

# パブリッククラウドを活用した LinuC学習環境構築

AWS EC2で簡単Linuxサーバー構築



LinuC取得者応援サイト「リナスク」主宰 水澤泰敬

## ■水澤泰敬

日経新聞関連企業、商社などへの転職を重ねたのち、フリーランスを経て会社設立。10数年にわたり数々の情報まとめサイト運営に携わる。キュレーションサイトへの転職、事業譲渡によるTSUTAYA関連企業への転籍を経て再びフリーランスとなる。2020年、日本のエンジニア育成に携わりたいと思い、エンジニアスクールで企業研修講師を務めながら約半年の準備期間を経てLinux技術者認定資格取得を目指す人を応援する情報サイト「リナスク」を立ち上げ。

**取得資格：** LinuCレベル1、LinuCレベル2、LinuCレベル3（304）、Linux Essentials、CCNA、Java Bronze、情報セキュリティマネジメント、情報セキュリティ管理士、個人情報保護士等

## LPI-Japan関連セミナー講師歴

- 2021/1/30 OSC 2021 Online/Osaka  
「LinuCの学習にも使える、ブラウザだけで使えるクラウド環境紹介ハンズオン」
- 2021/3/26 LPI-Japan技術解説セミナー  
「Linux研修【ハンズオン】コンテナを体験してみよう！」
- 2021/7/15 LPI-Japan技術解説セミナー  
「Linux研修【ハンズオン】ネットワークの基礎を学ぼう！ / ネットワークインターフェイスとDNSの設定」
- 2021/7/31 OSC 2021 Online/Kyoto  
「LinuCの学習にも使える、ブラウザで動かすクラウドLinuxハンズオン」

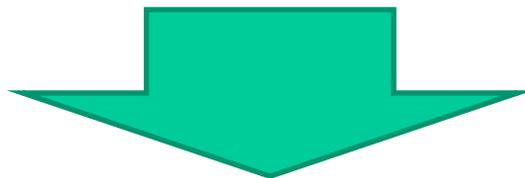


Linux技術者認定資格取得を目指す人を応援する情報サイト

<https://linuc.spa-miz.com/>



- 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。
- Webサーバー(Apache)とコンテナアプリ(Docker)をインストールし動作確認。



- ✓クラウドサービスを利用したLinux学習環境が構築できる
- ✓セキュアなりモート接続でクラウドサービスを利用する
- ✓クラウドセキュリティの基礎を学ぶ足掛かりにする

ハンズオン形式ではありませんが、AWSにアカウントをお持ちの方は、実際に手を動かしながら学習環境が構築できる内容にしています。

## ■主題1.01：Linuxのインストールと仮想マシン・コンテナの利用

- 1.01.1 Linuxのインストール、起動、接続、切断と停止
- 1.01.2 仮想マシン・コンテナの概念と利用

## ■主題1.10：セキュリティ

- 1.10.3 暗号化によるデータの保護
- 1.10.4 クラウドセキュリティの基礎

本セミナーは学習環境の構築方法の解説が主となるため、出題範囲の解説は参考程度になります。

- LinuCとは
- LinuC学習環境の構築方法（全般）
- クラウドサービスとは
- AWSの利用方法
- AWS上でLinux環境を構築する手順
- 他のクラウドサービスを使う場合
- クラウド学習環境ではカバーが難しいLinuCレベル 1 出題範囲
- まとめ

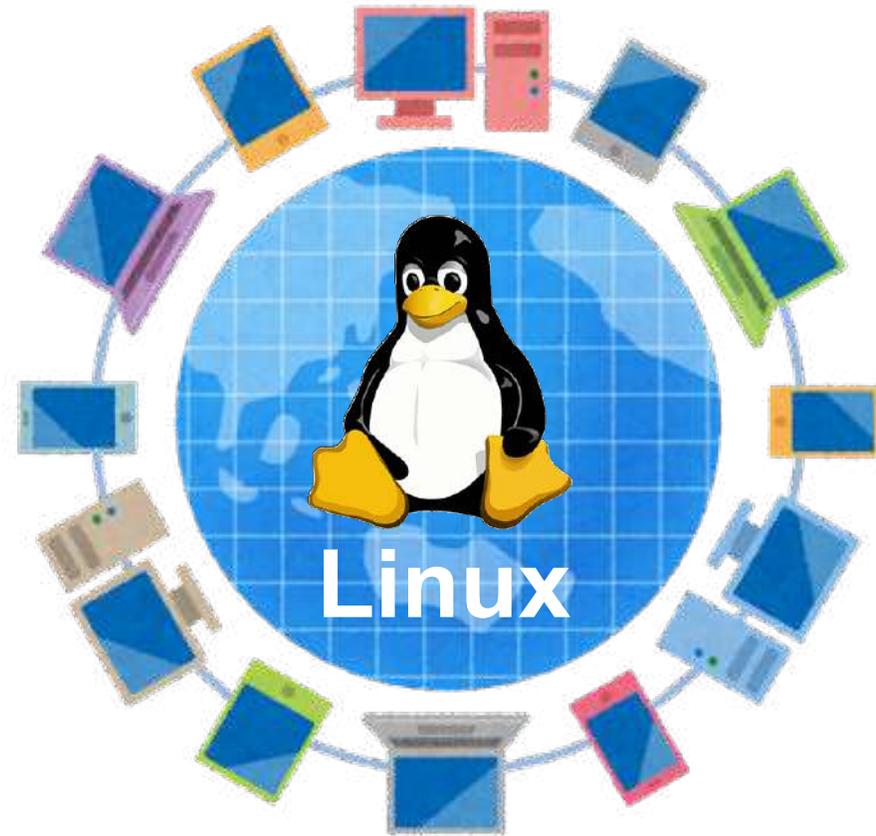
- 現在、クラウドサービスを使ったシステム開発が普通に行われるようになってきました。



- 驚くほどの短時間でサーバー環境が出来上がる今、なぜLinuxを学ぶ必要があるのでしょうか？



- Linuxはクラウド時代の様々なITサービスを支える技術です。



- 主要なクラウドサービスもほとんどがLinuxベースで運用されています。



- Linuxは、コンピューターのアーキテクチャから学べて本質的な技術力(=土台となる知識)を身につけるのに最良のOS。
- Linuxの学びを通じて、ITエンジニアに必須となるサーバー・ネットワーク技術やコンピュータの仕組みなど、基礎から知識を習得できます。

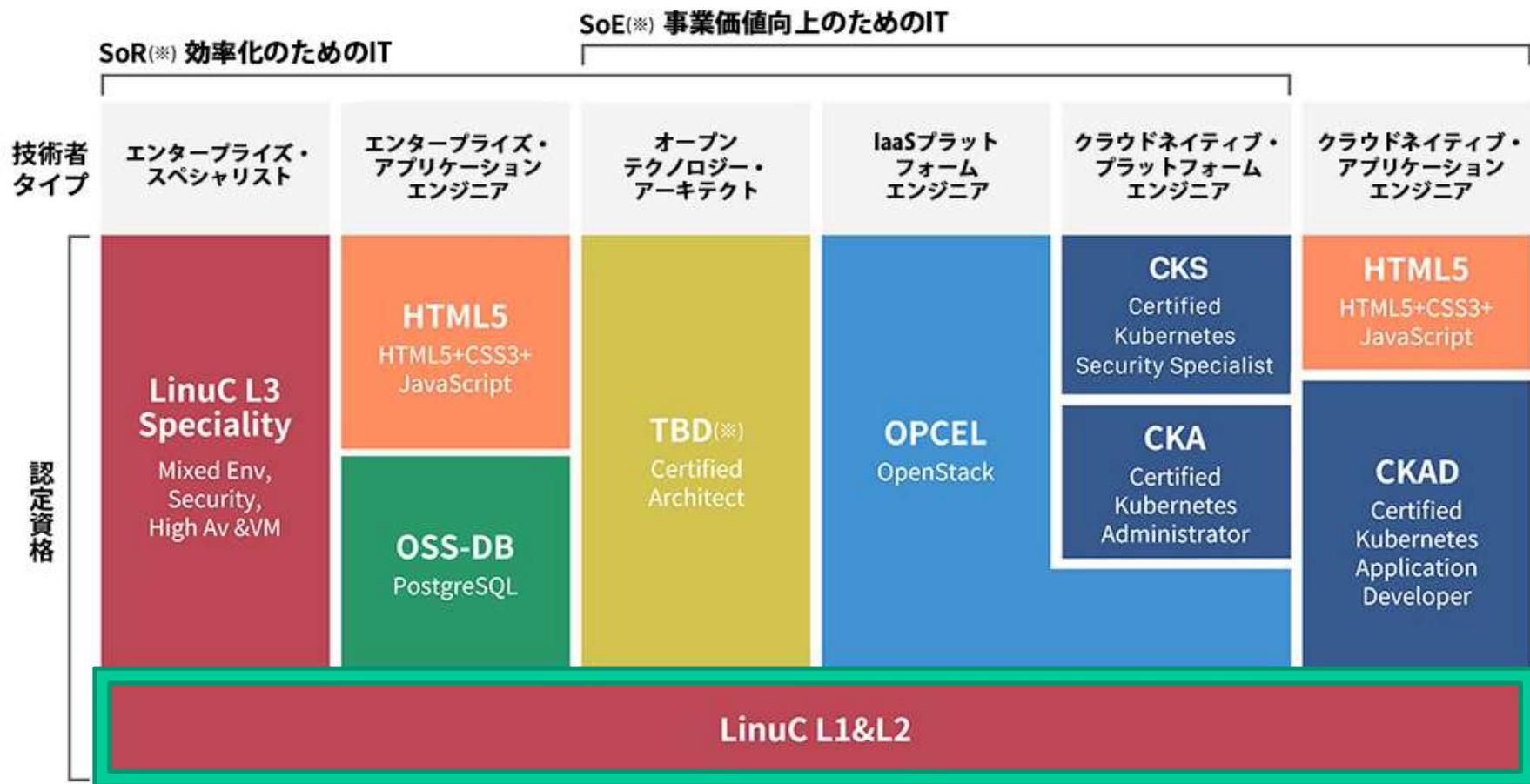


# Linux

- Linuxエンジニアに求められる技術力を証明できる認定



## ■ LinuC取得がITエンジニアの基礎技術所有を証明



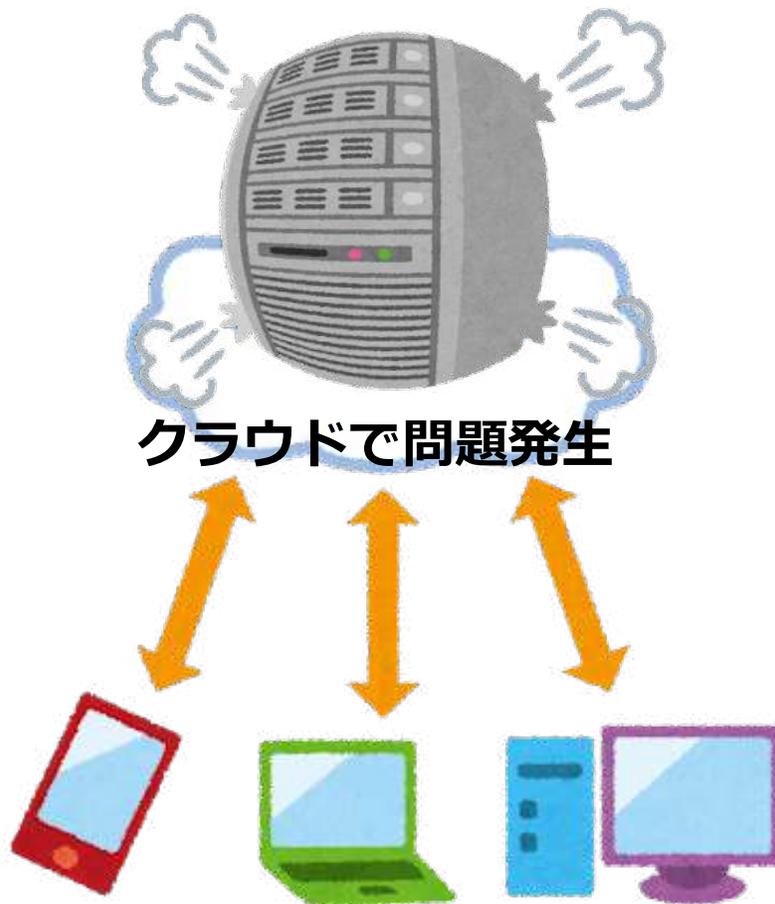
※SoR: System of Records, SoE: System of Engagement, TBD: To Be Determined

2021年7月時点

## ■Linuxの知識がトラブルや様々な問題の解決に役立つ



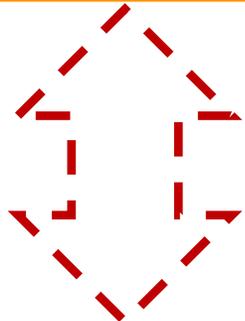
Linuxの知識がない人



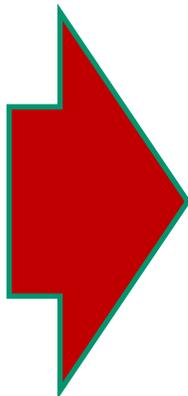
Linuxの知識がある人

- 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】



## ■LinuCの3つのレベル

SKILL UP

オンプレミス環境

仮想環境

  
**LinuC-3**  
 version10.0  
 (開発中)

高度な Linux システムのアーキテクチャ設計・構成・構築ができる

システム要件・非機能要件を満たす適切な設計・構築・運用をリードできる

スキル例) ディレクトリ /DB/API/WEB などのアーキテクチャ設計・構築など

  
**LinuC-2**  
 version10.0

オンプレ / 仮想の環境を問わず様々な Linux サーバを構築できる

必要な設定・構築が行える自立したエンジニアに

スキル例) 管理 /DNS/ ファイル /Web/ メールなどそれぞれのサーバ構築や設定など

  
**LinuC-1**  
 version10.0

オンプレ / 仮想を問わず、Linux サーバの運用・管理ができる

確かな知識で現場の即戦力エンジニアとして活躍できる

スキル例) 構築済みサーバ / 構築済みコンテナ (仮想環境) の運用管理対応など

## ■アカデミック認定校に通う

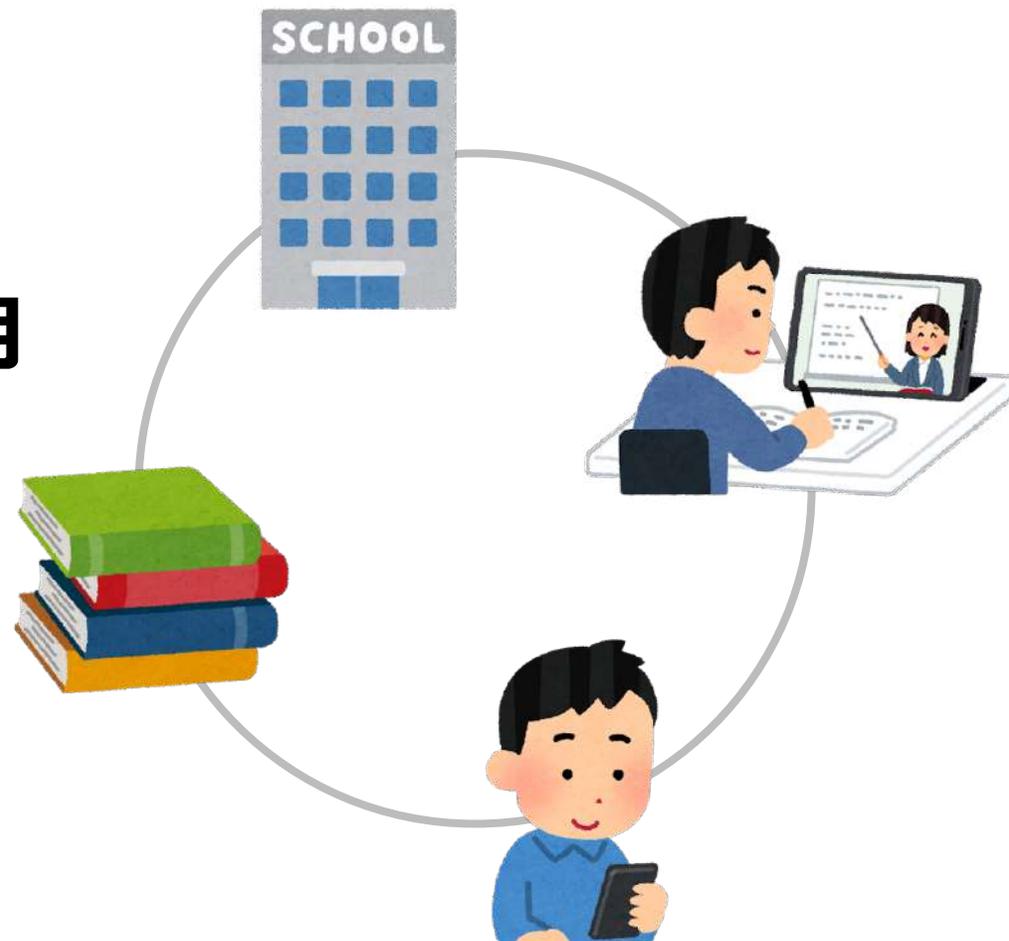
- 大学、専門学校
- PCスクール
- 企業研修機関

## ■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



# LinuCの学習は実際に手を動かしながらの反復学習が効果的

## 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

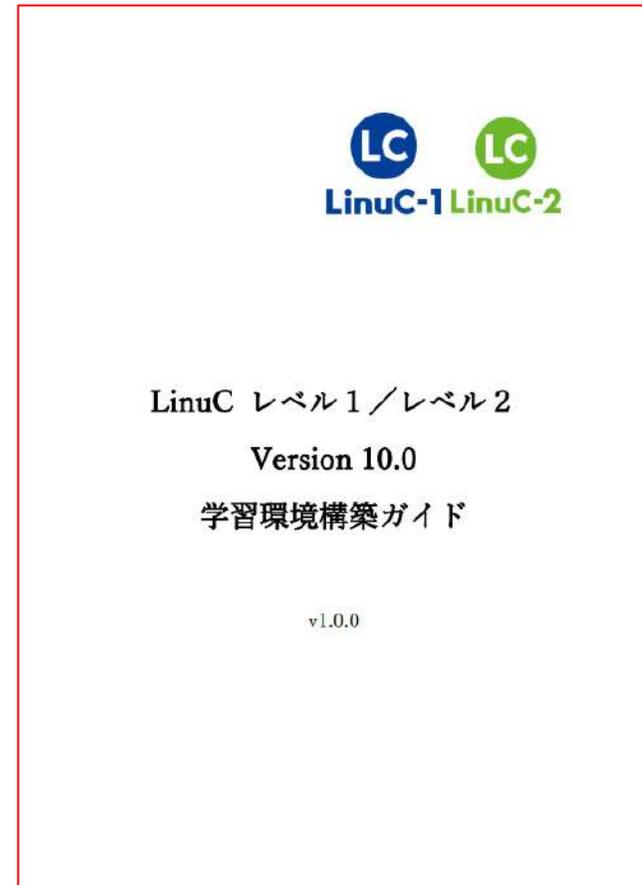
学習環境の構築 (Linux環境で、手を動かしながら学ぶ)

## ■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
- 他にパブリッククラウド上にLinux環境を構築する方法もあることを記載しているが、具体的な構築方法の解説は無い。



本セミナーにて解説



## ■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 a screenshot of the Katacoda website. The top section features the O'Reilly Katacoda logo and navigation links. Below, a blue banner reads "Learn new technologies using real environments right in your browser". The main content area highlights "Katacoda enhances your technical sales, training and internal education process" with three key features: "Learn & Discover the latest technologies and tooling", "Enable Users to Test Drive Your Products Within Seconds", and "Self-Paced Interactive Learning For Your Teams".

At the bottom, a browser window displays the "Ubuntu 20.04 Playground" interface. It includes a terminal window with the following output:

```

Terminal
Visualise Host
Your Interactive Bash Terminal. A safe place to learn and execute c
$ 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**

■クラウドコンピューティングサービスは、インターネット経由でIT関連の様々なリソースを提供するサービス。

- パブリッククラウド
- プライベートクラウド

■システムの構築や運用に必要なインフラリソースだけでなく、インフラを動かすためのOSやサーバー構築に必要なソフトウェアなどをまとめて借りることができるのが特徴。

■IaaS/PaaS/SaaSなどの形態がある。

- 「IaaS」はInfrastructure as a Serviceの略で、CPUやストレージなどのリソースをネットワーク経由で利用するもの
- 「PaaS」はPlatform as a Serviceの略で、アプリケーション開発に必要な実行環境をネットワーク経由で利用できるもの
- 「SaaS」はSoftware as a Serviceの略で、ソフトウェアやアプリケーションをネットワーク経由で離れたところから利用できるもの



## ■背景

- ITインフラの高性能化、低価格化、高速化
- 仮想化環境の成熟

## ■メリット

- ハードウェアの調達・保守が不要
- 数分でサーバー環境が構築できる
- サービスを組み合わせて利用できる
- 必要なものを必要な分だけ借りられる（従量制）
- 専門知識がなくても利用できる
- グローバル展開が容易

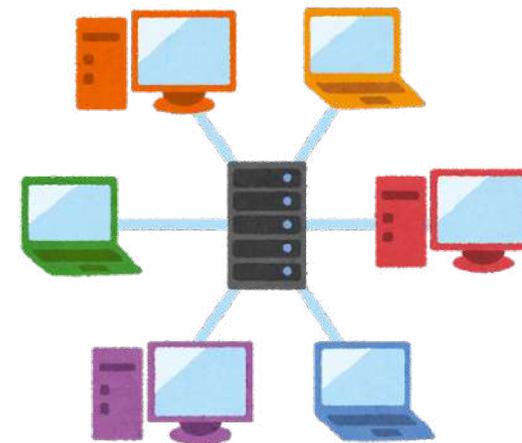


**初期導入コストを抑えつつ、サービスの提供に必要なシステムを素早く構築できる。**

- 専門的な知識がなくても利用はできるが、サーバーやネットワークに関する基礎知識は必須。

## → Linux/LinuCの知識が役に立つ

- 導入に当たっては、
    - ランニングコストがかかる
    - 全てに対して管理権限があるわけではない
    - 事業者側の方針やトラブルの影響を受ける（利用者側からの対処はできない）
- といったことの理解も必要。



クラウドサービスを利用したLinuxの学習環境構築のメリットと一般的なクラウドサービスを利用する目的とメリットは必ずしも一致しない。無料枠が活用できる期間はコスト的なメリットがあるが、終了後は有償になるため、学習環境を維持し続けるのであれば常にランニングコストを意識する必要がある。

3  
大  
ク  
ラ  
ウ  
ド



**Amazon Web Service(AWS)**



**Google Cloud**



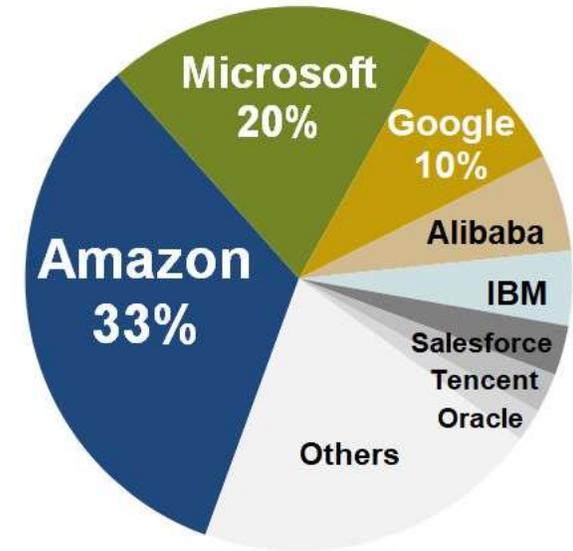
**Microsoft Azure**



**IBM Cloud**



**Alibaba Cloud**



Market Share Q2 2021

Source: Synergy Research Group

<https://www.srgresearch.com/articles/quarterly-cloud-market-leaps-to-42b-amazon-microsoft-google-pocket-63-of-dollars-spent>

参考：パブリッククラウドの国内市場は1兆円を突破（MM総研発表：2021/6時点）

<https://www.m2ri.jp/release/detail.html?id=500>

PaaS利用者全体では、AWSは37.4%、Azureは30.6%、GCPは15.9%。

IaaS利用者全体では、AWSが40.3%、Azureが26.3%、GCPが13.7%。

## AWSアカウント作成の流れ

<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。



### ステップ 1: AWS アカウントの作成



セキュリティの観点からも、登録ユーザーとは別のユーザーを作成し、そこからAWSにサインインします。



登録ユーザー



運用ユーザー

AWSアカウントには1つ以上のIAM(AWS Identity and Access Management)ユーザーを作成できます。

IAMユーザーはマネジメントコンソールで作成できます。



## AWS Identity and Access Management (IAM)

ユーザーを追加



ユーザー詳細の設定

同じアクセスの種類とアクセス権限を使用して複数のユーザーを一度に追加できます。 [詳細はこちら](#)

ユーザー名\*

[別のユーザーの追加](#)

AWS アクセスの種類を選択

これらのユーザーが主に AWS にアクセスする方法を選択します。プログラムによるアクセスのみを選択しても、ユーザーは引き受けたロールを使用してコンソールにアクセスすることはできません。アクセスキーと自動生成されたパスワードは、最後のステップで提供されます。 [詳細はこちら](#)

- AWS 認証情報タイプを選択\*
- アクセスキー - プログラムによるアクセス  
AWS API、CLI、SDK などの開発ツールの アクセスキー ID と シークレットアクセスキー を有効にします。
  - パスワード - AWS マネジメントコンソールへのアクセス  
ユーザーに AWS マネジメントコンソールへのサインインを許可するための パスワード を有効にします。

- コンソールのパスワード\*
- 自動生成パスワード
  - カスタムパスワード

パスワードの表示

- パスワードのリセットが必要  ユーザーは次回のサインインで新しいパスワードを作成する必要があります。ユーザーは、自動的に IAMUserChangePassword ポリシーを取得し、自分のパスワードを変更できるようにします。

今回は「linuxstudy」ユーザーを新規に作成します。

ユーザー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ユーザーだけでなく、IAMユーザーにも設定するようにしましょう。

作成した「linuxstudy」ユーザーでAWSにサインインします。

項目	内容
ユーザー名	linuxstudy
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

AWSでLinux環境を構築するにはAmazon EC2（以下EC2）を利用します。

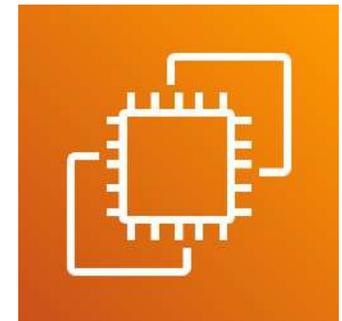
EC2は、Elastic Compute Cloudの略称で、スケーラブルなコンピューティング環境を提供します。

EC2の使用により、ハードウェアに事前投資する必要がなくなり、アプリケーションをより速く開発およびデプロイできます。

必要に応じて仮想サーバーの起動、セキュリティおよびネットワーキングの構成、ストレージの管理ができます。

要件変更や需要増に応じてスケールアップまたはスケールダウンできるため、トラフィック予測の必要性を軽減できます。

インフラのみの提供になるアンマネージドサービス(IaaS)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

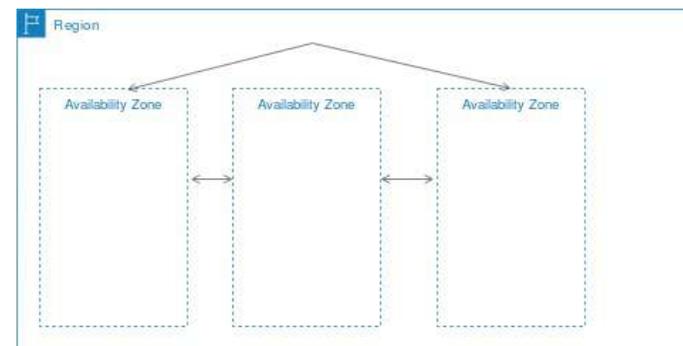
- リージョン→地理的な分類（世界26地域\*に設置されたデータセンター群）
- アベイラビリティゾーン(AZ)→各リージョン内の複数の独立した場所

リージョンごとにアベイラビリティゾーンと呼ばれる複数の独立した場所があります。インスタンスを起動するときに、アベイラビリティゾーンを自分で選択するか、自動的に選択されるようにできます。

インスタンスを複数のアベイラビリティゾーンに配布する場合は、1つのインスタンスで障害が発生したら別のアベイラビリティゾーンのインスタンスが要求を処理するように、アプリケーションを設計できます。

EC2リソースを作成するたびに、リソースのリージョンを指定できます。リソースのリージョンマネジメントコンソールやコマンドラインを使用して指定できます。

日本には東京(ap-northeast-1)と大阪(ap-northeast-3)、2つのリージョンがあります。



## 事前準備

AWSにIAM(linuxstudy)ユーザーでログイン後、マネジメントコンソールからEC2サービスを選択。リージョンは東京を選択しておきます。



AWS Management Console

## 1. EC2インスタンスを起動

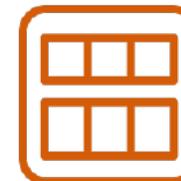
- Amazon マシンイメージ (AMI) を選択
- インスタンスのハードウェア構成を選択
- セキュリティグループの作成
- キーペアを作成する



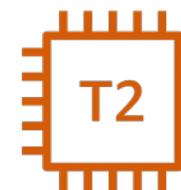
Amazon Elastic Compute Cloud (Amazon EC2)

## 作成するインスタンスの構成

項目	設定内容
AMI	Amazon Linux 2 (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
タグ	キー:Name 値:Linux_Study
セキュリティグループ	SSH(22) HTTP(80)

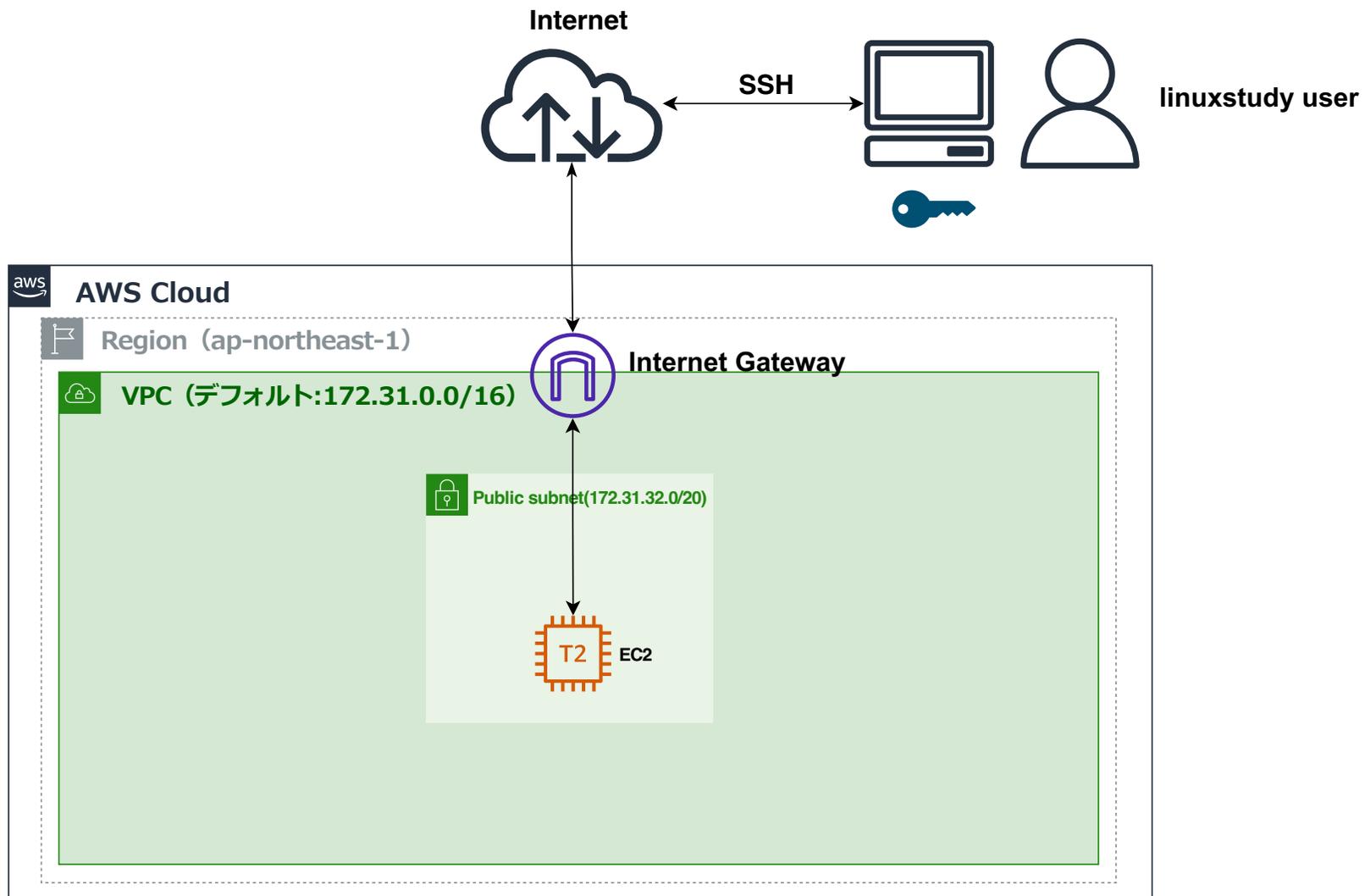


AMI

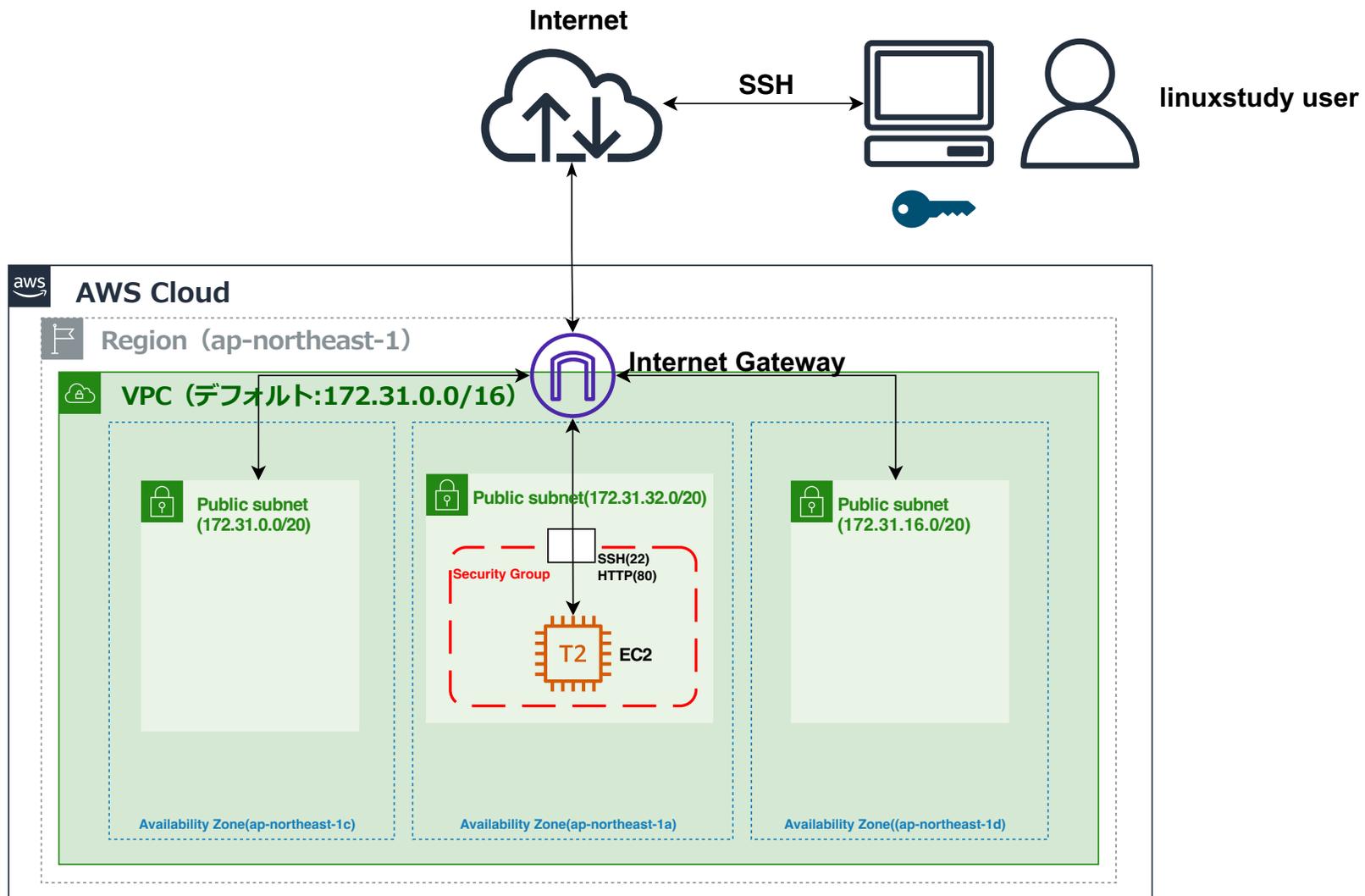


T2 instance

## 構成図(簡易)



## 構成図(詳細)



## 2. EC2インスタンスに接続する

### 接続手段

- SSHクライアント(Macはターミナル、WindowsはTera TermやPowerShellなど)
- EC2 Instance Connect
- AWS Systems Manager Session Manager

### SSHクライアントの場合

1. 接続の前にプライベートキーファイルのパーミッションを変更する。

```
$ chmod 400 Inxstdkey.pem
```

2. パーミッション変更後、**ec2-user**ユーザーでインスタンスのパブリックURLにSSH接続。

```
$ ssh -i <プライベートキーファイル> ec2-user@<パブリックDNS or パブリックIP>
```

注 : **ec2user**ではなく**ec2-user**。Amazon Linux以外のインスタンスの場合はログイン用ユーザーアカウントが異なるので注意。

## 3. EC2インスタンスにソフトウェアのインストールや設定を行う

インストール作業にはroot権限が必要なので、sudoを頭につける。  
面倒な場合は一時的にrootユーザーに切り替える。

### 一時的なrootユーザーへの切り替え

```
$ sudo -i
```

### Webサーバーソフトのインストール、起動、動作状態の確認

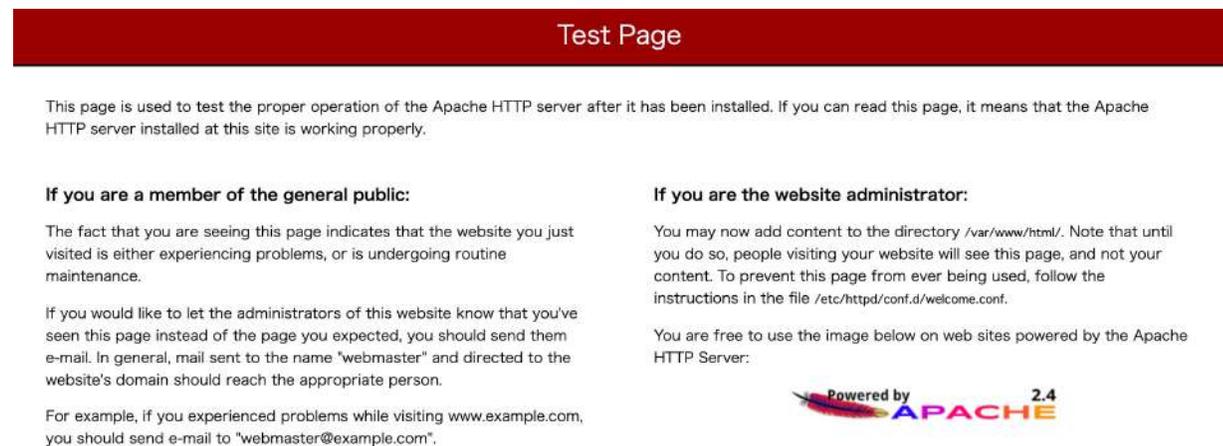
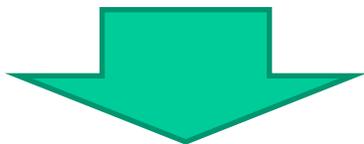
```
# yum install httpd (-y)
# systemctl start httpd
# systemctl status httpd
```

## Webサーバーの動作確認

ローカルマシンからWebブラウザでパブリックIPアドレスを指定してhttp接続(※)。

http://パブリックIPアドレス

ApacheのTestページが表示されればOK。



テストページが表示されるのは、セキュリティグループでhttp(ポート80)を有効にしているから。

※https接続の場合は別途設定が必要になる→AWSにチュートリアルが用意されている。

[https://docs.aws.amazon.com/ja\\_jp/AWSEC2/latest/UserGuide/ec2-tutorials.html](https://docs.aws.amazon.com/ja_jp/AWSEC2/latest/UserGuide/ec2-tutorials.html)

## Dockerのインストールと起動

RedHat系の場合、通常は

```
# yum install -y yum-utils device-mapper-persistent-data lvm2
# yum-config-manager --add-repo
https://download.docker.com/linux/centos/docker-ce.repo
# yum install -y docker-ce docker-ce-cli containerd.io
```

等の一連のコマンドで導入するが、Amazon Linux 2にはDockerのリポジトリが含まれているので、

```
# yum install docker
```

だけでOK。

念のため、

```
# yum repolist
# yum info docker
```

でリポジトリとパッケージの情報を確認。

## Dockerの動作確認

- ✓バージョン確認
 

```
# docker version
```
- ✓起動
 

```
# systemctl start docker
```
- ✓状態確認
 

```
# systemctl status docker
```
- ✓サンプルコンテナの実行
 

```
# docker container run hello-world
```

```
# docker container run docker/whalesay cowsay hello-LinuxC
```
- ✓コンテナの起動状態を確認（停止中含め全て）
 

```
# docker container ls -a
```
- ✓コンテナイメージの確認
 

```
# docker image ls
```

## DockerでWebサーバを動かす

前にインストールしたApacheと競合しないポートを指定して実行。

```
# docker container run -d --name apa01 -p 8080:80 httpd
# docker container ls
```

コンテナの動作確認後、Webブラウザで<http://パブリックURL:8080>にて動作確認。

ただし、このままアクセスしても繋がらない。

セキュリティグループでインバウンドルールにカスタムTCPポート8080をルールに追加して、再度ブラウザで確認。(It works!というテキストのみが表示されていればOK)

## 4. セキュリティ関連のログを確認

不正アクセスの痕跡があるかどうか。

```
# less /var/log/secure
```

```
# grep "Invalid user" /var/log/secure
```

## 5. EC2インスタンスをクリーンアップする

EC2ダッシュボードからインスタンスを指定して「終了」する。

終了することでインスタンスが削除できコストが発生しなくなる。

※停止の場合はEC2の課金は止まるが同時作成したボリューム(EBS)は課金対象

不要なインスタンスは終了する (= 破棄) → コストを抑える → クラウドサービスならではの  
利用方法

## EC2無料利用枠：750時間/月×12か月間

750 / 24 = 31.25 (=約1ヶ月) なので、1年間連続稼働でもほぼほぼ無料になる計算。

### 参考：EC2インスタンスの料金

#### インスタンスの使用量 + EBSの料金 + 通信料金 + オプション料金

- インスタンスの使用量：稼働時間×単価
  - EBSの料金：容量×単価
  - 通信料金：インスタンスからインターネットに向けた通信(アウトバウンド)に対してのみ発生
  - オプション料金：Elastic IPやRoute53など
- ✓ AWSアカウントを作成してから12か月間が経過しておらず、EC2の無料利用枠を使い切っていない場合、無料利用枠内で利用できるオプションを選択すると一切費用がかかりません。
  - ✓ 通常はインスタンスを起動したときから、インスタンスを削除するまで、アイドル状態のままでも標準の使用料が発生します。
  - ✓ AWS無料利用枠外でインスタンスを起動した場合は、インスタンスのステータスが休止または終了になるとインスタンスの課金が停止します。

**AWSでできることは他のクラウドサービスでもできます。**

**参考：他のクラウドサービスでEC2に相当するサービス**

クラウドサービス名	サービス名
AWS	EC2
Google Cloud	Compute Engine
Microsoft Azure	Virtual Machines

サービスの申し込み、ダッシュボードの利用方法等はそれぞれ異なりますが、互いに影響しあっているため、Linux学習環境の構築という観点では、いずれかのサービスでインスタンスが作成できれば、他のサービスでもさほど苦勞することなく環境構築ができるはずです。

## ■101試験

- 1.01.1 Linuxのインストール（インストール作業自体が無い）
- 1.01.2 仮想マシンの概念と利用（インスタンスタイプによる）
- 1.01.3 ブートプロセス（ログは確認できる）
- 1.01.5 デスクトップ環境の利用（条件によってはカバー可能）
- 1.05.1 ハードウェアの基礎知識と設定（ハードウェアは全て仮想化されている）
- 1.05.2 ハードディスクのレイアウトとパーティション（同上）
- 1.05.3 ファイルシステムの作成と管理、マウント

## ■102試験

- 1.11.1 オープンソースの概念とライセンス
- 1.11.2 オープンソースのコミュニティとエコシステム

クラウド学習環境ではカバーが難しい出題範囲については、実機またはVirtualBoxなど仮想マシン環境で補うようにしましょう。

- クラウドサービスを使うとリモート操作で短時間でLinuxサーバー環境が構築できるが、実際の運用にはLinuxを扱う上での基本的な知識が必須。  
→LinuC認定取得はクラウドエンジニアとして飛躍するための第一歩
- リモート接続が前提なので自分が何をしているのかイメージできるようにする。リモートでの複数ノードの管理などで混乱しないようにしたい。
- クラウド上でのLinux学習環境構築を通じて、LinuCの試験範囲にもなっているパブリッククラウドのセキュリティポイントを学ぶ。
- LinuC取得を目指すなら、クラウド上の学習環境だけではカバーしきれない出題範囲もあるので、実機や仮想の環境も構築して可能な限り試してみよう。
- パブリッククラウドサービスの無料枠を最大限活用して事業者ごとの違いを体感してみる。基本的には従量制のサービスなのでコストコントロールも意識する。