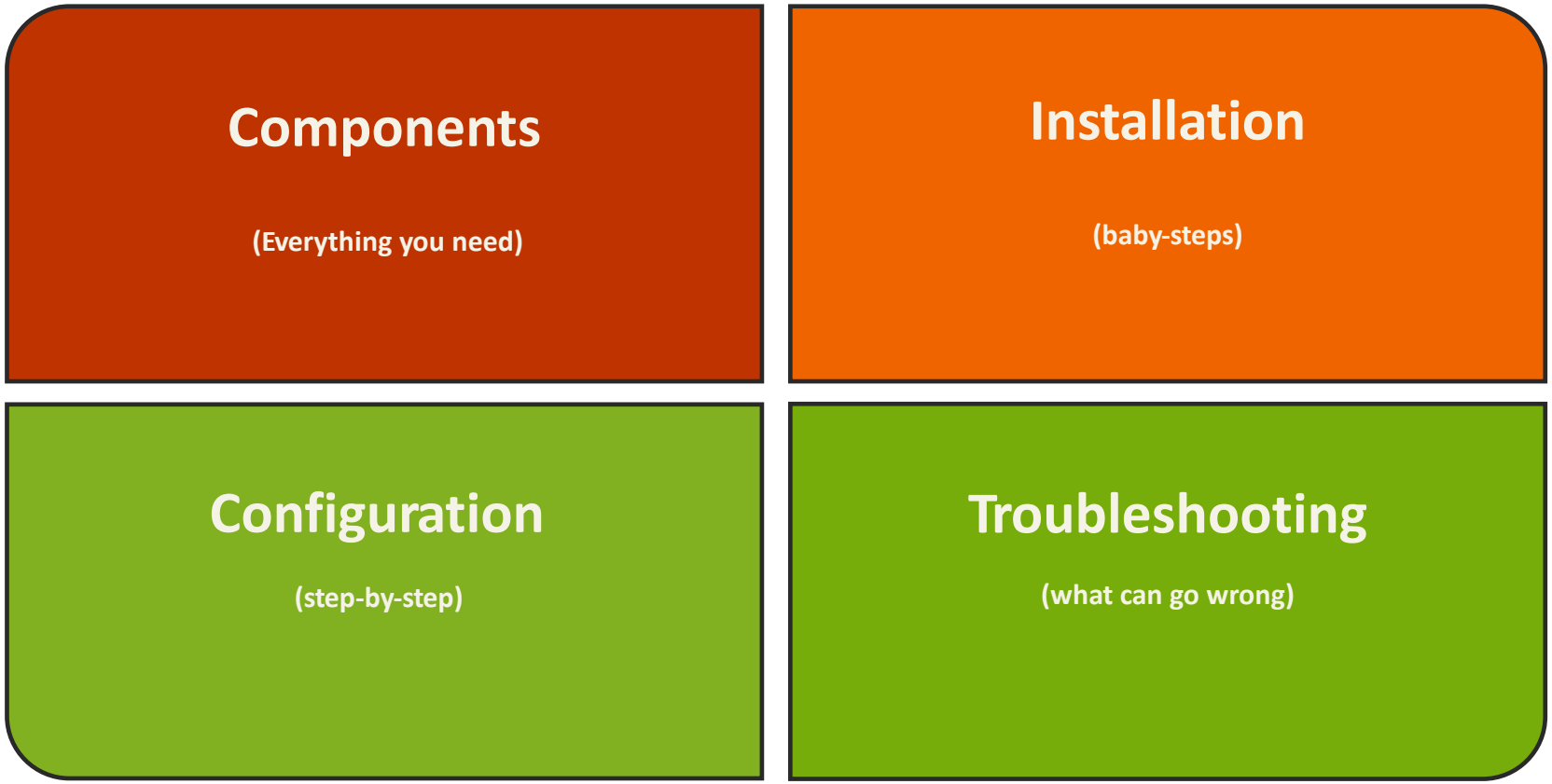
Troubleshooting Java Connection to MySQL

Rodrigo Trindade
Service Delivery Manager

Talk Topics



Timeline History

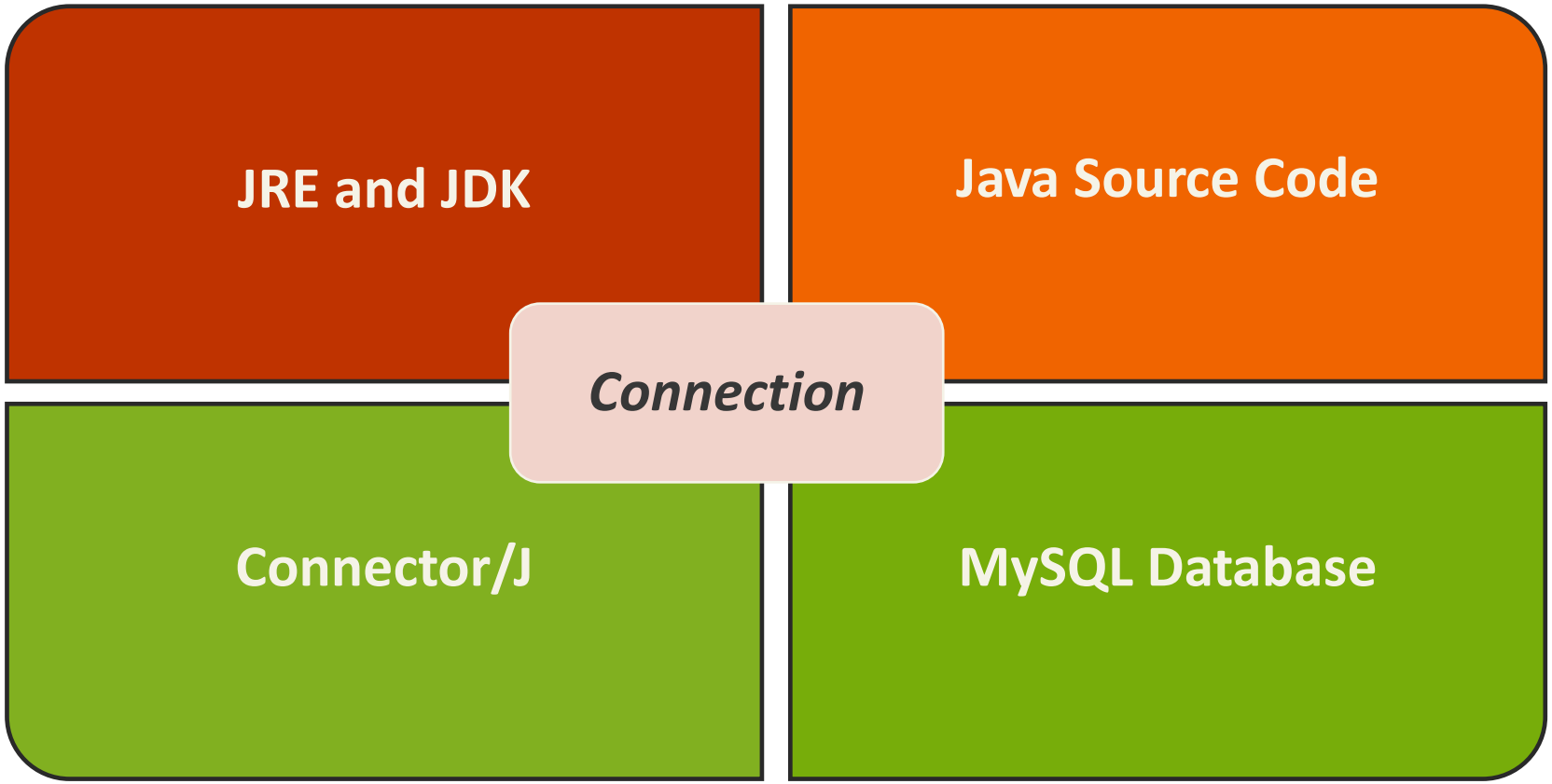
Universities - Companies:

- Certifications
- Brazil 2 USA

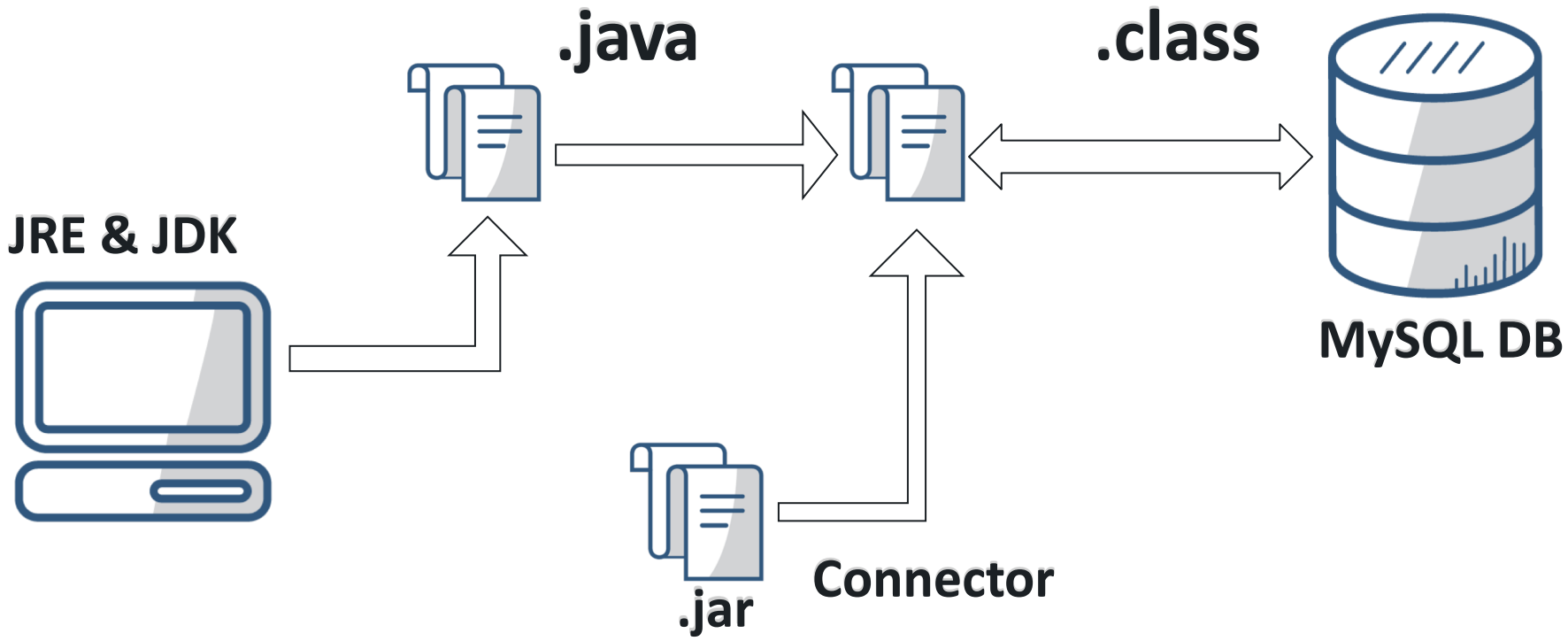
Components

Everything you need to get Java connection to MySQL

Components



OS Sandbox



Installation

Detailed steps for Components Installation

Installation

1- Install CentOS VMs

https://www.virtualbox.org/

https://www.vagrantup.com/downloads.html

Vagrant file

```
# -*- mode: ruby -*-
# vi: set ft=ruby :
Vagrant.configure(2) do |config|
  config.vm.define "node-1" do |node|
    node.vm.box = "centos/7"
    node.vm.host_name = "node1"
    node.vm.network "private_network", ip: "192.168.33.10"
  end
end
Vagrant.configure(2) do |config|
  config.vm.define "jvm-1" do |node|
    node.vm.box = "centos/7"
    node.vm.host_name = "jvm1"
    node.vm.network "private_network", ip: "192.168.33.11"
  end
end
```

VMs Installation

$ vagrant up

```
Bringing machine 'node-1' up with 'virtualbox' provider...
==> node-1: Checking if box 'centos/7' is up to date...
(...)
==> node-1: Booting VM...
==> node-1: Waiting for machine to boot. This may take a few minutes...
(...)
==> node-1: Machine booted and ready!
(...)
==> node-1: Setting hostname...
==> node-1: Configuring and enabling network interfaces...
    node-1: SSH address: 127.0.0.1:2222
    node-1: SSH username: vagrant
    node-1: SSH auth method: private key
```

VMs Installation

```
Bringing machine 'jvm-1' up with 'virtualbox' provider...
==> jvm-1: Importing base box 'centos/7'...
==> jvm-1: Matching MAC address for NAT networking...
==> jvm-1: Checking if box 'centos/7' is up to date...
==> jvm-1: Setting the name of the VM: CentOS7_jvm_1542309057193_62910
(...)
==> jvm-1: Booting VM...
==> jvm-1: Waiting for machine to boot. This may take a few minutes...
(...)
==> jvm-1: Machine booted and ready!
(...)
==> jvm-1: Setting hostname...
==> jvm-1: Configuring and enabling network interfaces...
    jvm-1: SSH address: 127.0.0.1:2200
    jvm-1: SSH username: vagrant
    jvm-1: SSH auth method: private key
```

Installation

2- Install Percona Server following these steps.

After booting VMs successfully, run the following to connect to node-1:

$ vagrant ssh node-1

Percona Server Installation

```
[root@node1 ~]# yum install http://www.percona.com/downloads/percona-release/redhat/0.1-6/percona-release-0.1-6.noarch.rpm
Loaded plugins: fastestmirror
 (...)
Is this ok [y/d/N]: y
Downloading packages:
(...)
Running transaction
  Installing : percona-release-0.1-6.noarch
  Verifying  : percona-release-0.1-6.noarch
 Installed:
  percona-release.noarch 0:0.1-6
 Complete!
```

Percona Server Installation

```
[root@node1 ~]# yum install Percona-Server-server-57
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
(...)
Resolving Dependencies
--> Running transaction check
---> Package Percona-Server-server-57.x86_64 0:5.7.23-24.1.el7 will be
installed
(...)
Installed:
Percona-Server-server-57.x86_64 0:5.7.23-24.1.el7
Percona-Server-shared-compat-57.x86_64 0:5.7.23-24.1.el7
(...)
 Complete!
```

Percona Server Installation

```
[root@node1 ~]# systemctl start mysql
[root@node1 ~]# ps -ef | grep mysqld
mysql      23699      1 18 20:55 ?        00:00:01 /usr/sbin/mysqld --daemonize --pid-file=/var/run/mysqld/mysqld.pid
[root@node1 ~]# grep -i 'password' /var/log/mysqld.log
2018-11-15T20:55:36.680897Z 1 [Note] A temporary password is generated for root@localhost: ;s_ZcHwfR5r1
```

Percona Server Installation

```
[root@node1 ~]# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.7.23-24
(...)
mysql> set PASSWORD = 'Passw0rd!';
Query OK, 0 rows affected (0.00 sec)
mysql> exit
Bye
```

Installation

3- Install Java Open JDK:

$ yum install java-1.8.0-openjdk-devel.x86_64

$ javac -version
javac 1.8.0_191

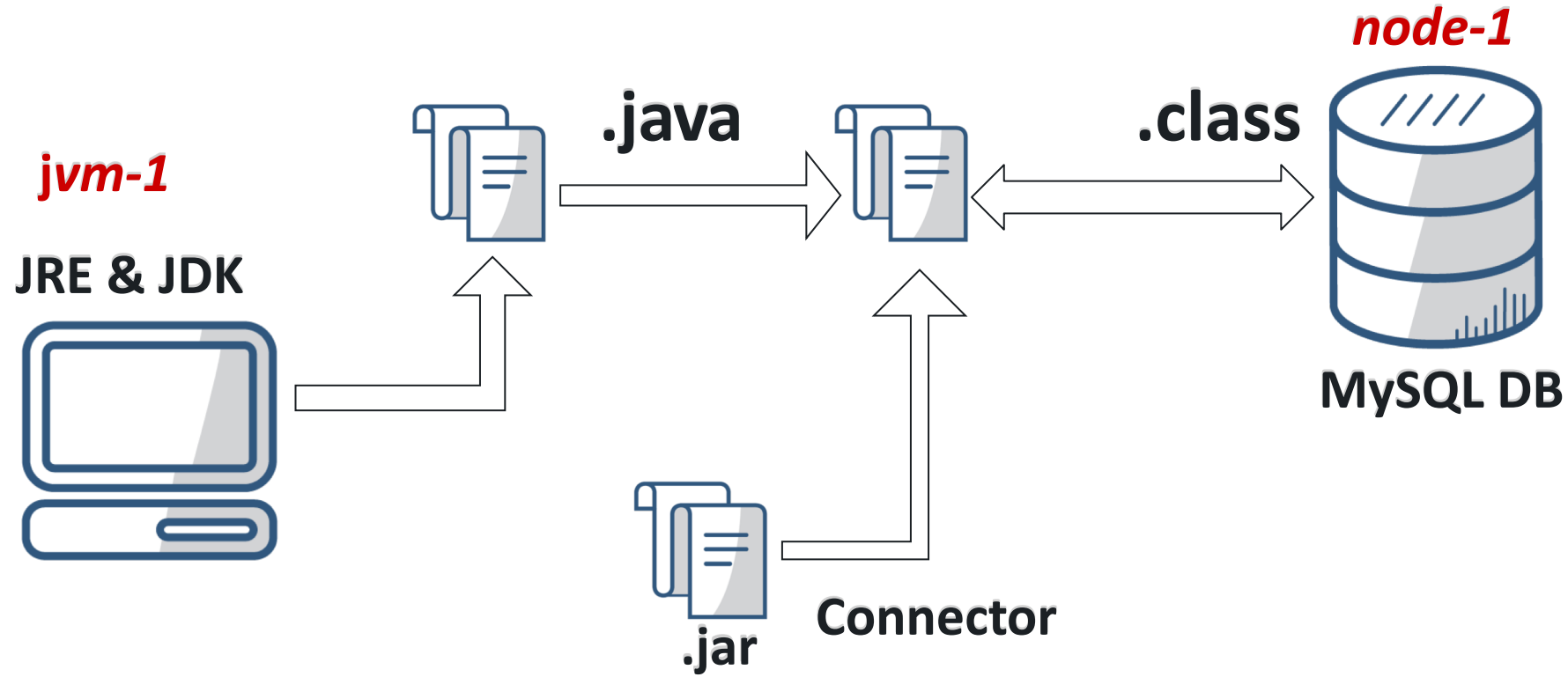
Installation

4 - Download *mysql-connector.jar*

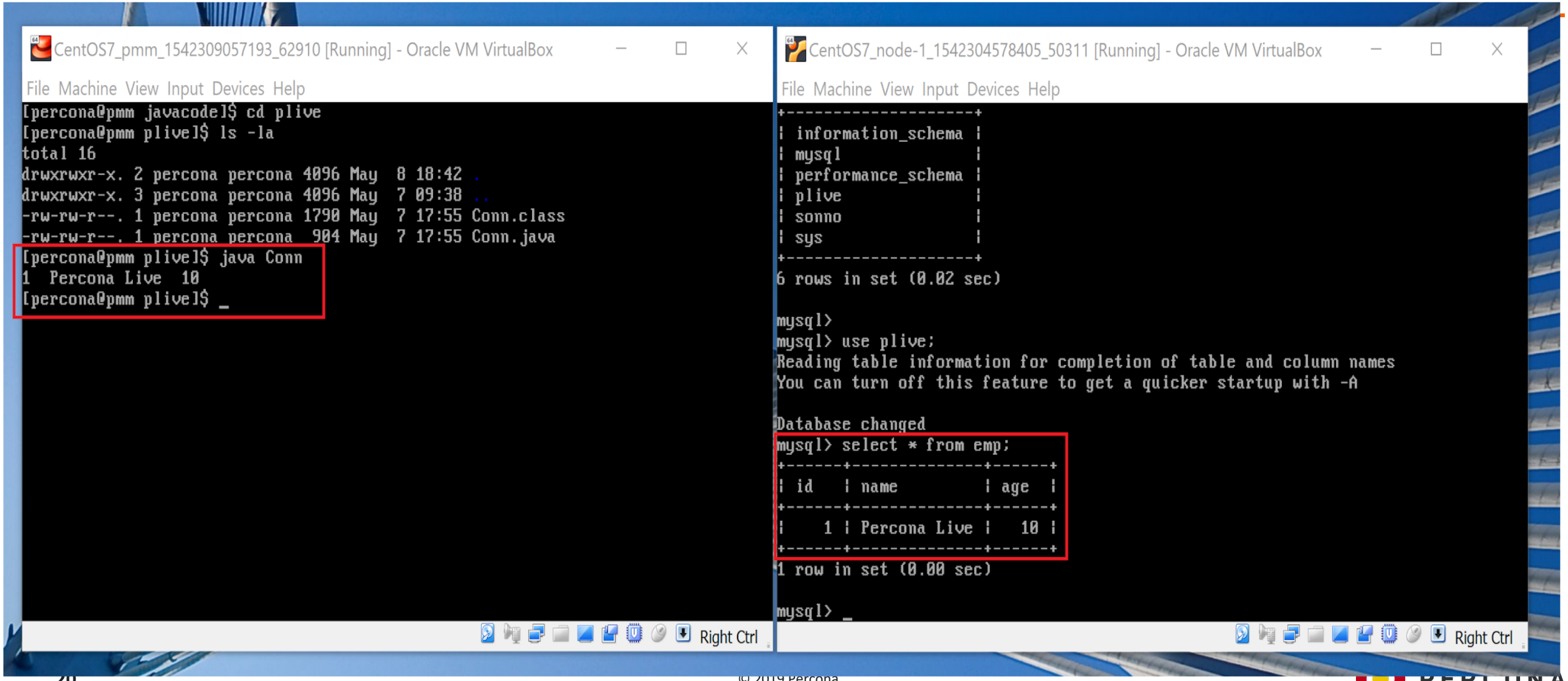
https://www.mysql.com/products/connector/

JDBC Driver for MySQL (Connector/J)

Sandbox Installed (open source)



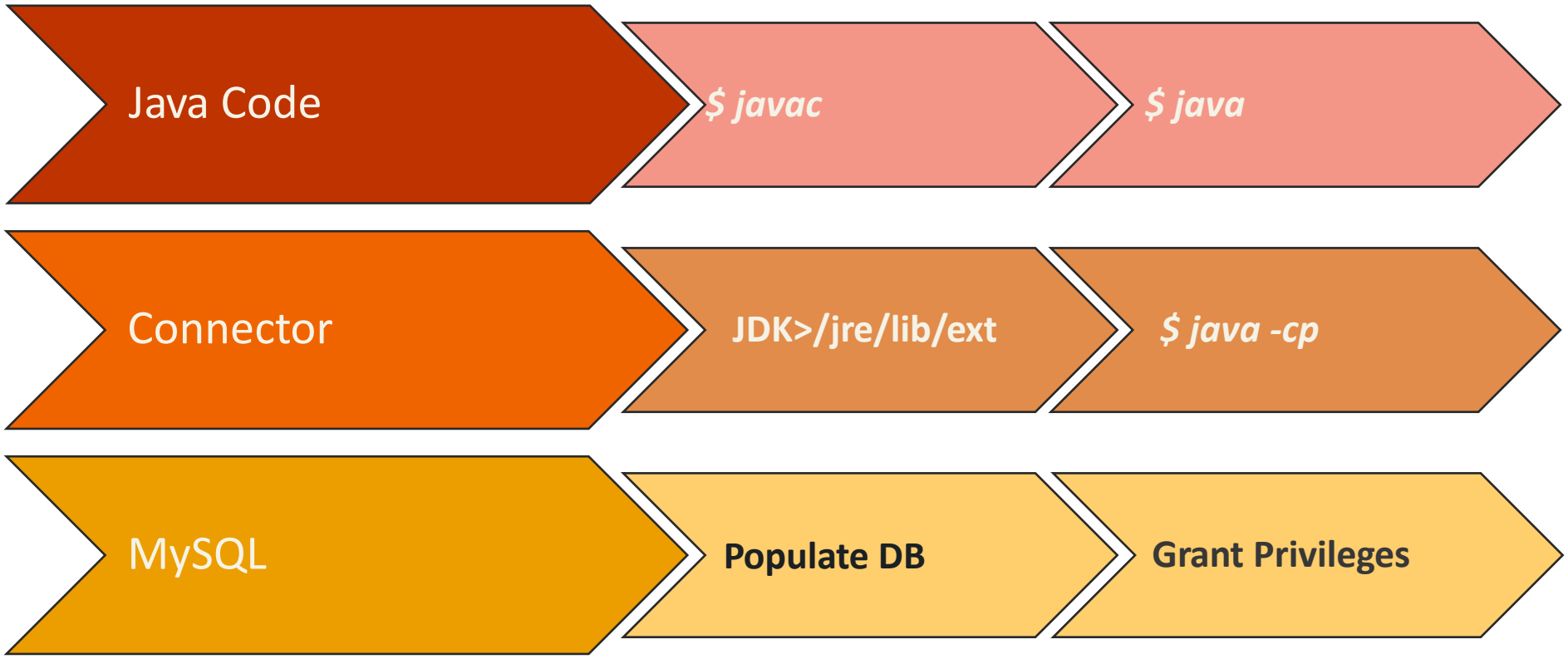
Sandbox Snapshot



Configuration

Details on Configuration of the Components

Configuration



MySQL JDBC Connector

JDBC provides a standard interface for interacting with any RDBMS.

The API consists of the following 4 main components:

1. **JDBC Driver**
2. **Connection**
3. **Statement**
4. **ResultSet**

Java Source code *Conn.java*

```
import java.sql.*;
class Conn {
private static String Driver = "com.mysql.jdbc.Driver";
private static String ConnectionURL = "jdbc:mysql://192.168.33.10:3306/plive";
private static String User = "root";
private static String Password = "Passw0rd!";
private static String SQL_Statement = "SELECT * FROM emp";

public static void main(String args[]){
try {
Class.forName(Driver);
Connection con=DriverManager.getConnection(ConnectionURL,User,Password);
Statement stmt=con.createStatement();
ResultSet rs=stmt.executeQuery(SQL_Statement);

while(rs.next())
System.out.println(rs.getInt(1)+"  "+rs.getString(2)+"  "+rs.getString(3));
con.close();
}
catch(Exception e){ System.out.println(e);}
}
}
```

Java Source code

1. *Class.forName(Driver);*
2. *Connection con=DriverManager.getConnection(ConnectionURL,User,Password);*
3. *Statement stmt=con.createStatement();*
4. *ResultSet rs=stmt.executeQuery(SQL_Statement);*

Java Source code

Where does that come from???

Class.forName(Driver);

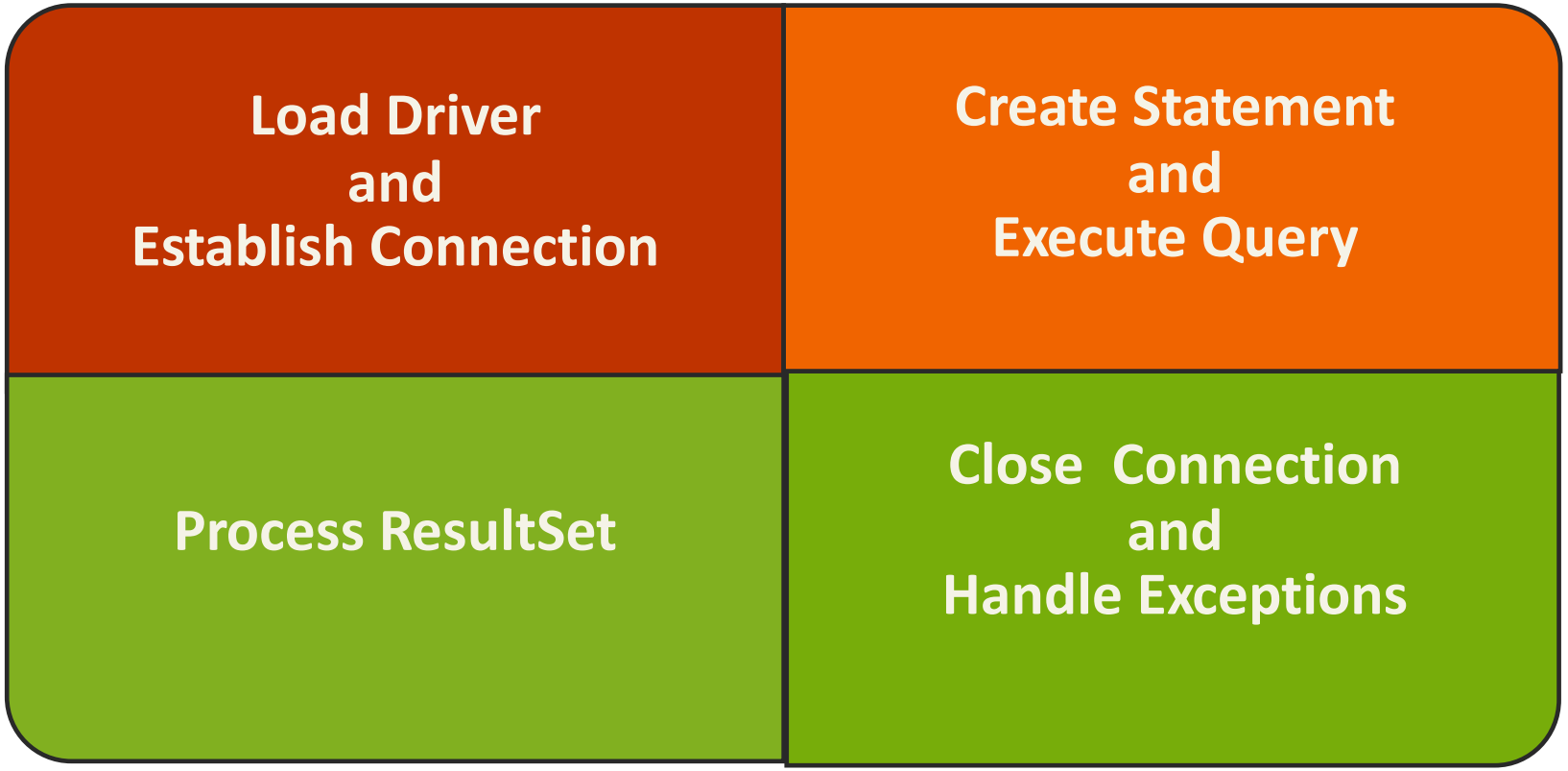
Class.forName loads a class using Classloader, including running its static initializers.

- ***The goal is to allow instanciation of a Connection from a static string, potentially from a configuration file.***
- ***Takes a Class or Interface as parameter.***

MySQL Sample Data Creation

```
mysql> create database plive;
mysql> use plive;
Database changed
mysql> create table emp(id int(10),name varchar(40),age int(3));
mysql> insert into emp(id,name,age) values (1,'Percona Live',10);
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| plive              |
| sys                |
+--------------------+
6 rows in set (0.00 sec)
```

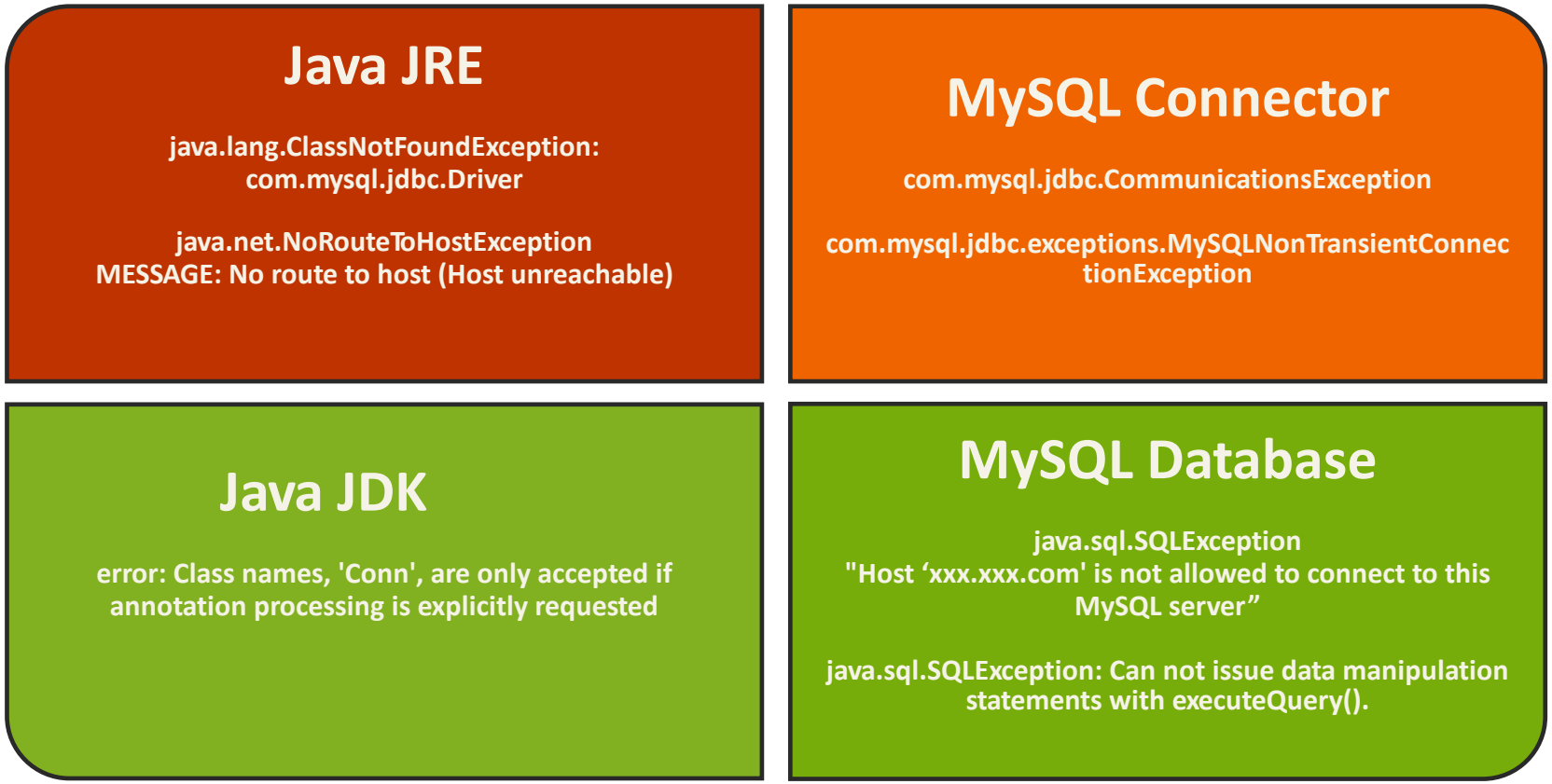
Java MySQL Connection Recap



Troubleshooting

Potential issues that you may encounter

Exceptions/Errors per Component



Java JDK

```
$ javac Conn
error: Class names, 'Conn', are only accepted if annotation processing is explicitly requested
1 error

$ javac Conn.java
$ ls -la Conn.*
-rw-rw-r--. 1 percona percona 1505 Apr 17 15:16 Conn.class
-rw-rw-r--. 1 percona percona  686 Mar 18 16:54 Conn.java
```

Java JRE

$ java Conn
java.net.ConnectException
MESSAGE: Connection refused (Connection refused)

Problem: Database Host is up but MySQL not listening to the port.

Solution: Change the JDBC connection string. Connection being attempted at port 3307. By default MySQL listening to port 3306.

Java JRE

$ java Conn
java.lang.ClassNotFoundException: com.mysql.jdbc.Driver

Problem: Java Runtime Environment can not find the Connector jar.

Solution: Use java -classpath with full path or place jar under ext.
$ pwd
/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.191.b12-1.el7_6.x86_64/jre/lib/ext
$ ls -la mysql-connector.jar
-rw-r--r--. 1 root root 540852 Jan 25 14:16 mysql-connector.jar

MySQL Connector

$ java Conn

com.mysql.jdbc.CommunicationsException: Communications link failure due to underlying exception:
java.net.NoRouteToHostException
MESSAGE: No route to host (Host unreachable)

Problem: Database Host or Network is down .

MySQL Connector

$ java Conn

com.mysql.jdbc.CommunicationsException: Communications link failure due to underlying exception:
java.net.SocketException
MESSAGE: No route to host (Read failed)
(...)
Last packet sent to the server was 952026 ms ago.

Problem: Network failure while reading a table.

MySQL Connector

$ java Conn *com.mysql.jdbc.exceptions.MySQLNonTransientConnectionException: Data source rejected establishment of connection, message from server: "Too many connections".*

Problem: MySQL Database has reached the maximum number of connections.

Solution: Increase MySQL *max_connections* (default:151)

MySQL Database

$ java Conn

java.sql.SQLException: null, message from server: "Host '192.168.33.11' is not allowed to connect to this MySQL server"

Problem: User *root* does not have MySQL privilege

Solution: Grant the required privilege at MySQL level so that the connection can be established:

mysql> GRANT ALL PRIVILEGES ON *.* TO 'root'@'%' IDENTIFIED BY 'PASSWORD'

MySQL Database

$ java Conn
java.sql.SQLException: Can not issue data manipulation statements with executeQuery().

Problem: Using *executeQuery* method with an *Update Statement*

Solution: Use *executeUpdate* method

```
Statement stmt=con.createStatement();
ResultSet rs=stmt.executeUpdate(SQL_Query);
```

Why it is so important to handle Exceptions?

Java MySQL Connection is not reliable!

“There is no safe method of reconnecting to MySQL server without r corruption of the connection or database state”.

- conn.createStatement().execute("UPDATE checking_account SET balance = balance - 1000.00 WHERE customer='Smith'");
- conn.createStatement().execute("UPDATE savings_account SET balance = balance + 1000.00 WHERE customer='Smith'");
- conn.commit();

What can be done to avoid such errors?

Try Catch SQLException, Retry, Rollback

“There is no safe method of reconnecting to MySQL server without risking corruption of the connection or database state”.

```
try { Logic }  catch (SQLException sqlEx) {
String sqlState = sqlEx.getSQLState();
if ("08S01".equals(sqlState) || "40001".equals(sqlState)) { retry; }   }
```

The two SQL states 'retry-able' are:

- 08S01 for a communications error.
- 40001 for deadlock.

What about other exceptions/errors?

Try Catch from more specific to general Exception

Watch out for code bugs NPEs, ClassCastException, etc .

```
try { Logic }
catch (SQLException se) {
    System.out.println("SQL  State is = " + se.getSQLState());
    System.out.println("Error Code = " + se.getErrorCode());
    System.out.println("Error Message = " + se.getMessage());
  }
catch (Exception e) {
    System.out.println(e);
  }
```

Technical Reference

MySQL Connector/J 8.0 Developer Guide

Questions?

Any questions or concerns that you may have

Thank You!!!

Please reach out for any questions

rodrigo.trindade@percona.com

Champions of Unbiased
Open Source Database Solutions