
FARM
COURSE
Quick and cheap introduction to Sanger's HPC

Revision

2023-10-10	Singularity and modules.
2024-04-01	Updated references farm5 to farm22.
2024-09-01	Fix typos. Draft multi-node gpu.
2024-10-18	Add elastic instructions. Change “Revision” format.
2024-10-28	Add conda instructions.
2025-07-01	Changed cellgen/conda for ISG/conda

Table of contents

1a

Why use it?

Why and when to use the High Performance Compute Environment.

The need for more resources

Research that uses computing resources can easily outgrow the desktop or laptop computer where it started.

There might **not be enough memory, not enough disk space or it may take too long to run computations**.

Typical cases of when it may be beneficial to use a cluster:

1. computations need much more memory than what is available
2. computations can be speed up with more CPUs for parallel processing
3. the size of the data generated doesn't fit in conventional storages

1b

What is it?

What’s a compute cluster?

Defining a FARM

A **computer cluster** is a set of computers that work together so that they can be viewed as a single system. Each node can perform the same tasks, controlled and scheduled by software.

Clusters are just a grouping of computers with similar components

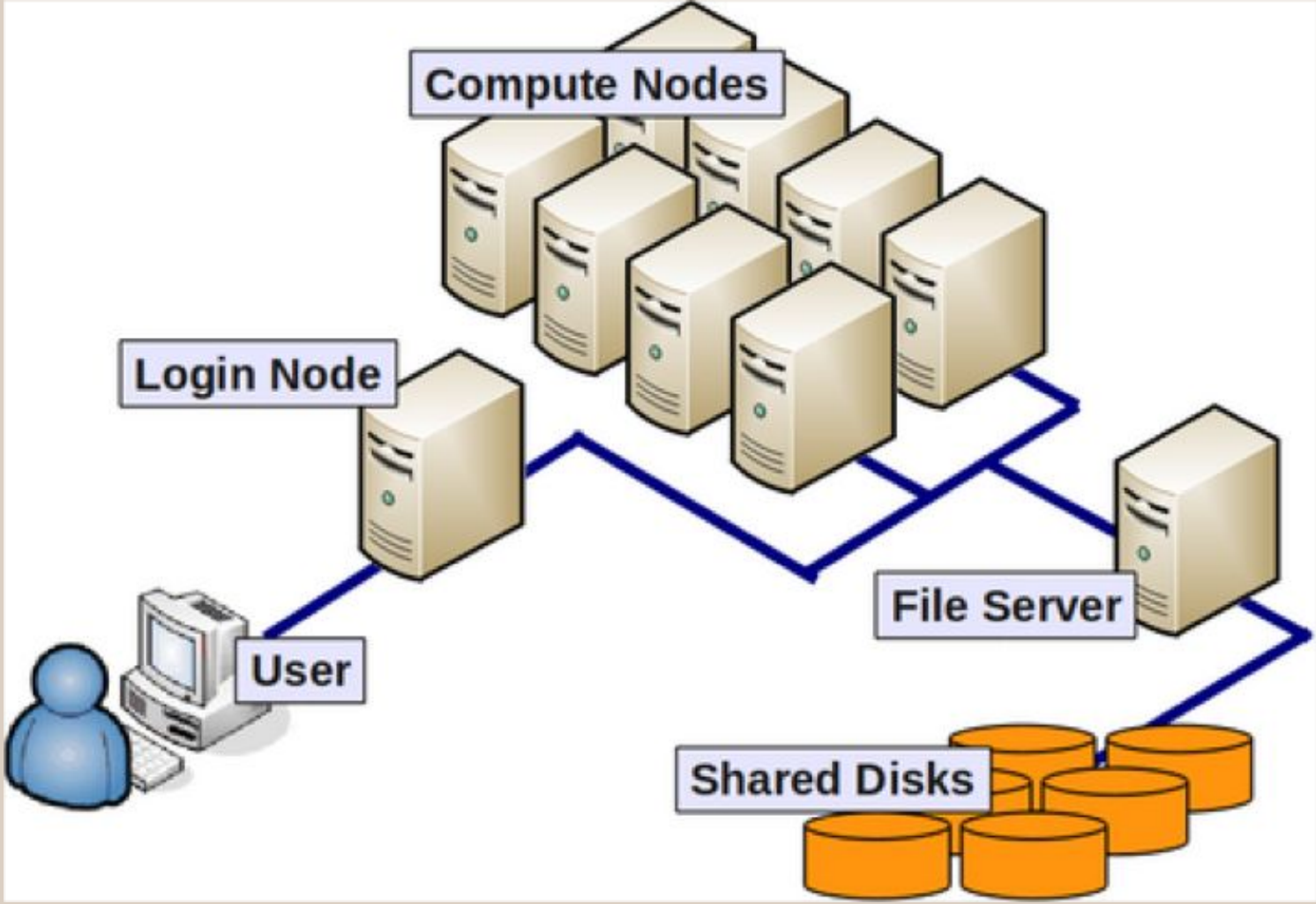


Image stolen from https://hbctraining.github.io/Intro-to-shell-flipped/lessons/08_HPC_intro_and_terms.html

Nodes and Cores

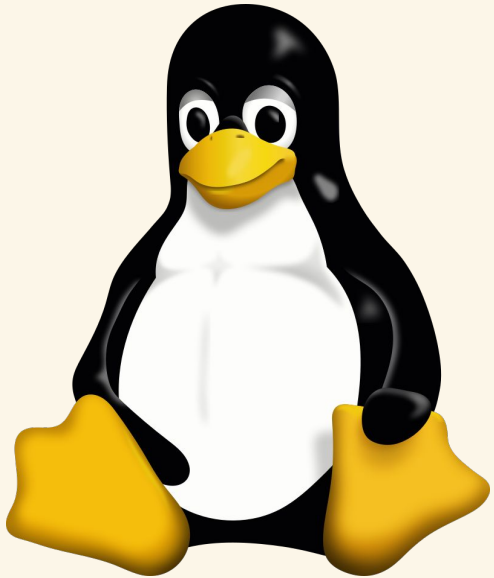
Each individual computer in a cluster a “**node**”. Inside each node processors do the actual computation (cores).

A typical node in our cluster will have from 65-256 cores and ~700 GB of RAM.

Sanger environment

The FARM compute nodes run on Ubuntu 22.04.

If you are new to Unix please consider looking into a Linux introduction course (*not covered in this talk*).



Basic flow

(1) user accesses the cluster via one or more **login nodes**, and
(2) **submits jobs** to the scheduler, which will dispatch to and collect the completed work from the compute nodes and
(3) results are stored in **shared file systems**

Access to the Compute Nodes is done via a Login or Head node

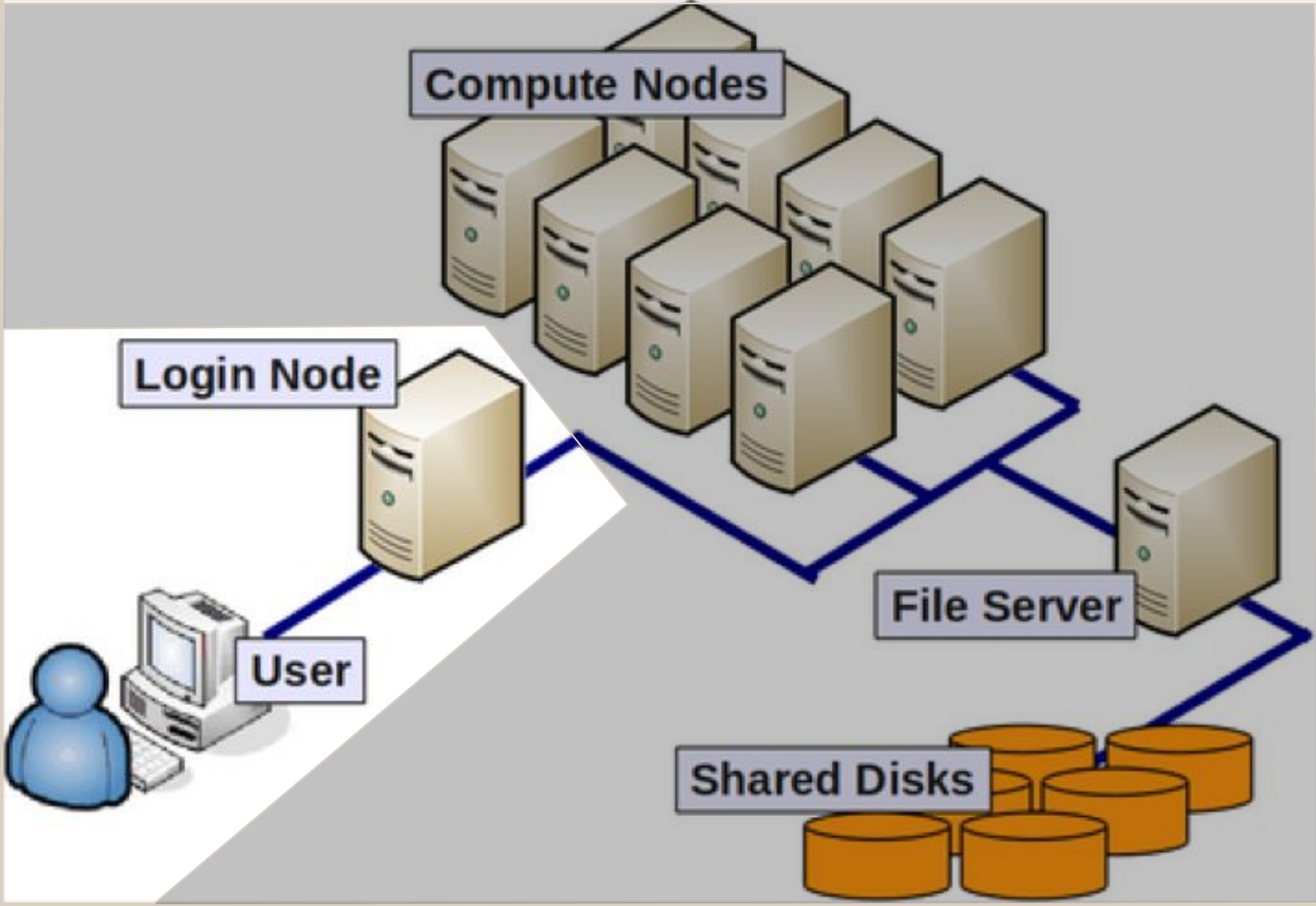
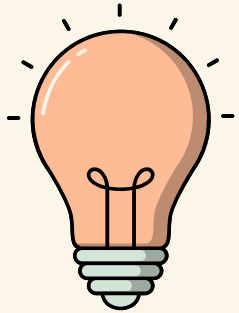


Image stolen from https://hbctraining.github.io/Intro-to-shell-flipped/lessons/08_HPC_intro_and_terms.html

Logging in to the Farm

To access the compute cluster you'll need a Sanger account and an SSH client. This is doable from:

- On-site (WTSI network)
- Remotely:
  - VPN
  - Teleport

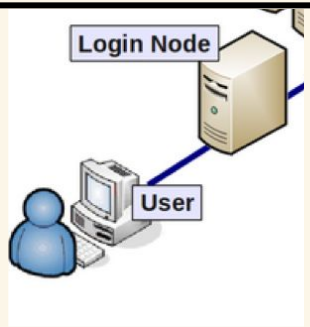


For this course

To logging to the farm you'll need one of these setups:

- OnSite/VPN + Mac or Unix
- Account on https://jhub.cellgeni.sanger.ac.uk
- OnSite/VPN + https://jupyter.internal.sanger.ac.uk

SSH Command




```
ssh user@gen22
```

Your sanger username. It's the first part of your email address. Example: **ob1**

Head node (machine) you're connecting to.
For training **gen22**
For work **farm22**

Head Nodes (do)

Head nodes are special nodes in the cluster you can connect to. Their main use is to:

- submit your LSF jobs
- copy data between file systems
- do simple housekeeping tasks
- start terminal sessions (screen/tmux)

Head Nodes (dont)

Head nodes have a specific propose and some things must not be run in them. **Do not**:

- Run cpu intensive tasks (unzip, run tools, etc)
- Leave code running
- Aggressively download files

2

LSF

IBM Spectrum LSF (formerly known as Load Sharing Facility)

Jobs

On a cluster, we need to run commands or scripts (executable tasks) as jobs.

Your scripts need resources (memory/cpu) to run. However, the resources required to run the **jobs** may not always be available straight away, so the jobs get submitted into a **queue**

Queues

A queue is a list of jobs which are waiting for resources (***pending***) or being executed (***running***). As jobs in the queue finish executing, the resources they were using become available again and the next job in the queue will start running.

Scheduling

Many factors control when and where the job starts to run: Active time window of the queues, Resource requirements of the job, Availability of eligible hosts, Job dependency conditions, Fair share constraints (configured user share policies), etc.

Fairshare

Fair share scheduling policy dispatches jobs between users based on assigned user shares, resource usage, or other factors. (ie: how many are running for the user? How many were submitted? etc.)

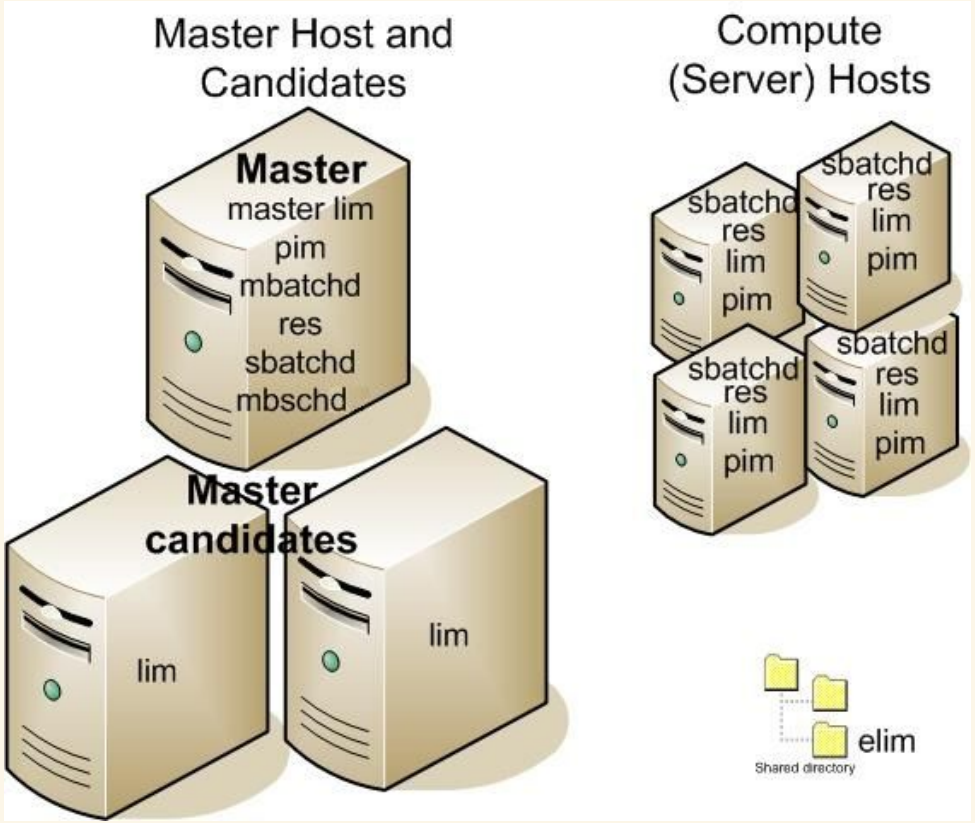
Job scheduling and execution is controlled by the platform load sharing facility (LSF) which manages the workload.

Workload management

LSF is a workload management platform, job scheduler, for distributed high performance computing owned by IBM.

The software coordinates the execution of multiple jobs on the FARM. Its primary function is to efficiently allocate available resources to incoming jobs based on various **scheduling policies**.

LSF Cluster Architecture



LSF processes running on each compute node

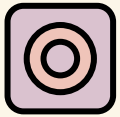
Cluster-level Architecture

lim load information manager

Collects load and config information

pim process information manager

Collects info of jobs running on the host



res remote execution server

accepts remote job execution requests

sbatchd

receives the request and manages local execution

LSF processes running on master nodes

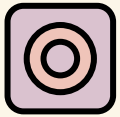
Cluster-level Architecture

lim/pim/res

Just like on compute nodes

mbatchd

Responsible for the overall state of jobs in the system



mbschd

makes scheduling decisions

sbatchd

receives the request and manages local execution

Cluster Daemons

Daemon	Role
mbatchd	Job requests and dispatch
mbschd	Job scheduling
sbatchd	Job execution
res	Job execution
lim	Host information
pim	Job process information
elim	Dynamic load indexes

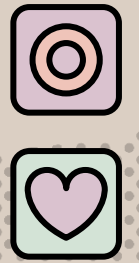
15 min break

Because you've been doing so well!

3a

Storage

Shared filesystems and where to put your data



What is a filesystem?

A file system is the way in which files are named and where they are placed logically for storage and retrieval.

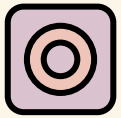
Physical disks are bundled together into a virtual volume; this virtual volume may represent one filesystem, or may be divided up, or partitioned, into multiple filesystems.

How to access

Filesystems are accessed over the network through mount points.

There are multiple storage/file systems options available for you to do your work and they will be mounted on the farm. Most notable are: /**nfs**, /**lustre**, and, /**warehouse**

What sets us apart?



LUSTRE

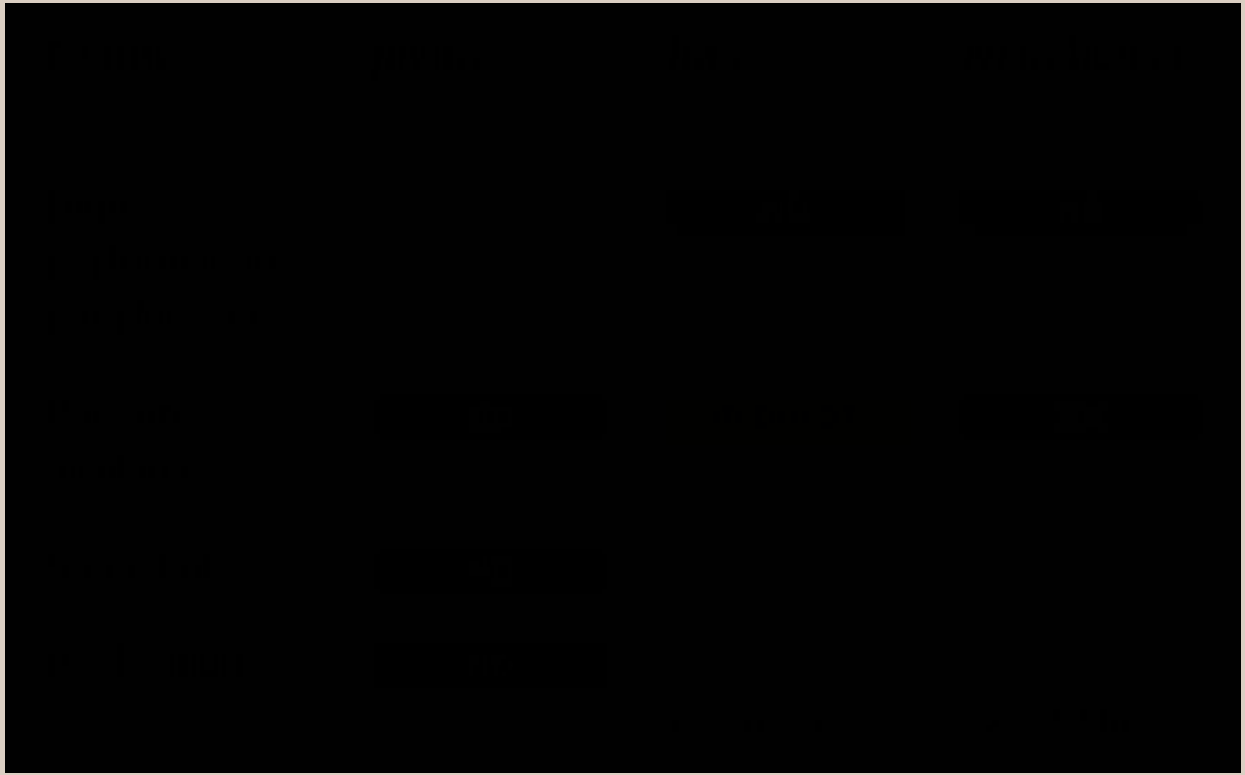
Super fast temporary working space, not backed up

NFS

Slower access, snapshotted, replicated. Used for long term storage

WAREHOUSE

Slow access snapshotted, replicated. Accessible from head nodes.



Storage system feature table

Filesystem summary

NFS (`/nfs/users/nfs_x/xx`): Home folders are small, **up to 20 GB** for usr exclusive access. **Don't store large files or conda envs.**

NFS (`/nfs/team`): Team/Project folders have larger quota and are for slow longer term storage. You can read/write from compute nodes. Multiple writes will break NFS **be gentle**.

Filesystem summary

WAREHOUSE (team/project): longer storage, slow, can’t be accessed from head nodes so it’s a safer place to put your data into after finishing analysis.

Filesystem summary

NFS: (team/project) **DON’T RUN HIGH THROUGHPUT JOBS AGAINST NFS!** Many concurrent read or writes can affect NFS, keep reading and writing to NFS from jobs to a minimum.

Do not store your conda environments here.

Filesystem summary

LUSTRE: (team/project) scratch space, temporary working space, **not backed up**.

DON'T STORE THINGS HERE LONG TERM! Output of currently active pipelines, delete after finish using. Supports concurrent read/writes better than NFS.

Do not store your conda environments here.

Check your team's quota

LUSTRE:

```
lfs quota -h -g <team> /lustre/scratch126
```

NFS:

```
df -h /nfs/<team>
```

Check your UNIX groups

```
id <user>
```

Switching between groups

Change group (starts a new bash session):

```
newgrp <team>
```

Copying data between groups

rsync:

```
rsync -avzhP --chown user:<team> /from/src /to/dest
```

Rsync stands for “remote synchronization”. It is used to efficiently transfer files. When you belong to more than one group, if you want to copy files from one place to another, you should be cautious of using the right group (team).

Use the --chown option specifying your username and the group you want the file to belong in the destination.

3

b

iRODS

Integrated Rule-Oriented Data System (iRODS)
Genomic data archiving

iRODS Integrated Rule Oriented Data Systems

iRODS is the software Sanger uses for managing genomic data generated on site.

The data grid system acts as the data management system for the institute. Users and can then query the metadata to find and track data.

Objects (d) and Collections (C) correspond to files and directories. There are multiple client types but the most used are the icommands on the command line.

iRODS data

The data on irods is not mounted like a usual filesystem. Meaning you can't see it like you do /lustre or /nfs. To navigate the collections and data objects you need to use specific tools named **icommands**.

The commands you will use most often are:

- **`ils`**: list information about a file
- **`iget`**: retrieve a file
- **`iput`**: upload a file
- **`imeta`** - search for file or directory with matching metadata

```
module load ISG/IRODS
```

iRODS list and download file

```
$ils /Sanger1
/Sanger1:
  C- /Sanger1/home
  C- /Sanger1/training
  C- /Sanger1/trash
```

List contents of /Sanger1 collection

```
$ils /Sanger1/training
/Sanger1/training:
  irodscourse_.JqmFYITGTG.txt
  irodscourse_.JRY66Yo0wN.txt
```

List contents of /Sanger1/training collection

```
$ iget -Kv /Sanger1/training/irodscourse_.JqmFYITGTG.txt .
  irodscourse_.JqmFYITGTG.t      0.000 MB | 0.014 sec | 0 thr |  0.014 MB/s
```

Download specific file

iRODS metadata

User-defined metadata can be applied to objects and collections. Metadata are attribute-value-units triples (AVUs) that can be queried to find matching objects.

The **imeta** command allows you to search for specific metadata keys and find your genomic data using, for example, study or sample names or identifiers.

Read more about iRODS here:
https://gitlab.internal.sanger.ac.uk/training/irods-training-course/-/blob/main/iRODSCourse.ipynb

iRODS metadata search

imeta qu -d -z /seq sample = ‘SAM12’

Search for a file (-d) with metadata key ‘sample’ and value ‘SAM12’ in the /seq zone

imeta qu -C -z /seq study = ‘STD99’

Search for a folder (-C) with metadata key ‘study’ and value ‘STD99’ in the /seq zone

imeta qu -d -z /archive project = ‘P678’

Search for a file (-d) with metadata key ‘project’ and value ‘P678’ in the /archive zone

15 min break

Last one

4

LSF commands

View cluster information, submit jobs and track your job’s status

Before we continue

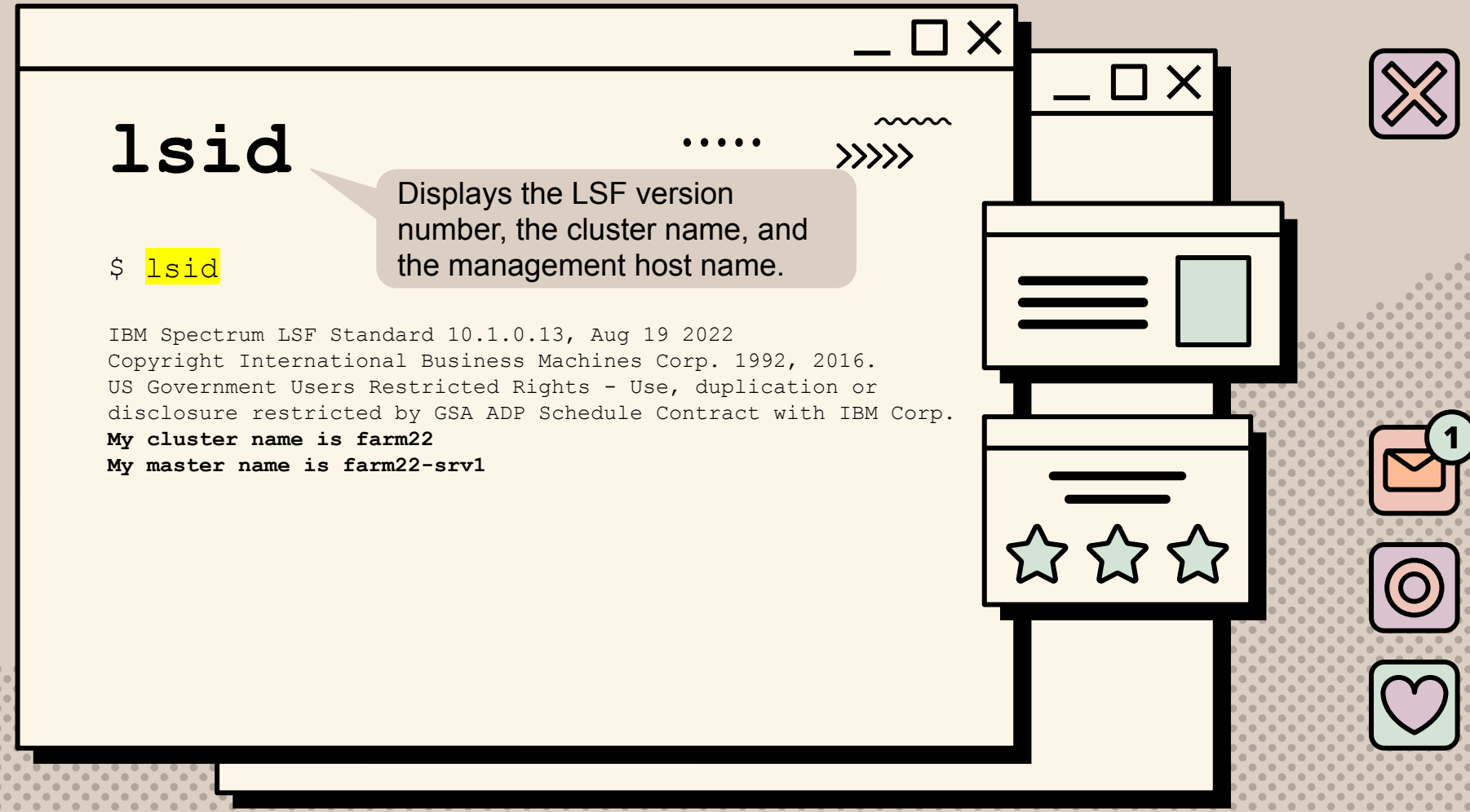
Let’s check:

1. Environment access? (onsite/vpn)
2. Access to terminal?
3. Access to SSH client?

lsid

Displays the LSF version number, the cluster name, and the management host name.

```
$ lsid

IBM Spectrum LSF Standard 10.1.0.13, Aug 19 2022
Copyright International Business Machines Corp. 1992, 2016.
US Government Users Restricted Rights - Use, duplication or
disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
My cluster name is farm22
My master name is farm22-srv1
```



bhosts

Displays hosts and their static and dynamic resources.

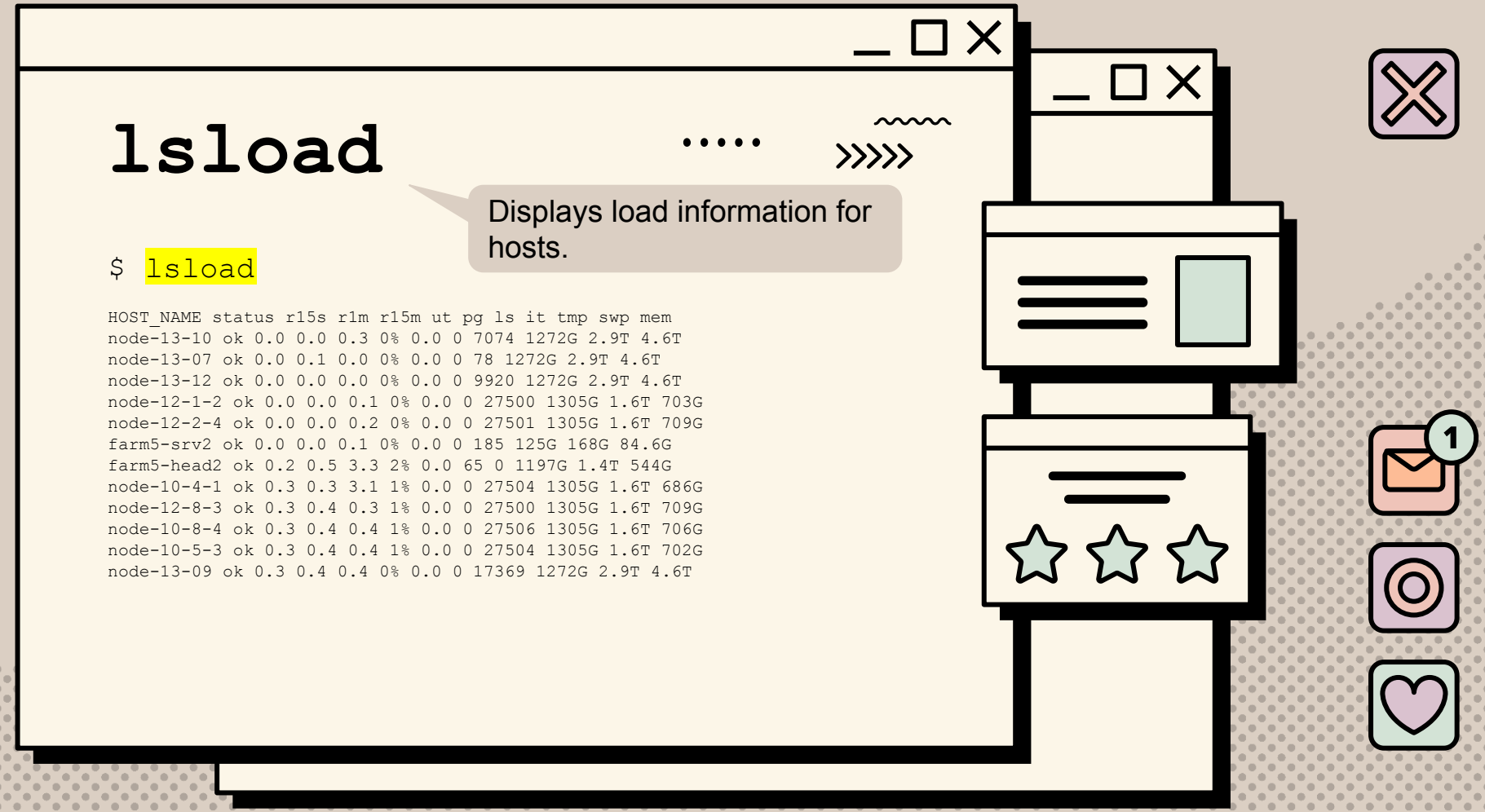
```
$ bhosts

HOST_NAME          STATUS       JL/U    MAX  NJOBS    RUN  SSUSP  USUSP
RSV
farm22-head1       ok              -      1      0      0      0      0
0
farm22-head2       ok              -      1      0      0      0      0
0
node-10-1-1        ok              -     64     17     17      0      0
0
node-10-1-2        ok              -     64     60     60      0      0
0
node-10-1-3        ok              -     64     14     14      0      0
0
node-10-1-4        ok              -     64     25     25      0      0
0
node-10-2-1        ok              -     64     16     16      0      0
0
node-10-2-2        ok              -     64     25     25      0      0
0
node-10-2-3        ok              -     64     49     49      0      0
0
0
node-10-3-1
0
```

lsload

Displays load information for hosts.

```
$ lsload
```

```
HOST_NAME status r15s r1m r15m ut pg ls it tmp swp mem
node-13-10 ok 0.0 0.0 0.3 0% 0.0 0 7074 1272G 2.9T 4.6T
node-13-07 ok 0.0 0.1 0.0 0% 0.0 0 78 1272G 2.9T 4.6T
node-13-12 ok 0.0 0.0 0.0 0% 0.0 0 9920 1272G 2.9T 4.6T
node-12-1-2 ok 0.0 0.0 0.1 0% 0.0 0 27500 1305G 1.6T 703G
node-12-2-4 ok 0.0 0.0 0.2 0% 0.0 0 27501 1305G 1.6T 709G
farm5-srv2 ok 0.0 0.0 0.1 0% 0.0 0 185 125G 168G 84.6G
farm5-head2 ok 0.2 0.5 3.3 2% 0.0 65 0 1197G 1.4T 544G
node-10-4-1 ok 0.3 0.3 3.1 1% 0.0 0 27504 1305G 1.6T 686G
node-12-8-3 ok 0.3 0.4 0.3 1% 0.0 0 27500 1305G 1.6T 709G
node-10-8-4 ok 0.3 0.4 0.4 1% 0.0 0 27506 1305G 1.6T 706G
node-10-5-3 ok 0.3 0.4 0.4 1% 0.0 0 27504 1305G 1.6T 702G
node-13-09 ok 0.3 0.4 0.4 0% 0.0 0 17369 1272G 2.9T 4.6T
```



bugroup

Displays information about user groups.

```
$ bugroup
```

```
GROUP_NAME    USERS
abscr         or3 os9 yr1 dh24 pa15 af24
analysis-cgp ad26 av8 yx2 sr17 pr10 drj go1 dw9 rpe kd7 dg17 ra11 im2 tb14
mj3 kb3 dl8 dmb jt14 snz jy3 sf5 jw32 ti1 pc8 mp17 kr2 em4 las rr11 msd
cgppipe my4 cp16 am3 la2 cl6 nr3
team113-grp  mdc1 lr15 pyc1 am32 aw28 kw10 pb19 t113admin lvdw mv7 oc6 cw6
da15 so16 ad33 sc46 ds39 iv3 jb62 vo1 im13 pg20 ag35 jl17
team154-grp  zs3 hm8 sl31 em16 yw2 ma18 ji2 ac69 kd7 mp34 hj6 an11 tb14 yl3
sg18 ai5 dl8 cd23 sl17 mr18 rt11 nv3 pc8 mn12 da15 hx2 nb15 ky3 ap31 tjm
team163-grp  mg31 em5 sk24 mf13 cl15 pq1 mg16 mm23 sp15 bd7 dg23 vm11
team176-grp  se3
team215-grp  km29 mt26 gg20 sb43 hl7 cd7 jgrg st25 mc32 ch24
team219-grp  ck14 ub1 mh28 tq2 rs26 so7
team228-grp  jd14 gk11
team238-grp  data-cosmic hb7 sy2 tm6 bh4 data-cosmic-dev hp8 ds33 nb7 cyk
sw23
```

bqueues

List all available queues.

```
$ bqueues
```

```
QUEUE_NAME      PRIO STATUS          MAX JL/U JL/P JL/H NJOBS  PEND   RUN
SUSP
gpu-basement    500  Open:Active       -    -    -    -    36     0    36
0
yesterday       500  Open:Active      50    7    -    -    14     0    14
0
basement        300  Open:Active    2000 2000    -    -  1772    59  1713
0
week            100  Open:Active    3000    -    -    -    67     0    67
0
gpu-huge        100  Open:Active       -    -    -    -     4     1     3
0
gpu-normal      100  Open:Active       -    -    -    -   644   614    30
0
hugemem         100  Open:Active       -    -    -    -   219   218     1
0
long            100  Open:Active    5000    -    -    -  1788    50  1734
4
normal          100  Open:Active       -    -    -    -  1663   659  1003
1
parallel        100  Open:Active                                  20     0    20
0
small
0
```

bqueues

QUEUE_NAME - the name of the queue
PRIO - the priority of the queue
STATUS - the status of the queue
MAX - the maximum number of job slots available
JL/U, JL/P and JL/H - the job slot limit for users, processors and hosts respectively
NJOBS - the total number of tasks for all jobs in the queue
PEND - the number of pending jobs in the queue
RUN - the number of running jobs in the queue
SUSP - the number of suspended jobs in the queue

bqueues

```
$ bqueues -l normal
```

```
QUEUE: normal
  -- For jobs running for around an hour or two (up to 12 hours)  This is the
default queue.

…

RUNLIMIT
 720.0 min
MEMLIMIT
 683.5 G

…

HOSTS:  dynamic/ standard_hosts/
```

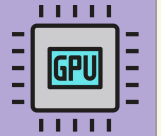
queue limits

MEMLIMIT: maximum memory available to request
RUNLIMIT: maximum time a job can run before it's terminated

```
bqueues -o 'queue_name runlimit memlimit cpulimit'
```

queue limits

QUEUE_NAME	MAX_RUNLIMIT	MAX_MEMLIMIT
`normal`	12 hours (720 min)	683.5G
`long`	48 hours (2880 min)	683.5G
`week`	7 days (10080 min)	683.5G
`basement`	30 days (43200 min)	683.5G
`hugemem`	15 days (21600 min)	727.5G
`teramem`	15 days (21600 min)	2.9T
`transfer`	30 days (43200 min)	683.5G
`gpu-normal`	12 hours (720 min)	683.5G
`gpu-huge`	48 hours (2880 min)	683.5G
`gpu-basement`	15 days (21600 min)	683.5G



bjobs

```
$ bjobs -u all
```

JOBID	USER	STAT	QUEUE	FROM_HOST	EXEC_HOST	JOB_NAME	SUBMIT_TIME
*996738	mercury	RUN	normal	node-14-17	node-13-14	*13076377)	Jun 26 13:29
*996897	mercury	RUN	normal	node-14-17	node-13-14	*13076376)	Jun 26 13:33
*996898	mercury	RUN	normal	node-14-17	node-13-14	*_GT_UKBB)	Jun 26 13:33
*991425	tr11	RUN	long	farm5-head2	node-13-16	*file[524]	Jun 26 10:31
*991425	tr11	RUN	long	farm5-head2	node-13-16	*file[525]	Jun 26 10:31
*995389	cc47	RUN	normal	node-14-26	node-13-16	*592fc0d4c	Jun 26 12:23
*995390	cc47	RUN	normal	node-14-26	node-13-16	*b2d6e64e2	Jun 26 12:24
*995588	kr23	RUN	long	farm5-head2	node-13-16	*n/bash -l	Jun 26 12:31

bjobs

JOBID - unique numerical job identifier used to keep track of the job
USER - username of the person who submitted the job
STAT - job state
QUEUE - which queue the job was submitted to
FROM_HOST - which host the job was submitted from
EXEC_HOST - which host the job is running on (blank if job is pending)
JOB_NAME - name of the job
SUBMIT_TIME - when the job was submitted

STAT - job state

PEND - the job is waiting in the queue to be scheduled and dispatched
RUN - the job has been dispatched to a host *(node)* and is running
DONE - the jobs finished normally (exit status is 0)
PSUSP - job was suspended by the owner or admin while pending
USUSP - job was suspended by the owner or admin after dispatched
SSUSP - job was suspended by LSF after dispatched

Submitting an interactive job

```
bsub
    -G farm-course
    -q normal
    -n1
    -M4000
    -R"select[mem>4000] rusage[mem=4000]"
    -Is
    /bin/bash
```

Submitting a job

bsub

Submits a job to LSF by running the specified command and its arguments and assigns a unique job ID.
LSF will run the job on a host that satisfies all requirements. If LSF cannot run the job immediately, LSF scheduling policies determine the order of dispatch. By default the current working directory is used for the job.

Submitting a job

`-G farm-course`

For fair share scheduling. **Associates the job with the specified group** or excludes the job from the specified groups. LSF uses **LSB_DEFAULT_USERGROUP** environment variable to allow you to set a default group. Instead of passing the -G every time use:

```
export LSB_DEFAULT_USERGROUP=YourGroup
```

In this case, the group is “farm-course”. Usually each team gets its own group.

Submitting a job

`-q normal`

Submits the job to one of the **specified queues.**

The specified queues must be defined for the local cluster. For a list of available queues in your local cluster, use bqueues.

When a list of queue names is specified, LSF attempts to submit the job to the first queue listed.

Submitting a job

`-n1`

Submits a parallel job and specifies the number of tasks in the job. **The number of tasks is used to allocate a number of slots/processors for the job.**

In this case, we're requesting 1 slot (CPU/core) to use for our job.

Submitting a job

```
-M4000
-R"select[mem>4000] rusage[mem=4000]"
```

-M Sets a memory limit in megabytes [MB] for all the processes that belong to the job

-R Simple resource requirement strings:

- **select**: select a host that has the resources specified. Select a host with [mem>4000] greater than 4GB of memory available.
- **rusage**: reserve the resource for the task. Reserve in the host [mem=4000] 4GB of memory for this job.

Submitting a job

`-Is`

Submits an **interactive job** and creates a pseudo-terminal with shell mode when the job starts.

Jobs can be interactive or headless. Some applications require a pseudo-terminal in order to run correctly. **Interactive jobs allow you to manually input the commands you want to run on the job** instead of using a predefined script (headless).

Submitting a job

```
/bin/bash
```

Command to run. It can be a tool, a script or any other executable/interpretable command and arguments.

In this case, because it’s an interactive job, we want to open a bash shell (/bin/bash) to interact with.

Submitting an non-interactive job

```
bsub
    -G farm-course
    -q normal
    -n1
    -M4000
    -R"select[mem>4000] rusage[mem=4000]"
    -o output%J.log
    -e error%J.log
    myscript.sh
```

Submitting a job

```
-o output%J.log
-e error%J.log
```

-o Appends the standard output of the job to the specified file path (output%J.log). **Mandatory in non-interactive mode.**
-e Appends the standard error output of the job to the specified file path (error%J.log).

%J *in the name of the output file is replaced by the job ID of the job*

Wrapper Scripts

submit.sh

```
#!/usr/bin/env bash

CORES=1
RAM="4G"
QUEUE="normal"
GROUP="farm-course"
SCRIPT="/path/to/myscript.sh"

bsub \
   -G "${GROUP}" -q "${QUEUE}"  -n ${CORES} \
   -M ${RAM} -R "select[mem>${RAM}] rusage[mem=${RAM}]" \
   -o "%J.output" -e "%J.error" \
   "${SCRIPT}"
```

Alternative #BSUB notation

```
--- myjob.lsf ---
  #BSUB -G farm-course
  #BSUB -q "normal"
  #BSUB -n 1
  #BSUB -M 4000
  #BSUB -R "select[mem>4000] rusage[mem=4000]"
  #BSUB -o "output%J.log"
  #BSUB -e "error%J.log"

  echo "Hi!"
```

```
--- submit command ---
  bsub < myjob.lsf
```

Submitting a GPU job

```
bsub
    -G farm-course
    -q gpu-normal
    -n1
    -M4000
    -R"select[mem>4000]
    rusage[mem=4000]"
    -gpu "mode=shared:num=1:gmem=6000"
    -o output%J.log
    -e error%J.log
    myscript.sh
```

Submitting a GPU job

```
-gpu "num=1:gmem=6000:mode=shared"
```

-num The number of physical GPUs required by the job.

-gmem Specify the GPU memory required by the job. The format is in MB. If not specified it, nothing will be reserved for this job and will only use what is available.

-mode The GPU mode when the job is running, either ‘shared’ or ‘exclusive_process’.

Submitting a GPU job

- **mode** When possible, **prioritize using "mode=shared"** to allow more use out of the GPU. In particular if you're only **doing dev work or inference**.

- **gpack When using shared mode use "gpack=yes"**. Tells LSF to pack multiple shared-mode GPU jobs to allocated GPUs with other shared jobs.

```
-gpu "num=1:gmem=6000:mode=shared:gpack=yes"
```

Submit a multi GPU job

```
-gpu "num=8:mode=exclusive_process"
```

LSF distributes all the allocated GPUs of a job as blocks when the number of cores is bigger than the requested number of GPUs. Strict CPU-GPU affinity binding is enabled by default.

If your job requires more than 1 GPU (using a single node)you may run into affinity issues. There is a proportion between CPU-GPU for a job to start. You need to book roughly 25CPU for each 1GPU.

Submit a multi-node GPU job

```
#BSUB -q gpu-normal          # name of the queue
#BSUB -gpu 'mode=exclusive_process:num=8' # gpus per host
#BSUB -o %J.out              # output file
#BSUB -e %J.err              # error file
#BSUB -M 8G                  # 8G RAM memory per host
#BSUB -R "select[mem>8G] rusage[mem=8G]" # same as above
#BSUB -n 4                   # number of cores in total
#BSUB -R "span[ptile=2]" # split total cores per host 2 on each
```

ptile is what controls the distribution of total cores ('tasks' in LSF) across hosts/nodes. Example 4 cores in total, split them 2 cores per host. Each host will get the same Memory and GPU resources stated in the bsub options (in this case 8GB RAM and 2 GPUs)

Submit a multi-node GPU job

```
#BSUB -q gpu-normal
#BSUB -gpu 'mode=exclusive_process:num=8'
#BSUB -R "span[ptile=2]"
#...

module load cuda-12.1.1
module load ISG/openmpi

export NCCL_DEBUG=INFO
export NCCL_IB_DISABLE=1
export UCX_IB_MLX5_DEVX=n

mpirun --display-allocation python training.py
```

Submission requires to use mpirun to launch the executable. This is to make sure the world and local rankings are properly set and managed.

Job Arrays

Job Arrays submit a large number of identical tasks as a single job with a single Job ID.

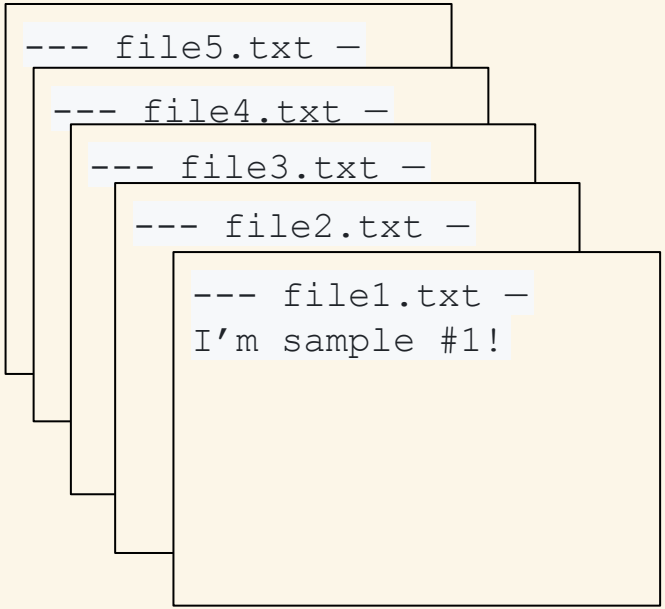
Instead of going through a loop and submitting multiple individual jobs, we can submit one job controlled by one Job ID.

```
bsub \
     -G farm-course \
     -J"[1-5]" \
     -o array%I_%J.log \
     script.sh
```

```
for i in 1..5;
do
  bsub \
     -o array%I_%J.log \
     script.sh $i
done
```

Job Arrays

```
bsub -J"[1-5]" -o array%I_%J.log /path/to/script.sh
```

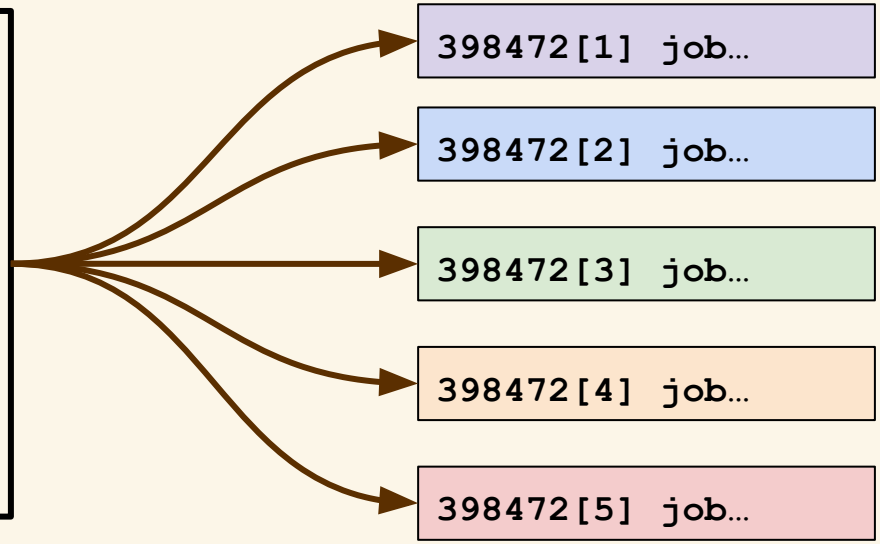


--- script.sh ---

```
echo "I’m job $LSB_JOBID"
echo "My index is $LSB_JOBINDEX"
echo "The contents of are:"

cat "file${LSB_JOBINDEX}.txt"
```

Job Arrays

```
bsub \
  -J"[1-5]" \
  -o array%I_%J.log \
  -q "normal"
  -n 1 \
  -M 4000 \
  -R "select[mem>4000]" \
  -R "rusage[mem=4000]" \
  /path/to/script.sh
```



JOB ID = **398472**

Job Arrays good practices

Using the job arrays it is possible to fill up the queues with a lot of jobs. Remember that there are other users on the system. **Specify that you want at most a certain number of jobs to simultaneously run on the system**:

```
bsub -J "[1-20]%5"
```

In this way you request a job array of 20 jobs, but with at most 5 running simultaneously.

⚠ **Don’t use the same output file for you jobs**. This causes file lock contention and can cause the file system to stop responding.

⚠ **Don’t rely on the order of execution**. The jobs in a job array are all completely independent, and they will not necessarily run in the order you expect.

Modules

Modules help with **dynamic modification of the user's environment** via modulefiles. Each modulefile contains the information needed to configure the shell for an application.

Modules on the farm are installed under: `/software/modules/`

More information: https://ssg-confluence.internal.sanger.ac.uk/display/FARM/Software+modules

Modules

Setting up your environment to use installed modules:

```
export MODULEPATH=$MODULEPATH:/software/modules/
module avail
module load cellgen/samtools
```

Export statement can be added to your **~/.bashrc** file to have modules available without specifying the path

Bsub + Modules example

```
#BSUB -G farm-course
#BSUB -q "normal"
#BSUB -n 1
#BSUB -M 4000
#BSUB -R "select[mem>4000] rusage[mem=4000]"
#BSUB -o "output%J.log"
#BSUB -e "error%J.log"

module load cellgen/samtools

samtools view -S -b sample.sam > sample.bam
samtools sort sample.bam -o sample.sorted.bam
samtools index sample.sorted.bam
```

Containers

A Unix operating system is broken into two primary components, the kernel space, and the user space. The Kernel talks to the hardware. The user space is the environment that most people are most familiar with.

Containers change the user space into a *swappable component*. This means that the entire user space portion of a Linux operating system, including programs, custom configurations, and environment can be independent of whether your system is running. This way users don't have to worry about dependencies and requirements.

Containers (singularity)

Singularity is a container platform. It allows you to create and **run containers that package up pieces of software** in a way that is portable and reproducible.

Singularity containers are a single file, and you don't have to worry about how to install all the software you need on each different operating system and system. Singularity was created to run complex applications on HPC clusters without requiring additional permissions or privileges.

Singularity shell

The **singularity shell** command can be used to start a container and run an interactive shell within it.

```
$ python3 -V
Python 3.10.12

$ module load cellgen/singularity
$ singularity shell /nfs/cellgeni/singularity/images/python-3.7.sif
Singularity> python3 -V
Python 3.7.2
```

Singularity path binding

You will need to **bind additional host system directories into a container** when using those not bound by default (current working directory and home folder).

There may be a shared dataset in lustrer that you need access to in the container. For that, you'll need to use the **--bind** or **-B** option with your singularity command.

Singularity path binding

```
$ module load cellgen/singularity
$ singularity shell /path/to/image.sif
Singularity> ls /lustre
ls: cannot access '/lustre': No such file or directory
```

```
$ module load cellgen/singularity
$ singularity shell --bind /lustre /path/to/image.sif
Singularity> ls /lustre
scratch114  scratch116 scratch118  scratch123      scratch125  scratch127
scratch115  scratch117 scratch119  scratch124      scratch126
```

Singularity exec

Singularity exec runs a command within a container.

```
$ module load cellgen/singularity
$ singularity exec \
      -B /lustre,/nfs \
      /nfs/cellgeni/singularity/images/python-3.7.sif \
      python3 /path/to/script.py
```

Bsub + Singularity example

```
#BSUB -G farm-course
#BSUB -q "normal"
#BSUB -n 1
#BSUB -M 4000
#BSUB -R "select[mem>4000] rusage[mem=4000]"
#BSUB -o "output%J.log"
#BSUB -e "error%J.log"

module load cellgen/singularity

singularity exec \
   -B /lustre,/nfs \
   /nfs/cellgeni/singularity/images/python-3.7.sif \
   python3 /path/to/script.py
```

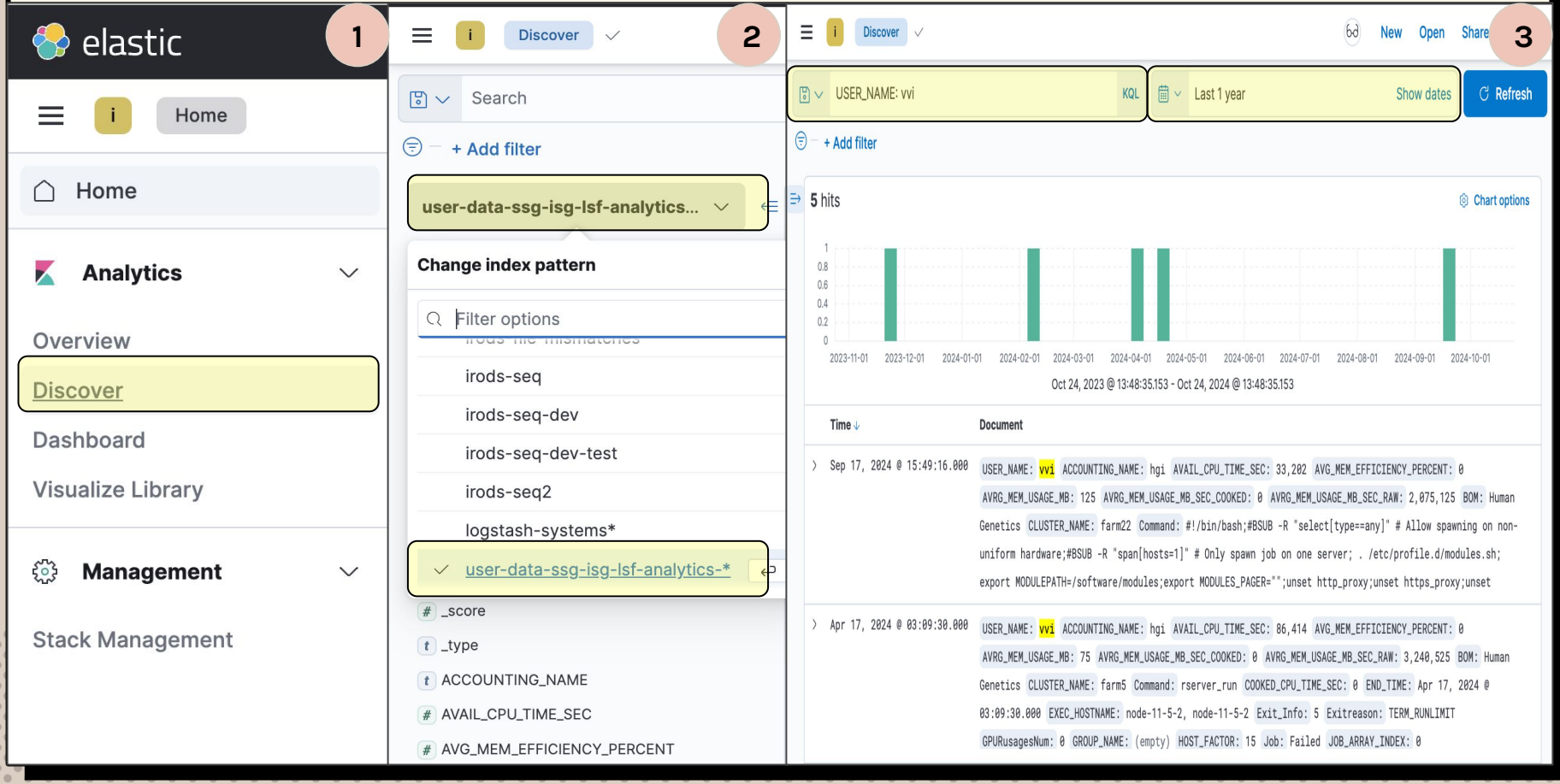
Metrics log for jobs

Sanger maintains a log of all the jobs metrics in an Elasticsearch index. You can use this database to query metrics from past jobs.

The stats will be reported similarly to what the LSF output shows. They are aggregated at the end of each job and not timepoints across your jobs.

https://elastic.internal.sanger.ac.uk/
User: public
Password: public

https://elastic.internal.sanger.ac.uk/



Conda

DO NOT INSTALL YOUR OWN CONDA

You must use the centralized conda module.

```
module load ISG/conda
conda create -n <YourEnv> python=3.11
conda activate <YourEnv>
```

The conda module will create all new environments in the right place:
ie: ***/software/conda/users/<USER>***
ℹ️ Conda commands (install/remove/update) need to be done on head nodes because **/software is not writable from working nodes**.

Conda

The institute has a block on anaconda.com packages. You must only use community channels. These repositories allowed are **conda-forge** and **bioconda**. Contact your informatics team if you are unsure about a channel. **DO NOT USE ANY OF THE FOLLOWING CHANNELS:**

- `defaults`
- `anaconda`
- `main`
- `free`
- `r`
- `pro`
- `msys2`

Bsub + Conda example

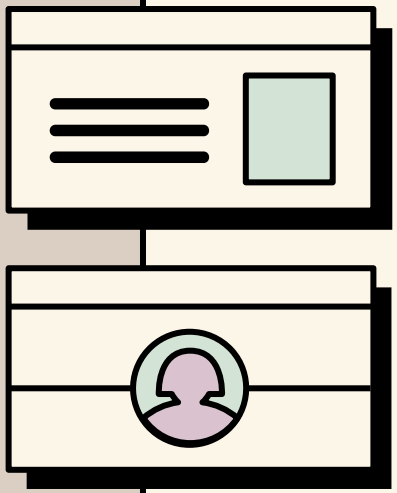
```
#BSUB -G farm-course
#BSUB -q "normal"
#BSUB -n 1
#BSUB -M 4000
#BSUB -R "select[mem>4000] rusage[mem=4000]"
#BSUB -o "output%J.log"
#BSUB -e "error%J.log"

module load ISG/conda
conda activate MyEnv
python my_custom_script.py
```

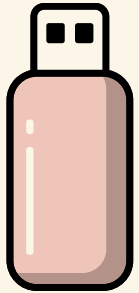
Additional links

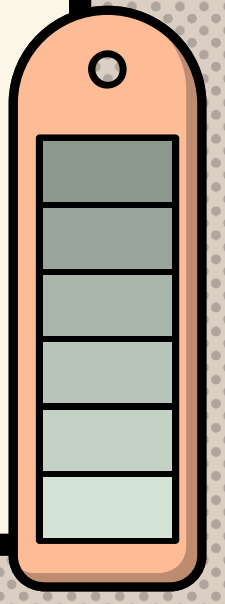
- **FARM FAQ:**
  - https://ssg-confluence.internal.sanger.ac.uk/display/FARM/Farm+Home
- **IBM official docs:**
  - https://www.ibm.com/docs/en/spectrum-lsf/10.1.0?topic=bsub-options
- **Fred docs:**
  - https://fred.wellcomegenomecampus.org/page/3941
- **Workday Fram Course:**
  - https://wd103.myworkday.com/sanger/learning/course/ee6d0406d4951000b681f6583e850000?type=9882927d138b100019b6a2df1a46018b

Thanks!

Questions?
cellgeni@sanger.ac.uk





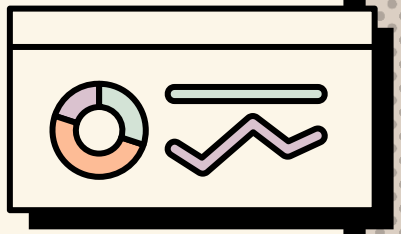


About us

This can be the part of the presentation where you can introduce your company or agency. Keep it short and go straight to the point. Your audience will appreciate it

Meet us

Our evolution



Mercury

Mercury is the closest planet to the Sun

2XXX

Venus

Venus is the second planet from the Sun

2XXX

Earth

Earth is the beautiful planet where we all live

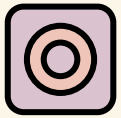
2XXX

Mars

Despite being red, Mars is a very cold planet

2XXX

What sets us apart?



Jupiter

Jupiter is a gas giant and the biggest planet in the Solar System. It was named after the Roman god

Saturn

Saturn is a gas giant and has several rings. It was named after the Roman god of wealth and agriculture

Neptune

Neptune is the farthest planet from the Sun. It's also the fourth-largest planet by diameter in the Solar System

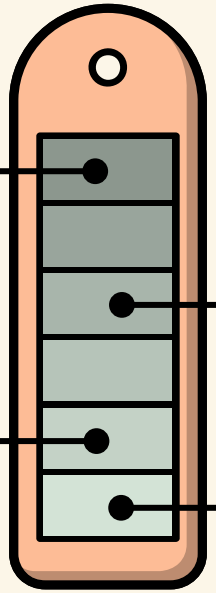
What do we do?

Mercury

Mercury is the closest planet to the Sun and the smallest of them all

Venus

Venus has a beautiful name and is the second planet from the Sun



Earth

Earth is the third planet from the Sun and the only one that harbors life

Mars

Despite being red, Mars is actually a cold place. It's full of iron oxide dust

How do we do it?

Jupiter

Jupiter is the biggest planet in the Solar System

Saturn

Saturn is a gas giant and has several rings

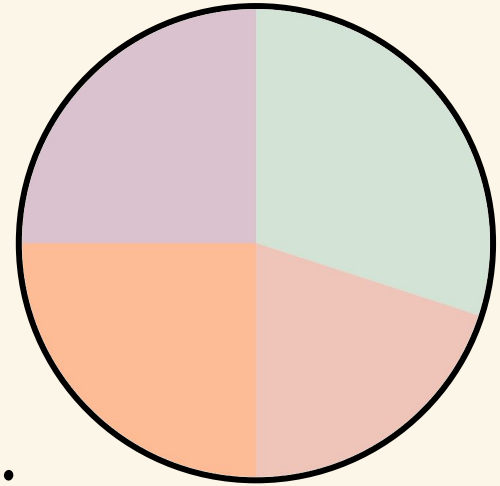
Neptune

Neptune is the farthest planet from the Sun

Sun

The Sun is the star at the center of the Solar System

What do we offer?



Follow the link in the graph to modify its data and then paste the new one here. **For more info, click here**

20%

Mercury

Mercury is the very small

25%

Venus

Venus has a beautiful name

25%

Earth

Earth is where we all live

30%

Mars

Mars is actually a cold place

Our clients

Mercury

Mercury is the closest planet to the Sun

Venus

Venus is the second planet from the Sun

Mars

Despite being red, Mars is actually cold place

Jupiter

Jupiter is the biggest planet of them all

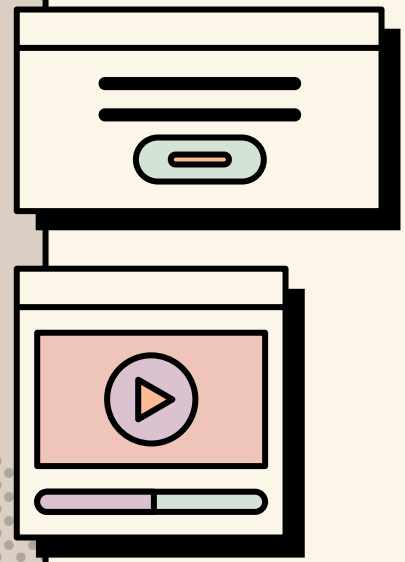
Saturn

Saturn is composed of hydrogen and helium

Neptune

Neptune is the farthest planet from the Sun

What do they say about us?



“Mercury is the closest planet to the Sun and also the smallest one in the Solar System”

—Laura Patterson

“Venus has a beautiful name and is the second planet from the Sun. It’s terribly hot, even hotter than Mercury”

—Monique Nelson

“Earth is the third planet from the Sun and the only one that harbors life in the Solar System”

—John James

Case studies

Mars	Despite being red, Mars is actually a cold place	It's full of iron oxide dust, which gives the planet its reddish cast
Jupiter	Jupiter is a gas giant and the biggest planet	It was named after the Roman god of the skies and lightning
Saturn	Saturn is a gas giant composed mostly of hydrogen and helium	It was named after the Roman god of wealth and agriculture
Neptune	Neptune is the farthest planet from the Sun	It's also the fourth-largest planet by diameter in the Solar System

Awards

- Name of the award (Organization), 2XXX
- Name of the award (Organization), 2XXX
- Name of the award (Organization), 2XXX
- Name of the award (Organization), 2XXX
- Name of the award (Organization), 2XXX
- Name of the award (Organization), 2XXX
- Name of the award (Organization), 2XXX
- Name of the award (Organization), 2XXX
- Name of the award (Organization), 2XXX
- Name of the award (Organization), 2XXX
- Name of the award (Organization), 2XXX

Meet the team

John Doe

You can replace the image with your own

Meet

Helena James

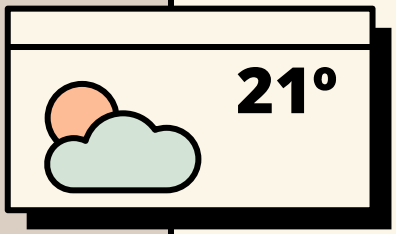
You can replace the image with your own

Meet

Awesome words

An image reinforces the concept

Images reveal large amounts of data, so remember: use an image instead of a long text. Your audience will appreciate it



141,994

Big numbers catch your audience’s attention

9h 55m 23s

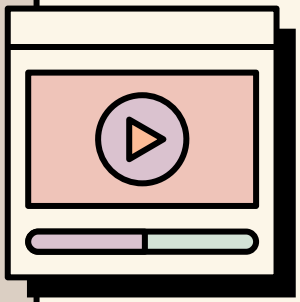
Jupiter's rotation period

333,000

The Sun’s mass compared to Earth’s

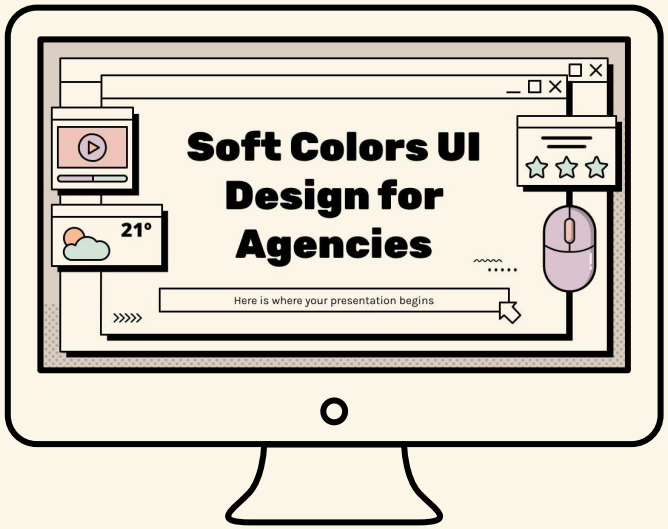
386,000 km

Distance between Earth and the Moon

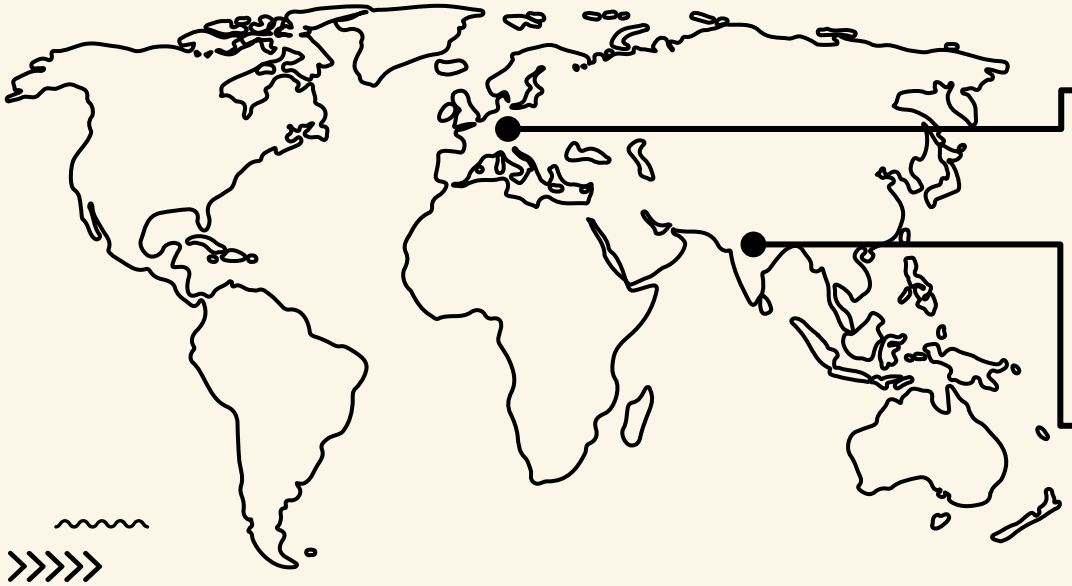


Mockup

You can replace the image on the screen with your own. Just right-click on it and select “Replace image”



We are international



Netherlands

Despite being red, Mars is actually cold place

India

Neptune is the farthest planet from the Sun



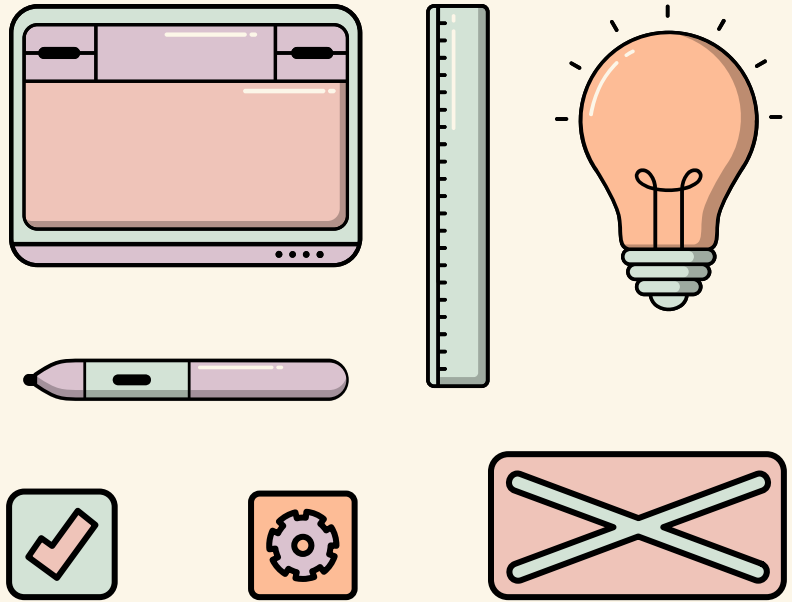
“This is a quote, words full of wisdom that someone important said and can make the reader get inspired.”

—Someone Famous

Alternative resources

Here’s an assortment of alternative resources whose style fits the one of this creative template:

- Linear flat ui/ux background
- Graphic design creative process



Resources

Did you like the resources on this template? Get them for free at our other amazing websites:

PHOTOS

- People working as a team company I
- People working as a team company II
- People working as a team company III
- People working as a team company IV

VECTORS

- Linear flat ui/ux background
- Graphic design creative process