University of Salford, MSc Data Science

Module: Big Data Tools and Techniques

Date: Trimester 2, 2024-2025

Session: Workshop Week 1

Topic: Linux operating system

Tools: Databricks Community Edition

Instructors: Dr Taha Mansouri, Nathan Topping, and Dr Kaveh Kiani

Objectives:

After completing this workshop, you will be able to:

- Running a Databricks notebook
- Running Some Linux commands
- Working with Local Filesystem

Table of Contents

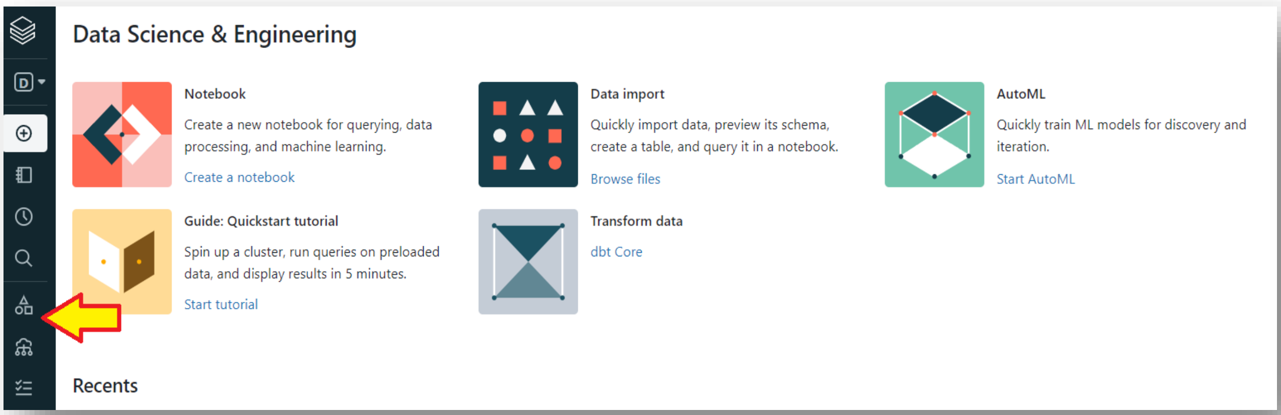
Important Note: In all BDTT workshop notes, if the screenshots appear slightly different from what you see on your screen, there's no need to worry. The Databricks platform is highly dynamic, constantly updating its contents and procedures. Please proceed with your work as usual and we will let you know if any major changes arise.

Part 1: Fire up the Databricks workspace

1. Log in to your Databricks Community Edition account if already you have an account:

   https://community.cloud.databricks.com/login.html

2. You will need to create a cluster to start your analysis – this gives you access to a machine to use. Click on "Compute" on the main page.



Data Science & Engineering

Notebook
Create a new notebook for querying, data processing, and machine learning.
Create a notebook

Data import
Quickly import data, preview its schema, create a table, and query it in a notebook.
Browse files

AutoML
Quickly train ML models for discovery and iteration.
Start AutoML

Guide: Quickstart tutorial
Spin up a cluster, run queries on preloaded data, and display results in 5 minutes.
Start tutorial

Transform data
dbt Core

Recents

3. Click on "Create Compute" and type in a new name for the cluster in the "compute name" box, any name that you like.
4. Select the most recent runtime (currently, runtime 14.2 (Scala 2.12, Spark 3.5.0) and press create compute

   Note that it will take a few minutes to create the cluster. After some time, the green circle next to the cluster name will gain a green tick meaning the cluster has successfully started up.

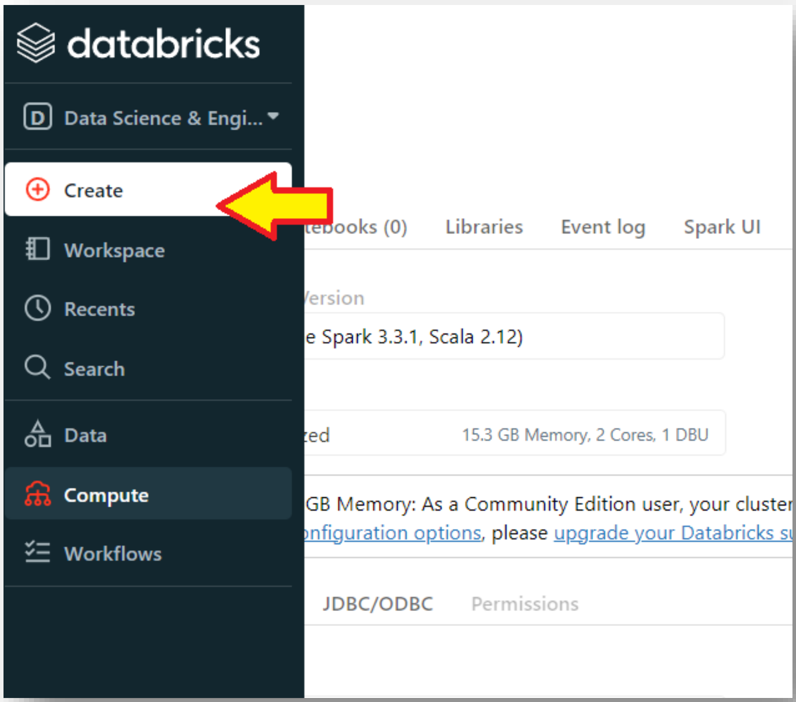
Part 2: Create a Python notebook

Databricks programs are written in Notebooks. A notebook is a collection of runnable cells which contain your commands. Look at

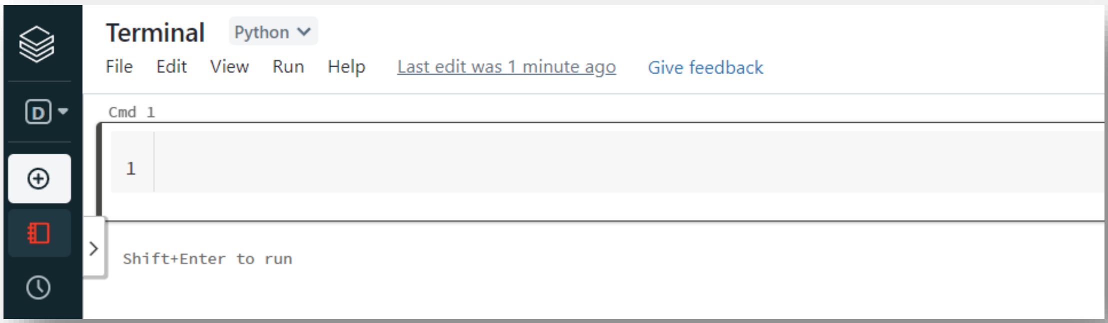
https://docs.databricks.com/notebooks/notebooks-use.html

for an overview of how to use a Databricks notebook.

1. Click on the button “Create” and select “Notebook”.


databricks
Data Science & Engi...
Create
Workspace
Recents
Search
Data
Compute
Workflows
tebooks (0)
Libraries
Event log
Spark UI
ersion
e Spark 3.3.1, Scala 2.12)
ed
15.3 GB Memory, 2 Cores, 1 DBU
GB Memory: As a Community Edition user, your cluster
nfiguration options, please upgrade your Databricks su
JDBC/ODBC
Permissions

2. Name your notebook, e.g. “Terminal”, leave (or select) the language to be “Python” and the cluster should be your currently created cluster. This will give you access to a notebook which looks something like this:



Part 3: LINUX Operating System (OS)

1) What is LINUX

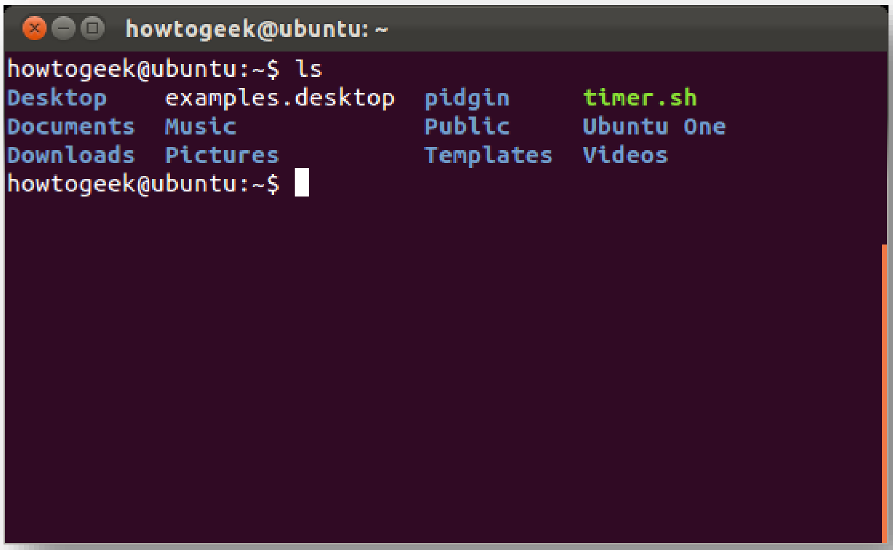
Linux is an open-source **operating system** (OS). So, LINUX is an operating system but unlike the Microsoft Windows, is a free and open-source OS. Data Scientists use Linux because:

- Effective help with quick manipulation and analysis of data
- Data science tools come with a command line interface
- Repeatable systems (more about this later)

When you use a computer, you should work with a OS. Most of us are used to work with Windows or macOS. You know that an **Operating System (OS)** is a software that acts as an interface between computer hardware components and the user. Every computer system must have at least one operating system to run other programs. Applications like Browsers, MS Office, etc. Databricks have assigned you a computer system on the Cloud (The cluster that you created). To communicate with this cluster, you must use Linux OS.

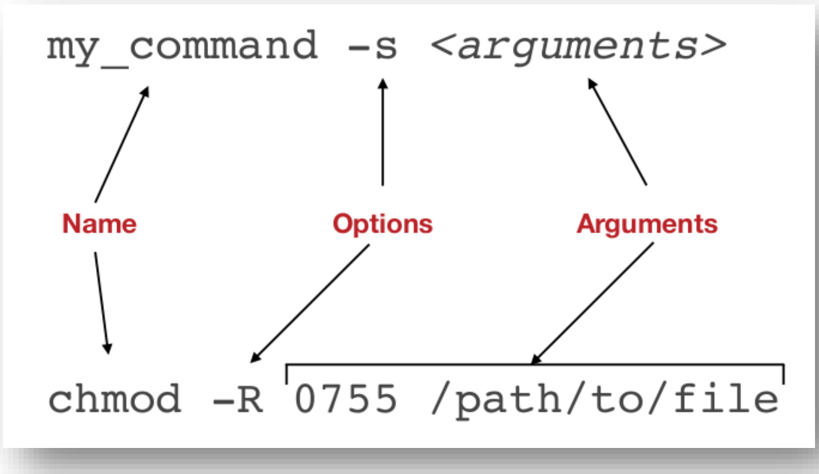
Unlike Windows, in Linux, you need to write commands to interact with the OS. If you need to create a folder, you can't right-click and then click on the New Folder tab, you should write a command. The same applies to Copy and Paste and so on. But where you can write your commands?

The **command line** is an interface that allows you for typing commands directly to a computer's operating system.



```
howtogeek@ubuntu:~$ ls
Desktop    examples.desktop  pidgin     timer.sh
Documents  Music             Public     Ubuntu One
Downloads  Pictures          Templates  Videos
howtogeek@ubuntu:~$
```

A Linux basic command is made up of three parts: Name, Options, and Arguments:

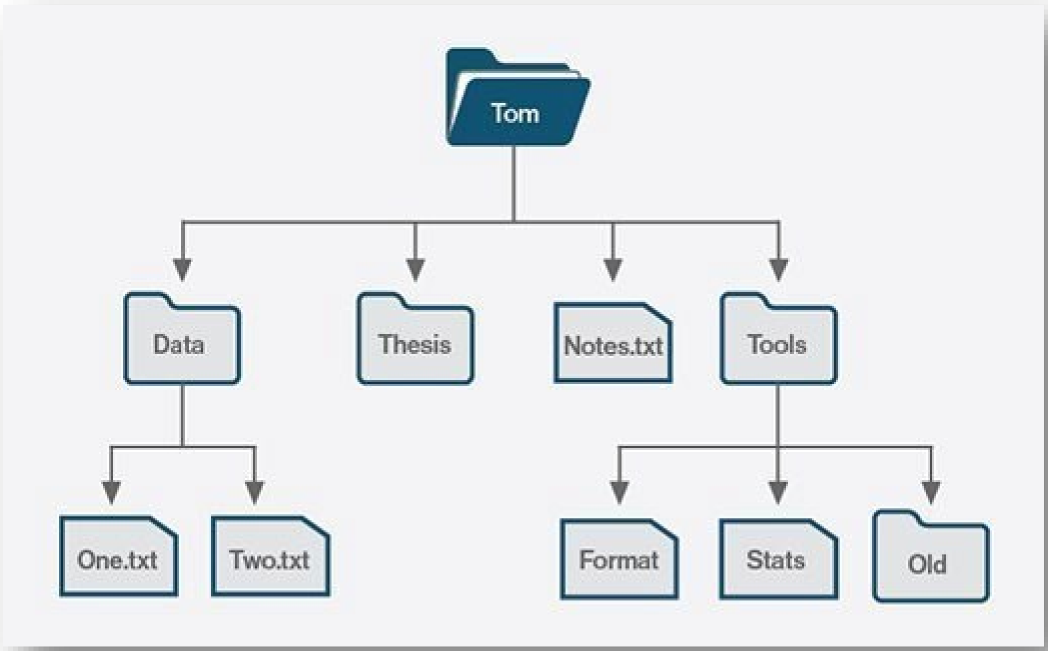


2) What is a Filesystem (FS)?

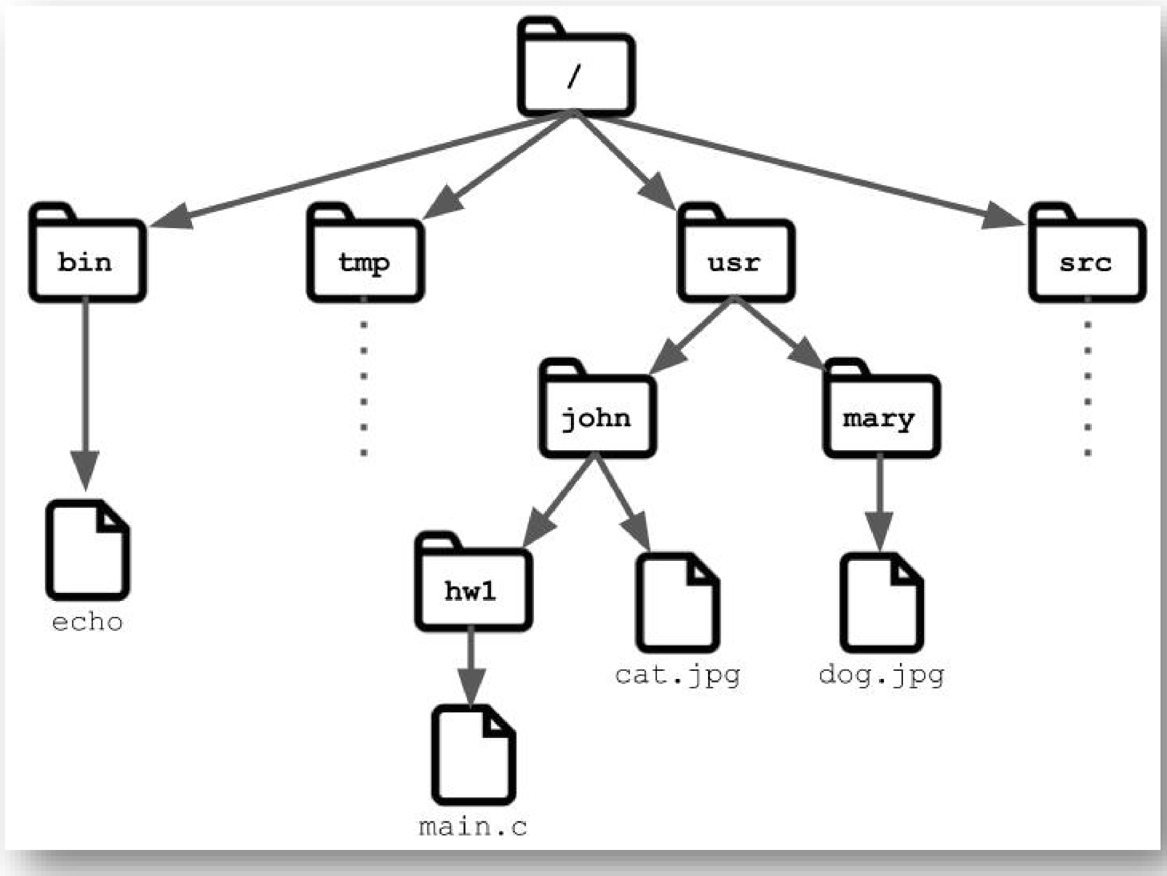
Computers use particular kinds of filesystems (FS) to store and organize data in drives, such as a hard drive or flash drive, or the CDs, DVDs, or Clouds.

A filesystem can be thought of as an index or a database containing the address of the physical location of every piece of data on the device. The data is usually organized in **folders** called **directories**, which can contain other directories and **files**. For example, Windows OS offers three filesystems: NTFS, FAT32 and FAT16.

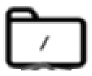
Imagine Tom uses Windows OS and wants to open/copy/delete the “One.txt” file. Tom must click on the “Tom” folder, then “Data” folder and finally do the intended operation on the “One.txt”.



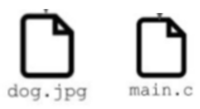
Also, Linux has a filesystem (FS) and to interact with any content that has been stored in the computer you should use its filesystem logic to do so. Hierarchy of the Linux filesystem is like:



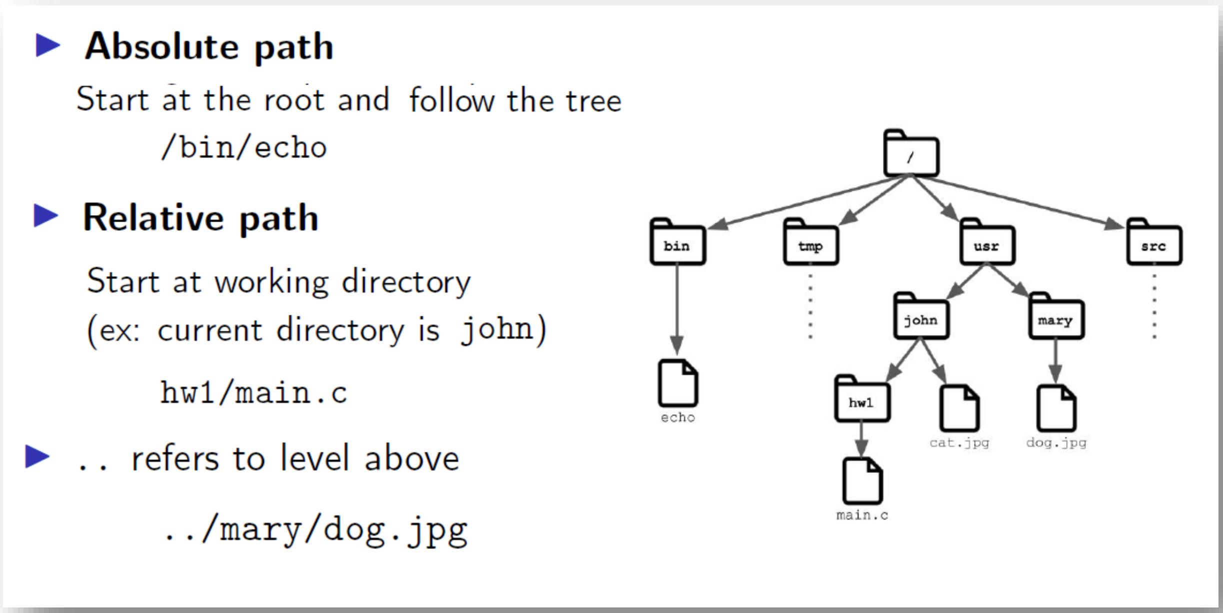
In this example:

1) You can see the “**root**” 

2) You can see multiple “**directories**” …

3) You can see multiple “**files**”  …

To get access to each of these entities you should type the address (**path**) of the entity in the command line.



- **Absolute path**

  Start at the root and follow the tree

  `/bin/echo`

- **Relative path**

  Start at working directory (ex: current directory is `john`)

  `hw1/main.c`

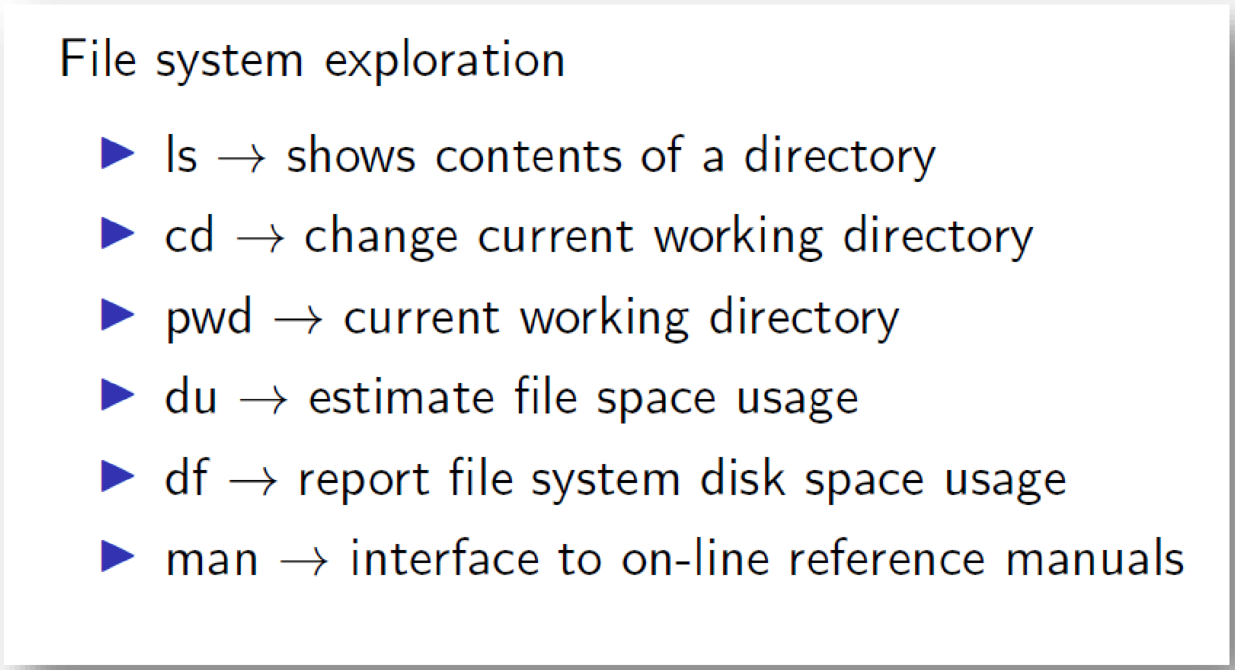
- `..` refers to level above

  `../mary/dog.jpg`

3) Linux commands

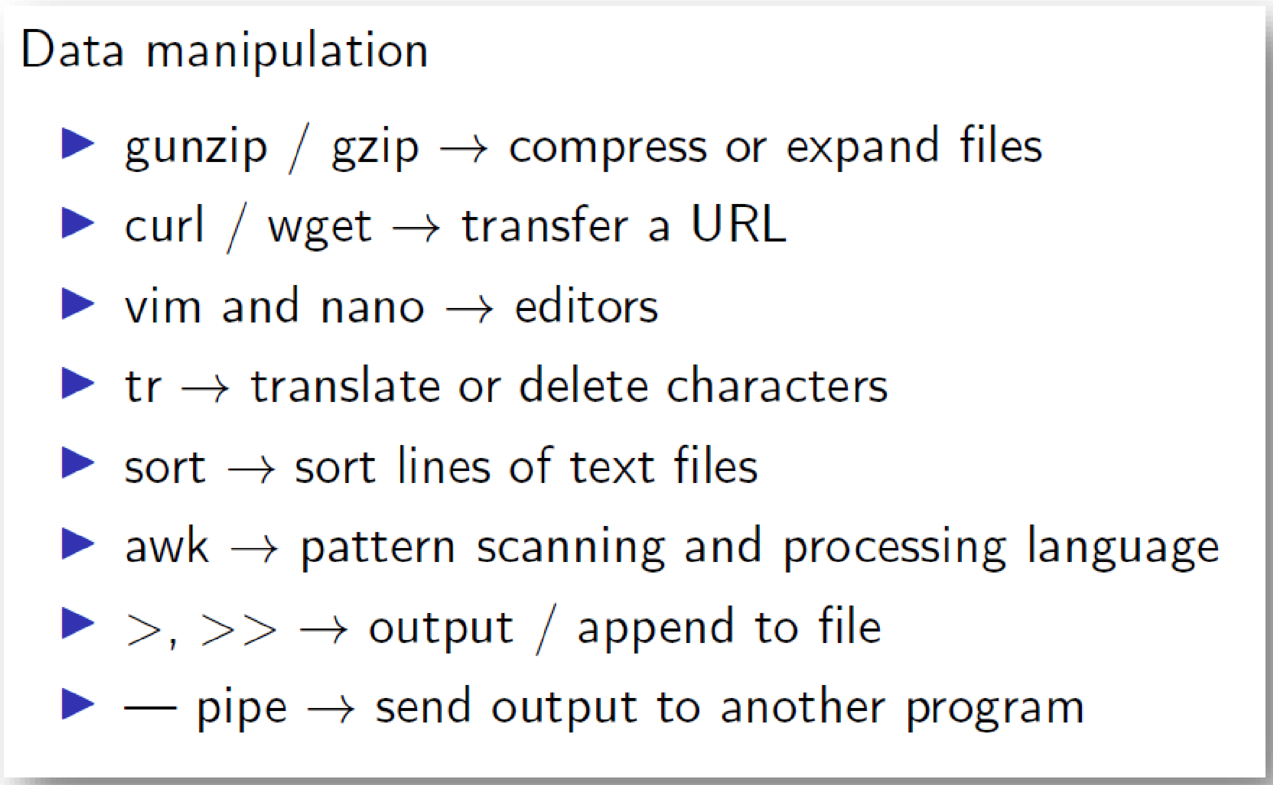
Main Linux commands are:

(You don't need to learn all this commands now, but we will use some of them during the workshops)



File system exploration

- ls → shows contents of a directory
- cd → change current working directory
- pwd → current working directory
- du → estimate file space usage
- df → report file system disk space usage
- man → interface to on-line reference manuals



Data manipulation

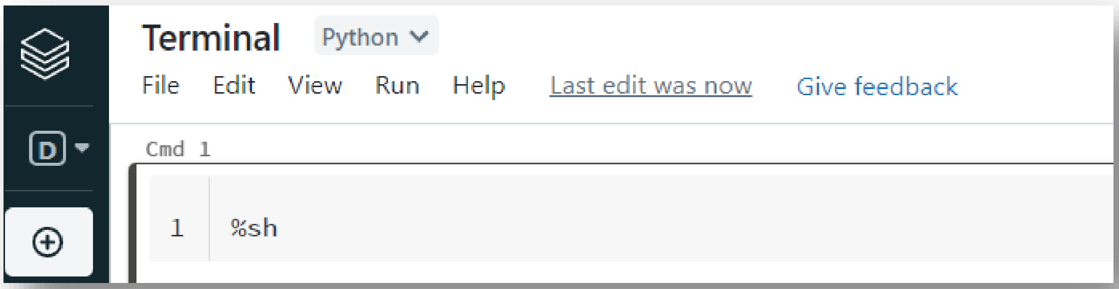
- gunzip / gzip → compress or expand files
- curl / wget → transfer a URL
- vim and nano → editors
- tr → translate or delete characters
- sort → sort lines of text files
- awk → pattern scanning and processing language
- >, >> → output / append to file
- — pipe → send output to another program

Part 4: Running some Linux commands in the Databricks notebook

1) How to use Linux commands in a notebook

The cells in the notebook expect you to be typing python code, but you can change the type of the cell by using “cell magic”.

To switch to “shell” and write Linux commands, you can type **%sh** at the start of your cell.

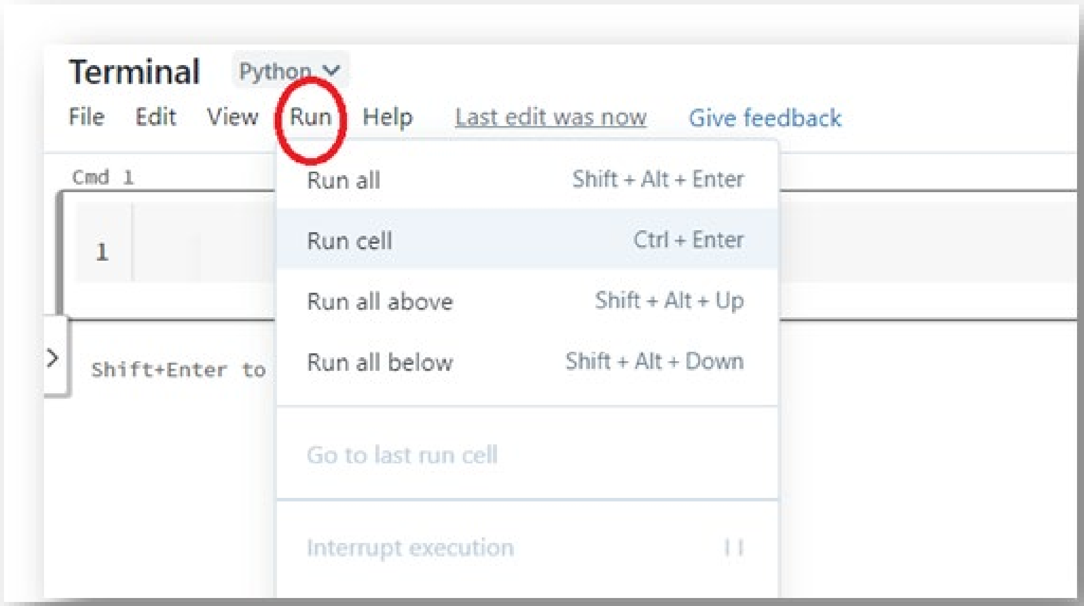


```
%sh
```

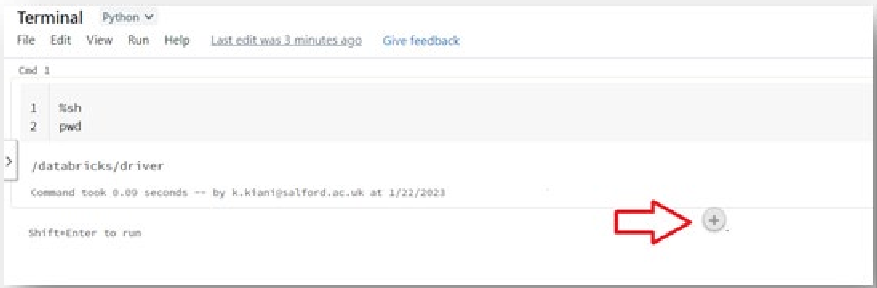
2) Explore a directory

Note 1: To run a cell use Ctrl+Enter or shift+Enter (shift+Enter will open a new cell as well)

To run a cell or specific cells or all cells, use:



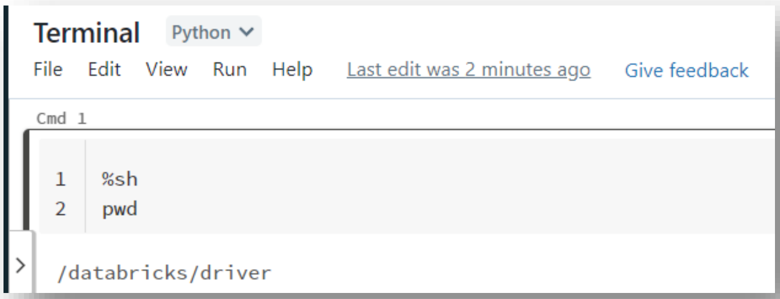
Note 2: To open a new cell press the **+** sign:



```
%sh
pwd
```

```
/databricks/driver
Command took 0.09 seconds -- by k.kiani@salford.ac.uk at 1/22/2023
```

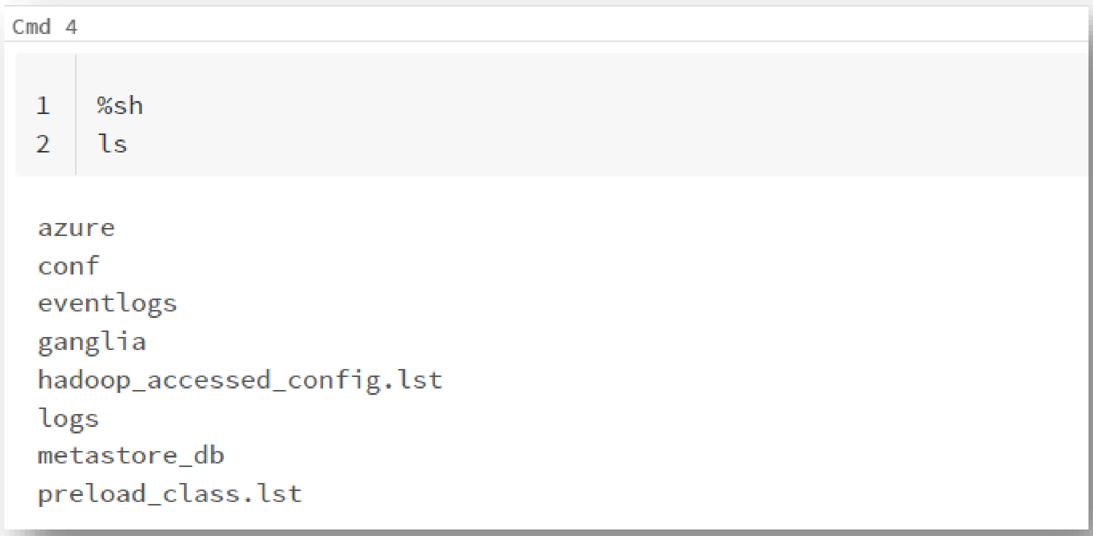
1. Check that you are currently in ***/databricks/drivers*** directory using pwd command.



```
%sh
pwd
```

```
/databricks/driver
```

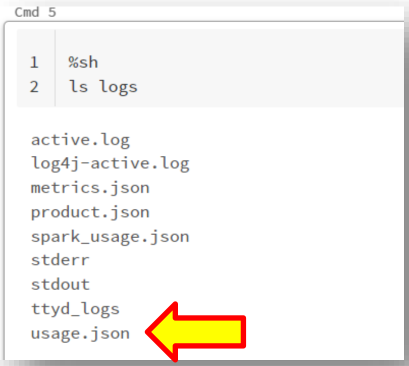
2. As in the cell shown above, you will need to ensure that the top of each cell containing shell commands starts with a %sh (this is only required once per cell). Now use the **ls** command to tell you the contents of the directory.



```
%sh
ls
```

```
azure
conf
eventlogs
ganglia
hadoop_accessed_config.lst
logs
metastore_db
preload_class.lst
```

3. What does ls -l in the same place tell you? (You can use Google “ls command Linux" to give you information about the command.)
4. The logs directory looks interesting, look at what's in there using:

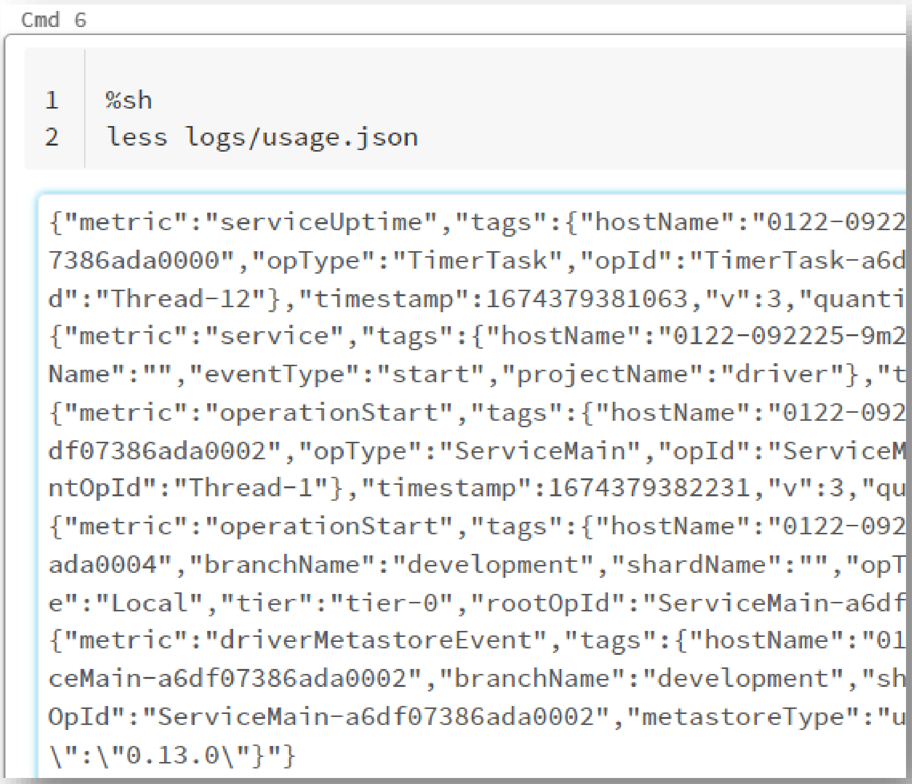


```
%sh
ls logs
```

```
active.log
log4j-active.log
metrics.json
product.json
spark_usage.json
stderr
stdout
ttyd_logs
usage.json
```

3) Explore files content.

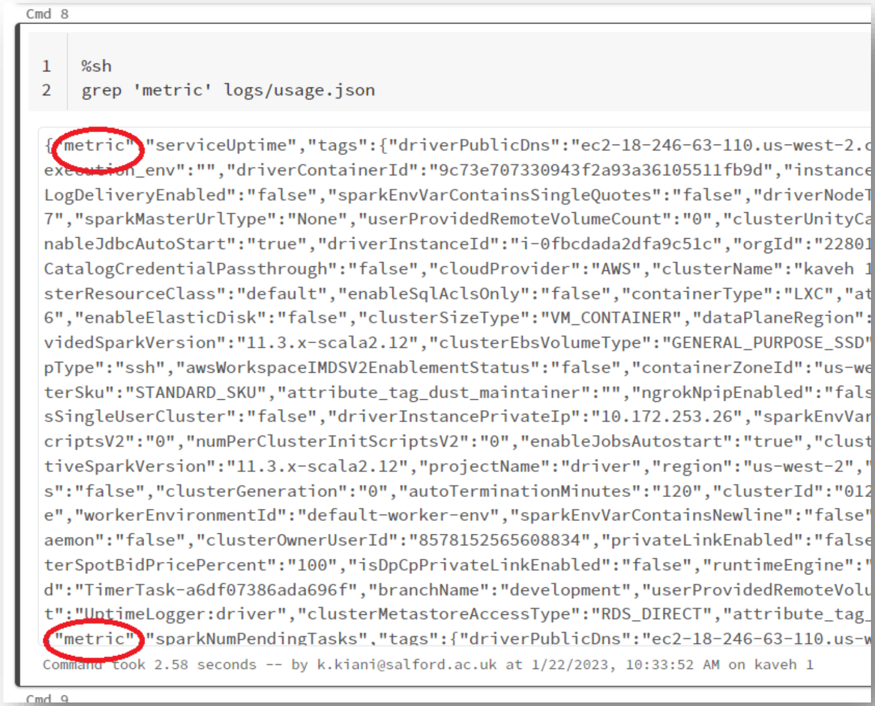
5. What command would you use to view the contents of the “logs/usage.json” file?



```
Cmd 6

%sh
less logs/usage.json
```

```
{"metric":"serviceUptime","tags":{"hostName":"0122-0922
7386ada0000","opType":"TimerTask","opId":"TimerTask-a6d
d":"Thread-12"},"timestamp":1674379381063,"v":3,"quanti
{"metric":"service","tags":{"hostName":"0122-092225-9m2
Name":"","eventType":"start","projectName":"driver"},"t
{"metric":"operationStart","tags":{"hostName":"0122-092
df07386ada0002","opType":"ServiceMain","opId":"ServiceM
ntOpId":"Thread-1"},"timestamp":1674379382231,"v":3,"qu
{"metric":"operationStart","tags":{"hostName":"0122-092
ada0004","branchName":"development","shardName":"","opT
e":"Local","tier":"tier-0","rootOpId":"ServiceMain-a6df
{"metric":"driverMetastoreEvent","tags":{"hostName":"01
ceMain-a6df07386ada0002","branchName":"development","sh
OpId":"ServiceMain-a6df07386ada0002","metastoreType":"u
\":\"0.13.0\"}"}
```

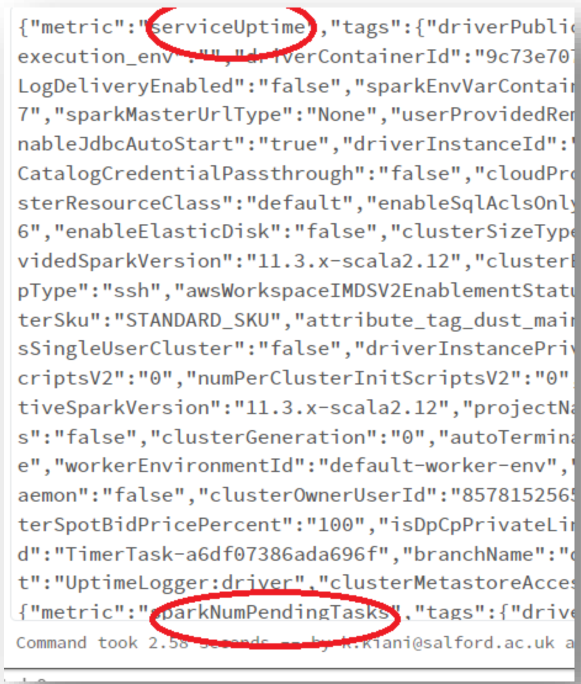
6. Extract the values containing the word “**metric**" from the logs/usage.json file.



```
Cmd 8

%sh
grep 'metric' logs/usage.json
```

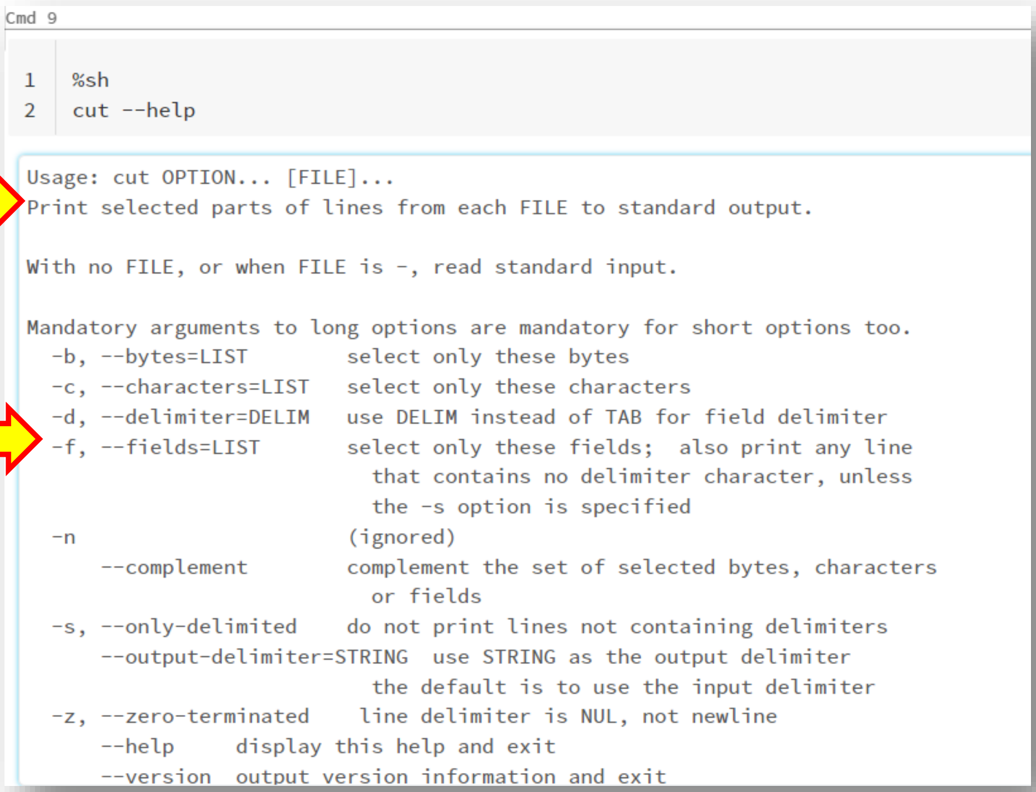
```
{"metric":"serviceUptime","tags":{"driverPublicDns":"ec2-18-246-63-110.us-west-2.c
execution_env":"","driverContainerId":"9c73e707330943f2a93a36105511fb9d","instance
LogDeliveryEnabled":"false","sparkEnvVarContainsSingleQuotes":"false","driverNodeT
7","sparkMasterUrlType":"None","userProvidedRemoteVolumeCount":"0","clusterUnityCa
nableJdbcAutoStart":"true","driverInstanceId":"i-0fbcdada2dfa9c51c","orgId":"22801
CatalogCredentialPassthrough":"false","cloudProvider":"AWS","clusterName":"kaveh 1
sterResourceClass":"default","enableSqlAclsOnly":"false","containerType":"LXC","at
6","enableElasticDisk":"false","clusterSizeType":"VM_CONTAINER","dataPlaneRegion":
videdSparkVersion":"11.3.x-scala2.12","clusterEbsVolumeType":"GENERAL_PURPOSE_SSD"
pType":"ssh","awsWorkspaceIMDSV2EnablementStatus":"false","containerZoneId":"us-we
terSku":"STANDARD_SKU","attribute_tag_dust_maintainer":"","ngrokNpipEnabled":"fals
sSingleUserCluster":"false","driverInstancePrivateIp":"10.172.253.26","sparkEnvVar
criptsV2":"0","numPerClusterInitScriptsV2":"0","enableJobsAutostart":"true","clust
tiveSparkVersion":"11.3.x-scala2.12","projectName":"driver","region":"us-west-2","
s":"false","clusterGeneration":"0","autoTerminationMinutes":"120","clusterId":"012
e","workerEnvironmentId":"default-worker-env","sparkEnvVarContainsNewline":"false"
aemon":"false","clusterOwnerUserId":"8578152565608834","privateLinkEnabled":"false
terSpotBidPricePercent":"100","isDpCpPrivateLinkEnabled":"false","runtimeEngine":"
d":"TimerTask-a6df07386ada696f","branchName":"development","userProvidedRemoteVolu
t":"UptimeLogger:driver","clusterMetastoreAccessType":"RDS_DIRECT","attribute_tag_
{"metric":"sparkNumPendingTasks","tags":{"driverPublicDns":"ec2-18-246-63-110.us-w
Command took 2.58 seconds -- by k.kiani@salford.ac.uk at 1/22/2023, 10:33:52 AM on kaveh 1
Cmd 9
```

7. Now we want to extract values of the “metric”.



```
{"metric":"serviceUptime","tags":{"driverPubli
execution_env":"","driverContainerId":"9c73e70
LogDeliveryEnabled":"false","sparkEnvVarContai
7","sparkMasterUrlType":"None","userProvidedRe
nableJdbcAutoStart":"true","driverInstanceId":
CatalogCredentialPassthrough":"false","cloudPr
sterResourceClass":"default","enableSqlAclsOnl
6","enableElasticDisk":"false","clusterSizeTyp
videdSparkVersion":"11.3.x-scala2.12","cluster
pType":"ssh","awsWorkspaceIMDSV2EnablementStat
terSku":"STANDARD_SKU","attribute_tag_dust_mai
sSingleUserCluster":"false","driverInstancePri
criptsV2":"0","numPerClusterInitScriptsV2":"0"
tiveSparkVersion":"11.3.x-scala2.12","projectN
s":"false","clusterGeneration":"0","autoTermin
e","workerEnvironmentId":"default-worker-env",
aemon":"false","clusterOwnerUserId":"857815256
terSpotBidPricePercent":"100","isDpCpPrivateLi
d":"TimerTask-a6df07386ada696f","branchName":"
t":"UptimeLogger:driver","clusterMetastoreAcce
{"metric":"sparkNumPendingTasks","tags":{"driv
Command took 2.58 seconds -- by k.kiani@salford.ac.uk a
```

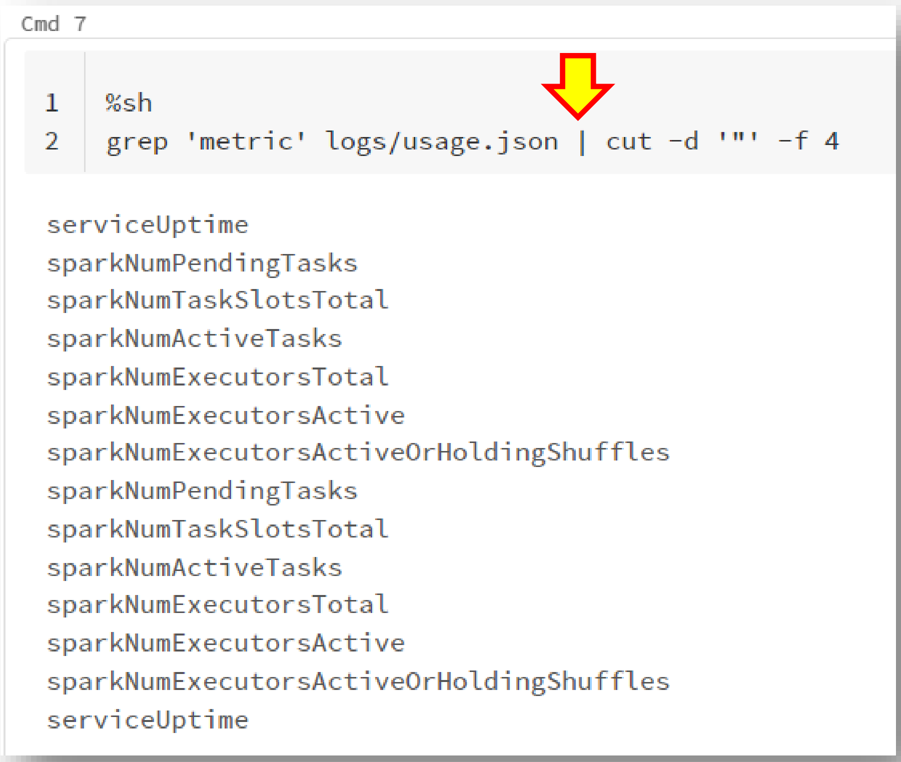
We want to use “cut” command. Run the following help command to learn “cut” functionality.



```
Cmd 9

%sh
cut --help
```

```
Usage: cut OPTION... [FILE]...
Print selected parts of lines from each FILE to standard output.

With no FILE, or when FILE is -, read standard input.

Mandatory arguments to long options are mandatory for short options too.
  -b, --bytes=LIST        select only these bytes
  -c, --characters=LIST   select only these characters
  -d, --delimiter=DELIM   use DELIM instead of TAB for field delimiter
  -f, --fields=LIST       select only these fields;  also print any line
                            that contains no delimiter character, unless
                            the -s option is specified
  -n                      (ignored)
      --complement        complement the set of selected bytes, characters
                            or fields
  -s, --only-delimited    do not print lines not containing delimiters
      --output-delimiter=STRING  use STRING as the output delimiter
                            the default is to use the input delimiter
  -z, --zero-terminated    line delimiter is NUL, not newline
      --help     display this help and exit
      --version  output version information and exit
```

Here is the command:

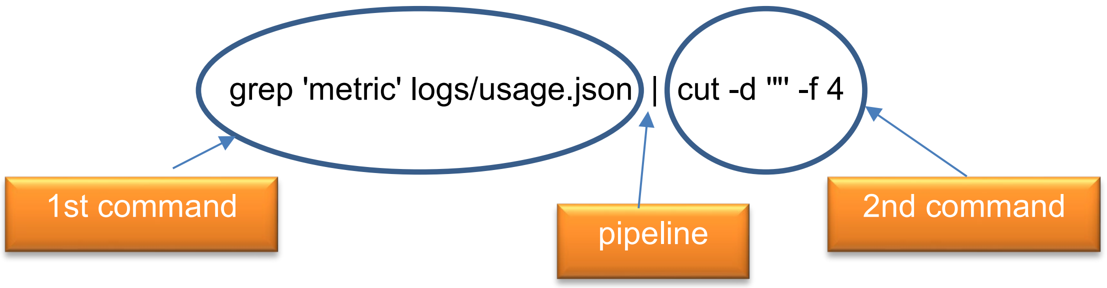

Cmd 7

```
%sh
grep 'metric' logs/usage.json | cut -d '"' -f 4
```

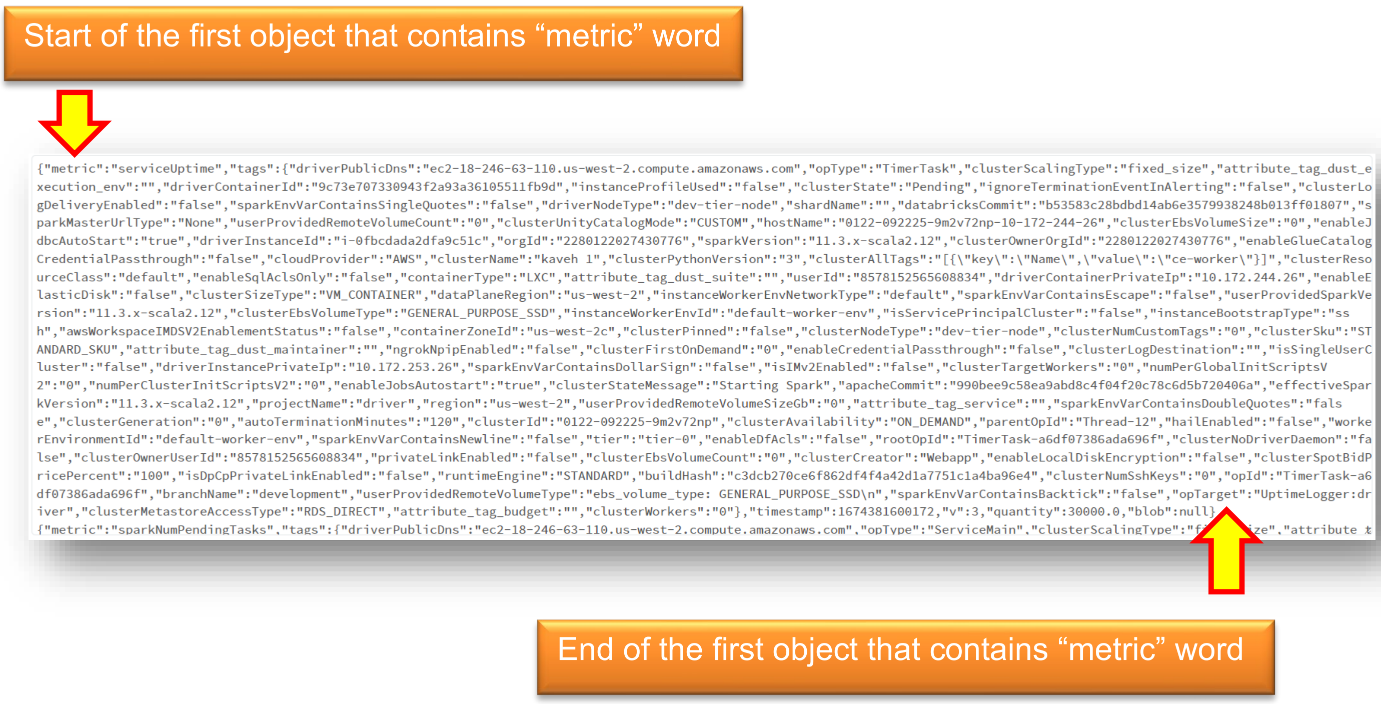
```
serviceUptime
sparkNumPendingTasks
sparkNumTaskSlotsTotal
sparkNumActiveTasks
sparkNumExecutorsTotal
sparkNumExecutorsActive
sparkNumExecutorsActiveOrHoldingShuffles
sparkNumPendingTasks
sparkNumTaskSlotsTotal
sparkNumActiveTasks
sparkNumExecutorsTotal
sparkNumExecutorsActive
sparkNumExecutorsActiveOrHoldingShuffles
serviceUptime
```

What is the usage of the vertical bar "I" in between of the code? "I" is the pipeline and like a pipe passes result of the first part to the second part. The vertical bar connects the commands together, making it possible to create a chain of related but separate processes.

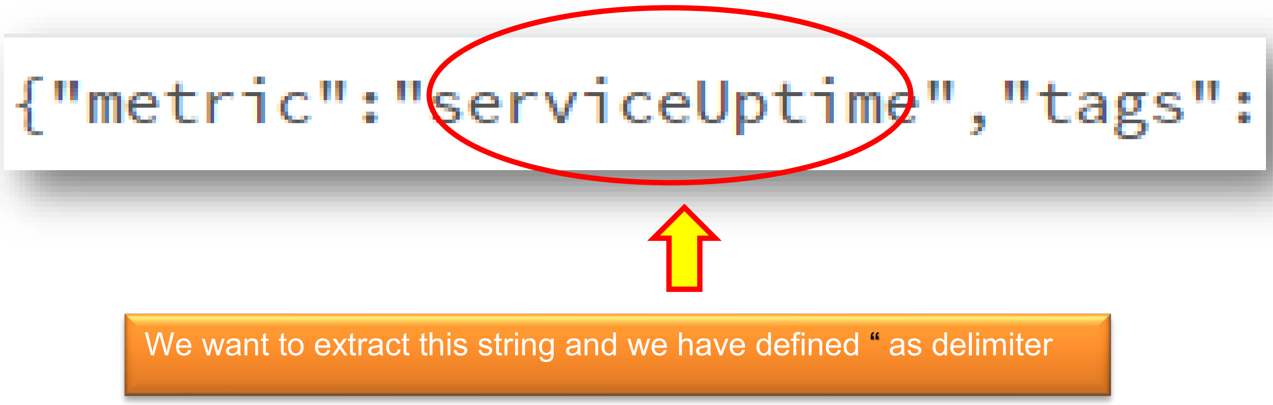
First part extracts the values containing the word "**metric**" from the logs/usage.json file and then second part receive outputs of the first part and extracts intended part of the inputs.



In the second command (cut -d '"' -f 4) we ask Linux to extract (or cut) based on " delimiter and give use the forth part of the each object. Let's take a close look to one of the objects that has been started with the word "metric".

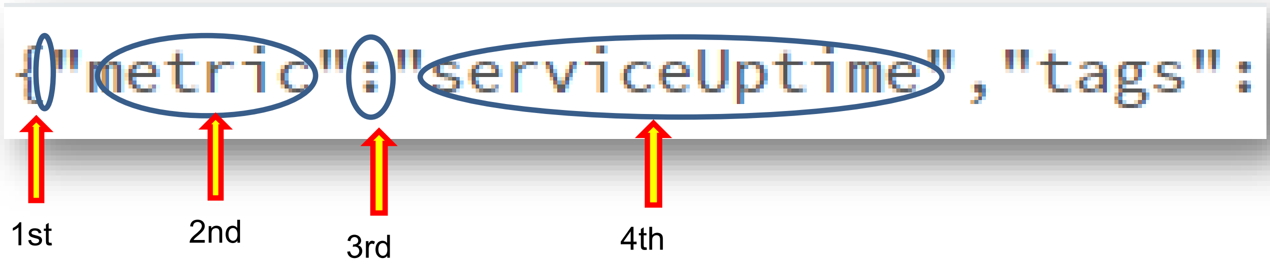

Start of the first object that contains “metric” word
End of the first object that contains “metric” word

```
{"metric":"serviceUptime","tags":{"driverPublicDns":"ec2-18-246-63-110.us-west-2.compute.amazonaws.com","opType":"TimerTask","clusterScalingType":"fixed_size","attribute_tag_dust_e
xecution_env":"","driverContainerId":"9c73e707330943f2a93a36105511fb9d","instanceProfileUsed":"false","clusterState":"Pending","ignoreTerminationEventInAlerting":"false","clusterLo
gDeliveryEnabled":"false","sparkEnvVarContainsSingleQuotes":"false","driverNodeType":"dev-tier-node","shardName":"","databricksCommit":"b53583c28bdbd14ab6e3579938248b013ff01807","s
parkMasterUrlType":"None","userProvidedRemoteVolumeCount":"0","clusterUnityCatalogMode":"CUSTOM","hostName":"0122-092225-9m2v72np-10-172-244-26","clusterEbsVolumeSize":"0","enableJ
dbcAutoStart":"true","driverInstanceId":"i-0fbcdada2dfa9c51c","orgId":"2280122027430776","sparkVersion":"11.3.x-scala2.12","clusterOwnerOrgId":"2280122027430776","enableGlueCatalog
CredentialPassthrough":"false","cloudProvider":"AWS","clusterName":"kaveh 1","clusterPythonVersion":"3","clusterAllTags":"[{\"key\":\"Name\",\"value\":\"ce-worker\"}]","clusterReso
urceClass":"default","enableSqlAclsOnly":"false","containerType":"LXC","attribute_tag_dust_suite":"","userId":"8578152565608834","driverContainerPrivateIp":"10.172.244.26","enableE
lasticDisk":"false","clusterSizeType":"VM_CONTAINER","dataPlaneRegion":"us-west-2","instanceWorkerEnvNetworkType":"default","sparkEnvVarContainsEscape":"false","userProvidedSparkVe
rsion":"11.3.x-scala2.12","clusterEbsVolumeType":"GENERAL_PURPOSE_SSD","instanceWorkerEnvId":"default-worker-env","isServicePrincipalCluster":"false","instanceBootstrapType":"ss
h","awsWorkspaceIMDSV2EnablementStatus":"false","containerZoneId":"us-west-2c","clusterPinned":"false","clusterNodeType":"dev-tier-node","clusterNumCustomTags":"0","clusterSku":"ST
ANDARD_SKU","attribute_tag_dust_maintainer":"","ngrokNpipEnabled":"false","clusterFirstOnDemand":"0","enableCredentialPassthrough":"false","clusterLogDestination":"","isSingleUserC
luster":"false","driverInstancePrivateIp":"10.172.253.26","sparkEnvVarContainsDollarSign":"false","isIMv2Enabled":"false","clusterTargetWorkers":"0","numPerGlobalInitScriptsV
2":"0","numPerClusterInitScriptsV2":"0","enableJobsAutostart":"true","clusterStateMessage":"Starting Spark","apacheCommit":"990bee9c58ea9abd8c4f04f20c78c6d5b720406a","effectiveSpar
kVersion":"11.3.x-scala2.12","projectName":"driver","region":"us-west-2","userProvidedRemoteVolumeSizeGb":"0","attribute_tag_service":"","sparkEnvVarContainsDoubleQuotes":"fals
e","clusterGeneration":"0","autoTerminationMinutes":"120","clusterId":"0122-092225-9m2v72np","clusterAvailability":"ON_DEMAND","parentOpId":"Thread-12","hailEnabled":"false","worke
rEnvironmentId":"default-worker-env","sparkEnvVarContainsNewline":"false","tier":"tier-0","enableDfAcls":"false","rootOpId":"TimerTask-a6df07386ada696f","clusterNoDriverDaemon":"fa
lse","clusterOwnerUserId":"8578152565608834","privateLinkEnabled":"false","clusterEbsVolumeCount":"0","clusterCreator":"Webapp","enableLocalDiskEncryption":"false","clusterSpotBidP
ricePercent":"100","isDpCpPrivateLinkEnabled":"false","runtimeEngine":"STANDARD","buildHash":"c3dcb270ce6f862df4f4a42d1a7751c1a4ba96e4","clusterNumSshKeys":"0","opId":"TimerTask-a6
df07386ada696f","branchName":"development","userProvidedRemoteVolumeType":"ebs_volume_type: GENERAL_PURPOSE_SSD\n","sparkEnvVarContainsBacktick":"false","opTarget":"UptimeLogger:dr
iver","clusterMetastoreAccessType":"RDS_DIRECT","attribute_tag_budget":"","clusterWorkers":"0"},"timestamp":1674381600172,"v":3,"quantity":30000.0,"blob":null}
{"metric":"sparkNumPendingTasks","tags":{"driverPublicDns":"ec2-18-246-63-110.us-west-2.compute.amazonaws.com","opType":"ServiceMain","clusterScalingType":"f[illegible]ze","attribute_
```


We want to extract this string and we have defined “ as delimiter

```
{"metric":"serviceUptime","tags":
```

(cut -d '"' -f 4) command says to extract the 4th part of the object where “ is delimiter (separator). Let’s count together:



```
{"metric":"serviceUptime","tags":
```

Challenge 1:

Now try to extract values for `"driverPublicDns:`

```
"driverPublicDns":"ec2-18-246-63-110.us-west-2.compute.amazonaws.com"
```

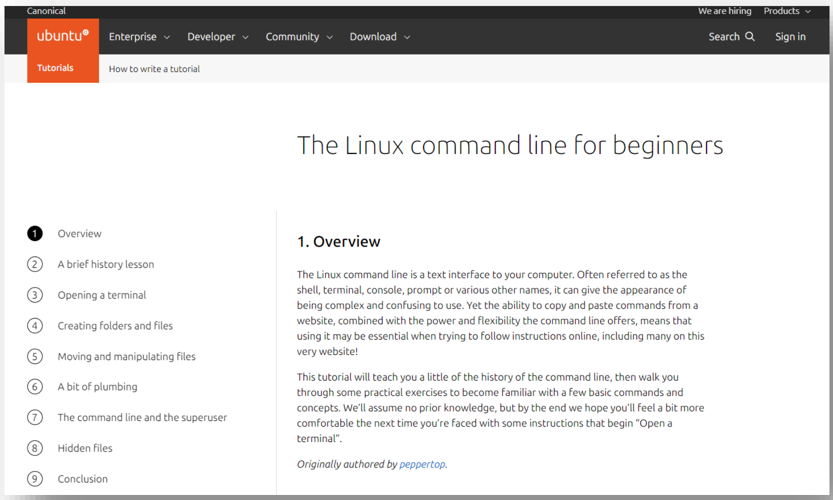
Challenge 2:

Can you think of a way to print each of the values only once?

4) Final touch

Ready to level up? Head over to this link and explore the website. It's a great way to test your skills, refresh your memory and learn more as much as you can and no need to learn every single detail of it for this workshop.

https://ubuntu.com/tutorials/command-line-for-beginners#1-overview



The Linux command line for beginners

1. Overview
2. A brief history lesson
3. Opening a terminal
4. Creating folders and files
5. Moving and manipulating files
6. A bit of plumbing
7. The command line and the superuser
8. Hidden files
9. Conclusion

1. Overview

The Linux command line is a text interface to your computer. Often referred to as the shell, terminal, console, prompt or various other names, it can give the appearance of being complex and confusing to use. Yet the ability to copy and paste commands from a website, combined with the power and flexibility the command line offers, means that using it may be essential when trying to follow instructions online, including many on this very website!

This tutorial will teach you a little of the history of the command line, then walk you through some practical exercises to become familiar with a few basic commands and concepts. We'll assume no prior knowledge, but by the end we hope you'll feel a bit more comfortable the next time you're faced with some instructions that begin "Open a terminal".

Originally authored by peppertop.

References and Resources:

https://docs.databricks.com/files/index.html

https://docs.databricks.com/notebooks/notebooks-use.html

https://ubuntu.com/tutorials/command-line-for-beginners#1-overview