

## 1. Glassfish server installation and running java servlet application:

I. Checking if java is installed in the system:

**# java -version**

ii. If java is not installed we can install it by running the command :

**#sudo apt-get install openjdk-8-jre**

```
manish@manish-pc:~$ java -version
openjdk version "1.8.0_292"
OpenJDK Runtime Environment (build 1.8.0_292-8u292-b10-0ubuntu1~20.04-b10)
OpenJDK 64-Bit Server VM (build 25.292-b10, mixed mode)
manish@manish-pc:~$
```

iii. We need to locate java's installation directory, export a **JAVA\_HOME** variable from the ~/.profile file and append its path to bin directory to our PATH variable in ~/.profile file.

```
export FISH="/opt/glassfish5/bin"
export GO_PATH="/usr/local/go/bin"
export JAVA_HOME="/usr/lib/jvm/java-1.8.0-openjdk-amd64"
export CHROME_DRIVER="/home/manish/Documents/apps"
export PATH="$FISH:$PATH:$FLUTTER_PATH:$JAVA_HOME/bin:$ADB_PATH:$CHROME_DRIVER:$GO_PATH"
```

We can apply the changes running :

**# source ~/.profile**

iv. Now we can download and install java glassfish server:

The java glassfish server 4.4.1 zip file was downloaded from:

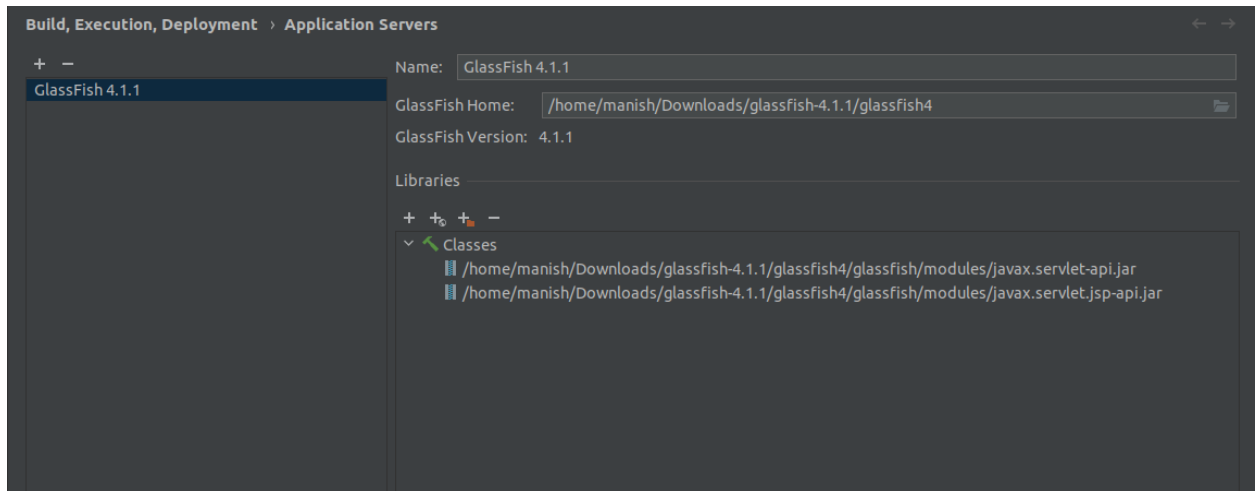
<https://download.oracle.com/glassfish/4.1.1/release/glassfish-4.1.1.zip>

The glassfish-4.1.1.zip file was extracted using the **unzip** command:

**# unzip glassfish-4.1.1.**

V. Created a new java 8 ee web application with maven using IntelliJ idea

Vi. In IntelliJ's application server settings, added glassfish's extracted path.



Vii. Edited the pom.xml to set packaging to war  
# <packaging> war</packaging>

```
<groupId>com.example</groupId>
<artifactId>demo</artifactId>
<version>1.0-SNAPSHOT</version>
<name>demo</name>
<packaging>war</packaging>
<properties>
```

Viii. Edited the domain.xml file and edited the http server port from 8080 to 8088.  
The config file is present in :  
**glassfish-4.1.1/glassfish4/glassfish/domains/domain1/config/**

```
manish@manish-pc: ~/Downloads/glassfish-4.1.1/glassfish4/...  
    <ssl classname="com.sun.enterprise.security.ssl.GlassfishSSLImpl" ce  
rt-nickname="s1as"></ssl>  
  </protocol>  
  <protocol name="admin-listener">  
    <http encoded-slash-enabled="true" max-connections="250" default-vir  
tual-server="__asadmin">  
      <file-cache></file-cache>  
    </http>  
  </protocol>  
</protocols>  
<network-listeners>  
  <network-listener protocol="http-listener-1" port="8088" name="http-li  
stener-1" thread-pool="http-thread-pool" transport="tcp"></network-listener>  
  <network-listener protocol="http-listener-2" port="8181" name="http-li  
stener-2" thread-pool="http-thread-pool" transport="tcp"></network-listener>  
  <network-listener protocol="admin-listener" port="4848" name="admin-li  
stener" thread-pool="admin-thread-pool" transport="tcp"></network-listener>  
</network-listeners>  
<transports>  
  <transport name="tcp"></transport>  
</transports>  
</network-config>  
<thread-pools>  
:X
```

ix. opened the admin page of the glassfish server and uploaded the demo war file generated in the source project folder using maven's war plugin.

```
[INFO] Building demo 1.0-SNAPSHOT  
[INFO] -----[ war ]-----  
[INFO]  
[INFO] --- maven-war-plugin:3.3.1:war (default-cli) @ demo ---  
[INFO] Packaging webapp  
[INFO] Assembling webapp [demo] in [/home/manish/IdeaProjects/demo/target/demo-1.0-SNAPSHOT]  
[INFO] Processing war project  
[INFO] Copying webapp resources [/home/manish/IdeaProjects/demo/src/main/webapp]  
[INFO] Building war: /home/manish/IdeaProjects/demo/target/demo-1.0-SNAPSHOT.war  
[INFO] -----  
[INFO] BUILD SUCCESS  
[INFO] -----  
[INFO] Total time: 1.132 s  
[INFO] Finished at: 2021-11-12T21:27:35+05:45  
[INFO] -----  
Process finished with exit code 0
```

Deploy Applications or M x

localhost:4848/common/index.jsf

profiles icon gh-act iPh json comp qt kube npuc s regex cour kube gre ui inern vim store app Other Bookmarks

Home About... Help

User: admin Domain: domain1 Server: localhost

GlassFish™ Server Open Source Edition

Tree

Common Tasks

Domain

server (Admin Server)

Clusters

Standalone Instances

Nodes

Applications

Lifecycle Modules

Monitoring Data

Resources

Concurrent Resources

Connectors

JDBC

JMS Resources

JNDI

JavaMail Sessions

Resource Adapter Configs

Configurations

default-config

server-config

Update Tool

Deploy Applications or Modules

OK Cancel

Specify the location of the application or module to deploy. An application can be in a packaged file or specified as a directory.

\* Indicates required field

Location: ☒ Packaged File to Be Uploaded to the Server

demo-1.0-SNAPSHOT.war

☐ Local Packaged File or Directory That Is Accessible from GlassFish Server

Type:

Context Root:

Path relative to server's base URL.

Application Name:

Virtual Servers:

Associates an Internet domain name with a physical server.

Status: ☒ Enabled

Allows users to access the application.

Implicit CDI: ☒ Enabled

Implicit discovery of CDI beans

