

Bot Libre Enterprise Bot Platform Installation Guide

Overview

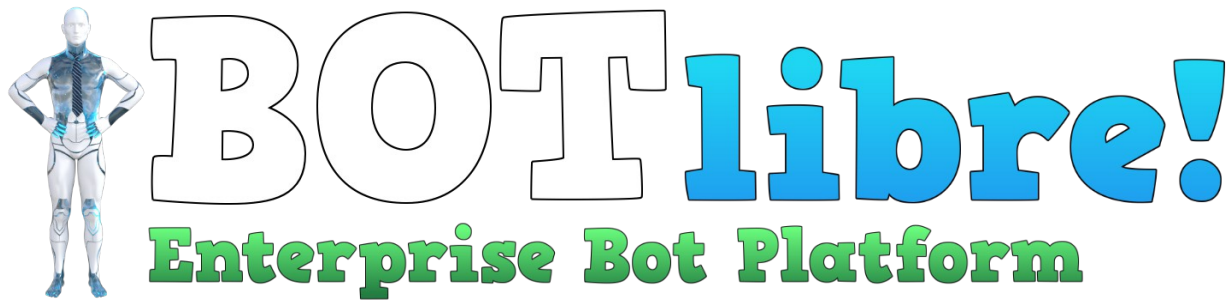
This installation guides give an overview for installing the Bot Libre Enterprise Bot Platform, its web server, database, and configuring the server.

The Bot Libre Enterprise Bot Platform has the following components:

- A Java web application – This runs on a web server and requires the Tomcat Java web server. Tomcat is a free and open source software product. Bot Libre is a Java “war” application that is deployed to the web server. To host Bot Libre a server is required to run the web server.
- A database -- The Bot Libre web application requires a PostgreSQL database. PostgreSQL is a free and open source software product. The database should normally run on the same server as the web server.
- A python web server – This runs in Python using Flask and is used to integrate with deep learning and Tensorflow. This web server is internal and only accessed from Tomcat.
- A web interface -- Bot Libre provides a web (HTML/JavaScript) interface that runs from its web server. This is the users primary interface to create, and administer their content.
- A web API – Bot Libre provides a web API that allows web and mobile clients to access content on its web server.
- An SDK – Bot Libre provides an open source software development kit in JavaScript, Java, Android, and objective C (iOS). This SDK gives access to Bot Libre's services from web and mobile applications.
- An AI engine – Bot Libre uses an artificial intelligence engine to service chat bots requests on its web server. Paphus develops the AI engine under BotLibre.org, and released the product as open source in September 2015.
- Third party libraries – Bot Libre uses several free open source third party libraries such as EclipseLink , Twitter4J, Facebook4J, and MaryTTS.

Hardware

Bot Libre requires a Java web server, Python, and a database to run. To host Bot Libre you require a dedicated server or VM with at least 8gb of RAM.



A single server cannot support an unlimited number of clients. Clients with excessive requests may require a dedicated server. The BotLibre.com server normally processes between 100,000 and 500,000 chat interactions per day.

The recommended production installation is as follows:

- a dedicated physical server
- 32gb ram
- 2 1tb disks
- Linux Centos 7.7
- Tomcat 8.5
- PostgreSQL 9.4
- Java 8

Bot Libre currently uses the OVH ISP

<https://www.ovh.com/ca/en/dedicated-servers/details-servers-range-ENT-id-2016-SP-32.xml>

Development machines and small installations have smaller hardware requirements.

You can run Bot Libre on Windows or Mac for development and testing. Linux CentOS 7.7 is recommended for a server.

Installation

This installation guide is based on Linux Centos 7.7. The installation process may vary depending on the operating system type and version. For installation of the web server and database you can also review the documentation for Apache Tomcat, and PostgreSQL online.

- <http://tomcat.apache.org/>
- <https://www.postgresql.org/>

To install on a development desktop machine, do not follow this process, as it is meant for a Linux server. For Windows or Mac, simply install Tomcat and PostgreSQL using their install wizards from their website.

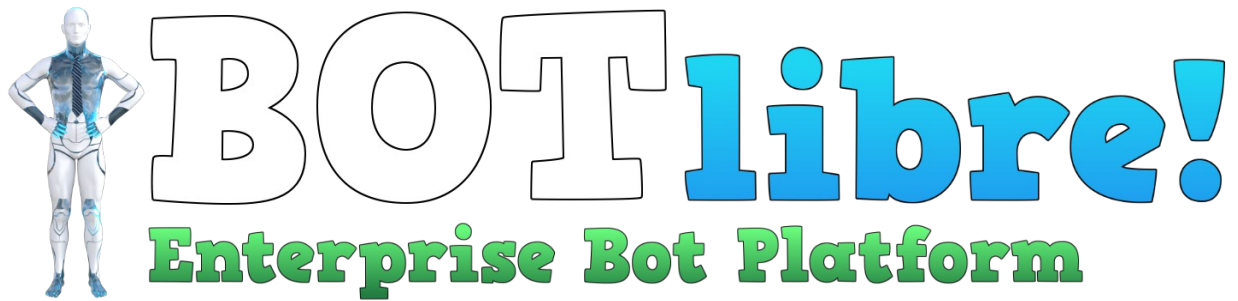
Step 1 – Review

Before you start verify your OS is configured as desired.

Check OS version:

```
> cat /etc/*release*
```

Check disk partitions:



```
> fdisk -l
```

```
> lsblk -io KNAME,TYPE,SIZE,MODEL,MOUNTPOINT
```

** It is recommended to have 2 disks, with tomcat installed on disk 1 and the database installed on disk 2, ensure both partitions have sufficient size (/usr/local/ and /data)

Step 2 – Firewall

Ensure your server has a firewall running and blocks all ports that are not required.

Required ports are:

- 22 – ssh ** blocking this port will prevent you from accessing your server, so be careful
- 80 – http
- 443 – https
- 21 – ftp
- 25 – smtp
- 5901, 5902 - VNC
- 465 - smtp
- 110 – pop3
- 995 - pop3
- 143 - imap
- 993 - imaps
- 6665 - IRC

The following is an example firewall configuration using iptables.

On Centos 7.7 you may need to install and enable iptables or configure systemctl.

```
>>
```

```
systemctl disable firewalld
```

```
yum install iptables-services
```

```
systemctl enable iptables
```

```
service iptables save
```



BOTlibre!

Enterprise Bot Platform

>>

iptables

iptables -L -n

service iptables restart

iptables -P INPUT ACCEPT

iptables -F

iptables -A INPUT -i lo -j ACCEPT

iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

iptables -A INPUT -p icmp --icmp-type echo-request -j ACCEPT

iptables -A INPUT -p tcp --dport 22 -j ACCEPT

iptables -A INPUT -p tcp --dport 5901 -j ACCEPT

iptables -A INPUT -p tcp --dport 5902 -j ACCEPT

iptables -A INPUT -p tcp --dport 80 -j ACCEPT

iptables -A INPUT -p tcp --dport 443 -j ACCEPT

iptables -A INPUT -p tcp --dport 25 -j ACCEPT

iptables -A INPUT -p tcp --dport 465 -j ACCEPT

iptables -A INPUT -p tcp -m tcp --dport 110 -j ACCEPT

iptables -A INPUT -p tcp -m tcp --dport 995 -j ACCEPT

iptables -A INPUT -p tcp -m tcp --dport 143 -j ACCEPT

iptables -A INPUT -p tcp -m tcp --dport 993 -j ACCEPT

iptables -A INPUT -p tcp --dport 21 -j ACCEPT

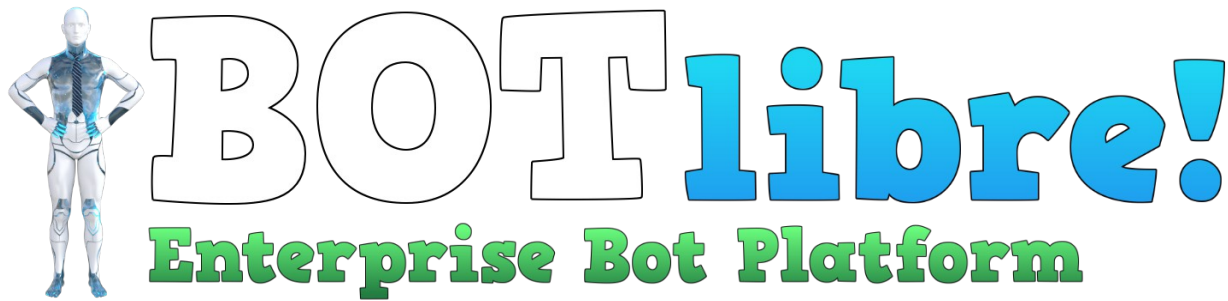
iptables -A INPUT -p tcp --dport 6665 -j ACCEPT

iptables -P INPUT DROP

iptables -P FORWARD ACCEPT

iptables -P OUTPUT ACCEPT

service iptables save



Step 3 – Java

Install the Java JDK 1.8 (latest dot release).

The following is an example shell command to install Java, you may need to change the commands for your OS, or the paths for the latest patch release.

>>

```
cd /opt/
```

```
wget --no-cookies --no-check-certificate --header "Cookie: gpw_e24=http%3A%2F%2Fwww.oracle.com%2F; oraclelicense=accept-securebackup-cookie"
"http://download.oracle.com/otn-pub/java/jdk/8u251-b08/3d5a2bb8f8d4428bbe94aed7ec7ae784/jdk-8u251-linux-x64.tar.gz"
```

```
tar xzf jdk-8u251-linux-x64.tar.gz
```

```
cd /opt/jdk1.8.0_251/
```

```
alternatives --install /usr/bin/java java /opt/jdk1.8.0_251/bin/java 2
```

```
alternatives --config java
```

```
export JAVA_HOME=/opt/jdk1.8.0_251
```

```
export JRE_HOME=/opt/jdk1.8.0_251/jre
```

```
export PATH=$PATH:/opt/jdk1.8.0_251/bin:/opt/jdk1.8.0_251/jre/bin
```

Step 4 – Tomcat

Install the Tomcat 8.5 web server. Tomcat is a Java web server, it should be installed as the server's main webserver and listen on port 80. It is possible to also run the Apache web server on the server, but not recommended. Tomcat 8.5 is recommended.

The following is an example shell command to install Tomcat, you may need to change the commands for your OS, or the paths for the latest patch version.

>>

```
wget https://downloads.apache.org/tomcat/tomcat-8/v8.5.55/bin/apache-tomcat-8.5.55.tar.gz
```

```
tar xzf apache-tomcat-8.5.55.tar.gz
```

```
mv apache-tomcat-8.5.55 /usr/local/tomcat8
```

Once installed you must configure Tomcat.

Give the JVM as much memory as your server can allocate, (%60-%80 ram).

>> (you may wish to install a text editor)



BOTlibre!

Enterprise Bot Platform

```
yum install nano
```

```
>>
```

```
cd /usr/local/tomcat8/bin
```

```
nano setenv.sh
```

```
>> insert these JVM options to increase the max memory size in your setenv.sh
```

```
export JAVA_OPTS="-server -Xmx8g"
```

```
>>
```

```
cd /usr/local/tomcat8/conf
```

```
nano web.xml
```

```
>> insert the following XML in your web.xml to enable CORS filters (required for JavaScript access), enable security filter, and disable iframes except for select pages
```

```
<filter>
  <filter-name>CorsFilter</filter-name>
  <filter-class>org.apache.catalina.filters.CorsFilter</filter-class>
  <init-param>
    <param-name>cors.support.credentials</param-name>
    <param-value>>false</param-value>
  </init-param>
  <init-param>
    <param-name>cors.allowed.origins</param-name>
    <param-value>*</param-value>
  </init-param>
</filter>
<filter-mapping>
  <filter-name>CorsFilter</filter-name>
  <url-pattern>/rest/*</url-pattern>
</filter-mapping>

<filter>
```



BOTlibre!

Enterprise Bot Platform

```
<filter-name>FrameResourceFilter</filter-name>
<filter-class>org.botlibre.web.servlet.FrameResourceFilter</filter-class>
</filter>
<filter-mapping>
  <filter-name>FrameResourceFilter</filter-name>
  <url-pattern>/chat</url-pattern>
</filter-mapping>
<filter-mapping>
  <filter-name>FrameResourceFilter</filter-name>
  <url-pattern>/livechat</url-pattern>
</filter-mapping>
<filter-mapping>
  <filter-name>FrameResourceFilter</filter-name>
  <url-pattern>/forum</url-pattern>
</filter-mapping>
<filter-mapping>
  <filter-name>FrameResourceFilter</filter-name>
  <url-pattern>/forum-post</url-pattern>
</filter-mapping>
<filter-mapping>
  <filter-name>FrameResourceFilter</filter-name>
  <url-pattern>/browse-post</url-pattern>
</filter-mapping>
<filter-mapping>
  <filter-name>FrameResourceFilter</filter-name>
  <url-pattern>/index.jsp</url-pattern>
</filter-mapping>
</filter>
```



BOTlibre!

Enterprise Bot Platform

```
<filter-name>HeaderSecurityFilter</filter-name>
<filter-class>org.apache.catalina.filters.HttpHeaderSecurityFilter</filter-class>
</filter>
```

```
<filter-mapping>
  <filter-name>HeaderSecurityFilter</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
```

```
<error-page>
  <error-code>404</error-code>
  <location>/404.jsp</location>
</error-page>
```

>>

nano server.xml

>> update connector port 8080->80, 8433->433

>> add to connector, URIEncoding="UTF-8"

>> update host- > autoDeploy=false

>> to support https also uncomment the https connector on port 433, you will also need to purchase, register, and generate an https certificate (leave this for later)

Test starting tomcat:

>>

cd /usr/local/tomcat8/bin

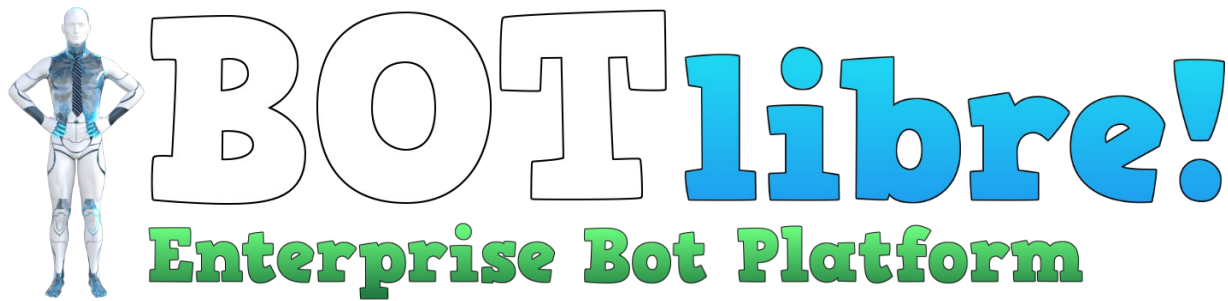
./startup.sh

Verify no errors in /usr/local/tomcat8/logs/catalina.out

Verify you can connect to the webserver, http://<server-ip>/

Stop tomcat:

./shutdown.sh



Troubleshooting: If you had modified the tomcat-users.xml to add a user, remove this configuration as it will make the server fail to start once you delete the tomcat console from webapps.

Step 5 – PostgreSQL

Install the PostgreSQL 9.4 database. It is important to install 9.4, if you install another version ensure you update the PostgreSQL JDBC jar in lib.

It is recommended to store the database on a separate disk on /data.

The following is an example shell command to install PostgreSQL, you may need to change the commands for your OS, or the paths for the latest patch version.

>> first disable yum updates in case yum uses an older version

```
nano /etc/yum.repos.d/CentOS-Base.repo
```

>> add lines to exclude postgresql from yum

```
[base]
```

```
...
```

```
exclude=postgresql*
```

```
[updates]
```

```
...
```

```
exclude=postgresql*
```

>> install postgresql 9.4

```
rpm -Uvh http://yum.postgresql.org/9.4/redhat/rhel-6-x86_64/pgdg-centos94-9.4-3.noarch.rpm
```

```
yum install postgresql94 postgresql94-server postgresql94-contrib
```

Only if you have 2 disks, you may want to use the 2nd disk for the database and change the postgres data directory. (this is not required, you can use the default data directory)

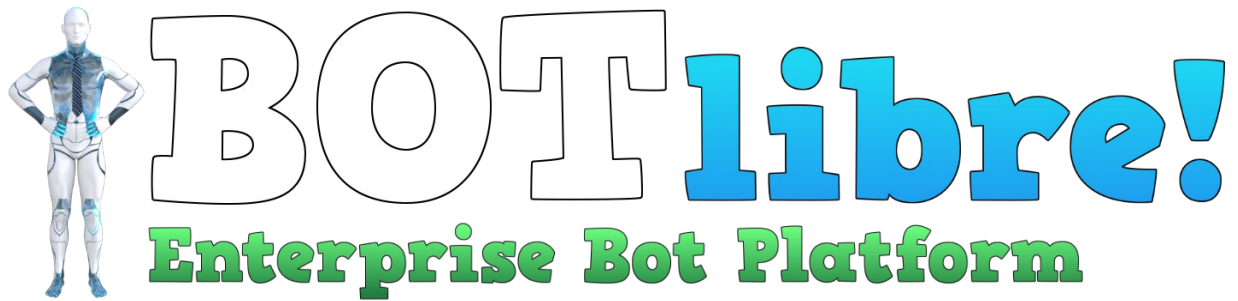
```
nano /usr/lib/systemd/system/postgresql-9.4.service
```

>> change the line "Environment=PGDATA=/var/lib/pgsqli/9.4/data/" to
"Environment=PGDATA=/data/postgres/"

```
cd /data
```

```
mkdir postgres
```

```
chown postgres postgres
```



>> initialize the database

```
su - postgres
```

```
/usr/pgsql-9.4/bin/initdb /data/postgres
```

```
exit
```

>> or “/usr/pgsql-9.4/bin/initdb //var/lib/pgsql/9.4/data” if using the default data directory

Verify database starts/stops:

```
service postgresql-9.4 start
```

```
service postgresql-9.4 stop
```

Configure restart:

```
chkconfig --levels 235 postgresql-9.4 on
```

Set database password:

```
service postgresql-9.4 start
```

```
sudo -u postgres psql postgres
```

```
\# \password postgres
```

```
\q
```

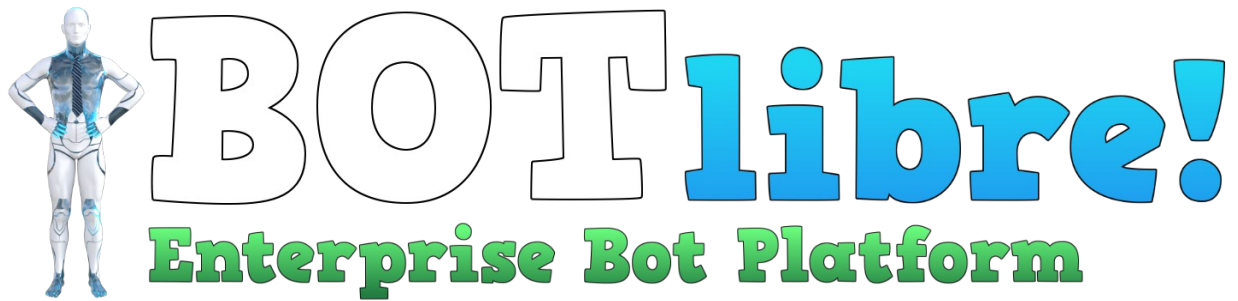
Test the password works using:

```
psql -U postgres -W
```

Troubleshoot password: (if you cannot connect to the database, you may need additional configuration)

```
nano /var/lib/pgsql/9.4/data/pg_hba.conf
```

>> change “ident”/”peer” to “trust” for all



Step 6 – Python

Bot Libre uses a separate Python web server to support deep learning analytics. Python is not required for bots or NLP, only deep learning analytics such as image classification. If you do not require deep learning you can skip this step.

There are several installation steps:

- Update packages
- Install Python
- Install Pip
- Install Tensorflow
- Install Flask
- Install Psycopg2

```
yum update
```

```
yum install yum-utils
```

```
yum groupinstall development
```

```
yum -y install https://repo.ius.io/ius-release-el7.rpm
```

```
yum -y install python36u
```

```
yum -y install python36u-pip
```

```
yum -y install python36u-devel
```

```
pip3.6 install "numpy<1.17"
```

```
pip3.6 install "tensorflow<1.8"
```

```
pip3.6 install flask
```

```
pip3.6 install psycopg2-binary
```

```
>> for object detection
```

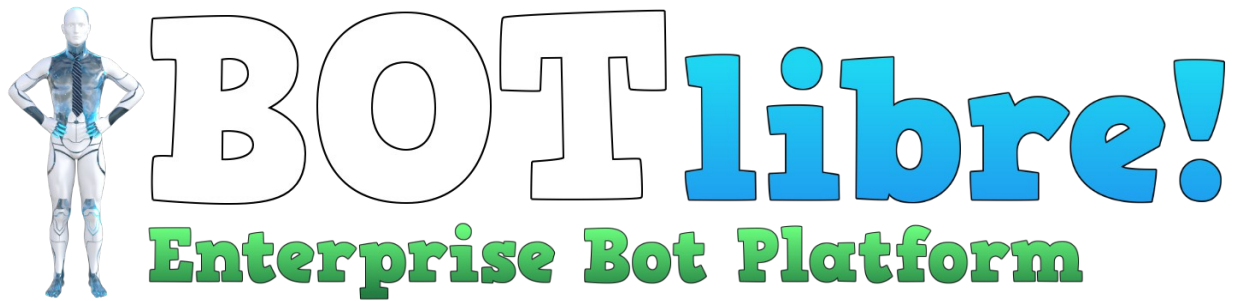
```
pip3.6 install pillow
```

```
pip3.6 install lxml
```

```
pip3.6 install Cython
```

```
pip3.6 install jupyter
```

```
pip3.6 install matplotlib
```



```
pip3.6 install pandas
```

```
pip3.6 install opencv-python
```

To test python run the following:

```
python3.6 -V
```

To test Tensorflow run the following:

```
python3.6 -c 'import tensorflow as tf; print(tf.__version__)'
```

Copy the python directory from the Bot Libre install to /usr/local/python.

To run the server type,

```
nohup python3.6 app.py &
```

To test the server run the follow to access the Flask website, check index.html,

```
wget http://localhost:6777/
```

Troubleshooting: If you cannot access the server check your nohup file for errors.

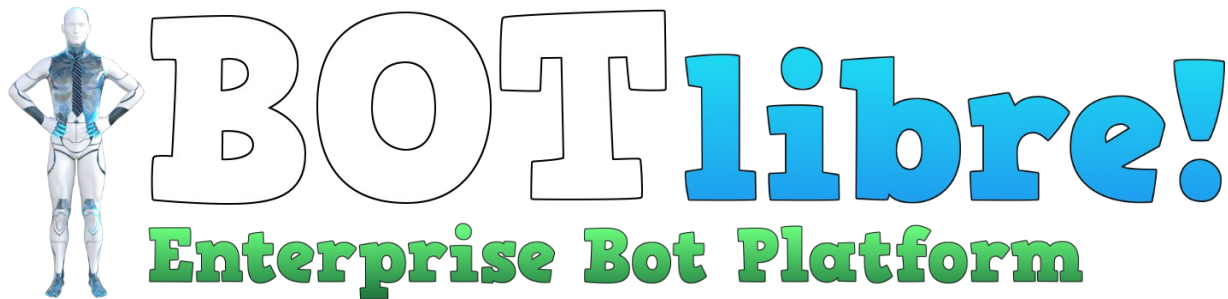
If there are no errors, you may need to change the last line of the app.py file to,

```
app.run(host='0.0.0.0', port=6777, debug=False, threaded=True)
```

Step 7 – Bot Libre

The Bot Libre application is a Java “war” file. A war file is a zip file that contains a website and Java libraries.

There are several installation steps:

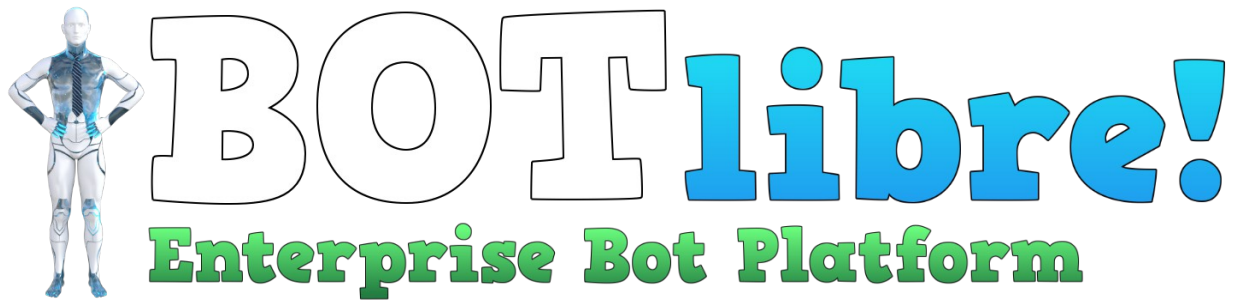


- Copy library jars to /usr/local/tomcat8/lib (from bot libre lib sub directories, **do not copy** botlibre-ai.jar, **do not copy** botlibreplatform.jar, do not overwrite any files already in tomcat lib) – ** you can also download these jars from <https://www.botlibre.biz/download/lib.zip> which has the exact jars that should be put into tomcat/lib
- Copy speech voice files to /usr/local/tomcat8/bin/lib (unzip speech voice zip into lib voices directory) – ** you can download the voice files from <https://www.botlibre.biz/download/voices52.zip> extract to tomcat/bin/lib (this will create a voices directory in lib)
- Configure server.xml, web.xml
- Ensure war file is configured with correct database password, domain name, and branding (this must be set in the Site.java file, and the war must be recompiled using the build.xml ant task in Eclipse) – ** This is no longer required, as the website will let you change these settings in the Admin Console.
- Deploy war file (rename to ROOT.war, delete all files from tomcat/webapps)
- Start tomcat
- Login as “admin” “password” (** change password)
- (no longer required in Bot Libre 8) If not created already, create the first bot named “template” with “Is Template” clicked
 - Go to the bots Admin Console, and go to its Knowledge, Status page, click “Delete All” (only do this for the default template, or to reset a bot)
 - Go to the Learning & Settings page and disable learning, set correction to administrators, disable comprehension, consciousness, and grammar
 - Go to the Scripts page, click “Rebootstrap”, delete the “Understanding”, “WhatIs”, and “WhereIs” scripts.
- Create a new test bot using the default template, test the bot is functioning.
- (optional) Import the avatars from the install /avatars directory from the website Avatars section
- (optional) Import the scripts from the install /scripts directory from the website Scripts section

>> deploying war file

```
cd /usr/local/tomcat8/bin
```

```
./shutdown.sh
```



```
cd /usr/local/tomcat8/webapps
rm -rf *
cp /home/ftp/botlibreplatform.war ROOT.war
cd /usr/local/tomcat8/bin
./startup.sh
```

Backups

Ensure you backup any important data on the server on a regular basis.

Most of the software can be reinstalled, so is not important to backup.

The database should be backed up.

The following is an example of backing up the database. It backup is stored on the 1st disk, where the database is on the 2nd disk for fault tolerance. It is recommended to also copy the backup file to a remote server or remote backup service.

```
>>
chmod a+rwX /backup
cd /usr/pgsql-9.4/bin/
su postgres
pg_dumpall > /backup/backup-2018-01-01.sql
```