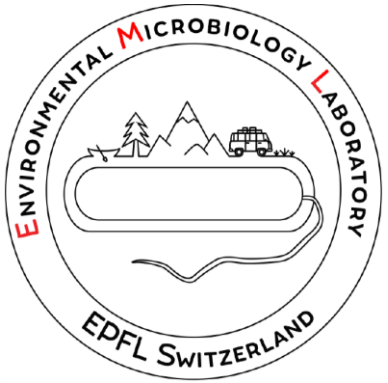


HPC & metaomics

ENV-621

Nicolas Jacquemin

EPFL

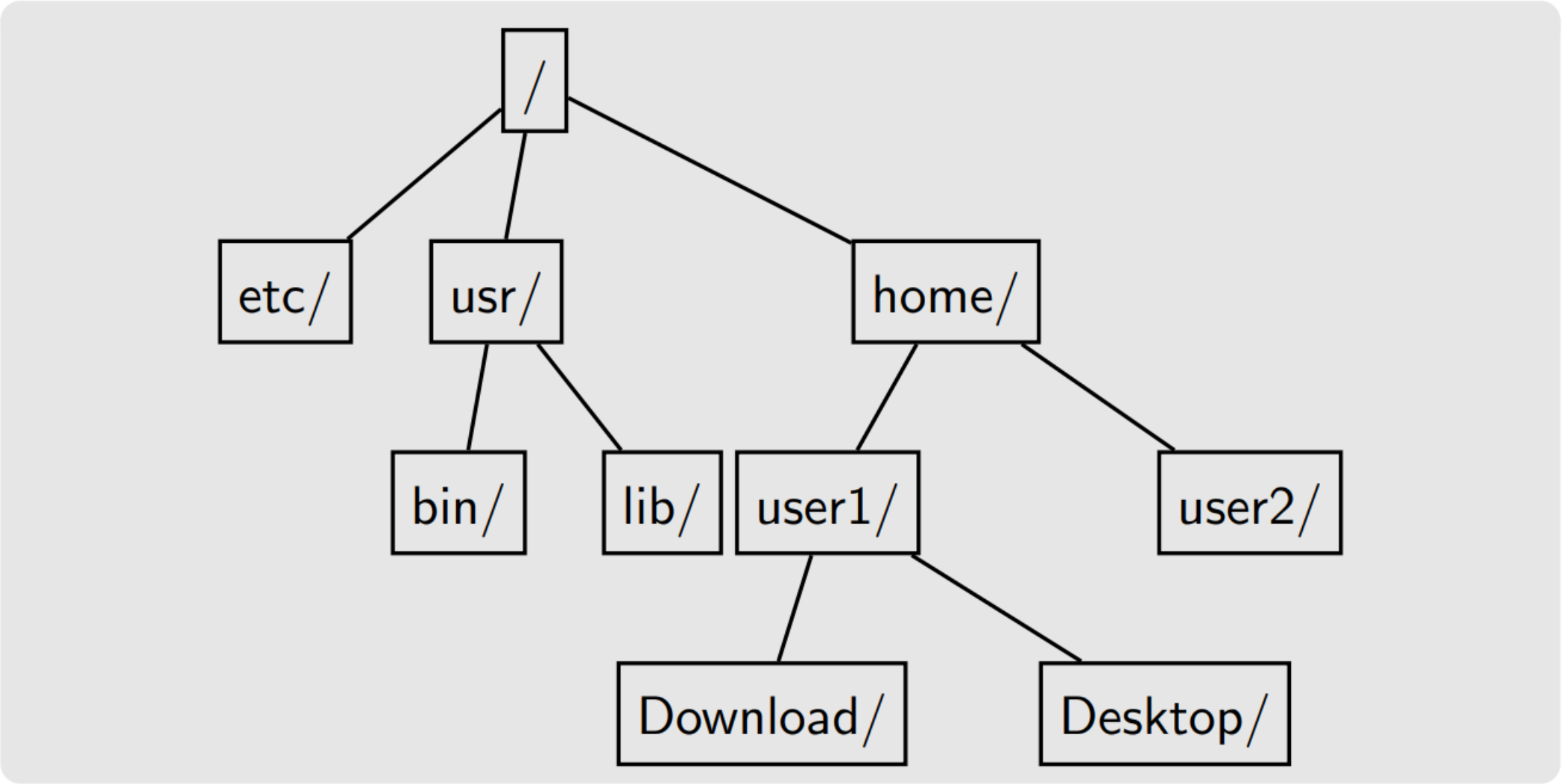
INFORMATIQUE SCIENTIFIQUE & SUPPORT APPLICATIF

SCITAS

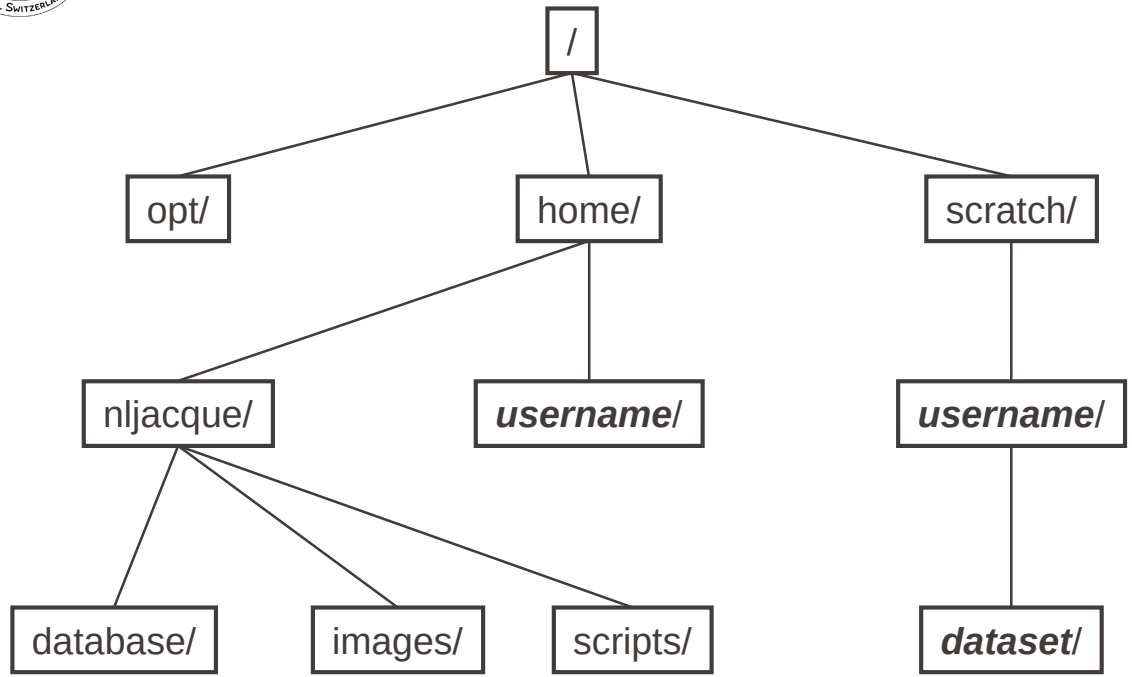
SCIENTIFIC IT AND APPLICATION SUPPORT

https://scitas-doc.epfl.ch/

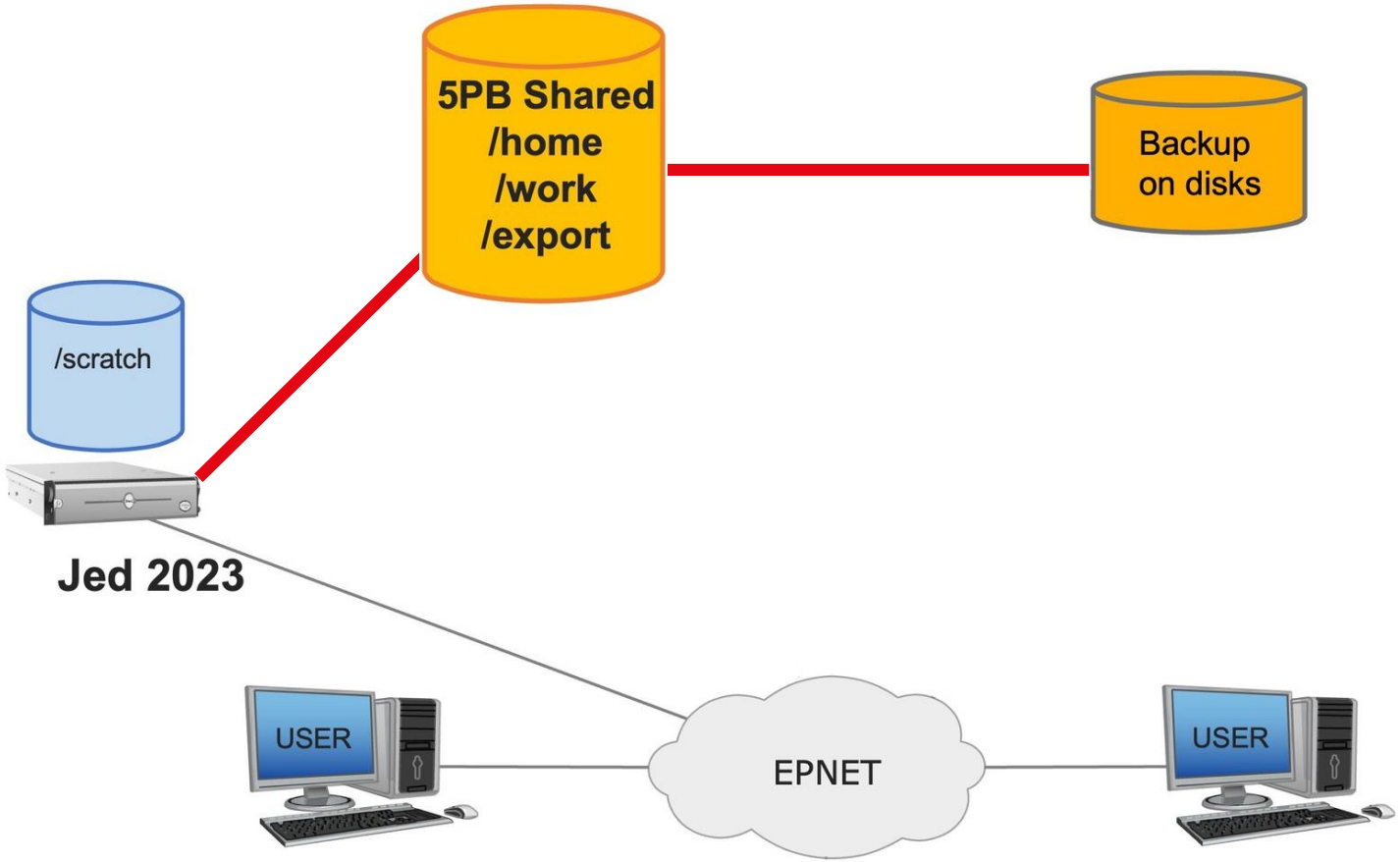
UNIX - Filesystem



SCITAS - Filesystem



SCITAS - Filesystem - GPFS

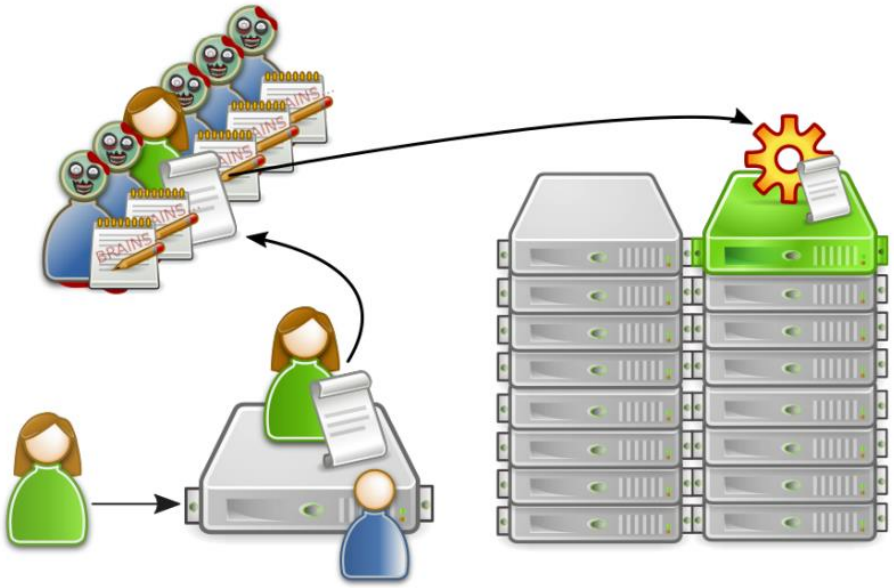


HPC - File system

File system, Environment variable	Purpose	Type, Size	Backup, Snapshots	Lifetime	Cleanup	Quota	Available for
/home $HOME	Store source files, input data, small files. Globally accessible from login and compute nodes.	GPFS 100TB	Backup to disks and snapshots	Account lifetime	No cleanup	100GB/user	All users
/work $WORK	Collaboration space for a group. Common software, result files and data sets. Globally accessible from login and compute nodes	GPFS 2.0PB	Backup upon request, at cost price. Snapshots	Group lifetime	No cleanup	50GB* /group	All EPFL units (upon request) Not available for Bachelor/Master students.
/scratch $SCRATCH	Temporary huge result files. Accessible from the frontend and compute nodes within one cluster.	Jed: GPFS 481TB	No backup, no snapshots	2 weeks	Automatic cleanup of files (see Scratch automatic cleanup)	A priori no quota	
/tmp/${SLURM_JOB_ID} $TMPDIR	Temporary, local file space for jobs on compute nodes. Not available on login nodes.	Local to node.	No backup, no snapshots	Job execution	Content is deleted after job ends	No	All users

- https://scitas-doc.epfl.ch/user-guide/using-clusters/file-system/

HPC - Job - Slurm



sbatch

This is the fundamental command used to submit jobs to the batch system. Normally returns immediately, as all it does is read the script and check your requirements.

Workflow

A typical workflow looks like this:

- create a job-script
- submit it to the batch system (with `sbatch`)
- *it will get executed*
- look at the output

The job **will wait in the queue** until resources are available to run it.

HPC – Sbatch directives

```
#!/bin/bash
#SBATCH --chdir /scratch/<username>/
#SBATCH --nodes 1
#SBATCH --ntasks 1
#SBATCH --cpus-per-task 28
#SBATCH --mem 120G
#SBATCH --time 00:30:00
#SBATCH --account scitas-courses
#SBATCH --reservation intro2clusters

./runme_xeon64
```

#SBATCH --something

This is how you tell Slurm what resources the job needs.

HPC – Sbatch directives

```
#!/bin/bash
#SBATCH --chdir /scratch/<username>/
#SBATCH --nodes 1
#SBATCH --ntasks 1
#SBATCH --cpus-per-task 28
#SBATCH --mem 120G
#SBATCH --time 00:30:00
#SBATCH --account scitas-courses
#SBATCH --reservation intro2clusters

./runme_xeon64
```

The number of nodes per job

Examples:

```
--nodes 1
--nodes 64
```

If not specified then the default is 1.

HPC – Sbatch directives

```
#!/bin/bash
#SBATCH --chdir /scratch/<username>/
#SBATCH --nodes 1
#SBATCH --ntasks 1
#SBATCH --cpus-per-task 28
#SBATCH --mem 120G
#SBATCH --time 00:30:00
#SBATCH --account scitas-courses
#SBATCH --reservation intro2clusters

./runme_xeon64
```

The number of MPI tasks per job

Examples:

```
--ntasks 1
--ntasks 256
```

If not specified then the default is 1.

HPC – Sbatch directives

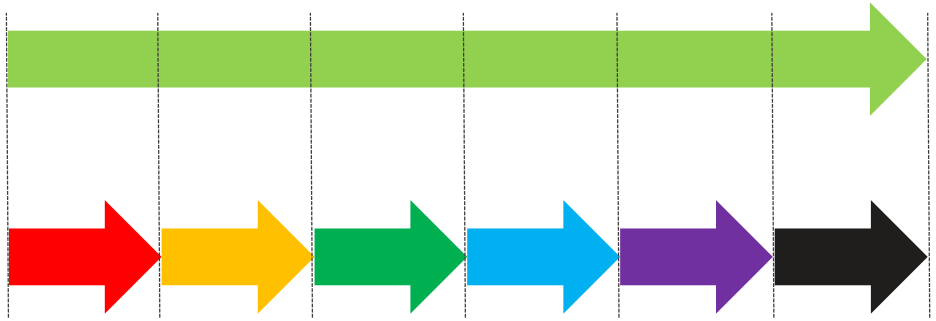
```
#!/bin/bash
#SBATCH --chdir /scratch/<username>/
#SBATCH --nodes 1
#SBATCH --ntasks 1
#SBATCH --cpus-per-task 28
#SBATCH --mem 120G
#SBATCH --time 00:30:00
#SBATCH --account scitas-courses
#SBATCH --reservation intro2clusters

./runme_xeon64
```

The number of CPUs per task for multithreaded applications

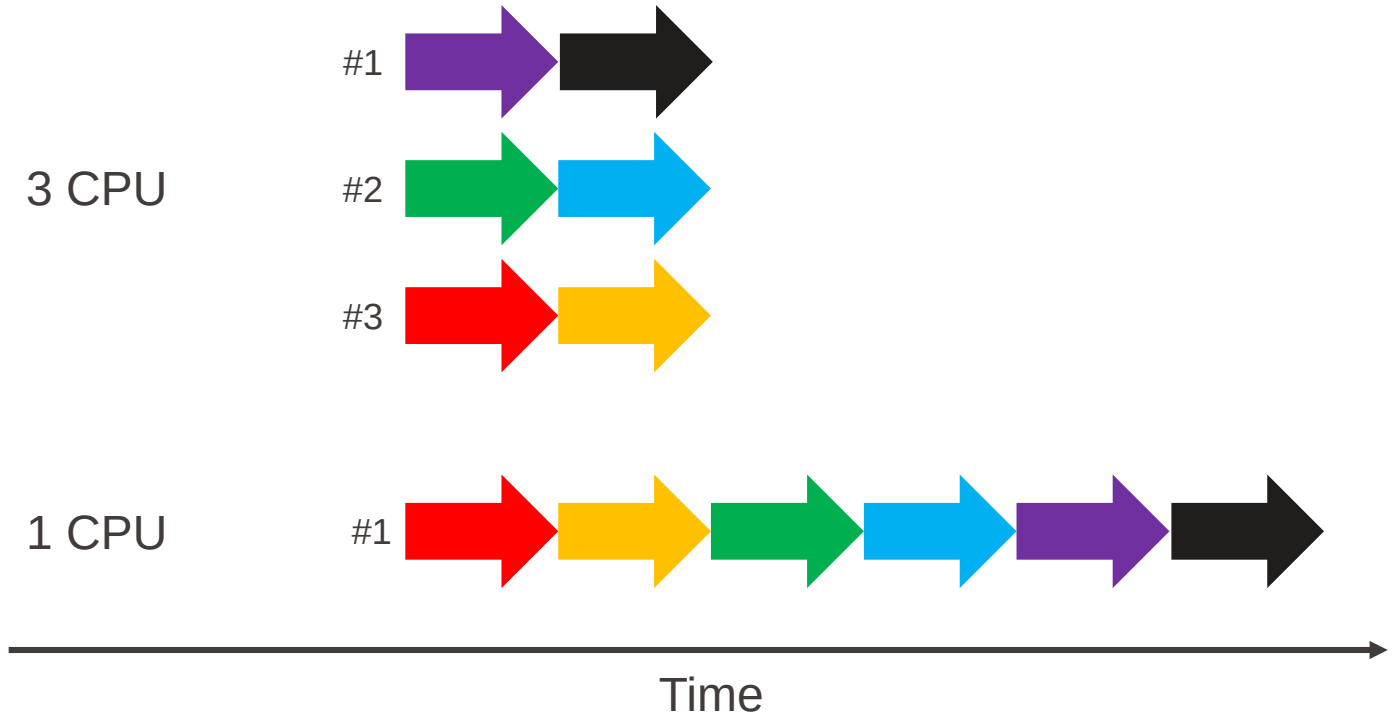
Examples:

```
--cpus-per-task 1
--cpus-per-task 28
```

If not specified then the default is 1.
Cannot be more than the number of cores/cpus in a compute node!



6 Threads



HPC – Sbatch directives

```
#!/bin/bash
#SBATCH --chdir /scratch/<username>/
#SBATCH --nodes 1
#SBATCH --ntasks 1
#SBATCH --cpus-per-task 28
#SBATCH --mem 120G
#SBATCH --time 00:30:00
#SBATCH --account scitas-courses
#SBATCH --reservation intro2clusters

./runme_xeon64
```

The required memory per node

Examples:

```
--mem 4096M
--mem 120G
```

If not specified then the default is 4096MB per CPU.

Beware of edge cases!

For example, on `fidis` if you ask for 128G, you are targetting 192G or 256G nodes, as our *128G* nodes only have 125G.

HPC – Sbatch directives

```
#!/bin/bash
#SBATCH --chdir /scratch/<username>/
#SBATCH --nodes 1
#SBATCH --ntasks 1
#SBATCH --cpus-per-task 28
#SBATCH --mem 120G
#SBATCH --time 00:30:00
#SBATCH --account scitas-courses
#SBATCH --reservation intro2clusters

./runme_xeon64
```

How long will your job run for?

Examples:

```
--time 06:00:00
--time 2-23
```

If not specified then the default is 15 minutes.

HPC – Sbatch directives

```
#!/bin/bash
#SBATCH --chdir /scratch/<username>/
#SBATCH --nodes 1
#SBATCH --ntasks 1
#SBATCH --cpus-per-task 28
#SBATCH --mem 120G
#SBATCH --time 00:30:00
#SBATCH --account scitas-courses
#SBATCH --reservation intro2clusters

./runme_xeon64
```

Flag to put in the submission scripts

#SBATCH --reservation=bioinformatics_meta-omics1

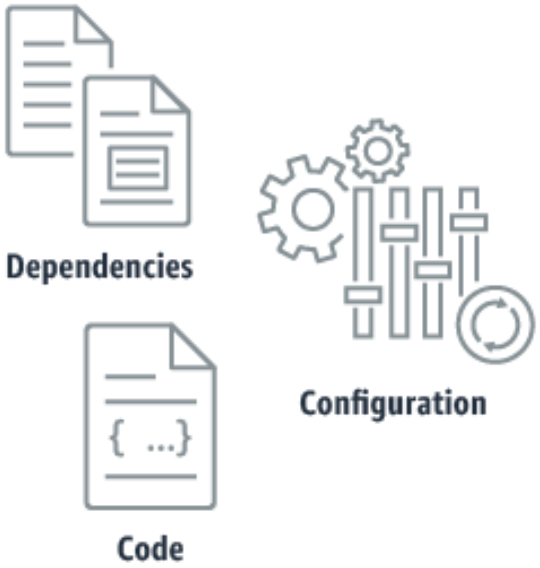
HPC – Sbatch directives

```
#!/bin/bash
#SBATCH --chdir /scratch/<username>/
#SBATCH --nodes 1
#SBATCH --ntasks 1
#SBATCH --cpus-per-task 28
#SBATCH --mem 120G
#SBATCH --time 00:30:00
#SBATCH --account scitas-courses
#SBATCH --reservation intro2clusters

./runme_xeon64
```

Your commands

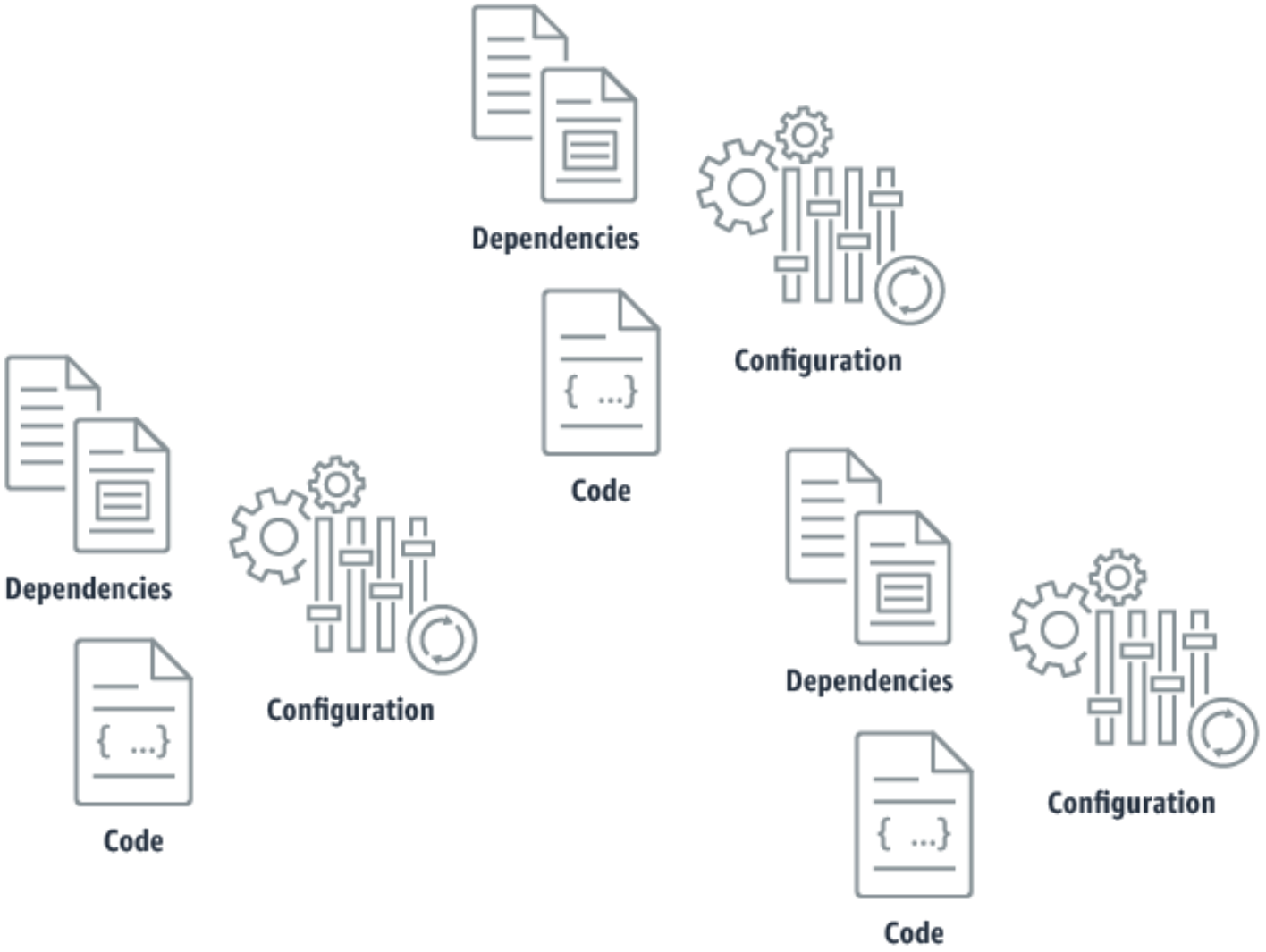
HPC – Software installation

- FastQC
- MultiQC
- Fastp
- MEGAHIT
- SPAdes
- seqkit
- BBMap
- Strobealign
- CONCOCT
- MetaBAT2
- DASTool
- DRep
- CheckM
- CheckM2
- GTDB-Tk
- METABOLIC
- CoverM



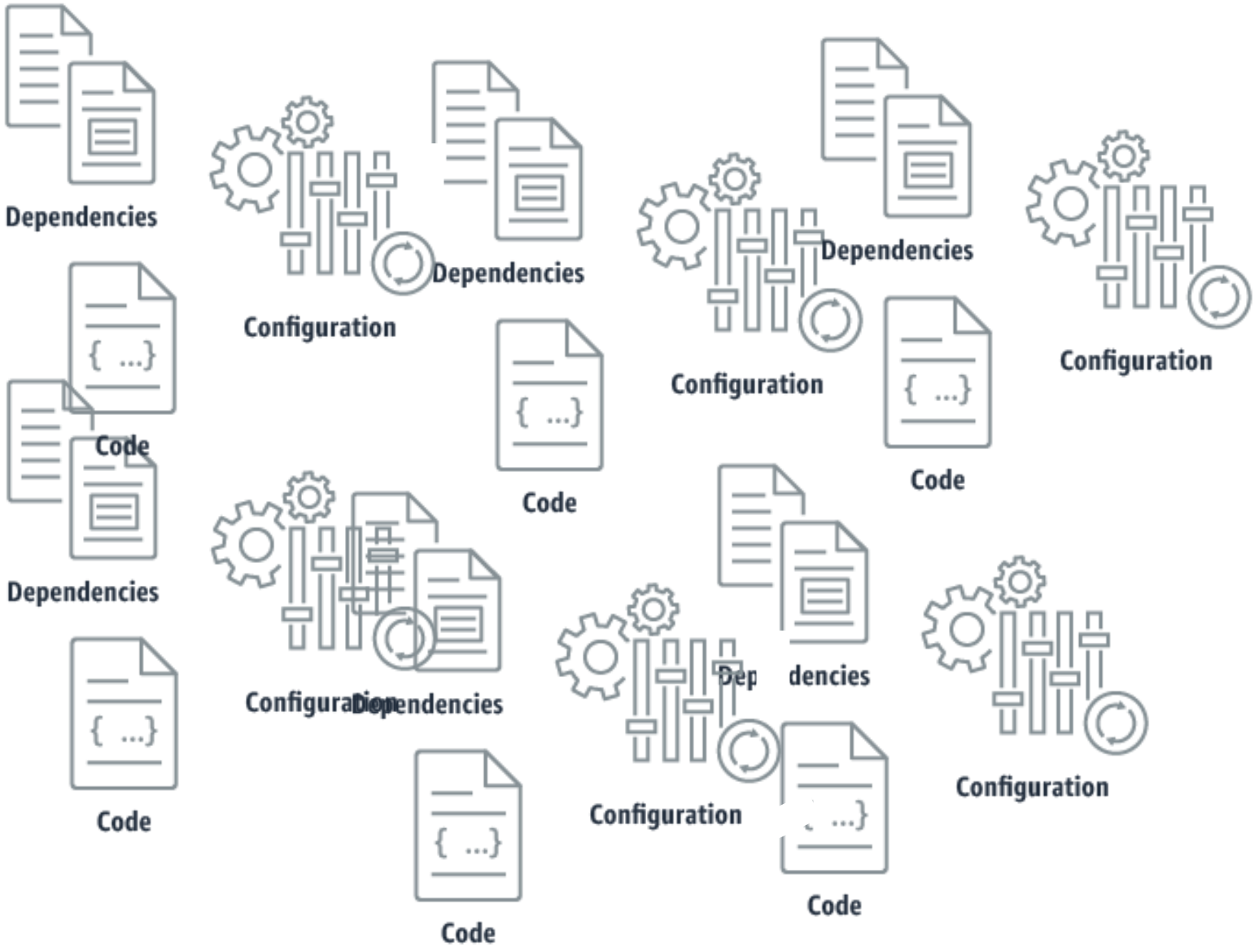
HPC – Software installation

- FastQC
- MultiQC
- Fastp
- MEGAHIT
- SPAdes
- seqkit
- BBMap
- Strobealign
- CONCOCT
- MetaBAT2
- DASTool
- DRep
- CheckM
- CheckM2
- GTDB-Tk
- METABOLIC
- CoverM



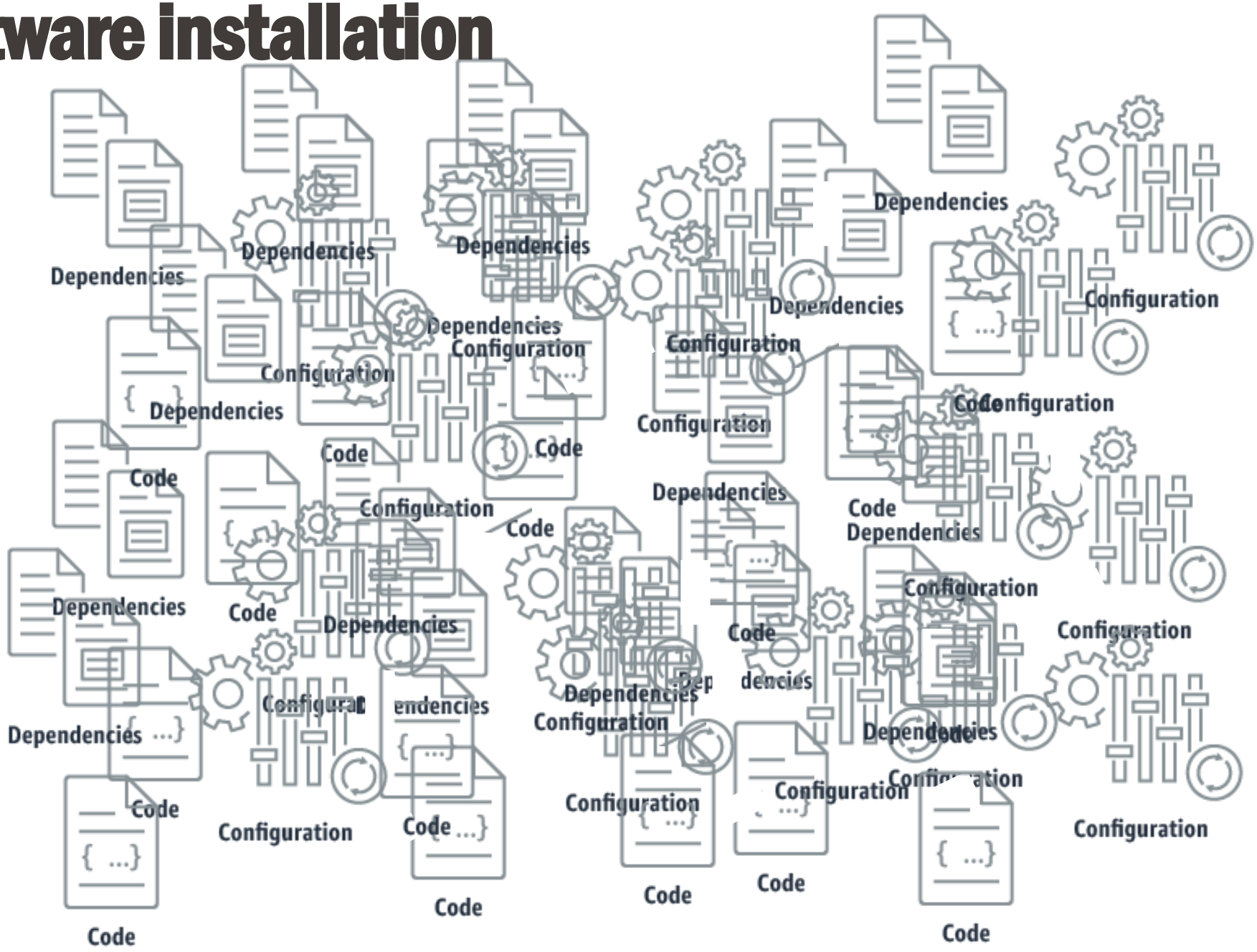
HPC – Software installation

- FastQC
- MultiQC
- Fastp
- MEGAHIT
- SPAdes
- seqkit
- BBMap
- Strobealign
- CONCOCT
- MetaBAT2
- DASTool
- DRep
- CheckM
- CheckM2
- GTDB-Tk
- METABOLIC
- CoverM



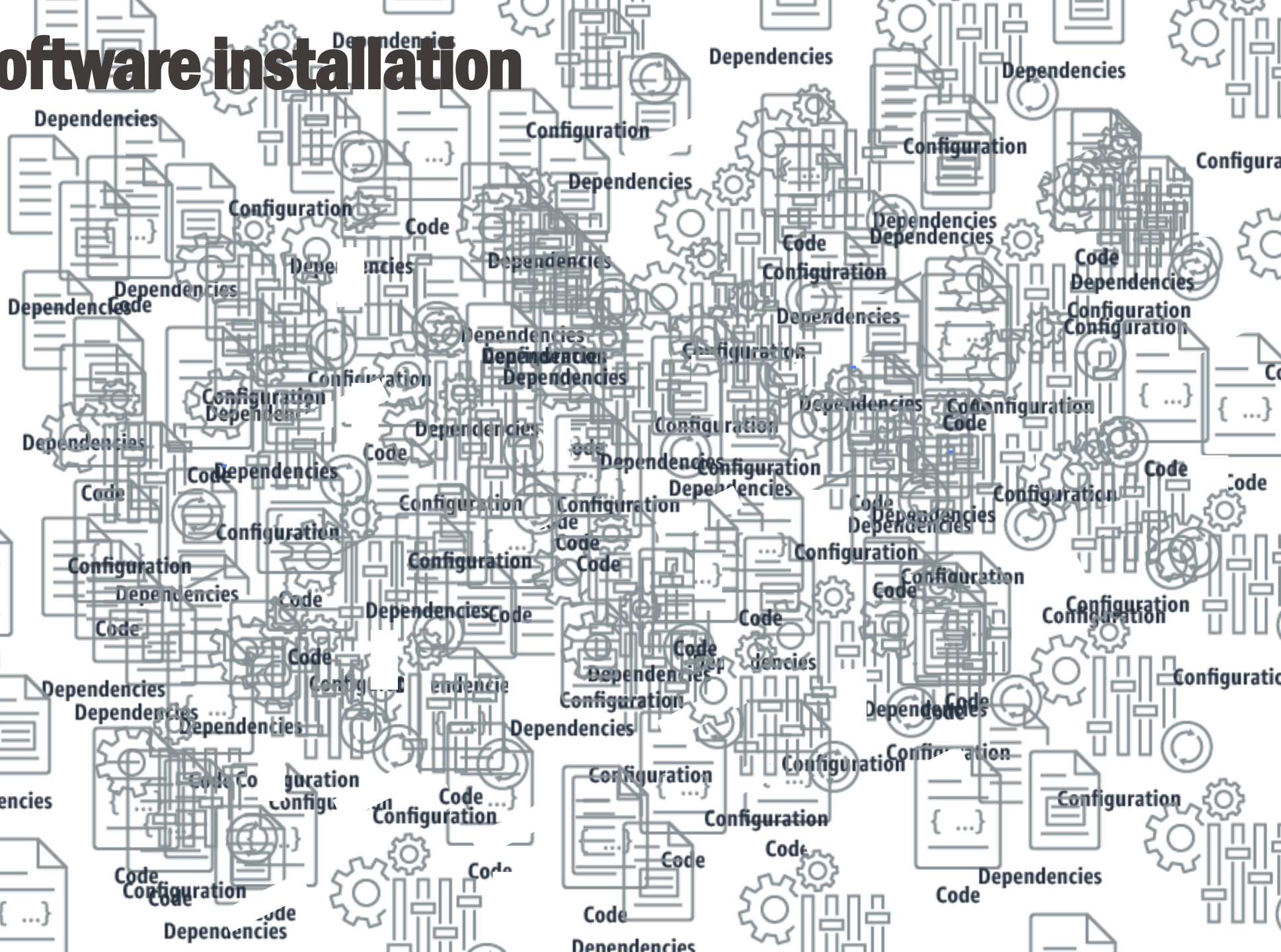
HPC – Software installation

- FastQC
- MultiQC
- Fastp
- MEGAHIT
- SPAdes
- seqkit
- BBMap
- Strobealign
- CONCOCT
- MetaBAT2
- DASTool
- DRep
- CheckM
- CheckM2
- GTDB-Tk
- METABOLIC
- CoverM



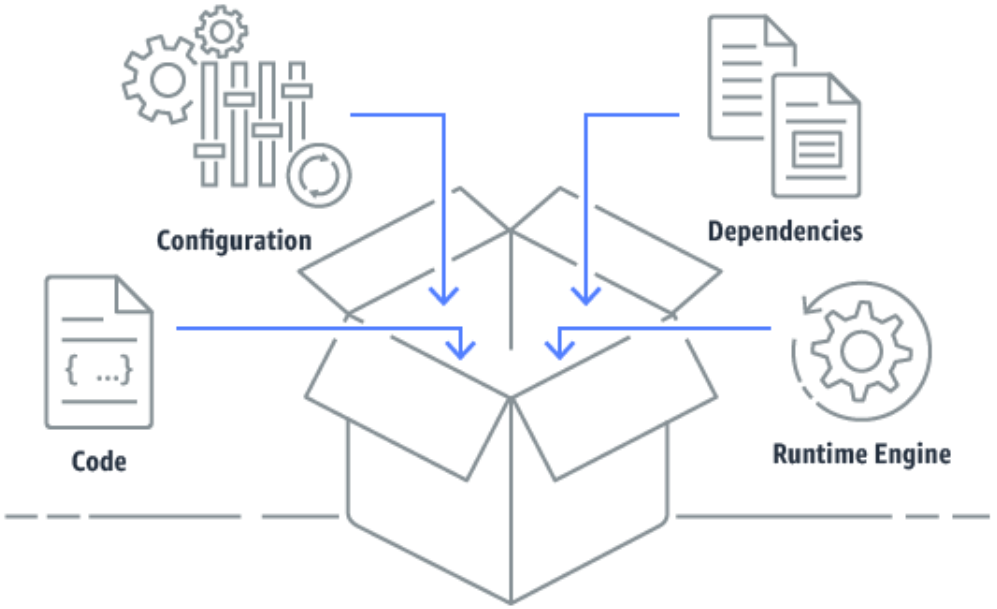
HPC – Software installation

- FastQC
- MultiQC
- Fastp
- MEGAHIT
- SPAdes
- seqkit
- BBMap
- Strobealign
- CONCOCT
- MetaBAT2
- DASTool
- DRep
- CheckM
- CheckM2
- GTDB-Tk
- METABOLIC
- CoverM

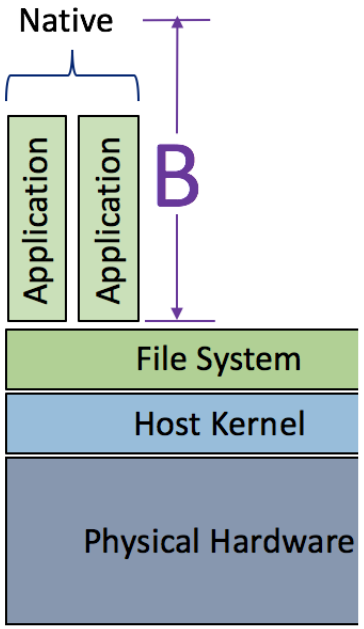


HPC – Software installation

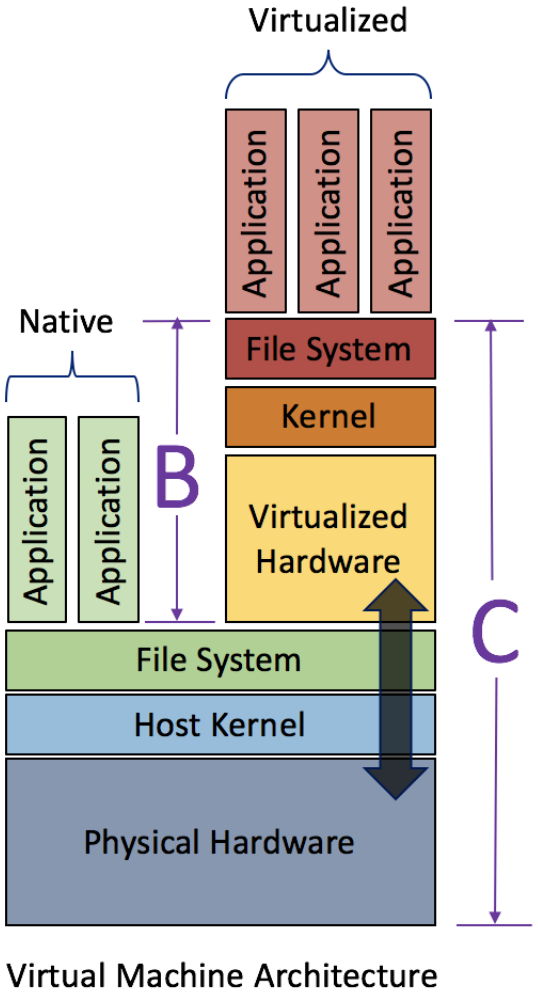
- FastQC
- MultiQC
- Fastp
- MEGAHIT
- SPAdes
- seqkit
- BBMap
- Strobealign
- CONCOCT
- MetaBAT2
- DASTool
- DRep
- CheckM
- CheckM2
- GTDB-Tk
- METABOLIC
- CoverM



HPC – Apptainer

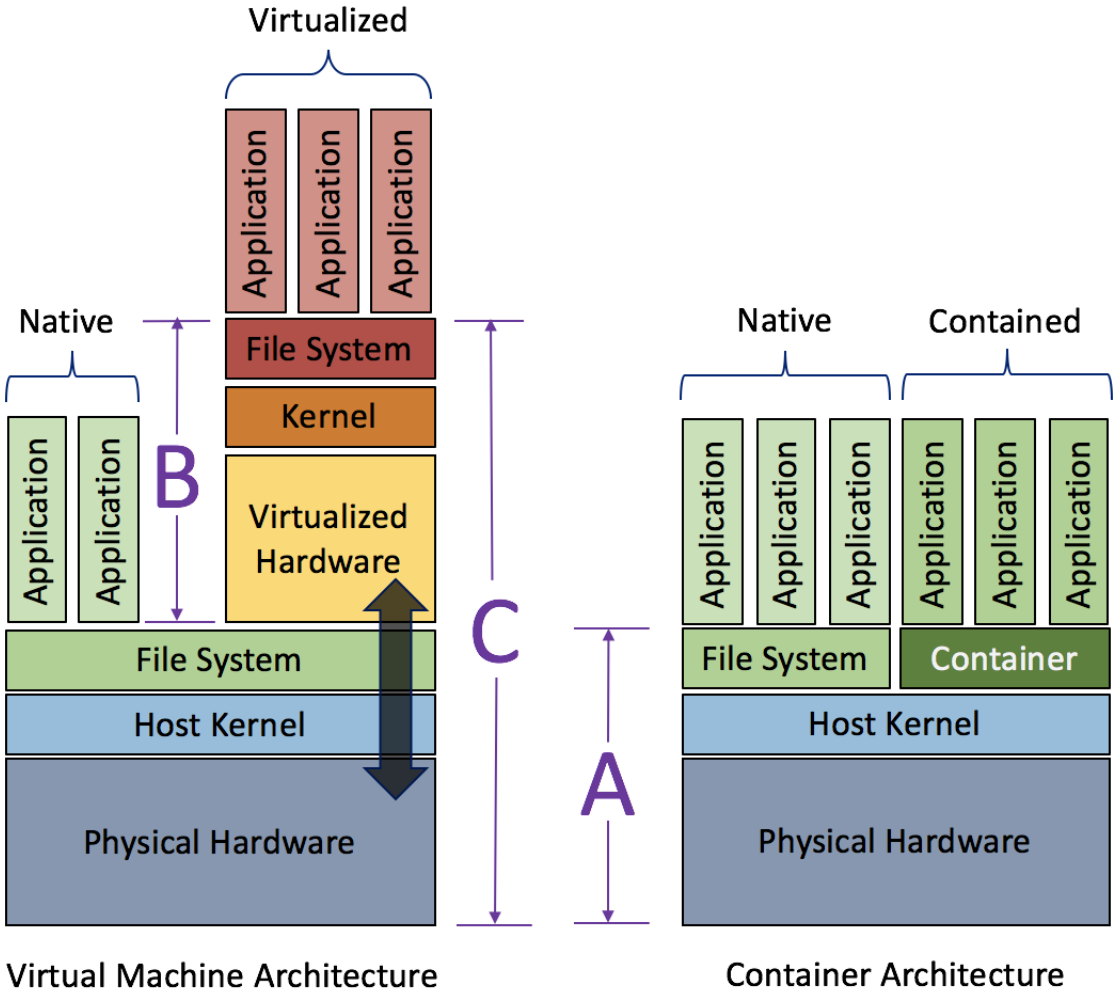


HPC – Apptainer



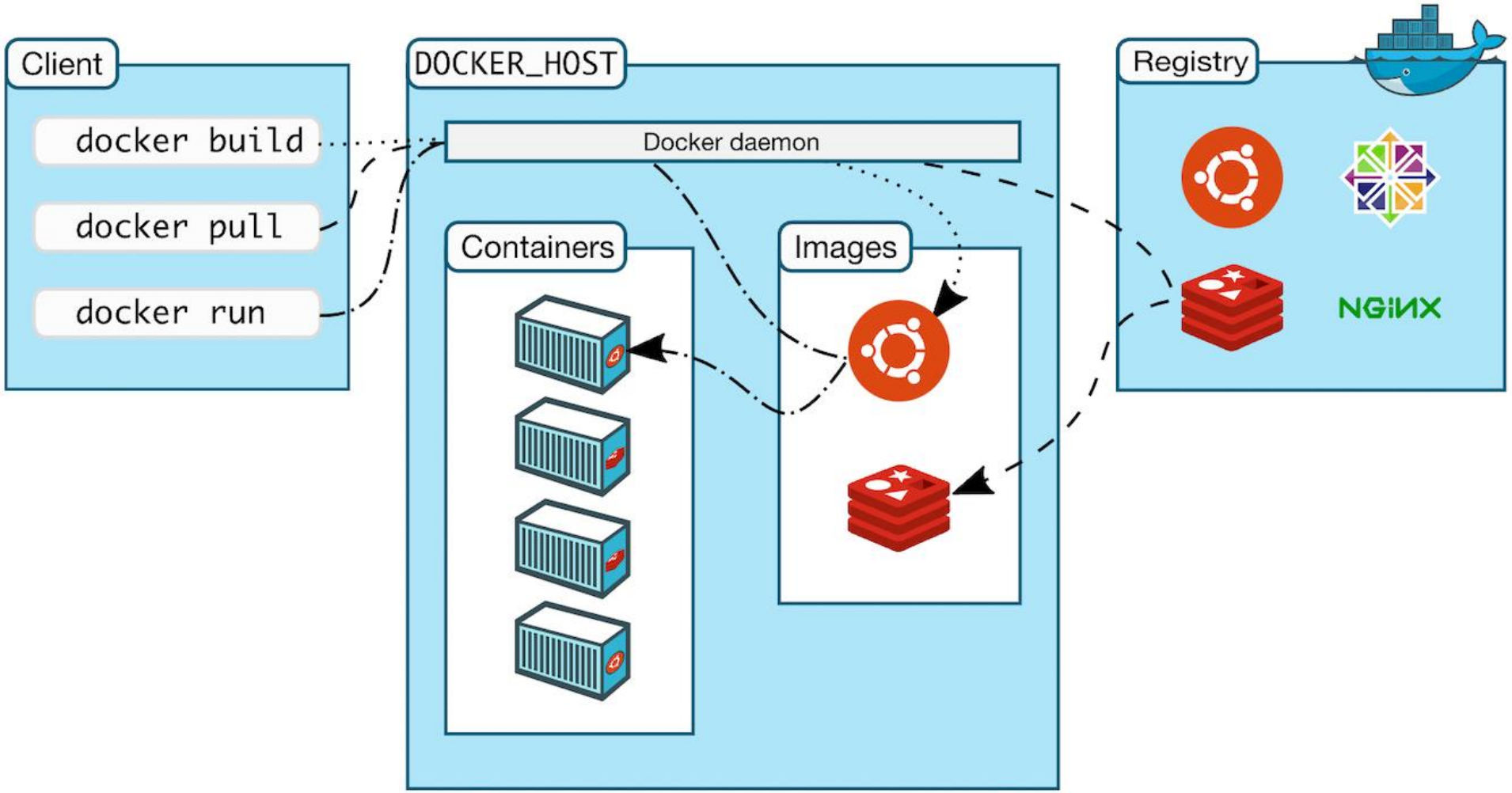
Virtual Machine Architecture

HPC – Apptainer

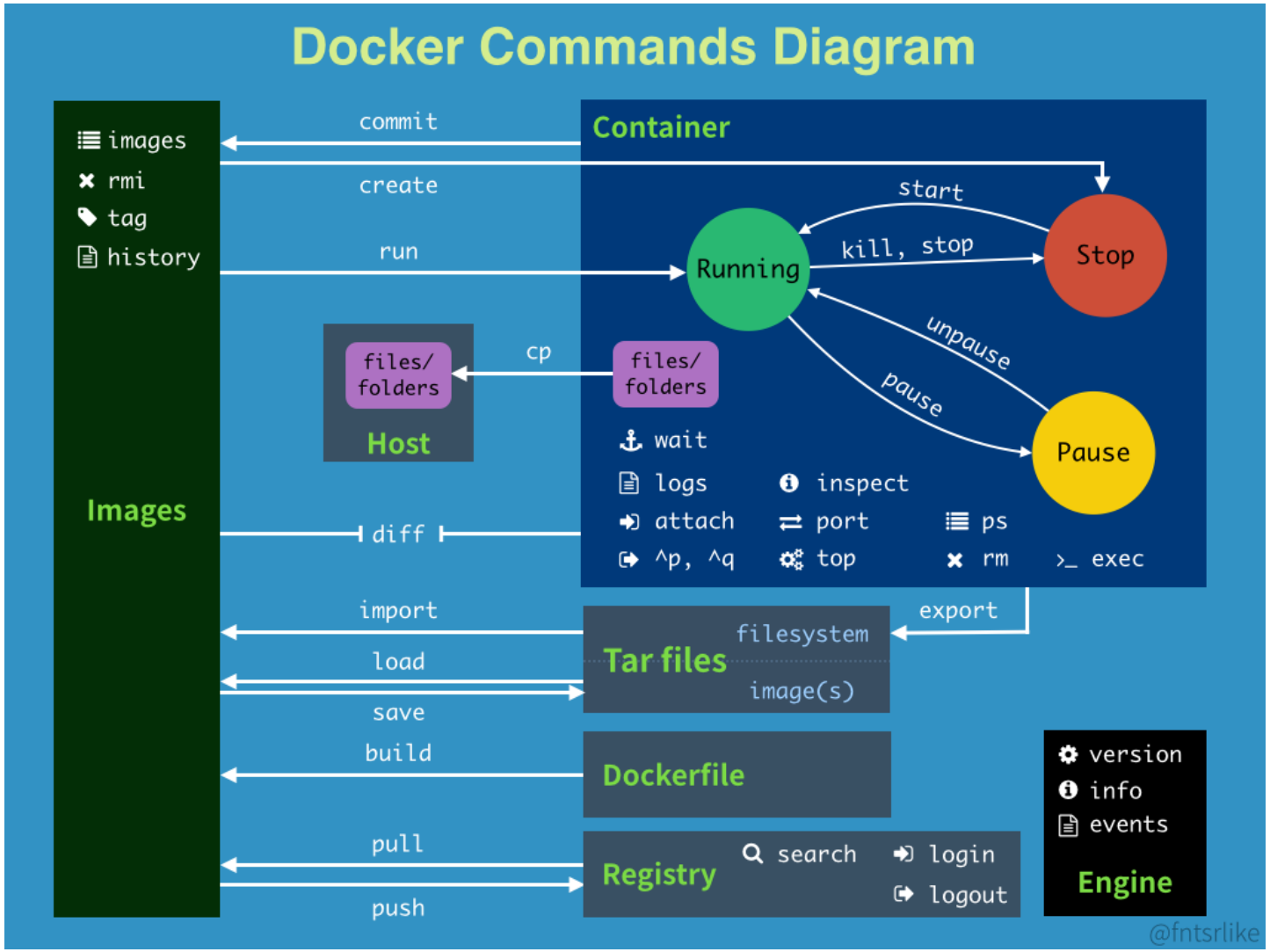


- Apptainer/Singularity is the most widely used container system for HPC.
- It is designed to execute applications at bare-metal performance.
- Secure
- Portable
- Reproductible

HPC – Docker container system



HPC – Docker/apptainer images

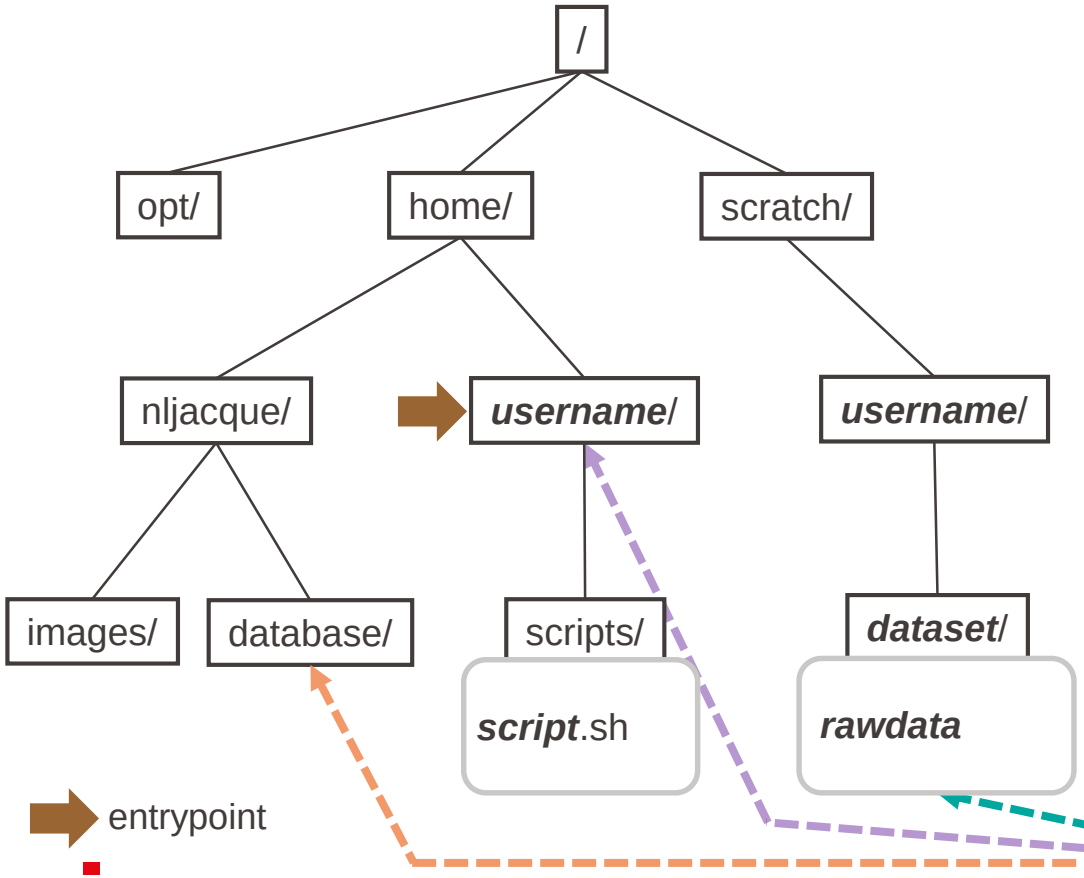


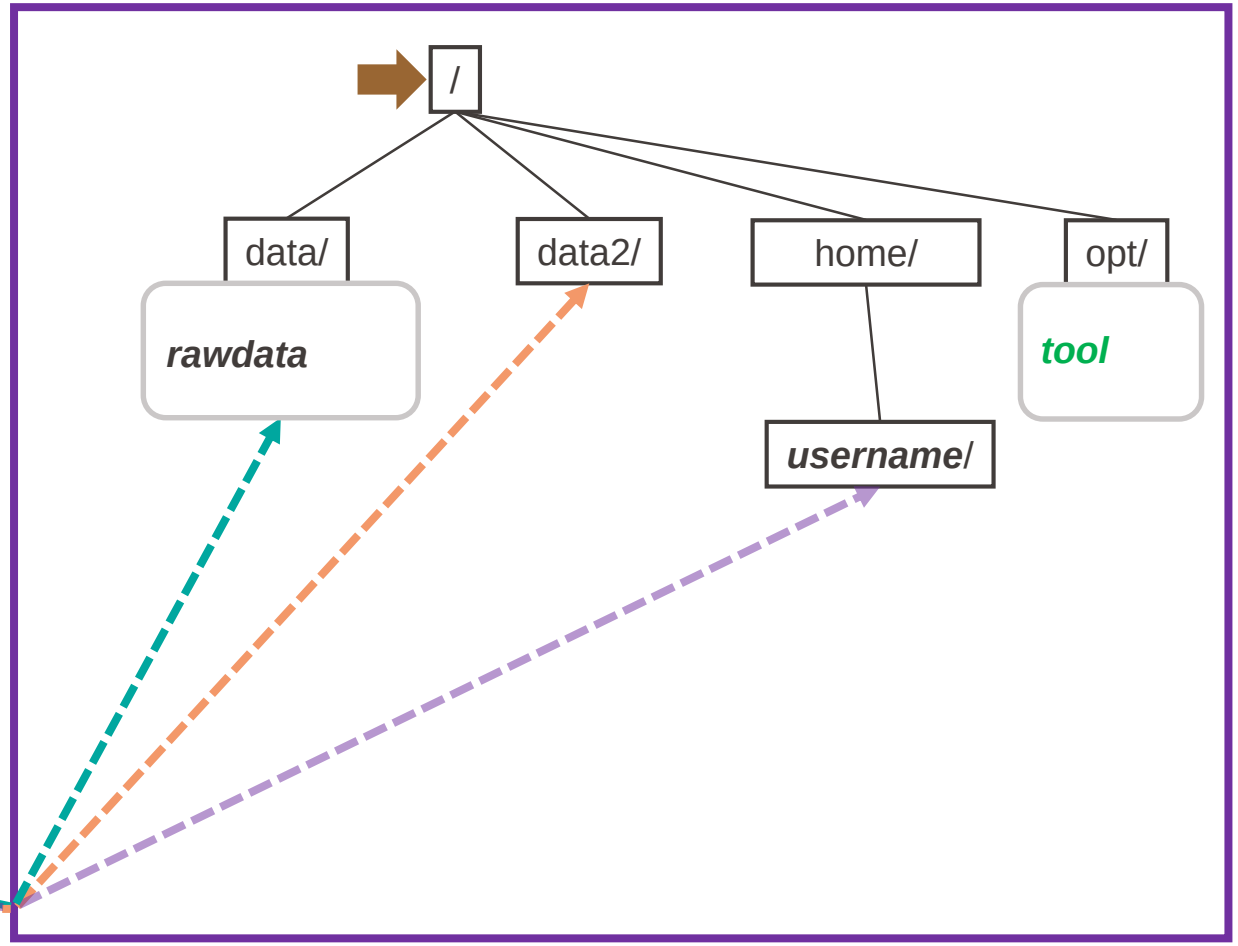
SCITAS – Filesystem & container

```
apptainer exec \
--bind /scratch/username/dataset/:/data \
--bind /home/nljacque/database:/data2 \
/home/nljacque/images/image.sif \
bash script.sh
```

script.sh

```
cd /data

tool --option rawdata
```





HPC – Submit a job which executes a script

cp /home/nljacque/scripts/sbatch_example.sh .

cp sbatch_example.sh sbatch_***tool***.sh

nano/vim sbatch_***tool***.sh → *Set up :*
- the sbatch directives
- the path to the image
- the path to the script

nano/vim ***tool***.sh → *Copy paste code from github*

sbatch sbatch_***tool***.sh → *Submit your job ☺*

sacct