Data Management Lab

Lecture 0: introduction to the lab environment

Who I Am

Alessandro Tundo

U14 – 1st floor – SAL1 (Room 1039)

alessandro.tundo@unimib.it

@tunale

The Linux Shell ▶

What is the shell?

Simply put, the shell is a program that takes commands from the keyboard and gives them to the operating system to perform. In the old days, it was the only user interface available on a Unix-like system such as Linux. Nowadays, we have *graphical user interfaces (GUIs)* in addition to *command line interfaces (CLIs)* such as the shell.

What is the terminal?

It's a program called a *terminal emulator*. This is a program that opens a window and lets you interact with the shell. There are a bunch of different terminal emulators you can use. Most Linux distributions supply several, such as: gnome-terminal, konsole, xterm, and many more…

What is bash?

Bash is the shell, or command language interpreter, for the GNU operating system. The name is an acronym for the 'Bourne-Again SHell', a pun on Stephen Bourne, the author of the direct ancestor of the current Unix shell sh, which appeared in the Seventh Edition Bell Labs Research version of Unix.

Sources: http://linuxcommand.org/lc3_lts0010.php, https://www.gnu.org/software/bash/manual/html_node/What-is-Bash_003f.html

Some useful shell commands

Command	Description
ls *dir*	Lists all files in the dir directory
pwd	Prints out the full path of the current directory
cd *dir*	Moves into the dir directory
cd *..*	Goes back to the parent directory
cp *fileA fileB*	Copies fileA to fileB (like copy&paste)
mv *fileA fileB*	Moves fileA to fileB (like cut&paste)
rm *fileA*	Removes the fileA definitely
rm *-r dir*	Removes the dir directory and its content definitely (note the -r option, which is recursive)
touch *fileA*	Creates a new empty file named fileA in the current directory
mkdir *dir*	Creates a new dir directory in the current directory
cat *fileA*	Shows the content the file fileA
head *-n fileA*	Shows the first n lines of fileA
tail *-n fileA*	Shows the last n lines of fileA
grep *”something” fileA*	Shows all the lines which contain "something"

The Lab Environment

Azure Lab Services

Small VMs

- 2 vCPU
- 3.5GB RAM
- 128GB Disk
- Debian 10 OS

We will use them for the first lessons. These VMs are targeted for small tasks and services with little resource consumption.

https://labs.azure.com/register/ynu36zub

Click here to get your VM!

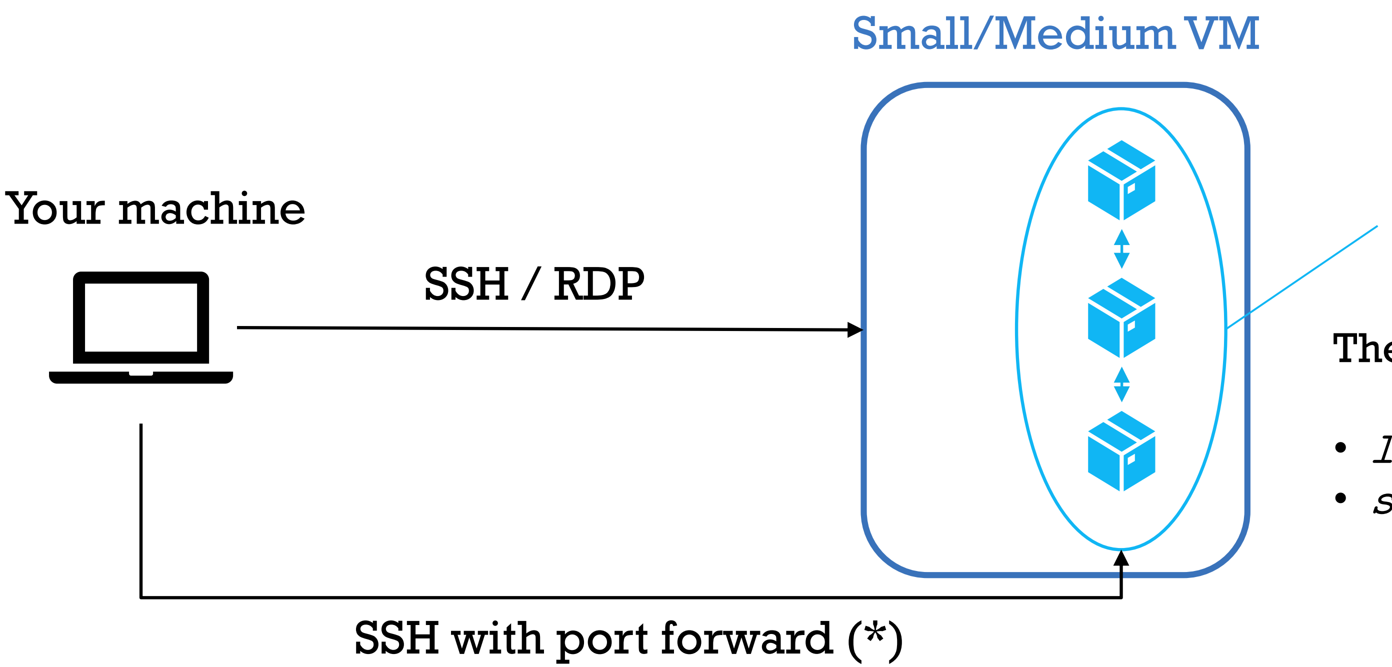
Medium VMs

- 4 vCPU
- 12GB RAM
- 128GB Disk
- Debian 10 OS

We will use them for Big Data tasks later on. On these VMs you can run "heavy" services such as Apache Hadoop, Apache HBase, etc...

The registration link will be provided

Architecture overview



Docker Containers
that host services

They are accessible within the VM at

- *`localhost:servicePort`*
- *`serviceName:servicePort`*

(*) *With the port forward, you will access to the services running in the containers as they are on your machine at* `localhost:servicePort`*. You will get further details later…*

What are the available services?

- **MongoDB** v4.2 (with **Mongo Express** WebUI)
- **Neo4j** v3.5
- **ArangoDB** v3.5
- **JupyterLab** (datascience notebook release)
- Apache **Kafka** v2.3.0
- Apache **ZooKeeper** v3.4.10
- Apache **NiFi** v1.10

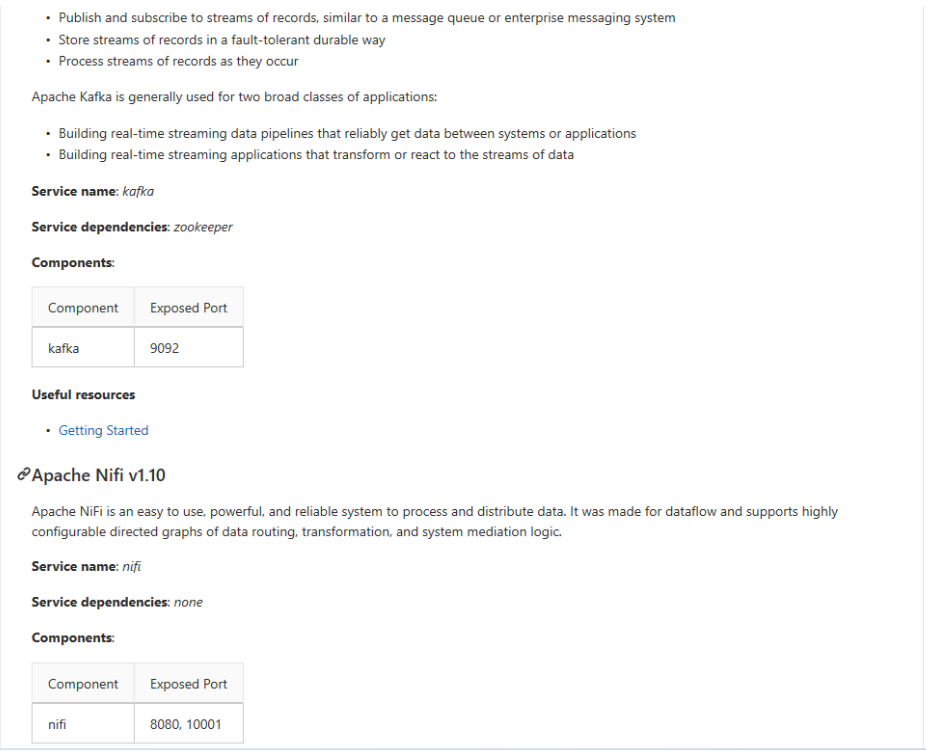
*WHEN PROCESS **LARGE AMOUNTS OF DATA** (ORDER OF MAGNITUDE OF GIGABYTES) PERFORM BETTER ON THE **MEDIUM** VM!*

- Apache **Hadoop** v2.7.4
- Apache **Hbase** v1.2.6
- Apache **Hive** v2.3.2

*USE THEM **ONLY** ON THE **MEDIUM** VM!*

How can I manage the services?

We will dig into details in the next lectures, in the meanwhile you can check out this link: https://gitlab.com/aletundo/data-management-lab



- Publish and subscribe to streams of records, similar to a message queue or enterprise messaging system
- Store streams of records in a fault-tolerant durable way
- Process streams of records as they occur

Apache Kafka is generally used for two broad classes of applications:

- Building real-time streaming data pipelines that reliably get data between systems or applications
- Building real-time streaming applications that transform or react to the streams of data

Service name: *kafka*

Service dependencies: *zookeeper*

Components:

Component	Exposed Port
kafka	9092

Useful resources

- Getting Started

Apache Nifi v1.10

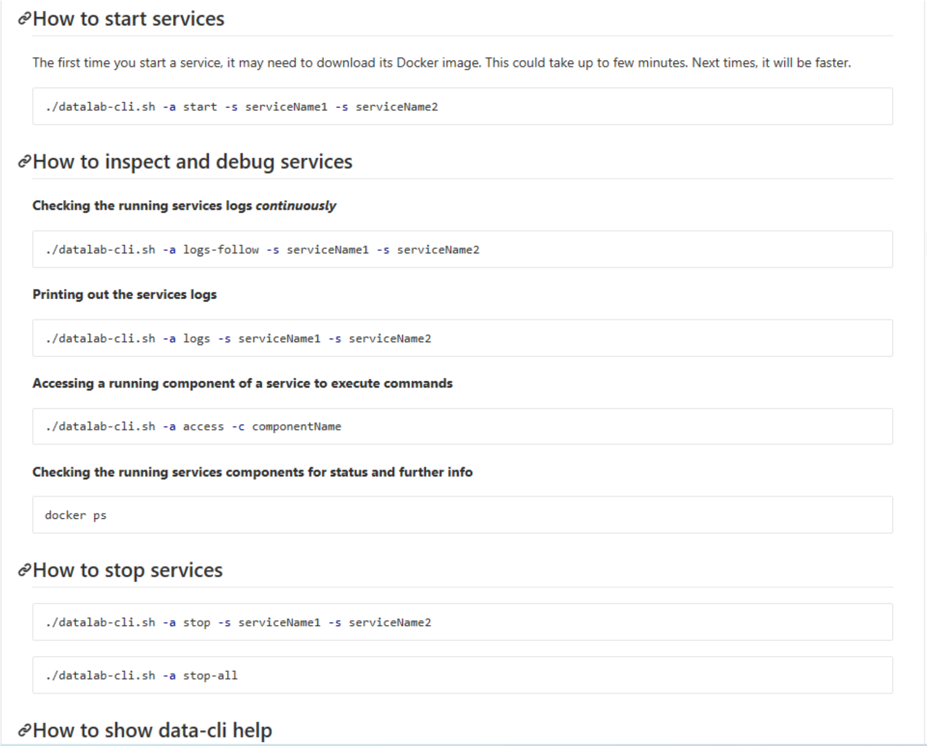
Apache NiFi is an easy to use, powerful, and reliable system to process and distribute data. It was made for dataflow and supports highly configurable directed graphs of data routing, transformation, and system mediation logic.

Service name: *nifi*

Service dependencies: *none*

Components:

Component	Exposed Port
nifi	8080, 10001



How to start services

The first time you start a service, it may need to download its Docker image. This could take up to few minutes. Next times, it will be faster.

```
./datalab-cli.sh -a start -s serviceName1 -s serviceName2
```

How to inspect and debug services

Checking the running services logs *continuously*

```
./datalab-cli.sh -a logs-follow -s serviceName1 -s serviceName2
```

Printing out the services logs

```
./datalab-cli.sh -a logs -s serviceName1 -s serviceName2
```

Accessing a running component of a service to execute commands

```
./datalab-cli.sh -a access -c componentName
```

Checking the running services components for status and further info

```
docker ps
```

How to stop services

```
./datalab-cli.sh -a stop -s serviceName1 -s serviceName2
```

```
./datalab-cli.sh -a stop-all
```

How to show data-cli help

How to access your VM

Secure SHell (SSH)

- Provides a secure channel over an unsecured network connecting an *SSH client* application with an *SSH server*
- Used to log into a remote machine and execute commands
- Supports also *tunneling*, *forwarding TCP* ports and X11 connections
- Can **transfer files** using the associated SSH file transfer (SFTP) or secure copy (***SCP***) protocols

Remote Desktop Protocol (RDP)

- Protocol developed by Microsoft (open source implementation exist!), which provides a user with a **graphical interface to connect to another computer** over a network connection
- User employs **RDP client** software for this purpose, while the other computer must run RDP server software.

How to access your VM: the SSH way

1. Access to https://labs.azure.com
2. Start the VM (if not already started) and click on ***Connect via SSH***
3. Copy the command provided and paste it in your terminal (see below)
4. Press Enter

Unix-like OS 😎

You are lucky (and a *good* person!), you already have an SSH client on your machine.

Copy&Paste the command in a terminal and you are done!

Windows OS 😟

You are unlucky (and *bad* person, sorry!), you must do some additional steps.

Please, read the next slide!

Windows SSH client (for *"bad"* people only)

Windows 10

There is an SSH client as an *"optional feature"*

- Go ***Settings*** -> ***Apps*** and click *"**Manage optional features**"* under *Apps & Features*
- Click "***Add a feature***" at the top of the list of installed features
- Select the "***OpenSSH Client***" option and click **"Install"**

Read more at:
https://www.howtogeek.com/336775/how-to-enable-and-use-windows-10s-built-in-ssh-commands/

Other versions

You can install **PuTTy** to create SSH connections

- Download it from **here**, select the x64 version

How to access your VM: the RDP way

1. Access to https://labs.azure.com
2. Start the VM (if not already started) and click on ***Connect via RDP***
3. Download the RDP file
4. Open your RDP client and import the file (or double click on the file, it should run directly the RDP client)
5. Connect to the VM

Linux OS

- Install an RDP client for your Linux distribution if you do not have it already (e.g.: ***Remmina***)

MacOs

- Download and install ***Microsoft Remote Desktop 10*** from the *Mac App Store*

Windows 10 OS

- Look for ***Remote Desktop Client*** in your *Apps* or download it from the *Windows Store*

How to copy files from your machine to the VM

Unix-like OS and Windows 10: Secure Copy (scp)

```
scp -P 1234 filepath studente@hostname:/home/studente/my-data
        A       B             C
```

A. Change it with the port provided in the ssh connection command
B. Change it with your filepath, add the option **-r** if it is a directory
C. Change it with the hostname provided in the ssh connection command

Other Windows versions: WinSCP

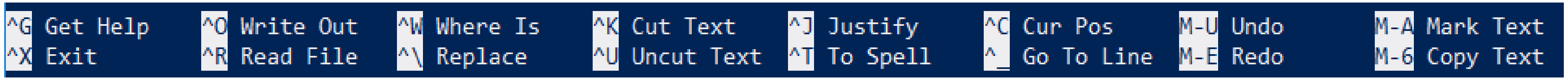
- Download WinSCP at **https://winscp.net/eng/download.php**
- Install WinSCP
- You can drag&drop your files after the connection setup
- Learn more about WinSCP at the following links: **docs**, **tutorial**

Hands-on: getting started with the shell

▶

nano editor: a quick introduction

- Launch the editor: **nano**
- Open a new file: **nano new_file**
- Open an existing file: **nano existing_file**
- Save: **CTRL+O**
- Exit: **CTRL+X**
- Some useful commands are explained at the bottom of the terminal



```
^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify     ^C Cur Pos     M-U Undo       M-A Mark Text
^X Exit        ^R Read File   ^\ Replace     ^U Uncut Text  ^T To Spell    ^_ Go To Line  M-E Redo       M-6 Copy Text
```

Exercise 0: create and copy

1. Create a CSV file on your machine (fill it as you prefer…)
2. Copy the CSV file within the VM into `/home/studente/my-data` directory
   1. See slide "How to copy files from…"

Exercise 1: fiddle with files and directories

1. Access to your VM with SSH
2. Enter into `/home/studente/my-data` directory (check where you are with **pwd**)
3. Create a new directory named `new-dir`
4. Go back to `/home/studente/my-data`
5. Copy the CSV file within the new directory `new-dir`
6. Delete the original CSV file
7. Rename (move…) `new-dir` to `renamed-dir`
8. Go back to your home directory `(/home/studente`)

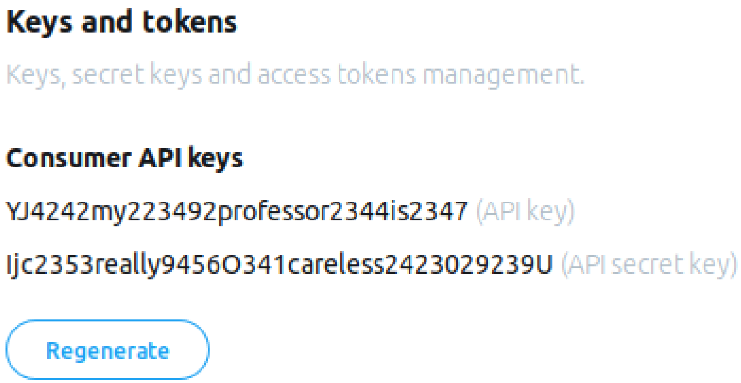
Get your Twitter API key

Twitter Developer Account

1. Visit https://developer.twitter.com/en/apply-for-access.html
2. Click the **Apply for a developer account** button
3. Log in with your account credential
   1. You need to create a new Twitter account if you don't already have it
4. You must add a valid phone number to verify your account
   1. Read more about your privacy here
5. Select **I am requesting access for my own personal use (if requested)**
6. Select the cases you are interested in (e.g.: Student project) and explain why you need to use Twitter APIs
   1. I'm using Twitter's APIs during a hands-on session …
   2. I plan to use Tweets to learn how to use data management tools …
   3. I don't need to perform tweeting or retweeting … (we want to learn how to manage streamings!)
   4. I plan to run some queries to my local document database, and the results will be displayed in aggregate only to professors/colleagues…

Twitter Developer Account (continue…)

7. Verify your email
8. Create an application to obtain your key
   1. Choose a name and describe your application
   2. Fill in the Website URL field even if we will not publish our app…
   3. **Do not** check the **Enable Sign In with Twitter**
   4. Ignore all the other fields, but explain (once more) how this app will be used (I will use this app during my hands-on session…)
9. Accept and close the dialog (you should also read its content, shouldn't you?)
10. Move to the **Keys and tokens** tab to find your **Consumer keys**



Keys and tokens

Keys, secret keys and access tokens management.

Consumer API keys

YJ4242my223492professor2344is2347 (API key)

Ijc2353really9456O341careless2423029239U (API secret key)