

Glassfish:

- Install Glassfish server and change HTTP port to 8088.
- Create a demo Java (11) servlet application with maven.
- Generate war packages.
- Deploy the war using glassfish app server.

Glassfish server is installed in Ubuntu 20.04, also default jdk package is installed which is default java development package.

'Sudo apt update'

sudo apt install default-jdk

Then an environment variable is set as:

nano /etc/environment

And this part is added to the file :

JAVA_HOME="/usr/lib/jvm/java-11-openjdk-amd64"

Source /etc/environment to load the variable.

Then latest Glassfish.zip file is downloaded using command:

'wget <http://download.oracle.com/glassfish/5.0.1/nightly/latest-glassfish.zip>'

The file is then extracted

Sudo unzip latest-glassfish.zip

The glassfish.service file is created:

Sudo nano /etc/systemd/system/glassfish.service

Inside that file, this part is appended:

```
GNU nano 4.8 /etc/systemd/system/glassfish.service
[Unit]
Description = GlassFish Server v5.0
After = syslog.target network.target

[Service]
ExecStart=/opt/glassfish5/bin/asadmin start-domain domain1
ExecReload=/opt/glassfish5/bin/asadmin restart-domain domain1
ExecStop=/opt/glassfish5/bin/asadmin stop-domain domain1
Type = forking
TimeoutSec=180

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

Then the daemon was reloaded and following are commands to start glassfish:

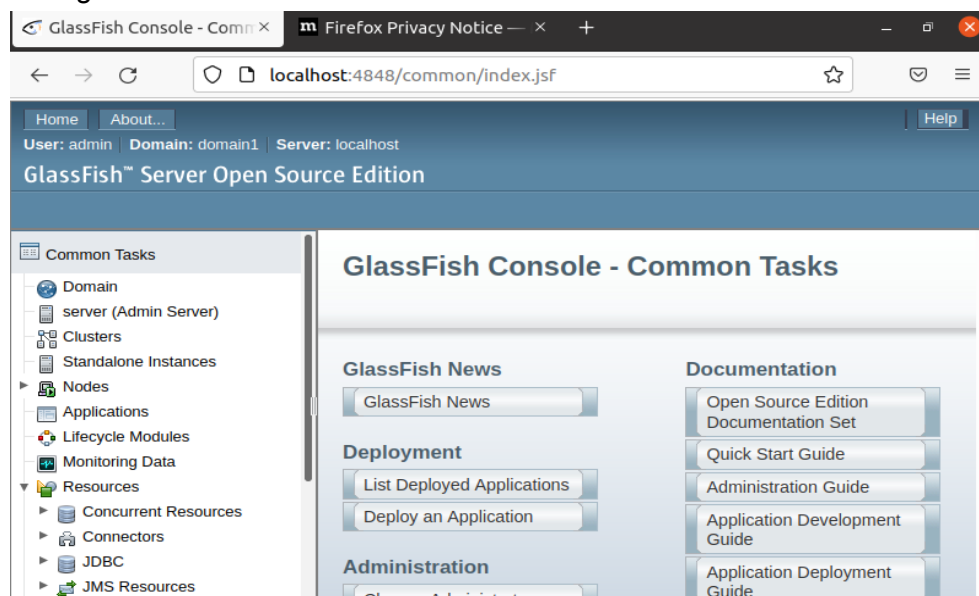
```
sudo systemctl daemon-reload  
sudo systemctl enable glassfish  
sudo systemctl start glassfish  
sudo systemctl status glassfish
```

After this, we can see the status of glassfish:

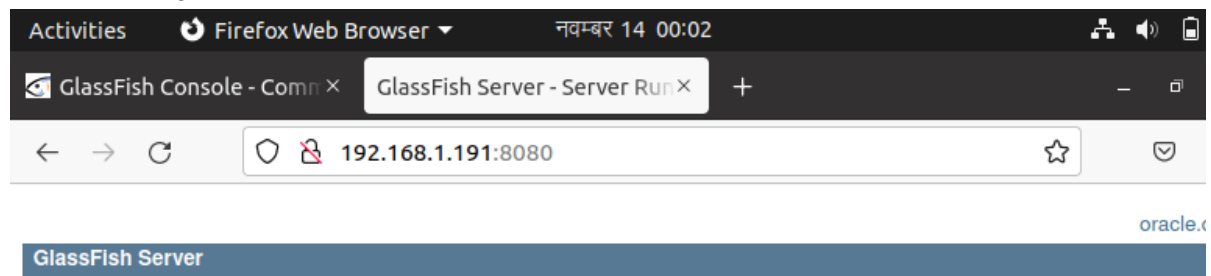
```
~  
lines 1-11/11 (END)  
^C  
root@bjay:/opt# systemctl restart glassfish  
root@bjay:/opt# systemctl status glassfish  
● glassfish.service - GlassFish Server v5.0  
   Loaded: loaded (/etc/systemd/system/glassfish.service; enabled; vendor preset: en  
   Active: active (running) since Sat 2021-11-13 23:04:33 +0545; 5s ago  
   Process: 2491 ExecStart=/opt/glassfish5/bin/asadmin start-domain domain1 (code=exi  
   Main PID: 2503 (java)  
   Tasks: 91 (limit: 2299)  
   Memory: 178.8M  
   CGroup: /system.slice/glassfish.service  
           └─2503 /usr/lib/jvm/java-8-openjdk-amd64/bin/java -cp /opt/glassfish5/gla  
  
नवम्बर 13 23:04:26 bjay systemd[1]: Starting GlassFish Server v5.0...  
नवम्बर 13 23:04:32 bjay asadmin[2491]: Waiting for domain1 to start .....  
नवम्बर 13 23:04:32 bjay asadmin[2491]: Successfully started the domain : domain1  
नवम्बर 13 23:04:32 bjay asadmin[2491]: domain Location: /opt/glassfish5/glassfish/doma  
नवम्बर 13 23:04:32 bjay asadmin[2491]: Log File: /opt/glassfish5/glassfish/domains/doma  
नवम्बर 13 23:04:32 bjay asadmin[2491]: Admin Port: 4848  
नवम्बर 13 23:04:32 bjay asadmin[2491]: Command start-domain executed successfully.  
नवम्बर 13 23:04:33 bjay systemd[1]: Started GlassFish Server v5.0.  
lines 1-18/18 (END)
```

After the glassfish server is active, we can access the admin panel with port 4848.

If we go to a browser and enter localhost:4848 then we can see this screen:



To go to the launch the app, we can enter the IP of the VM.
I.e. 192.168.1.191 in our case and default http port for glassfish service is 8080.
So when we go to 192.168.1.191:8080 we can launch the app,

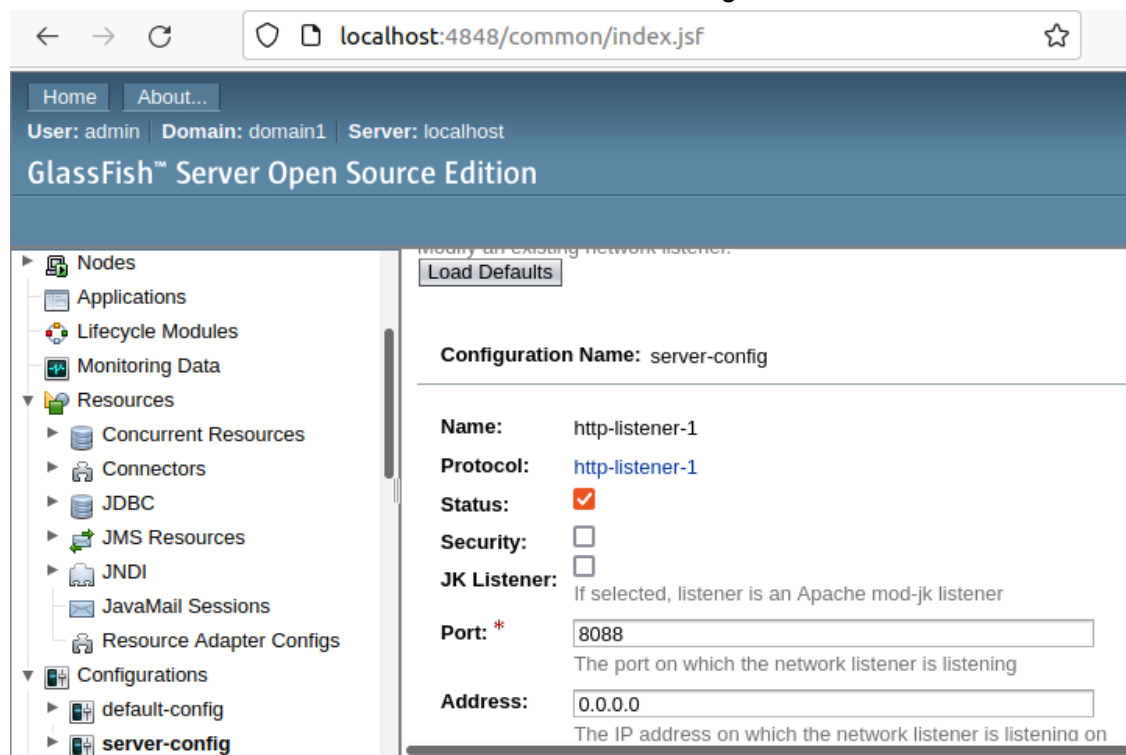


Your server is now running

To replace this page, overwrite the file `index.html` in the document root folder of this server. The document root folder for this server is the `docroot` subdirectory of this server's domain directory.

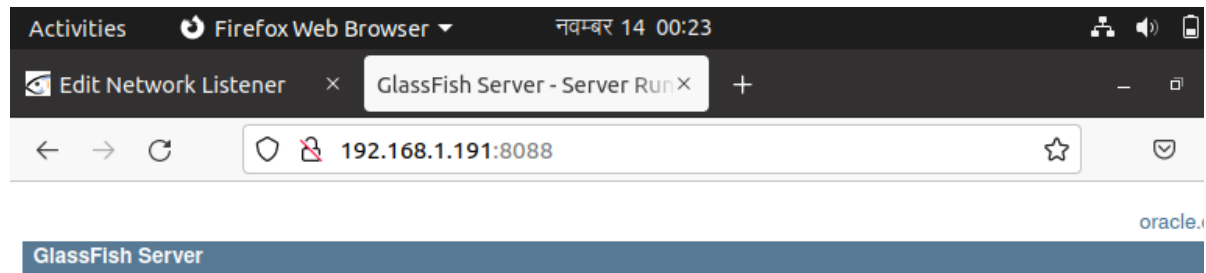
To manage a server on the **local host** with the **default administration port**, go to the [Administration Console](#).

The default HTTP port of the glassfish server is: 8080
And the http port is changed from the admin server.
In the left side of admin server, we can see server config:



Inside server-config >NetworkListeners>http-listener-1
We changed the port from 8080 to 8088 and saved the settings.

Now we can access the app from port 8088.



After the server is running, maven is installed in our system using command:

Sudo apt install maven

A maven package is downloaded and extracted in /opt drive.

wget

https://www.apache.org/dist/maven/maven-3/3.6.3/binaries/apache-maven-3.6.3-bin.tar.gz
-P /tmp

sudo tar xf /tmp/apache-maven-*.tar.gz -C /opt

sudo ln -s /opt/apache-maven-3.6.3 /opt/maven

Then maven.sh file is created

Sudo nano /etc/profile.d/maven.sh

And inside this file, this content is appended:

export JAVA_HOME=/usr/lib/jvm/openjdk-11-amd64

export M2_HOME=/opt/maven

export MAVEN_HOME=/opt/maven

export PATH=\${M2_HOME}/bin:\${PATH}

After that that maven.sh file is made executable and environment variables are loaded using source command as:

sudo chmod +x /etc/profile.d/maven.sh

source /etc/profile.d/maven.sh #environment-variables are loaded

Now maven is initialized in our system.

Now maven is used to generate a sample war file using commands:

Mvn archetype:generate

Then go to dir containing pom.xml file and enter

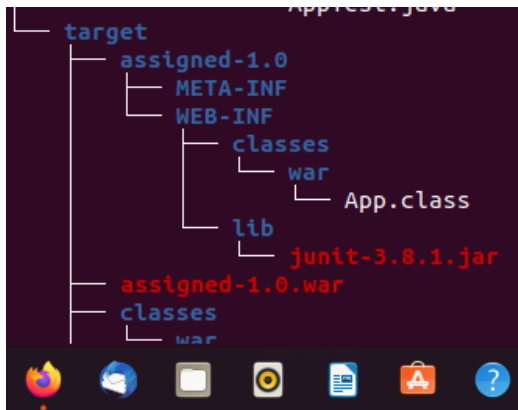
Mvn validate

Mvn compile

Mvn package

Mvn clean install

Then a war file is created in the target directory.



This war file is then selected from admin server page and launched and deployed.

