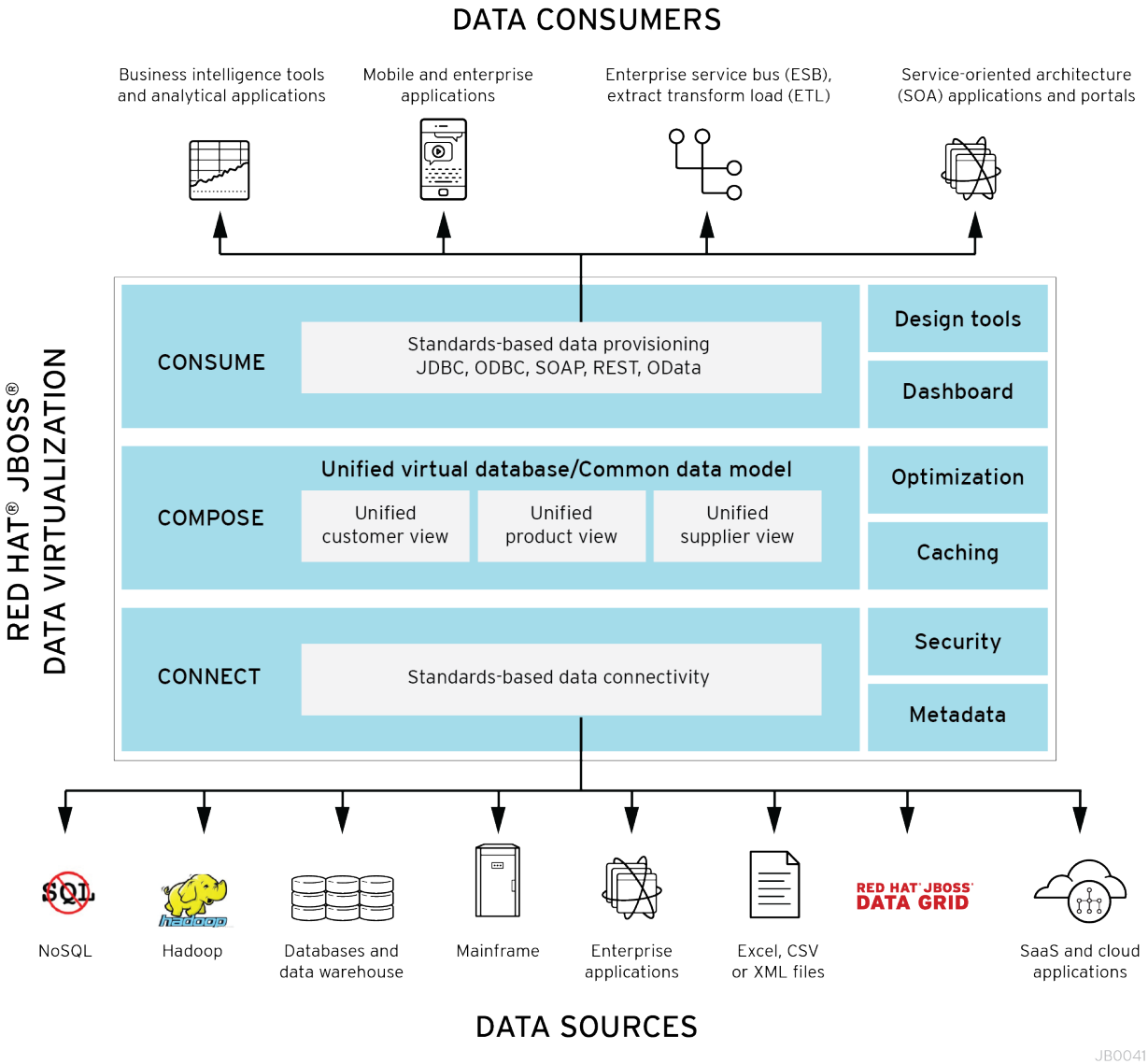
Lab 0. Introduction

Red Hat JBoss Data Virtualization, formerly known as Red Hat JBoss Enterprise Data Services Platform (EDS), is a complete data provisioning, federation, integration and management solution that enables organizations to gain actionable and unified information. Red Hat JBoss Data Virtualization enables agile data utilization in three easy steps:

1. Connect: Access data from multiple, heterogeneous data sources.
2. Compose: Easily create reusable, business-friendly logical data models and views by combining and transforming data.
3. Consume: Make unified data easily consumable through open standard interfaces.



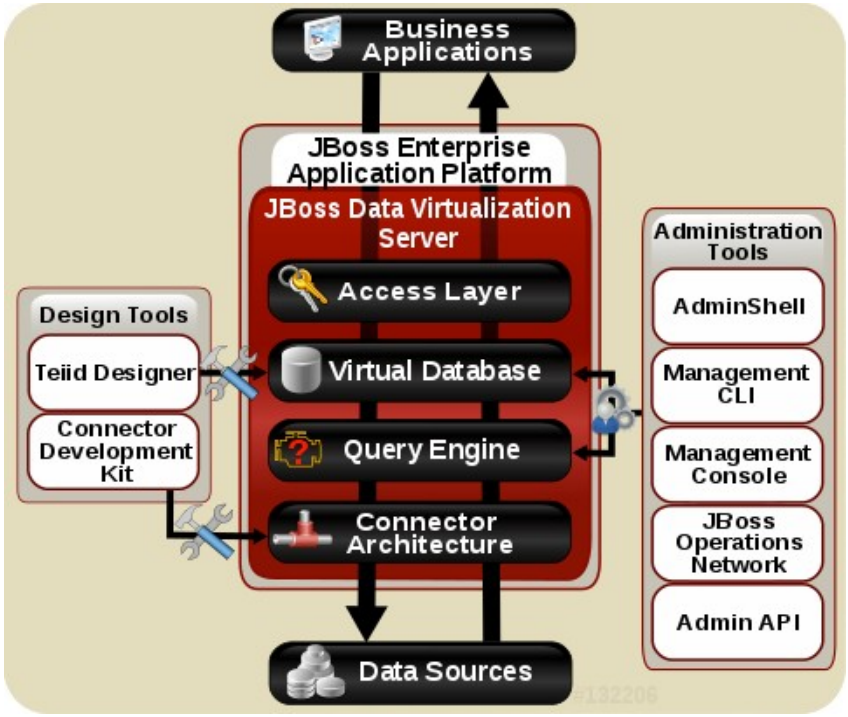
JBoss Data Virtualization includes:

- Tools for creating data views that are accessible through standard protocols (the Teiid Designer plug-in for JBoss Developer Studio (JBDS)).
- A robust runtime environment that provides enterprise-class performance, data integrity, and security (the JBoss Data Virtualization Server, which executes as a process within the JBoss Enterprise Application Platform (EAP)).
- A repository for storing metadata (ModeShape)

JBoss Data Virtualization is based on the following community projects:

- Teiid (http://www.jboss.org/teiid)
- Teiid Designer (http://www.jboss.org/teiiddesigner)
- Modeshape (http://www.jboss.org/modeshape)

The figure below depicts the architectural overview of JBoss Data Virtualization:



Query Engine	The heart of JBoss Data Virtualization Server is a high-performance query engine that processes relational, XML, XQuery and procedural queries from federated datasources. Features include support for homogeneous schemas, heterogeneous schemas, transactions, and user defined
Embedded	An easy-to-use JDBC Driver that can embed the Query Engine in any Java application.
Server	An enterprise ready, scalable, manageable, runtime for the Query Engine that runs inside JBoss EAP that provides additional security, fault-tolerance, and administrative features.

Connectors	JBoss Data Virtualization Server includes a rich set of Translators and Resource Adapters that enable access to a variety of sources, including most relational databases, web services, text files, and ldap. Need data from a different source? A custom translators and resource adaptors can easily be developed.
Tools	• **Create** - Use Teiid Designer to define virtual databases containing views, procedures or even dynamic XML documents. • **Monitor & Manage** - Use the Management Console with JBoss EAP or use the JBoss Data Virtualization JBoss Operations Network (JON) plugin to control any number of servers. • **Script** - Use the AdminShell to automate administrative and testing tasks.

0.1 What is expected of you

Please feel free to raise your hands with any questions that you have about the lab; feel free to ask why it is you are doing something, or if something does not feel right. Please know that all care was made in creating this user guide, but all screen shots and steps along the way might be off by just a little so please be patient with any issues.

0.2 Prerequisites

Oracle JDK 1.6, 1.7 or OpenJDK 1.6 or 1.7

PostgreSQL server and/or MySQL server running

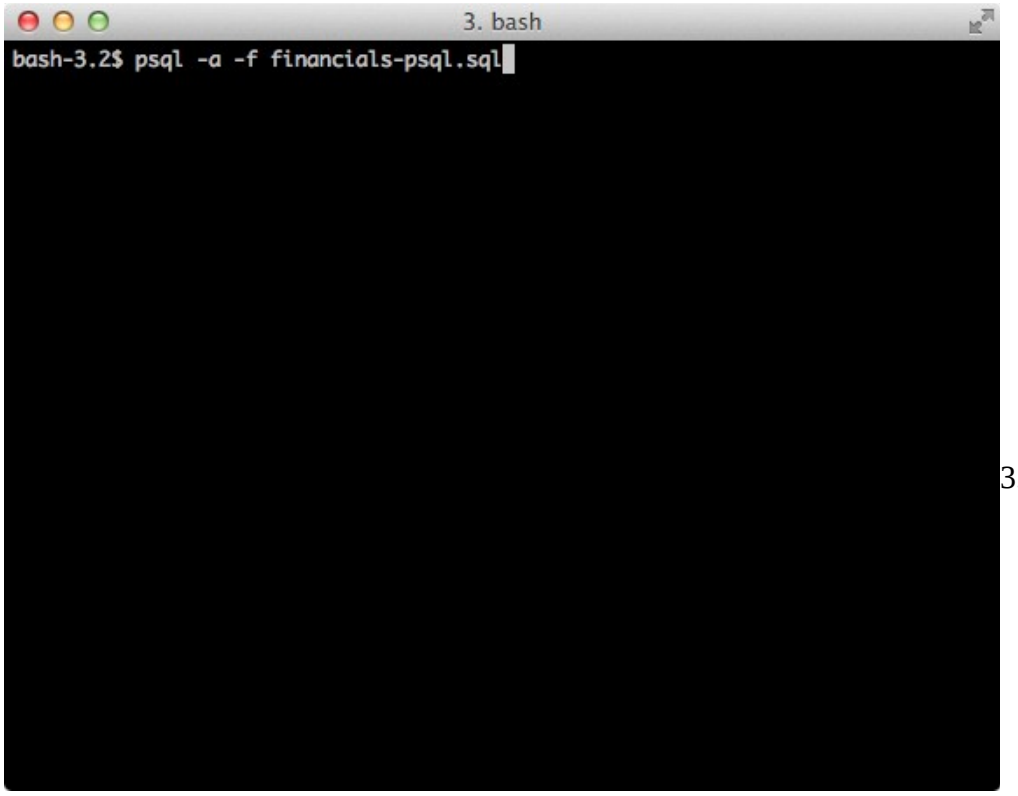
PostgreSQL and/or MySQL client tools installed if using a remote database

0.3 Setup demo environment

In order to use the labs we need to prepare the database. We provide database SQL scripts for PostgreSQL and MySQL to load initial data into the database. The next paragraph will describe how to install the database and load the demo data into the database.

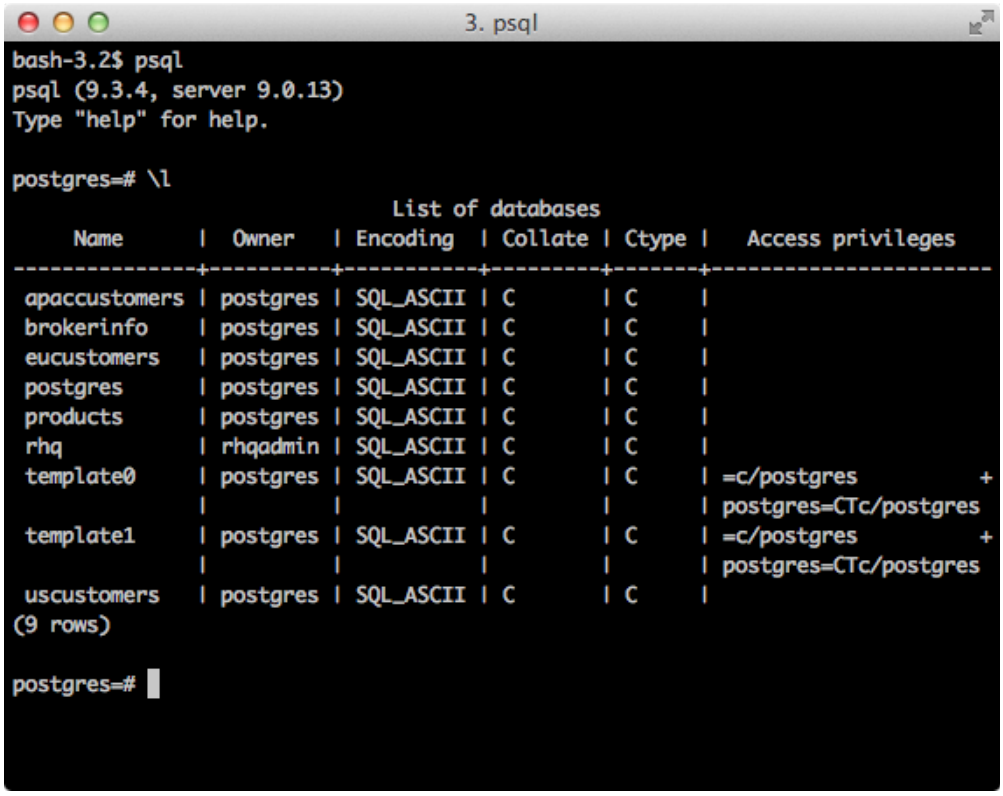
0.3.1 PostgreSQL

- Installation
  The easiest way to install PostgreSQL is to use the pre-built binary packages which are available for a number of operating systems. See http://www.postgresql.org/download/ for more information and downloads.
- Post-install steps
  1. If *nix or MacOSX switch to user postgres (Hint: su postgres)
  2. Go to the DVWorkshop/dv-docker/demo directory and tun the following command as depicted in the picture below.



3.

3. If Step 2 is successfully executed the the PostgreSQL environment contains the following databases. Hint: start the psql command line utility and type the “\l” to list the databases in PostgreSQL database.



Type “\q” to exit the postgres=# shell.

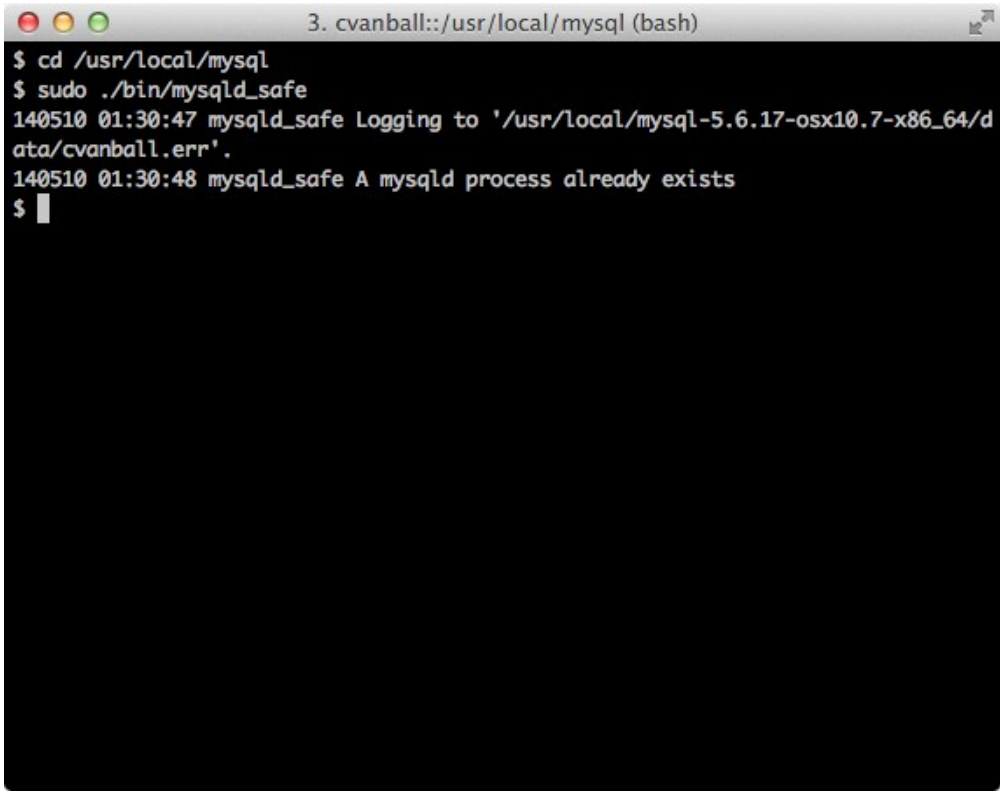
The labs will use the following databases:

- apaccustomer
- eucustomers
- uscustomers

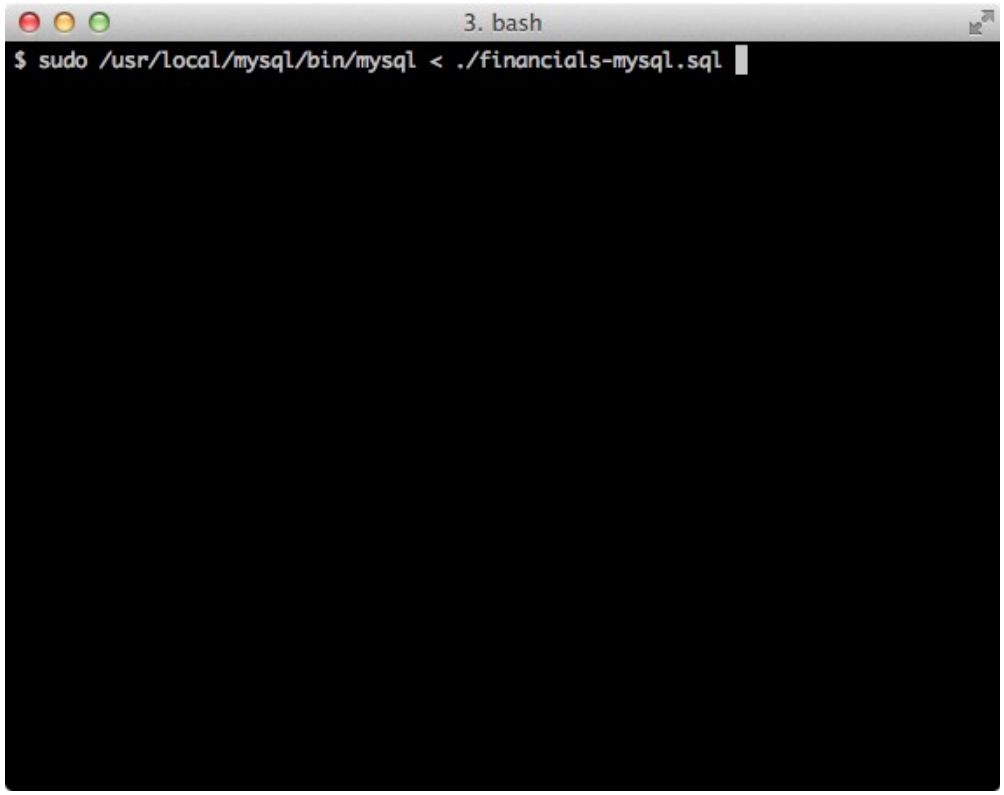
0.3.2 MySQL

- Installation
  The easiest way to install MySQL is to use the pre-built binary packages which are available for a number of operating systems. See http://dev.mysql.com/downloads/mysql/ for more information and downloads.

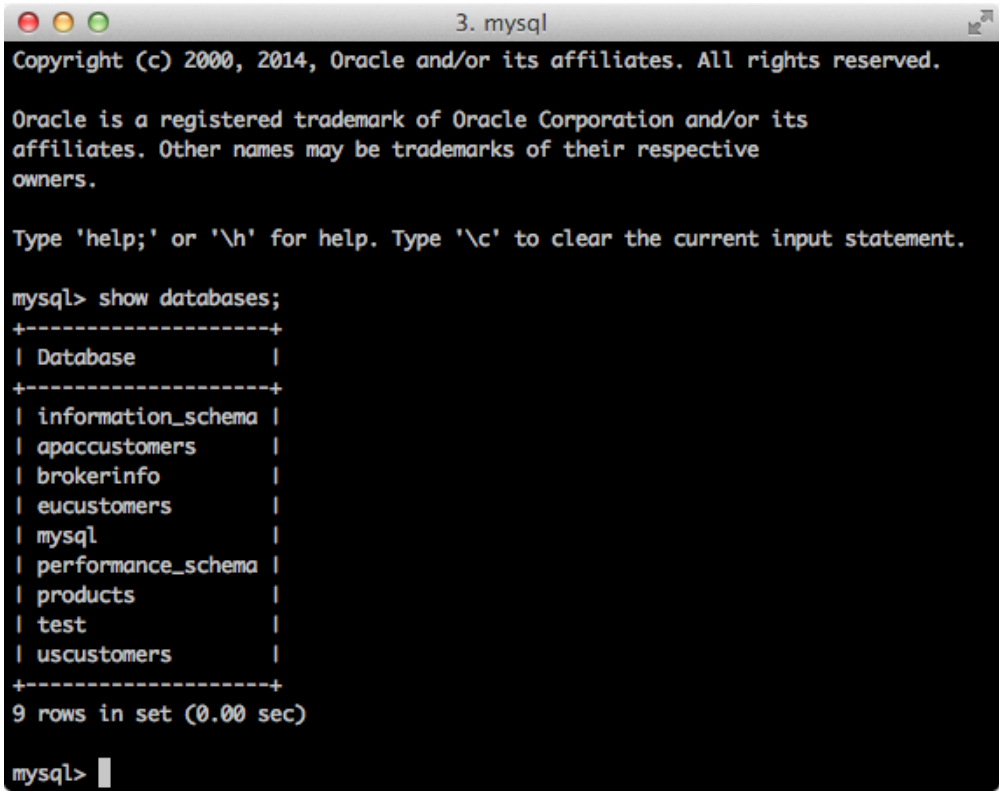
- Post-install steps
  1. If *nix or MacOSX go to the /usr/local/mysql directory and start mysqld_safe



  2. Go to the DVWorkshop/dv-docker/demo directory and tun the following command as depicted in the picture below.



3. If Step 2 is successfully executed the the MySQL environment contains the following databases. Hint: start the mysql command line utility and type the “show databases;” to list the databases in MySQL database.



Type “exit” to exit the mysql> shell.

Congratulations, you can now proceed with Lab 1.