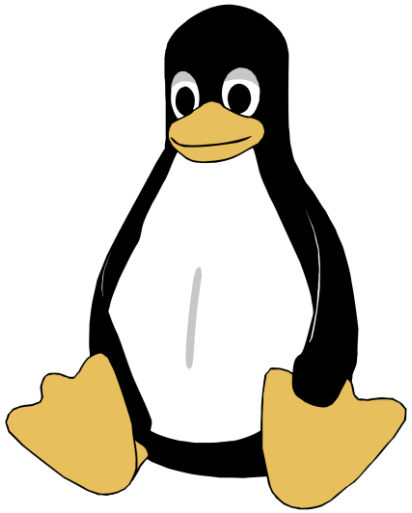
ZMT LEIBNIZ CENTRE for Tropical Marine Research

Leibniz Association

Linux command line

Tutorial 11.03.2019

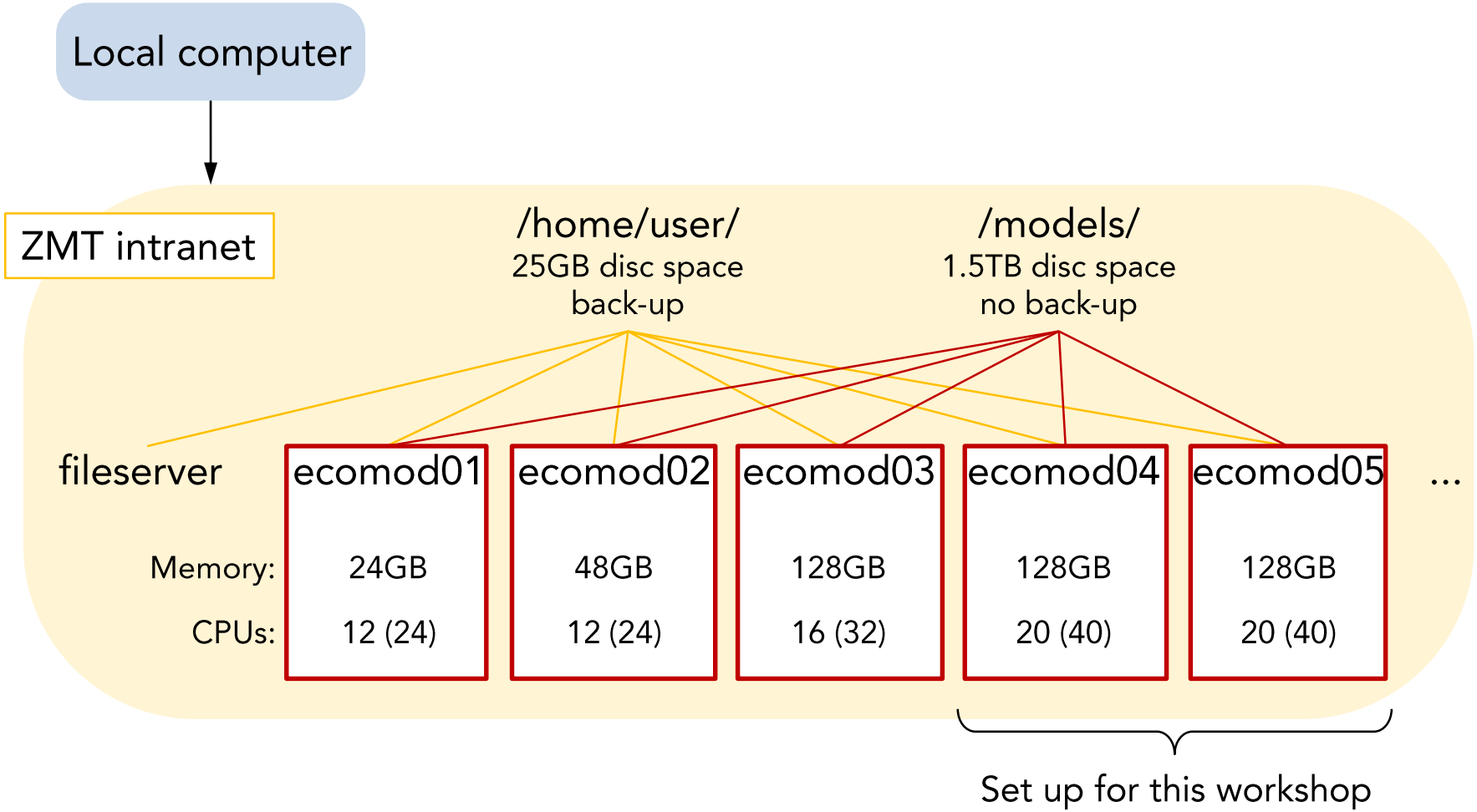


Outline

- ZMT computing resources
- Configuring your account
- SSH keys
- Resource managers
- Sequence file formats
- Basic linux commands, paths
- Input/output, redirect/pipe, line endings
- Regular expressions
- Variables
- Working environment (modules)
- Screens

https://zmtcloud.zmt-bremen.de/index.php/s/Mkgty4KxUpJ3qsi

ZMT computing resources



Configuring your account

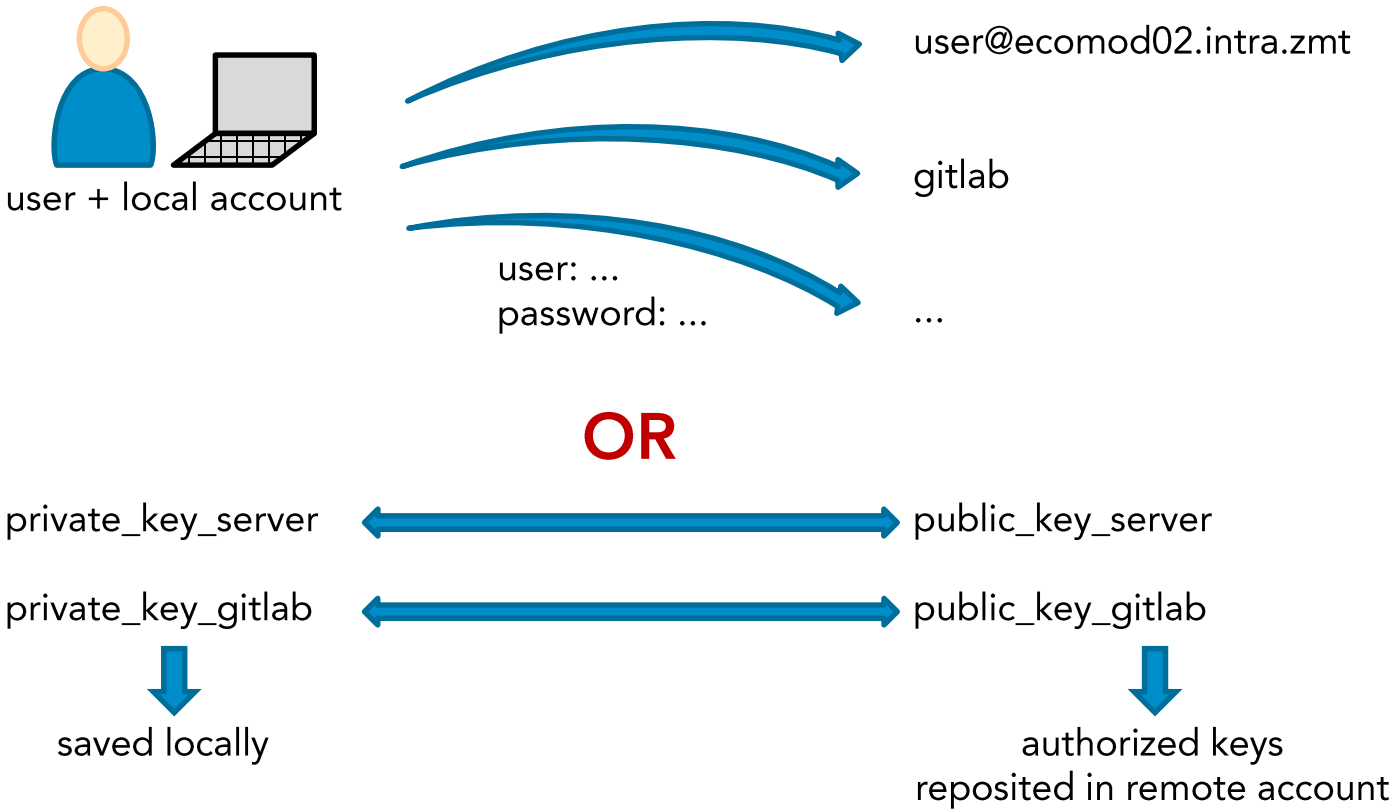
- Get access to ZMT intranet (or your institute/university network)
  - ZMT computer: direct access
  - Alternative: VPN connection

- Windows:
  - PuTTY (http://www.putty.org/) or MobaXterm
  - Xming (https://sourceforge.net/projects/xming/)
  - Tutorial on: http://www.geo.mtu.edu/geoschem/docs/putty_install.html
  - File transfer: FileZilla or WinSCP

- Mac:
  - Start Xquartz
  - Open command line
  - Connect via: ssh -X user@ecomod05.leibniz-zmt.zmt
  - File transfer: scp (secure copy)

SSH keys

- Instead of using passwords



- E.g. PuTTy gen (windows), ssh-keygen (linux, mac)

Resource managers

→ Only use half of the available CPUs

→ Monitor memory usage

→ `top` or `htop`

- Torque, **SLURM**, Sun Grid Engine, ...

Simple Linux Utility for Resource Management

=

The Respectful Way to Run your Analysis

Resource managers

squeue: display information on running and queued jobs

```
squeue -o "%.18i %.9P %.8j %.8u %.2t %.10M %.6D %R %C %m"
```

jobid partition name user status time nodes nodelist cpus memory

default

scancel *jobid*: cancel execution of queued and running jobs

srun: submit jobs to queue

--cpus-per-task=4	reserve 4 cpus for that taks (default: 1 CPU)
--mem-per-cpu=2000	each CPU needs at least 2GB of memory
-x ecomod01,ecomod02	do not run on ecomod01 and ecomod02
--exclusive=ecomod05	run exclusively on ecomod05 (no other users allowed)
-J *name*	specify job name
--mail-type=BEGIN,END	send email at when job is started and when it has finished
--error *file* –output *file* (or 1>*file* 2>*file*)	redirect error and output to files
--time	specify run time limit for job (default: no limit)

Sequence file formats

- Fasta:

>Sequence accession
Sequence...

```
>SNL168:111:H2YY7BCXY:1:1101:11589:1946 1:N:0
TGGGGAATATTGCACAATGGGGGAAACCCTGATGCAGCCATGCCGCGTGTATGAAGAAGG
```

- Fastq:

@Sequence accession
Sequence...
+
Base quality scores....

```
@SNL168:111:H2YY7BCXY:1:1101:11589:1946 1:N:0
TGGGGAATATTGCACAATGGGGGAAACCCTGATGCAGCCATGCCGCGTGTATGAAGAAGG
+
GGDEC:@<DCDFGGGGGGGGG7C,9,<,6:668+6=,,<,,,<9,5FA8C,,,,<CCF,5
```

→ Simple text files!

Basic linux commands

- General commands (selection): http://linuxcommand.org/tlcl.php

List contents of directory - including hidden files - all attributes - by date modfied (in reverse order) - human-readable output - list in 1 column - in numerical order - list all fasta files (* wildcard)	ls -a -l -tr -h -1 -v *.fasta
Change directory - home - root - scratch directory	cd ~ / /models/

- Navigate your file system:
  - Absolute file paths: always starting with /
  - Relative file paths: starting from yout current location
  - Display current location: `pwd`

Basic linux commands

- General commands (selection): http://linuxcommand.org/tlcl.php

List contents of directory - including hidden files - all attributes - by date modfied (in reverse order) - human-readable output - list in 1 column - in numerical order - list all fasta files (* wildcard)	ls -a -l -tr -h -1 -v *.fasta
Change directory - home - root - scratch directory	cd ~ / /models/
Create directory	mkdir
Copy file or directory	cp
Move/rename file or directory	mv
Remove file - remove by force	rm -rf
Remove empty directory	rmdir
Change permissions - add read and write permissions for all categories - remove write permissions (for everyone) - add executable permissions (for group)	chmod +rw a-w g+x

Basic linux commands

- General commands (continued): http://linuxcommand.org/tlcl.php

Show first/last n lines of file	head/tail -n
Scroll through file - truncate lines to window size - scroll forwards - scroll backwards	less -S [space key] [b key]
Count words - count lines	wc -l
Search for pattern - count lines with match - show 5 lines before and after match - lines without match - read patterns from file	grep "*pattern*" -c -B5 -A5 -v -f
Stream editor - substitute text - delete line with pattern	sed 's/*search*/*replace*/' '/*pattern*/d'
For loops	for i in *instances* do *command* done
While loops	while read line do *command* done < input.file

Input/output

- Standard input: from keyboard
- Standard output: to console

- Redirect input/output:

Redirect to file	>
Append to file (or create file if not existing)	>>
Pipe to next command	|

Line endings

- Windows: ^M \r \n
- Linux: $ \n

→ May cause incompatibilities of scripts and search patterns

Regular expressions

Most often used with grep and sed (and perl) to define search pattern

Special characters:

Any number, lowercase, uppercase letter n number of times	[0-9], [a-z], [A-Z]{n}
Escape special meaning	\
Any character any number of times	.*
Beginning of line	^
End of line	$
Comment	#
Soft quotation (allow special characters)	"
Hard quotation (do not allow special characters)	'

Examples:

`ls Sample[0-9]*_R[1-2].fastq`	`Sample1_R1.fastq`
	`Sample1_R2.fastq`
	`Sample23_R1.fastq`
	`Sample23_R2.fastq`

`sed "s/^>/>"${SID}"\_/" ${SID}".fasta"`	`>Sample1_skldjfhksjfd`
	`AGCTAGATCGATC`
	`>Sample1_ksjdhflkhasd`
	`GCTATGTACCATGG`

Variables

- Environment variables:

  Location of program executables: `echo ${PATH}`

- User-specified variables:

  Location of scripts for 16S amplicon analysis:

```
SCRIPTS="/models/tmm/Scripts/bioinf"
echo ${SCRIPTS}
```

  Loops:

```
while read line
do
  echo ${line}
done < input.file
```

Setting up your environment

- .bashrc:
  - Containes command to configure your environment
  - E.g. $PATH, aliases
  - Loaded (sourced) when you open a new interactive session

- Modules:
  - Dynamic modification of your $PATH
  - add and remove programs → switch between different versions of the same program
  - https://gitlab.leibniz-zmt.de/chh/bioinf/wikis/using-the-module-system

`module avail`	list available modules
`module load`	load a module (add location of program to your $PATH)
`module list`	list loaded modules
`module unload`	unload module

→ Reload modules for every new interactive session!

Screen

- Keep your interactive session from crashing if the network connection is lost

`screen -ls`	list screens
`screen -S` <name>	start named screen
ctrl + a + d	detach screen
`screen -r` <name>	resume
`screen -D`	detach attached screen after crash
ctrl + a shift + k	kill attached screen

Let‘s get started

Task 1: Open a terminal, navigate to a suitable location and create a directory for this workshop

Task 2: Move/Copy the example data set to this directory https://zmtcloud.zmt-bremen.de/index.php/s/Mkgty4KxUpJ3qsi

Task 3: Unzip the files

Task 4: List the files, have a quick look at the contents, count the R1 and R2 files

Task 5: Extract all sequences with “@MISEQ:41:000000000-A9A9U:1:1101“ in the header in CH_A_1_SB_clip_R1.fastq, and save them in a new fastq file

Task 6: Shorten file names by removing “_SB_clip”, and generate a text file with sample names

Task 7: Count the number of sequences per sample

Task 8: Move the sequence files into a new directory corresponding to the stage of analysis they are in

Task 9: Check what is in your .bashrc, modify it if necessary

Task 10: Check what is in your $PATH, locate R and python 2 (using modules)