# AVR2 Installation Instructions

# Introduction

This document serves as documentation to build out the AVR2 datawarehouse using Pentaho5 and PostgreSQL 9.2. The documentation assumes you already have an instance of PostgreSQL running with a copy of the populated Trisano database schema on it.

# Prerequsites

These instructions assume you are running AVR2 on a CentOS Linux system or other Linux derivative. The software stack for AVR2 is 100% Java and can run on any JVM platform.

* Java 7 (Open JDK 1.7.0\_\*). AVR2 was developed and tested on 1.7.0\_51, although any 1.7 version of Java should work. Note: Many of the Pentaho BI tools are packaged with their own JVM.
* Access to the uniphi server (to retrieve software)
* Pentaho 5

# Download Necessary Files and Unzip Archives

- From uniphi (uniphi.chpc.utah.edu), scp all files in /var/local/export to the intended target system

* move pdi.tgz and pentaho.tgz to /var/local
* sudo tar -xzvf pdi.tgz
* sudo tar -xzvf pentaho.tgz

# Create Pentaho Linux User and Working Environment

# sudo groupadd pentaho

# sudo useradd pentaho -g pentaho

# sudo chown -R pentaho /var/local/pentaho

# sudo chown -R pentaho /var/local/pentaho

# sudo chgrp -R pentaho /var/local/pentaho

# sudo chown -R pentaho /var/local/pdi-ce-5.0.1-stable

# sudo chgrp -R pentaho /var/local/pdi-ce-5.0.1-stable

# sudo ln -s pdi-ce-5.0.1-stable/ pdi

# AVR2 GIT Project

# cd /var/local

# sudo mkdir project

# sudo chown pentaho project

# sudo chgrp pentaho project

# cd project

# git clone https://github.com/UTDOH/AVR.git

# Configure PDI

# sudo su pentaho

# cd ~

# ls -alt

# (if .kettle doesn't exist, create it)

# cd .kettle

# vi kettle.properties

# add the line: pentaho.solutions.root=/var/local/project/AVR/pentaho/datawarehouse (replace /var/local/project with the absolute path for the installtion environment)

# Database Setup

Login to postgresql and enter the following commands:

postgres=# create user pentaho with password '\*\*\*\*\*'; (replace the \*\*\*\*\*\* with whatever the password should be)

postgres=# create user hibuser with password '\*\*\*\*\*'; (replace the \*\*\*\*\*\* with whatever the password should be)

postgres=# create database hibernate;

CREATE DATABASE

postgres=# grant all privileges on database hibernate to hibuser;

postgres=# \c hibernate

postgres=# grant all on all tables in schema public to hibuser;

GRANT

postgres=# create database jackrabbit;

CREATE DATABASE

postgres=# create user jcr\_user with password '\*\*\*\*\*'; (replace the \*\*\*\*\*\* with whatever the password should be)

postgres=# grant all privileges on database jackrabbit to jcr\_user;

postgres=# alter database jackrabbit owner to jcr\_user;

postgres=# \c jackrabbit

postgres=# grant all on all tables in schema public to jcr\_user;

GRANT

postgres=# create database pentaho;

CREATE DATABASE

postgres=# grant all privileges on database pentaho to pentaho;

postgres=# \c pentaho

postgres=# grant all on all tables in schema public to pentaho;

GRANT

postgres=# create user pentaho\_user with password '\*\*\*\*\*'; (replace the \*\*\*\*\*\* with whatever the password should be)

postgres=# create database quartz;

CREATE DATABASE

postgres=# grant all privileges on database quartz to pentaho\_user;

postgres=# \c quartz

Kettle / PDI config

- sudo su pentaho

- cd ~

- (if .kettle doesn't exist, create it)

- cd .kettle

- vi kettle.properties

- add the line: pentaho.solutions.root=/var/local/project/AVR/pentaho/datawarehouse

- (replace /var/local/project with the absolute path for the installtion environment)postgres=# grant all on all tables in schema public to pentaho\_user;

GRANT

postgres=# create database trisano\_olap;

CREATE DATABASE

postgres=# grant all privileges on trisano\_olap to pentaho;

postgres=# \c trisano\_olap

trisano\_olap=# create schema warehouse;

postgres=# grant all on all tables in schema warehouse to pentaho;

trisano\_olap=# create schema warehouse\_util;

postgres=# grant all on all tables in schema warehouse\_util to pentaho;

GRANT

\q

# Restore Database Schemas

cd /var/local/import

sudo -u postgres psql -d hibernate -h 127.0.0.1 < hibernate.backup

sudo -u postgres psql -d jackrabbit -h 127.0.0.1 < jackrabbit.backup

sudo -u postgres psql -d pentaho -h 127.0.0.1 < pentaho.backup

sudo -u postgres psql -d quartz -h 127.0.0.1 < quartz.backup

# JNDI / JDBC Configuration

Now the database schemas and users have been setup, the configuration for each JNDI and JDBC connection can be made for both Pentaho server as well as Pentaho Data Integration

## PDI Configuration

Open /var/local/pdi/simple-jndi/jdbc.properties

**Connection**: WebTrisanoTXDS

Update the url, user, and password to match the configuration for the OLTP (Trisano) database

**Connection**: TrisanoOLAPDS

Update the url, user, and password to match the configuration for the OLAP database (trisano\_olap above).

# cd /var/log/

# sudo mkdir pdi

# sudo chown pentaho pdi

# sudo chgrp pentaho pdi

## Pentaho Configuration

Database: hibernate

Open /var/local/pentaho/pentaho-solutions/system/hibernate/postgresql.hibernate.cfg.xml

* Update the connection.url to match the host, port and database name for the hibernate database
* Update the connection.username and connection.password to match what was set in the Database Setup section.

Open /var/local/pentaho/tomcat/webapps/pentaho/META-INF/context.xml and update each JNDI connection to match what was configured in the “Database Setup” section.

# Pentaho Server Logging Setup

cd /var/log

sudo mkdir pentaho

sudo chown -R pentaho pentaho

sudo chgrp -R pentaho pentaho

cd /var/local/pentaho/tomcat/

sudo mv logs logs\_prev

sudo ln -s /var/log/pentaho/ logs

# Apache AJP ProxyPass Setup

These instructions specify how to setup the AJP proxypass in Apache2. The instructions may vary slightly due to the installation of Apache. Use uniphi as the reference implementation.

cd /var/local/pentaho/tomcat/conf/

sudo vi server.xml

Ensure the AJP connector is configured properly:

e.g. <Connector URIEncoding="UTF-8" port="8010" protocol="AJP/1.3" redirectPort="8543" />

Save server.xml

cd /etc/apache2/mods-enabled

sudo ln -s ../mods-available/proxy\_ajp.load proxy\_ajp.load

sudo vi sudo vi proxy.conf

Add the following proxy definition:

<Location /pentaho/>

ProxyPass ajp://localhost:8010/pentaho/

</Location>

\*\* Update the AJP port to match the AJP port specified in the tomcat server.xml

# Build / Rebuild Datawarehouse

The ETL scripts for AVR2 are built to both build the datawarehouse from scratch and / or pick up where they last left off “upserting” records to the datawarehouse facts and dimensions

cd /var/local/pdi

./kitchen.sh -file /var/local/project/AVR/pentaho/datawarehouse/etl/load\_data\_warehouse\_main.kjb > /var/log/pdi/load\_data\_warehouse\_main.log 2>&1

The aforementioned script can be placed into the pentaho user's crontab for regular rebuilding of the datawarehouse data.

**Please note**: The 2>&1 is important to include as in Pentaho 5 Data Integration, errors (including database errors) are redirected to stderr. If the scripts are executed from crontab, stderr will redirect to the crontab user's session and not the intended log file.