Ubuntu-Linux Operation: Shell Scripting (Hands-on Practice)

Zainal Abidin

DR. Danang Eko Nuryanto

BMKG

PUSDIKLAT

BMKG Corporate University

WORLD METEOROLOGICAL ORGANIZATION

Regional Training Centre - Indonesia

UNESCO

United Nations Educational, Scientific and Cultural Organization

Intergovernmental Oceanographic Commission

OceanTeacher GLOBAL ACADEMY

Specialized Training Centre - Indonesia

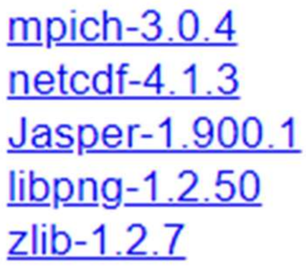
wget

`$ wget [url]`	download a file
`$ wget -c [url]`	resuming if the file is not completed
`$ wget [url] -O output.ext`	rename the output as need
`$ wget [url] -i list.txt`	download files from a list (contains many urls)

`curl` command also can do the same thing and much more make a list of libraries that needed for WRF compilation

https://www2.mmm.ucar.edu/wrf/OnLineTutorial/compilation_tutorial.php#STEP2

copy and paste the url from these links, put in a list.txt



mpich-3.0.4
netcdf-4.1.3
Jasper-1.900.1
libpng-1.2.50
zlib-1.2.7

Compress Data Command

`zip`/`unzip` is commonly used for files or folders

```
$ zip [folder] out.zip
$ unzip file.zip
```

`gzip`/`gunzip` is best used in file

```
$ gzip file
$ gunzip file.gz
```

tar compress/extract

`$ tar zxfv file.tar.gz`	extract a tar.gz file
`$ tar zcfv folder/ file.tar.gz`	compress a folder
`$ tar ztfv file.tar.gz`	view the file list

find Search File

```
$ find . -name *.F
```

search files in current directory
that have .F extension

```
$ find . -name *.txt -exec cp -v {} folder/
```

search files in current directory
that have .F extension and then
exec a command that copy the files to a folder

`find` has many useful options in common cases, consult the manual for details

Search File from Index

`$ locate file.ext`	search the file from file list index of the system
`$ sudo updatedb`	update the file list index of the system

“search” binary files

`$ whatis command`	shows the simple description of the command
`$ whereis command`	shows the binary, source, and manual page files for a command
`$ which command`	shows path of the binary

Soft link

In windows is something link shortcut.

If the link is modified, it will also modify the source file.

If the link is deleted, it doesn't delete the source file

```
$ ln -s SOURCE_FILE SYMBOLIC_LINK
```

It is useful

Pipe |

| Pipe enables us to combines two or more commands, the output of a command acts as an input to another command

In US keyboard, this key is located on the Enter key, combined with \

```
$ first_command | second_command | third_command
```

```
$ grep Chicago us-500.csv| cut -d, -f3
```

Filter us-500.csv with text “Chicago” and show only the company name (3rd column)

Redirection > or >>

`$ ls > filelist.txt`	put **`ls`** output to filelist.txt file
`$ ls WRF/ >> filelist.txt`	append **`ls WRF/`** output to filelist.txt

the **>>** redirection can be monitored by **`tail -f`**

Redirection with Error Message

`$ ls %`	this command will produce an error
`$ ls % >> out.log`	out.log will still empty
`$ ls % 2> out.log`	the error message will be written out.log

Redirection > or >>

```
$ echo “hello” > print.txt
```

put “hello” word in print.txt file

```
$ echo “good morning” >> print
```

append “good morning” in print.txt file

```
$ cat print.txt
hello
good morning
```

the **>>** redirection can be monitored by `tail -f`

Package Management

- Red Hat Package Management (rpm)
  - Red Hat Enterprise Linux (RHEL), CentOS, RockyLinux
- Debian (deb)
  - Ubuntu, Debian, Raspberry Pi OS
- Arch User Repository (aur)
  - Arch Linux, Manjaro, ArcoLinux, Artix Linux

Package Management Command

RPM uses **`dnf`** or **`yum`** command

Deb uses **`apt`** command

Package Management Command

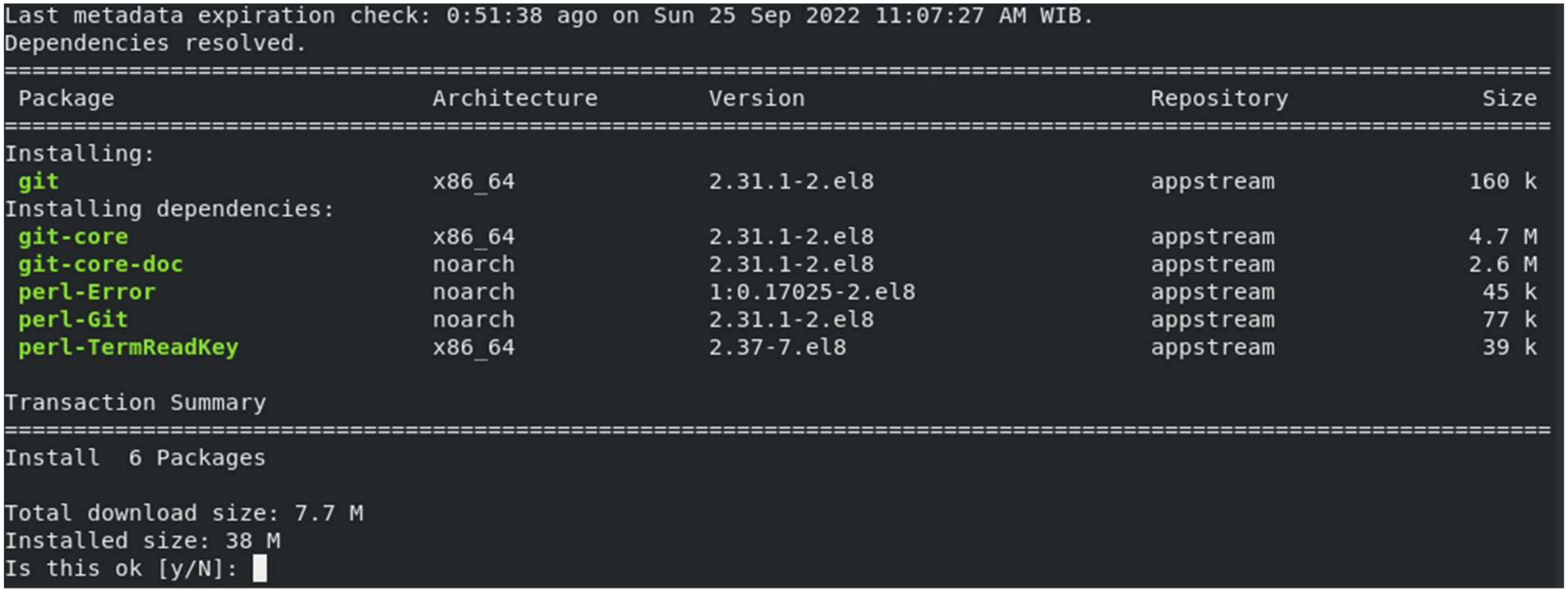
Update

Install Package(s)

- sudo dnf install *package_name*
- sudo apt install *package_name*
- sudo dnf update
- sudo apt update

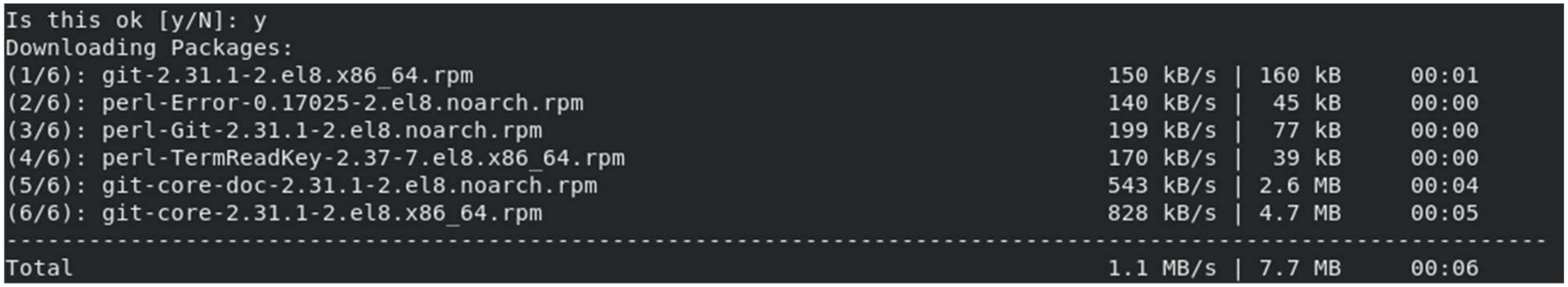
Install Package

$ sudo dnf install git



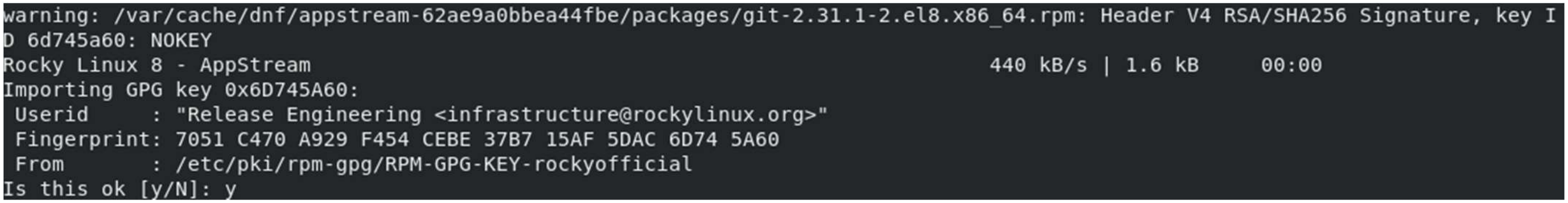
```
Last metadata expiration check: 0:51:38 ago on Sun 25 Sep 2022 11:07:27 AM WIB.
Dependencies resolved.
================================================================================================================
 Package                        Architecture        Version                         Repository             Size
================================================================================================================
Installing:
 git                            x86_64              2.31.1-2.el8                    appstream             160 k
Installing dependencies:
 git-core                       x86_64              2.31.1-2.el8                    appstream             4.7 M
 git-core-doc                   noarch              2.31.1-2.el8                    appstream             2.6 M
 perl-Error                     noarch              1:0.17025-2.el8                 appstream              45 k
 perl-Git                       noarch              2.31.1-2.el8                    appstream              77 k
 perl-TermReadKey               x86_64              2.37-7.el8                      appstream              39 k

Transaction Summary
================================================================================================================
Install  6 Packages

Total download size: 7.7 M
Installed size: 38 M
Is this ok [y/N]:
```

Downloading packages



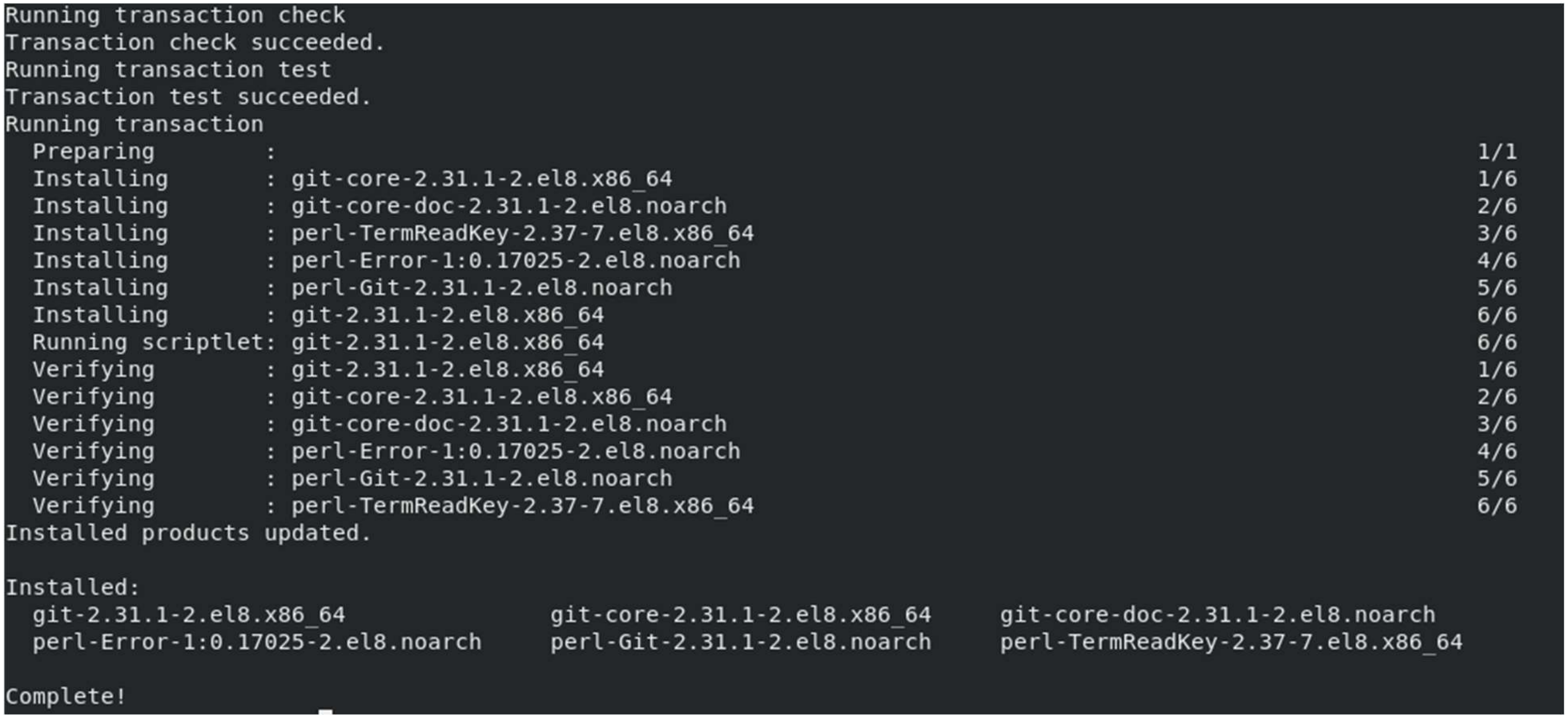
```
Is this ok [y/N]: y
Downloading Packages:
(1/6): git-2.31.1-2.el8.x86_64.rpm                                        150 kB/s | 160 kB     00:01
(2/6): perl-Error-0.17025-2.el8.noarch.rpm                                140 kB/s |  45 kB     00:00
(3/6): perl-Git-2.31.1-2.el8.noarch.rpm                                   199 kB/s |  77 kB     00:00
(4/6): perl-TermReadKey-2.37-7.el8.x86_64.rpm                             170 kB/s |  39 kB     00:00
(5/6): git-core-doc-2.31.1-2.el8.noarch.rpm                               543 kB/s | 2.6 MB     00:04
(6/6): git-core-2.31.1-2.el8.x86_64.rpm                                   828 kB/s | 4.7 MB     00:05
---------------------------------------------------------------------------------------------------------
Total                                                                     1.1 MB/s | 7.7 MB     00:06
```

Sometime shows GPG key warning, just type “y”



```
warning: /var/cache/dnf/appstream-62ae9a0bbea44fbe/packages/git-2.31.1-2.el8.x86_64.rpm: Header V4 RSA/SHA256 Signature, key I
D 6d745a60: NOKEY
Rocky Linux 8 - AppStream                                                    440 kB/s | 1.6 kB     00:00
Importing GPG key 0x6D745A60:
 Userid     : "Release Engineering <infrastructure@rockylinux.org>"
 Fingerprint: 7051 C470 A929 F454 CEBE 37B7 15AF 5DAC 6D74 5A60
 From       : /etc/pki/rpm-gpg/RPM-GPG-KEY-rockyofficial
Is this ok [y/N]: y
```

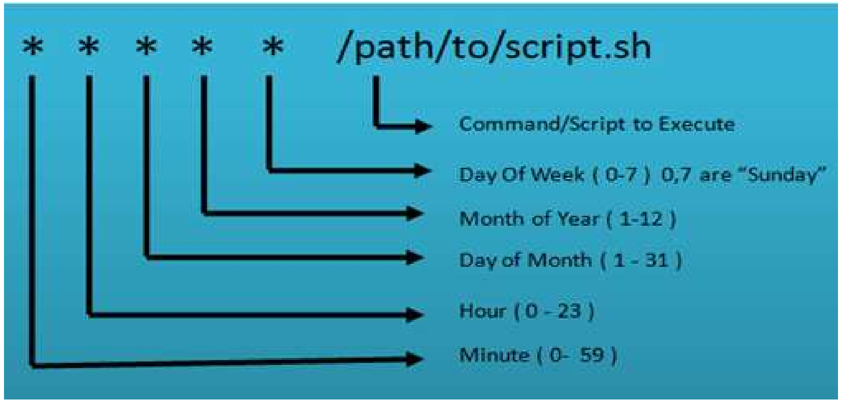
Installing packages



```
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                                                        1/1
  Installing       : git-core-2.31.1-2.el8.x86_64                                                           1/6
  Installing       : git-core-doc-2.31.1-2.el8.noarch                                                       2/6
  Installing       : perl-TermReadKey-2.37-7.el8.x86_64                                                     3/6
  Installing       : perl-Error-1:0.17025-2.el8.noarch                                                      4/6
  Installing       : perl-Git-2.31.1-2.el8.noarch                                                           5/6
  Installing       : git-2.31.1-2.el8.x86_64                                                                6/6
  Running scriptlet: git-2.31.1-2.el8.x86_64                                                                6/6
  Verifying        : git-2.31.1-2.el8.x86_64                                                                1/6
  Verifying        : git-core-2.31.1-2.el8.x86_64                                                           2/6
  Verifying        : git-core-doc-2.31.1-2.el8.noarch                                                       3/6
  Verifying        : perl-Error-1:0.17025-2.el8.noarch                                                      4/6
  Verifying        : perl-Git-2.31.1-2.el8.noarch                                                           5/6
  Verifying        : perl-TermReadKey-2.37-7.el8.x86_64                                                     6/6
Installed products updated.

Installed:
  git-2.31.1-2.el8.x86_64               git-core-2.31.1-2.el8.x86_64     git-core-doc-2.31.1-2.el8.noarch
  perl-Error-1:0.17025-2.el8.noarch     perl-Git-2.31.1-2.el8.noarch     perl-TermReadKey-2.37-7.el8.x86_64

Complete!
```

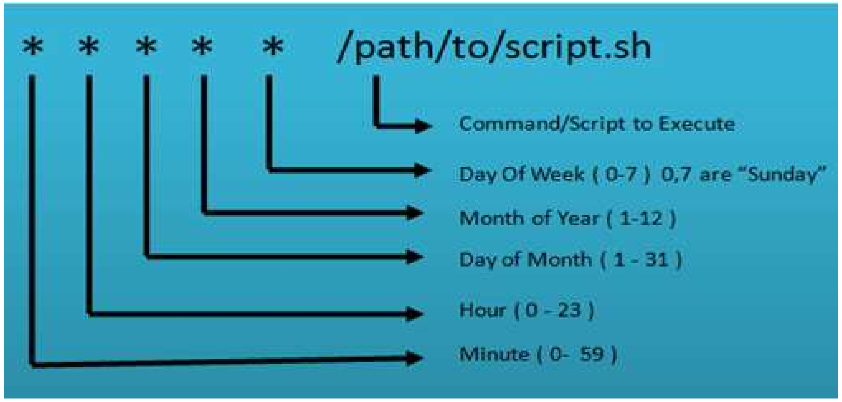
Task Scheduling (crontab)

crontab -l	list cron
crontab -e	modify cron list


* * * * * /path/to/script.sh
Command/Script to Execute
Day Of Week (0-7) 0,7 are "Sunday"
Month of Year (1-12)
Day of Month (1 - 31)
Hour (0 - 23)
Minute (0- 59)

Useful website: https://crontab.guru/

crontab


* * * * * /path/to/script.sh
Command/Script to Execute
Day Of Week (0-7) 0,7 are "Sunday"
Month of Year (1-12)
Day of Month (1 - 31)
Hour (0 - 23)
Minute (0- 59)

`* * * * *`	script runs every minute
`0 * * * *`	script runs every hour at minute 0
`0 6 * * *`	script runs every day at 06:00
`0 12 1 * *`	script runs every 1st day of the month at 12:00
`*/10 * * * *`	script runs every 10 minutes
`30 5 * * 0`	script runs every Sunday at 05:30

The time is based on system clock

Useful website: https://crontab.guru/

Write the cron

```
$ crontab -e
```

Editing cron with Vim

```
* * * * * command
* * * * * /path/to/script.sh
```

Linux Shell

bash	Bash shell (by default in many linux distribution)
csh	C shell (the commands look like C programming language)
zsh	Z shell
fish	Fish shell (modern shell with many additional features)

Create Shell Script

```
$ vim/nano begin.sh
```

```
#!/bin/bash

echo “hello world”
```

save file and quit

```
$ chmod +x begin.sh
$ ./begin.sh
```

Shell Script

Enable us to write linux commands to run specific objective.

Example: download files, put in to a folder, backup the folder, send the backup file to another machine

It has feature like common programming language i.e, variable, loop, conditional

Let’s Practice with Sample Script

`gfs_download.sh` and `gfs_filter_download.sh`

Another Useful Linux Feature

`sed`

`awk`

Shell as programming language

End of Day 2

Good luck for the next trainings!