の選択
ステップ3: 検証とリクエスト
ステップ4: 検証

① 進行中のリクエスト
証明書のリクエストが作成されました。その状況は検証保留中になっています。証明書の検証と承認を完了するには、追加のアクションが必要があります。

検証状態
ドメイン 検証保留中
hoge.tk

ドメインのDNS設定に次のCNAMEレコードを追加します。CNAMEレコードを追加する手順は、お使いのDNSサービスプロバイダによって異なります。[詳細はこちる。](#)

名前	タイプ	値
_d55247aa17c44f915ba0cd576040c4ad.hoge.tk.	CNAME	_1d2ac06e77cc25edd7cf00405d4d09c.olprtswtu.acm-validations.aws.

注意: DNS設定を変更すると、ACMは、DNSレコードが存在する限り、このドメイン名用の証明書を発行できるようになります。アクセス許可は、レコードを削除することでいつでも取り消すことができます。[詳細はこちる。](#)

[DNS設定をファイルにエクスポート](#) すべてのCNAMEレコードをファイルにエクスポート

クリック

続行



▲図 9.20 「DNS設定をファイルにエクスポート」リンクをクリック

DNS設定がダウンロードできました。このDNSレコードをRoute53で作成する必要

第9章 HTTPSでアクセスできるようにしよう

があります。

Domain Name	Record Name	Record Type	Record Value
hoge.tk	_d55247aa17c44f915ba0cd576040c4ad.hoge.tk.	CNAME	_1d2ac06e77cc25edd7cf0ee0405d4d09c.olprtlswtu.acm-validations.aws.

▲図 9.21 DNS 設定をダウンロード

Route53 のダッシュボードを開いたら、「ホストゾーン」をクリックします。



1 クリック

▲図 9.22 Route53 で「ホストゾーン」をクリック

対象のドメインを選択し、「レコードセットに移動」をクリックします。

9.3 デジタル証明書を発行しよう

クリック

▲図 9.23 「レコードセットに移動」をクリック

先ほどダウンロードした DNS 設定のレコードセットを作成します。名前に「Record Name」の文字列（サブドメインの部分のみ）を入力します。タイプは「CNAME」を選択、値に「Record Value」の文字列を入力します。

「作成」をクリックします。

①クリック

②名前を入力

③CNAMEを選択

④値を入力

⑤クリック

▲図 9.24 レコードセットを作成

CNAME のレコードが作成できました。

少し時間がたつと、Certificate Management のドメイン検証が成功するはずです。



▲図 9.25 CNAME のレコードが作成できた

9.4 ロードバランサーで HTTPS の設定をしよう

作成した証明書をロードバランサーに登録します。ロードバランサーは EC2 のダッシュボードでしたね。

EC2 のダッシュボードを開いたら、左のメニューで「ロードバランサー」を選択します。「start-infra-on-aws」のロードバランサーを選択し、「リスナー」タブを表示し、「リスナーの追加」をクリックします。



▲図 9.26 「リスナーの追加」をクリック

プロトコルは「HTTPS」を選択し、「アクションの追加」から「転送先...」を選択しま

9.4 ロードバランサーで HTTPS の設定をしよう

しよう。



▲図 9.27 転送先を選択

転送先に ap を選択します。



▲図 9.28 ap を選択

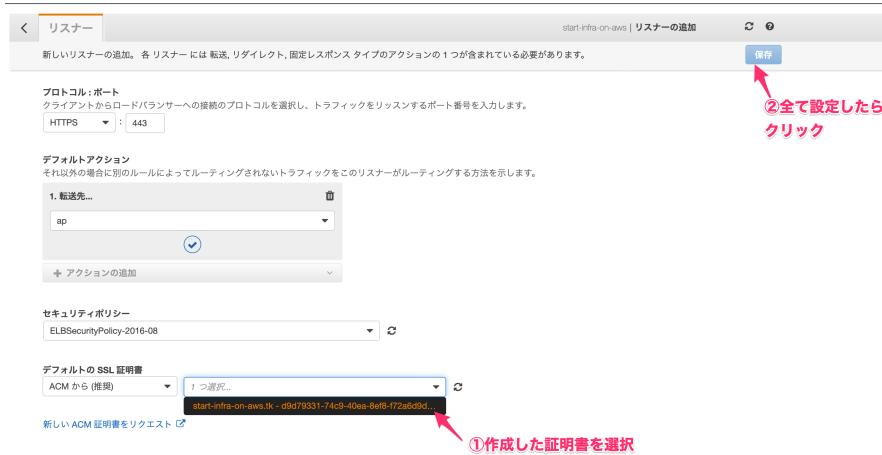
セキュリティポリシーのプルダウンを押してみるとたくさんのポリシーが表示されますね。ポリシーごとに、利用可能な暗号スイートが決まっています。

今回はデフォルト設定のままで OK です。

デフォルトの SSL 証明書は、先ほど作成した証明書を選択しましょう。

設定したら、一番上にある「保存」をクリックします。

第9章 HTTPS でアクセスできるようにしよう



▲図 9.29 「保存」をクリック

HTTPS 用のリスナーが作成できました。



▲図 9.30 HTTPS リスナーが作成できた

ロードバランサーのリスナータブで、リスナー一覧を表示します。

HTTP のリスナーはもう不要なので削除しましょう。HTTP のリスナーを選択し、「削除」をクリックします。

9.4 ロードバランサーで HTTPS の設定をしよう



▲図 9.31 HTTP のリスナーを削除

本当に削除してよいのか確認画面が出るので、「はい、削除する」をクリックします。



▲図 9.32 はい、削除する」をクリック

HTTPS のリスナーを見ると、警告マークがでています。クリックして確認するとわかりますが、HTTPS の通信をするためには、ロードバランサーのセキュリティグループで HTTPS 通信を許可する必要があります。

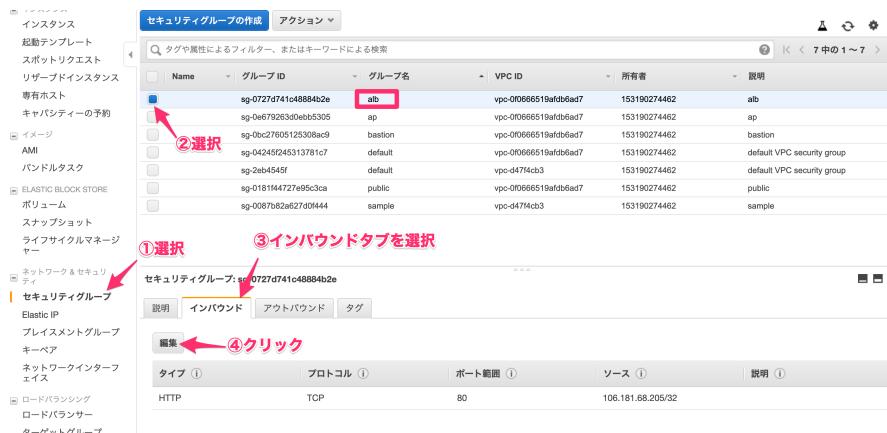
第9章 HTTPS でアクセスできるようにしよう



▲図 9.33 HTTPS のリスナーに警告マークが出ている

同じ EC2 ダッシュボードで、左のメニューから「セキュリティグループ」を選択します。

alb のセキュリティグループを選択し、「インバウンド」タブで「編集」をクリックしましょう。



▲図 9.34 alb のセキュリティグループを編集

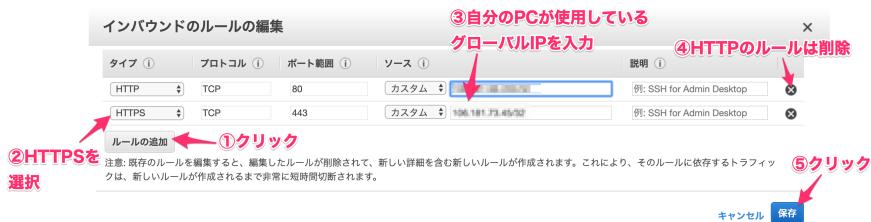
HTTPS のレコードを追加します。

9.4 ロードバランサーで HTTPS の設定をしよう

タイプ「HTTPS」を選択、ポートは「443」、ソースは自分が使用しているグローバルIPをCIDR表記（/32）で入力します。

不要になったHTTPのレコードは削除しましょう。

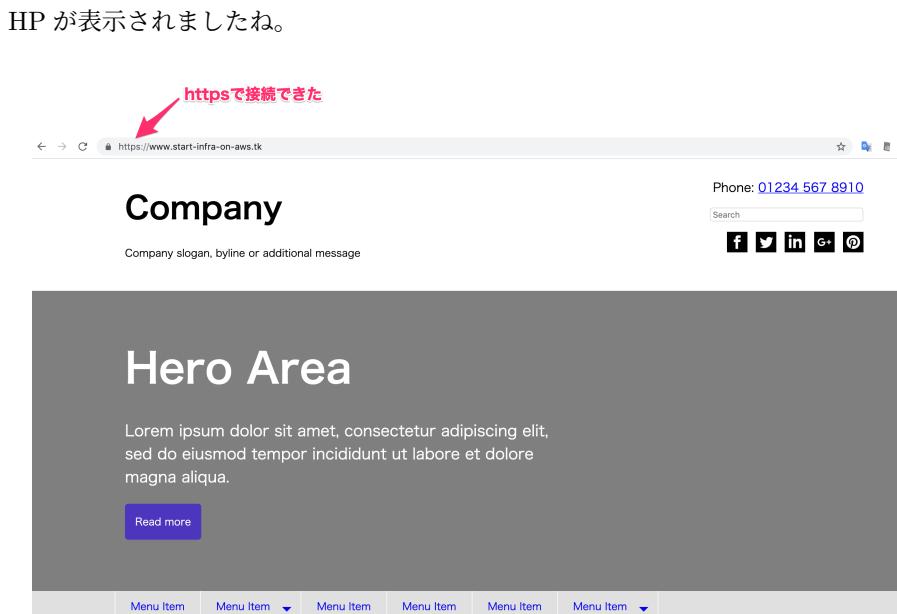
修正したら、「保存」をクリックします。



▲図 9.35 HTTPS の通信を許可して保存

ブラウザから、URLをhttpsに変更して接続してみましょう。

<https://www.start-infra-on-aws.tk>



▲図 9.36 HP が表示された

これにて終了になります！お疲れ様でした！

勉強後は、作成したEC2インスタンスやALB、Route53のホストゾーンなどは無駄なコストがかからないように削除しておきましょう。AWSのマネジメントコンソールで、作成した対象を選択し、「アクション」から削除できます。

AWSでインフラことはじめ

2019年9月22日 技術書典7版 v1.0.0

著 者 maki
編 集 maki
発行所 ひよこ開発室出版部

(C) 2019 ひよこ開発室出版部