**Purpose:**

To manually install the most recent Jetty as a standalone service on RHEL 7+. This process describes a command-line installation, so should work on most Linux flavours. It closely follows that outlined by the Jetty instructions for CentOS7/RHEL7. It creates a ‘jetty’ user. It also includes installation of geonetwork, since that is a reasonably complex application example.

Despite many apparent steps, there are actually the following phases:

1. download the jetty webserver application
2. create a jetty user
3. create necessary folders, parameters, files and links to run jetty as a service. Jetty options will be located in /etc/default/jetty, and link to startup file jetty.sh (or actual file) is placed in /etc/init.d/jetty
4. start the jetty application to auto-run as a service. This involves linking init.d/jetty to the runLevel.

The result could be one of the following directory structures, both of which allow for easy upgrade of jetty because of separation of the (read only) installation directory, and the base directory. In both cases, startup variables are in /etc.

***Install goal (manual or apt).***

The read-only java jars are placed in a separate Jetty\_Home location. The logs and webapps folders are located elsewhere in a Jetty\_Base location (e.g. /var/www) but symbolically linked in the JETTY\_HOME directory, as are the /etc variables.

|  |  |  |  |
| --- | --- | --- | --- |
| **/usr/share/java/jetty-distro**  **(JETTY\_HOME, read only)** | | **/var/www** | **/var/www/jetty**  **(JETTY\_BASE)** |
| |\_ start.d/ | |\_resources/ | |\_jetty/ | |\_lib/ |
| |\_ bin/ | |\_ logs/ | |\_temp/ | |\_resources/ |
| |\_ lib/ | |\_ start.ini | |\_geowebcache/ | |\_start.d/ |
| |\_logs/ | |\_webapps | |\_gn\_dir | |\_webapps/ |
| |\_etc/ |  | |\_gs\_dir/ |  |

***Jetty-Java version compatibility***

| **Version** | **Years** | **Home** | **Min JVM** | **Servlet** | **JSP** | **Status** | **Protocols** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 11 | 2020- | Eclipse | 11 (2) | 5.0 | 3.0 | **Stable** | HTTP/1.1 (RFC 7230), HTTP/2 (RFC 7540), WebSocket (RFC 6455, JSR 356), FastCGI, **JakartaEE Namespace**(1) |
| 10 | 2020- | Eclipse | 11 (2) | 4.0 | 2.3 | **Stable** | HTTP/1.1 (RFC 7230), HTTP/2 (RFC 7540), WebSocket (RFC 6455, JSR 356), FastCGI |
| 9.4 | 2016- | Eclipse | 1.8 | 3.1 | 2.3 | Stable | HTTP/1.1 (RFC 7230), HTTP/2 (RFC 7540), WebSocket (RFC 6455, JSR 356), FastCGI |

| **Step** | **Major Activity** | **References, Forms and Details** |
| --- | --- | --- |
| **1** | Download the most recent java-version-compatible Jetty package from  *http://www.eclipse.org/jetty/download.html* | * Select the .tgz version * Max version 9-series if using java8 * Copy the path, and use with wget |
| **2** | Create folders  *sudo mkdir –p /usr/share/java/*  *sudo mkdir –p /var/www/temp*  *sudo mkdir –p /var/www/{mybase}* | * The “-p” option causes intermediate folders to be generated as needed * ***/usr/share/java/*** will hold the Jetty distroas a subfolder * ***/var/www/temp***temporary directory assigned to Java by the Service Layer * ***/var/www/{mybase}***is where the specific set of webapps will be located, including all server configuration. The name ‘{mybase}’ should be replaced with a desired name (e.g. jetty). |
| **3** | Extract the downloaded archive  *sudo tar zxvf <jettyDistro>.tar.gz –C /usr/share/java/* | * <jettyDistro>.will be used without modification * present jettyDistro=*jetty-distribution-9.4.21.v20190926* * Customizations will occur in the mybase folder |
| **4** | Set up symlinks to folders in a normal startup location (e.g. ~/.bashrc )  *export JETTY\_HOME=/usr/share/java/{jettyDistro}*  *export JETTY\_BASE=/var/www/{mybase}*  *export JETTY\_PORT=8080 (optional)*  *export JETTY\_HOST={ip-address} (optional)*  Then exit to home and issue  . ~/.bashrc  Check: env | grep JETTY | * “dot-space-squiggle-slash-dot-bashrc” causes the file to be re-read |
| **5** | Create a user called jetty to run the webserver on system start-up  *sudo useradd --shell /bin/false \*  ***--****home-dir /$JETTY\_BASE/temp jetty*  *sudo usermod -a -G jetty {mySysID}* | * user jetty, in group jetty * no shell access * home directory as shown * add whomever (replace {mySysID}) as user(s) to group jetty |
| **6** | List users and groups:  for user in $(awk -F: \  '{print $1}' /etc/passwd); \  do groups $user; \  done | * should see user jetty as a member of group jetty * should also see all {mySysID} users as members of group jetty |
| **7** | Change ownership of jetty directories  *sudo chown -R root:root $JETTY\_HOME*  *sudo chown -R jetty:jetty $JETTY\_BASE* |  |
| **8** | Set folder properties:  sudo chmod –R 755 $JETTY\_BASE | jetty rwx, others rx |
| **9** | Jetty test – try to start demos:  *cd $JETTY\_HOME/demo-base*  *java* ***-****jar $JETTY\_HOME/start.jar*  then use browser to http://localhost:8080 | Feedback should describe   * logging initiated * deployment monitoring location * server start on port 8080 * in browser, should see splash screen with several examples accessible * May need to clear browser cache * You are now ready to establish jetty as a daemon (to start on reboot) |
| **10** | Establish the jetty control file   * Change to /etc/systemd/system * Use an editor to create jetty.service | * need root privileges to edit * create a document similar to 330-F20   + memory usage   + java as headless   + geowebcache location and name   + console logging geoserver loction and name. if used   + geonetwork support parameters |
| **11** | Initialize $JETTY\_BASE directory  *cd $JETTY\_BASE*  *sudo* ***-****u jetty java* ***-j****ar $JETTY\_HOME/start.jar \*  ***--****create-startd \*  ***--****add-to-start=http,deploy,jndi,jdbc,jsp,console-capture,logging-log4j,plus*  Check with  *java –jar $JETTY\_HOME/start.jar –llist-config* | Result should be:   * *webapps/* directory (empty) * *logs/* directory (empty) * *start.d/* directory with \*.ini files * “add-to-start” generates the ini files in start.d folder. *No spaces* in “add-to-start” string * console-capture enables logging (**important)** * jsp for geoserver * plus for jndi if using that type of databaseconnection * https and ssl for secure access   webserver not yet running |
| **12** | Add WAR files to  *$JETTY\_BASE/webapps*/*<war file>* | The *<war file>* could be:   * geonetwork.war * geoserver.war * any add-ins |
| **13** | Expand the WAR files via unzip  Move the war file out of webapps, or it will take precedence | * often modify the expanded WAR files to point at a database like PostGIS |
| **14** | Configure the service to auto start, by :  sudo systemctl enable jetty.service |  |
| **15** | Reboot computer for clean start   * *sudo reboot*   Check that jetty is running:  *service jetty status* | can take a minute or two, then the command line response resumes |
| **16** | Check the jetty log and the geonetwork log for errors  cd $JETTY\_BASE/logs |  |
| **17** | Jetty and apps can be accessed via  *http://{ip-address}:8080* |  |