Data Analysis and Integration

Lab 0: Preparing the virtual machine

Note: This lab will prepare the virtual machine (VM) to be used in the labs and in the project.

Installing VirtualBox

1. Open the following URL in your Web browser:
   - https://www.virtualbox.org/wiki/Downloads

2. Download the appropriate **platform package** for your operating system.
   Note: Only the platform package is required; the Extension Pack and the SDK are not needed.

3. Run the installer to install Oracle VM VirtualBox on your system.
   Note: When selecting the features to install, you can disable VirtualBox Python Support.

Downloading the virtual machine

4. Open the following URL in your Web browser:
   - http://groups.tecnico.ulisboa.pt/aid-meic/virtualbox/

5. Right-click the **.vbox** file and select ***Save link as...*** to download the configuration file for the VM.

6. Right-click the **.vdi** file and select ***Save link as...*** to download the disk image file for the VM.
   Note: The disk image file is quite large and may take a long time to download.

Setting up the virtual machine

7. Open **Oracle VM VirtualBox** in your system.

8. In the menu bar, select **Machine > Add**.

9. Browse to the folder where the **.vbox** file is located and select it.

10. Right-click the newly added VM and select **Settings**.

11. In the **System** page, notice that the VM is configured to use 4 GB of RAM (**Base Memory**).
    - This assumes that you have at least 8 GB of RAM in your computer.
    - Depending on your system, you might be able to increase the amount of memory that will be dedicated to the VM.

- The VM should work fine with 4 GB. However, if you have enough resources, you might want to increase it up to 8 GB. There is no need to increase it further than that.

Note: As a rule of thumb, the base memory for the VM should not exceed half of the physical RAM available in the computer.

12. Change to the **Processor** tab, and notice that the VM is configured to use 2 CPUs.
    - This assumes that you have at least 4 CPU cores in your computer, either physical or virtual.
    - Depending on your system, you might be able to increase the number of CPUs that will be dedicated to the VM.
    - The VM should work fine with 2 CPUs. However, if you have enough resources, you might want to increase it up to 4 CPUs. There is no need to increase it further than that.

    Note: As a rule of thumb, the number of CPUs for the VM should not exceed half of the CPUs available in the computer.

13. Close the **Settings** windows with **OK** or **Cancel**, depending on whether you made any changes or not.

Starting the VM

14. Click **Start** to boot up the VM.

15. Once the VM boots up, open Firefox inside the VM.

16. Note that Firefox is redirecting to the page where you can find the lab guides for this course. In subsequent labs, you can access the lab guide directly from Firefox inside the VM.

17. Optionally, you can continue this lab inside the VM, by opening this same lab guide in Firefox.

    *In case you run into problems starting the VM check the **Troubleshooting** section at the end of this document.*

Checking the software: MySQL

18. Open a terminal and execute the following command: **mysql -u aid -p**
    Password: **aid**

19. You are now connected to MySQL. Execute the command **show databases;** (with semicolon) to show the databases available in MySQL.
    Note: For the moment, these are just the system databases. In this course, we will be creating several MySQL databases.

20. Press **Ctrl-D** to exit the MySQL prompt.

Checking the software: Pentaho Data Integration

21. In the terminal, change to the Pentaho folder with the command: **cd Pentaho**

22. Check the contents of that folder with the command: **ls -las**
    Note: These folders contain the tools that we will be using in this course.

23. Change to the **data-integration** folder and check its contents.

24. Start Pentaho Data Integration (PDI) with the command: **./spoon.sh**

25. We will be using Pentaho Data Integration (PDI) to create and run data transformations. With this tool, we can read data from one or more data sources (files, databases, etc.), transform those data, and store the results somewhere else (in another file or database, for example).

26. Close Pentaho Data Integration (PDI).

Checking the software: DataCleaner

27. Change to the **../DataCleaner** folder and check its contents.

28. Start DataCleaner with the command: **./datacleaner.sh**

29. We will be using DataCleaner for data profiling, i.e. collecting information about the data that we are dealing with, such as missing values, value ranges, value distributions, string lengths, etc.

30. Close DataCleaner.

Checking the software: Pentaho Schema Workbench

31. Change to the **../schema-workbench** folder and check its contents.

32. Start Pentaho Schema Workbench (PSW) with the following command: **./workbench.sh**

33. We will be using Pentaho Schema Workbench (PSW) to define data cubes when working with a data warehouse. This tool is basically an XML editor that saves cube definition in an XML file.

34. Close Pentaho Schema Workbench (PSW).

Checking the software: Pentaho Server & Saiku Analytics

35. Change to the **../pentaho-server** folder and check its contents.

36. Start Pentaho Server with the following command: **./start-pentaho.sh**
    Note: Pentaho Server will start running on the background. It may take some time for its startup to complete. You can use System Monitor to check CPU activity and memory usage.

37. Pentaho Server is a Web application server based on Apache Tomcat. We will be using Pentaho Server as a container for Saiku Analytics, a tool that we will use to perform multidimensional analysis over a data cube.

38. Open Firefox and navigate to: **http://localhost:8080/**

39. On the **Welcome** page, press **Log in as an evaluator** and **Log in** as **Administrator**.

40. In the **File** menu at the top left corner, select **New > Saiku Analytics**.

41. This is the tool that we were referring to (Saiku Analytics). We will be using this tool to run MDX queries over a data warehouse. MDX is an SQL-like query language for data warehouses.

42. Close Firefox.

43. Back in the terminal, stop Pentaho Server with the following command: **./stop-pentaho.sh**

Checking the software: Pentaho Report Designer

44. Change to the **../report-designer** folder and check its contents.

45. Start Pentaho Report Designer (PRD) with the following command: **./report-designer.sh**

46. We will be using Pentaho Report Designer (PRD) to design and generate reports based on data in a database or in data warehouse.

47. Close Pentaho Report Designer (PRD).

48. Press **Ctrl-D** to exit the terminal.

Checking the software: other tools

49. There are other tools that we will be using occasionally. In Applications, locate or search for LibreOffice Calc and open it.

50. We will be using LibreOffice Calc to open CSV files (text files with comma-separated values).

51. Close LibreOffice Calc.

52. We will also be using MySQL Workbench. In Applications, locate or search for MySQL Workbench and open it.

53. In the main window, click **Local instance** to connect to MySQL.

54. In some cases, when we want to explore a given database, it might be easier to do it with MySQL Workbench rather than using the MySQL prompt in the terminal.

55. Close MySQL Workbench.

56. Finally, most software that we will be using is based on Java. Open a terminal and execute the command **java -version** to check the Java version that we are using.
Note: The Java version has been selected according to the software requirements.

Troubleshooting

There are a few known issues that may prevent VirtualBox from correctly starting a virtual machine. The common message "Failed to open a session for the virtual machine" is shown in these cases.

On the **Linux** operating system the following aspects must be considered:

a. VirtualBox requires the compiler with which the Kernel was compiled to be installed. Check the correct compiler version by issuing the command `cat /proc/version`. More information in this post.

b. On systems with Secure Boot active, the VirtualBox kernel modules must be signed. Check this recipe for signing. An alternative is to simply deactivate Secure Boot.

VirtualBox runs natively only on Intel architectures, thus it is unusable on different chipsets, particularly **ARM**. On those other architectures a few options can be tried, albeit with limitations:

a. Run VirtualBox on a emulator such as QEMU.

b. Experiment with an ARM development snapshot.