

Using ElasticBean Stalk, Installed Glassfish Docker on Amazon AWS Linux GF instance wasn't able to access it via admin console page on port 4848 to upload war file.

Editing/configuring glassfish was difficult as it needed to be configured via the Docker commands. Tried configuring Docker glassfish for a week but remote admin console for glassfish was not accessible.

So switched directly to manual setup of Operating system on EC2 on which everything can be configured as our local system.

Only the respective ports needed for application are required to be opened up for providing worldwide access. The respective ports can be opened via security groups inbound rule.

Installing Amazon Linux on EC2

1. Launch instance using EC2 on AWS select all default. For OS choose AMI Linux

Configuring Putty

1. After instance is launched connect to instance using SSH.
 - 1.1 Steps for connecting via SSH
 - 1.2 Create a key pair in AWS
 - 1.3 Link that key pair with instance in AWS
 - 1.4 On Linking download the file private_key.pem from AWS
 - 1.5 Use private_key.pem in putty to authenticate yourself to AWS instance
 - 1.6 Set User: ec2-user
 - 1.7 Set hostname: : ec2-13-235-115-132.ap-south-1.compute.amazonaws.com

Configuring WinSCP

1. For file transfer use WinSCP as root for read/write/execute access to all directories
 - 1.1 Set hostname: ec2-13-235-115-132.ap-south-1.compute.amazonaws.com
 - 1.2 Set protocol :SFTP
 - 1.3 Find SFTP server location in putty console via
 - 1.3.1 *root cmd> cat /etc/ssh/sshd_config |grep -i sftp-server*
output: Subsystem sftp /usr/libexec/openssh/sftp-server
 - 1.4 Set SFTP server: sudo su -c /usr/libexec/openssh/sftp-server
 - 1.5 Set private key file: private_key.pem
 - 1.6 Set shell:sudo -su

Installing and Configuring JAVA

Download jdk-8u241-linux-x64.rpm on Windows

Move it to location home/ec2-user using WinScp

Install downloaded JDK

ec2 user cmd> sudo rpm -i jdk-8u241-linux-x64.rpm

Add the manual installed java to the alternatives list

ec2 user cmd> sudo /usr/sbin/alternatives --install /usr/bin/java java /usr/java/jdk1.8.0_241-amd64/bin/java 3000

SET JAVA Default version

ec2 user cmd>sudo /usr/sbin/alternatives --config java

Installing Glassfish

Download glassfish full platform glassfish-4.1.2.zip on Windows

Move to /usr/share/ via WinSCP

GOTO /usr/share location and unzip the glassfish zip file

ec2 user cmd>sudo unzip -q glassfish-4.1.2.zip

Setting up JDBC Driver

For JDBC Connections MYSQL driver is required

Copy mysql-connector-java-8.0.19.jar at location ../../usr/share/glassfish5/glassfish/lib/ via WinSCP

Adding users for Glassfish Server

Move to directory `usr/share/` and in putty command execute below commands

```
Cmd>sudo su
```

Add Group

```
Cmd>groupadd groupgf
```

Assign bash login shell to user `usergf`

```
Cmd>useradd -s /bin/bash -g groupgf usergf
```

Change user ownership of `glassfish5` and group ownership to root

```
Cmd>chown -R usergf: glassfish5
```

Change the password for the 'glassfish' user:

```
Cmd>passwd usergf
```

Adding glassfish as a service(OPTIONAL SKIP IF NOT REQUIRED)

Create a start/stop/restart script:

```
nano /etc/init.d/glassfish
```

Add the below lines in glassfish file

```
#!/usr/bin/env bash

# description: Glassfish start/stop/restart

# processname: glassfish

# chkconfig: 2445 20 80

JAVA_HOME= ../../usr/java/jdk1.8.0_241-amd64

export JAVA_HOME

PATH=$JAVA_HOME/bin:$PATH

export PATH

GLASSFISH_HOME=../../usr/share/glassfish5/glassfish/

GLASSFISH_USER=ec2-user

case $1 in

start)

    su $GLASSFISH_USER -c "$GLASSFISH_HOME/bin/asadmin start-domain domain1"

;;

stop)

    su $GLASSFISH_USER -c "$GLASSFISH_HOME/bin/asadmin stop-domain domain1"

;;

restart)

    su $GLASSFISH_USER -c "$GLASSFISH_HOME/bin/asadmin stop-domain domain1"

    su $GLASSFISH_USER -c "$GLASSFISH_HOME/bin/asadmin start-domain domain1"

;;

esac

exit 0
```

Save the script and make it executable:

```
Cmd>chmod +x /etc/init.d/glassfish
```

If you want to start up your GlassFish application on boot execute the following commands:

```
chkconfig --add glassfish
```

```
chkconfig glassfish on
```

```
reboot
```

To start/stop/restart Glassfish use the command:

```
# /etc/init.d/glassfish start/stop/restart
```

Configuring Glassfish for Remote access

```
cmd>sudo su
cmd>cd usr/share/glassfish/glassfish/bin/
cmd>asadmin
asadmin>change-master-password --savemasterpassword //default pswd is changeit
asadmin>change-admin-password //default pswd is blank
asadmin>start-domain
asadmin>enable-secure-admin
asadmin>restart-domain
```

Note: Add port 8080 and 4848 in security group inbound rules and set access from anywhere

Configuring JDBC Connection Pool

```
<jdbc-connection-pool driver-classname="com.mysql.jdbc.Driver" name="awspool" description="" res-
type="java.sql.Driver">
```

```
  <property name="password" value="password"></property>
```

```
  <property name="user" value="user"></property>
```

```
  <property name="URL" value="jdbc:mysql://db-1.cvgjpg.ap-south-
1.rds.amazonaws.com:3306/dbname"></property>
```

```
  <property name="allowPublicKeyRetrieval" value="true"></property>
```

```
  <property name="driverClass" value="com.mysql.cj.jdbc.Driver"></property>
```

```
  <property name="useSSL" value="false"></property>
```

```
  <property name="databaseName" value="holidays"></property>
```

```
</jdbc-connection-pool>
```

```
<jdbc-resource pool-name="awspool" jndi-name="awsresource"></jdbc-resource>
```

Deploying JAVA WAR file;

Keep war file at location glassfish5/glassfish using WinSCP

```
CMD> cd ../../usr/share/glassfish5/glassfish/
```

```
CMD> bin/asadmin deploy zipfilename.war
```

For undeploying

```
CMD> bin/asadmin undeploy zipfilename
```

OPTIONAL(With Context Root)

```
CMD>bin/asadmin --port=4848 deploy --contextroot "/WebApplication2" ../../../../home/ec2-user/WebApplication2.war
```

Installing ReactJS

```
CMD>cd usr/share
```

```
CMD> sudo su
```

```
CMD> yum install -y gcc-c++ make
```

```
CMD> curl -sL https://rpm.nodesource.com/setup_13.x | sudo -E bash -
```

```
CMD> yum install nodejs
```

```
CMD> node -v
```

```
CMD> npm -v
```

Creating React Application

```
CMD>npm install -g create-react-app
```

```
CMD>create-react-app todoapp
```

```
CMD>cd todoapp
```

copy todoapp local project code to this folder including package.json

```
CMD>npm install
```

```
CMD>npm install -g serve
```

```
CMD>serve -s build
```

Note: Open port 5000 in AWS security group

For stopping application CTRL+C OR

```
CMD> sudo fuser -k <portno>/tcp
```