

Installation of Pentaho components and setting up connections

Pentaho can download from <http://sourceforge.net/projects/pentaho/files/Business%20Intelligence%20Server/6.1/biserver-ce-6.1.0.1-196.zip/download>.

1) installation steps are in dockerfile

a) for local database server

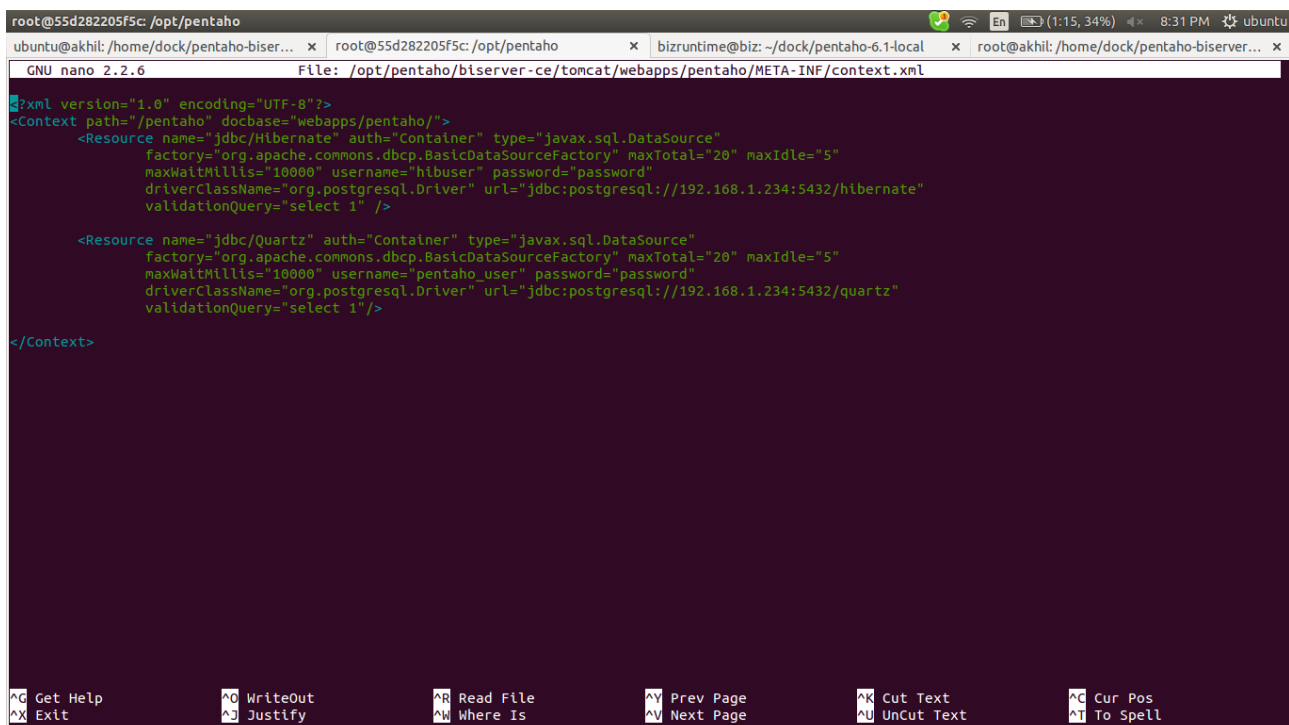
links : git clone -b master <https://github.com/akhilrajmailbox/pentaho-6.1-local.git>

b) for remote database server

links : git clone -b master-remote-db <https://github.com/akhilrajmailbox/pentaho-6.1-local.git>

configuration files which i changed for remote database (192.168.1.234) are given below for make local database “192.168.1.234” changed to “localhost”

2) */opt/pentaho/biserver-ce/tomcat/webapps/pentaho/META-INF/context.xml*



```
root@55d282205f5c: /opt/pentaho
ubuntu@akhil:/home/dock/pentaho-biser... x root@55d282205f5c: /opt/pentaho x bizruntime@biz: ~/dock/pentaho-6.1-local x root@akhil:/home/dock/pentaho-biserver... x
GNU nano 2.2.6 File: /opt/pentaho/biserver-ce/tomcat/webapps/pentaho/META-INF/context.xml
<?xml version="1.0" encoding="UTF-8"?>
<Context path="/pentaho" docbase="webapps/pentaho/">
  <Resource name="jdbc/Hibernate" auth="Container" type="javax.sql.DataSource"
    factory="org.apache.commons.dbcp.BasicDataSourceFactory" maxTotal="20" maxIdle="5"
    maxWaitMillis="10000" username="hibuser" password="password"
    driverClassName="org.postgresql.Driver" url="jdbc:postgresql://192.168.1.234:5432/hibernate"
    validationQuery="select 1" />

  <Resource name="jdbc/Quartz" auth="Container" type="javax.sql.DataSource"
    factory="org.apache.commons.dbcp.BasicDataSourceFactory" maxTotal="20" maxIdle="5"
    maxWaitMillis="10000" username="pentaho_user" password="password"
    driverClassName="org.postgresql.Driver" url="jdbc:postgresql://192.168.1.234:5432/quartz"
    validationQuery="select 1" />
</Context>

^G Get Help      ^O WriteOut     ^R Read File    ^Y Prev Page    ^K Cut Text     ^C Cur Pos
^X Exit          ^J Justify      ^W Where Is     ^V Next Page    ^U UnCut Text   ^T To Spell
```

3) */opt/pentaho/biserver-ce/pentaho-solutions/system/applicationContext-spring-security-
hibernate.properties*

```
root@55d282205f5c: /opt/pentaho
ubuntu@akhil:/home/dock/pentaho-biser... x root@55d282205f5c: /opt/pentaho x bizruntime@biz: ~/dock/pentaho-6.1-local x root@akhil:/home/dock/pentaho-biserver... x
GNU nano 2.2.6 File: /opt/pentaho/biserver-ce/pentaho-solutions/system/applicationContext-spring-security-hibernate.properties
jdbc.driver=org.postgresql.Driver
jdbc.url=jdbc:postgresql://192.168.1.234:5432/hibernate
jdbc.username=hibuser
jdbc.password=password
hibernate.dialect=org.hibernate.dialect.HSQLDialect

^G Get Help      ^O WriteOut      ^R Read File      Read 5 lines      ^Y Prev Page      ^K Cut Text
^X Exit          ^J Justify       ^W Where Is       ^V Next Page      ^U UnCut Text
sanath dm has gone offline
```

4) */opt/pentaho/biserver-ce/pentaho-solutions/system/hibernate/hibernate-settings.xml*

```
root@55d282205f5c: /opt/pentaho
ubuntu@akhil:/home/dock/pentaho-biser... x root@55d282205f5c: /opt/pentaho x bizruntime@biz: ~/dock/pentaho-6.1-local x root@akhil:/home/dock/pentaho-biserver... x
GNU nano 2.2.6 File: /opt/pentaho/biserver-ce/pentaho-solutions/system/hibernate/hibernate-settings.xml
<?xml version='1.0' encoding='utf-8'?>
<settings>

<!--
 * This setting allows the deployment to specify where to find the
 * database-specific hibernate configuration. The samples supplied
 * Include the following:
 *
 * system/hibernate/postgresql.hibernate.cfg.xml
 * system/hibernate/mysql5.hibernate.cfg.xml
 * system/hibernate/postgresql.hibernate.cfg.xml
 * system/hibernate/oracle10g.hibernate.cfg.xml
 *
 -->
  <config-file>system/hibernate/postgresql.hibernate.cfg.xml</config-file>

<!--
 *
 * managed should be set to true if running the BI Platform
 * in a managed environment (like JBoss, Orion, etc). In this configuration,
 * you should specify another location for the hibernate.cfg.xml (see below)
 * instead of simply using the default one provided. This setting essentially
 * tells the HibernateUtil class to use JNDI to locate the factory class for
 * getting sessions. This allows the platform to use Hibernate across boundaries
 * in message beans (for example).
 *
 * <managed>false</managed>
 -->
  <managed>false</managed>
</settings>
```

5) */opt/pentaho/biserver-ce/pentaho-solutions/system/simple-jndi/jdbc.properties*

```
root@55d282205f5c: /opt/pentaho
ubuntu@akhil:/home/dock/pentaho-biser... x root@55d282205f5c:/opt/pentaho x bizruntime@biz: ~/dock/pentaho-6.1-local x root@akhil:/home/dock/pentaho-biserver... x
GNU nano 2.2.6 File: /opt/pentaho/biserver-ce/pentaho-solutions/system/simple-jndi/jdbc.properties Modified

# Copyright 2008 - 2010 Pentaho Corporation. All rights reserved.
# This program is free software; you can redistribute it and/or modify it under the
# terms of the GNU Lesser General Public License, version 2.1 as published by the Free Software
# Foundation.
#
# You should have received a copy of the GNU Lesser General Public License along with this
# program; if not, you can obtain a copy at http://www.gnu.org/licenses/old-licenses/lgpl-2.1.html
# or from the Free Software Foundation, Inc.,
# 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.
#
# This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY;
# without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
# See the GNU Lesser General Public License for more details.

#SampleData/type=javax.sql.DataSource
#SampleData/driver=org.hsqldb.jdbcDriver
#SampleData/url=jdbc:hsqldb:hsq://localhost/sampledata
#SampleData/user=pentaho_user
#SampleData/password=password

Hibernate/type=javax.sql.DataSource
Hibernate/driver=org.postgresql.Driver
Hibernate/url=jdbc:postgresql://192.168.1.234:5432/hibernate
Hibernate/user=hibuser
Hibernate/password=password

Quartz/type=javax.sql.DataSource
Quartz/driver=org.postgresql.Driver
Quartz/url=jdbc:postgresql://192.168.1.234:5432/quartz
Quartz/user=pentaho_user
Quartz/password=password

#Shark/type=javax.sql.DataSource
#Shark/driver=org.hsqldb.jdbcDriver
#Shark/url=jdbc:hsqldb:hsq://localhost/shark
#Shark/user=sa
#Shark/password=

Get Help WriteOut Read File Prev Page Cut Text Cur Pos
Exit Justify Where Is Next Page UnCut Text To Spell
```

6) */opt/pentaho/biserver-ce/pentaho-solutions/system/hibernate/postgresql.hibernate.cfg.xml*

```
root@55d282205f5c: /opt/pentaho
ubuntu@akhil:/home/dock/pentaho-biser... x root@55d282205f5c:/opt/pentaho x bizruntime@biz: ~/dock/pentaho-6.1-local x root@akhil:/home/dock/pentaho-biserver... x
GNU nano 2.2.6 File: /opt/pentaho/biserver-ce/pentaho-solutions/system/hibernate/postgresql.hibernate.cfg.xml

<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE hibernate-configuration
PUBLIC "-//Hibernate/Hibernate Configuration DTD//EN"
"http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
<session-factory>
<property name="cache.provider_class">net.sf.ehcache.hibernate.SingletonEhCacheProvider</property>
<property name="hibernate.generate_statistics">true</property>
<property name="hibernate.cache.use_query_cache">true</property>
<!-- Postgres 8 Configuration -->
<property name="connection.driver_class">org.postgresql.Driver</property>
<property name="connection.url">jdbc:postgresql://192.168.1.234:5432/hibernate</property>
<property name="dialect">org.hibernate.dialect.PostgreSQLDialect</property>
<property name="connection.username">hibuser</property>
<property name="connection.password">password</property>
<property name="connection.pool_size">10</property>
<property name="show_sql">>false</property>
<property name="hibernate.jdbc.use_streams_for_binary">true</property>
<!-- replaces DefinitionVersionManager -->
<property name="hibernate.hbm2ddl.auto">update</property>
<!-- load resource from classpath -->
<mapping resource="hibernate/postgresql.hbm.xml" />
</session-factory>
</hibernate-configuration>
```

7) */opt/pentaho/biserver-ce/pentaho-solutions/system/applicationContext-spring-security-jdbc.xml*

```
root@55d282205f5c: /opt/pentaho
ubuntu@akhil:/home/dock/pentaho-biser... x root@55d282205f5c: /opt/pentaho x bizruntime@biz: ~/dock/pentaho-6.1-local x root@akhil:/home/dock/pentaho-biserver... x
GNU nano 2.2.6 File: /opt/pentaho/biserver-ce/pentaho-solutions/system/applicationContext-spring-security-jdbc.xml
<property name="roleMapper" ref="jdbcRoleMapper" />
<constructor-arg ref="tenantedUserNameUtils"/>
<pen:publish as-type="INTERFACES">
  <pen:attributes>
    <pen:attr keys="providerName" value="jdbc"/>
  </pen:attributes>
</pen:publish>
</bean>

<bean id="dataSource" class="org.apache.commons.dbcp.BasicDataSource">
  <property name="driverClassName" value="org.postgresql.Driver" />
  <property name="url" value="jdbc:postgresql://192.168.1.234:5432/hibernate" />
  <property name="username" value="hibuser" />
  <property name="password" value="password" />
  <!-- the following are optional -->
  <property name="validationQuery" value="${datasource.validation.query}" />
  <property name="maxWait" value="${datasource.pool.max.wait}" />
  <property name="maxActive" value="${datasource.pool.max.active}" />
  <property name="maxIdle" value="${datasource.max.idle}" />
  <property name="minIdle" value="${datasource.min.idle}" />
</bean>

<bean class="org.springframework.beans.factory.config.PropertyPlaceholderConfigurer">
  <property name="location" value="applicationContext-spring-security-jdbc.properties" />
</bean>

<bean id="jdbcPasswordEncoder"
  class="org.springframework.security.providers.encoding.PlaintextPasswordEncoder" />
</beans>
```

8) Edit the file */opt/pentaho/biserver-ce/tomcat/webapps/pentaho/WEB-INF/web.xml* to prevent HSQLDB to start

```
root@55d282205f5c: /opt/pentaho
ubuntu@akhil:/home/dock/pentaho-biser... x root@55d282205f5c: /opt/pentaho x bizruntime@biz: ~/dock/pentaho-6.1-local x root@akhil:/home/dock/pentaho-biserver... x
GNU nano 2.2.6 File: /opt/pentaho/biserver-ce/tomcat/webapps/pentaho/WEB-INF/web.xml
<?xml version="1.0"?>
<!DOCTYPE web-app PUBLIC
  "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
  "http://java.sun.com/dtd/web-app_2_3.dtd">

<web-app>

  <context-param>
    <param-name>solution-path</param-name>
    <param-value></param-value>
  </context-param>

  <context-param>
    <param-name>contextClass</param-name>
    <param-value>org.pentaho.platform.web.http.context.PentahoSolutionSpringApplicationContext</param-value>
  </context-param>

  <context-param>
    <param-name>contextConfigLocation</param-name>
    <!-- This file is relative to the "system" folder within your pentaho solutions folder. -->
    <!-- Instead of adding files here, add them as imports to pentaho-spring-beans.xml. -->
    <param-value>pentaho-spring-beans.xml</param-value>
  </context-param>

  <!-- [BEGIN HSQLDB DATABASES] -->
  <context-param>
    <param-name>hsqldb-databases</param-name>
    <param-value>sampledata@../data/hsqldb/sampledata,hibernate@../data/hsqldb/hibernate,quartz@../data/hsqldb/quartz</param-value>
  </context-param>
  <!-- [END HSQLDB DATABASES] -->

  <!-- Insert additional context-params -->

  <filter> <!-- This must be the first filter listed in the web.xml -->
    <filter-name>Set Character Encoding Filter</filter-name>
    <filter-class>org.pentaho.platform.web.http.filters.PentahoAwareCharacterEncodingFilter</filter-class>
    <init-param>
      <param-name>ignore</param-name>
      <param-value>yes</param-value>
    </init-param>
  </filter>
```

change

<!-- [BEGIN HSQLDB DATABASES] --> to *<!-- [BEGIN HSQLDB DATABASES]*

and

<!-- [END HSQLDB DATABASES] --> to *[END HSQLDB DATABASES] -->*

```
root@55d282205f5c: /opt/pentaho
ubuntu@akhil: /home/dock/pentaho-biser... x root@55d282205f5c: /opt/pentaho x bizruntime@biz: ~/dock/pentaho-6.1-local x root@akhil: /home/dock/pentaho-biser... x
GNU nano 2.2.6 File: /opt/pentaho/biserver-ce/toncat/webapps/pentaho/WEB-INF/web.xml

</listener>
<!-- enables session and request scoped object creation in Spring -->
<listener>
<listener-class>org.springframework.web.context.request.RequestContextListener</listener-class>
</listener>

<!-- [BEGIN HSQLDB STARTER]
<listener>
<listener-class>org.pentaho.platform.web.http.context.HsqlDbStartupListener</listener-class>
</listener>
[END HSQLDB STARTER] -->

<listener>
<listener-class>org.pentaho.platform.web.http.session.PentahoHttpSessionListener</listener-class>
</listener>

<!-- NEXT TWO LISTENERS ARE ORDER DEPENDENT -->
<listener>
<listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
</listener>

<listener>
<listener-class>org.pentaho.platform.web.http.context.SolutionContextListener</listener-class>
</listener>
<!-- END OF ORDER DEPENDENCE -->

<listener>
<listener-class>org.springframework.security.ui.session.HttpSessionEventPublisher</listener-class>
</listener>

<listener>
<listener-class>org.pentaho.platform.web.http.context.PentahoCacheContextListener</listener-class>
</listener>

<listener>
<listener-class>org.pentaho.platform.web.http.session.PentahoCacheSessionListener</listener-class>
</listener>

<listener>
<listener-class>net.sf.ehcache.constructs.web.ShutdownListener</listener-class>
</listener>

Get Help WriteOut Read File Prev Page Cut Text Cur Pos
Exit Justify Where Is Next Page UnCut Text To Spell
```

change

<!-- [BEGIN HSQLDB STARTER] --> to <!-- [BEGIN HSQLDB STARTER]
and
<!-- [END HSQLDB STARTER] --> to [END HSQLDB STARTER] -->

9) Make the .sh files executable
sudo chmod +x /opt/pentaho/biserver-ce/*.sh

Make the .sh files executable

/opt/pentaho/biserver-ce/start-pentaho.sh

Go to

http://ipaddress-pentaho-server:8080

10) refer links for auto start up

a) cron job

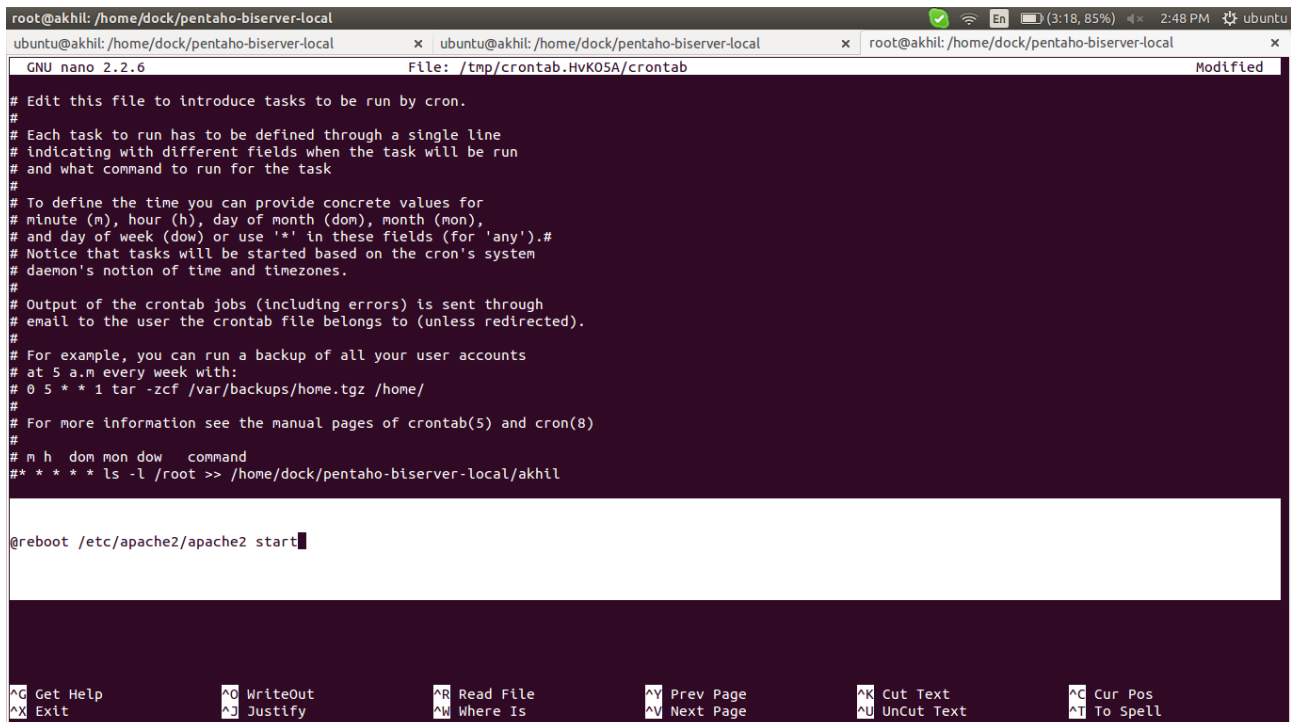
Use a crontab option to make your script run after reboot,
you can do it by adding @reboot code in cron

@reboot /opt/pentaho/biserver-ce/start-pentaho.sh

<http://askubuntu.com/questions/290099/how-to-run-a-script-during-boot-as-root>

<http://www.cyberciti.biz/faq/howto-linux-unix-start-restart-cron/>

eg: apache start up script



```
root@akhil: /home/dock/pentaho-biserver-local
ubuntu@akhil: /home/dock/pentaho-biserver-local x ubuntu@akhil: /home/dock/pentaho-biserver-local x root@akhil: /home/dock/pentaho-biserver-local x
GNU nano 2.2.6 File: /tmp/crontab.HvK05A/crontab Modified

# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow command
## * * * * ls -l /root >> /home/dock/pentaho-biserver-local/akhil

@reboot /etc/apache2/apache2 start

^G Get Help      ^O WriteOut      ^R Read File     ^Y Prev Page     ^K Cut Text      ^C Cur Pos
^X Exit          ^J Justify       ^W Where Is      ^V Next Page     ^U UnCut Text    ^T To Spell
```

here /etc/apache2/apache2 is the path to that script

<http://askubuntu.com/questions/814/how-to-run-scripts-on-start-up>

b) put service in /etc/init.d

<https://www.digitalocean.com/community/tutorials/how-to-configure-a-linux-service-to-start-automatically-after-a-crash-or-reboot-part-1-practical-examples>

<https://embraceubuntu.com/2005/09/07/adding-a-startup-script-to-be-run-at-bootup/>

```
root@akhil: /etc/init.d
ubuntu@akhil: /home/dock/pentaho-biserver-local x ubuntu@akhil: /home/dock/pentaho-biserver-local x root@akhil: /etc/init.d
root@akhil: /home/dock/pentaho-biserver-local# cd /etc/init.d/
root@akhil: /etc/init.d# ls
acpid          console-setup  halt           nmbd           rc             samba          sudo
anacron        cron           irqbalance    ndemand        rc.local       samba-ad-dc    sysstat
apache2        cups          kerneloops    pdns           rcs           saned          tomcat7
apparmor       cups-browsed  killprocs     postfix        README        sendsigs      udev
appport        dbus          kmod          pppd-dns       reboot        single         umountfs
avahi-daemon   dns-clean     lightdm        procs          resolvconf    skeleton      umountnfs.sh
binfmt-support docker         mysql          pulseaudio     rpcbind       smb           umountroot
bluetooth      friendly-recovery networking     quota          rsync         speech-dispatcher unattended-upgra
brltty         grub-common   nfs-kernel-server quotarpc        rsyslog       ssh           urandom
root@akhil: /etc/init.d#
```

copy that script to /etc/init.d/

eg : `cp -r /etc/apache2/apache2 /etc/init.d/`

jconsole monitoring

install openjdk-7-jdk or openjdk-6-jdk in server side of jconsole not in pentaho server

the jconsole can find in “/usr/lib/jvm/java-7-openjdk-amd64/bin”

`sudo jconsole >` shows a graphical window for monitoring

edit **`/opt/pentaho/biserver-ce/start-pentaho.sh`**

JMX Configuration

In order to access Mondrian metrics from a JMX client, you'll need to first configure the JVM process running biserver. For example, to set up unsecured remote JMX access on port 7199, add the following JVM parameters to the CATALINA_OPTS variable in start-pentaho.sh:

```
-Dcom.sun.management.jmxremote.port=7199
-Dcom.sun.management.jmxremote.authenticate=false
-Dcom.sun.management.jmxremote.ssl=false
```

Link:

[http://wiki.pentaho.com/display/analysis/Monitoring+Mondrian+System+Metrics+with+Java+Management+Extensions+\(JMX\)](http://wiki.pentaho.com/display/analysis/Monitoring+Mondrian+System+Metrics+with+Java+Management+Extensions+(JMX))

```
root@8f1a29c31a8d: /opt/pentaho
ubuntu@akhil: ~
GNU nano 2.2.6 File: /opt/pentaho/biserver-ce/start-pentaho.sh
rm "$DIR/promptuser.sh"
fi
if [ "$errCode" = 0 ]; then
cd "$DIR/toncat/bin"
CATALINA_OPTS="-Xms2048m -Xmx6144m -XX:MaxPermSize=256m -Dsun.rmi.dgc.client.gcInterval=3600000 -Dcom.sun.management.jmxremote.port=7199 -Dcom.sun.management.jmxremote.authenticate=false -Dcom.sun.management.jmxremote.ssl=false"
export CATALINA_OPTS
JAVA_HOME=$PENTHO_JAVA_HOME
sh startup.sh
fi
Get Help WriteOut Read File Prev Page Cut Text Cur Pos
Exit Justify Where Is Next Page UnCut Text To Spell
```

Java Monitoring & Management Console

ubuntu@akhil: ~

1986 docker attach 8f1a29c31a8d
1987 docker start 8f1a29c31a8d
1988 docker attach 8f1a29c31a8d
1989 docker ps
1990 docker attach 8f1a29c31a8d
1991 sudo jconsole
1992 sudo jconsole
1993 docker ps
1994 docker stop 8f1a29c31a8d
1995 docker ps
1996 docker ps
1997 docker start 8f1a29c31a8d
1998 docker attach 8f1a29c31a8d
1999 sudo jconsole
2000 history
ubuntu@akhil:~\$ ^C
ubuntu@akhil:~\$
ubuntu@akhil:~\$ sudo jconsole

Connection Window Help

JConsole: New Connection

New Connection

☐ Local Process:

Name	PID
sun.tools.jconsole.JConsole	27567
org.apache.catalina.startup.Bootstrap start	2938

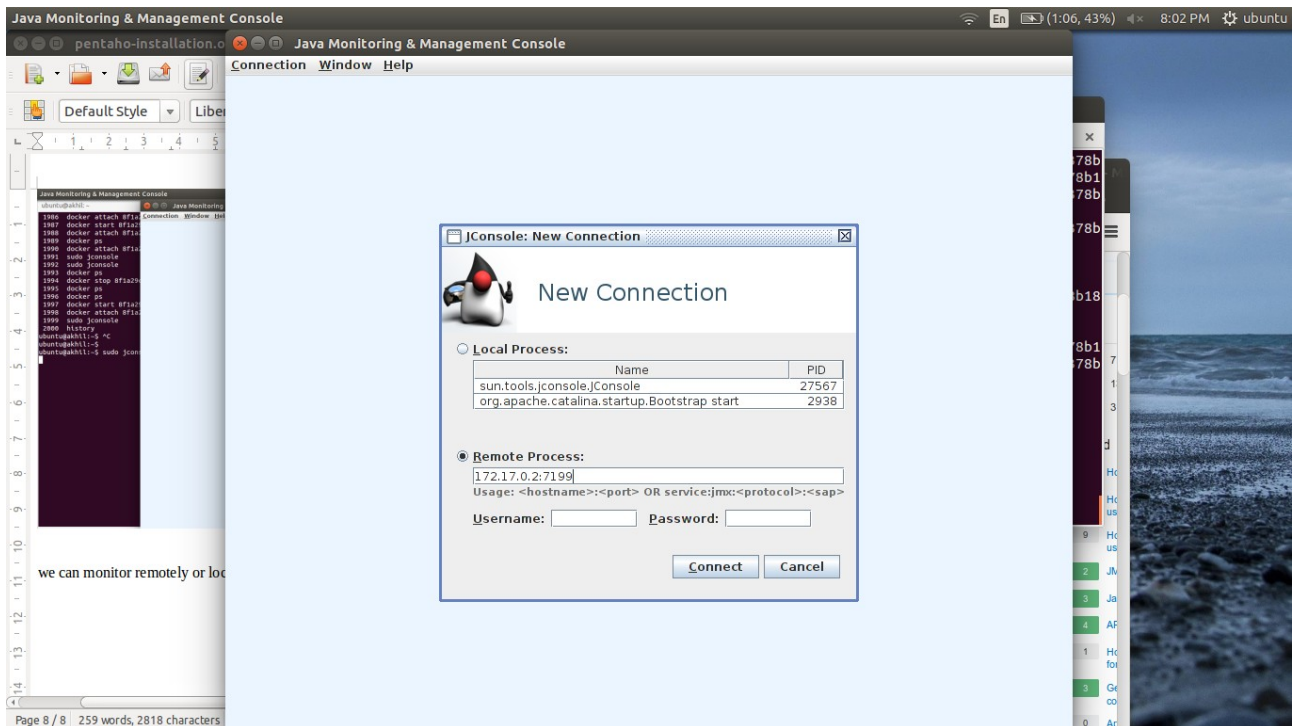
☐ Remote Process:

Usage: <hostname>:<port> OR service:jmx:<protocol>:<sap>

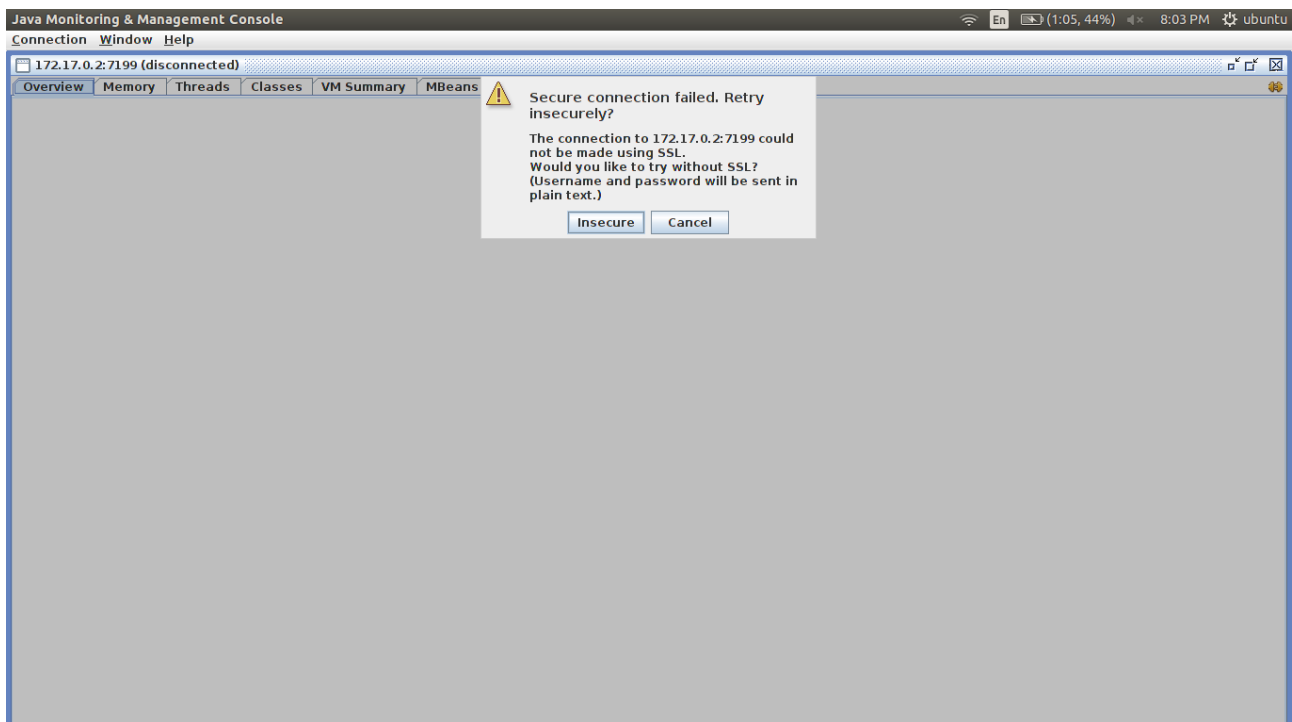
Username: Password:

Connect Cancel

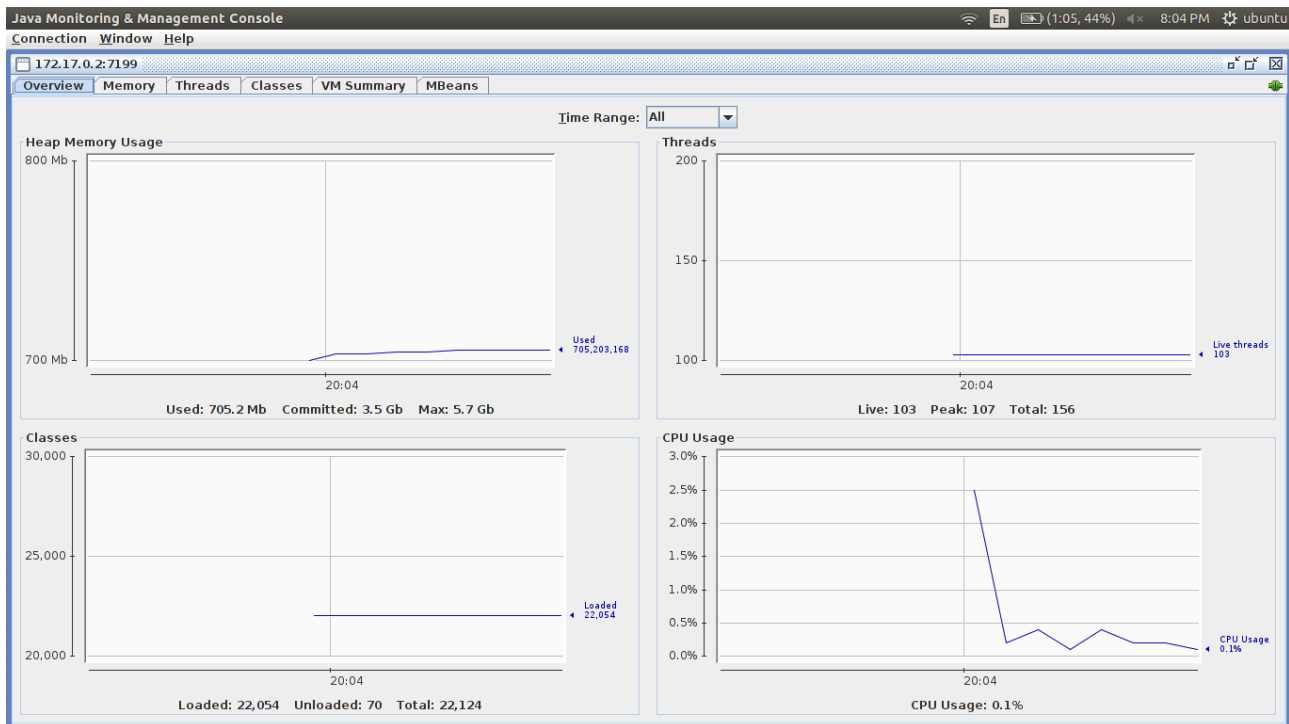
we can monitor remotely or locally



here i use remote because our pentaho running remotly 172.17.0.2 and i provide a port 7199 for jconsole accessing.



Click on insecure if configuration in pentaho is insecure



jconsole manual testing:
download a jar file and run it with

```
java -Dcom.sun.management.jmxremote \  
-Dcom.sun.management.jmxremote.port=9010 \  
-Dcom.sun.management.jmxremote.local.only=false \  
-Dcom.sun.management.jmxremote.authenticate=false \  
-Dcom.sun.management.jmxremote.ssl=false \  
-Djava.rmi.server.hostname=192.168.1.234 \  
-jar Notepad.jar
```

here i use **Notepad.jar** for testing,

so in jconsole provide 192.168.1.234:9010 so it will connected.

Links::>>>>

<http://stackoverflow.com/questions/856881/how-to-activate-jmx-on-my-jvm-for-access-with-jconsole>

<http://stackoverflow.com/questions/2591516/why-has-it-failed-to-load-main-class-manifest-attribute-from-a-jar-file>

<http://www.oracle.com/technetwork/articles/java/jconsole-1564139.html>

http://www.wellho.net/mouth/2081_Connecting-jconsole-remotely-the-principles.html

CATALINA_OPTS parameters

-Xmx for maximum heap size, and

-Xms for initial heap size

Increase maximum perm size for web base applications to 4x the default amount

-XX:MaxPermSize

eg: -XX:PermSize=128m

-XX:MaxPermSize=512m

6 Common Errors in Setting Java Heap Size

Two JVM options are often used to tune JVM heap size: -Xmx for maximum heap size, and -Xms for initial heap size. Here are some common mistakes I have seen when using them:

- Missing m, M, g or G at the end (they are case insensitive). For example,

```
java -Xmx128 BigApp
java.lang.OutOfMemoryError: Java heap space
```

The correct command should be: `java -Xmx128m BigApp`. To be precise, -Xmx128 is a valid setting for very small apps, like HelloWorld. But in real life, I guess you really mean -Xmx128m

- Extra space in JVM options, or incorrectly use =. For example,

```
java -Xmx 128m BigApp
Invalid maximum heap size: -Xmx
Could not create the Java virtual machine.
java -Xmx=512m HelloWorld
Invalid maximum heap size: -Xmx=512m
Could not create the Java virtual machine.
```

The correct command should be `java -Xmx128m BigApp`, with no whitespace nor =. -X options are different than -Dkey=value system properties, where = is used.

- Only setting -Xms JVM option and its value is greater than the default maximum heap size, which is 64m. The default minimum heap size seems to be 0. For example,

```
java -Xms128m BigApp
Error occurred during initialization of VM
Incompatible initial and maximum heap sizes specified
```

The correct command should be `java -Xms128m -Xmx128m BigApp`. It's a good idea

to set the minimum and maximum heap size to the same value. In any case, don't let the minimum heap size exceed the maximum heap size.

- Heap size is larger than your computer's physical memory. For example,

```
java -Xmx2g BigApp
Error occurred during initialization of VM
Could not reserve enough space for object heap
Could not create the Java virtual machine.
```

The fix is to make it lower than the physical memory: `java -Xmx1g BigApp`

- Incorrectly use mb as the unit, where m or M should be used instead.

```
java -Xms256mb -Xmx256mb BigApp
Invalid initial heap size: -Xms256mb
Could not create the Java virtual machine.
```

- The heap size is larger than JVM thinks you would ever need. For example,

```
java -Xmx256g BigApp
Invalid maximum heap size: -Xmx256g
The specified size exceeds the maximum representable size.
Could not create the Java virtual machine.
```

The fix is to lower it to a reasonable value: `java -Xmx256m BigApp`

- The value is not expressed in whole number. For example,

```
java -Xmx0.9g BigApp
Invalid maximum heap size: -Xmx0.9g
Could not create the Java virtual machine.
```

The correct command should be `java -Xmx928m BigApp`

links : >>>

<http://javahowto.blogspot.in/2006/06/6-common-errors-in-setting-java-heap.html>

<https://gist.github.com/terrancesnyder/986029>

<https://www.experts-exchange.com/questions/23159662/Difference-between-CATALINA-OPTS-and-JAVA-OPTS-in-TOMCAT.html>