

Programms Cheat Sheet

Console Toolkit II

Formatting

Text which is underlined has to be replaced by the associated information.

Text which is grey, is optional.

SSH

```
ssh -p port username@address
```

```
ssh ssh://username@address:port
```

SCP

```
scp -p port source target
```

source & target: user@address:path

RSYNC

```
rsync options source destination
```

source & target: user@address:path

-a	archival
-v	verbose
-u	skip files which are newer in target
-n	dry run
--delete	deletes files in source
-b	back up deleted files
--backup-dir=/backupdirectory	backs up deleted files
--progress	shows progress
-c	checksum instead of modtime & size

ssh-keygen

```
ssh-keygen -f output -t encryption -b bits
```

Encryption^[1]

rsa

An old algorithm based on the difficulty of factoring large numbers. A key size of at least 2048 bits is recommended for RSA; 4096 bits is better. RSA is getting old and significant advances are being made in factoring. Choosing a different algorithm may be advisable. It is quite possible the RSA algorithm will become practically breakable in the foreseeable future. All SSH clients support this algorithm.

dsa

An old US government Digital Signature Algorithm. It is based on the difficulty of computing discrete logarithms. A key size of 1024 would normally be used with it. DSA in its original form is no longer recommended.

ecdsa

A new Digital Signature Algorithm standardized by the US government, using elliptic curves. This is probably a good algorithm for current applications. Only three key sizes are supported: 256, 384, and 521 (sic!) bits. We would recommend always using it with 521 bits, since the keys are still small and probably more secure than the smaller keys (even though they should be safe as well). Most SSH clients now support this algorithm.

ed25519

This is a new algorithm added in OpenSSH. Support for it in clients is not yet universal. Thus its use in general purpose applications may not yet be advisable.

Git

Git config

```
Git --config -global user.name «Username»
```

```
Git --config -global user.email «email address»
```

Working with git

<code>git init</code>	Initialize a git repository
<code>git add <u>file</u></code>	Add files you want to include in a commit
<code>git commit</code>	Create a commit for the selected changes
<code>git push</code>	Push changes to server

Ranger

<code>?</code>	help
<code>q</code>	exit

hjkl, arrow keys	move
r	open with
gn, ctrl+n	open new tab
H, L	move backwards/forwards in History
gg	move to top
G	move to bottom
:	console
!	shell
:set show_hidden true	show hidden files
:set show_hidden false	hide hidden files

MPV

arrow keys move in time

9, 0	Volume up/down
p, Space	pause
[,]	decrease/increase playback speed

Pipes.sh

<https://github.com/pipeseroni/pipes.sh>

```
git clone https://github.com/pipeseroni/pipes.sh
```

```
cd pipes.sh
```

```
make install
```

Archives and compression

Tar and gzip

Create and compress:

```
tar -czvf archive.tar.gz directory -exclude-directory=/directory
```

Inflate:

```
tar -xzvf archive.tar.gz -c /directory
```

Tar and bzip2

Create and compress:

```
tar -cjvf archive.tar.gz directory -exclude-directory=/directory
```

Inflate:

```
tar -xjvf archive.tar.gz -c /directory
```

zip and unzip

Compress:

```
zip -r file.zip /directory
```

Inflate:

```
unzip file.zip -d /directory
```

Continuing Literature

X-forwarding for SSH

Using SSH-keys for ssh and git

Using Euler: scicomp.ethz.ch

Automate stuff: Bash scriptin

Feedback

Feedback always welcome at: noah.marti@thealternative

Sources

1: copied from: <https://www.ssh.com/ssh/keygen/>, 26.10.2020