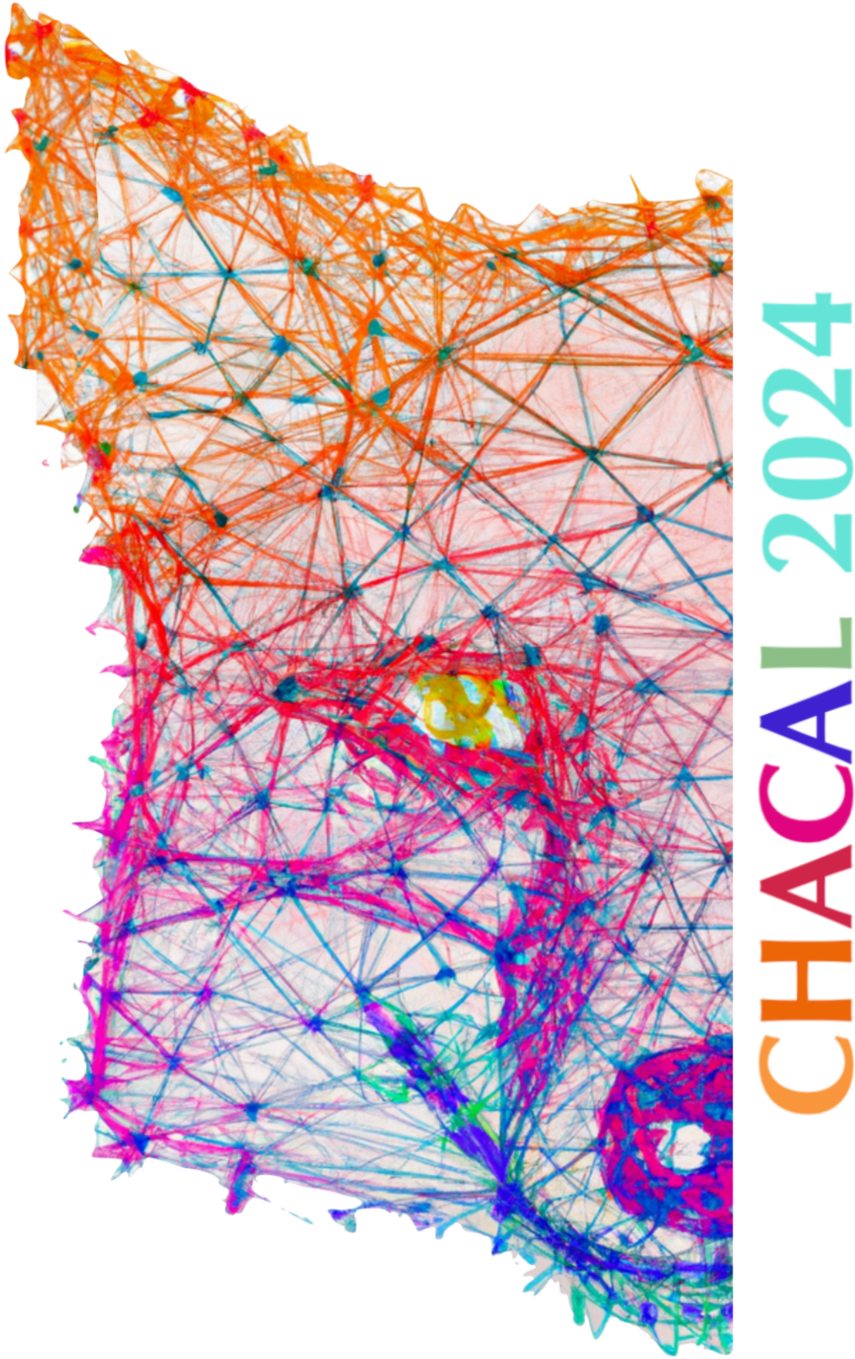
Scientific Computing Basics

What you'll need to make CHACAL a success

Louie Dartmoor Corpe (LPC Clermont)

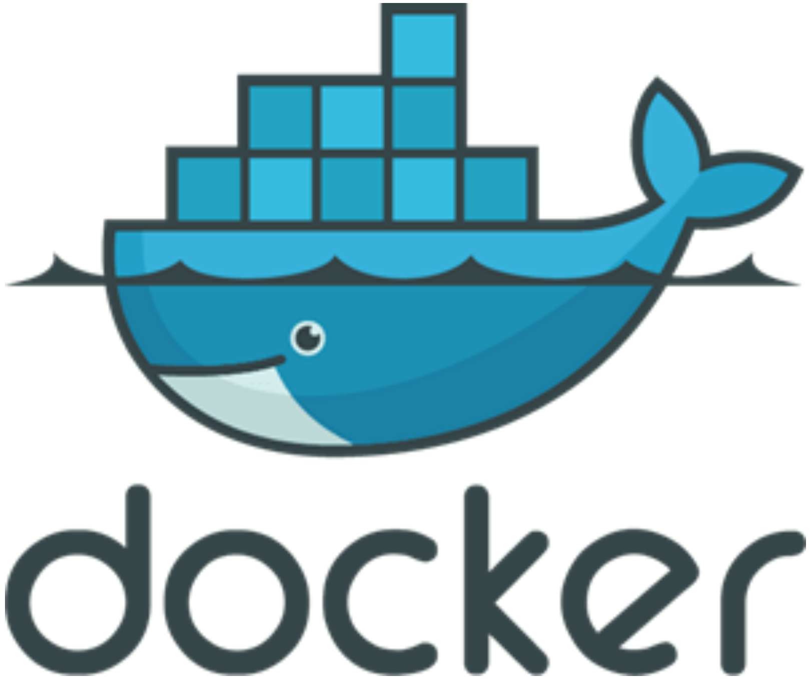
15 Jan 2024



Welcome to the tutorial

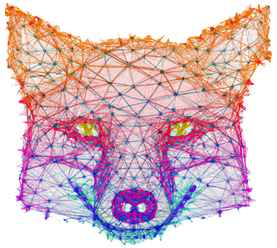
- We will now try to put into practice some of the things we saw this morning.
- It's also a trial run to make sure you have the core software set up that you will need over the course of CHACAL.
- Questions are welcome and encouraged.
- Also don't hesitate to work in groups and work together!

Docker and Bash



```
Last login: Tue Jan  9 17:27:34 on ttys004
louiecorpe_cern2@clratlmac02 ~ % 
```

Task 1. Install Docker desktop and download an image

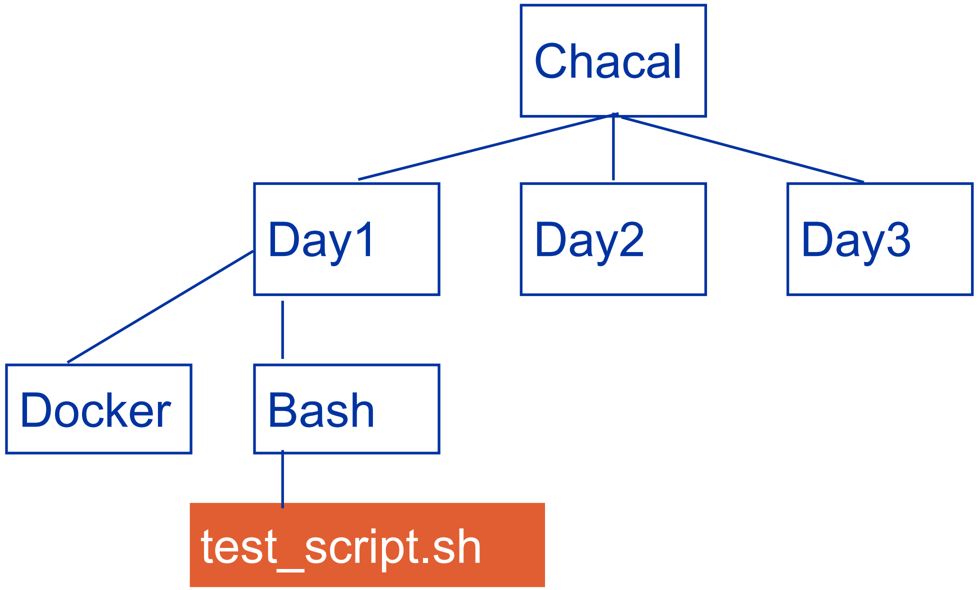


- You first job is to install Docker desktop.
- Follow the instructions here relative to your machine https://docs.docker.com/engine/install/
- Download the "`alpine:latest`" image
- Use the command line to open a container with that image in interactive mode
- Inside the image, install emacs and vim like this

```
apk add vim emacs
```

Task 2. Create some directories

- Make some directories and files in the following structure.



Chacal

Day1

Day2

Day3

Docker

Bash

test_script.sh

Task 3. Pipe to a file and copy

- Navigate to the /Chacal/Day1/Bash directory
- Append the list of environment variables to a file called "vars.txt"
- Create a new environment variable called "CHACALYEAR" and set it to value "24"
- Append the new list of environment variables to "vars2.txt"
- Copy both these files to /Chacal/Day1/Docker but rename them "varlist1.txt" and "varlist2.txt"

Task 4. Write a script

- Navigate to the /Chacal/Day1/Docker directory
- Using a text editor of your choice, write a script which prints the following to the screen:
  - "Welcome to CHACAL" followed by the year (from your env variable)
  - Print the operating system (you can obtain this from `cat /etc/os-release`)
  - Print the current directory
  - Print list of files in the current directory

Don't forget to exit the docker container!

Basic python

Task 5 - Getting started with Jupyter

- Install Jupyter notebook using "pip install jupyter"
- While you are at it, also install numpy, matplotlib and pandas
- Create a new notebook and call it "CHACAL24 Day 1"
- Play around with the functionality:
  - Can you figure out how to make some cells **text** and other cells **code**?
  - Load the numpy, matplotlib and pandas libraries

Task 6 - Basic python

- Look up the 10 largest cities in South Africa by population
- Fill a dictionary with keys as the city names, and the value as population
- Use a loop to print the South African cities and their percentage of the total South African population
- Write a function which takes a list of city names, and prints only those containing "c". Use that on the list of keys from your dictionary.

Task 7 - Basic python

- Look up the 10 largest cities in South Africa by population
- Fill a dictionary with keys as the city names, and the value as population
- Use a loop to print the South African cities and their percentage of the total South African population
- Write a function which takes a list of city names, and prints only those containing "c". Use that on the list of keys from your dictionary.

Task 7 - Basic python

- Look up the 10 largest cities in South Africa by population
- Fill a dictionary with keys as the city names, and the value as population
- Use a loop to print the South African cities and their percentage of the total South African population
- Write a function which takes a list of city names, and prints only those containing "c". Use that on the list of keys from your dictionary.

Task 7 - Numpy, Dataframes and Matplotlib

- On the indico page, there are two csv files, one each for the top cities in France and South Africa
- Import pandas and open a dataframe for each of those two files.
- Show the top 20 cities of each country using **dataframe.head()**
- Where does Clermont-Ferrand rank? And Johannesburg ?
- Use a vectorised operations **(NO LOOPS)** to calculate the mean longitude and latitude of French and South African cities

Task 8 - Numpy, Dataframes and Matplotlib

- Plot the distributions of the two countries' city populations on the same graph (use plt.hist).
- Use a mask to select only the entries with more than 100 000 people in each dataframe
- Make a 2D scatter plot with longitude on the x-axis and latitude on the y-axis, featuring both the countries' large cities.
- Make the same graph but where the marker size corresponds to population.
- Add a new column in each dataframe to calculate the distance with respect to the largest city ($\sqrt{\Delta(\text{latitude}) - \Delta(\text{longitude})}$.
- What is the average value of this distance in France ? In South Africa ?