

パブリッククラウドを活用したLinux 学習環境構築



株式会社ALJ
プロジェクトリーダー
廣出 啓太

■講師プロフィール



- ◆ 名前 : 廣出 啓太(ひろで けいた)
- ◆ 出身 : 愛知県出身
- ◆ 年齢 : 26歳
- ◆ 興味ある分野 : セキュリティ(主に暗号系)

- ◆ 使用言語 : Python, VBA, Google AppScript, JavaScript

- ◆ 使用環境 : Linux, Windows, Cisco, Yamaha, Fortigate, Seiko, Nokia, AWS, MS Azure

- ◆ 資格 : ネットワークスペシャリスト, 応用情報技術者認定, CCNA, 危険物取扱者乙4,乙3

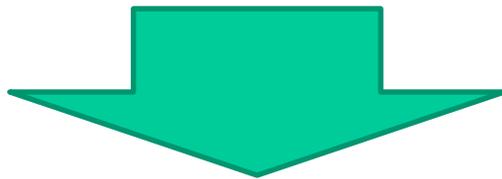
- ◆ 経歴
 2021年から新卒で入った会社でネットワークエンジニアとしてのキャリアを開始。
 エンタープライズ、モバイルコアの設計構築案件に従事したのち
 フリーランスを経て、株式会社エーエルジェイに入社。
 休日は専門学校講師として学生にCCNA, Linuxを教育し、
 本業のネットワークエンジニアの仕事に加え、PowerAutomate, Python , GASを使用
 した不動産会社、保険代理店のRPA開発を兼任。

今回のテーマ

パブリッククラウドを活用した Linux学習環境構築

- **LinuCについての説明**
- **パブリッククラウドについての概要説明**
- **オンプレミス・パブリッククラウドの違い**
- **AWSを使用したLinux環境構築方法(Alma Linux)**
- **AWSのインスタンス上へのssh,scpなどのログイン方法**
- **VScodeとクラウドのインスタンスの接続**
- **Appendix**
- **まとめ**

- パブリッククラウドサービスを用いたLinuC/Linux学習環境構築方法。
- Amazon Web Service(AWS)のEC2サービスを活用し、リモート操作で実際にLinux環境を構築。
- WinscpやVS Codeを使用し、クラウド上のインスタンスへの簡易的なアクセス方法。



- ✓クラウドサービスを利用したLinux学習環境が構築できる
- ✓セキュアなリモート接続でクラウドサービスを利用する
- ✓クラウド環境上に好みの環境を簡単に構築できる

■Linuxとは

Linuxは、オープンソースの**オペレーティングシステム (OS)** です。

オペレーティングシステムとは、CPU、メモリ、ストレージなどのシステムのハードウェアとリソースを直接管理するソフトウェアのことで、アプリケーションとハードウェアの間に位置し、すべてのソフトウェアと作業を実行する物理リソースをつないでいます。

(RedHat. Linux とは.Linuxについて参照)

簡単に言うと・・・

✓ 皆さんが普段使ってる**Windows**や**MAC**などの仲間です

例として出すのであれば、普段みなさんが使用されてるPCも
 動画再生ボタンを押したら、画面が動いたり、音楽が流れたりしますよね
 いろいろ**物理的なものとの仲介屋**をして**くれているシステム**ということですよ



■ 普段身の回りに隠れているLinux

- Android(携帯)
- テレビ
- コピー機
- ネットワーク機器 など。



実は意識していないだけで**Linux**というOSは普段から身の回りに多数存在しています

また今回学習する**クラウドサービス**のほとんどがLinuxベースで運用されています。



■LinuCとは

クラウド／DX時代のITエンジニアに求められるシステム構築から運用管理に必要なスキルを証明する技術者認定です。

✓クラウド活用に役立つスキルの習得

- オンプレミス／仮想化・コンテナを問わず様々な環境下でのサーバー構築
- 他社とのコラボレーションの前提となるオープンソースへの理解

✓習得できるスキルが実践的

問題作成にはトップエンジニアも参加するコミュニティ内の意見を取り込むことで、本当に必要な内容を網羅的に盛り込んでいます。

✓上流工程を担うアーキテクトの領域までカバー

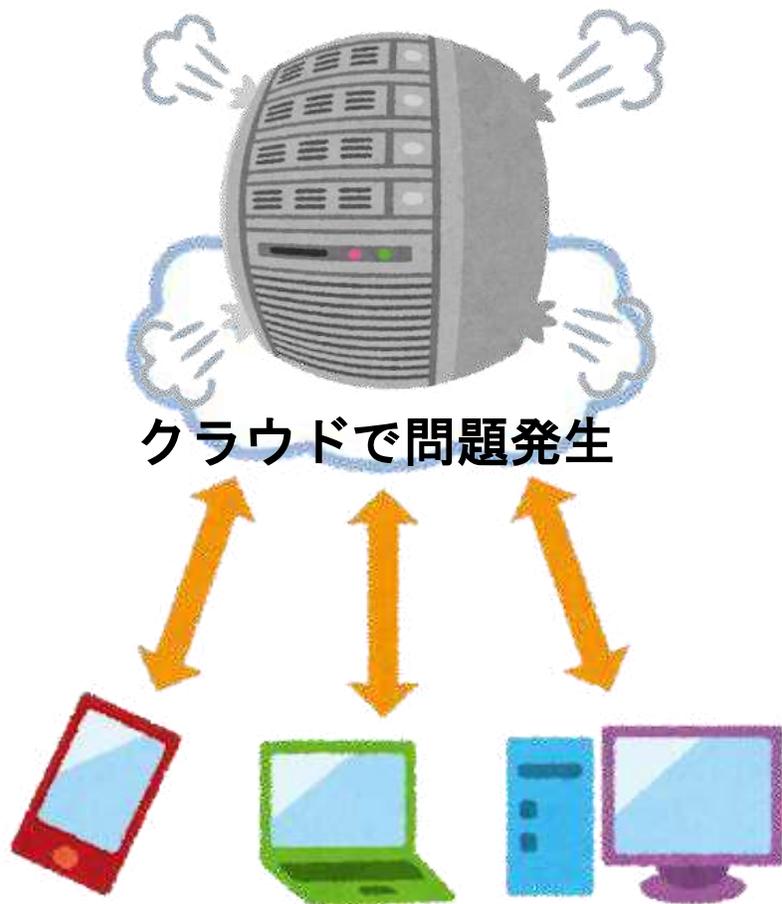
システムの運用管理からアーキテクチャ設計までの4つのレベルをひとつずつ習得していくことで、活躍できるエンジニアとして必要なスキルを網羅的に身につけていくことができます



■Linuxの知識がトラブルや様々な問題の解決に役立つ

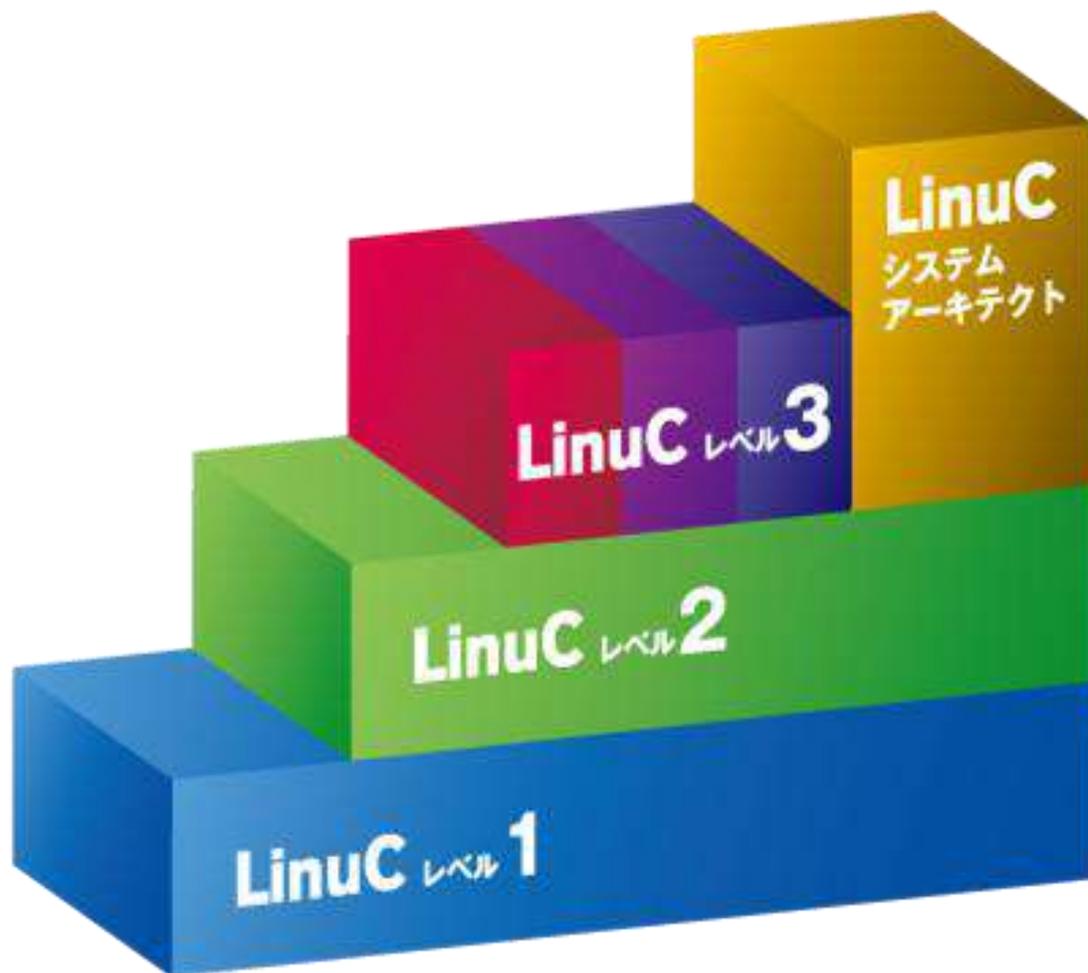


Linuxの知識がない人



Linuxの知識がある人

LinuCは、サーバーの運用管理からアーキテクト設計まで、システム開発・運用に必要な知識とスキルを体系立てて習得することができます。



LinuC システムアーキテクト

ITプロジェクトを成功に導く上級エンジニア

SA01試験

SA02試験

LinuC レベル3

高度な技術力を備えた特定分野のスペシャリスト

304試験 (仮想化&高可用性)

300試験
(現在環境)

303試験
(セキュリティ)

LinuC レベル2

仮想マシン・コンテナを含むLinuxシステム、ネットワークの設計・構築

201試験

202試験

LinuC レベル1

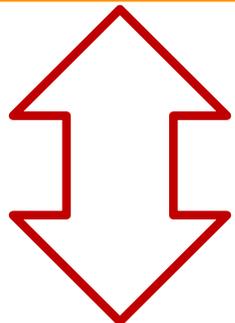
物理/仮想Linuxサーバーの操作・運用

101試験

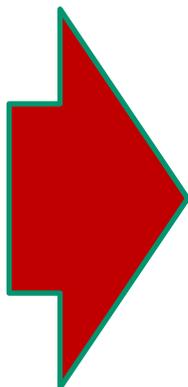
102試験

- LinuCにはクラウドを活用できるITエンジニアに必須の技術がまとまっています。

AWSなどの
パブリッククラウド
を活用するための技術



間
が
欠
け
て
い
る
状
態



AWSなどの
パブリッククラウド
を活用するための技術

仮想マシン／コンテナ技術
、クラウドセキュリティ、
アーキテクチャ、ほか

オンプレミスの
サーバーサイドLinux技術

【今まで／その他】

LC LinuC Version 10.0



■ アカデミック認定校に通う

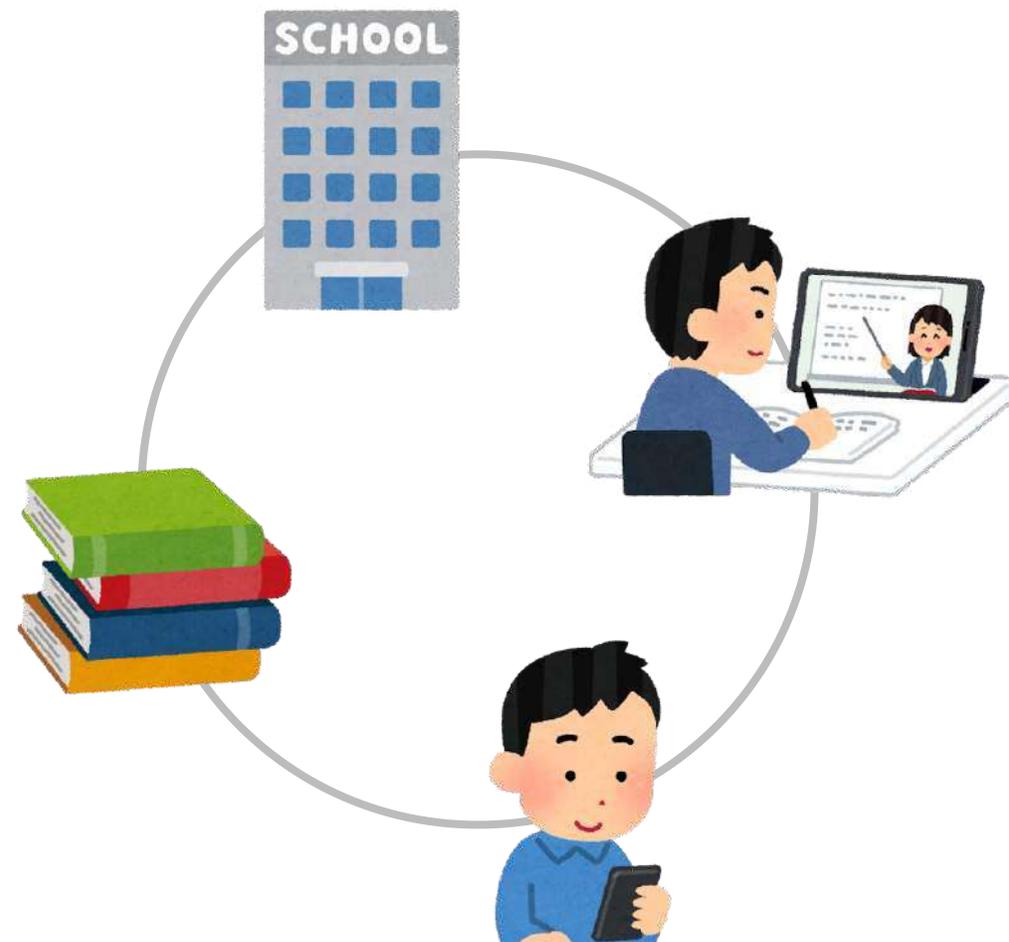
- 大学、専門学校
- PCスクール
- 企業研修機関

■ LPI-Japanの教材・サービスを活用

- Linux標準教科書、学習補助教材
- 技術解説セミナー、同セミナー動画

■ 市販の認定教材を活用

- 書籍（電子書籍含む）
- オンライン動画・問題集
- スマートフォンアプリ



学習

- LinuC v10.0をしっかりと学べる **LPI-Japan認定校**
- LinuC v10.0を体系的に学べる **LPI-Japan認定教材** (書籍、動画、電子教材)



反復学習



演習

- LinuC v10.0の出題範囲の理解を確認できる **LPI-Japan認定問題集** (書籍、Web教材、スマホアプリ)



無償補助教材

- 最新技術差分が効率良く学べる **LinuC Version10.0 新出題範囲学習補助教材**



- Linux基礎学習に最適な入門書 **Linux標準教科書**



- Linuxの学習環境が作れる **学習環境構築ガイド**



- 出題形式になれるための **サンプル問題集**

- 理解しにくい分野の理解に最適 **LinuC技術解説セミナー ライブ** 又は **youtube配信**



- 素朴な疑問の理解に役立つ **豆知識**

学習環境の構築 (Linux環境で、手を動かしながら学ぶ)

■ LPI-Japan提供の学習環境構築ガイドを活用

- LinuCを受験される方が出題範囲を学習しやすくするために、Linux実習環境を構築するための方法を案内するもので、以下の4種類の環境の構築方法を説明している。
 - Linux専用コンピュータを構築
 - 外付けSSDにLinuxをインストールして起動ディスクにする
 - WindowsやMac上にLinux仮想環境を構築
 - Windows 10以降ならWSL2を利用



■ パブリッククラウド上にLinux環境を構築



■ オンラインのクラウド型学習環境を利用

The image shows the Katacoda website and its interactive terminal interface. The top part is a screenshot of the website's homepage, which features the O'Reilly logo and navigation links. The main content area highlights "Learn new technologies using real environments right in your browser" and "Katacoda enhances your technical sales, training and internal education process". Below this, there are three icons representing learning, testing, and self-paced learning. The bottom part of the image shows a browser window displaying the Katacoda interface for an "Ubuntu 20.04 Playground". The terminal window shows the following output:

```

$ uname -a
Linux host01 5.4.0-52-generic #57-Ubuntu SMP Thu Oct 15 10:5
7:00 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
$
  
```



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

学習環境	難易度	メリット	デメリット
Linux専用コンピュータ	★★★	制約無しに網羅的に学べる。	学習専用PCの調達が必要。
外付けSSDにLinuxインストール	★★★	既存OS環境を維持しながら専用マシンと同じLinux環境が構築できる。	Linux OS使用時は既存OS環境が使えない。
WindowsやMac上にLinux仮想環境	★★	既存OS環境と併用でき、完全に切り離しも可能。複数の仮想マシンの作成も可能。	構築する内容によっては、ホスト環境への負荷が大きくなり、パフォーマンスが落ちる。
Windows 10以降のPCでWSL2	★★	Windows環境と混在、併用可能。 Windows10以降の標準機能なので無償で利用可能。	使用するPCのCPUに仮想化支援機能が必要。公式にはRedHat系ディストリビューションが無い。
パブリッククラウド上にLinux環境	★	OSインストールが不要ですぐに使える。 様々なLinux OSが使える。既存OSマシンのWebブラウザやターミナルソフト経由で使える。	クラウドサービスの内容、設定の把握が必要。無償枠もあるが基本的に有償。
オンラインのクラウド型学習環境	★	OSインストールが不要でWebブラウザだけですぐに使える。無償で使える。	環境が保存ができない。時間制限がある。一部制約がある。

LinuCを取得するなら、実機、仮想、クラウド、それぞれを可能な限り試してみよう

パブリッククラウドについての概要説明

■クラウドとは何ですか？

めちゃくちゃ固く言うと・・・

クラウド（クラウドコンピューティング）とは、インターネットを通じてコンピュータのリソース（サーバー、ストレージ、ソフトウェアなど）を利用できる技術です。

従来の**オンプレミス型**（自社で設備を管理する方式）に比べ、柔軟性やコスト削減が期待できます。

クラウドの特徴として以下のようなものがあります。

- オンデマンドで利用可能 - 必要なときに必要な分だけリソースを使える。
- スケーラビリティ - ユーザーの需要に応じて自動的にリソースを増減できる。
- コスト削減 - 自社で設備を持たずに、利用した分だけ料金を支払う方式（従量課金）。
- アクセスの自由度 - インターネット経由でどこからでも利用できる。



■代表的なクラウドって何があるんですか？

主に使用されているのは以下のプラットフォームです。

- AWS(Amazon Web Service)
- Microsoft Azure
- Google Cloud Platform など

ほかにも富士通クラウド、さくらのクラウドであったり様々な企業が自社クラウドサービスを展開しています。



■ どういったところでクラウドって使用されているんですか？

ものすごく身近な場所でいうと

- iCloud
- Google Drive
- Dropbox
- Gmail, Outlook, OneDrive
- Steam など

気にしていないだけで実はいろいろな場所で私たちはクラウドサービスの恩恵を浴びています。



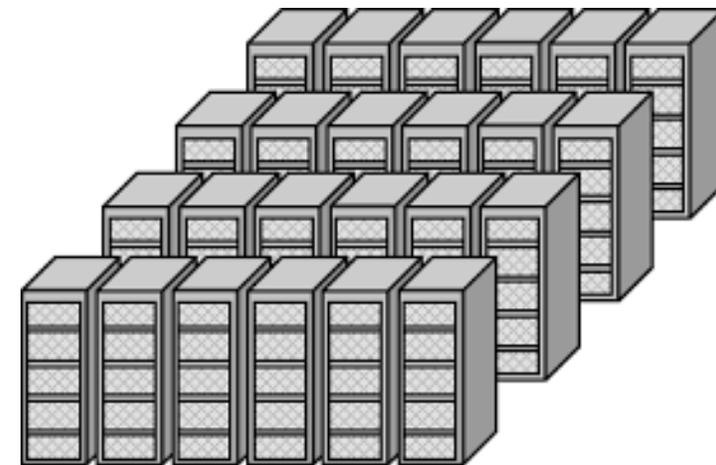
■パブリッククラウドっていうけどプライベートもあるんですか？

実は・・・あります！

プライベートクラウドは社内のデータセンターなどに仮想化されたクラウド基盤を構築し閉鎖的なクラウド環境を疑似的に作成する技術です。

そのため、**パブリッククラウド**よりセキュアですがその分、運用負担や、初期コストが高くなってしまいます。

拡張性も乏しいため、企業のニーズに応じて**ハイブリッドクラウド**という選択肢を取ることも一般的で**プライベートクラウド**のセキュリティと**パブリッククラウド**の柔軟性を組み合わせて最適な環境を構築することが可能になります。



■じゃあ結局パブリッククラウドってなんですか？

簡単に言うと、めちゃくちゃでかい企業が提供している無制限に近い物理リソースを柔軟に拡張し使用することができるめちゃくちゃ便利なサービス。

プライベートクラウドは結局リソース追加する場合は、SSDなどの物理リソースの追加が必要になるところ、**パブリッククラウド**はそれが不要！

だからこそ、それぞれの良いところを使い分けていきたいと思いますというお話になります！



オンプレミス・パブリッククラウドの違い

■ オンプレミスとパブリッククラウドって何が違うんですか？

● オンプレミス

自社でサーバ、ネットワーク機器を管理し運用する方法
初期コストは高いが柔軟な構成変更が可能

● クラウド

クラウドサービスを使用し、必要なリソースだけ使用可能
初期コストも安く、可用性が高く運用負担が軽減。



オンプレミスはエンジニアがインフラ部分をすべて用意し、環境を構築する必要がある！

■ 表にすると以下のような感じ

項目	オンプレミス	クラウド
初期コスト	サーバ、ネットワーク、ストレージの機器購入の費用が非常に高つく	機器購入の費用が掛からず、使用した分だけの従量課金
管理	自社管理が必要で、機器が壊れていないかなどの定期点検が必要	クラウド事業者(AWS, Microsoftなど)がやってくれる
拡張性	サーバを置くスペースなどの物理的な制約はあるが、自由に拡張できる	基本的には無限に拡張することができる
セキュリティ	自社で管理する	クラウド事業者が管理する

■ オンプレミスで用意する機器ってどんなものがあるんですか？

オンプレミスの環境構築するためには非常に多くのネットワーク機器、サーバ、ストレージなど多くの構成要素が必要になります。

代表的なベンダーとして以下のベンダーがあげられます。

- Cisco Systems
- Dell Technologies
- F5 networks
- Paloalto Networks など

またこれらの機器たちもLinuxベースで構成されています！



ARISTA

FORTINET

JUNIPER
NETWORKS



■ メリット/デメリットをわかりやすく教えてください！

項目	オンプレミス	クラウド
メリット	独立したネットワークを持つため、非常に高いセキュリティを保てる	簡単にクラウドリソース調達することができる
	自由にネットワークなどを変更できるのでカスタマイズ性が高い	初期コストが不要で簡単にビジネスを開始することができる
	法規制やコンプライアンス対応が容易	機器の管理が不要
	すべての機器の完全な管理権限を持てる	サーバなどの状態を簡単に把握することができる
デメリット	初期コストが高い	クラウド事業者のインフラに依存するため、そちらで障害が発生した場合に巻き込まれる
	拡張するときには機器を調達し物理リソースの追加が必要	セキュリティはクラウド事業者による
	性能向上させるためには時間とコストが必要	継続的に利用すると、コストが非常に高くなる

■ 結論

オンプレミスとクラウドのそれぞれにメリット・デメリットがあり
どちらが優位性があるかなどはなく、状況に合わせた使い分けが必要になります。

そのため、どちらかだけを重点的に勉強すればよいという話ではなく
ITエンジニアとして幅広い知識を得て、最適なソリューションを提案できるようになる必要があります！

今回はクラウドを使用した**Linux環境構築**を勉強しますが、実際に物理PCにインストールして勉強するのも、今後のためには学習をお勧めします。

オンプレミスの**Linux環境構築手順**も**LinuC**のページにて公開しておりますので
興味ある方はそちらを参考にいただければと思います

詳しくはこちら！  

AWSを使用したLinux環境構築方法 (Alma Linux)

■ AWSアカウント作成の流れについて教えてください！

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

1. AWSアカウントの作成
2. 連絡先情報の入力
3. 請求情報の入力
4. SMSまたは音声電話による本人確認
5. AWSサポートプランの選択

全ての手続きを終えれば、数分ほどで登録メールアドレス宛に確認のメールが届き、「AWS マネジメントコンソールにお進みください」ボタンをクリックするとAWSの利用を開始できます。

3.では有効なクレジットカードまたはデビットカード情報を入力する必要がありますが、AWS 無料利用枠内の利用に対し料金は発生しません。5.では無料枠の利用でも有償のサポートを選択すると課金がスタートするので注意してください。



■ ユーザの種類について教えてください！

AWSには、すべての権限を行使することができる「**ルートユーザ**」と通常の運用で扱う「**IAMユーザ**」の2種類のユーザがいます。それぞれの**特徴**は以下のようになっています。

ルートユーザ	IAMユーザ
AWSアカウント作成時に自動生成	アクセス権限を個別設定可能
すべての権限を持つ（制限なし）	必要な操作のみ許可（例：管理者、開発者、監視者など）
重要な設定変更や請求管理が可能	ルートユーザを使わずに安全に運用
推奨される使用頻度は最小限（セキュリティの観点から）	セキュリティを強化しつつ、チームで適切に管理できる

※ユーザすべてに対して管理者権限を与えてしまうと、運用中に**重大なインシデント**が発生してしまう恐れもあるので、適切なユーザ管理が必要になります。

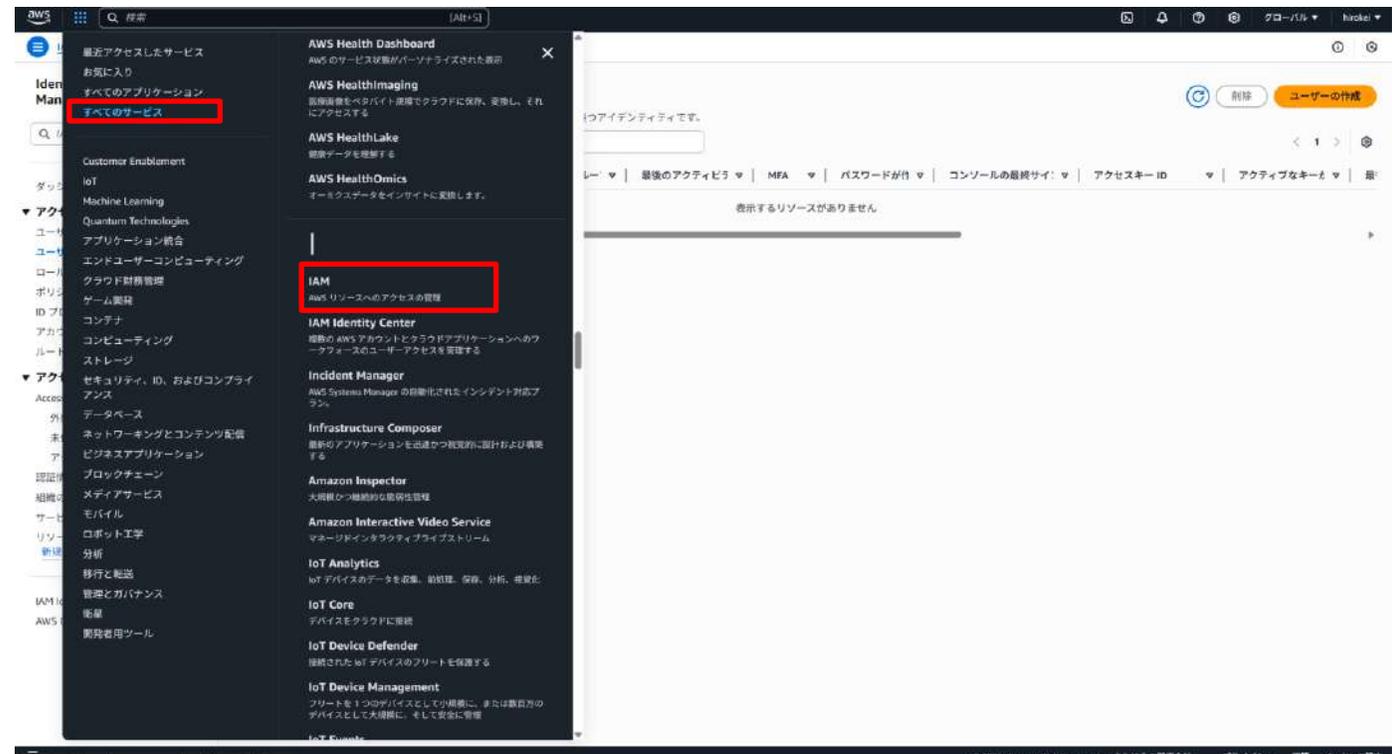
■ どうやってユーザを作成すればいいですか？

AWSにはAWS Identity and Access Management (IAM)



と呼ばれるサービスが提供されており、そちらでユーザを作成して適切な権限を与えることができます。

「すべてのサービス」 > 「IAM」
からユーザ作成画面に移動することができます。



■ 今回の環境構築に使うユーザは何ですか？

今回は実際の運用という名目で「**ルートユーザ**」では実施せず、初めに「**IAMユーザ**」を作成した後にAWS上にLinuxの学習環境を作成していきたいと思います！



ルートユーザ



IAMユーザ

また「**IAMユーザ**」は複数作成でき、様々な権限を持ったユーザを作成でき、「**開発者**」「**運用者**」「**監視者**」など用途に合わせてユーザを作成することができます。

■ それぞれの設定値は何ですか？

今回は「**linuxUser**」ユーザを新規に作成します。ユーザには様々な権限を付与できますが、今回はあくまでテスト用ということで、契約関連以外の強い権限を持つ「**AdministratorAccess**」を付与します。

AWSでは多要素認証 (MFA) の使用が推奨されています。**rootユーザ**だけでなく、**IAMユーザ**にも設定するようにしましょう。

作成した「**linuxUser**」ユーザで**AWS**にサインインします。

項目	内容
ユーザー名	LinuxUser
AWS認証情報タイプ	パスワード - 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	カスタムパスワード(値は任意)
パスワードのリセットが必要	解除
アクセス権限	既存ポリシーで AdministratorAccessのみ
タグ(キー)	linuxstudy
タグ(値)	for Linux Study

■ 何を使用して、Linuxの学習環境を構築すればいいですか？

AWSでLinux環境を構築するなら、**Amazon EC2** を使うのが一般的です。

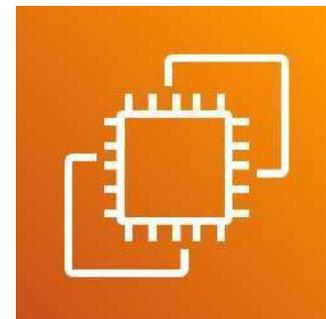
EC2は「クラウド上で仮想サーバを簡単に作れるサービス」です。これを使うと、物理サーバを購入する必要がなく、好きなときにサーバを作成・削除 できます。

また、**EC2**では セキュリティ設定やネットワーク構成 が簡単にでき
ストレージ管理も自由に行えます。

サーバの負荷に応じて 増やしたり減らしたり できるので
事前に通信量を明確に予測する必要がありません。

EC2は インフラの管理だけを提供する**IaaS (Infrastructure as a Service)** という
サービスで、OSの設定やアプリのインストールはユーザーが行います。

簡単に言うと、**EC2**は「クラウド上で自由にサーバを使える便利なサービス」です！



Amazon Elastic
Compute Cloud
(Amazon EC2)

項目	内容
インスタンス	仮想コンピューティング環境(=仮想サーバー)
AMI(Amazon Machine Image)	サーバー構築に必要なインスタンス用テンプレート
インスタンスタイプ	インスタンス用の様々な構成(CPU、メモリ、ストレージなど)
キーペア	インスタンス用の安全なログイン情報(認証鍵)
インスタンスストアボリューム	インスタンスを停止、休止、または終了するときに削除される一時的ストレージボリューム
EBSボリューム	データ用の永続的ストレージボリューム
リージョンおよびアベイラビリティゾーン	インスタンスやEBSボリュームなどのリソース用の複数の物理的な場所(データセンター)
セキュリティグループ	インスタンスのプロトコル、ポート、ソース IP 範囲を指定できる仮想ファイアウォール
Elastic IP	動的クラウドコンピューティング用の静的なIPv4アドレス
タグ	EC2リソースに割り当てることができるメタデータ
仮想プライベートクラウド(VPC)	AWS内で論理的に分離されたユーザー独自の仮想ネットワーク

● リージョン

AWSのデータセンターは 世界26地域 に分かれておりこれを「リージョン」と呼びます。

リージョンごとに独立したインフラを持つため地理的に最適な場所を選んでリソースを配置できます。

● アベイラビリティゾーン(AZ)

リージョン内には 複数のAZ(独立したデータセンター)が存在し、インスタンスを分散配置できます。

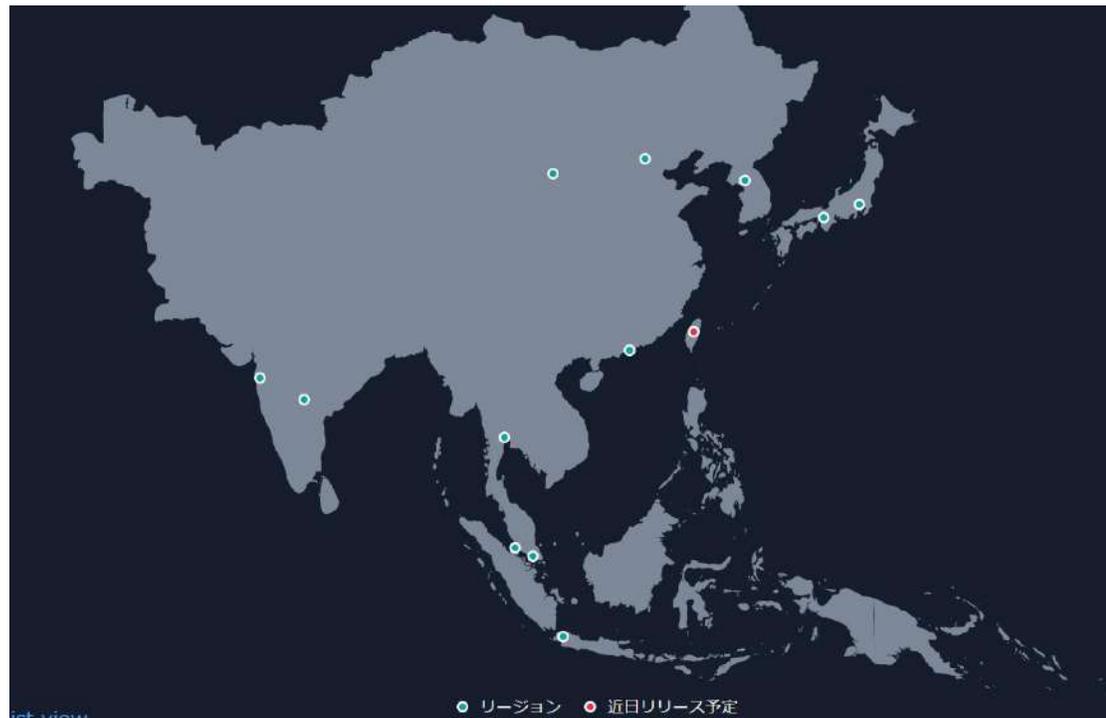
これにより 障害が発生しても他のAZが処理を継続できるようになります。

インスタンスの配置は 手動または自動で選択 可能です。

● 日本のリージョン

日本には **東京(ap-northeast-1)**と **大阪(ap-northeast-3)**の 2つのリージョン があります。EC2などのリソース作成時には、リージョンを選択し、適切な環境を構築できます。

AWSではリージョンとAZを活用することで、「**障害耐性**」や「**パフォーマンス**」を向上 させることが可能です！



※2025/5現在
Asia pacific

■ 今回使用するディストリビューションについて

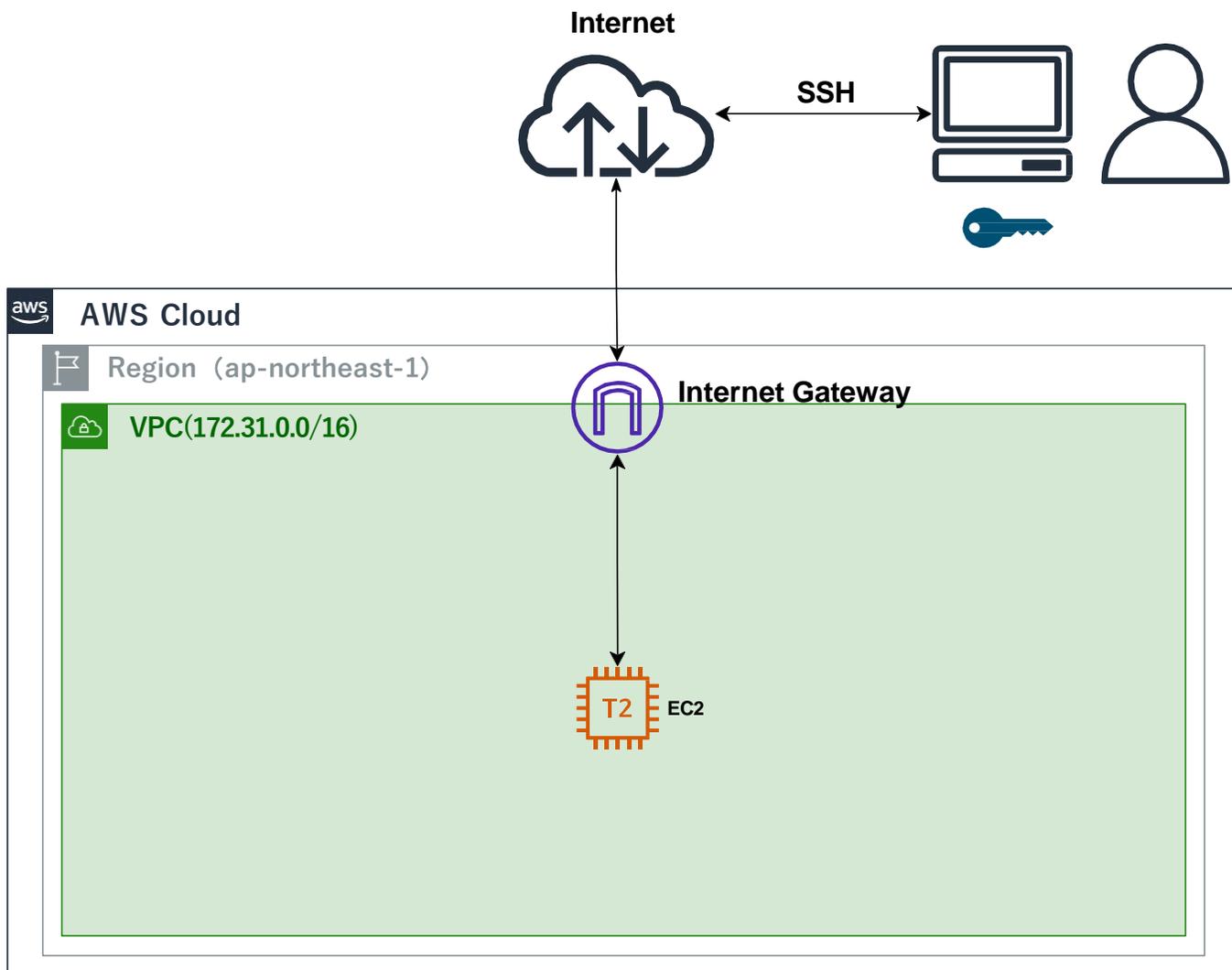
今まで広く使用されてきた**CentOS7**が2024年6月にサポート終了したことに伴い、現在、ディストリビューション選びに困っている方が非常に多くなってきていると思います。本セミナーでは**Alma Linux**を使用して環境構築を実施していきます。

AlmaLinuxは、以下の特徴から**CentOS**の後継として注目されています。

- **RHEL互換**: 既存アプリ・ソフトがそのまま利用可能
- **無料**: 商用ライセンス不要でコスト削減
- **活発なコミュニティ**: グローバルな開発・サポート
- **定期アップデート**: 最新のセキュリティ対策
- **高い安定性**: 企業環境にも適した運用



今回の構成図

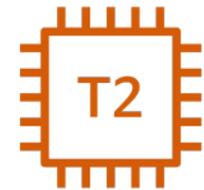


作成するインスタンスの構成

項目	設定内容
AMI	Alma Linux (RedHat互換)
インスタンスタイプ	t2.micro (無料利用枠)
リージョン	東京
インスタンス数	1
ネットワーク	デフォルト
ストレージ	8GB汎用SSD(gp3)
セキュリティグループ	SSH(22) HTTP(80)



AMI

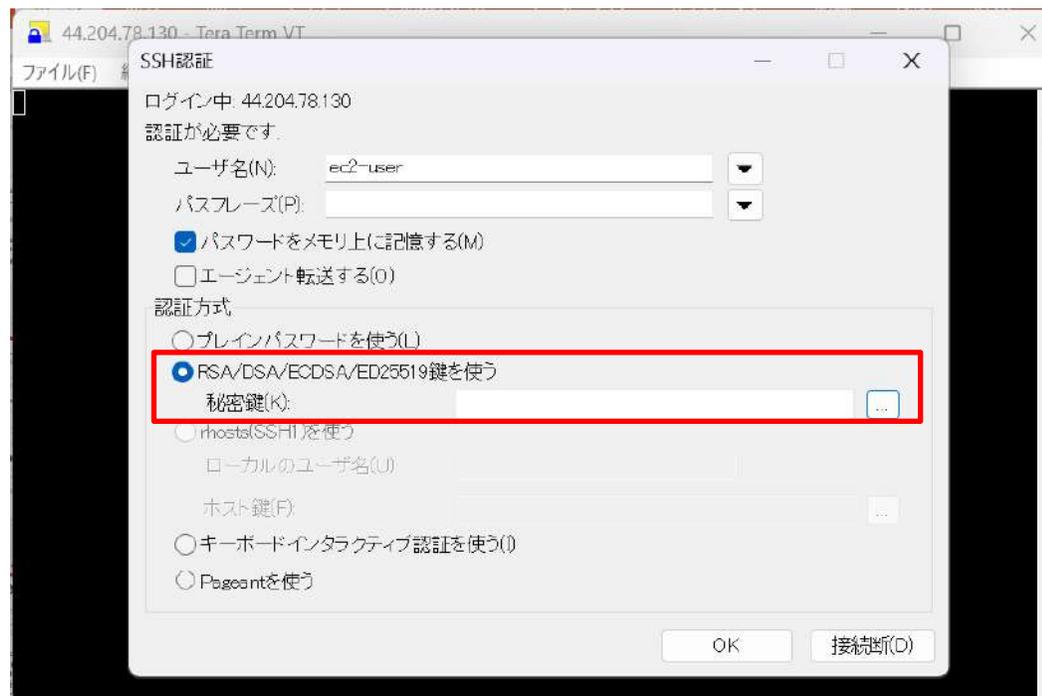


T2 instance

- 先ほどダウンロードしたpemファイルを使用してログインする

接続手段

- ① TeraTermを使用して、クラウド上のインスタンスへログインをする。
- ② 「RSA/DSA/ECDSA/ED25519鍵を使う」から先ほどダウンロードしたpem(秘密鍵)を選択する
- ③ 「OK」を押下し、ログイン完了！

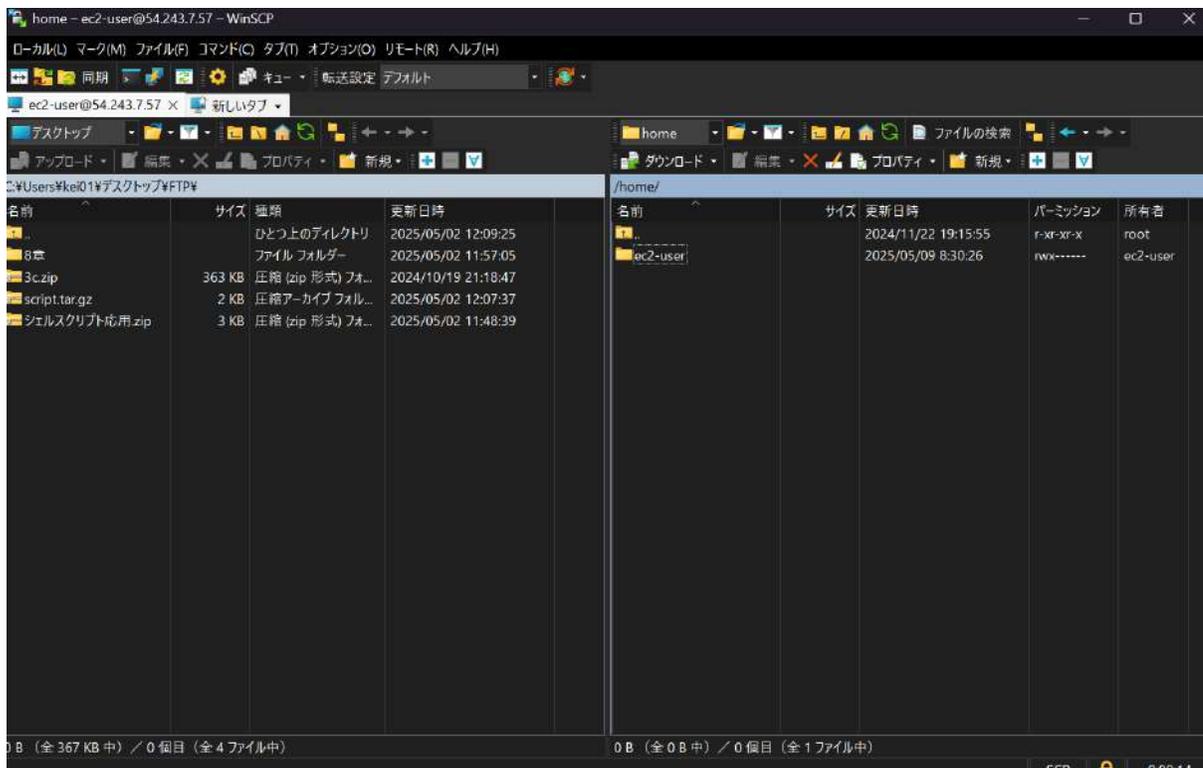


AWSのインスタンス上へのssh,scpなどの ログイン方法

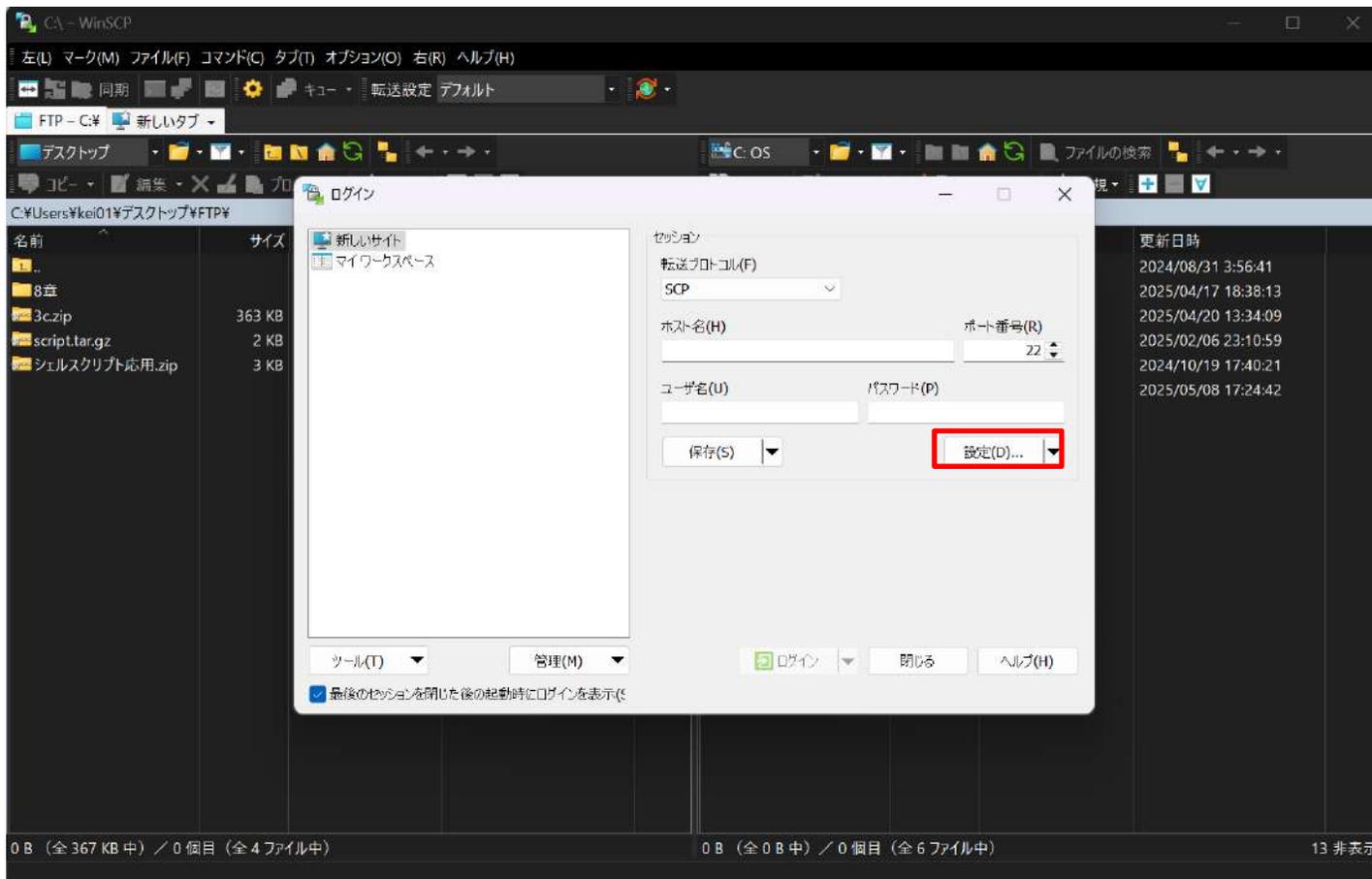
■ WinSCPについて

WinSCPは「SFTP,FTP,SCP,WEBDAV,AmazonS3」の通信をGUIで直感的に実行することができるツールです。

WinSCPを使用すると、AWS上のLinuxインスタンスへ簡単にファイルなどを送信することができます。

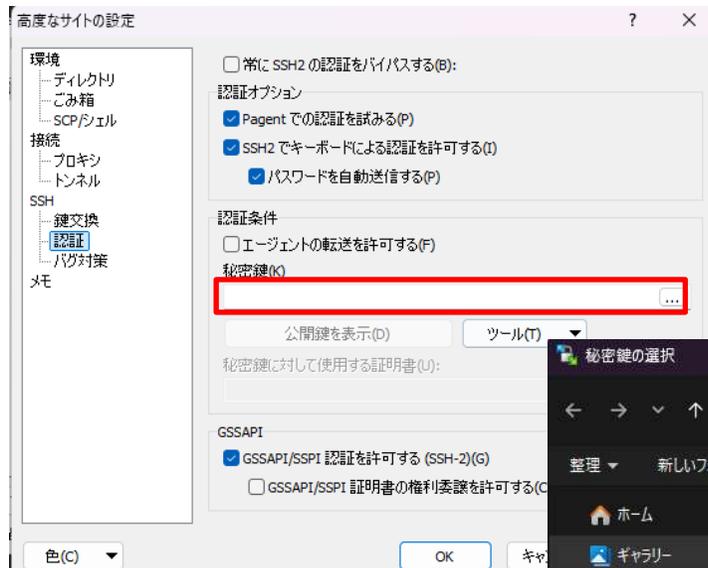


■ WinSCPってどうやって使用するのですか？

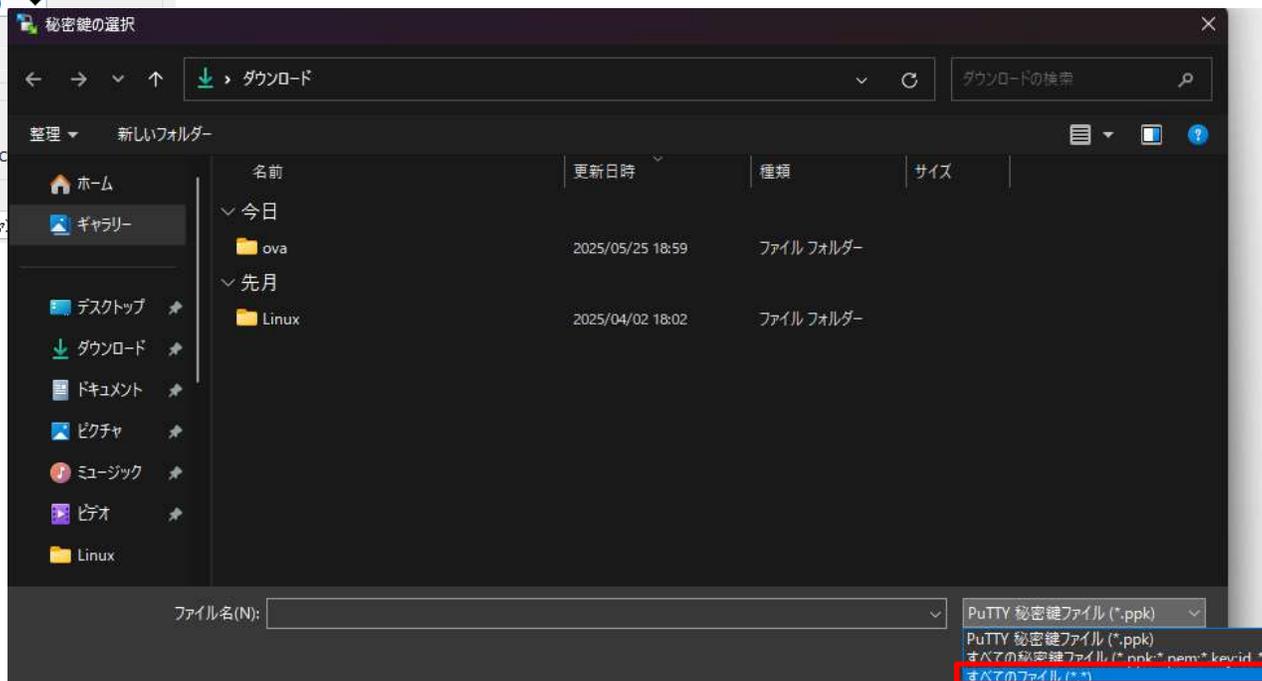


まずログインする前に
先ほどダウンロードした鍵を指
定するために赤枠の「設定」を
クリックします！

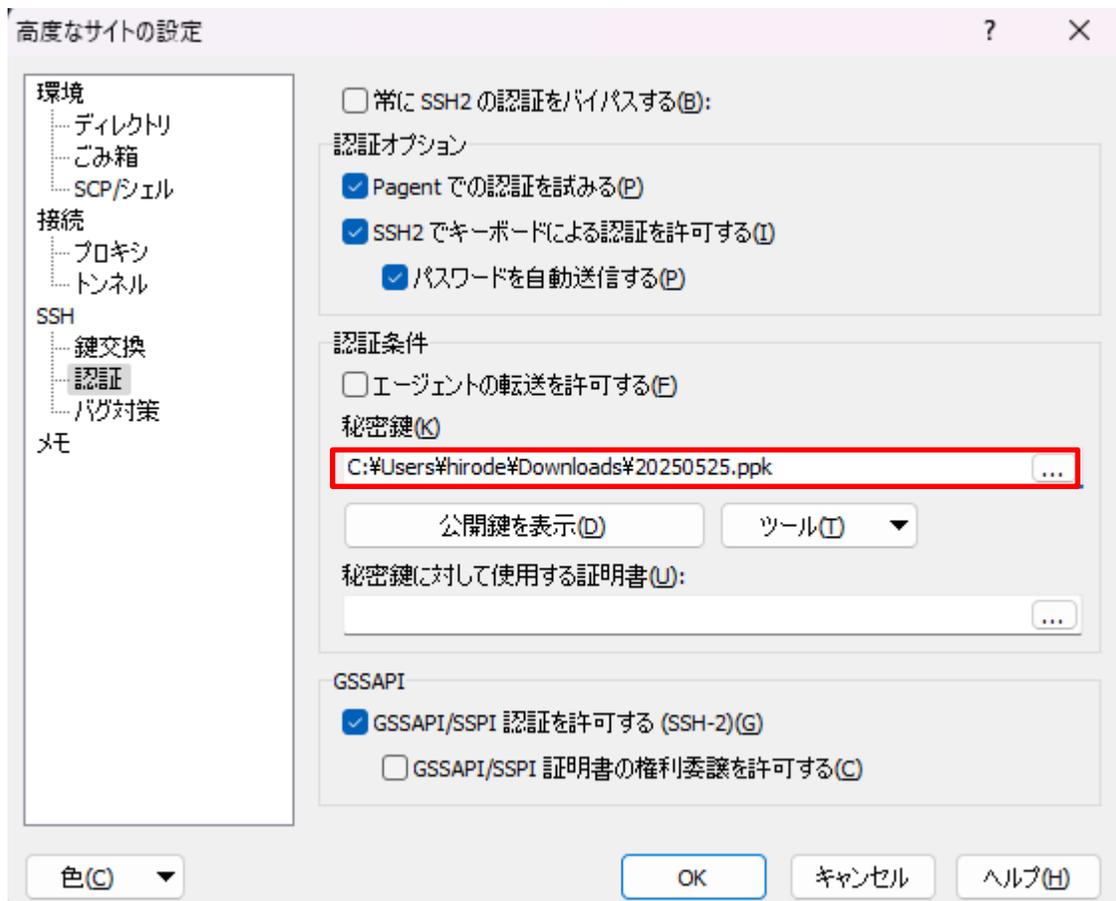
■ WinSCPの鍵認証の指定方法について



認証のタブから**秘密鍵**を選択し
 エクスプローラーが開いたら**すべてのファイルを表示**
 し、先ほどの鍵を**putty形式**に変換します。



■ WinSCPの鍵認証の指定方法について



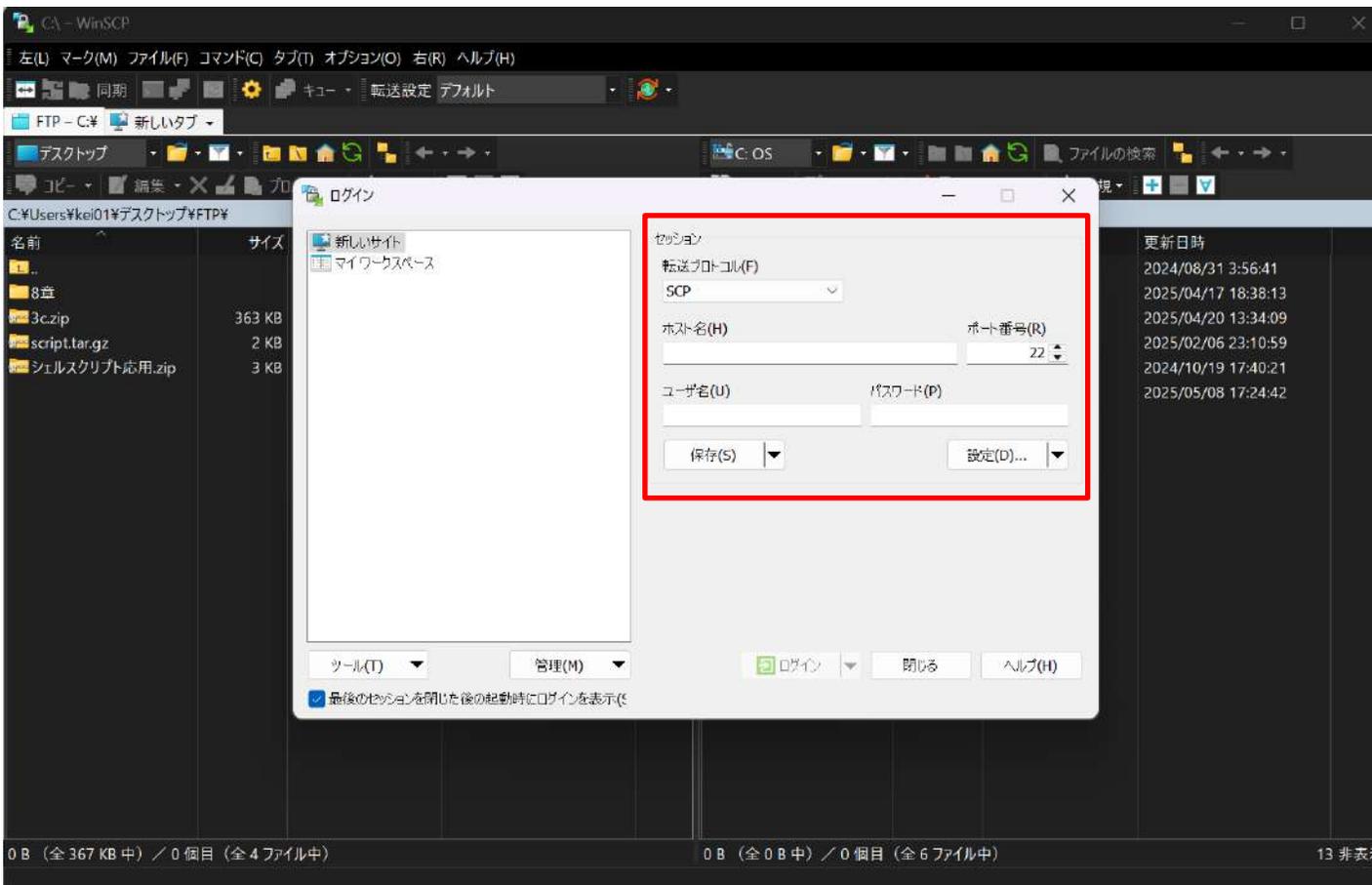
Putty形式に変換できて、
左図のように指定されていれば
準備OKです！
ファイルは .ppkの拡張子です。

問題なければ下の「OK」を押下
し元の画面に戻ってください

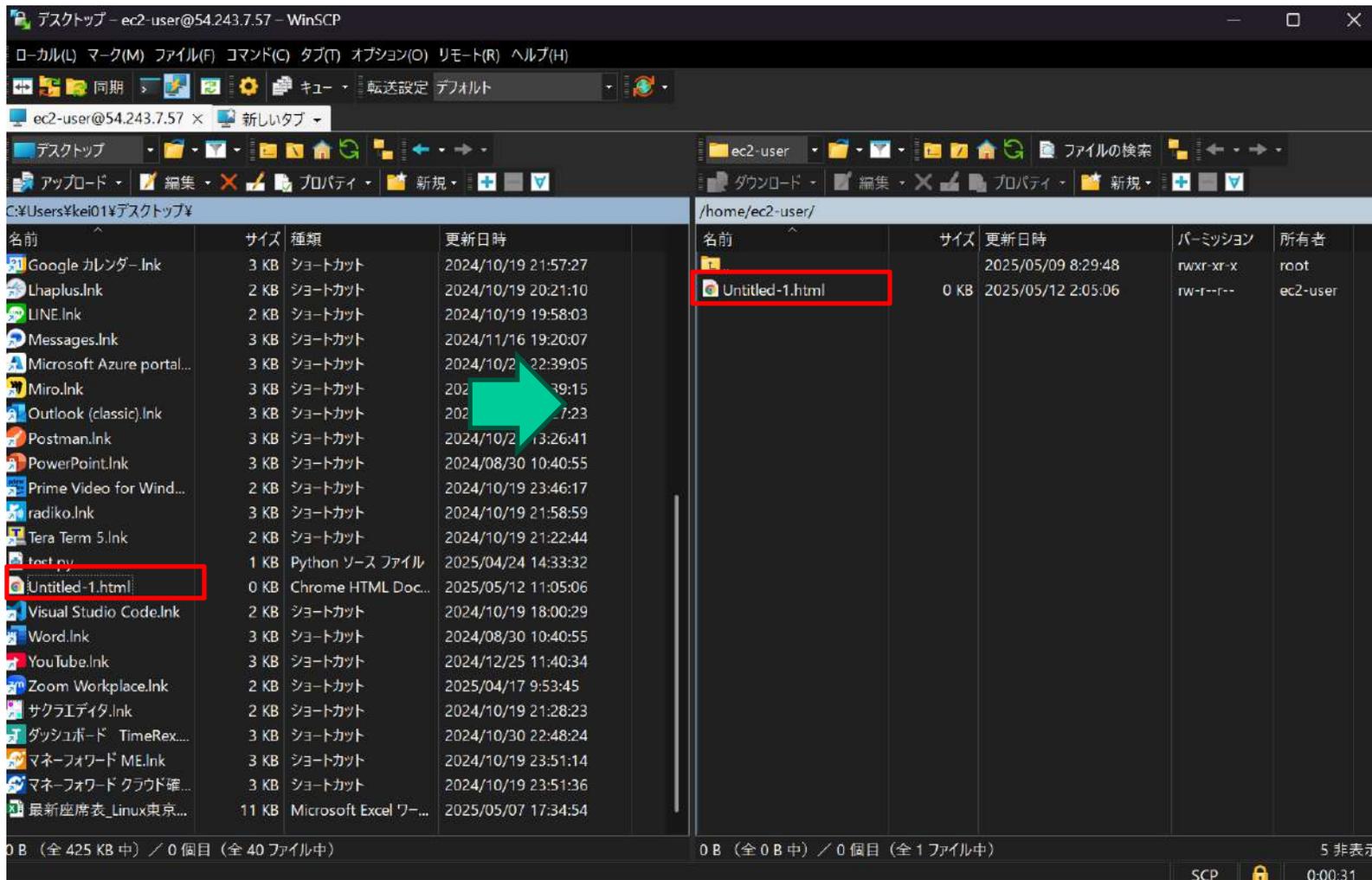
■ WinSCPの使用方法について

次に以下の設定をしログインボタンを押下します。

項目	設定内容
転送プロトコル	SCP
ホスト名	IPアドレス
ユーザ名	ec2-user
パスワード	※鍵認証のため空



■ ファイルの転送方法



転送したいファイルを
「ドラッグアンドドロップ」
することでファイルを転送
することができます！

これらができるとローカル環
境で作成したプログラムを
Linux上へすぐ移動するこ
とができます！

■ Linux上へHTTPパッケージをインストールする

以下のコマンドを実行し、「httpd」パッケージをインストールします。

```
sudo dnf -y install httpd
```

インストールが成功すると「**Complete!**」と出力されます。

Redhat系列のディストリビューションでは「**Apache**」のパッケージ名は「**httpd**」とされています。

Debian系列のディストリビューションをいじる場合は「**Apache2**」というパッケージ名になるため、ご自身の環境で実施される場合は、ディストリビューションの違いに注意してください。



■ Apacheについて

世界中で広く利用されているオープンソースのWebサーバーソフトウェアです。
Webサイトを公開するためのサーバーとして、多くの企業や個人が採用しています。

1. オープンソース

- ✓ 無料で利用可能
- ✓ 世界中の開発者による継続的な改良

2. 高い互換性

- ✓ Linux、Windows、macOS など様々なOSで動作
- ✓ PHP, Python, Perl との連携が簡単

3. 拡張性と柔軟な設定

- ✓ .htaccess を使ったアクセス制御
- ✓ モジュール追加で機能を拡張

4. 安定性と信頼性

- ✓ 長年の運用実績と大規模環境での採用
- ✓ 企業のWebサービスにも広く導入



■ ディレクトリの所有者を変更する

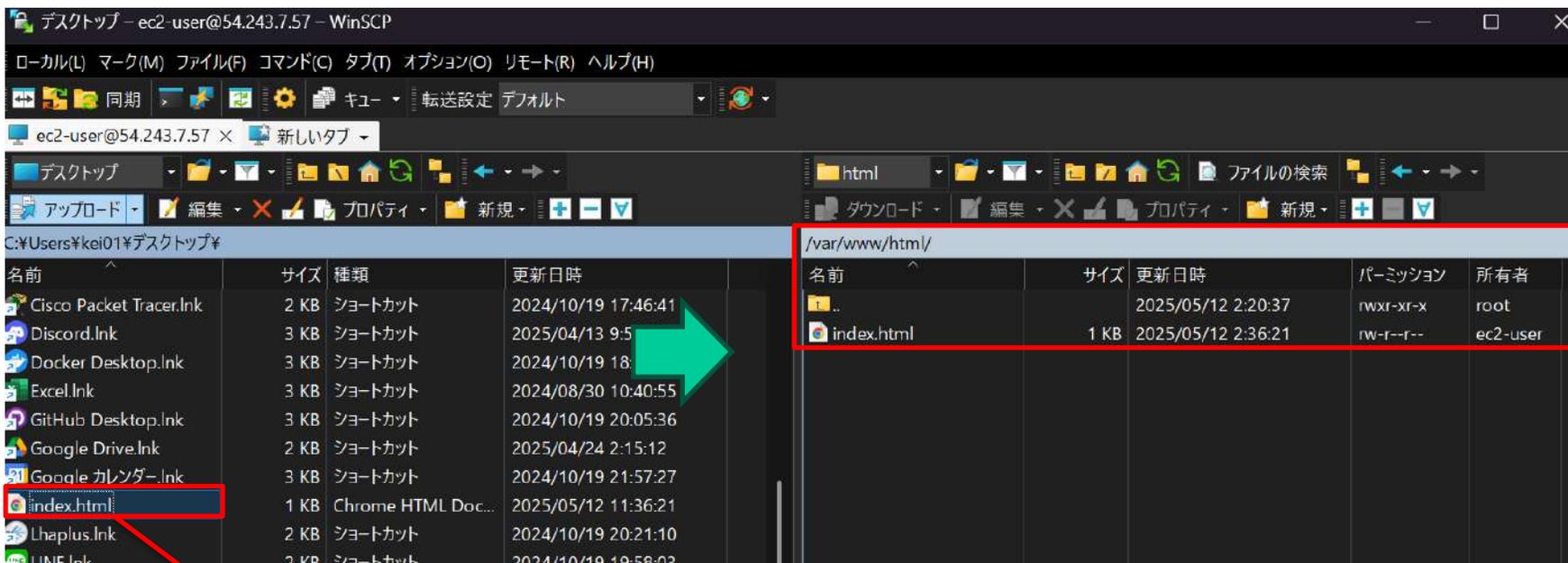
以下のコマンドを実行し、ウェブページ公開用のディレクトリの所有者を「ec2-user」に変更します。

```
sudo chown ec2-user /var/www/html
```

※これによって、「**/var/www/html**」ディレクトリの所有者をrootからec2-userに変更したため、scpでファイルを転送する際に「**Permission Denied**」されるのを防ぐことができます。

Chmodで権限を変えるのでも可能ですが、その他のユーザに権限を与えすぎるのはセキュリティ的に弱まるため、今回はオーナーの変更にしています。

- HTMLファイルを用意し、WinscpでAWS上にuploadする
アップロード対象の「/var/www/html」ディレクトリへhtmlファイルをUploadする。



```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <meta name="viewport" content="width=device-width, initial-scale=1.0">
6    <title>Document</title>
7  </head>
8  <body>
9    <h1>20250526 Linucセミナー</h1>
10   <h2>これはLinucセミナーように作成したWEBサイトです。</h2>
11 </body>
12 </html>

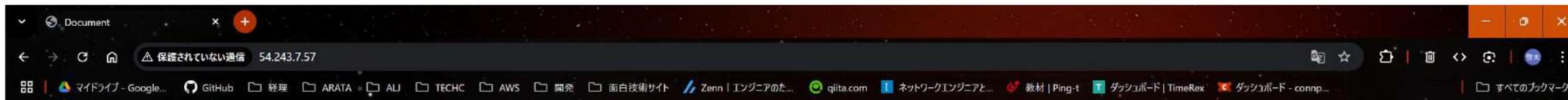
```

■ サービスを開始し、WEBページで確認する。

①以下のコマンドを実行し、httpdサービスを開始する。

```
sudo systemctl start httpd
```

②ブラウザの検索欄に「<http://IPアドレス>」と入力する。



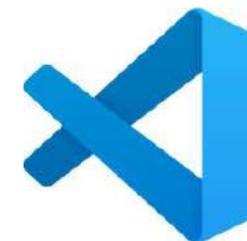
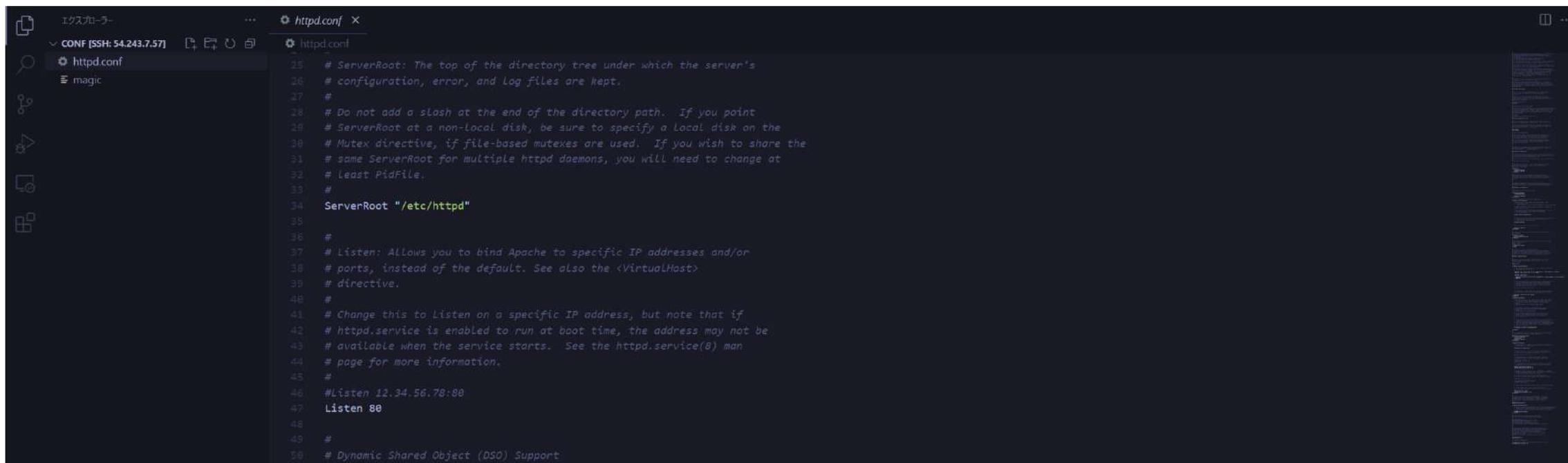
20250526 Linucセミナー

これはLinucセミナーように作成したWEBサイトです。

VScodeとクラウドのインスタンスの接続

■ リモート環境のファイルを編集するには「vi」しかないのですか？

Vscodeの「Remote SSH」を使用することで簡単にクラウド上の設定ファイルなどを編集することができます。

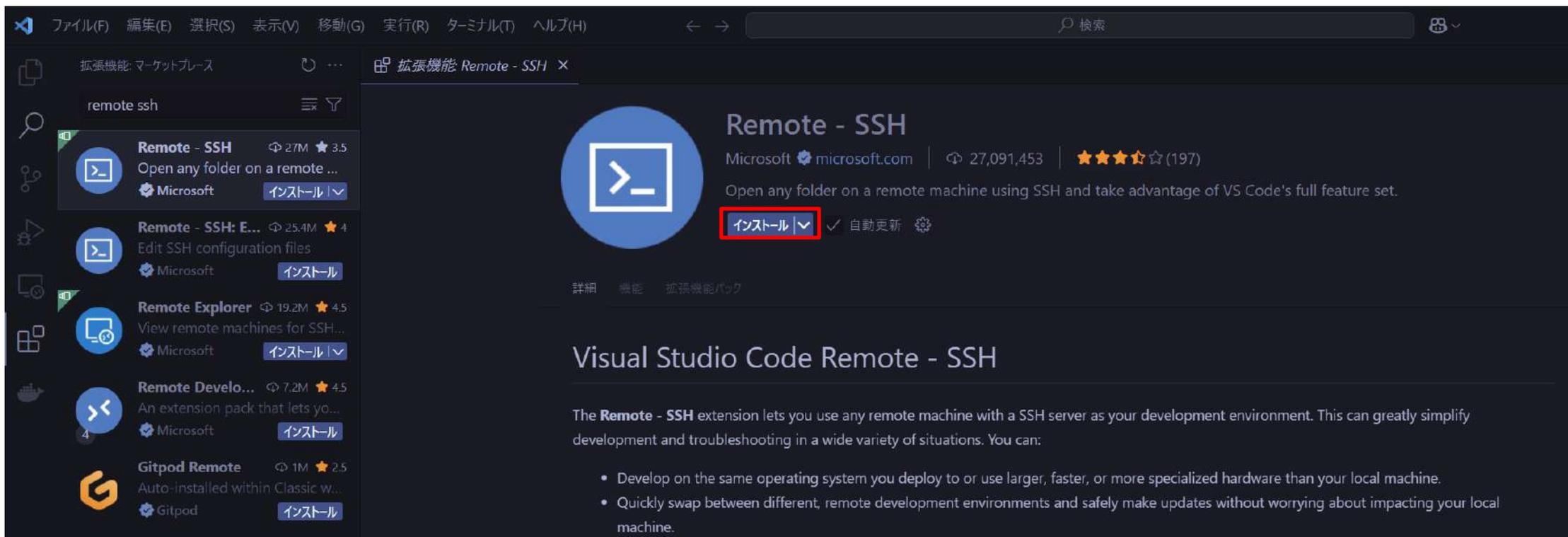
```

25 # ServerRoot: The top of the directory tree under which the server's
26 # configuration, error, and Log files are kept.
27 #
28 # Do not add a slash at the end of the directory path. If you point
29 # ServerRoot at a non-local disk, be sure to specify a local disk on the
30 # Mutex directive, if file-based mutexes are used. If you wish to share the
31 # same ServerRoot for multiple httpd daemons, you will need to change at
32 # least PidFile.
33 #
34 ServerRoot "/etc/httpd"
35
36 #
37 # Listen: Allows you to bind Apache to specific IP addresses and/or
38 # ports, instead of the default. See also the <VirtualHost>
39 # directive.
40 #
41 # Change this to Listen on a specific IP address, but note that if
42 # httpd.service is enabled to run at boot time, the address may not be
43 # available when the service starts. See the httpd.service(8) man
44 # page for more information.
45 #
46 #Listen 12.34.56.78:80
47 Listen 80
48
49 #
50 # Dynamic Shared Object (DSO) Support

```

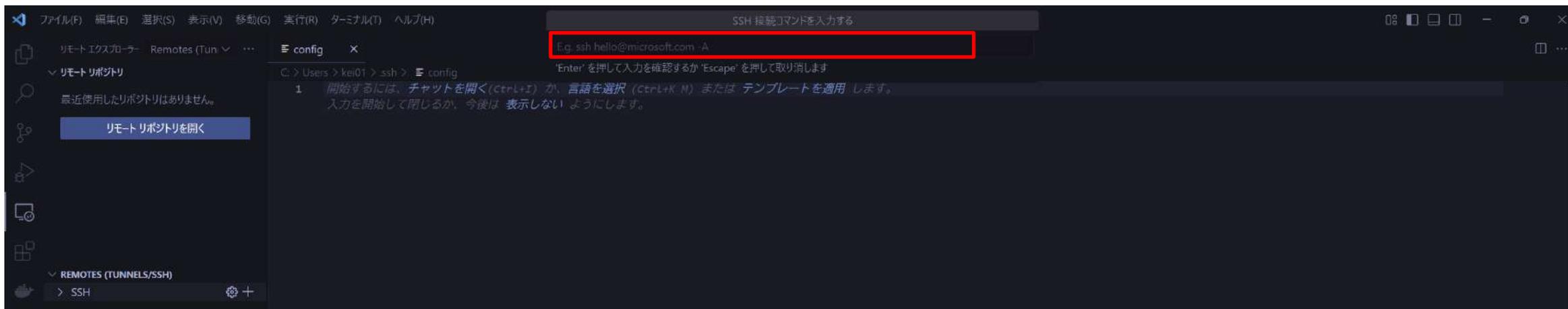
■ 具体的にやり方を教えてください！

①VS Code でRemote SSHと検索し、「インストール」をクリックします。

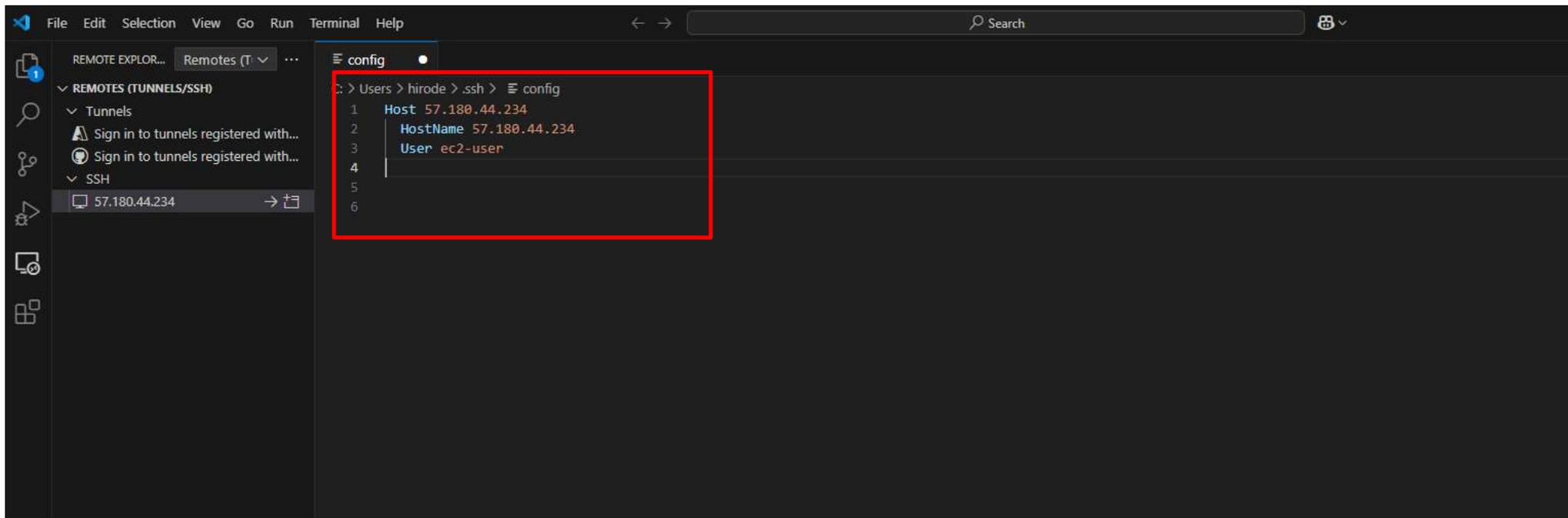


②インストールが終了したら「SSH」の右にある「+」ボタンを押下し、赤枠箇所に**宛先IPアドレス**を入力します。

ec2-user@IPアドレス



③入力後、Enterを押すことでPCの「.ssh/config」にIPアドレスとユーザ名情報が追加されるはずですが。



The screenshot shows the VS Code interface with the Remote Explorer on the left and a terminal window open to the .ssh/config file. The terminal content is as follows:

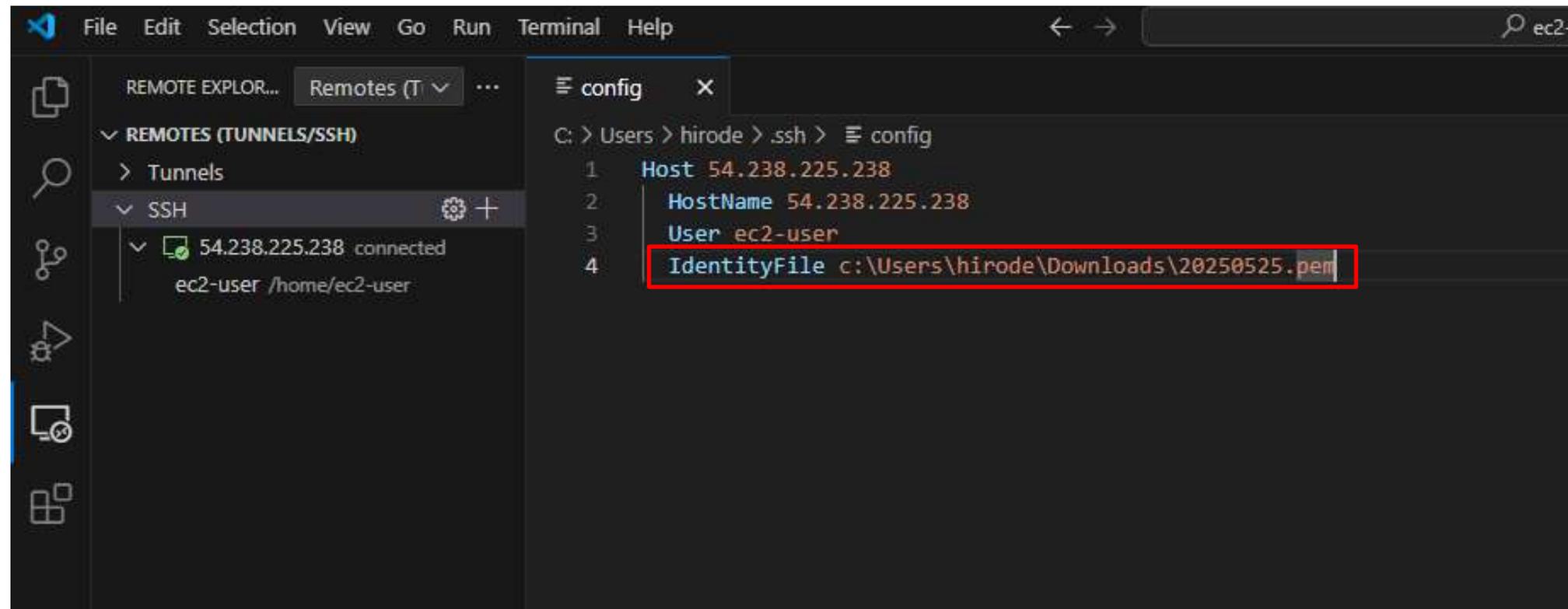
```

C:\Users> Users > hirode > .ssh > config
1 Host 57.180.44.234
2   HostName 57.180.44.234
3   User ec2-user
4
5
6
  
```

④今開いている「**.ssh¥config**」というファイルは、SSHクライアントがログイン先の情報を保存しておくために使用されるファイルです。

ここに以下のように先ほどの**.pemファイル**を指定してあげます。

IdentifyFile pemファイルの絶対パス



The screenshot shows the Visual Studio Code interface with a remote SSH connection. The left sidebar shows the 'Remote Explorer' with a tree view containing 'Tunnels' and 'SSH'. Under 'SSH', there is a connection to '54.238.225.238' with the user 'ec2-user'. The main editor window shows the contents of the '.ssh/config' file. The file content is as follows:

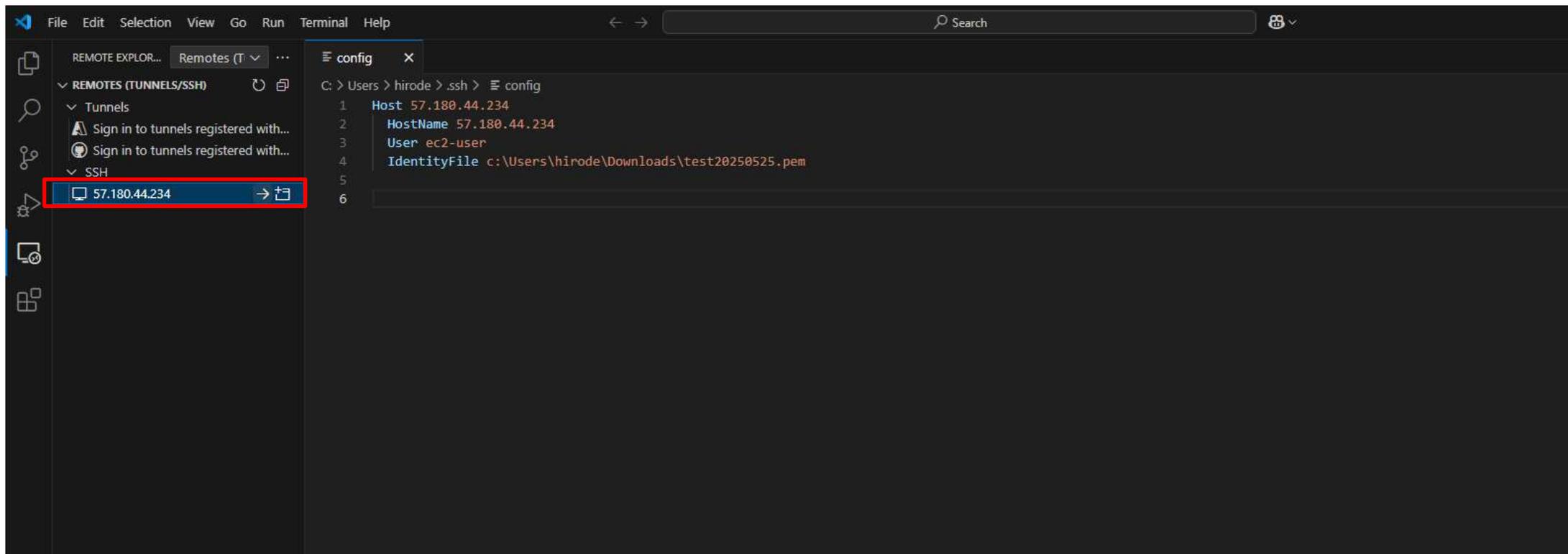
```

1 Host 54.238.225.238
2   HostName 54.238.225.238
3   User ec2-user
4   IdentityFile c:\Users\hirode\Downloads\20250525.pem

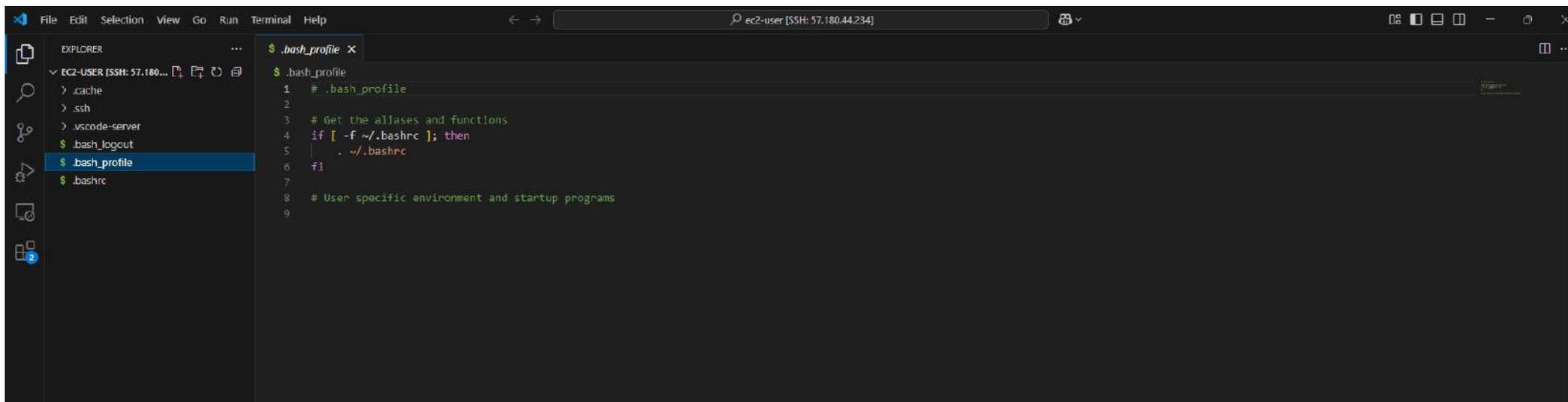
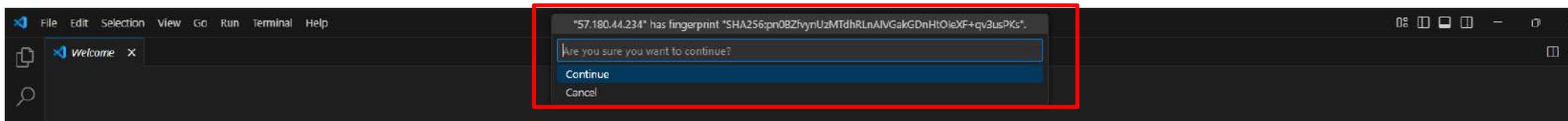
```

The line `IdentityFile c:\Users\hirode\Downloads\20250525.pem` is highlighted with a red box, indicating the path to the PEM file.

⑤問題なく設定できたら「→」のアイコンをクリックします。

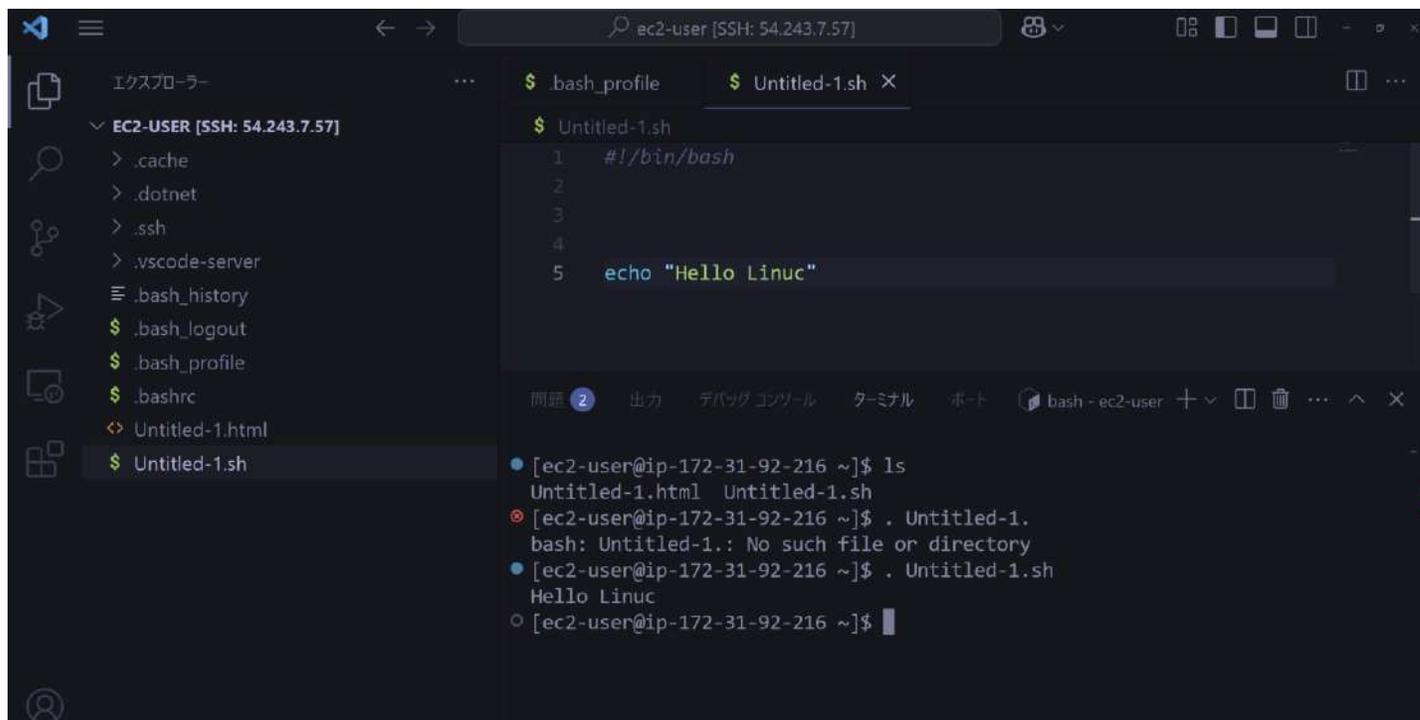


⑥初回ログイン時は、接続先が正しいか確認されますが
Continueを選択すれば**ログイン成功**です！



■ 結局これをするとながができるのですか？

VSCodeで見えると何が良いのか？ というと、クラウド上のLinuxマシンをローカル環境のように動かすことができるため、**LinuC**の勉強でシェルスクリプトの練習時などにスクリプトの記述が楽になるため、勉強がはかどります！！



```

$ bash_profile
$ Untitled-1.sh X
$ Untitled-1.sh
1  #!/bin/bash
2
3
4
5  echo "Hello Linuc"

[ec2-user@ip-172-31-92-216 ~]$ ls
Untitled-1.html  Untitled-1.sh
[ec2-user@ip-172-31-92-216 ~]$ . Untitled-1.
bash: Untitled-1.: No such file or directory
[ec2-user@ip-172-31-92-216 ~]$ . Untitled-1.sh
Hello Linuc
[ec2-user@ip-172-31-92-216 ~]$

```

Appendix

■ 従量課金は怖いので月々定額でインスタンス立ち上げられませんか？

実はAWSには、簡単にインスタンスを作成できて
月々定額でクラウドのインスタンスを立てることができるサービスがあります。

「**Amazon LightSail**」を使用すると、月々最安 5ドルで使い放題になります。

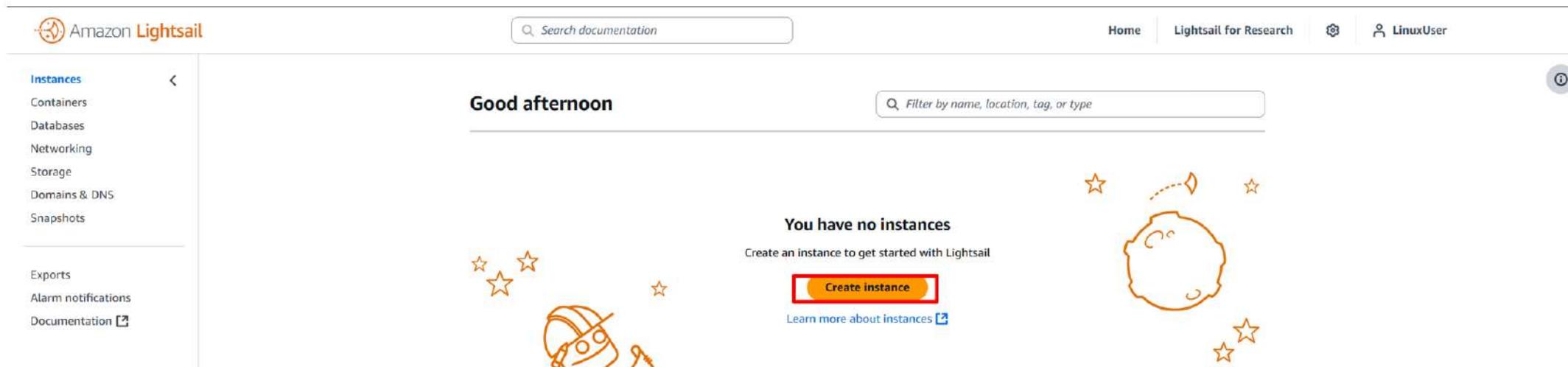
ただ注意点として、作成後の拡張などはできないため
初期段階からコストは**定額**になる一方、後々スペックが足りなくなった場合などには対応できません。

ただLinuxの学習だけであればスペックの拡張などは不必要になると思いますので
もし電源消し忘れが怖い等あれば「**LightSail**」の使用をお勧めします。



■ 作成方法について教えてください！

① 「インスタンスの作成」をクリックします。



② 「Alma Linux」を指定して「インスタンスの作成」を押下します。

インスタンスの場所 情報

東京、ゾーン A (ap-northeast-1a) でこのインスタンスを作成しようとしています
[AWS リージョンとアベイラビリティゾーンを変更](#)

インスタンスイメージを選択 情報

選択するインスタンスイメージによって、オペレーティングシステムと、インスタンスにアプリケーションが含まれるかどうかが決まります。

プラットフォームの選択

- Linux/Unix
27 個の設計図
- Microsoft Windows
6 個の設計図

設計図の選択

アプリ + OS | **オペレーティングシステム (OS) のみ**

- | | | | |
|--------------------------------------------------------------|--------------------------------------------------------|----------------------------------------------|-------------------------------------------|
| <input type="radio"/> Amazon Linux 2023
2023.7.20250428.1 | <input type="radio"/> Amazon Linux 2
2.0.20250428.0 | <input type="radio"/> Ubuntu
24.04 LTS | <input type="radio"/> Ubuntu
22.04 LTS |
| <input type="radio"/> Debian
12.8 | <input type="radio"/> Debian
11.11 | <input type="radio"/> FreeBSD
14.2 | <input type="radio"/> FreeBSD
13.4 |
| <input type="radio"/> openSUSE
15.6 | <input checked="" type="radio"/> AlmaLinux
9.4 | <input type="radio"/> CentOS
CS9-20230110 | |

AlmaLinux 9.4

コミュニティが所有および管理するオープンソースの無料エンタープライズ Linux ディストリビューションでは、長期的な安定性を重視し、堅牢なプロダクショングレードのプラットフォームを提供しています。AlmaLinux OS は RHEL® およびプレストリーム CentOS とは 1:1 のバイナリー互換性があります。

[AWS Marketplace](#) の AlmaLinux について詳細を見る

本ソフトウェアを使用することにより、[本ソフトウェアのエンドユーザーライセンス契約](#) に同意したものとみなされます。

サイズを選択

次でソート: 月次料金 ▼

- | | | | |
|---------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| <input type="radio"/> \$5
1 か月あたりの USD

512 MB メモリ
2仮想 CPU処理
20 GB SSD ストレージ
1 TB 転送
初回 90 日間無料 | <input type="radio"/> \$7
1 か月あたりの USD

1 GBメモリ
2仮想 CPU処理
40 GB SSD ストレージ
2 TB 転送
初回 90 日間無料 | <input checked="" type="radio"/> \$12
1 か月あたりの USD

2 GBメモリ
2仮想 CPU処理
60 GB SSD ストレージ
3 TB 転送
初回 90 日間無料 | <input type="radio"/> \$24
1 か月あたりの USD

4 GBメモリ
2仮想 CPU処理
80 GB SSD ストレージ
4 TB 転送 |
| <input type="radio"/> \$44
1 か月あたりの USD

8 GBメモリ
2仮想 CPU処理
160 GB SSD ストレージ
5 TB 転送 | <input type="radio"/> \$84
1 か月あたりの USD

16 GBメモリ
4仮想 CPU処理
320 GB SSD ストレージ
6 TB 転送 | <input type="radio"/> \$164
1 か月あたりの USD

32 GBメモリ
8仮想 CPU処理
640 GB SSD ストレージ
7 TB 転送 | <input type="radio"/> \$384
1 か月あたりの USD

64 GBメモリ
16仮想 CPU処理
1,280 GB SSD ストレージ
8 TB 転送
最大プラン |

Lightsail の新規のお客様は、期間限定で、選択したプランを 90 日間無料でお試しください。Lightsail の無料トライアルの詳細はこちら。

AWS サービスの使用には [AWS カスタマーアグリーメント](#) が適用されます。

インスタンスの作成

③正しく作成できている場合は、**インスタンス**が存在し、「実行中」となっているはずです。

The screenshot shows the Amazon Lightsail console interface. On the left is a navigation menu with options like 'Instances', 'Containers', 'Databases', etc. The main area is titled 'Good afternoon' and shows a list of instances. One instance is highlighted with a red box:

Instance Name	Specifications	Status	Public IP	Private IP	Location
AlmaLinux-1	2 GB RAM, 2 vCPUs, 60 GB SSD	Running	52.196.171.198	2406:da14:1ca1:bd00:b3dd:df71:2d03:1dff	Tokyo, Zone A

③ オレンジの箇所「SSHを使用して接続」をクリックするとブラウザですぐにLinuxのインスタンスに接続することができます。

AlmaLinux-1 情報

2 GB RAM、2 vCPU、60 GB SSD

削除 再起動 停止

<p>AWS リージョン 東京、ゾーン A (ap-northeast-1a)</p> <p>ネットワークタイプ デュアルスタック ネットワークタイプを変更</p>	<p>パブリック IPv4 アドレス 52.196.171.198</p> <p>プライベート IPv4 アドレス 172.26.8.247</p> <p>パブリック IPv6 アドレス 2406:da14:1ca1:bd00:b3dd:df71:2d03:1dff</p>	<p>インスタンスステータス 実行中</p>
----------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------

[接続](#) | [メトリクス](#) | [スナップショット](#) | [ストレージ](#) | [ネットワーク](#) | [ドメイン](#) | [タグ](#) | [履歴](#)

インスタンスに接続 情報

ブラウザまたは独自の互換性のある SSH クライアントを使用して接続できます。

ブラウザを使用 情報

ブラウザベースの SSH クライアントを使用して接続します。

SSH を使用して接続

独自の SSH クライアントを使用 情報

インスタンスに接続するには、次の認証情報を使用します。

パブリック IPv4 アドレス
52.196.171.198

パブリック IPv6 アドレス
2406:da14:1ca1:bd00:b3dd:df71:2d03:1dff

ユーザー名
ec2-user

SSH キー
このインスタンスは東京 (ap-northeast-1) リージョンのデフォルトの SSH キーを使用します。

[デフォルトキーのダウンロード](#)

もちろん「デフォルトキーのダウンロード」をクリックし、鍵認証にてTeraTermなどでログインすることも可能です！

パブリッククラウドを活用したLinuC学習環境構築 最後に

■ いかがでしたでしょうか？

クラウドというものに苦手意識を持っている方もいたかと思いますが、一回作ってしまえば、2回目からは簡単に作成することができ、様々な**ディストリビューション**の学習環境を簡単に作成することができます。

もっと言うと外出先や携帯からログインしたいなど様々な欲求にこたえてくれるのが、**クラウドサービス**です。

AWSなどのクラウドは仮想マシンを立ち上げるだけであれば非常に簡単に使いこなすことができますので、どんどん使って、**LinuC**の学習を進めていきましょう。



ご静聴ありがとうございました！