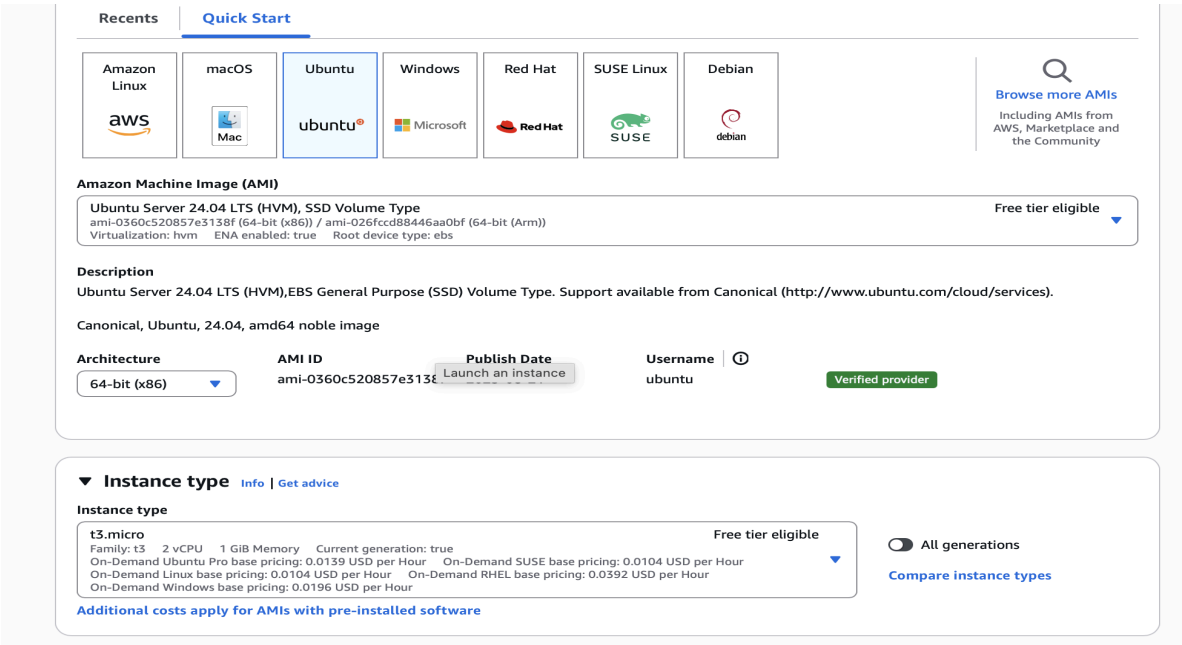
Java EE Application Deployment Guide: Build Server to Deploy Server Workflow

1. Project Context and Goal

This guide details the process of creating, building, and deploying the classic JBoss/WildFly **"Member Registration"** quickstart application (the one shown in your screenshot) using a two-basic server model with minimal instance setup: a dedicated **Build Server** and a dedicated **Deployment Server**.

Server	Role	Key Components
Build Server	Compiles source code, runs tests, and produces the deployable artifact (.war).	JDK, Apache Maven, Git
Deploy Server	Hosts the application server (WildFly/JBoss) and runs the application.	JDK, WildFly/JBoss Application Server

Basic Ubuntu server with a t3.micro instance type



This is the final instance setup

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP	IPv6 IPs	Mo
build-server	i-0c19e23cd0302f46d	Running	t3.micro	3/3 checks passed	View alarms +	us-east-1a	ec2-54-159-91-225.co...	54.159.91.225	–	–	dis
deploy-server	i-077ad8c590dbfc8d8	Running	t3.micro	3/3 checks passed	View alarms +	us-east-1a	ec2-54-90-239-214.co...	54.90.239.214	–	–	dis

2. Prerequisites and Environment Setup

Ensure the following tools are installed on both servers (or your local machine, if simulating servers using different directories/ports).Make sure that before installing the tools update the servers using command

``` sudo apt -y update ```

Prerequisites (Both Servers)

1. **Java Development Kit (JDK 8 or newer):** Must be installed in build server and configured correctly.

```
ubuntu@ip-172-31-20-12:~$ java 8
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-172-31-20-12:~$ sudo apt install openjdk-8-jre-headless
```

2. **Maven:** should also be installed in the build server only once the compatible java version is installed so that maven wen installs configures its settup according
```

```
ubuntu@ip-172-31-20-12:~$ sudo apt install maven
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libaopalliance-java libapache-pom-java libatinject-jsr330-api-java libcdi-api-java libcommo
  liberror-prone-java libgeronimo-annotation-1.3-spec-java libgeronimo-interceptor-3.0-spec-j
```

3. **WildFly or JBoss AS 7/EAP:** Download and extract the application server distribution. For this project, **WildFly 10+** is the modern equivalent of JBoss AS 7.

```
ubuntu@ip-172-31-18-94:~$ wget https://download.jboss.org/wildfly/10.0.0.Final/wildfly-10.0.0.Final.zip
--2025-10-08 07:04:33--  https://download.jboss.org/wildfly/10.0.0.Final/wildfly-10.0.0.Final.zip
Resolving download.jboss.org (download.jboss.org)... 23.53.11.147, 23.53.11.141, 2600:1408:c400:e::17cd:6a11, ...
Connecting to download.jboss.org (download.jboss.org)|23.53.11.147|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 138158431 (132M) [application/zip]
Saving to: 'wildfly-10.0.0.Final.zip'

wildfly-10.0.0.Final.zip          100%[===================================================================================>] 131.76M  79.8MB/s    in 1.7s

2025-10-08 07:04:35 (79.8 MB/s) - 'wildfly-10.0.0.Final.zip' saved [138158431/138158431]

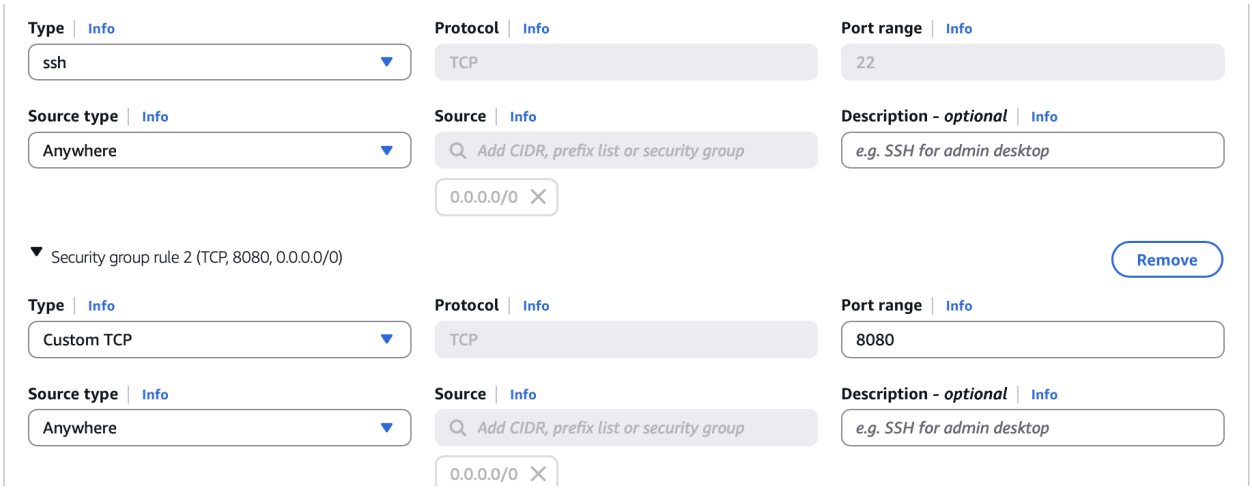
ubuntu@ip-172-31-18-94:~$ ls
wildfly-10.0.0.Final.zip
ubuntu@ip-172-31-18-94:~$
```

```
ubuntu@ip-172-31-18-94:~$ sudo apt install zip
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  unzip
```

```
ubuntu@ip-172-31-18-94:~$ unzip wildfly-10.0.0.Final.zip
Archive:  wildfly-10.0.0.Final.zip
   creating: wildfly-10.0.0.Final/
   creating: wildfly-10.0.0.Final/.installation/
   creating: wildfly-10.0.0.Final/appclient/
   creating: wildfly-10.0.0.Final/appclient/configuration/
   creating: wildfly-10.0.0.Final/bin/
   creating: wildfly-10.0.0.Final/bin/client/
   creating: wildfly-10.0.0.Final/docs/
```

```
ubuntu@ip-172-31-18-94:~$ mv wildfly-10.0.0.Final ~/wildfly-10
ubuntu@ip-172-31-18-94:~$ ls
wildfly-10  wildfly-10.0.0.Final.zip
ubuntu@ip-172-31-18-94:~$
```

4. **Networking:** Ensure both servers can communicate on the necessary ports (typically port 8080 for HTTP access).



Setting up the Servers (Logical Separation)

To simulate the two-server environment on a single machine (if required), follow these steps:

1. **Define Server Directories:**
   - **Build Server Directory:** ~/javaee-build-server/
   - **Deploy Server Directory:** ~/wildfly-deploy-server/ (This is where you extract the WildFly/JBoss zip file).
2. **Run Deploy Server (WildFly):** Start the application server on the Deploy Server. We will use the default standalone.xml configuration.
   # On the Deploy Server machine (or inside the extracted WildFly directory)
   $ cd ~/wildfly-deploy-server/bin
   $ ./standalone.sh

3. Project Creation and Build (On Build Server)

All steps in this section are performed exclusively on the **Build Server**.

Step 3.1: Get the Project Source Code

The project is based on the famous JBoss "kitchensink" quickstart. It can be initialized via Maven or cloned directly. We will clone a common repository structure.
# 1. Navigate to the Build Server directory
$ cd ~/javaee-build-server/

2. Clone the representative quickstart project
$git clone
https://github.com/jboss-developer/jboss-as-quickstarts.git$ cd jboss-as-quickstarts/kitchensink

Step 3.2: Configure and Inspect the Project

The project is a standard Java EE 6 application that uses JPA (for data), Bean Validation (for constraints), JAX-RS (for a REST API), and JSF/Facelets (for the web UI).

- **Key Files:**
  - src/main/java/org/jboss/as/quickstarts/kitchensink/model/Member.java: Contains the JPA entity with Bean Validation annotations (e.g., @NotNull, @Size, @Email).
  - src/main/resources/import.sql: Used by JPA (Hibernate) to populate initial data, including the "John Smith" record visible in the screenshot.
  - src/main/webapp/index.xhtml: The JSF/Facelets view containing the registration form and member list.

Step 3.3: Build the Artifact (.war file)

Use Apache Maven to compile the source code, run any unit tests, and package the final deployable artifact.
# On the Build Server
$ mvn clean package

Output:

```
ubuntu@ip-172-31-20-12:~/javaee-project$ mvn package
[INFO] Scanning for projects...
Downloading from central: https://repo.maven.apache.org/maven2/org/jboss/spec/jboss-javaee-web-6.0/2.0.0.Final/jboss-javaee-web-6.0-2.0.0.Final.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/jboss/spec/jboss-javaee-web-6.0/2.0.0.Final/jboss-javaee-web-6.0-2.0.0.Final.pom (5.5 kB at 13 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/jboss/spec/jboss-specs-parent/1.0.0.Beta2/jboss-specs-parent-1.0.0.Beta2.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/jboss/spec/jboss-specs-parent/1.0.0.Beta2/jboss-specs-parent-1.0.0.Beta2.pom (3.3 kB at 53 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/jboss/jboss-parent/5/jboss-parent-5.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/jboss/jboss-parent/5/jboss-parent-5.pom (19 kB at 280 kB/s)
Downloading from jboss-public-repository: https://repository.jboss.org/nexus/content/groups/public/org/jboss/spec/jboss-javaee6-specs-bom/2.0.0.Final/jboss-javaee6-specs-bom-2.0.
0.Final.pom
Downloaded from jboss-public-repository: https://repository.jboss.org/nexus/content/groups/public/org/jboss/spec/jboss-javaee6-specs-bom/2.0.0.Final/jboss-javaee6-specs-bom-2.0.0
.Final.pom (8.4 kB at 89 kB/s)
[INFO] ------------------------------------------------------------------------
[INFO] Reactor Build Order:
[INFO]
[INFO] javaee-project application                                         [pom]
```

```
[INFO] Installing /home/ubuntu/javaee-project/javaee-project-ear/pom.xml to /home/ubuntu/.m2/repository/com/mews/javaee-project-ear/1.0/javaee-project-ear-1.0.pom
[INFO] ------------------------------------------------------------------------
[INFO] Reactor Summary for javaee-project application 1.0:
[INFO]
[INFO] javaee-project application ......................... SUCCESS [  1.731 s]
[INFO] javaee-project EJB module ......................... SUCCESS [  2.250 s]
[INFO] javaee-project Web module ......................... SUCCESS [  1.196 s]
[INFO] javaee-project EAR module ......................... SUCCESS [  0.262 s]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  5.819 s
[INFO] Finished at: 2025-10-08T06:52:50Z
[INFO] ------------------------------------------------------------------------
ubuntu@ip-172-31-20-12:~/javaee-project$ ls
README  README.md  javaee-project-ear  javaee-project-ejb  javaee-project-web  pom.xml
ubuntu@ip-172-31-20-12:~/javaee-project$ cd javaee-project-ear/
ubuntu@ip-172-31-20-12:~/javaee-project/javaee-project-ear$ ls
pom.xml  target
ubuntu@ip-172-31-20-12:~/javaee-project/javaee-project-ear$ cd target/
ubuntu@ip-172-31-20-12:~/javaee-project/javaee-project-ear/target$ ls
application.xml  javaee-project  javaee-project.ear  maven-archiver
ubuntu@ip-172-31-20-12:~/javaee-project/javaee-project-ear/target$
```

4. Transfer and Deployment (To Deploy Server)

This is the critical step where the built artifact is moved from the Build Server to the Deploy Server.

Step 4.1: Transfer the Artifact

We will use the **Secure Copy Protocol (scp)** as a standard, secure way to transfer the file across a network. This command needs to be executed on the **Build Server**.

Assumptions:

- **Deploy Server IP:** 192.168.1.100 (Replace with the actual IP address of your Deploy Server).
- **Deploy Server User:** deploy_user
- **Deployment Directory Path:** The folder that WildFly/JBoss constantly monitors for new application files is standalone/deployments/.
- **On the Build Server**

First get your pem file to build server from local machine

```
ubuntu@ip-172-31-20-12:~/javaee-project/javaee-project-ear/target$ exit
logout
Connection to ec2-54-159-91-225.compute-1.amazonaws.com closed.
➜  Downloads scp -i key.pem key.pem ubuntu@54.159.91.225:~
The authenticity of host '54.159.91.225 (54.159.91.225)' can't be established.
ED25519 key fingerprint is SHA256:KOFpNGWVZe8YWUnDFAaK36pd+VrAnpq3mw3THrnvveA.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:207: ec2-54-159-91-225.compute-1.amazonaws.com
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '54.159.91.225' (ED25519) to the list of known hosts.
key.pem                                                                                                   100% 1674     4.7KB/s   00:00
➜  Downloads ssh -i "key.pem" ubuntu@ec2-54-159-91-225.compute-1.amazonaws.com
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-1011-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Wed Oct  8 06:59:10 UTC 2025

  System load:  0.0               Temperature:           -273.1 C
  Usage of /:   37.4% of 6.71GB   Processes:             113
  Memory usage: 26%               Users logged in:       0
  Swap usage:   0%                IPv4 address for ens5: 172.31.20.12


Expanded Security Maintenance for Applications is not enabled.

47 updates can be applied immediately.
35 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


Last login: Wed Oct  8 06:39:01 2025 from 104.28.233.87
ubuntu@ip-172-31-20-12:~$ ls
javaee-project  key.pem
ubuntu@ip-172-31-20-12:~$
```

Than transfer the build file from the build server to the deploy server using scp -i

```
ubuntu@ip-172-31-20-12:~$ scp -i key.pem  /home/ubuntu/javaee-project/javaee-project-ear/target/javaee-project.ear ubuntu@54.90.239.214:~/wildfly-10/standalone/deployments/
The authenticity of host '54.90.239.214 (54.90.239.214)' can't be established.
ED25519 key fingerprint is SHA256:Ch3UJZMXnkfAXsG0Q9sVuRDyrJ/g0kCmxWmK6BZVjo4.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '54.90.239.214' (ED25519) to the list of known hosts.
javaee-project.ear                                                                                        100%   60KB  31.5MB/s   00:00
ubuntu@ip-172-31-20-12:~$ Connection to ec2-54-159-91-225.compute-1.amazonaws.com closed by remote host.
Connection to ec2-54-159-91-225.compute-1.amazonaws.com closed.
```

Step 4.2: Automatic Deployment on Deploy Server

As soon as the kitchensink.war file is successfully placed into the standalone/deployments/ folder on the Deploy Server, **WildFly/JBoss detects the new file and automatically attempts to deploy it.**
In the WildFly console (the terminal where you ran ./standalone.sh), you will see log messages confirming the deployment:
[org.jboss.as.server] (ServerService Thread Pool -- 29) WFLYSRV0010: Deployed "kitchensink.war" (runtime-name : "kitchensink.war")

If the deployment is successful, WildFly creates a marker file named kitchensink.war.deployed.

5. Verification

To verify that the deployment was successful and matches your screenshot, open a web browser and navigate to the Deploy Server's IP address and the application's context root.

1. **Access URL:**
   http://<DEPLOY_SERVER_IP>:8080/kitchensink/

   - (Replace <DEPLOY_SERVER_IP> with localhost if running on the same machine, or the actual IP address otherwise).
2. Expected Output:
   You will see the Member Registration page, which includes the section:
   - **Register (Bean Validation example)**
   - A form to register a new member.
   - A list displaying the initial data from import.sql (e.g., **John Smith**).

The successful display of this page confirms that:

- The Build Server successfully compiled the artifact.
- The artifact was correctly transferred and deployed to the Deploy Server.
- All Java EE components (CDI, JPA, Bean Validation, JAX-RS) are running correctly.

Private

Not Secure — 54.90.239.214

Instances | EC2 | us-east-1

Java EE 6 Starter Application

JBoss Application Server 7

Welcome to JBoss AS 7!

You have successfully deployed a Java EE 6 web application on JBoss AS 7.

Register (Bean Validation example)

Enforces annotation-based constraints defined on the model class.

Name:
Email:
Phone #:

Register

Members

Id	Name	Email	Phone #	REST URL
0	John Smith	john.smith@mailinator.com	2125551212	/rest/members/0

REST URL for all members: /rest/members

Find out more

Learn about JBoss AS 7.

- JBoss AS 7 Getting Started Developing Applications Guide
- JBoss AS 7 project site

Learn about the Java EE 6 platform and the component model it provides.

- Java EE 6 tutorial
- JSR-299: CDI specification
- CDI Source

Dive into Weld, the CDI reference implementation, and discover portable extensions Seam 3 offers.

- Weld reference guide
- Weld project
- Seam 3 project
- User forums
- Mailing lists
- Archetype issue tracker

Explore JavaServer Faces, the component-oriented UI framework in Java EE 6.

- JSF community site

If you have an add-on, please let us know and consider contributing it back to the community!

JBoss Application Server 7

This project was generated from a Maven archetype from JBoss.