

Centre for  
Research  
Training



# SFI Centre for Research Training in **GENOMICS DATA SCIENCE**

HOST INSTITUTION



**NUI Galway**  
OÉ Gaillimh

PARTNER INSTITUTIONS



University College Cork, Ireland  
Coláiste na hOllscoile Corcaigh



University College Dublin  
An Coláiste Ollscoile, Baile Átha Cliath



Trinity College Dublin  
Coláiste na Tríonóide, Baile Átha Cliath  
The University of Dublin



RCSI



**QUEEN'S  
UNIVERSITY  
BELFAST**



# Introduction to Linux

Speaker: **Declan Bennett**

[BennettD@universityofgalway.ie](mailto:BennettD@universityofgalway.ie)

# Who am I

- 2016 - BSc Biotechnology -> psychiatric genetics project
- 2017 - MSc Biomedical Genomics, Genetic variation in the somatic mutation rate
- 2017-2018 – Bioinformatician EMBL-EBI – Accelerating medicines project
- 2018- PhD bioinformatics. Inference of somatic mutation in 200,000 UK biobank exomes.





# Preliminary advice

- Data analysis can be very frustrating, you will make mistakes, and get error messages:
- Expect to spend a large part of your time on Google / forums or learning to use new tools / techniques...
- Don't run things blindly, always make sure you know how tools / packages work, the stats / biases behind them...
- Data analysis can be as experimental as wet-lab science!
- Tools and applications are constantly evolving, best practices are extremely hard to come by. Computation however doesn't consume samples / reagents, so don't be afraid to try new things...



# History of UNIX



# What is Linux/Unix

Unix (originally developed at Bell labs in 1960s/70s) is a family of operating systems with some powerful features:

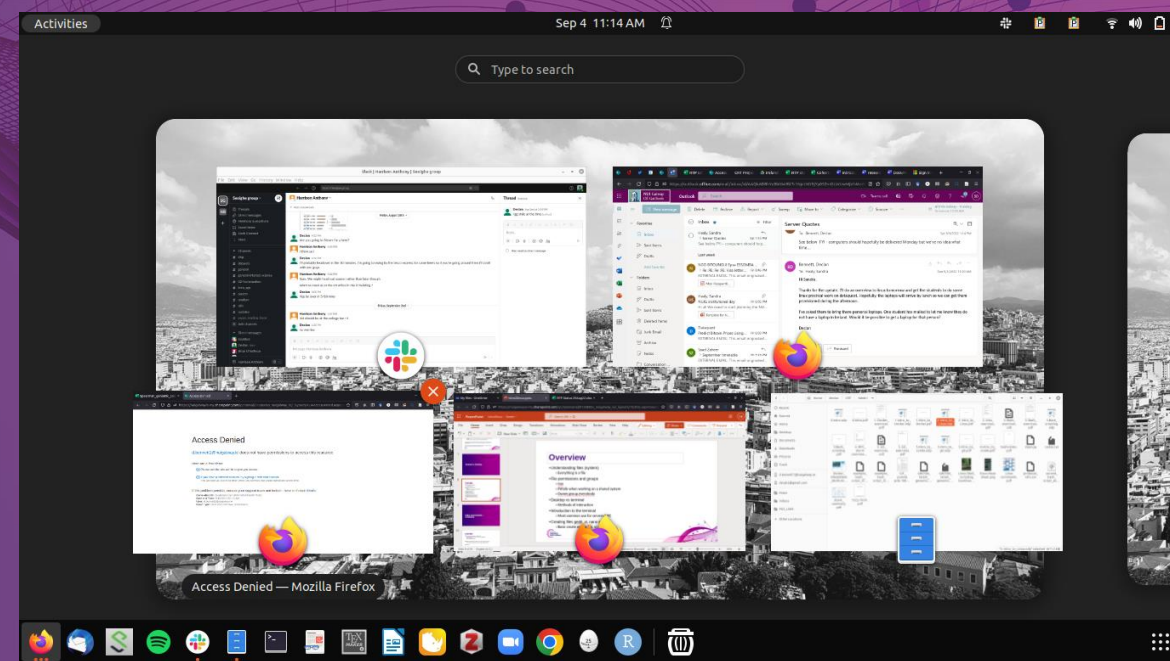
- Stable / Secure - Generally less prone to crashes / hacks
- Efficient multitasking - Designed for a multiuser environment
- Minimalist, modular code (“Do one thing and do it well”) written mostly in C – portable
- Unix shell – command line interpreter/interface (CLI), user enters text in a window to execute commands
- Unified File System – “everything is a file” (documents/directories/devices/)

Linux is an open-source Unix-like OS which comes in various distributions - RedHat, Fedora, Debian, etc., etc. Modern variants typically use X11 Windows System plus a desktop environment to provide a GUI.

Most compute clusters (supercomputers) run headless Unix / Linux OS – we usually need to use these types of systems to handle large-scale genomics analyses.



# Terminal vs Desktop



```
declan@somu: ~  
declan@somu  
-----  
OS: Ubuntu 21.10 x86_64  
Host: Latitude 5590  
Kernel: 5.13.0-52-generic  
Uptime: 20 mins  
Packages: 4041 (dpkg), 25 (s  
Shell: bash 5.1.8  
Resolution: 1366x768  
DE: GNOME 40.5  
WM: Mutter  
WM Theme: Adwaita  
Theme: Yaru [GTK2/3]  
Icons: DMZ-Black [GTK2/3]  
Terminal: gnome-terminal  
CPU: Intel i7-8650U (8) @ 4.  
GPU: Intel UHD Graphics 620  
Memory: 4546MiB / 31964MiB  
declan@somu:~$ |
```



# Overview

- ❑ Understanding files (system)
  - Everything is a file
- ❑ File permissions and groups
  - rwx
  - Pitfalls when working on a shared system
  - Owner, group, everybody
- ❑ Desktop vs terminal
  - Methods of interaction
- ❑ Introduction to the terminal
  - Most common use for servers/HPC
- ❑ Creating files gedit, vi, nano etc..
  - Basic create empty file, vi shortcuts





# Files

```
declan@somu:~/work/thesis/chapter2$ tree -d -L 1 /
/
├── bin -> usr/bin
├── boot
├── cdrom
├── dev
├── etc
├── home
├── lib -> usr/lib
├── lib32
├── lib64 -> usr/lib64
├── libx32 -> usr/libx32
├── lost+found
├── media
├── mnt
├── opt
├── proc
├── root
├── run
├── sbin -> usr/sbin
├── snap
├── srv
├── sys
├── tmp
├── usr
└── var
```

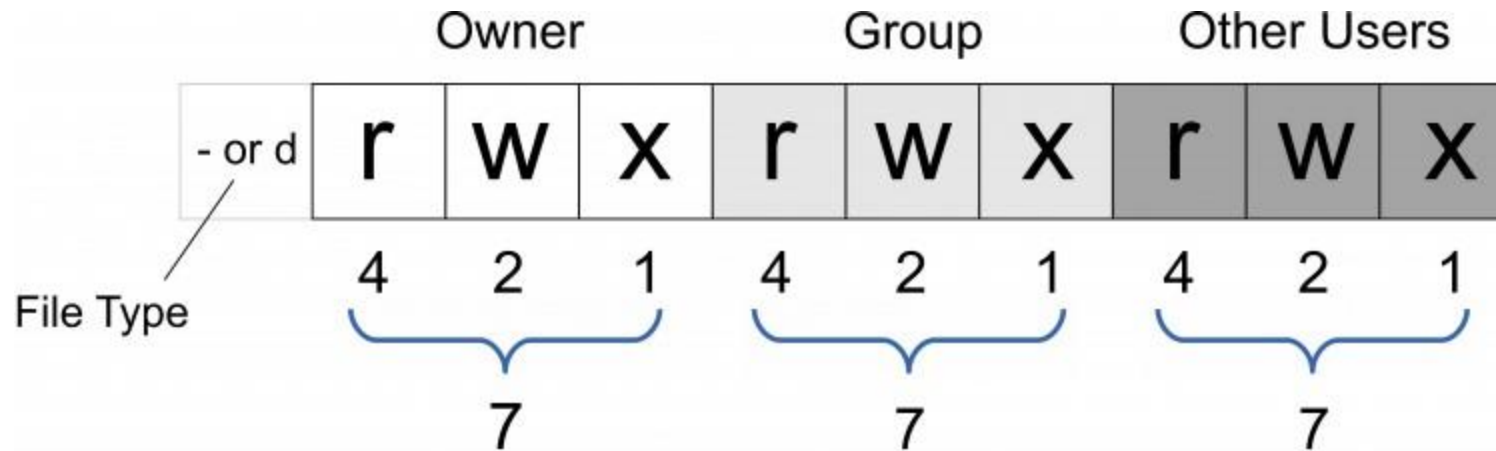
```
declan@somu:~/work/thesis/chapter2$ l
total 643M
-rw-rw-r-- 1 declan declan 4.1K May 24 10:44 Age_correlations.R
-rw-rw-r-- 1 declan declan 884 May 23 18:25 asymmetry_probes.R
drwxrwxr-x 2 declan declan 4.0K May 25 16:08 batch_corr
-rw-rw-r-- 1 declan declan 665 Jan 14 2022 Check_chr5_assoc.R
drwxrwxr-x 2 declan declan 4.0K May 25 13:00 data
drwxrwxr-x 3 declan declan 4.0K May 24 12:34 expression
drwxrwxr-x 5 declan declan 4.0K Feb 27 2022 gwas
-rw-rw-r-- 1 declan declan 643M Jan 14 2022 old_counts.norm
drwxrwxr-x 2 declan declan 4.0K May 19 11:42 pheno_norm
drwxrwxr-x 2 declan declan 4.0K Apr 5 11:48 pipeline
-rw-rw-r-- 1 declan declan 8.8K May 25 09:39 probe_asymmetry_df.txt
```

## Multiple user system

```
dbennett@lugh:/data/Seoighe_data$ ls -l
total 984
drwxrwxr-x 3 dbennett seoighe_group 45 Nov 15 2019 1KG
drwxrwxr-x 4 dbennett dbennett 42 Dec 10 2019 Apples
drwxrwxr-x 4 dbennett seoighe_group 8192 Apr 19 11:00 GTEx
drwxr-x--- 7 scleary seoighe_group 111 Feb 24 2022 ICGC
drwxrwxr-x 2 scleary seoighe_group 167 May 14 2020 mapability
drwxrwxr-x 2 scleary scleary 10 Nov 20 2019 Pennychuik
```



# File permissions



# Paths, environments + commands



# Overview

- Paths, environment, bashrc + profile, alias'
  - How does the computer know where an executable file is
  - How to specify
  - Some example bash alias'
  
- Example commands cd, ls, mkdir, rm, top, less, cat, grep, zcat, pipe
  - Moving about, making files, directories, zipping, peaking at files etc...
  
- Exercises



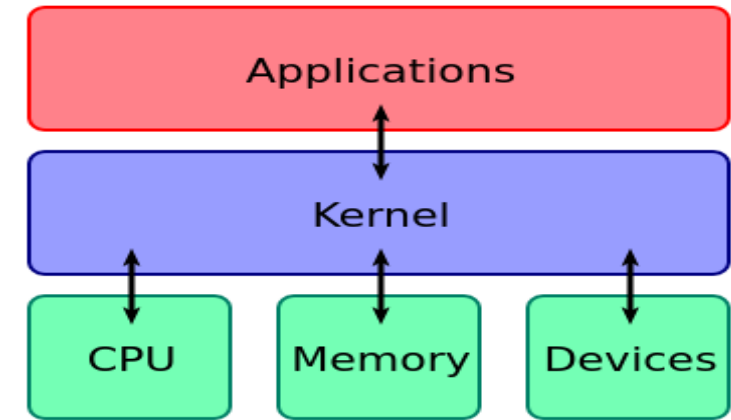


# Processes



# Overview

- ☐ All instructions from outside of the kernel space are executed in the context of processes
- ☐ A process can be seen as a set of instructions with controlled data attached to it
- ☐ The `top` command can be used to list these processes
- ☐ The processes information is stored under `/proc/PID/`



# Git



# "FINAL".doc



FINAL.doc!



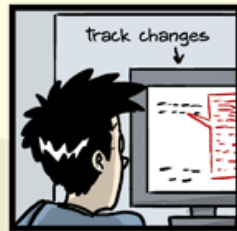
FINAL\_rev.2.doc



FINAL\_rev.6.COMMENTS.doc



FINAL\_rev.8.comments5.  
CORRECTIONS.doc



FINAL\_rev.18.comments7.  
corrections9.MORE.30.doc



FINAL\_rev.22.comments49.  
corrections.10.##\$%WHYDID  
ICOMETOGRADSCHOOL?????.doc

JORGE CHAM © 2012

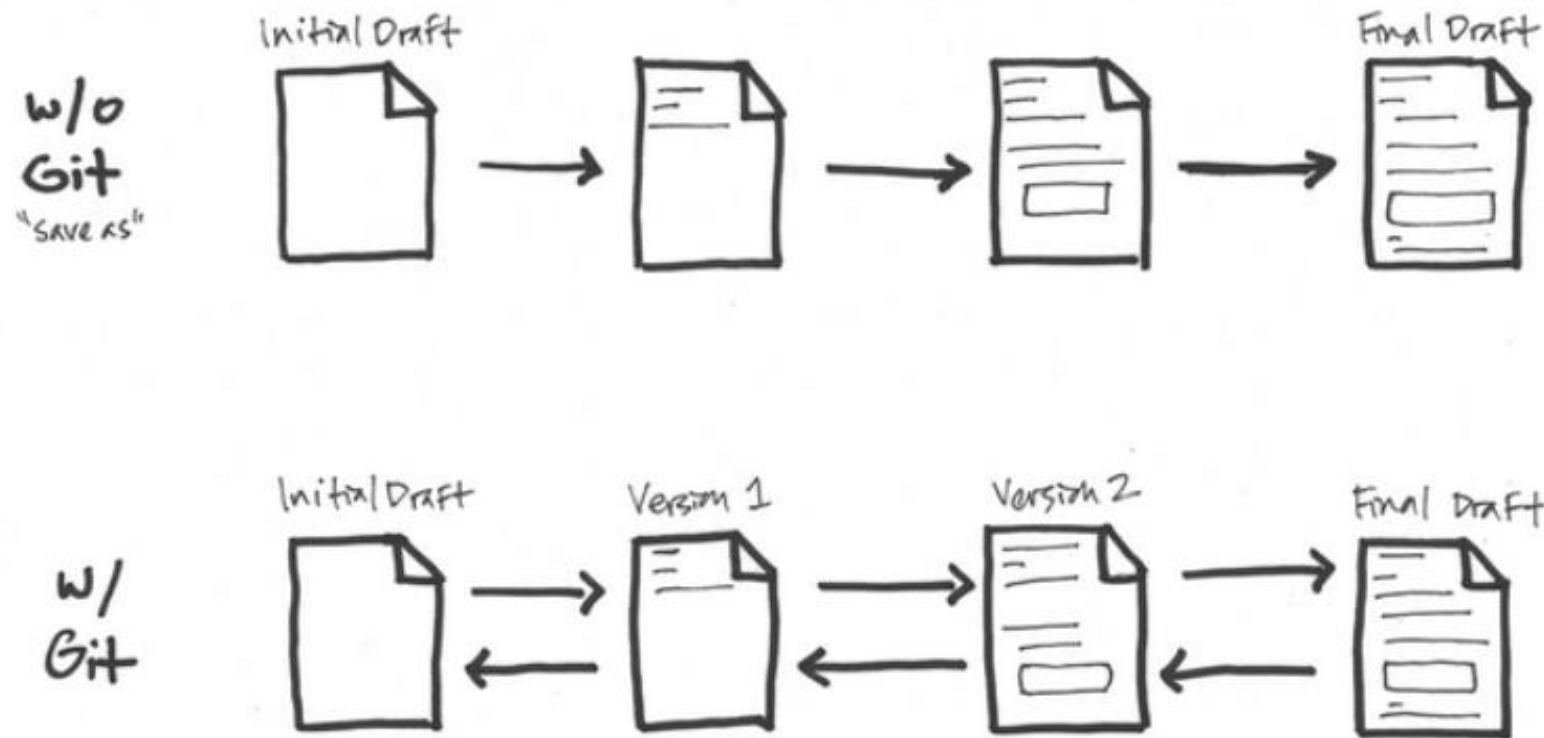
WWW.PHDCOMICS.COM





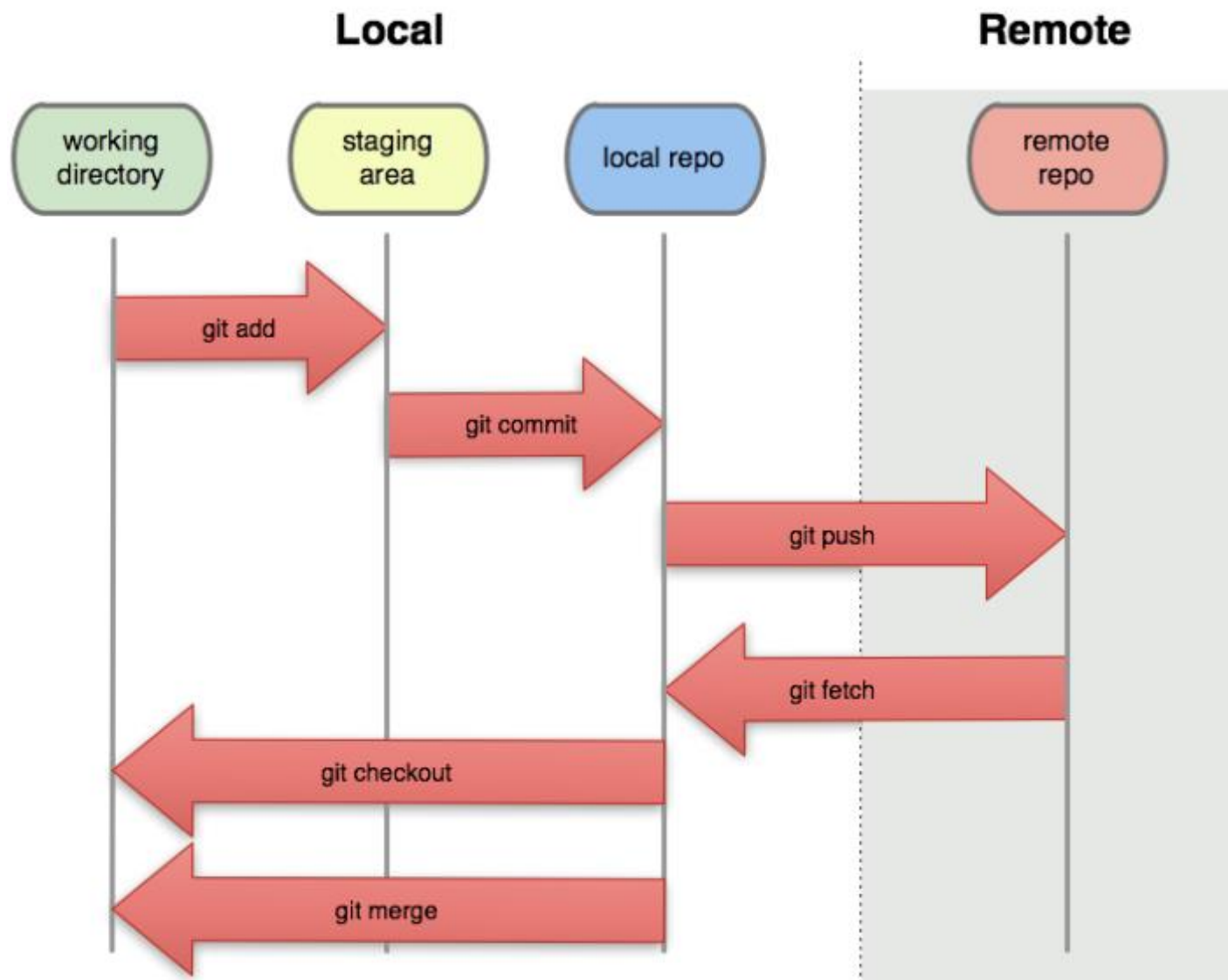
- Methodology in software development that ensures all changes to a software project (and code) are tracked in time.
- Advantages
  - you can revert back to specific 'versions' of your code
  - collaboration becomes practical, as specific changes and associated contributors are tracked
- The most commonly used version control systems is Git



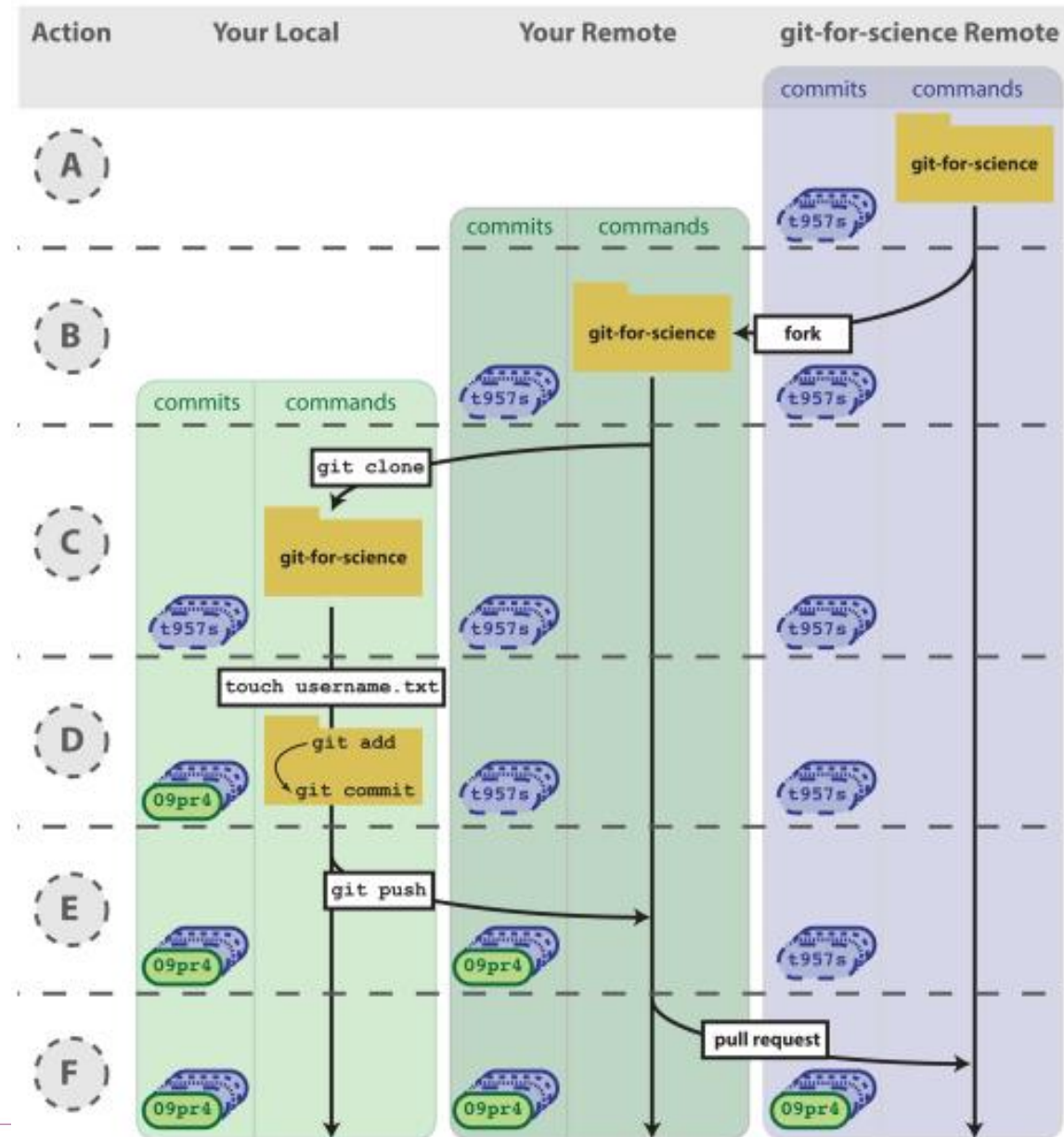


'Edits' etc. are easily forgotten - with *git* all changes are logged

# Version Control with Git



# Collaborative software





# Github readme



# Introduction to bash scripting



# Containers



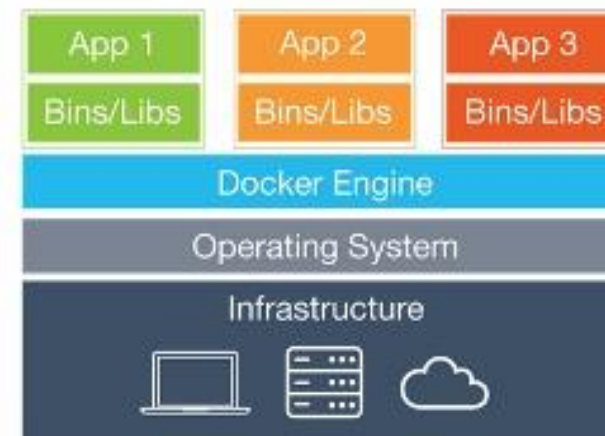


# Overview

- Difference between virtual machines and containers



Virtual Machines

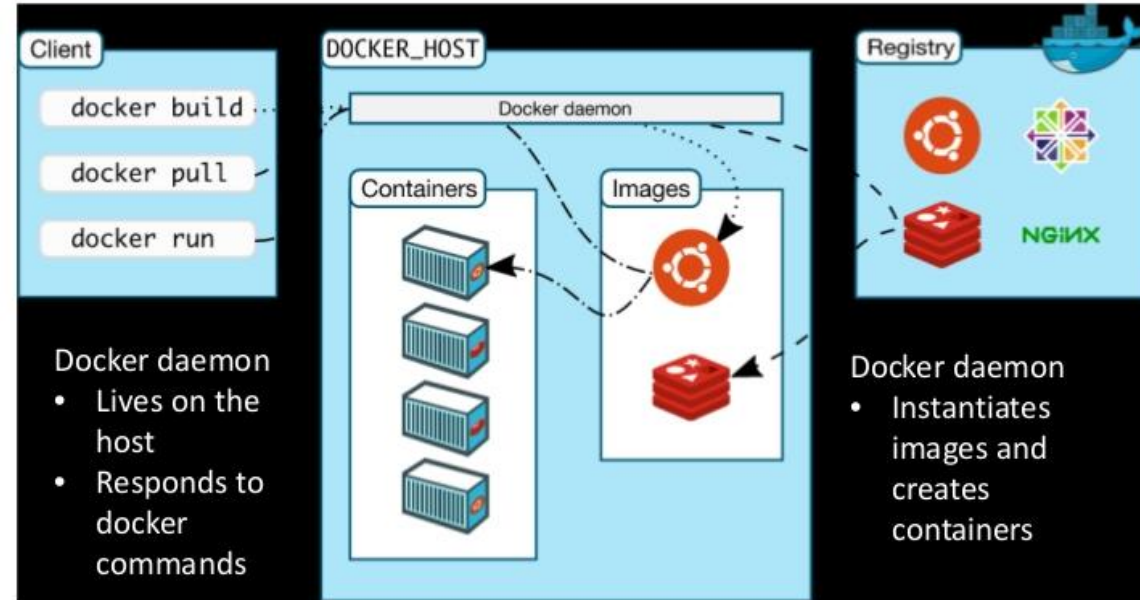


Containers



# Core concepts

- Docker Image: read-only template with instructions for creating a container
- Docker Container: a runnable instance of an image
- Docker Registry: 'App-store' for Docker images. Docker is configured to use Docker Hub by default.
- Dockerfile: set of instructions to build an image



# Core commands and options

command	description
<code>docker images</code> <code>docker history image</code>  <code>docker inspect image...</code>	list all local images show the image history (list of ancestors) show low-level infos (in json format)
<code>docker tag image tag</code>	tag an image
<code>docker commit container image</code>  <code>docker import url - [tag]</code>	create an image (from a container) create an image (from a tarball)
<code>docker rmi image...</code>	delete images

command	description
<code>docker create image [ command ]</code> <code>docker run image [ command ]</code>	create the container = <code>create</code> + <code>start</code>
<code>docker rename container new_name</code> <code>docker update container</code>	rename the container update the container config
<code>docker start container...</code> <code>docker stop container...</code> <code>docker kill container...</code> <code>docker restart container...</code>	start the container graceful <sup>2</sup> stop kill (SIGKILL) the container = <code>stop</code> + <code>start</code>
<code>docker pause container...</code> <code>docker unpause container...</code>	suspend the container resume the container
<code>docker rm [ -f<sup>3</sup> ] container...</code>	destroy the container

<sup>2</sup>send SIGTERM to the main process + SIGKILL 10 seconds later

<sup>3</sup>-f allows removing running containers (= `docker kill` + `docker rm`)





**Back up your data regularly !!!**