# **How To Install Glassfish 4.0 on Ubuntu 12.04.3**

### Pre-conditions

There are many tutorials available to help you install OpenJDK and JBoss. This is one on the latest concerning Oracle Java and Glassfish. Hopefully this will make deploying easier for Java EE developers.

You will need a droplet with Ubuntu 12.04.3 x64 that has been created with DigitalOcean. Login as root by ssh. This article assumes no Java installed and at least 1G memory, as Java EE servers are quite demanding.

**What is Glassfish?**

GlassFish is an open-source application server and the reference implementation of Java EE. GlassFish 4.0 release supports the latest Java Platform: Enterprise Edition 7. It supports Enterprise JavaBeans, JPA, JavaServer Faces, JMS, RMI, JavaServer Pages, servlets, etc.

## Step One: Install Oracle Java 7

Start by updating the package index:

sudo apt-get update

In order to get Oracle Installer of Java 7, we need to add a new apt repository. In order to use add-apt-repository, you need to install python-software-properties. Here's how to do it by apt-get:

sudo apt-get install python-software-properties

Now you can add the new repository and install from Oracle Installer:

sudo add-apt-repository ppa:webupd8team/java

Make source list up-to-date:

sudo apt-get update

Install Java 7 by apt-get:

sudo apt-get install oracle-java7-installer

After installing, confirm the current Java is Oracle version:

java -version

**You will see this:**

java version "1.7.0\_51"  
Java(TM) SE Runtime Environment (build 1.7.0\_51-b13)  
Java HotSpot(TM) 64-Bit Server VM (build 24.51-b03, mixed mode)

## Step Two: Install Glassfish 4.0

**Get Glassfish Zip file**

wget download.java.net/glassfish/4.0/release/glassfish-4.0.zip

Install unzip first before unpackage to /opt

apt-get install unzip

Create the directory /opt and then unzip the package to /opt:

unzip glassfish-4.0.zip -d /opt

For convenience, add export PATH=/opt/glassfish4/bin:$PATH to the end of ~/.profile.

**Start the glassfish server:**

asadmin start-domain

**You will see:**

Waiting for domain1 to start ...................  
Successfully started the domain : domain1  
domain Location: /opt/glassfish4/glassfish/domains/domain1  
Log File: /opt/glassfish4/glassfish/domains/domain1/logs/server.log  
Admin Port: 4848  
Command start-domain executed successfully.

A domain is a set of one or more GlassFish Server instances managed by one administration server. Default GlassFish Server’s port number: 8080. Default administration server’s port number: 4848. Administration user name: admin; password: none.

In order to visit admin page (your*server*id:4848) remotely, you need to enable secure admin:

asadmin enable-secure-admin

**You will see:**

Enter admin user name> admin  
Enter admin password for user "admin">   
You must restart all running servers for the change in secure admin to take effect.  
Command enable-secure-admin executed successfully.

Restart domain to make effect of secure admin:

asadmin restart-domain

**You will see:**

Successfully restarted the domain  
Command restart-domain executed successfully.

Now you can visit admin page (your*server*id:4848) in browser

**To stop the GlassFish server:**

asadmin stop-domain

**You will see:**

Waiting for the domain to stop .  
Command stop-domain executed successfully.

## Demo service: deploy hello.war on Glassfish

Download the sample application from Glassfish official samples:

wget https://glassfish.java.net/downloads/quickstart/hello.war

Deploy war file:

asadmin deploy /home/ee/glassfish/sample/hello.war

**You will see:**

Enter admin user name> admin  
Enter admin password for user "admin">   
Application deployed with name hello.  
Command deploy executed successfully.

Now you can visit your*server*id:8080/hello

To undeploy the application:

asadmin undeploy hello

**You will see:**

Enter admin user name> admin  
Enter admin password for user "admin">   
Command undeploy executed successfully.

In order to save typing "admin user name" and "password" every time you deploy or undeploy an application, create a password file pwdfile with content:

AS\_ADMIN\_PASSWORD=your\_admin\_password

Add --passwordfile in command:

asadmin --passwordfile pwdfile deploy /home/ee/glassfish/sample/hello.war

Now the prompt for user name/password won't appear.

## 

## Virtual Hosting using Apache as proxy to GlassFish

### Introduction

In general when people think about running a Java based application server like Glassfish with Apache as a frontend they almost always think about using mod\_jk. This works fine but can be cumbersome to configure and difficult to debug - especially when you are hosting many different web applications from multiple Glassfish servers. Glassfish is an excellent server in its own right and does not need to have static files served by Apache (the usual mod\_jk approach) so this article explains the mod\_proxy alternative.

### Typical Scenario and Requirements

* Linux based server
* Apache 2 is installed and working
* Glassfish 2.x+ is installed and working
* Root access to configure Apache

We assume that Apache is listening on port 80 and Glassfish has a correctly configured virtual host listening on port 8080. Configuration and setup of these two is beyond the scope of this article!

### Configuring mod\_proxy for Apache

Typical linux installations will have a directory structure under **/etc/apache2/** (have a look at the top of the /etc/apache2/httpd.conf file to see the directory structure in detail). The system configuration subdirectory (sysconfig.d) contains a loadmodule.conf file which lists all the Apache modules loaded on startup. Hopefully you will see something like the following lines:

LoadModule proxy\_module /usr/lib64/apache2-prefork/mod\_proxy.so  
LoadModule proxy\_http\_module /usr/lib64/apache2-prefork/mod\_proxy\_http.so

However if you have a default installation they will be missing. To include the mod\_proxy modules you will need to have root access to modify the /etc/sysconfig/apache2 file. Search for a line that starts with APACHE\_MODULES and looks like:

APACHE\_MODULES="actions alias auth\_basic *...lots more modules...***proxy proxy\_http** "

The **proxy proxy\_http** will be missing so add them as shown making sure you do not break the line.

### Restart Apache

Restart Apache by typing:

/etc/init.d/apache2 stop  
/etc/init.d/apache2 start

Do not use:

apache2ctl stop/start

as with some operating systems the loadmodule.conf may not be updated!

### Test Apache now has mod\_proxy loaded

You can do this by typing the following from the command line:

apache2ctl -M

This will hopefully list your proxy modules:

proxy\_module (shared)  
proxy\_http\_module (shared)

### Add Apache virtual host(s)

This is done by adding a .conf text file to the /etc/apache2/vhosts.d directory. I normally name the file to match the domain you are proxying (eg www.mydomain.com.conf) the contents of which should now contain:

<VirtualHost \*>  
ServerName mydomain.com  
ServerAlias www.mydomain.com  
ServerAdmin   
 info@mydomain.com  
ProxyRequests Off  
<Proxy \*>  
Order deny,allow  
Allow from all  
</Proxy>  
   
ProxyPass / http://localhost:8080/   
ProxyPassReverse / http://localhost:8080/  
</VirtualHost>

This is the minimum required. I recommend everyone has a careful look at the proxy technical pages on the Apache2 website to understand what these directives mean (<http://httpd.apache.org/docs/2.0/mod/mod_proxy.html>).

### Reload the Apache server to pick up your changes

apache2ctl restart

That's it! All requests coming to http://www.mydomain.com will now be proxied to your Glassfish virtual host listening on port 8080.

### JSF forms not rewritten correctly?

This issue arises when the project name is inserted between the domain name and the file name. The result is an invalid target URL when submitting forms (the form->action URL of JSF forms is not rewritten correctly)

The issue can been solved using apache's mod\_proxy\_html module. Using this you can rewrite the links (including form actions) within the rendered html page passed through the reverse proxy. Simply rewrite "/yourprojectname/\*" to "/\*" and everything should works as expected.