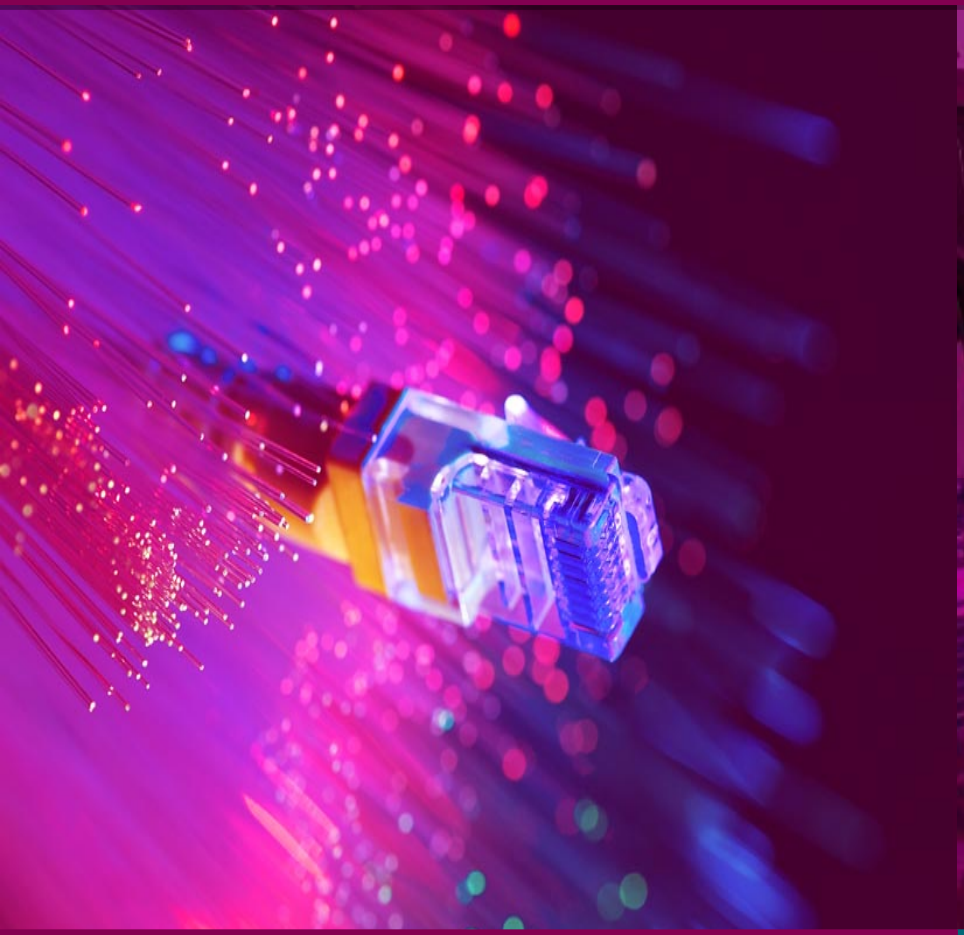
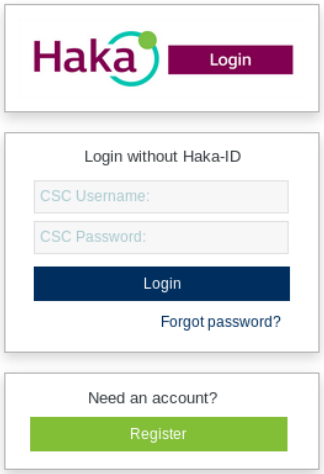
Three steps for getting access to CSC resources
(for non-commercial research)



Getting access: 1.Register yourself



- Register to get a CSC user account
  - Self service for Haka account users:
  - https://sui.csc.fi/web/guest/home
  - Others need to send mail to: servicedesk@csc.fi
  - You get a user account and a personal test project
  - Access to Chipster
  - Your institute account is different (used e.g. in HAKA)

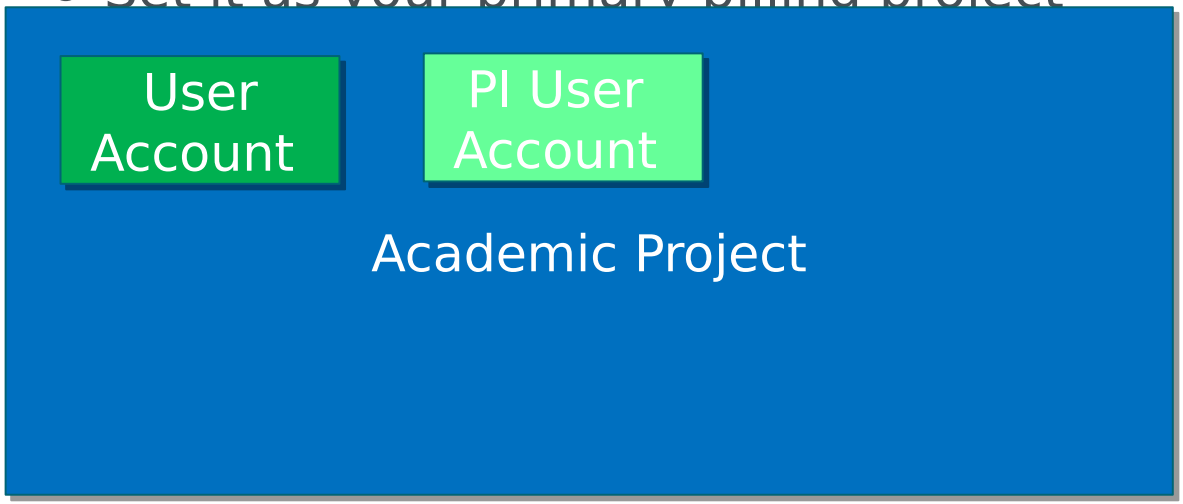
User Account

Getting access: 2. Project membership

- Apply for an Academic Project or join an existing one
  - You can apply for a computing project if you can act as the Principal Investigator

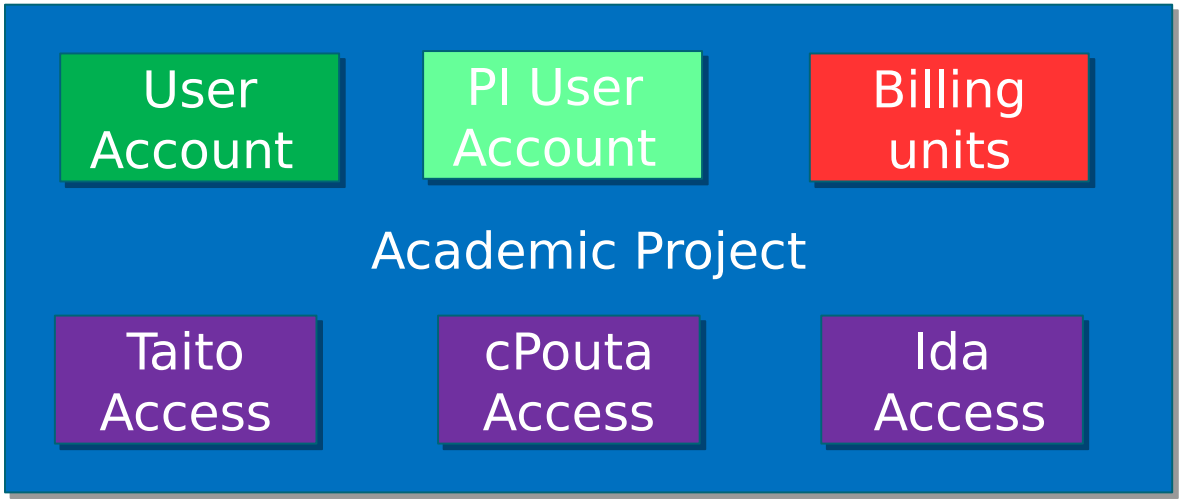
    https://sui.csc.fi/group/sui/resources-and-applications

  - ask PI of an existing CSC project to invite you to it
  - Set it as your primary billing project

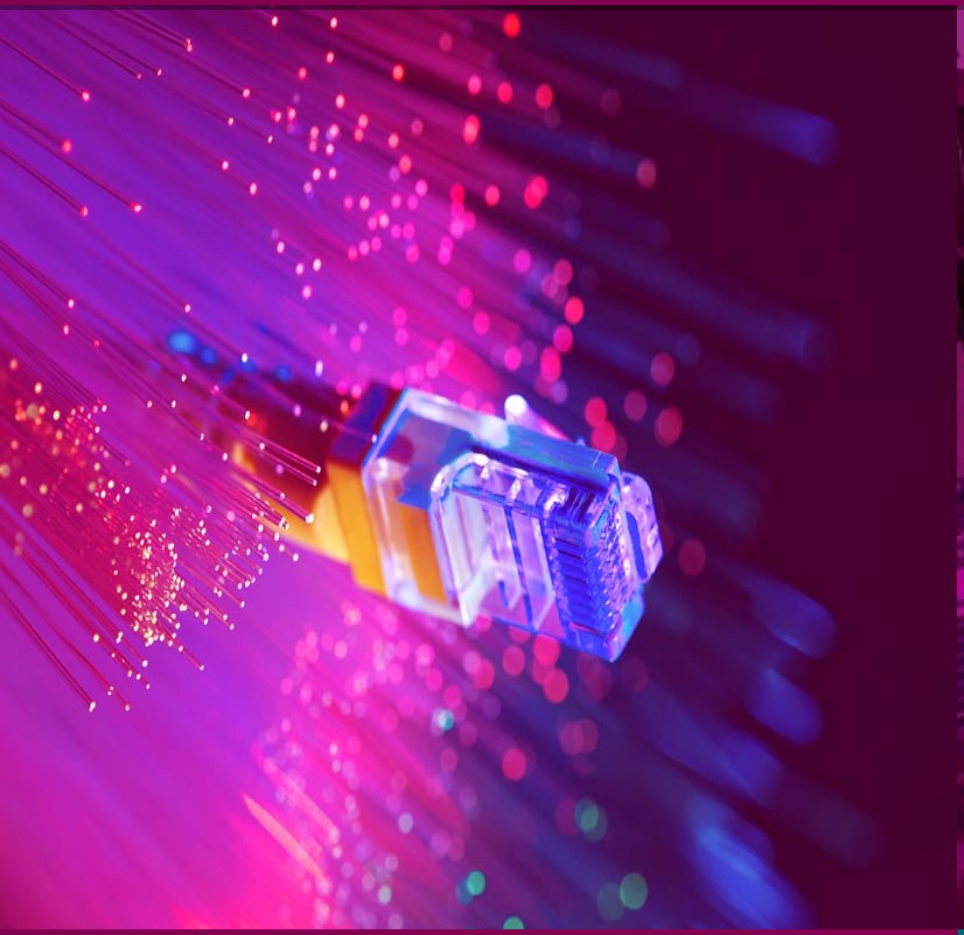


Getting access: 3. Apply for resources and services

- Principal Investigator can apply for billing units and access for services *e.g.* Taito cluster or cPouta access (https://sui.csc.fi/group/sui/resources-and-applications)
- When billing units are applied the PI must report the planned usage and the results obtained with the resources used so far (



Managing research data at CSC



Data services linked to your CSC account

Academic researchers in Finland can apply access for:

- High performance computing (HPC) environment
- cPouta cloud environment
- IDA research data storage service

https://research.csc.fi/accounts-and-projects

Usage is (mostly) free for academic research

Storage areas on the HPC environment of CSC

HPC environment:

- Sisu supercomputer (about 40 000 cores)
- Taito supercluster (about 18 000 cores)
- New servers coming on 2019

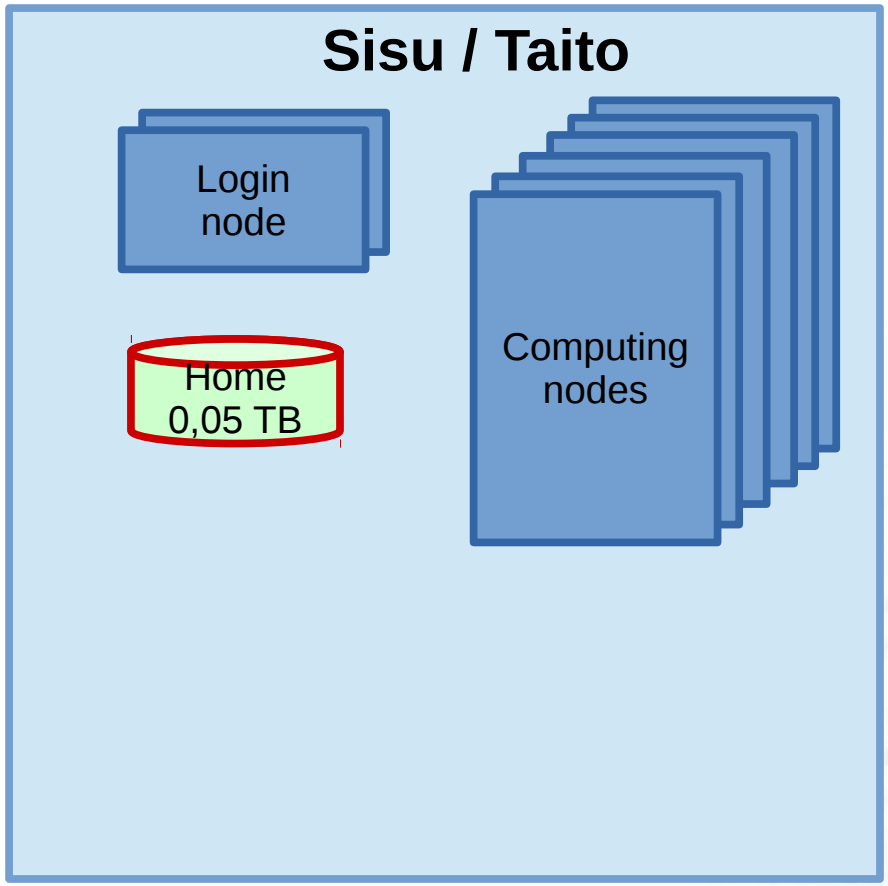
- Directories in the HPC environment are intended for processing your data - not for long term storage.

- Data can be shared with other CSC users (own group or all users) but it can't be accessed outside Taito and Sisu.

Common features for managing data at CSC

- Services are available as long as the account and project is valid.
- user owns the data in HPC,
- project owns the data cPouta and Ida
- no backup, no undo option
- no automatic data management available (if you don't create it)
- data can be technically anything but there are legal restrictions for example for sensitive data.

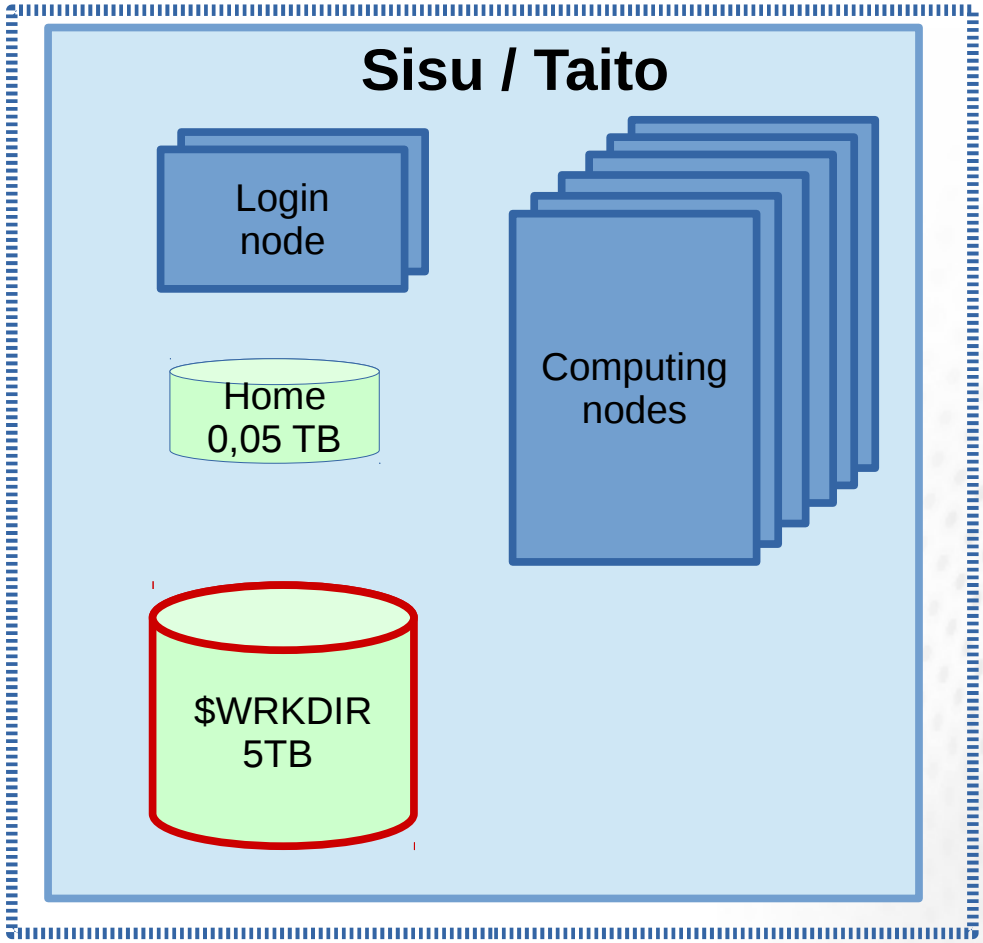
Storage areas on the HPC environment of CSC



$HOME

- 50 GB
- **daily backups**
- settings and small data (program codes, parameter files etc.)
- not for massive data

Storage areas on the HPC environment of CSC



$WRKDIR

- 5 TB
- intended for data analysis
- automatic cleaning of files that have not been used for 90 days
- includes DONOTREMOVE area that is not cleaned

Object Storage in cPouta

- Content Agnostic, Distributed, Scalable, Eventual Consistent and **Highly available** Data Storage.
- Access and data ownership is based on projects (no personal ownership)
- BU consumption same as existing block storage i.e. 3.5 BU/TiB/hr
- Initial quota for object storage/project is 1 TiB.
  - Buckets per project: 1000
  - Objects per Bucket: 100000
  - Object Size: 5GB
  - For objects > 5GB, split the object into smaller segments.
- Access control possible for buckets/objects.
- REST API available, S3 and Swift API compatible, Supports WebUI and Swift/S3 CLI tools.
- No automatic backup

IDA- research data storage

- Research data storage service for Finnish universities and research institutes.
- Project based quotas granted by local organizations but managet by CSC
- Quotas up to several terabytes
- Usage through **WWW interface** or **ida** command line client
- Part of the Fairdata.fi service: https://www.fairdata.fi/
- Other Fairdata tools include:
  - **Qvain** metadata mangement tool
  - **Etsin** reaserch data finder
  - **Fairdata PAS** data preservation service

Moving data to and from CSC

IDA
Reseach
Data storage
X00 GB

Etsin

Sisu / Taito

Login node

Home
0,05 TB

Computing nodes

$WRKDIR
5TB

Project directory
1-50TB

wget, curl

WWW interface

Browser

Scp
Rsync
FileZilla

s3cmd

Pouta Object
Storage
x TB

- WWW
- Local client
- s3cmd

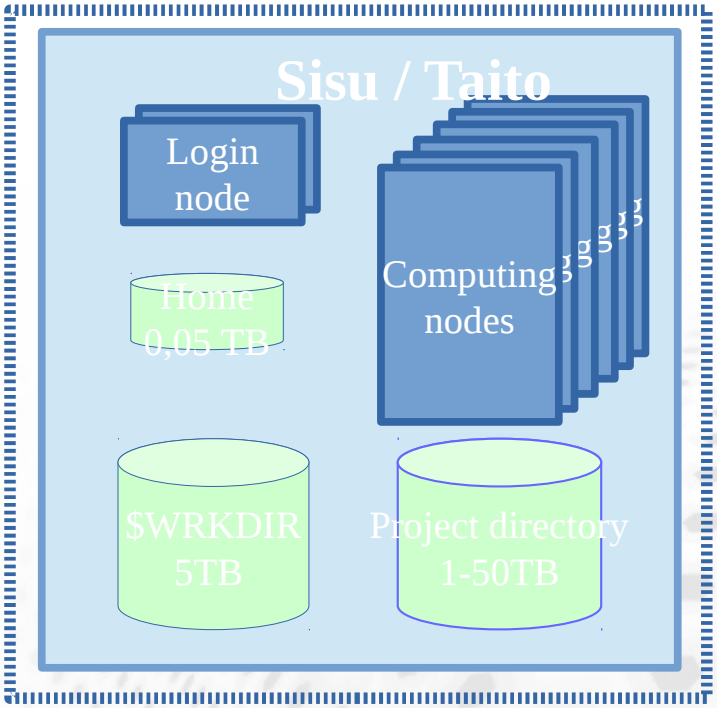
Your colleague

Stoage size is not the only issue, transfer rate matters too !

Task for Human Genome: **1 GB** (Fasta formatted, Gzip compressed uncompressed size would be: **49 GB**)	Time	Rate	Time for transfering one Terabyte
Download (wget) from EBI (UK) to Taito	4 m40 s	13 GB/h	3 days
Download (wget) from EBI (UK) to cPouta	34 s	105 GB/h	9,5h
Download (wget) from EBI (UK) to a laptop through slow network	45 min	1,3 GB/h	31days
Upload form Taito to Pouta OS	20 s	180 GB/h	5,5h
Copying from $WRKDIR to Project directory in Taito	3 s	1200 GB/h	50 min

Reasearch project dataflow example

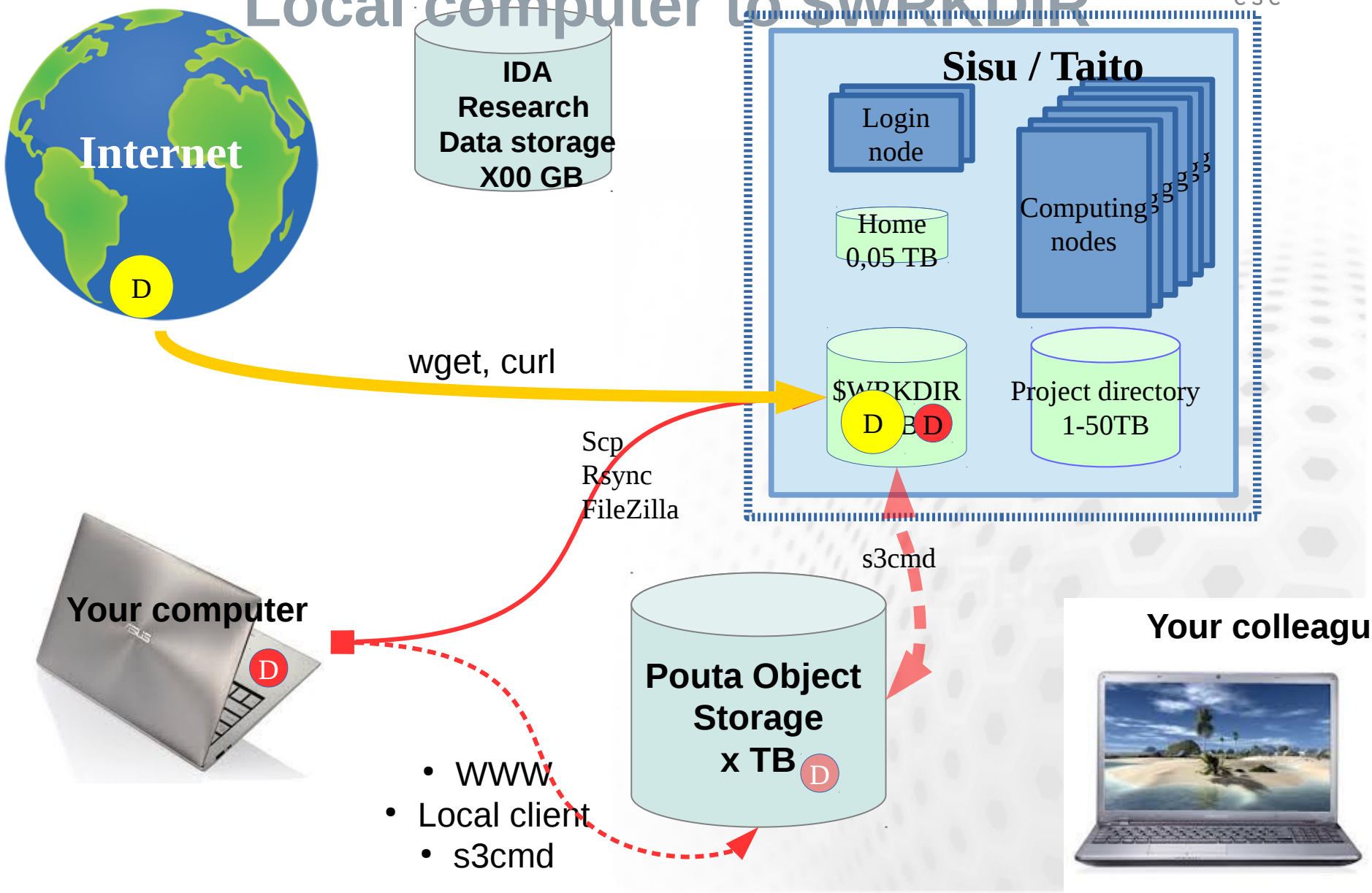
IDA
- Research
- Data storage
- X00 GB



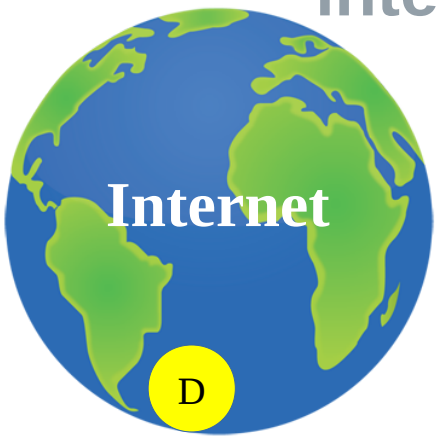
Pouta Object Storage x TB

Your colleague

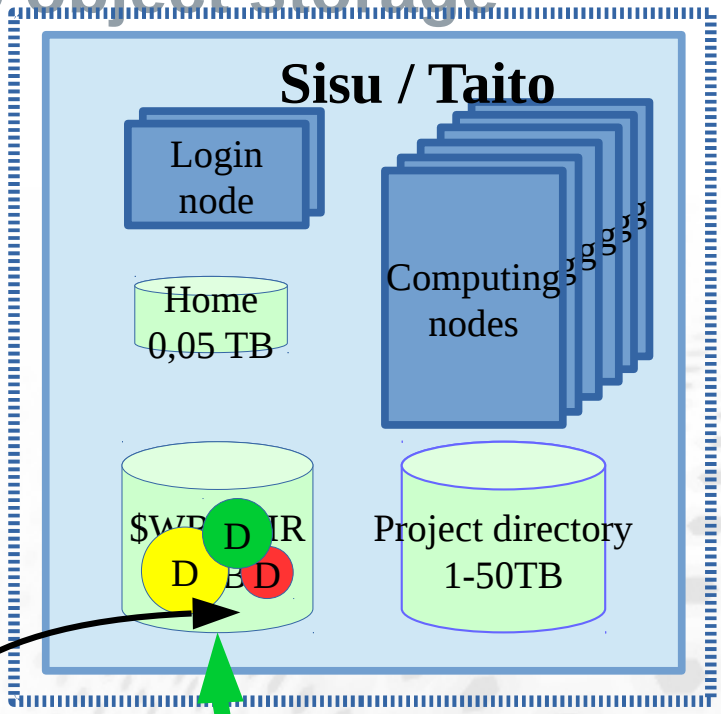
1. Copy the data from internet and your Local computer to $WRKDIR



2. Process your data at CSC and backup intermediate steps to object storage



IDA Research Data storage X00 GB



s3cmd

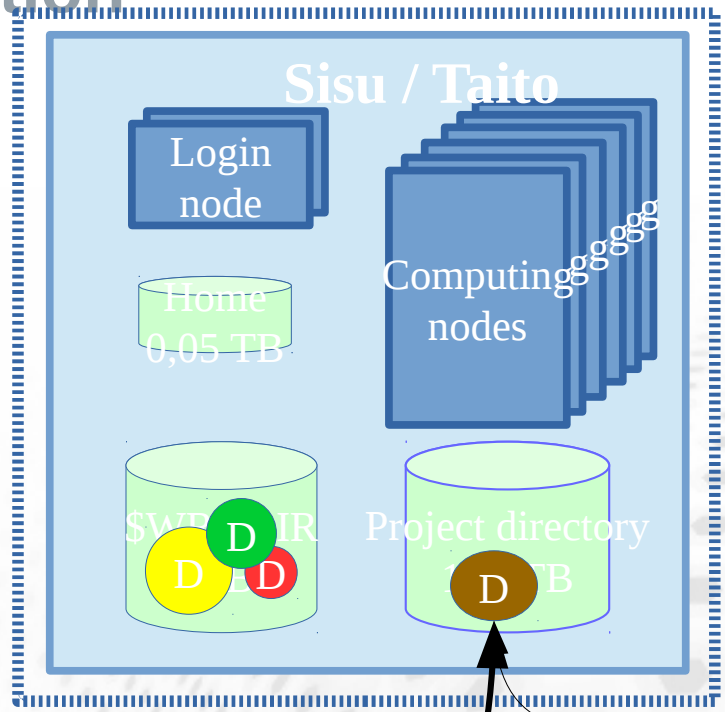
Pouta Object Storag x TB D D

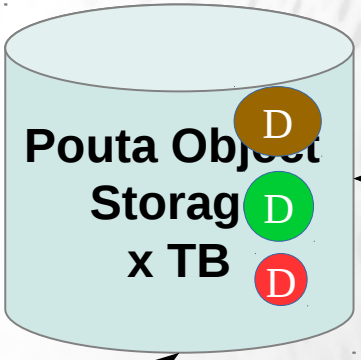
- WWW
- Local client
- s3cmd

Your colleague

3. Use Object Storage and project directory for collaboration

IDA
- Research
- Data storage
- X00 GB





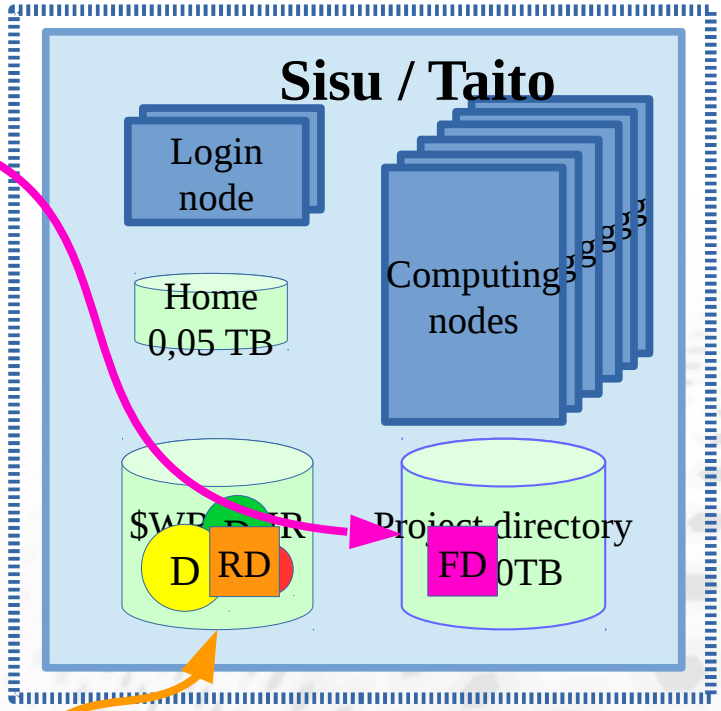
s3cmd

Your colleague

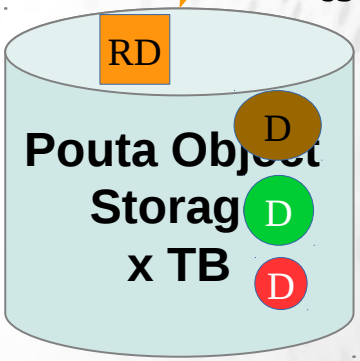
- WWW
- Local client
- s3cmd

4. Store your final resuts to IDA or object storage

IDA

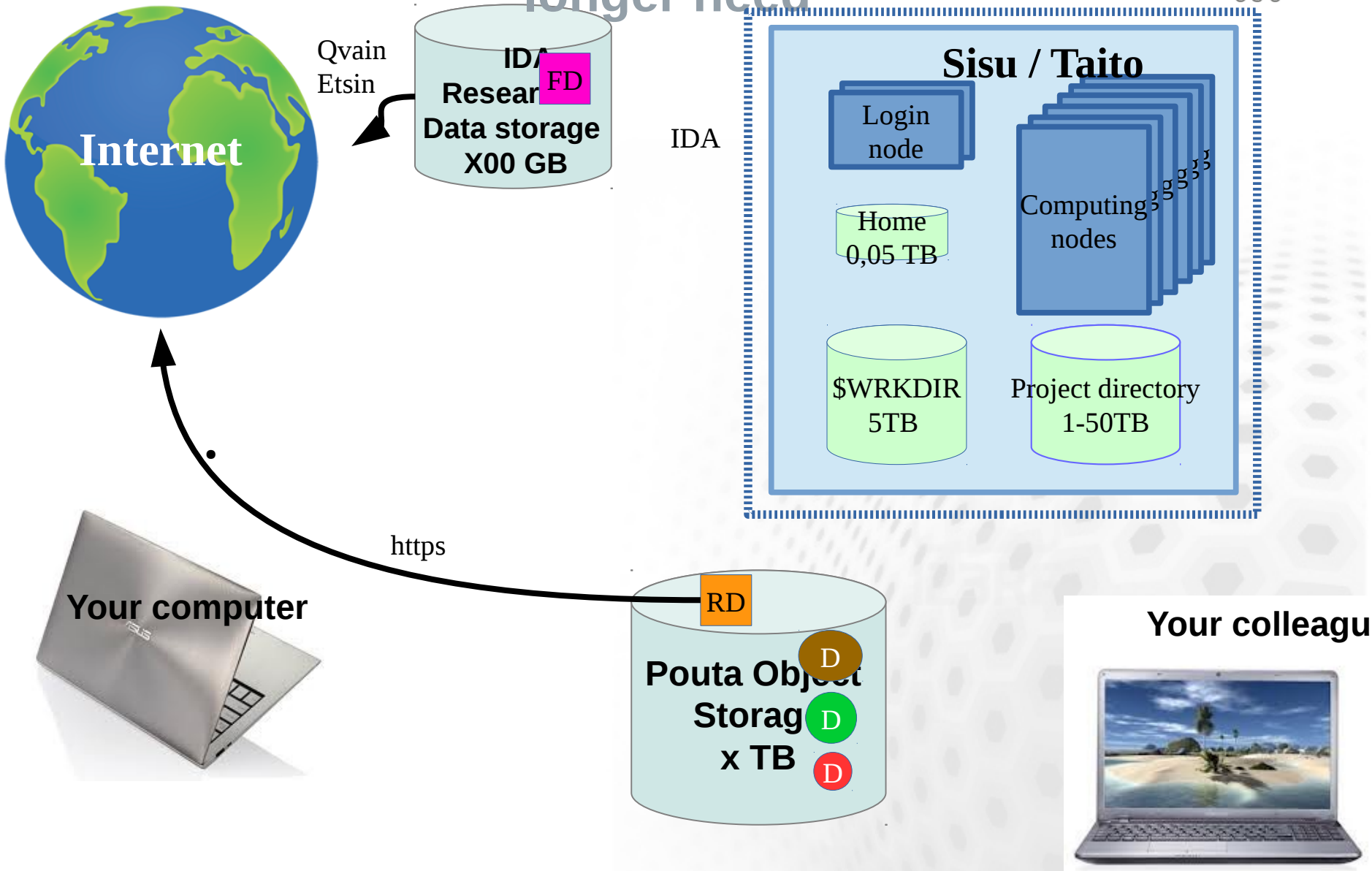


s3cmd



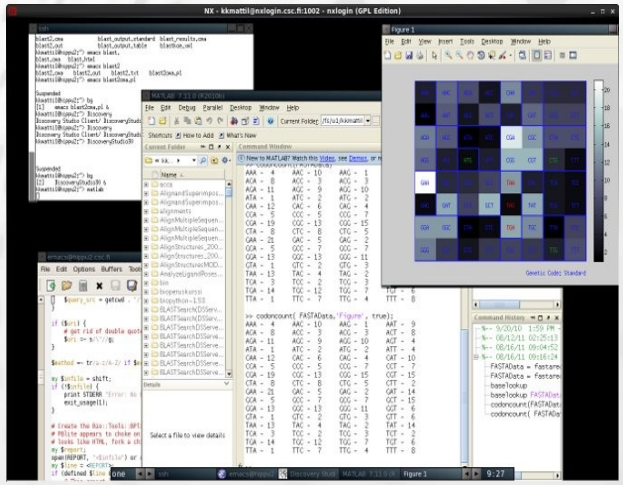
Your colleague

4. Publish your data and clean the files you no longer need



Connecting Servers of CSC

- Terminal connections (ssh, PuTTY, SUI)
  - usage through typed commands
  - Graphics requires Xterm connection
- Scientist's User Interface
  - Usage through web interface
  - Mostly used for managing your account and files
  - No bioscience applications
- NoMachine/FreeNX virtual desktop
  - requires local client installation
  - Norman terminal connections can be used
  - Enables using grapical interfaces and displaying images



http://research.csc.fi/csc-guide-connecting-the-servers-of-csc

Managing files in unix command line

http://research.csc.fi/csc-guide-linux-basics-for-csc

Unix/linux commands

Basic syntax:

comand -option argument

```
ls
ls -l
ls -l myDirectory
```

Use *man* command to get information about possible options

```
man ls
```

Commands for directories:

`cd`	change directory
`ls`	list the contents of a directory
`pwd`	print (=show) working directory
`mkdir`	make directory
`rmdir`	remove directory

Commands for files:

`cat`	print file to screen
`cp`	copy
`less`	view text file
`rm`	remove
`mv`	move/rename a file
`head`	show beginning of a file
`tail`	show end of a file
`grep`	find lines containing given text
`wc`	count number of words or lines
file	check the type of the file

Special characters:

*(asterisk), wild card, means any text

```
ls *.fasta
```

| (pipe) guides output of a command to an input of another commands

```
ls *.fasta | less
```

> Writes output to a new file

```
ls > files_of_the_directory.txt
```

>> Appends output to an existing file

ls *.fasrta >> `files_of_the_directory.txt`

~ (tilde) means your home directory as does $HOME

```
cp test.txt ~/file.txt
cp text.txt $HOME
```

& runs command in background

```
gzip my_big_file.tar  &
```

Quotes

- Different quotes have different fuctionalities

  ' ' Take text enclosed within quotes literally

  ` ` Take text enclosed within quotes as command and replace with output

  " " Take text within quotes literally after substituting any variables

- Compare the results of these commands:

```
set var = "test"; echo 'echo $var'
set var = "test"; echo `echo $var`
set var = "test"; echo "echo $var"
```

Some useful commands for parsing lines

Try these to see what they do!

sed

```
echo "one this two this three" | sed s/this/that/
echo "one this two this three" | sed s/this/that/g
```

awk

```
echo ”one two three” | awk ’{print $2}’
echo ”one;two;three” | awk –F”;” ’{print $2 $3}’
```

cut

```
echo ”123456789” | cut –c 4
echo ”123456789” | cut –c -4
echo ”123456789” | cut –c 4-
echo ”123456789” | cut –c 4-7
echo ”one_two_three” | cut –d ”_” –f 2
```

All of these have much more options. See man pages for details.