

SSL をはじめよう

証明書の発行からトラブルシューティングまで

mochikoAsTech 著

2020-03-01 版 mochikoAsTech 発行

はじめに

2020 年 2 月 mochikoAsTech

この本を手に取ってくださったあなた、はじめてまして。「SSL をはじめよう」の筆者、mochikoAsTech です。

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。

- よく分からぬいけど言われるがままに SSL の設定をしている人
- SSL と TLS の関係性がよく分かっていない人
- SSL 証明書がいったい何を証明しているのか知らない人
- SSL は聞いたことがあるけど TLS は知らないという人
- 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
- 「インフラがよく分からぬこと」にコンプレックスのある人
- 証明書の購入や設置はしたことがあるけど SSL はあまり分かっていない人
- サイトを HTTPS 化しなきゃ！ と思っている人

マッチしない読者層

本著は、こんな人が読むと恐らく「not for me だった…（私向けじゃなかった）」となります。

- SSL/TLS の通信を C 言語で実装したい人
- 「プロフェッショナル SSL/TLS」を読んで完全に理解できた人

本著の特徴

本著では実際にサーバを立てて SSL 証明書の設置を行い、HTTPS のサイトを作つてみます。手を動かして試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、

- こんな障害が起きたら原因は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。

- SSL 証明書がどんな役割を果たしているのか説明できる
- 証明書を買うときは何に注意してどんな手順で買ったらいいか分かっている
- 意図せず「保護されていない通信」と表示されてしまったときの対処法が分かる
- 障害が起きたときに原因を調査できる
- 読む前より SSL が好きになっている
- SSL/TLS と併記されている「TLS」の意味が分かっている

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、必要に応じて適宜改訂を行いますので GitHub の Issue や Pull request で筆者までお知らせいただけますと幸いです。

<https://github.com/mochikoAsTech/startSSL>

目次

| | |
|---|-----------|
| はじめに | 3 |
| 想定する読者層 | 3 |
| マッチしない読者層 | 3 |
| 本著の特徴 | 4 |
| 本著のゴール | 4 |
| 免責事項 | 4 |
| 第1章 Oracle Cloud のアカウントを作ろう | 9 |
| 1.1 ウェブサーバを立てよう | 10 |
| 1.1.1 サイトを作るのにどうしてサーバがいるの？ | 10 |
| 1.1.2 サーバを立てるにはお金が必要？ | 11 |
| 1.1.3 なんで AWS じゃなくて Oracle のクラウドを使うの？ | 11 |
| 1.2 Oracle Cloud でアカウント登録 | 12 |
| 1.2.1 無料でアカウントを作成 | 12 |
| 【コラム】どうしても SMS が届かない！ そんなときは？ | 18 |
| 1.2.2 Oracle Cloud のコンソールにサインイン | 23 |
| 第2章 Oracle Cloud でサーバを立てよう | 27 |
| 2.1 事前準備 | 28 |
| 2.1.1 お使いのパソコンが Windows の場合 | 28 |
| 【コラム】パスフレーズは設定すべき？ しなくてもいい？ | 37 |
| 2.1.2 お使いのパソコンが Mac の場合 | 38 |
| 【コラム】ターミナルでコピー＆ペーストするには？ | 40 |
| 2.2 コンピュートでサーバを立てる | 41 |
| 2.3 ドメイン名の設定 | 43 |
| 2.4 まずは HTTP でサイトを公開 | 43 |

| | | |
|--------------|--|-----------|
| 2.5 | 証明書を取得しよう | 43 |
| 2.5.1 | 秘密鍵を作ろう | 43 |
| 2.5.2 | CSR を作ろう | 43 |
| 2.5.3 | 証明書の取得申請 | 43 |
| 2.5.4 | 取得した証明書をサーバに置こう | 43 |
| 2.6 | HTTPS でサイトを公開 | 43 |
| 第 3 章 | 基本 | 45 |
| 3.1 | SSL ってなに? | 46 |
| 3.2 | TLS ってなに? | 46 |
| 3.3 | SSL と TLS の違いは? | 46 |
| 3.4 | SSL と SSH って似てる? 何が違うの? | 46 |
| 3.5 | HTTPS で始まるページで鍵のマークが壊れて表示された | 46 |
| 3.6 | 種類 | 46 |
| 3.6.1 | SSL サーバ証明書 | 46 |
| 3.6.2 | SSL クライアント証明書 | 46 |
| 3.7 | どんなシーンで使われている? | 46 |
| 3.8 | SSL 証明書は全然違う 2 種類の仕事をしている | 46 |
| 3.8.1 | Web サイトで送受信する情報を暗号化すること | 46 |
| 3.8.2 | Web サイト運営者の身元を証明すること | 46 |
| 3.9 | 鍵マークが壊れるケース | 46 |
| 3.9.1 | すべて HTTP で通信しているとき | 46 |
| 3.9.2 | HTTPS だけど一部が HTTPS じゃないとき | 46 |
| 3.10 | ウェブページが表示されるまで | 46 |
| 3.10.1 | 1 往復で表示されるわけじゃない | 46 |
| 3.11 | SSL 証明書は何を証明してくれるのか? | 46 |
| 3.11.1 | ネットバンクの事例 | 46 |
| 3.12 | 認証局事業者の身元は誰が証明する? | 46 |
| 3.12.1 | 身元保証の連鎖をつなぐ中間 CA 証明書とルート証明書 | 46 |
| 3.13 | SSL 証明書はどうしてあんなに値段に差があるの? | 46 |
| 3.14 | 同じ「SSL 証明書」という名前でも 3 つの種類がある | 46 |
| 3.14.1 | EV 証明書 | 46 |
| 3.14.2 | OV 証明書 | 46 |
| 3.14.3 | DV 証明書 | 46 |
| 3.14.4 | 3 つの違いは何か? | 46 |

| | |
|---|-----------|
| 3.14.5 ブラウザベンダーによる EV 証明書の扱いの変化 | 46 |
| 3.15 その他の証明書 | 46 |
| 3.15.1 中間証明書 | 46 |
| 3.15.2 クロスルート証明書 | 46 |
| 3.16 どの証明書を買えばいい? | 46 |
| 3.16.1 ワイルドカード証明書 | 46 |
| 3.16.2 www ありにリダイレクトしたいだけなのに www なしの証明書 もいるの? | 46 |
| 3.16.3 コモンネームが*.example.com の証明書は example.com で使え る? | 46 |
| 3.16.4 Let'sEncrypt | 46 |
| 3.17 CDN と証明書 | 46 |
| 3.17.1 CDN を使ったら古い端末でサイトが見られなくなった | 46 |
| 3.17.2 同じサーバで複数サイトを HTTPS 化したら古い端末で別サイ トが表示された | 46 |
| 3.17.3 SNI Server Name Indication | 46 |
| あとがき | 47 |
| PDF 版のダウンロード | 47 |
| Special Thanks: | 47 |
| レビュー | 47 |
| 参考文献 | 47 |
| 著者紹介 | 49 |

第 1 章

Oracle Cloud のアカウントを作 ろう

この章では Oracle Cloud というクラウドでアカウントを作ります。

SSL を理解するには、実際に手を動かしてやってみるのがいちばんです。実際に SSL 証明書を取得して、HTTPS のサイトを作ってみましょう。

HTTPS でサイトを作るのに必要な材料は次の 3 つです。

- ウェブサーバ
- ドメイン名
- SSL 証明書

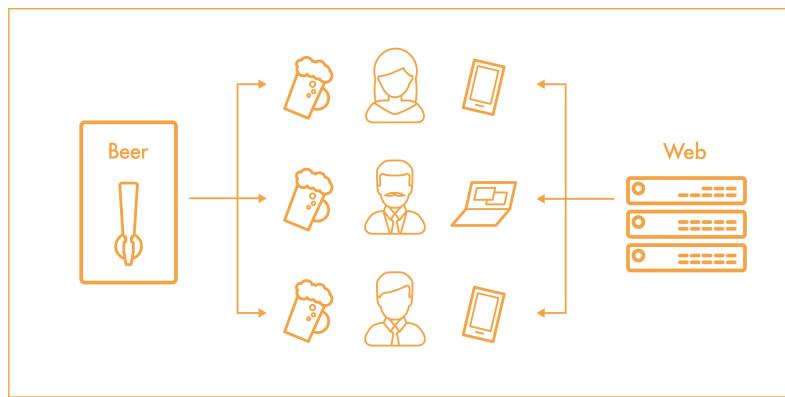
まずは 1 つめのウェブサーバを立てるため、アカウント作成から始めましょう。

1.1 ウェブサーバを立てよう

1.1.1 サイトを作るのにどうしてサーバがいるの？

これからウェブサーバを立てますが…どうしてサイトを作りたいだけなのに、ウェブサーバが必要なのでしょう？

そもそもですが、サーバとは**クライアントに対してサービスを提供するものです**。居酒屋にあるビアサーバに「ビールをください」というリクエストを投げる…つまりコックを「開」の方へひねると、ビールというレスポンスが返ってきます。同様にあなたがブラウザでURLを入力したり、リンクをクリックしたりして、ウェブサーバに対して「ウェブページを見せてください」というリクエストを投げたら、ウェブページというレスポンスが返ってきます。（図1.1）



▲図1.1 ビアサーバもウェブサーバもリクエストしたらサービスが提供される

つまり、せっかくHTMLや画像でサイトのコンテンツを作っても、それを載せておくウェブサーバがなければ、サイトはあなたのパソコンの中でしか見られず、インターネットで公開できないのです。^{*1}

というわけでウェブサイトを提供するために、まずはウェブサーバを立てましょう！

^{*1} サーバについては、はじめようシリーズの2冊目、「AWSをはじめよう」の「CHAPTER1 インフラとサーバってなに？」で、より詳しく解説しています。仮想サーバと物理サーバ、クラウドとオンプレミス、ホストサーバとゲストサーバなどサーバ周りの用語をもう少し理解したい！という方はそちらも併せて読んでみるのがお勧めです

1.1.2 サーバを立てるにはお金が必要？

ウェブサイトを作るにはサーバが必要です。そしてサーバを立てるには、普通はお金がかかります。ですがオラクルがやっている「Oracle Cloud（オラクル クラウド）」というサービスなら、なんと有効期限なしでずっと無料で使える「Always Free」という枠があります。「Always Free」の範囲内であれば、サーバも無料で立てて使えるので今回はそれを使いましょう。

オラクルがやっているクラウド、と言われても、そもそもクラウドがなんだか分からないといまいちピンと来ないかもしれません。あなたが「ウェブサイト作りたいなあ…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、従量課金ですぐに使って性能や台数の増減も簡単にできるのがクラウドです。**

Oracle Cloud とはオラクルがやっているクラウドなので、ブラウザでぽちぽちとスペックを選んでいくだけで、すぐにサーバが使えます。

1.1.3 なんで AWS じゃなくて Oracle のクラウドを使うの？

クラウドは Oracle Cloud だけではありません。かの有名な AWS こと Amazon Web Services や、Google の Google Cloud Platform^{*2}、Microsoft の Azure（アジュール）^{*3}、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^{*4}、お名前.com でお馴染み GMO グループの GMO クラウド^{*5}などたくさんあります。

2019年11月時点、クラウド市場では AWS がシェア約40%でトップを独走中^{*6}です。そのため仕事で AWS を使ったことがある、あるいはこれから使う予定だ、というエンジニアも多いと思います。

しかし最近は、Alibaba Cloud や Tencent Cloud といった中国のクラウド事業者も追い上げを見せています。こうした新興のクラウドは、先に行く AWS を見て学んだ上で生まれてきているだけあって、よりスマートな作りになっているのがいいところです。

たくさんのクラウドがある中でどこを選ぶのか、その理由は、本来であれば使う人やそ

^{*2} <https://cloud.google.com/>

^{*3} <https://azure.microsoft.com/ja-jp/>

^{*4} <https://cloud.sakura.ad.jp/>

^{*5} <https://www.gmocloud.com/>

^{*6} IaaS + PaaS クラウド市場、AWS の首位ゆるがず。AWS、Azure、Google、Alibaba の上位4社で市場の7割超。2019年第3四半期、Synergy Research Group — Publickey https://www.publickey1.jp/blog/19/iaaspasawsazuregooglealibaba4720193synergy_research_group.html

の上で動かすサービスによって異なるはずです。あなたが動かしたいサービスには、いったいどのクラウドが適しているのでしょうか？

本著では以下を目的としていますので、それに適した Oracle Cloud で学びを進めていきたいと思います。

- SSL 証明書を自分で取得して設置する一通りの流れを試したい
- お金をかけずに無料で試したい

1.2 Oracle Cloud でアカウント登録

先ずは Oracle Cloud のアカウントを作りますので次の 2 つを用意してください。

- クレジットカード
- SMS 受信が可能な携帯電話（電話番号認証で使用するため）^{*7}

なお Oracle Cloud を利用する際は、前述のとおり Always Free という無料枠^{*8}があります。

1.2.1 無料でアカウントを作成

「Oracle Cloud 無料」で検索（図 1.2）したら、いちばん上の [Oracle Cloud Free Tier | Oracle 日本]^{*9}をクリックします。

^{*7} ショートメッセージサービスの略。宛先に電話番号を指定してメッセージを送れるサービス

^{*8} 期限なしでずっと無料ですが、無料で利用できる範囲は決まっていて、何をどれだけ使っても無料という訳ではありませんので注意してください。Always Free の他に、30 日間だけ有効な 300 ドル分の無償クレジットも付いてきますので、Always Free の範囲外のサービスはそちらで試せます。詳細は <https://www.oracle.com/jp/cloud/free/> を確認してください

^{*9} <https://www.oracle.com/jp/cloud/free/>

1.2 Oracle Cloud でアカウント登録



[今すぐ始める（無償）] をクリックします。（図 1.3）



「Oracle Cloud へのサインアップ」と表示されました。それでは次の情報を入力して、使用条件を確認した上で【次】をクリックしましょう。(図 1.4) 後で分からなくならないように、登録した項目をメモしておきましょう。(表 1.1)



▲図 1.4 入力したら【次】をクリック

▼表 1.1 Oracle Cloud に登録した情報

| 項目 | 例 | あなたが登録した情報 |
|------------|-----------------------|------------|
| 電子メール・アドレス | startdns.01@gmail.com | |
| 国/地域 | 日本 | |

次は「アカウント詳細の入力」です。(表 1.2) 今回は仕事ではなく個人での利用ですので【アカウント・タイプ】は【個人使用】を選択してください。【クラウド・アカウント名】には任意のアカウント名を入力します。【クラウド・アカウント名】には英字小文字と数字のみ使えます。記号や英字大文字は使えないで注意してください。筆者は startdns01 にしました。この【クラウド・アカウント名】は、後で管理画面にサインインするときのアカウント URL になります。(図 1.5)

【ホーム・リージョン】は【日本東部(東京)】を選択してください。Oracle Cloud は世界の各地域にデータセンターを所有しており、サーバはそのデータセンターの中で元気に動いています。この【ホーム・リージョン】とは、**各地域の中でどこを使うか？を指定するものです。** ウェブサイトにアクセスするとき、パソコンのある場所からサーバまで物理

的に距離が遠いと、それだけ通信にも時間がかかるて応答時間も遅くなりますので、日本国内向けにウェブサイトを開設する場合は基本的にこの「東京リージョン」を選びましょう。ただし Oracle Cloud のサービスによってはまだ東京リージョンが使えないものもあります。その場合は次点として「米国東部(アッシュバーン)」を選択してください。

▼表 1.2 Oracle Cloud に登録した情報

| 項目 | 例 | あなたが登録した情報 |
|-------------|------------|------------|
| アカウント・タイプ | 個人使用 | - |
| クラウド・アカウント名 | startdns01 | |
| ホーム・リージョン | 日本東部(東京) | - |

▲図 1.5 [クラウド・アカウント名] には好きな名前を入力

続いて名前や住所を入力していきます。入力内容は日本語表記で構いません。個人利用なのですが「部門名」が必須であるため、ここでは「個人」と入力しておきましょう。[名]・[姓]・[部門名]・[住所]・[市区町村]・[都道府県]・[郵便番号]をすべて入力できましたか？（図 1.6）

The screenshot shows a form for creating an Oracle Cloud account. The address input section is highlighted with red boxes around the following fields: Name (名), Surname (姓), Department (部門名), Position (役職), Address (住所), City/Town/Village (市区町村), Prefecture/Province (都道府県), and Zip Code (郵便番号). The 'Address' field contains '新宿四丁目1番6号' and 'JR新宿ミライナタワー'. The 'City/Town/Village' field contains '新宿区' and the 'Prefecture/Province' field contains 'TOKYO'.

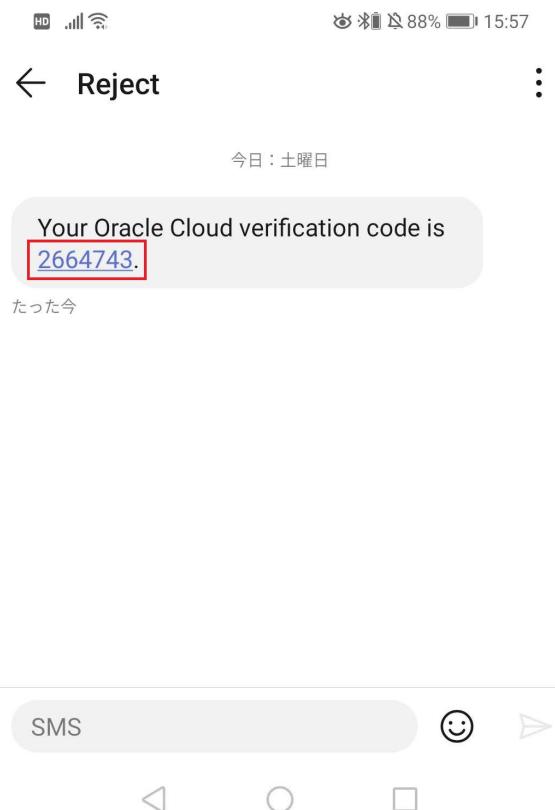
▲図 1.6 名前や住所を入力

では最後に「モバイル番号」です。国番号は「日本(81)」を選択して、自分の携帯電話番号を入力します。このとき電話番号の先頭の 0 は不要です。例えば「090-〇〇〇〇-〇〇〇〇」という携帯電話番号であれば「90-〇〇〇〇-〇〇〇〇」と入力してください。携帯電話番号を入力したら【次: モバイル番号の確認】をクリックしてください。(図 1.7)

The screenshot shows the 'Mobile Phone Number Verification' step. It displays the selected country code '日本 (81)' and the entered mobile number '90 [REDACTED]'. A note below explains that the first digit '0' is not required for Japanese mobile numbers. A green checkmark icon is visible next to the mobile number input field. Below the input fields, there is a button labeled '次: モバイル番号の確認' (Next: Mobile phone number confirmation).

▲図 1.7 携帯電話の番号を入力

数分以内に [Your Oracle Cloud verification code is ○○○○○○○○.] と書かれた SMS が届きます。(図 1.8)



▲図 1.8 コードの書かれた SMS が届いた

SMS で届いた「〇〇〇〇〇〇〇」の数字を [コード] に入力して、[コードの確認] をクリックします。(図 1.9)



▲図 1.9 SMS で届いた数字を [コード] に入力して [コードの確認] をクリック

【コラム】どうしても SMS が届かない！ そんなときは？

電話番号を入力したのに SMS が届かないときは、まず自分が契約している携帯キャリアの迷惑メール設定で、SMS をスパムとしてはじく設定をしていないか確認してみましょう。たとえば海外の事業者から送信された SMS を拒否する設定になっていたり、海外からの着信を拒否する設定になっていると、SMS が届かないことがあるようです。^{*10}

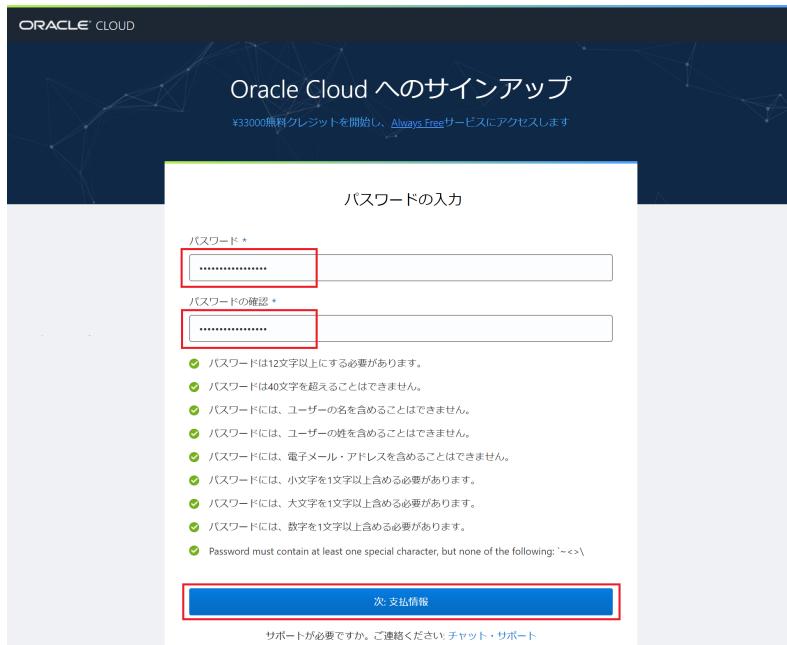
ちなみに筆者の場合は、特に設定変更をせず同じ番号で 2 回試してみたのですが、1 回目は届かず、もう 1 回試してようやく届きました。

迷惑メールの設定を確認して何回か試して、それでも SMS が届かなかったら、ページ下部の「[サポートが必要ですか。ご連絡ください: チャット・サポート]」からサポートにチャットで問い合わせてみましょう。

残念ながら英語でしか対応してもらえませんが、「I am trying to register on Oracle Cloud. But I can't receive SMS. What should I do?」（アカウント登録

しようとしてるけど SMS が届かないの、どうしたらいい?) という感じで聞いてみると、「じゃあ登録情報をこのチャットで教えて。そうしたらこちらでコードを発行して、チャットで伝えてあげる」(意訳) という感じでサポートしてもらえます。

正しいコードが入力できたら、[パスワードの入力] と表示されます。[パスワード] と [パスワードの確認] を入力して、[次: 支払情報] をクリックします。(図 1.10)



▲図 1.10 パスワードを入力して [次: 支払情報] をクリック

パスワードを入力すると、今度は [支払情報] のページが表示されます。(図 1.11) 繰り返しあ伝えしているとおり、Oracle Cloud には Always Free という無料枠があり、本書ではその無料枠の範囲内で Oracle Cloud を使っていくつもりですが、それでもクレジットカードは登録しておく必要があります。記載されているとおり、この後、管理画面で [アカウントのアップグレード] という作業をしない限り、請求は発生しませんので安

*10 「Oracle Cloud の SMS は海外の事業者から届く」という確証がある訳ではないです。あくまで SMS が届かないときによくある話と思ってください

心してカード情報を登録してください。[クレジット・カード詳細の追加] をクリックします。



▲図 1.11 [クレジット・カード詳細の追加] をクリック

[ご注文者様情報] はそのまま変更不要です。[カード情報] の [カードの種類] を選択し、[カードの番号]・[有効期限]・[CVN] を入力したら [Finish] をクリックします。
(図 1.12)*¹¹

*¹¹ Oracle Cloud では、クレジットカード登録時に「1 ドル認証」と呼ばれる認証方法で、そのクレジットカードが決済可能かをチェックしています。クレジットカードによってはこの 1 ドル認証を不審な決済と判断して通さないため、それによってエラーが発生することがあります。その場合は別のクレジットカードで試すか、Oracle Cloud のチャット・サポートで問い合わせてみてください

The screenshot shows a 'Card Information' form. At the top, there's a heading 'カード情報' with a small orange info icon. Below it, a section for 'Card Type' has four options: Visa, Mastercard, Amex, and JCB. The Visa and Mastercard options are highlighted with a red box. The next section is 'Card Number' with a red box around the input field. Below that is 'Expiration Date' with a red box around the dropdown menus. The final section is 'CVN' with a red box around the input field. To the right of the CVN field is a note: 'このコードは、クレジットカードの裏面または表面に印字されている3桁または4桁の番号です。' (This code is the three or four digit number printed on the back or front of the credit card). At the bottom right is a green 'Finish' button with a red box around it.

▲図 1.12 カード情報を入力して [Finish]

[クレジット・カード詳細をご提供いただきありがとうございます。] と表示（図 1.13）されたら、支払い情報の登録は完了です。Oracle Cloud の Service Agreement^{*12}を確認した上で、チェックボックスにチェックを入れて、[サインアップの完了] をクリックします。

*12 <https://www.oracle.com/goto/oraclecsa-jp-en>

第1章 Oracle Cloud のアカウントを作ろう



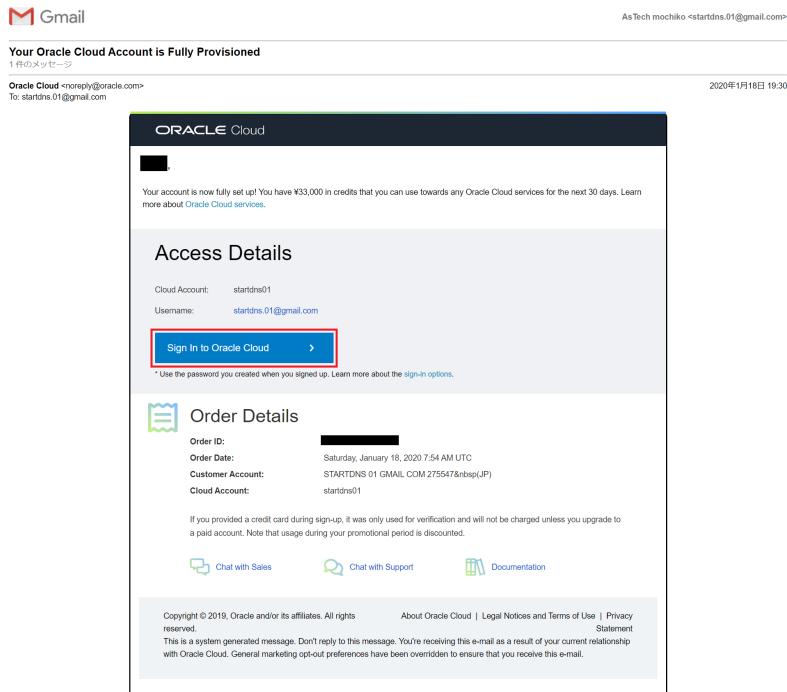
▲図 1.13 チェックを入れて [サインアップの完了] をクリック

これでアカウント登録の手続きはおしまいです。[アカウントの設定が完了するまでお待ちください。] と表示（図 1.14）されます。準備が整うとサインイン画面にリダイレクトされますが、この [アカウントの設定が完了するまでお待ちください。] の画面でかなり時間がかかるので一度ブラウザを閉じてしまって構いません。頑張った自分を褒めて一旦休みましょう。



▲図 1.14 アカウント登録の手続きはおしまい

数時間後^{*13}、[Your Oracle Cloud Account is Fully Provisioned] という件名で、準備完了を知らせるメールが届きます。メールの [Sign In to Oracle Cloud] をクリックしましょう。(図 1.15)



▲図 1.15 準備完了を知らせるメールが届いた

1.2.2 Oracle Cloud のコンソールにサインイン

メールの [Sign In to Oracle Cloud] をクリックすると、コンソールへのサインイン^{*14}画面が表示されます。(図 1.16) [ユーザー名] には先ほど登録したメールアドレスを入力します。^{*15} [パスワード] を入力して、[サイン・イン] をクリックしてください。

*13 筆者の場合は、メールが届くまで 2 時間半かかりました

*14 日本語だとログインの方が馴染みがあるかも知れませんが、サインインはログインと同じ意味です。

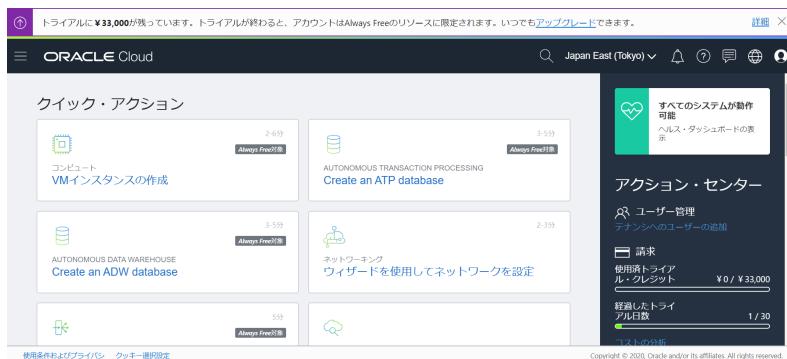
*15 メールにも書いてありますが、ここでの [ユーザー名] とは [クラウド・アカウント名] (筆者の場合は startdns01) ではなく、[メールアドレス] のことです。紛らわしいのでご注意ください

第1章 Oracle Cloud のアカウントを作ろう



▲図 1.16 [ユーザー名] と [パスワード] を入力して [サイン・イン]

おめでとうございます！ これでコンソールにサインインできました。



▲図 1.17 コンソールにサインインできた！

なお今後、コンソールにサインインしたくなったら、いちいちメールを探してリンクを踏む必要はありません。まずは Oracle のトップページ^{*16}を開いて、右上の人物マークから [クラウドにサインイン] をクリックしましょう。

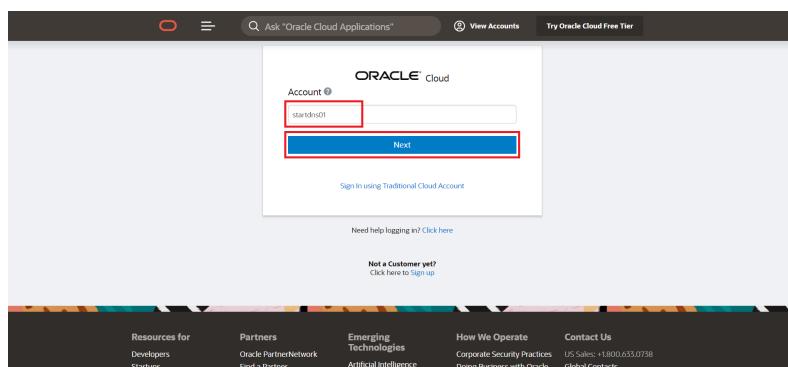
*16 <https://www.oracle.com/jp/>

1.2 Oracle Cloud でアカウント登録



▲図 1.18 右上の人マークから [クラウドにサインイン] をクリック

サインインのページ^{*17}で [Account] の欄にクラウド・アカウント名^{*18}を入力して [Next] をクリックすれば、メールのリンクを踏んだときと同じ [サイン・イン] のページにたどり着けます。あとは同じように [ユーザー名] にはメールアドレスを、[パスワード] にはパスワードを入力して、[サイン・イン] をクリックするだけです。



▲図 1.19 [Account] の欄にクラウド・アカウント名を入力して [Next] をクリック

*17 <https://www.oracle.com/cloud/sign-in.html>

*18 筆者の場合は startdns01 です。アカウント登録時に、あなたの [クラウド・アカウント名] をメモしているはずですでの、数ページ戻って確認してみましょう

第2章

Oracle Cloud でサーバを立てよう

この章では実際に Oracle Cloud でサーバを立てます。
インフラエンジニアのお仕事体験みたいできっと楽しいですよ！

2.1 事前準備

2.1.1 お使いのパソコンが Windows の場合

RLogin のインストール

Windows のパソコンを使っている方は、サーバを立てる前に「ターミナル」と呼ばれる黒い画面のソフトをインストールしておきましょう。サーバに接続するときにはこのターミナルを使うのですが、ターミナルのソフトには色々な種類があります。

- RLogin (<http://nanno.dip.jp/softlib/man/rlogin/>)
- Poderosa (<https://ja.poderosa-terminal.com/>)
- Tera Term (<https://ja.osdn.net/projects/ttssh2/>)
- PuTTYjp (<http://hp.vector.co.jp/authors/VA024651/PuTTYkj.html>)^{*1}



▲図 2.1 RLogin

本著ではいちばん上の RLogin (図 2.1) を使って説明していきますので、特にこだわりがない場合は RLogin を使うことをお勧めします。RLogin の「実行プログラム (64bit)^{*2}」(図 2.2) の URL、http://nanno.dip.jp/softlib/program/rlogin_x64.zip をクリックしてください。

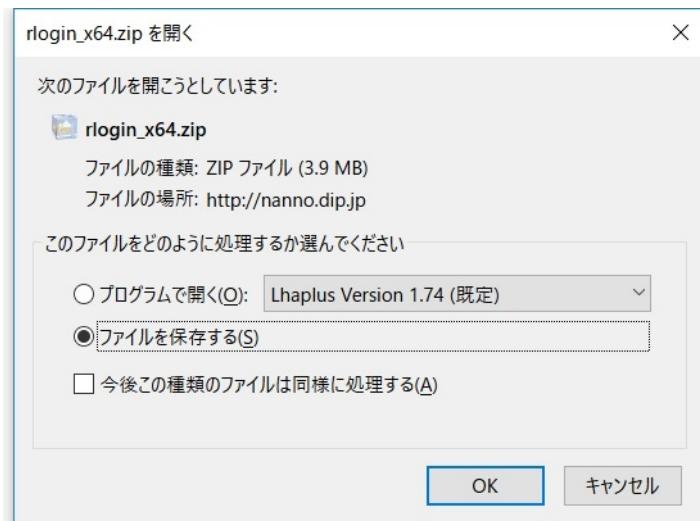
^{*1} PuTTYjp を使う場合、.pem の秘密鍵を PuTTYgen で.ppk に変換する必要が出てくるため、他のターミナルソフトに比べると一手間余計にかかります。

^{*2} もしパソコンの Windows が 32bit 版だった場合は「実行プログラム (32bit)」の URL をクリックしてください。



▲図 2.2 「実行プログラム(64bit)」の URL をクリックしてダウンロード

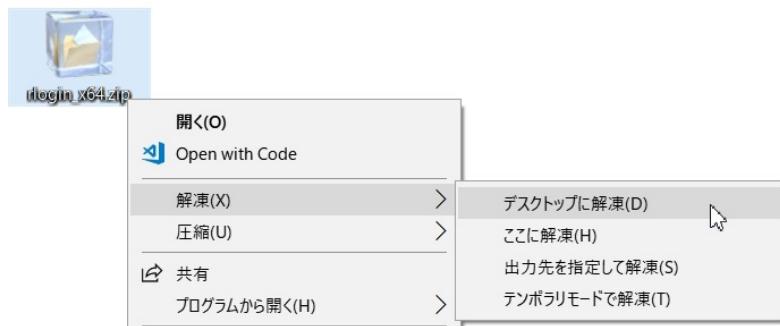
ダウンロードした ZIP ファイルを保存（図 2.3）します。保存場所はどこでも構いませんが、後でどこに置いたか分からなくなりそうな人はデスクトップに保存しておきましょう。



▲図 2.3 「ファイルを保存する」でパソコンに保存

デスクトップの ZIP ファイル (rlogin_x64.zip) を右クリック（図 2.4）して、[解凍]> [デスクトップに解凍]*3をクリックします。

*3 ZIP ファイルを右クリックしても「解凍」が見当たらないときは、圧縮・解凍の定番ソフトである Lhaplus をインストールしましょう。 <https://forest.watch.impress.co.jp/library/software/lhaplus/>



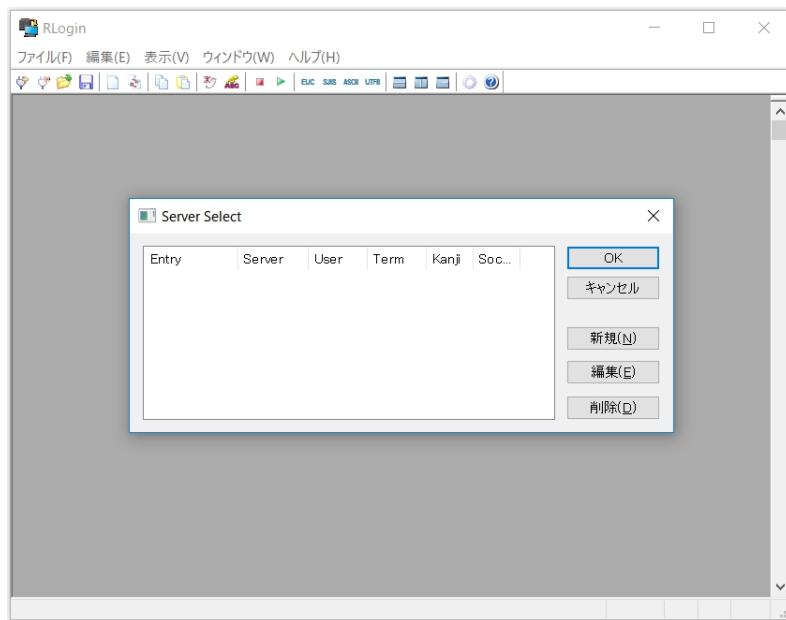
▲図 2.4 ZIP ファイルを右クリックして解凍>デスクトップに解凍

解凍したら、デスクトップにできた「rlogin_x64」というフォルダの中にある「RLogin.exe」^{*4}（図 2.5）をダブルクリックすれば RLogin が起動（図 2.6）します。



▲図 2.5 RLogin.exe をダブルクリック

^{*4} フォルダの中に RLogin はあるけど RLogin.exe なんて見当たらない・・・という場合、ファイルの拡張子が非表示になっています。この後も拡張子を含めてファイル名を確認する場面が何度かでできますので、表示されていない人は「拡張子 表示」で Google 検索して拡張子が表示されるように設定変更しておきましょう。

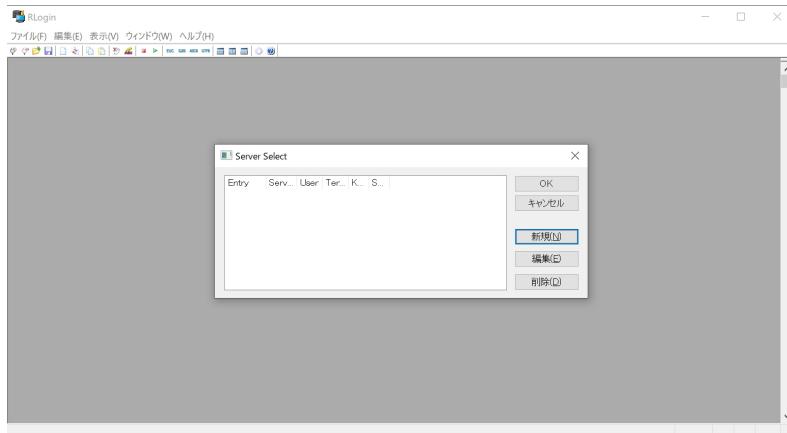


▲図 2.6 RLogin が起動した

これで RLogin のインストールは完了です。

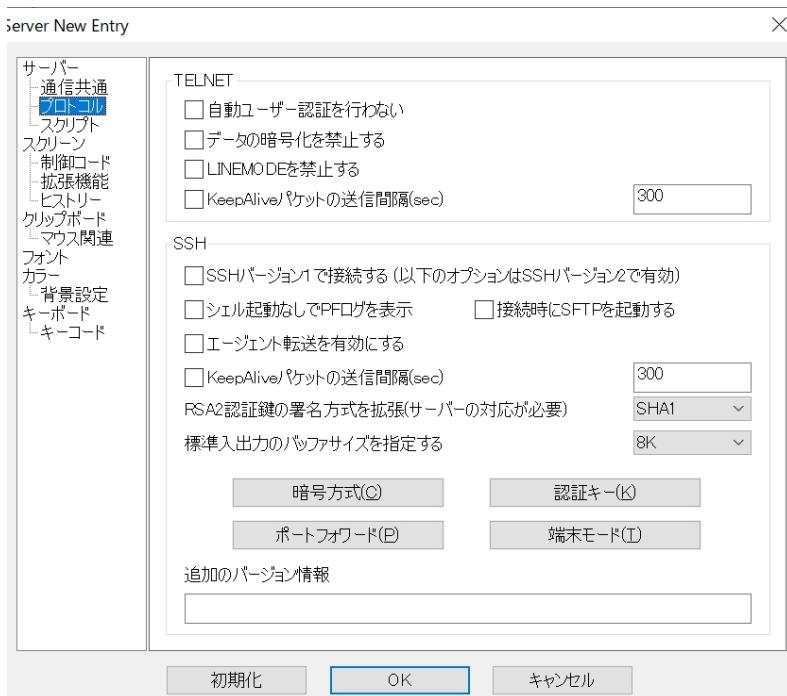
Windows で SSH のキーペア（秘密鍵・公開鍵）を作成する

Windows の方は、起動した RLogin で [新規 (N)] をクリックします。



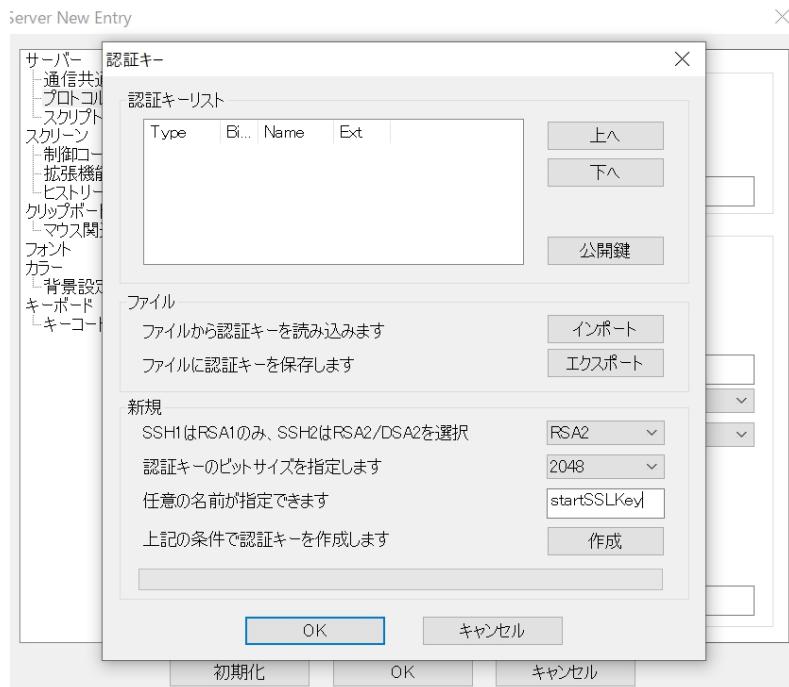
▲図 2.7 [新規 (N)] をクリック

左メニューの [サーバー>プロトコル] を選択して、[認証キー (K)] をクリックします。



▲図 2.8 [サーバー>プロトコル] を選択して [認証キー (K)] をクリック

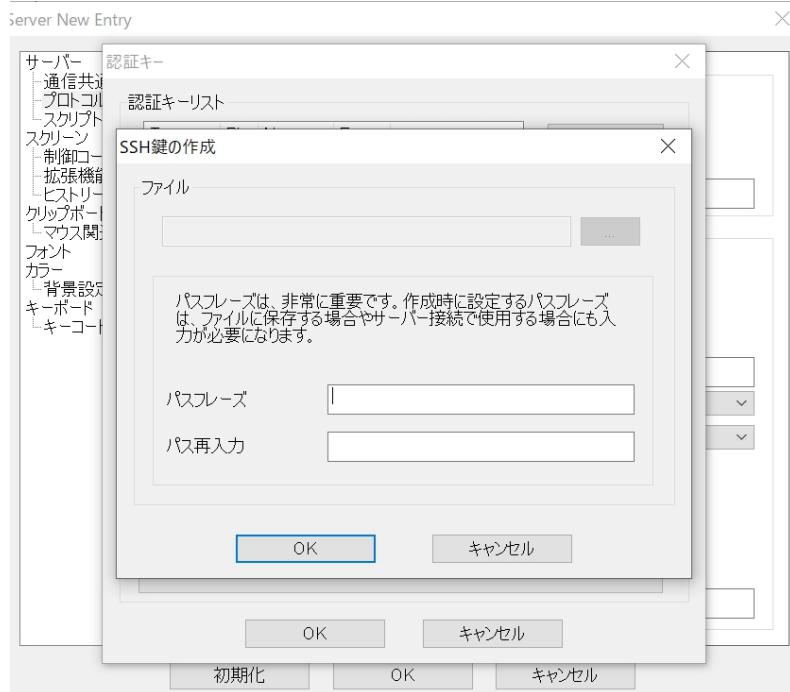
[任意の名前が指定できます] に [startSSLKey] を入力して、[作成] をクリックします。



▲図 2.9 [startSSLKey] を入力して [作成] をクリック

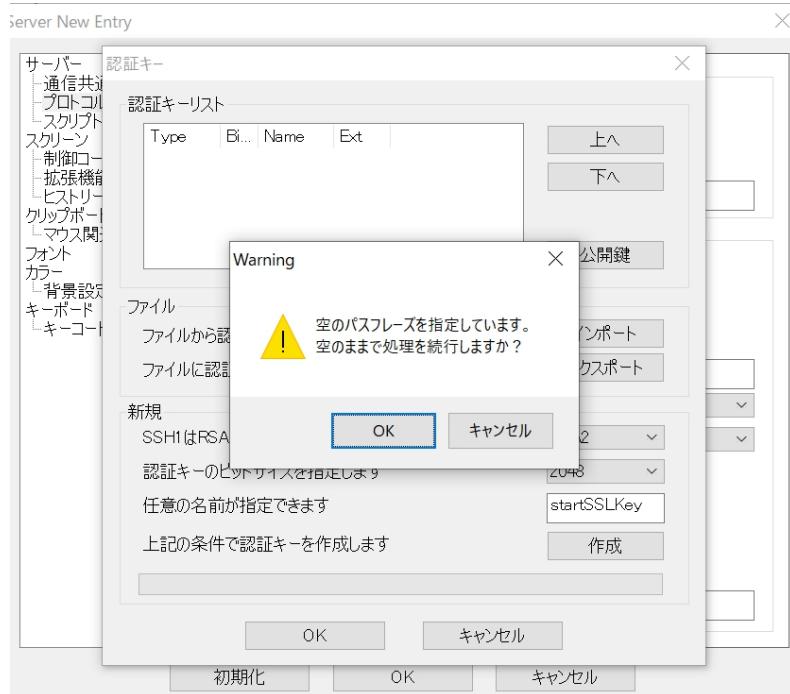
[パスフレーズ] と [パス再入力] には何も入力せず、[OK] をクリックします。^{*5}

^{*5} 「p@\$sw0rd」や「@dm1ni\$strat0r」のように、ひとつの単語でできているのがパスワードです。それに対して「This 1s P@ss\$ Phrase.」のように空白を挟んだ文章（フレーズ）で構成されているのものをパスフレーズと呼びます



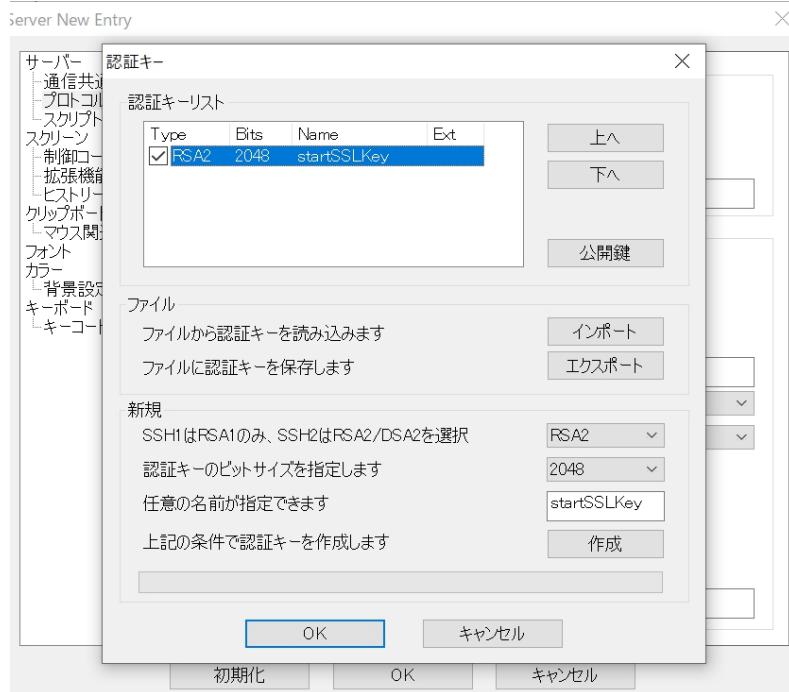
▲図 2.10 何も入力せず [OK] をクリック

[空のパスフレーズを指定しています。空のままで処理を続行しますか?] と表示されますが、そのまま [OK] をクリックします。



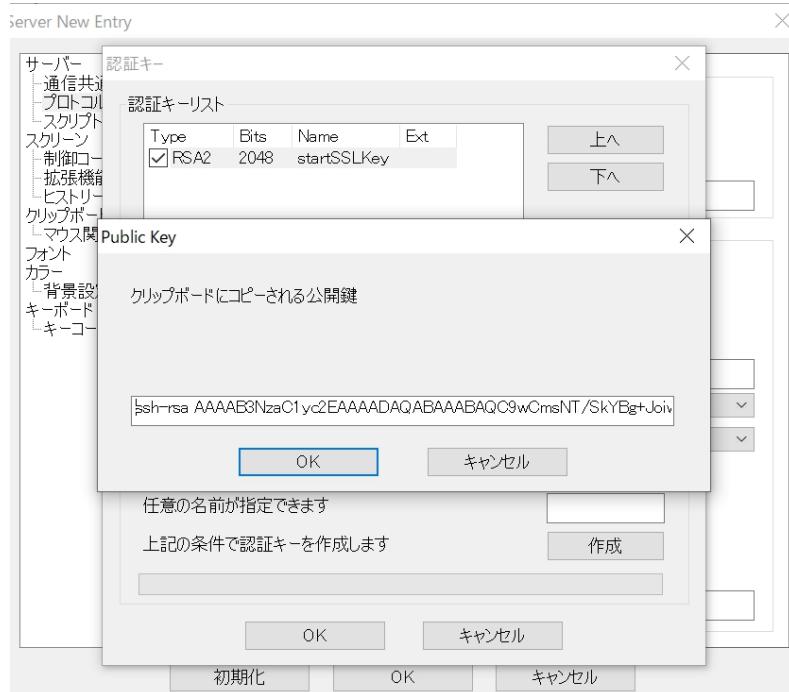
▲図 2.11 [OK] をクリック

[認証キーリスト] に、今作った [startSSLKey] が表示されたら、キーペア（秘密鍵・公開鍵）が無事できています。[公開鍵] をクリックしてください。（図 2.5）



▲図 2.12 キーペアが出来たら [キャンセル] して RLogin を閉じよう

この後すぐに使いますので、表示された公開鍵（ssh-rsa から始まる文字列）をまるごとコピーして、メモ帳などにペーストしておきましょう。



▲図 2.13 表示された公開鍵（文字列）はまるごとコピーしてメモ帳にペーストしておこう

公開鍵をメモしたら [キャンセル] を繰り返し 4 回クリックして、起動中の RLogin はいったん閉じてしまって構いません。RLogin はまた後で使いますので、デスクトップの「rlogin_x64」フォルダとその中にある「RLogin.exe」をごみ箱へ捨てないように注意してください。

【コラム】パスフレーズは設定すべき？ しなくてもいい？

秘密鍵に「パスフレーズ」を設定しておくと、鍵を使ってサーバに入ろうとしたとき、「鍵を発動するにはパスフレーズを叫べ…！」という感じでパスフレーズを聞かれます。

つまり、もしあなたの秘密鍵が盗まれて勝手に使われそうになっても、パスフレーズを設定していれば鍵の悪用が防げます。スマホが盗まれてしまっても、パスワードが分からなければロック画面が解除できず、勝手に使えないのと同じです。

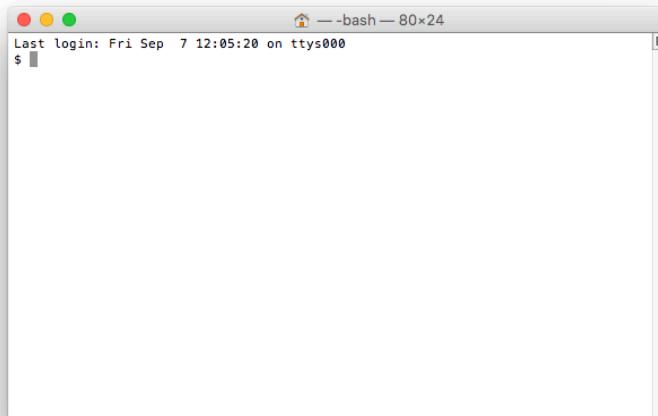
ただ「パスワード認証じゃなくて鍵認証なのに、パスフレーズも要るの…？」という点で、初心者を混乱に陥れやすいので、本著では秘密鍵をパスフレーズなしで作って使います。

パスフレーズを設定していれば絶対に安心！ というものではありませんが、上記の理由から、本来であれば設定した方がいいものです。

起動した RLogin はいったん「キャンセル」をクリックして閉じてしまつて構いません。また後で使いますので、デスクトップの「rlogin_x64」フォルダとその中にある「RLogin.exe」をごみ箱へ捨てないように注意してください。

2.1.2 お使いのパソコンが Mac の場合

Mac を使っている方は、最初から「ターミナル」（図 2.14）というソフトがインストールされていますのでそちらを利用しましょう。



▲図 2.14 最初からインストールされている「ターミナル」を使おう

ターミナルがどこにあるのか分からぬときは、Mac の画面で右上にある虫眼鏡のマークをクリックして、Spotlight で「ターミナル」と検索（図 2.15）すれば起動できます。



▲図 2.15 どこにあるのか分からなかつたら Spotlight で「ターミナル」と検索

Mac で SSH のキーペア（秘密鍵・公開鍵）を作成する

Mac の方は、ターミナルで次のコマンドを実行してください。^{*6}

```
ssh-keygen -f ~/startSSLKey
```

すると次のように、パスフレーズの入力待ち状態になります。何も入力せずに、2回 Enter を押してください。

```
$ ssh-keygen -f ~/startSSLKey
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase): ←何も入力せずに Enter
Enter same passphrase again: ←何も入力せずに Enter
```

次のように表示されたらキーペア（秘密鍵・公開鍵）の作成は完了です。

^{*6} ssh-keygen コマンドは名前のとおり、SSH の鍵（key）を生成（generate）するコマンドです。-f オプションでは、生成する鍵のファイル名を指定しています。～（チルダ）はホームディレクトリを表しますので、-f ~/startSSLKey は「/Users/<ユーザ名>/startSSLKey」という鍵を作つて、という意味です

```
$ ssh-keygen -f ~/startSSLKey
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/mochikoAsTech/startSSLKey.
Your public key has been saved in /home/mochikoAsTech/startSSLKey.pub.
The key fingerprint is:
a2:52:43:dd:70:5d:a8:4f:77:47:ca:f9:69:79:14:48 mochikoAsTech@ghana
The key's randomart image is:
+--[ RSA 2048]----+
|          . ooE. |
|          . + o   . |
|          . . .   . + |
|          . . . = o |
|          o . So   . +o |
|          . o .   +o |
|          . .     . |
|          .         |
+-----+
```

ホームディレクトリに秘密鍵（startSSLKey）と、公開鍵（startSSLKey.pub）ができるあがっているはずです。cat（キャット）コマンド^{*7}で公開鍵を表示してみましょう。

```
$ cat ~/startSSLKey.pub
ssh-rsa AAAAB3NzaC1yc2E=Unidb+6FjiLw== mochikoAsTech@mochikoMacBook-Air.local
```

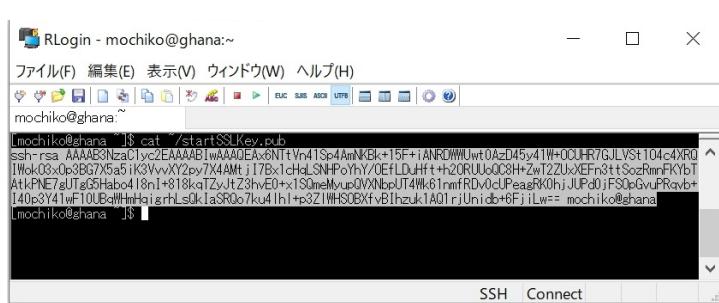
この後すぐに使いますので、表示された公開鍵（ssh-rsa から始まる文字列）をまるごとコピーして、メモ帳などにペーストしておきましょう。

以上で事前準備は完了です。お待たせしました。いよいよサーバを立てましょう。

【コラム】ターミナルでコピー＆ペーストするには？

ターミナルで表示されている内容をコピーしたいときは、コピーしたい部分をマウスで選択するだけです。（図 2.16）選択してから Ctrl+c を押す必要はありません。

^{*7} cat は猫ではなく「conCATenate files and print on the standard output」の略です

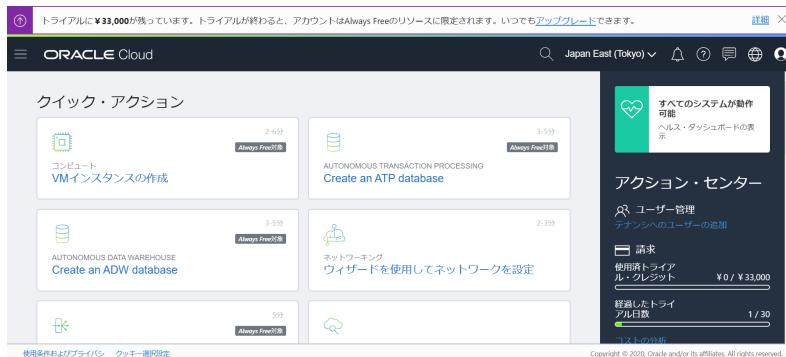


▲図 2.16 マウスで選択するだけでコピーできる

逆にコピーした内容をターミナルへペーストしたいときはターミナル上で**右クリック**するだけです。Ctrl+pは使えないで注意してください。

2.2 コンピュートでサーバを立てる

コンソールにサインインしたら、さっそくサーバを立てましょう。[VM インスタンスの作成] をクリックします。(図 2.17)



▲図 2.17 [VM インスタンスの作成] をクリック

[インスタンスの命名] に [startSSLInstance] と入力します。(図 2.18)



▲図 2.18 [インスタンスの命名] に [startSSLInstance] と入力

その下の [オペレーティング・システムまたはイメージ・ソースを選択します] は、何も変更せずそのまま構いません。

パソコンには OS という基本ソフトが入っていて、Word や Excel、Chrome といったソフトはその OS の上で動いています。皆さんのパソコンにも「Windows 10」や「Mac OS X Lion」などの OS が入っていますよね。

そしてパソコンと同じようにサーバにも「Linux」や「Windows Server」といったサーバ用の OS があります。サーバを立てるときには Linux を選択することが多いのですが、この Linux の中にもさらに「RHEL (Red Hat Enterprise Linux)」や「CentOS」、「Ubuntu」などいろいろなディストリビューション（種類）があります。

今回は、OS はデフォルトの [Oracle Linux 7.7] を使います。Oracle Linux なら Oracle Cloud のツールがあらかじめ入っていますので、**Oracle Linuxでサーバを立てるときは OSはOracle Linuxにすることをお勧めします**。Oracle Linux は Red Hat 系のディストリビューションですので、RHEL や CentOS のサーバを使ったことがある方なら違和感なく使えると思います。

Oracle Linux には 2020 年 1 月時点です

- Oracle Linux 6.10
- Oracle Linux 7.7

の 2 種類があります。名前のとおり、Oracle Linux 6.10 は CentOS 6 と同じ RHEL6 系、Oracle Linux 7.7 は CentOS 7 と同じ RHEL7 系なので、使い勝手はほぼ同じです。本著では Oracle Linux 7.7 を使用します。

[SSH キーの追加] は、[SSH キーの貼付け] を選択して、そこに先ほどメモしておいた公開鍵をペーストします。公開鍵は改行を含まず、先頭の「ssh-rsa」から末尾の「<ユー

ザ名>@<ホスト名>」のようなコメントまでで、まるごと 1 行です。(図 2.19)



▲図 2.19 [SSH キーの貼付け] を選択してメモしておいた公開鍵をペースト

公開鍵をペーストしたら [作成] をクリックします。"Out of host capacity."出てめっちゃ笑ってる。

https://docs.oracle.com/cd/E83857_01/get-started/subscriptions-cloud/csgsg/sign-your-account-oracle-cloud-website.html

2.3 ドメイン名の設定

2.4 まずは HTTP でサイトを公開

2.5 証明書を取得しよう

2.5.1 秘密鍵を作ろう

2.5.2 CSR を作ろう

2.5.3 証明書の取得申請

2.5.4 取得した証明書をサーバに置こう

2.6 HTTPS でサイトを公開

第3章

基本

3.1 SSL ってなに？

3.2 TLS ってなに？

3.3 SSL と TLS の違いは？

3.4 SSL と SSH って似てる？ 何が違うの？

3.5 HTTPS で始まるページで鍵のマークが壊れて表示された

3.6 種類

3.6.1 SSL サーバ証明書

3.6.2 SSL クライアント証明書

3.7 どんなシーンで使われている？

3.8 SSL 証明書は全然違う 2 種類の仕事をしている

3.8.1 Web サイトで送受信する情報を暗号化すること

3.8.2 Web サイト運営者の身元を証明すること

3.9 鍵マークが壊れるケース

3.9.1 すべて HTTP で通信している₄₆

3.9.2 HTTPS だけど一部が HTTPS じゃないとき

画像と CSS の指定が絶対パスだった

3.10 ウェブページが表示されるまで

あとがき

数ある技術書の中から「SSL をはじめよう」を手に取ってくださったあなたに感謝します。

2020年2月
mochikoAsTech

PDF 版のダウンロード

本著（紙の書籍）をお買い上げいただいた方は、下記の URL から PDF 版を無料でダウンロードできます。

- ダウンロード URL : <https://mochikoastech.booth.pm/items/xxxxxx>
- パスワード : **xxxxxx**

Special Thanks:

- ネコちゃん

レビュー

- Takeshi Matsuba

参考文献

- ぶんけん
 - ??

著者紹介

mochiko / @mochikoAsTech

元 Web 制作会社のシステムエンジニア。技術書典で出した本がきっかけで、テクニカルライターの仕事を始めた。モバイルサイトのエンジニア、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したのち、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。技術書典 4,5,6 で頒布した「DNS をはじめよう」「AWS をはじめよう」「技術をつたえるテクニック」「技術同人誌を書いたあなたへ」は累計で 7,800 冊を突破。

- <https://twitter.com/mochikoAsTech>
- <https://mochikoastech.booth.pm/>
- <https://note.mu/mochikoastech>
- <https://mochikoastech.hatenablog.com/>

Hikaru Wakamatsu

表紙デザインを担当。

Shinya Nagashio

挿絵デザインを担当。

SSLをはじめよう

証明書の発行からトラブルシューティングまで

2020-02-29/2020-03-01 技術書典 8 初版

著 者 mochikoAsTech

デザイン Hikaru Wakamatsu / Shinya Nagashio

発行所 mochikoAsTech

印刷所 日光企画

(C) 2020 mochikoAsTech