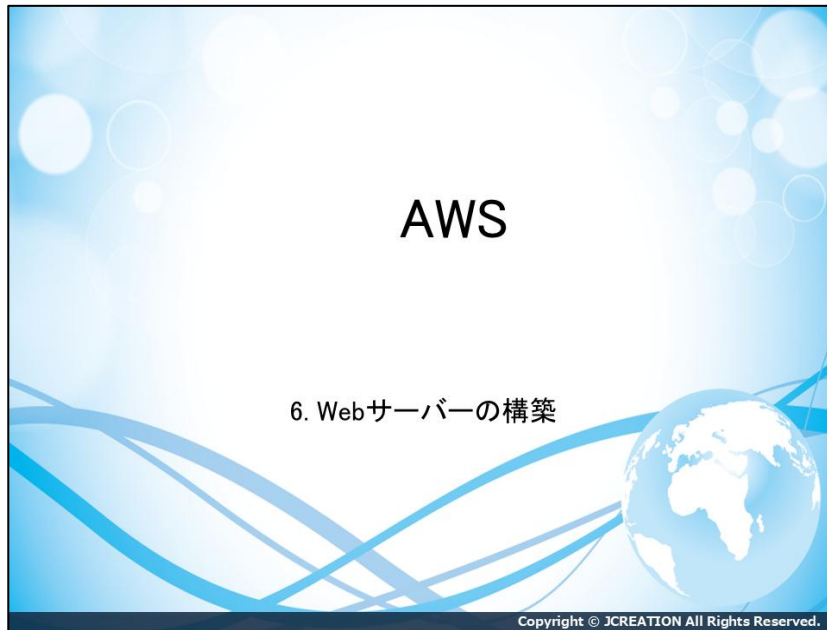
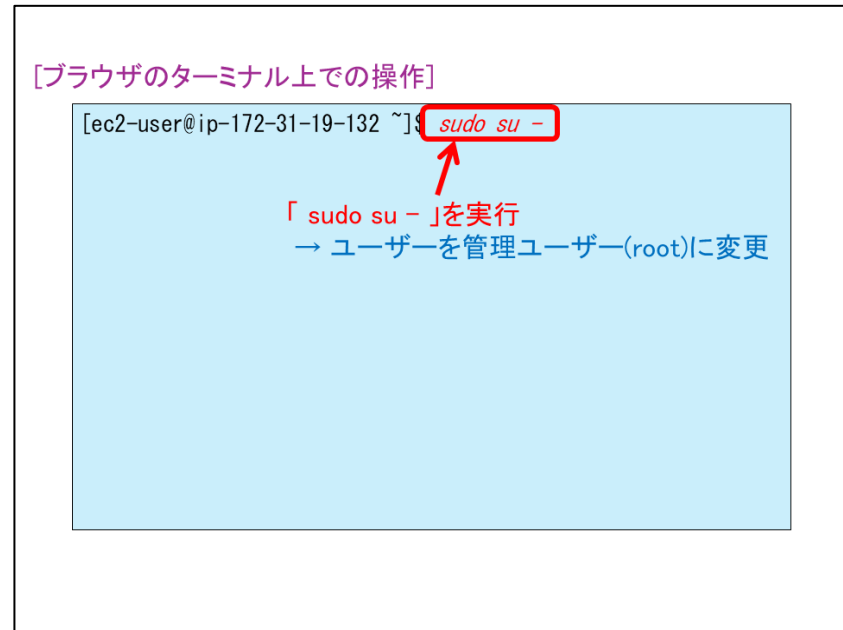
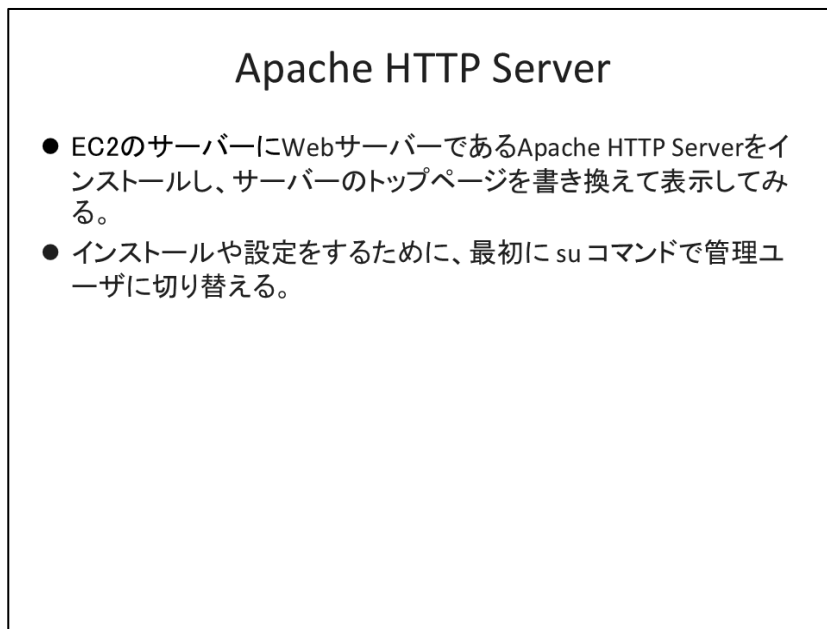




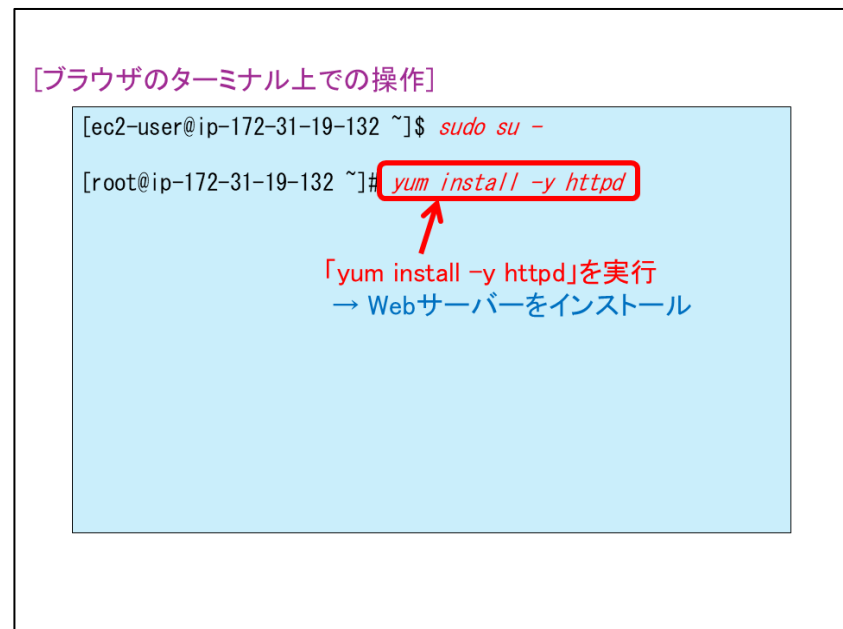
1



3



2



4

[ブラウザのターミナル上での操作]

```
[ec2-user@ip-172-31-19-132 ~]$ sudo su -  
[root@ip-172-31-19-132 ~]# yum install -y httpd  
  
Last metadata expiration check: 0:46:42 ago on Mon Aug 3  
04:05:16 2026.  
Dependencies resolved.  
=====
```

Package	Architecture
. . . (略) . . .	

```
Complete!  
[root@ip-172-31-19-132 ~]#
```

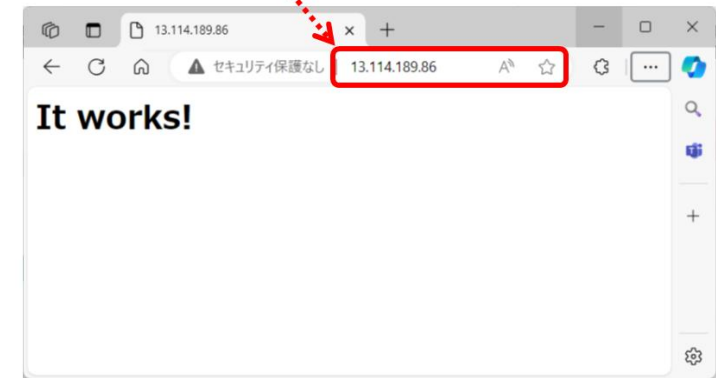
「Complete!」と表示されればインストール完了

5

インスタンスの概要

パブリック IPv4 アドレス
13.114.189.86 | [オープンアドレス](#)

IPv4アドレスをブラウザのURLに張り付ける



7

[ブラウザのターミナル上での操作]

```
[ec2-user@ip-172-31-19-132 ~]$ sudo su -  
[root@ip-172-31-19-132 ~]# yum install -y httpd  
  
Last metadata expiration check: 0:03:39 ago on Tue Aug 27  
07:56:10 2025.  
Dependencies resolved.  
=====
```

Package	Architecture
. . . (略) . . .	

```
Complete!  
[root@ip-172-31-19-132 ~]# systemctl start httpd
```

「systemctl start httpd」を実行
→ Webサーバーを起動

6

[ブラウザのターミナル上での操作]

```
[root@ip-172-31-19-132 ~]# echo "Yamada Taro" > /var/www/html/index.html
```

「echo "自分の名前" > /var/www/html/index.html」を実行
→ Webサーバーのトップページを書き換え

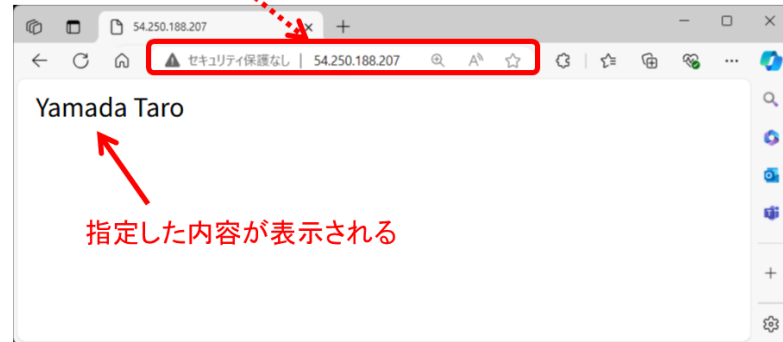
8

インスタンスの概要

パブリック IPv4 アドレス

13.114.189.86 | [オープンアドレス](#)

IPv4アドレスをブラウザのURLに張り付ける



指定した内容が表示される

9

タイムゾーンの設定

[ブラウザのターミナル上での操作]

- 管理ユーザーへの切り替え (既に実行済みの場合は不要)

```
$ sudo su -
```

- 現在のタイムゾーンの確認

時刻がUTCになっている

```
# timedatectl status
Local time: Sat 2025-08-03 05:50:17 UTC
Universal time: Sat 2025-08-03 05:50:17 UTC
RTC time: Sat 2025-08-03 05:50:17
Time zone: n/a (UTC, +0000)
System clock synchronized: yes
NTP service: active
RTC in local TZ: no
```

UTC: 協定世界時 (Universal time coordinated)
(イギリスで利用されているGMTがUTCと一致)

11

AWS

7. Linuxの設定

Copyright © JCREATION All Rights Reserved.

10

タイムゾーンの設定

[ブラウザのターミナル上での操作]

- 現在のタイムゾーンの変更と確認

時刻はJST (日本標準時)

```
# timedatectl set-timezone Asia/Tokyo ← タイムゾーンを設定
```

12

タイムゾーンの設定

[ブラウザのターミナル上での操作]

●現在のタイムゾーンの変更と確認

```
# timedatectl set-timezone Asia/Tokyo
# timedatectl status ← 確認
Local time: Sat 2025-08-03 14:51:17 JST
Universal time: Sat 2025-08-03 05:51:17 UTC
RTC time: Sat 2025-08-03 05:51:17
Time zone: Asia/Tokyo (JST, +0900)
System clock synchronized: yes
NTP service: active
RTC in local TZ: no
```

Local time: 地方時、現地時間

Universal time: 協定世界時

RTC time: リアルタイムクロック (real-time clock)

時刻はJST(日本標準時)



13

ロケールの変更

[ブラウザのターミナル上での操作]

●ロケール(言語・地域)の設定

```
# localectl status
System Locale: LANG=C.UTF-8
VC Keymap: (unset)
X11 Layout: (unset)

# localectl set-locale LANG=ja_JP.UTF-8 ← 日本語・日本に設定
```

15

ロケールの変更

[ブラウザのターミナル上での操作]

●ロケール(言語・地域)の設定

```
# localectl status ← 設定確認
System Locale: LANG=C.UTF-8
VC Keymap: (unset)
X11 Layout: (unset)
```

14

ロケールの変更

[ブラウザのターミナル上での操作]

●ロケール(言語・地域)の設定

```
# localectl status
System Locale: LANG=C.UTF-8
VC Keymap: (unset)
X11 Layout: (unset)

# localectl set-locale LANG=ja_JP.UTF-8

# localectl status ← 設定確認
System Locale: LANG=ja_JP.UTF-8 ← 日本語・日本になっている
VC Keymap: (unset)
X11 Layout: (unset)
```

16

Java SE21のインストール

[ブラウザのターミナル上での操作]

●利用可能なJavaパッケージの確認

```
# dnf list --available "java*dev*"
Last metadata expiration check: 0:11:29 ago on Sat Aug 3 06:18:10
2025.
Available Packages
java-1.8.0-amazon-corretto-devel.x86_64 1:1.8.0_422.b05-1.amzn2023
java-11-amazon-corretto-devel.x86_64 1:11.0.24+8-1.amzn2023
java-17-amazon-corretto-devel.x86_64 1:17.0.12+7-1.amzn2023.1
java-21-amazon-corretto-devel.x86_64 1:21.0.4+7-1.amzn2023.1
java-24-amazon-corretto-devel.x86_64 1:24.0.2+12-1.amzn2023.1
```

17

Java SE21のインストール

[ブラウザのターミナル上での操作]

●インストールされているJavaバージョンの確認

```
# java --version
openjdk 21.0.8 2025-07-15 LTS ← SE21がインストールされている
OpenJDK Runtime Environment Corretto-21.0.8.9.1
(build 21.0.8+9-LTS)
OpenJDK 64-Bit Server VM Corretto-21.0.8.9.1
(build 21.0.8+9-LTS, mixed mode, sharing)
```

19

Java SE21のインストール

[ブラウザのターミナル上での操作]

●Java SE21のインストール

```
# dnf -y install java-21-amazon-corretto-devel
Last metadata expiration check: 0:16:13 ago on Sat Aug 3 06:18:10
2025.
Dependencies resolved.
=====
Package                        Architecture      Version
=====
Installing:
java-21-amazon-corretto-devel  x86_64            1:21.0.4+7-1.
alsa-lib                       x86_64            1.2.7.2-1.am
cairo                          x86_64            1.17.6-2.ama
--- 略 ---
Complete! ← インストール完了
```

18

Tomcatのインストール

[ブラウザのターミナル上での操作]

●tomcat用ユーザーの作成

```
# useradd -r -s /sbin/nologin tomcat ← ユーザーの作成
```

20

Tomcatのインストール

[ブラウザのターミナル上での操作]

●tomcat用ユーザーの作成

```
# useradd -r -s /sbin/nologin tomcat

# id tomcat ← ユーザーの確認
uid=992(tomcat) gid=992(tomcat) groups=992(tomcat)
```

21

Tomcatのインストール

[ブラウザのターミナル上での操作]

●Tomcatパッケージのダウンロードとインストール

```
# wget https://dlcdn.apache.org/tomcat/tomcat-10/
/v10.1.57/bin/apache-tomcat-10.1.57.tar.gz } 1行で書く
--2025-08-03 07:41:09-- https://dlcdn.apache.org/tomcat/tomc
... 略 ...
```

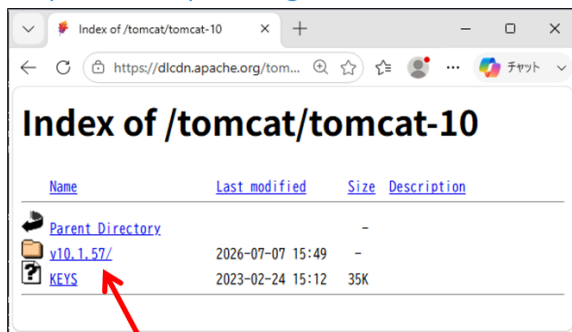
23

Tomcatのインストール

●最新パッケージの確認

次のURLでパッケージを確認できる。

<https://dlcdn.apache.org/tomcat/tomcat-10/>



最新バージョンは 10.1.57 (2026年8月3日時点)

22

Tomcatのインストール

今回用

[ブラウザのターミナル上での操作]

●Tomcatパッケージのダウンロードとインストール

```
# wget https://dlcdn.apache.org/tomcat/tomcat-10/
/v10.1.57/bin/apache-tomcat-10.1.57.tar.gz } 1行で書く
--2025-08-03 07:41:09-- https://dlcdn.apache.org/tomcat/tomc
... 略 ...
```

`wget https://javax.biz/apache-tomcat-10.1.57.tar.gz`

入力が長いので、今回用に用意した簡単なURLを使う

24

Tomcatのインストール

[ブラウザのターミナル上での操作]

●Tomcatパッケージのダウンロードとインストール

```
# wget https://dlcdn.apache.org/tomcat/tomcat-10/  
/v10.1.57/bin/apache-tomcat-10.1.57.tar.gz  
--2025-08-03 07:41:09-- https://dlcdn.apache.org/tomcat/tomc  
... 略 ...  
  
# tar xvfz apache-tomcat-10.1.57.tar.gz ← ファイルを展開  
apache-tomcat-10.1.57/conf/  
apache-tomcat-10.1.57/conf/catalina.policy  
apache-tomcat-10.1.57/conf/catalina.properties  
... 略 ...
```

25

Tomcatのインストール

[ブラウザのターミナル上での操作]

```
# mv apache-tomcat-10.1.57 /opt/apache-tomcat ← ファイルを移動  
カレントディレクトリにある apache-tomcat-10.1.57 を  
/optの下に apache-tomcat という名前で移動
```

27

Tomcatのインストール

[ブラウザのターミナル上での操作]

●Tomcatパッケージのダウンロードとインストール

```
# wget https://dlcdn.apache.org/tomcat/tomcat-10/  
/v10.1.57/bin/apache-tomcat-10.1.57.tar.gz  
--2025-08-03 07:41:09-- https://dlcdn.apache.org/tomcat/tomc  
... 略 ...  
  
# tar xvfz apache-tomcat-10.1.57.tar.gz  
apache-tomcat-10.1.57/conf/  
apache-tomcat-10.1.57/conf/catalina.policy  
apache-tomcat-10.1.57/conf/catalina.properties  
... 略 ...  
  
# ls ← ファイルの一覧表示  
apache-tomcat-10.1.57    apache-tomcat-10.1.57.tar.gz
```

展開して作成されたディレクトリ

ダウンロードしたファイル

26

Tomcatのインストール

[ブラウザのターミナル上での操作]

```
# mv apache-tomcat-10.1.57 /opt/apache-tomcat  
  
# chown -R tomcat:tomcat /opt/apache-tomcat ← 所有者を変更  
ファイルの所有者とグループを tomcat に変更
```

28

Tomcatのインストール

[ブラウザのターミナル上での操作]

```
# mv apache-tomcat-10.1.57 /opt/apache-tomcat
# chown -R tomcat:tomcat /opt/apache-tomcat
# vi /usr/lib/systemd/system/tomcat.service ←まだ実行しない！
```

↑
エディター vi で設定ファイルを作成→中身は次のスライド
→ vi が使いにくいようであれば nano エディターを使う

29

Tomcatのインストール 今回用

[ブラウザのターミナル上での操作]

```
# mv apache-tomcat-10.1.57 /opt/apache-tomcat ←ファイルを移動
# chown -R tomcat:tomcat /opt/apache-tomcat ←所有者を変更
# vi /usr/lib/systemd/system/tomcat.service
```

↑
エディター vi で設定ファイルを作成→中身は次のスライド
→ vi が使いにくいようであれば nano エディターを使う
編集は面倒なので、今回用に用意した簡単な設定を使う

```
# cd /usr/lib/systemd/system ←作業ディレクトリを移動
```

31

Tomcatのインストール(設定ファイル)

```
# Systemd unit file for default tomcat
[Unit]
Description=Apache Tomcat Web Application Container
After=syslog.target network.target

[Service]
Type=oneshot
PIDFile=/opt/apache-tomcat/tomcat.pid
RemainAfterExit=yes
ExecStart=/opt/apache-tomcat/bin/startup.sh
ExecStop=/opt/apache-tomcat/bin/shutdown.sh
ExecReStart=/opt/apache-tomcat/bin/shutdown.sh:/opt/apache-tomcat/bin/startup.sh
SuccessExitStatus=143
User=tomcat
Group=tomcat

[Install]
WantedBy=multi-user.target
```

エディターを使ってファイルを編集

viでは次のように操作する:
① i キーを押して挿入モードに切り替える
② 内容を編集
③ ESC キーを押して :wq と入力して終了

30

Tomcatのインストール 今回用

[ブラウザのターミナル上での操作]

```
# mv apache-tomcat-10.1.57 /opt/apache-tomcat ←ファイルを移動
# chown -R tomcat:tomcat /opt/apache-tomcat ←所有者を変更
# vi /usr/lib/systemd/system/tomcat.service
```

↑
エディター vi で設定ファイルを作成→中身は次のスライド
→ vi が使いにくいようであれば nano エディターを使う
編集は面倒なので、今回用に用意した簡単な設定を使う

```
# cd /usr/lib/systemd/system ←作業ディレクトリを移動
# wget https://javax.biz/tomcat.service ←ファイルのダウンロード
```

32

Tomcatのインストール

[ブラウザのターミナル上での操作]

●自動起動の設定

```
# systemctl enable tomcat.service ← 自動起動の設定
Created symlink /etc/systemd/... → /usr/lib/systemd/system/...
```

33

Tomcatのインストール

[ブラウザのターミナル上での操作]

●自動起動の設定

```
# systemctl enable tomcat.service
Created symlink /etc/systemd/... → /usr/lib/systemd/system/...

# systemctl start tomcat.service

# curl -s http://localhost:8080 ← 動作確認(ローカル接続)
                                (まだ外部からはつながらない...)
<!DOCTYPE html> ← HTMLのソースが表示されればOK
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <title>Apache Tomcat/10.1.57</title>
    <link href="favicon.ico" rel="icon" type="image/x-ico
    <link href="tomcat.css" rel="stylesheet" type="text/c
    ...略...
```

35

Tomcatのインストール

[ブラウザのターミナル上での操作]

●自動起動の設定

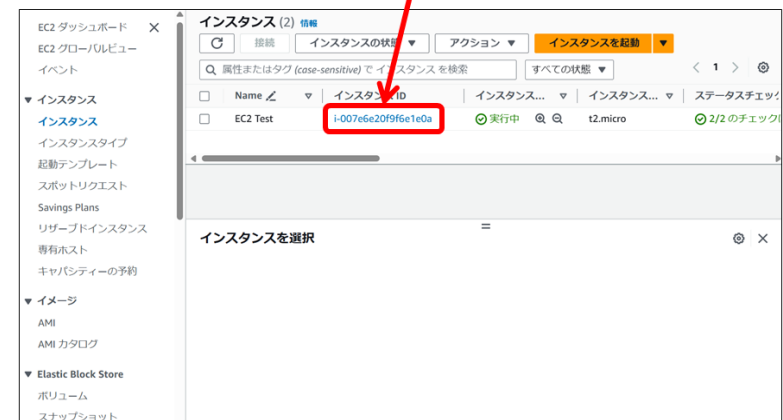
```
# systemctl enable tomcat.service
Created symlink /etc/systemd/... → /usr/lib/systemd/system/...

# systemctl start tomcat.service ← Tomcatの起動
```

34

外部接続の設定

インスタンスIDをクリック



36

外部接続の設定

ページをやや下に降りて「セキュリティ」をクリック

The screenshot shows the AWS Management Console interface. The top navigation bar has tabs for '詳細' (Details), 'ステータスとアラーム' (Status & Alarms), 'モニタリング' (Monitoring), 'セキュリティ' (Security), 'ネットワーク' (Network), 'ストレージ' (Storage), and 'タグ' (Tags). The 'セキュリティ' tab is highlighted with a red box and a red arrow pointing to it from the text above.

37

外部接続の設定

「インバウンドのルールを編集」をクリック

The screenshot shows the 'インバウンドルール' (Inbound Rules) section of the AWS Management Console. A red box highlights the 'インバウンドのルールを編集' (Edit inbound rules) button, with a red arrow pointing to it from the text above.

39

外部接続の設定

セキュリティグループ名をクリック

The screenshot shows the 'セキュリティ' (Security) section of the AWS Management Console. Under the 'セキュリティグループ' (Security Groups) heading, the security group 'sg-0c73c94462517e162 (launch-wizard-1)' is highlighted with a red box and a red arrow pointing to it from the text above.

38

外部接続の設定

「ルールを追加」をクリック

The screenshot shows the 'インバウンドのルールを編集' (Edit inbound rules) section of the AWS Management Console. A red box highlights the 'ルールを追加' (Add rule) button, with a red arrow pointing to it from the text below.

40

外部接続の設定

インバウンドのルールを編集 [情報](#)

インバウンドルールは、インスタンスに到達できる番付トラフィックをコントロールします。

セキュリティグループルール ID	タイプ	プロトコル	ポート範囲	ソース	説明 - オプション	削除
sg-061f53681cf59d331	SSH	TCP	22	カス...	0.0.0.0/0 X	
sg-0f29afd4722fd1a8	HTTP	TCP	80	カス...	0.0.0.0/0 X	
sg-0c28d4a5fe0a2bd42	HTTPS	TCP	443	カス...	0.0.0.0/0 X	
-	カスタム TCP	TCP	8080	カス...	0.0.0.0/0 X	

[ルールを追加](#)

「8080」入力

41

外部接続の設定

インバウンドのルールを編集 [情報](#)

インバウンドルールは、インスタンスに到達できる番付トラフィックをコントロールします。

セキュリティグループルール ID	タイプ	プロトコル	ポート範囲	ソース	説明 - オプション	削除
sg-061f53681cf59d331	SSH	TCP	22	カス...	0.0.0.0/0 X	
sg-0f29afd4722fd1a8	HTTP	TCP	80	カス...	0.0.0.0/0 X	
sg-0c28d4a5fe0a2bd42	HTTPS	TCP	443	カス...	0.0.0.0/0 X	
-	カスタム TCP	TCP	8080	Any...	0.0.0.0/0 X	

[ルールを追加](#)

キャンセル [変更をプレビュー](#) [ルールを保存](#)

「ルールを保存」をクリック

43

外部接続の設定

インバウンドのルールを編集 [情報](#)

インバウンドルールは、インスタンスに到達できる番付トラフィックをコントロールします。

セキュリティグループルール ID	タイプ	プロトコル	ポート範囲	ソース	説明 - オプション	削除
sg-061f53681cf59d331	SSH	TCP	22	カス...	0.0.0.0/0 X	
sg-0f29afd4722fd1a8	HTTP	TCP	80	カス...	0.0.0.0/0 X	
sg-0c28d4a5fe0a2bd42	HTTPS	TCP	443	カス...	0.0.0.0/0 X	
-	カスタム TCP	TCP	8080	カス...	0.0.0.0/0 X	

[ルールを追加](#)

枠をクリックして「0.0.0.0/0」を選択

42

外部接続の設定

aws サービス [Alt+S]

EC2 ダッシュボード X

EC2 グローバルビュー

イベント

▼ インスタンス

インスタンスタイプ

起動テンプレート

スポット リクエスト

Savings Plans

リザーブドインスタン

専用ホス

キャパシティーの予約

▼ イメージ

AMI

AMI カタログ

インバウンドセキュリティグループのルールが、セキュリティグループで正常に変更されました (sg-0c73c94462517e162 - launch-wizard-1)

詳細

EC2 > セキュリティグループ > sg-0c73c94462517e162 - launch-wizard-1

sg-0c73c94462517e162 - launch-wizard-1

詳細

セキュリティグループ名	セキュリティグループ ID	説明
launch-wizard-1	sg-0c73c94462517e162	launch-wizard-1 c 2024-08-03T00:29:11

所有者 043684925296

インバウンドルールカウン

4 アクセス許可エントリ

アウトバウンドル

1 アクセス許可エントリ

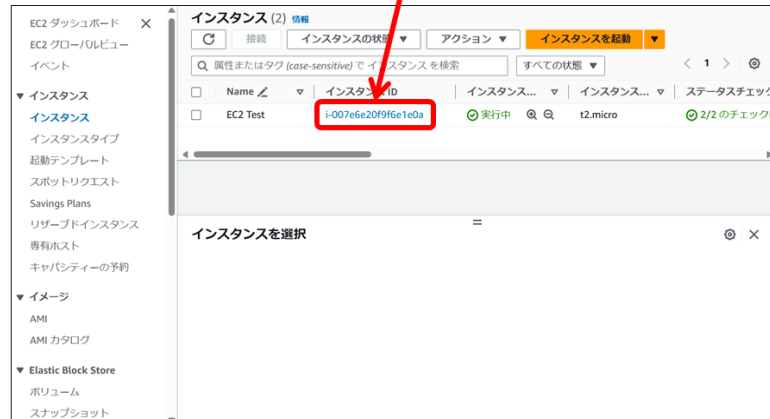
インバウンドルール アウトバウンドルール タグ

「インスタンス」をクリック

44

外部接続の設定

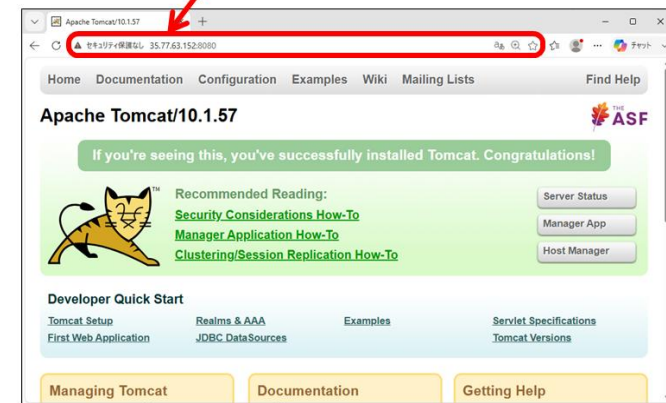
インスタンスIDをクリック



45

外部接続の設定

コピーしたアドレス: 8080



Tomcatに接続してページが表示される

47

外部接続の設定



「パブリックIPv4アドレス」をコピー

46

AWS

8. Webアプリの公開

Copyright © JCREATION All Rights Reserved.

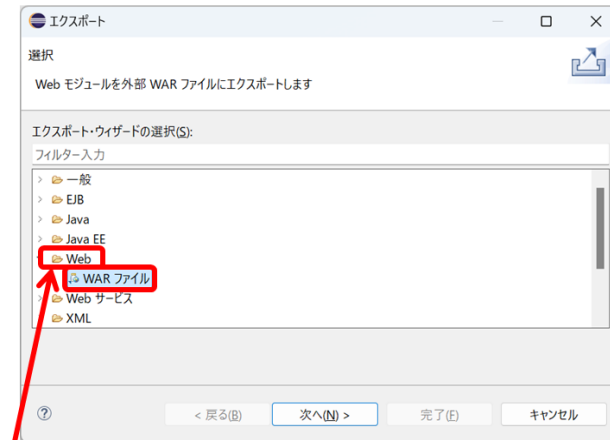
48

デプロイ

- Webアプリケーションをサーバー上に配置・展開し、利用可能な状態にすることをデプロイという。
- JavaベースのWebアプリケーションのデプロイにはWAR(Web Application Archive)ファイルが使用される。WARファイルには、Javaのクラスファイル、JARファイル、設定ファイル、JSPページ、HTMLページなどが含めることができる。
- ここでは、EclipseでWARファイルを作成し、AWSのEC2上にWebアプリを配置してインターネット上からアクセスできるようにする。

49

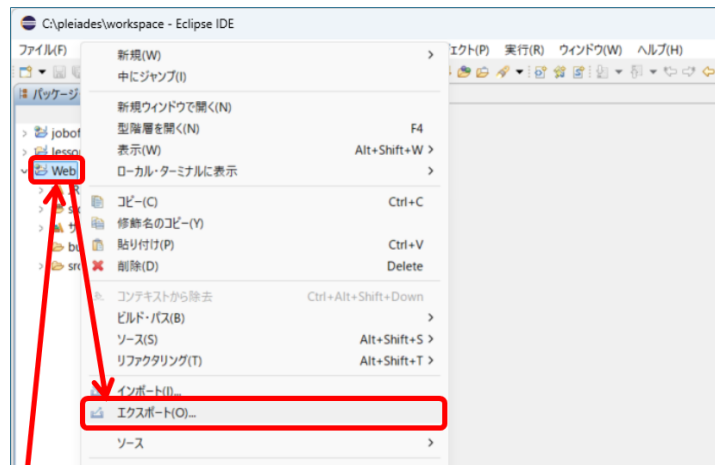
WARファイルの作成



「Web → WARファイル」をクリック

51

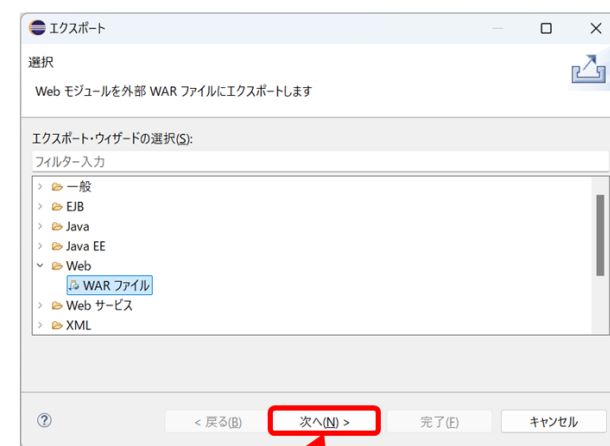
WARファイルの作成



プロジェクト名Webを右クリックして「エクスポート」を選択

50

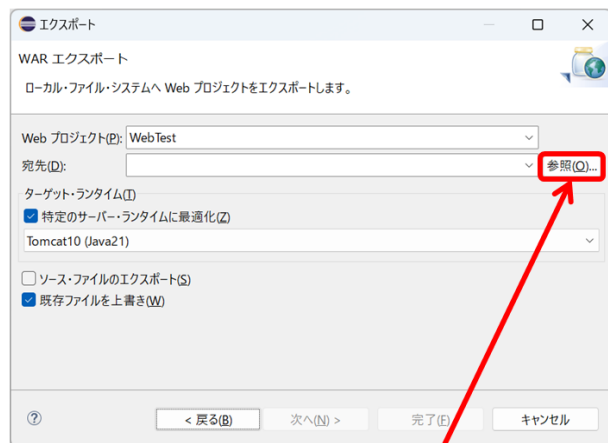
WARファイルの作成



「次へ」をクリック

52

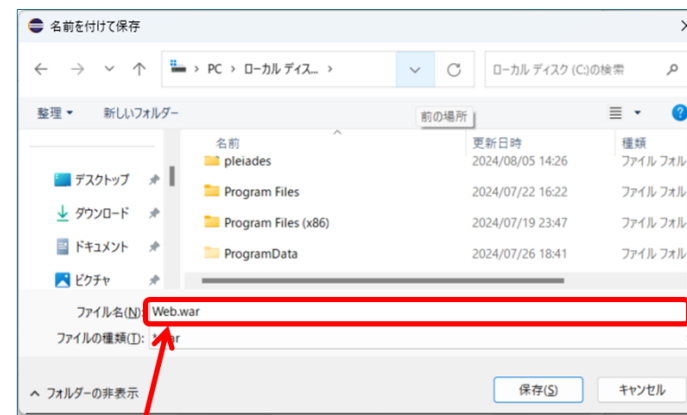
WARファイルの作成



「次へ」をクリック

53

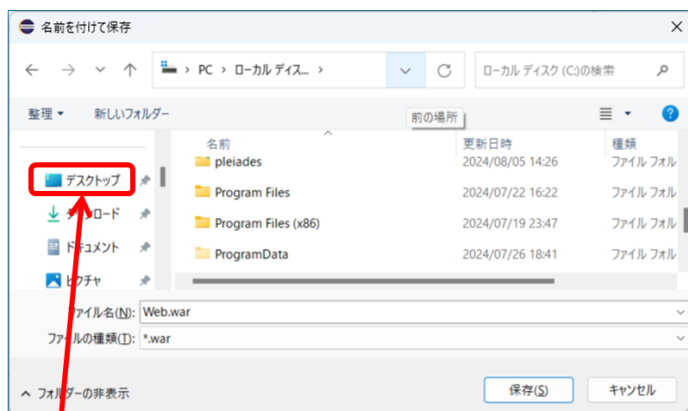
WARファイルの作成



ファイル名には「Web.war」が入力されている
(拡張子は表示されないこともある)

55

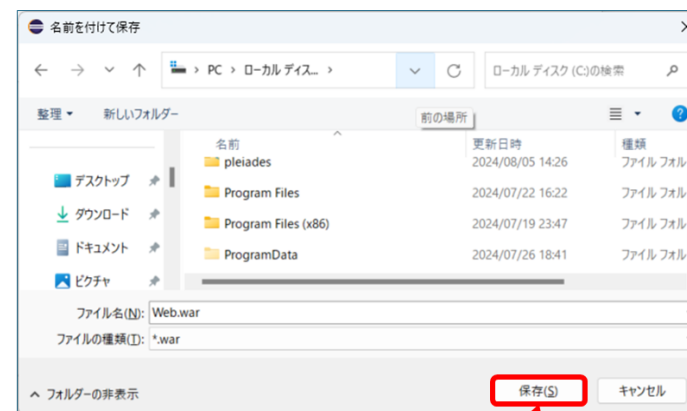
WARファイルの作成



「デスクトップ」をクリック

54

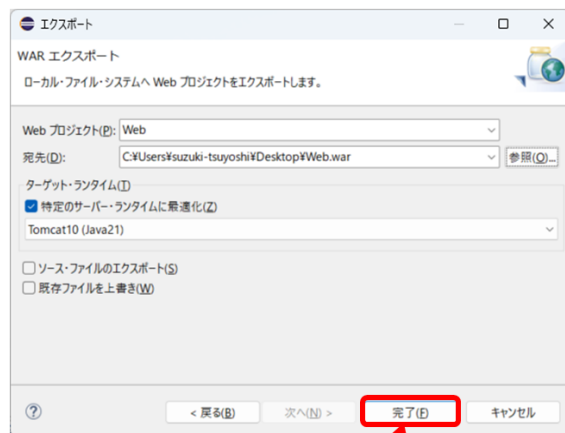
WARファイルの作成



「保存」をクリック

56

WARファイルの作成



「完了」をクリック

57

EC2へwarファイルをコピー

- サーバー側でSSH接続が可能な場合にはscpコマンドを使ってファイルのコピーを行うことができる。
- EC2にローカルにあるファイルをコピーするには次のようにする。

```
scp -i キーファイル コピー元ファイル ec2-user@IPアドレス:コピー先
```
- scpでコピーを行うにはコピー先のサーバーのアドレスを確認する必要がある。

59

WARファイルの作成

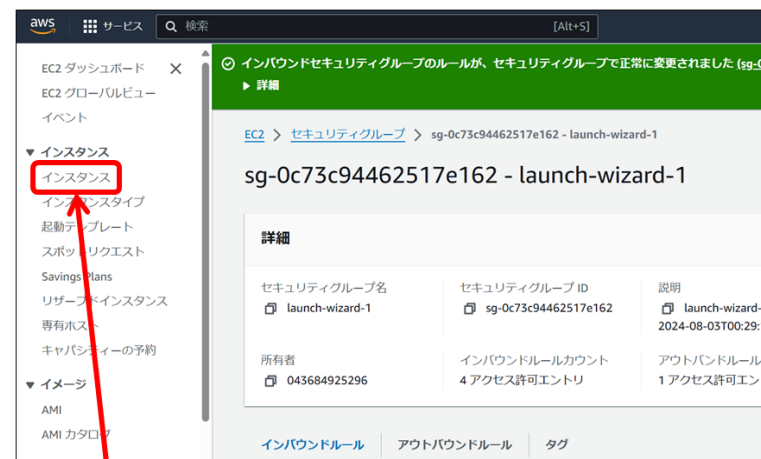


Web.war

デスクトップに Web.war が作成される

58

EC2のアドレスを確認

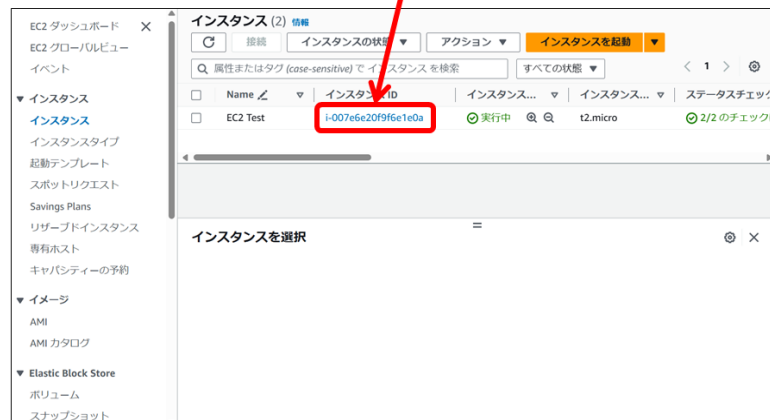


「インスタンス」をクリック

60

EC2のアドレスを確認

インスタンスIDをクリック



61

EC2へwarファイルをコピー

- サーバー側でSSH接続が可能な場合にはscpコマンドを使ってファイルのコピーを行うことができる。



Windowsのデスクトップの何もないところを右クリックして
「ターミナルで開く」を選択

63

EC2のアドレスを確認

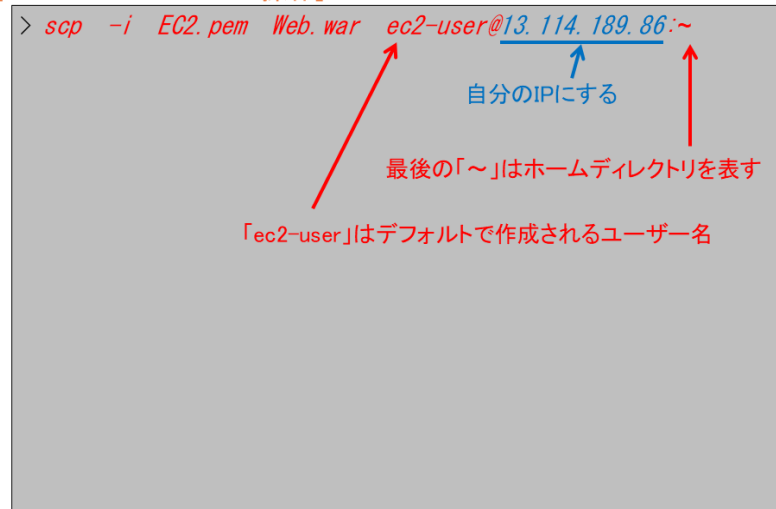


「パブリックIPv4アドレス」をコピー

62

ファイルのコピー

[コマンドプロンプトでの操作]



64

ファイルのコピー

[コマンドプロンプトでの操作]

```
> scp -i EC2.pem Web.war ec2-user@13.114.189.86:~
```

自分のIPにする

The authenticity of host '13.114.189.86 (13.114.189.86)' can't be established. ED25519 key fingerprint is SHA256: blpn9WlvxttrcdmOYwbhlq+qirOkBvm3f704HnNB3eM. This key is not known by any other names Are you sure you want to continue connecting (yes/no/[fingerprint])?

「yes」と入力して[Enter]を押す
→ 文字は表示されない

65

ファイルのコピー

[ブラウザのターミナル上での操作]

warファイルをtomcatの配置ディレクトリにコピー

```
# ls /home/ec2-user ← ファイルの一覧表示
Web.war ← コピーしたwarファイル

# cp /home/ec2-user/Web.war /opt/apache-tomcat/webapps/

# ls /opt/apache-tomcat/webapps ← ファイルの確認
ROOT Web Web.war docs examples host-manager ma
```

展開されたファイル → 自動的に展開される
コピーしたwarファイル

67

ファイルのコピー

[コマンドプロンプトでの操作]

```
> scp -i EC2.pem Web.war ec2-user@13.114.189.86:~
```

自分のIPにする

The authenticity of host '13.114.189.86 (13.114.189.86)' can't be established. ED25519 key fingerprint is SHA256: blpn9WlvxttrcdmOYwbhlq+qirOkBvm3f704HnNB3eM. This key is not known by any other names Are you sure you want to continue connecting (yes/no/[fingerprint])?

「yes」と入力して[Enter]を押す
→ 文字は表示されない

Warning: Permanently added '13.114.189.86' (ED25519) to the list of known hosts.

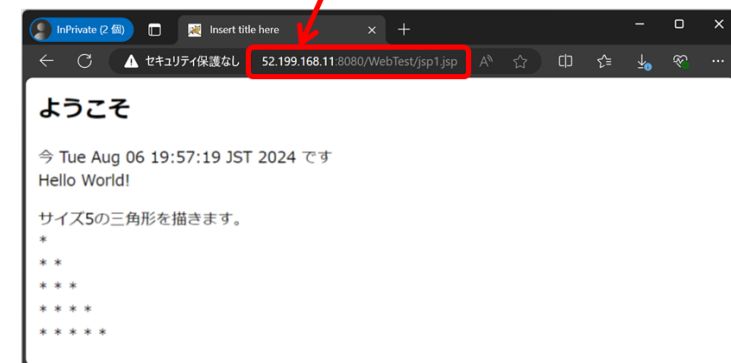
Web.war 100% 4596 405.6KB/s 00:00

100%と表示されれば完了

66

動作確認

http://(IPアドレス):8080/Web/jsp1.jsp



JSPの画面が表示される

68

演習問題

- 昨日作成したMyMemoアプリをAWSのEC2サーバーにデプロイし、インターネットから使用できるようにせよ。

注. アプリを更新したときにはtomcatの再起動が必要となる。
ブラウザのコンソールから管理者で次を実行する。
`systemctl restart tomcat`