

Jboss-Wildfly Deployment

Step:1 (Build server)

*Launch instance name as Build server.

Launch an instance [info](#)
Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags [info](#)
Name
 [Add additional tags](#)

▼ Application and OS Images (Amazon Machine Image) [info](#)
An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose **Browse more AMIs**.

Recents

Quick Start

Amazon Linux
aws

macOS
Mac

Ubuntu
ubuntu

Windows
Microsoft

Red Hat
Red Hat

SUSE Linux
SUSE

Debian
debian

>

[Browse more AMIs](#)
Including AMIs from AWS, Marketplace and the Community

▼ Summary

Number of instances [info](#)

Software Image (AMI)
Canonical, Ubuntu, 24.04, amd6...[read more](#)
ami-0a716d3f3b16d290c

Virtual server type (instance type)
t3.micro

Firewall (security group)
New security group

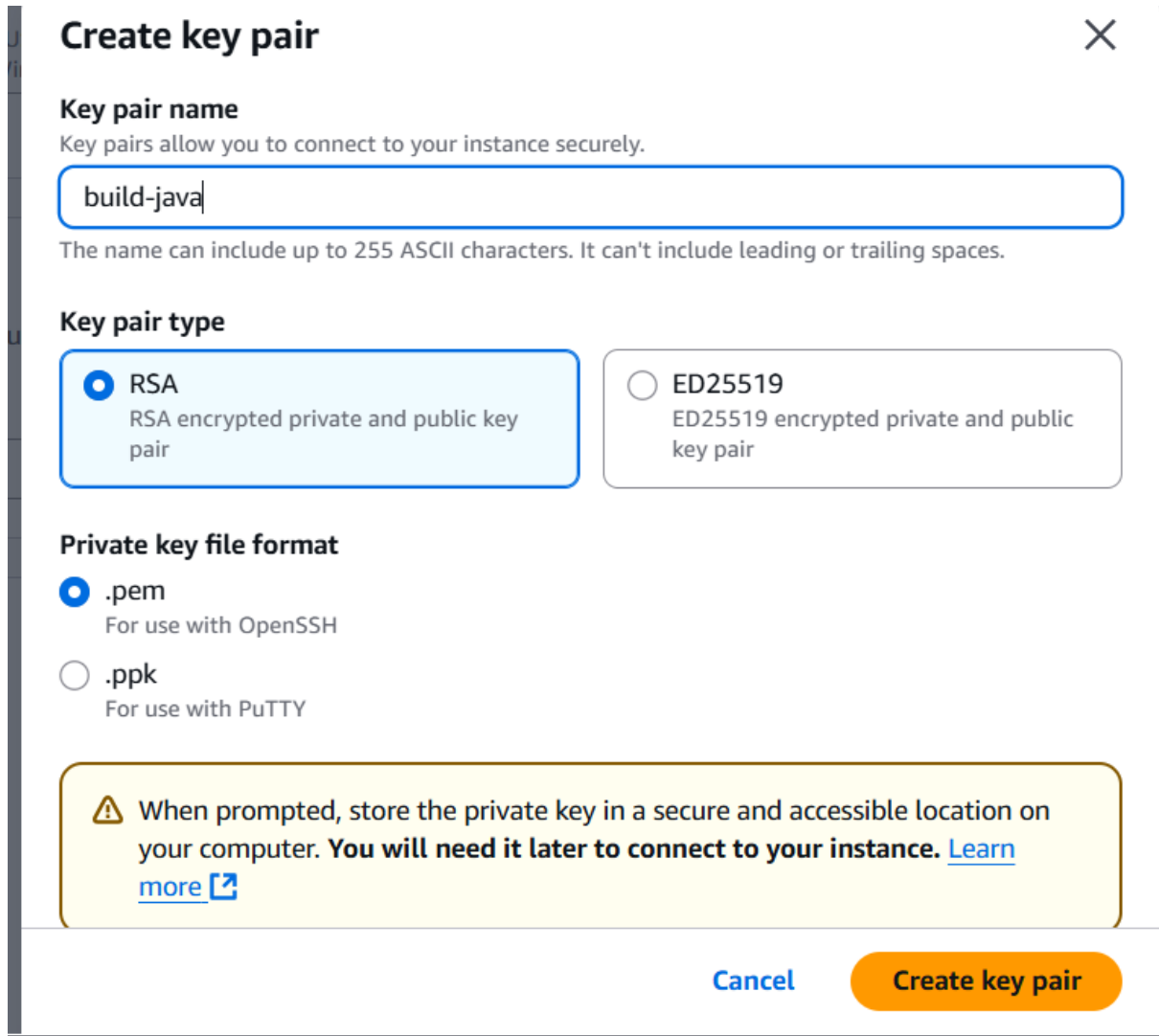
Storage (volumes)
1 volume(s) - 8 GiB

Cancel

Launch instance

[Preview code](#)

*Create key-pair.



The screenshot shows the 'Create key pair' dialog box. At the top, the title 'Create key pair' is followed by a close button (X). Below the title, the section 'Key pair name' has a text input field containing 'build-java'. A note below the field states: 'Key pairs allow you to connect to your instance securely. The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.' The 'Key pair type' section has two options: 'RSA' (selected with a blue radio button) and 'ED25519' (unselected). The RSA option is described as 'RSA encrypted private and public key pair'. The ED25519 option is described as 'ED25519 encrypted private and public key pair'. The 'Private key file format' section has two options: '.pem' (selected with a blue radio button) and '.ppk' (unselected). The .pem option is described as 'For use with OpenSSH'. The .ppk option is described as 'For use with PuTTY'. A yellow warning box at the bottom contains a warning icon and the text: 'When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance. [Learn more](#)'. At the bottom right, there are two buttons: 'Cancel' and 'Create key pair'.

Create key pair

Key pair name
Key pairs allow you to connect to your instance securely.
build-java
The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

☒ **RSA**
RSA encrypted private and public key pair

☐ **ED25519**
ED25519 encrypted private and public key pair

Private key file format

☒ **.pem**
For use with OpenSSH

☐ **.ppk**
For use with PuTTY

Warning: When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance. [Learn more](#)

Cancel **Create key pair**

Step 2

*sudo apt update -y

```
/usr/bin/xauth: file /home/ubuntu/.Xauthority does not exist
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

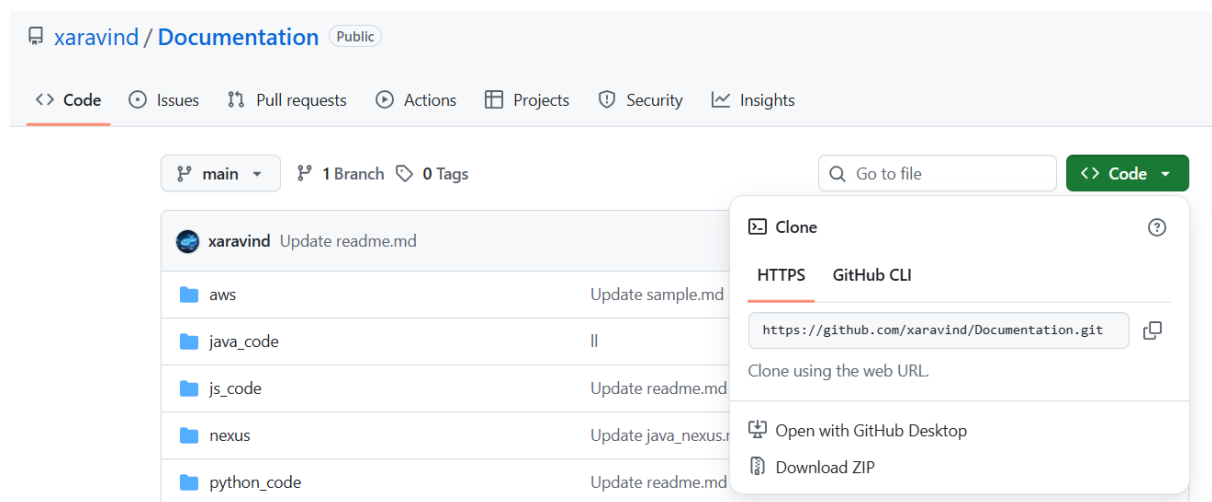
ubuntu@ip-172-31-16-28:~$ sudo apt update -y
```

*Install java

```
ubuntu@ip-172-31-16-28:~$ java --version
Command 'java' not found, but can be installed with:
sudo apt install openjdk-17-jre-headless # version 17.0.16+8~us1-0ubuntu1~24.04.1, or
sudo apt install openjdk-21-jre-headless # version 21.0.8+9~us1-0ubuntu1~24.04.1
sudo apt install default-jre # version 2:1.17-75
sudo apt install openjdk-11-jre-headless # version 11.0.28+6-1ubuntu1~24.04.1
sudo apt install openjdk-8-jre-headless # version 8u462-ga~us1-0ubuntu2~24.04.2
sudo apt install openjdk-19-jre-headless # version 19.0.2+7-4
sudo apt install openjdk-20-jre-headless # version 20.0.2+9-1
sudo apt install openjdk-22-jre-headless # version 22~22ea-1
ubuntu@ip-172-31-16-28:~$ sudo apt install openjdk-17-jre-headless
```

*Install maven

*Clone the code form github.



*git clone

```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-16-28:~$ git clone https://github.com/xaravind/Documentation.git
Cloning into 'Documentation'...
remote: Enumerating objects: 1845, done.
remote: Counting objects: 100% (30/30), done.
remote: Compressing objects: 100% (23/23), done.
remote: Total 1845 (delta 13), reused 1 (delta 1), pack-reused 1815 (from 3)
Receiving objects: 100% (1845/1845), 13.96 MiB | 28.60 MiB/s, done.
```

*mvn package(Build success)

```
[INFO] Packaging webapp
[INFO] Assembling webapp [WebCarRental] in [/home/ubuntu/Documentation/java_code/target/WebCarRental]
[INFO] Processing war project
[INFO] Copying webapp resources [/home/ubuntu/Documentation/java_code/src/main/webapp]
[INFO] Building war: /home/ubuntu/Documentation/java_code/target/WebCarRental.war
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 29.333 s
[INFO] Finished at: 2025-10-07T11:33:41Z
[INFO] -----
ubuntu@ip-172-31-16-28:~/Documentation/java_code$
```

Step-3

*Launch 2nd instance named as Jboss Deploy

Launch an instance [Info](#)

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags [Info](#)

Name

 [Add additional tags](#)

▼ **Application and OS Images (Amazon Machine Image)** [Info](#)

An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose [Browse more AMIs](#).

Recents **Quick Start**

Amazon Linux

macOS

Ubuntu

Windows

Red Hat

SUSE Linux

Debian

[Browse more AMIs](#)
Including AMIs from AWS, Marketplace and the Community

▼ **Summary**

Number of instances [Info](#)

Software Image (AMI)
Canonical, Ubuntu, 24.04, amd64...[read more](#)
ami-0a716d3f3b16d290c

Virtual server type (instance type)
t3.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

[Cancel](#) [Launch instance](#) [Preview code](#)

- Install jboss-wildfly

```
ubuntu@ip-172-31-20-253:~$ sudo wget https://github.com/wildfly/wildfly/releases/download/38.0.0.Beta1/wildfly-preview-38.0.0.Beta1.tar.gz
```

*untar jboss-wildfly

```
ubuntu@ip-172-31-20-253:~$ ls
wildfly-preview-38.0.0.Beta1.tar.gz
ubuntu@ip-172-31-20-253:~$ tar -xvf wildfly-preview-38.0.0.Beta1.tar.gz
```

*Rename wildfly.

```
ubuntu@ip-172-31-20-253:~$ ls
wildfly-preview-38.0.0.Beta1  wildfly-preview-38.0.0.Beta1.tar.gz
ubuntu@ip-172-31-20-253:~$ mv wildfly-preview-38.0.0.Beta1 wildfly
ubuntu@ip-172-31-20-253:~$ ls
wildfly  wildfly-preview-38.0.0.Beta1.tar.gz
```

*Start wildfly .

```
domain.conf.ps1      jdr.bat      wsprovide.sh
ubuntu@ip-172-31-20-253:~/wildfly/bin$ ./standalone.sh -b=0.0.0.0
=====
```

Step-4

*Create ssh-keygen.

*Here pub id will be created copy this keys and paste in wildfly authorized keys.

Step-5

*cd .ssh

*vi authorized keys

```
ubuntu@ip-172-31-16-28:~/.ssh$ ls
authorized_keys  id_ed25519  id_ed25519.pub
ubuntu@ip-172-31-16-28:~/.ssh$ cat id_ed25519.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAICwLa0pEiX+wXSNaDzDJ9gHa4qi4sUg5bldik9fh/7qU  ubuntu@ip-172-31-16-28
ubuntu@ip-172-31-16-28:~/.ssh$ cd
```

Step-6

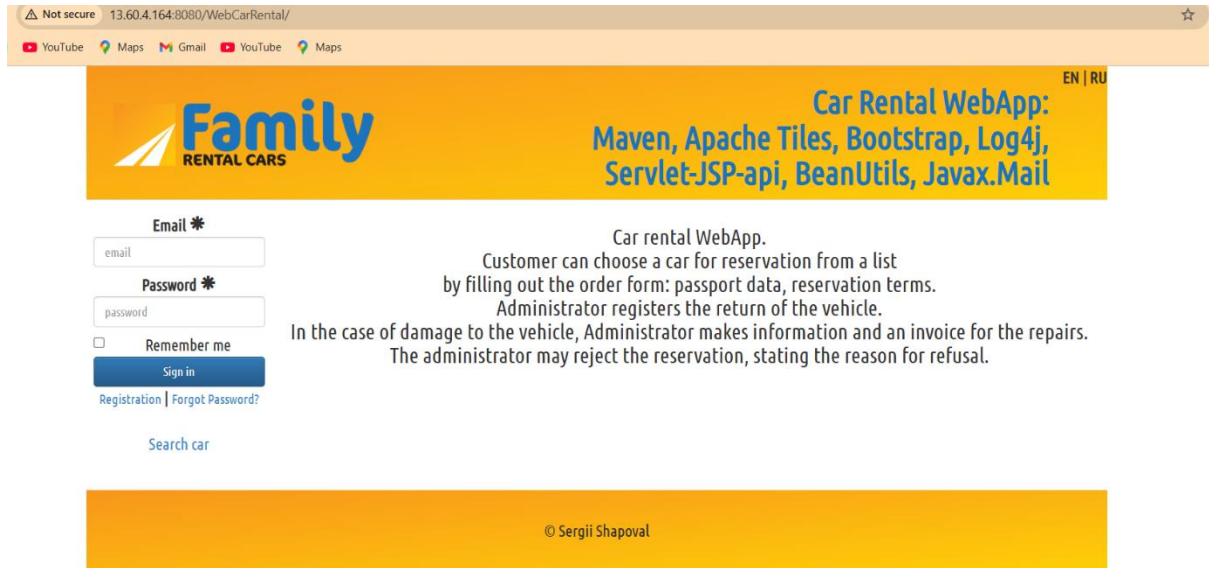
*Communicate this two servers by scp

```
ubuntu@ip-172-31-16-28:~/Documentation/java_code/target$ scp /home/ubuntu/Documentation/java_code/target/*.war ubuntu@13.60.4.164:/home/ubuntu/wildfly/standalone/deployments
The authenticity of host '13.60.4.164 (13.60.4.164)' can't be established.
ED25519 key fingerprint is SHA256:RvEcQ6cmIYjKKlhneR4N/XbHeL2id4ed3CxuwpSfDGs.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '13.60.4.164' (ED25519) to the list of known hosts.
WebCarRental.war                                100% 3932KB  28.2MB/s   00:00
ubuntu@ip-172-31-16-28:~/Documentation/java_code/target$ ^C
ubuntu@ip-172-31-16-28:~/Documentation/java_code/target$ █
```

GISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

* Here war file is created

* We can access wildfly by publicip ,port number 8080 and app name.



The screenshot shows a web browser window with the address bar displaying "13.60.4.164:8080/WebCarRental/". The page features a yellow header with the "Family RENTAL CARS" logo on the left and the text "Car Rental WebApp: Maven, Apache Tiles, Bootstrap, Log4j, Servlet-JSP-api, BeanUtils, Javax.Mail" on the right. Below the header, there is a login form on the left with fields for "Email" and "Password", a "Remember me" checkbox, and a "Sign in" button. To the right of the form, there is a description of the application: "Car rental WebApp. Customer can choose a car for reservation from a list by filling out the order form: passport data, reservation terms. Administrator registers the return of the vehicle. In the case of damage to the vehicle, Administrator makes information and an invoice for the repairs. The administrator may reject the reservation, stating the reason for refusal." Below the form, there is a "Search car" link. At the bottom of the page, there is a yellow footer with the text "© Sergii Shapoval".

* Now the app is deployed and exposed.