Installing LaBB-CAT on Red Hat Linux

Broad steps are:

    i. Install Apache Tomcat 9
   ii. Install MariaDB
  iii. Install third-party tools (Praat, ffmpeg, HTK if required)
  iv. Install LaBB-CAT webapp

Additional steps might include

- Setting up password protection
- Integrating Apache web server with Tomcat

Tomcat

Install Tomcat 9, later versions are currently not tested.

(1) Install JDK with the following command:
```
sudo dnf install wget java-11-openjdk-devel
```

(2) Download and unpack the Tomcat software with the following commands:
```
cd /usr/local
sudo wget https://downloads.apache.org/tomcat/tomcat-9/v9.0.95/bin//apache-tomcat-9.0.95.tar.gz
sudo tar -xvf apache-tomcat-9.0.95.tar.gz
sudo mv apache-tomcat-9.0.95 tomcat9
```

(3) Create a user for the Tomcat service with the following commands:
```
sudo useradd -r tomcat
sudo chown -R tomcat:tomcat /usr/local/tomcat9
```

(4) Create a service file */etc/systemd/system/tomcat.service* with your favourite text editor, e.g.
```
sudo vim /etc/systemd/system/tomcat.service
```
The content should be:

```
[Unit]
Description=Apache Tomcat Server
After=syslog.target network.target
[Service]
Type=forking
User=tomcat
Group=tomcat
```

```
Environment=CATALINA_PID=/usr/local/tomcat9/temp/tomcat.pid
Environment=CATALINA_HOME=/usr/local/tomcat9
Environment=CATALINA_BASE=/usr/local/tomcat9
ExecStart=/usr/local/tomcat9/bin/catalina.sh start
ExecStop=/usr/local/tomcat9/bin/catalina.sh stop
RestartSec=10
Restart=always
[Install]
WantedBy=multi-user.target
```

(5) Remove example webapps that are installed by default: `sudo rm -r /usr/local/tomcat9/webapps/docs` `sudo rm -r /usr/local/tomcat9/webapps/host-manager` `sudo rm -r /usr/local/tomcat9/webapps/`

(6) Enable/start the Tomcat service with the following commands: `sudo systemctl daemon-reload` `sudo systemctl enable tomcat.service` `sudo systemctl start tomcat.service`

(7) Ensure Tomcat can receive requests from external clients: `sudo firewall-cmd --zone=public --permanent --add-port=8080/tcp` `sudo firewall-cmd --reload`

At this point, Tomcat should be working. You can check this by opening your browser and browsing to your server's host name, on port 8080, something like:
`http://example.com:8080/`
(Substitute *example.com* with the host name of your server.).
This should display the Apache Tomcat default page (and not en error).

(8) Ensure Tomcat's *catalina.out* log file is rotated by creating a *logrotate* configuration file called */etc/logrotate.d/tomcat* with your favourite text editor, e.g.
`sudo vim /etc/logrotate.d/tomcat`

(9) Add the following content to the file, and save it:

```
/usr/local/tomcat9/logs/catalina.out {
    copytruncate
    daily
    rotate 7
    compress
    missingok
    size 5M
}
```

MariaDB

(10) Install MariaDB with the following command:
```
sudo yum install mariadb
```
(11) Enable and system its service with the following commands:
```
sudo systemctl enable mariadb
sudo systemctl start mariadb
sudo systemctl stop mariadb
```

Third Party Tools

LaBB-CAT integrates with various tird party software tools, which you can install now. These steps are optional.

Praat

If you want to user LaBB-CAT's batch Praat processing functionality, you must install Praat on the server, with the following commands:

1. `sudo wget https://www.fon.hum.uva.nl/praat/praat6420_linux-intel64-barren.tar.gz`
2. `sudo tar zxvf praat6420_linux-intel64-barren.tar.gz`
3. `sudo cp praat_barren /usr/bin/praat`

ffmpeg

If you want LaBB-CAT to be able to automatically convert media between formats (e.g. automatically extract a .wav audio file from uploaded videos) you must install Praat on the server, with the following commands:

1. `sudo dnf install epel-release`
2. `sudo dnf install https://download.fedoraproject.org/pub/epel/epel-release-latest-9.noarch.rpm`
3. `sudo dnf install https://dl.fedoraproject.org/pub/epel/epel-next-release-latest-9.noarch.rpm`
4. `sudo dnf install --nogpgcheck https://mirrors.rpmfusion.org/free/el/rpmfusion-free-release-$(rpm -E %rhel).noarch.rpm`
5. `sudo dnf install https://mirrors.rpmfusion.org/nonfree/el/rpmfusion-nonfree-release-$(rpm -E %rhel).noarch.rpm`
6. `sudo dnf install https://mirror.stream.centos.org/9-stream/CRB/x86_64/os/Packages/ladspa 1.13-28.el9.x86_64.rpm`
7. `sudo dnf install ffmpeg ffmpeg-devel`

HTK

If you want to user HTK for forced alginment, you must install HTK on the server, using the following steps.

1. Ensure you have registered a username and password on http://htk.eng.cam.ac.uk/register.shtml
2. Install prerequisites for coompiling HTK from source: `sudo yum -y install glibc-devel.i686 libX11-devel.i686 glibc-devel.i686 libstdc++-devel.i686`
3. Download and unpack the source code: `cd /tmp`
4. In the following command, replace *my-username* with the username you registered on the HTK site: `wget --user=my-username --ask-password http://htk.eng.cam.ac.uk/ftp/software/ 3.4.1.tar.gz`
5. Enter your HTK password when asked.
6. `tar -zxf HTK-3.4.1.tar.gz`
7. Prepare for compilation: `cd htk`
8. `./configure`
9. Open the *HLMTools/Makefile* file with your favourite text editor, e.g. `vim HLMTools/Makefile` …and replace spaces on line 77 with a tab character. Then save the file.
10. Compile and install the software with the following commands: `make all sudo make install`

LaBB-CAT webapp

Once Tomcat and MariaDB are installed, the LaBB-CAT web application can be installed, using the following steps:

(12) Download the most recent *labbcat-server_yyyymmdd.zip* file from SourceForge:
https://sourceforge.net/projects/labbcat/files/install/

(13) Unzip the resulting file, will with contain LaBB-CAT's Web Application Archive, called *labbcat.war*

(14) Copy *labbcat.war* into Tomcat's *webapps* directory with a command like:
`sudo cp labbcat.war /usr/local/tomcat9/webapps/`

Tomcat will automatically unpack the file into a subirectory with the same name as the .war file. You can check this is successful with the following command:
`sudo ls /usr/local/tomcat9/webapps/labbcat/`
…which should show a list of files, rather than returning an error.

(15) In your web browser, open the *labbcat* webapp page, with a URL like:
`http://example.com:8080/labbcat/`
(Substitute *example.com* with the host name of your server.)

You should see a page with the title: *web application self-installation* and a form asking for information for database information.

(16) Next to the title *LaBB-CAT Database Exists* there is an expandable section labelled:
A MySQL database and MySQL user can be pre-created
Click this section to expand it, revealing SQL statements for pre-creating the database.

There are four SQL commands that look something like this:

```
CREATE DATABASE IF NOT EXISTS labbcat CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;
CREATE USER 'labbcat'@'%' IDENTIFIED BY ...;
GRANT ALL PRIVILEGES ON labbcat.* TO 'labbcat'@'%';
GRANT PROCESS ON *.* TO 'labbcat'@'%';
```

(17) Back in you command shell on the server on which MariaDB is installed, start the *mysql* command line interface with the following command:
`sudo mysql`
This will display a command prompt something like: `MariaDB [(none)]>`

(18) Copy and paste the four SQL commands from your browser into the MariaDB command prompt, and hit `return` to execute all of them.

(19) Exit from the MariaDDB command prompt with the following command:
`exit`

(20) Back in your browser, on the *web application self-installation* page, tick the box labelled:
The above database already exists, so don't create it.

(21) Press the *install* button.

You should see a page listing steps that are being taken during the webapp installation, with
Installation is complete.
…printed at the bottom.

After a short delay, the page will reload, and the LaBB-CAT home page will be displayed.

LaBB-CAT is now installed on your server, and can be accessed with the URL shown in your browser.

Additional steps

Password Protection

When first installed, LaBB-CAT can be accessed by anyone who has a browser connection to the server. You probably want to password-protect LaBB-CAT to restrict access.

Follow these optional steps to enable password protection:

(22) Ensure that Tomcat itself can connect to the MariaDB database to authenticate users, using the following commands:
```
sudo cp /usr/local/tomcat9/webapps/labbcat/WEB-INF/lib/mysql-connector-
java-latest.jar /usr/local/tomcat9/lib/
sudo chown tomcat:tomcat /usr/local/tomcat9/lib/mysql-connector-java-
latest.jar
```
(23) Open LaBB-CAT's *WEB-INF/web.xml* file with your favourite text editor, e.g.
```
sudo vim /usr/local/tomcat9/webapps/labbcat/WEB-INF/web.xml
```
(24) Delete the lines that contain the phrase USER-SECURITY - there should be two lines, one that opens an XML comment, and one that closes it.
(25) Save the *web.xml* file.
(26) Restart Tomcat to ensure the changes take effect, using the followiing command:
```
sudo systemctl restart tomcat.service
```
(27) In your web browser, open LaBB-CAT's home-page (or reload the page if it's already displayed).
You should be asked for a username and password.
(If you see an error, close your browser completely down and start it up again)
(28) Enter the username `labbcat` and the password `labbcat`.
You should see a page asking you to change your password.
(29) Set your `labbcat` user's password to something memorable, and the LaBB-CAT home page will be displayed.

LaBB-CAT is now password protected.

You can add new users using the *users* link at the top of the page:

1. Fill in the *User ID* box at the top of the page.
2. Press the *New* button on the right.
3. Set an initial password for the new user by entering it twice.
4. Press the *Save* button that appears after you've entered the same new password twice.

The new user will have read-only access to LaBB-CAT. If you want them to have more privileges, you need to tick corresponding *Roles* checkboxes for their user:

- *edit*: the user can upload transcripts, media, and transcript/participant meta-data.
- *admin*: the user can do anything at all, and

- *export*: the user can export a copy of the whole LaBB-CAT instance.

Tomcat integration with Apache

Your server may already have Apache web server installed, e.g. for serving requests on the normal HTTP port and/or supporting encrypted HTTPS connections.

Tomcat can be integrated with Apache, so that HTTP/HTTPS requests received by Apache are handled by Tomcat.

Steps to integrate Apache Web Server with Tomcat are are:

1. Install package dependencies:
   ```
   sudo dnf install httpd-devel apr apr-devel apr-util apr-util-devel gcc
   gcc-c++ make autoconf libtool httpd-devel.x86_64
   ```

2. Download and install Jk:

   1. `sudo mkdir -p /opt/mod_jk/`
   2. `cd /opt/mod_jk/`
   3. `sudo wget https://downloads.apache.org/tomcat/tomcat-connectors/jk/tomcat-connectors-1.2.50-src.tar.gz`
   4. `sudo tar -xvzf tomcat-connectors-1.2.50-src.tar.gz`
   5. `cd tomcat-connectors-1.2.50-src/native`
   6. `sudo ./configure --with-apxs=/usr/bin/apxs`
   7. `sudo make`
   8. `sudo libtool --finish /usr/lib64/httpd/modules`
   9. `sudo cp ./apache-2.0/mod_jk.so /etc/httpd/modules/mod_jk.so`

3. Configure Tomcat to accept connections from Apache:

   1. Open */etc/tomcat/server.xml* with your favourite text editor, e.g.
      ```
      sudo vim /etc/tomcat/server.xml
      ```

   2. Change the *Engine* tag to be:
      ```
      <Engine name="Catalina" defaultHost="localhost" jvmRoute="jvm1">
      ```

   3. Comment **out** the connector for port 8080:
      ```
      <!-- <Connector port="8080" protocol="HTTP/1.1" redirectPort="8443"
      />  -->
      ```

   4. Comment **in** the connector for 8009, and set the attributes as follows:
      ```
      <Connector protocol="AJP/1.3"
                 address="127.0.0.1"
                 port="8009"
                 maxParameterCount="-1"
                 redirectPort="8443"
                 secretRequired="false"/>
      ```

5. Save the file.

4. Create the Jk module configuration file */etc/httpd/conf.d/mod_jk.conf* with the following content:

```
LoadModule jk_module "/etc/httpd/modules/mod_jk.so"
JkWorkersFile /etc/httpd/conf/workers.properties
JkShmFile     /var/run/httpd/mod_jk.shm
JkLogFile     /var/log/httpd/mod_jk.log
JkLogLevel    info
JkLogStampFormat "[%a %b %d %H:%M:%S %Y] "
#JkRequestLogFormat "%w %V %T"
#JkEnvVar SSL_CLIENT_V_START worker1
```

5. Create */var/run/mod_jk*:

   1. sudo mkdir -p /var/run/mod_jk
   2. sudo chown apache:apache /var/run/mod_jk

6. Create the */etc/httpd/conf/workers.properties* file with the following content:

```
workers.apache_log=/var/log/httpd
worker.list=app1Worker
worker.stat1.type=status
worker.app1Worker.type=ajp13
worker.app1Worker.host=127.0.0.1
worker.app1Worker.port=8009
```

7. Create the */etc/httpd/conf.d/tomcat.conf* file with the following content (substituting *example.com* with your server's host name):

```
<VirtualHost *:80>
  ServerName example.com
  ServerAdmin webmaster@example.com
  LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-agent}i\"" combined
  CustomLog /var/log/httpd/tomcat.log combined
  ErrorLog /var/log/httpd/tomcat.log
  <IfModule mod_jk.c>
     JkMount /* app1Worker
     # so that certbot works:
     JkUnMount /.well-known/*  app1Worker
  </IfModule>
</VirtualHost>
```

8. Restart Tomcat and Apache Web Server:

        1. `sudo service tomcat restart`
        2. `sudo service httpd restart`

9. If you are using SElinux, you may see an error when restarting `httpd`. If so, execute the following commands:

        1. `sudo ausearch -c 'httpd' --raw | audit2allow -M my-httpd`
        2. `sudo semodule -i my-httpd.pp`
        3. `sudo service httpd restart`

Now ordinary browser connections like `http://example.com/labbcat` should show the LaBB-CAT home page, and if you have HTTPS configured with an SSL certificat, encrypted connections like `https://example.com/labbcat` should work.