FUC Documentation

Release 0.1.0

Pingjun Chen

Dec 27, 2022

COMMANDS

CHAPTER

ONE

COMMANDS BASICS

1.1 Help commands

`$ man <command>` # man page

`$ whatis <command>` # whatis

`$ <command> -h` # option -h

1.2 Environment variables

Create variable in shell

`$ <env_var>="var_value" $ export <env_var>`

Define vairable in bash configuration file

`export <env_var>="var_value"` # add the setting to ~/.bashrc

`source ~/.bashrc` # load new environment variables to shell

Print variable

`$ echo $<env_var> $ printenv <env_var>`

1.3 History commands

`$ history` # check recent commands

`$ history | grep <string>` # find previous command with specific string

1.4 Directory change

`$ pwd` # check current directory

`$ cd ~` # go to home directory

`$ cd -` # go to previous directory

`$ cd ..` # go to parent directory

1.5 Download from the internet

```
$ wget <url>
$ curl <url>
```

CHAPTER

TWO

GIT COMMANDS

2.1 Delete branch

Delete local branch

```
$ git branch -D <branch_name>
```

Delete remote branch

```
$ git push <remote_name> --delete <branch_name>
```

2.2 Tag

Add tag to current submit

```
$ git tag -a <tag_name> -m <comment>
```

List all tags

```
$ git tag -l
```

Delete local tag

```
$ git tag -d <tag_name>
```

Delete remote tag

```
$ git push --delete origin <tag_name>
```

2.3 Remote

Set up upstream remote

```
git remote add upstream https://github.com/repo
```

Update local repo

```
git pull upstream <branch_name>
```

Push to upstream

```
git push upstream <branch_name>
```

CHAPTER

THREE

FILE OPERATIONS

3.1 Find files contain specifc text

```
$ grep -rin <text> .
```

3.2 Check specifc lines inside a text file

```
$ sed -n <start_line_num>:<end_line_num>p <text_file>
```

3.3 Get number of all files

When number of file is small, less than 10, 000

```
$ ls -ls *.<ext> | wc -l
```

When more than 10, 000 files

```
$ find -type f -name '*.<ext>' | wc -l
```

3.4 Storage usage

Check disk partition usage

```
$ df -h
```

Check the size of a directory

```
$ du -hs <directory>
```

3.5 Check size of all subdirectories

```
$ du -hs *
```

3.6 Searches directory recursively in subdirectories

```
$ find . -type d -name <directory>
```

3.7 Searches file recursively in subdirectories

```
$ find . -type f -name <file>
```

3.8 Delete file/directory recursively in subdirectories

```
$ find . -name <file/directory> -exec rm -rf {} ;
```

CHAPTER

FOUR

DOCKER MANAGEMENT

Docker is an open platform for developers and sysadmins to build, ship, and run distributed applications, whether on laptops, data center VMs, or the cloud.

4.1 Docker Installation

```
$ sudo apt install docker.io        # install docker
$ sudo usermod -aG docker $USER     # add user to docker group
```

4.2 Image management

Create **image** from Dockerfile

```
$ docker build -t <image_name>:[tag_name] .
```

Create **image** from a specific Dockerfile

```
$ docker build -f <dockerfile_name> -t <image_name>:[tag_name] .
```

Image management

```
$ docker save <image_name>:[tag_name] <image_name>.tar      # save image
$ docker load --input <image_name>.tar                      # load image
$ docker images                                             # list available images
$ docker rmi <image_name>                                   # remove image by name
$ docker rmi <image_id>                                     # remove image by id
$ docker images purge                                       # remove dangling images
```

4.3 Container management

Start container

```
$ docker run -it --restart always --name <container_name> <image_name>:[tag_name]
Options:
    -v <local_dir>:<docker_dir>:ro          # map local directory to docker
    --runtime=nvidia                        # expose NVIDIA GPUs
    -e NVIDIA_VISIBLE_DEVICES=5             # cuda device setting
```

(continues on next page)

(continued from previous page)

```
    --shm-size 16G                                  # set size of shared memory
    --rm                                            # remove container file system when exits
```

Copy files

```
$ docker cp <container_name>:<src_dir> <local_dst_dir>  # copy files from docker to local
$ docker cp <local_src_dir> <container_name>:<dst_dir>  # copy files from local to docker
```

Container management

```
$ docker ps                          # list all available containers
$ docker stop <container_name>       # stop specific container
$ docker rm <container_name>         # remove specific stopped container
```

4.4 Upload image to docker hub

Login to docker hub

```
$ export DOCKER_ID_USER="user_name"   # set docker hub username
$ docker login                        # login in to docker hub
```

Tag image

```
$ docker tag <image_name>:<version> $DOCKER_ID_USER/<image_name>:<version>
```

Push to docker cloud

```
$ docker push $DOCKER_ID_USER/<image_name>:<version>
```

4.5 Convert image to singularity

Create docker image tarball

```
$ docker save <image_name>:<version> -o <image_name>.tar
```

Build singularity from image tarball

```
$ singularity build <image_name>.sif docker-archive://<image_name>.tar
```

Build singularity from DockerHub image

```
$ singularity pull <image_name>.sif docker://<user_name>/<image_name>:<version>
```

CHAPTER

FIVE

USER MANAGEMENT

5.1 Add user with full profile

```
$ adduser <username>
```

5.2 Add user with only name

```
$ useradd <username>
```

5.3 Add user to sudo group

```
$ usermod -aG sudo <username>
```

5.4 List all user names

```
$ compgen -u
```

Delete user

```
$ userdel -r <username>
```

CHAPTER

SIX

COMPRESSION OPERATIONS

6.1 Tar and untar

Create tar archive File

```
$ tar -cvf tar-archive-name.tar <source>
```

Untar files in current directory

```
$ tar -xvf tar-archive-name.tar
```

Untar files to a specific directory

```
$ tar -xvf tar-archive-name.tar -C <destination>
```

List content of tar archive file

```
$ tar -tvf tar-archive-name.tar
```

6.2 Compress and uncompress

Compress folder to tar.gz

```
$ tar -czvf tar-archive-name.tar.gz <source>
```

Extract a tar.gz compressed archive

```
$ tar -xzvf tar-archive-name.tar.gz
```

Compress folder to tar.bz2

```
$ tar -cjvf tar-archive-name.tar.bz2 <source>
```

Extract a tar.bz2 compressed archive

```
$ tar -xjvf tar-archive-name.tar.bz2
```

CHAPTER

SEVEN

PROCESS MANAGEMENT

7.1 Check process

View your system's resource usage and see the processes that are taking up the most system resources.

```
$ top
```

Improved top

```
$ htop
```

7.2 Kill process

By process id

```
$ kill -9 <pid>
```

By application name

```
$pkill <app>
```

By filtering conditions

```
ps -ef | grep <command> | awk '{print $<col_num>}' | xargs kill -9
```

7.3 Check Ubuntu version

```
$ lsb_release -a
```

CHAPTER

EIGHT

SEADRAGON

Seadragon is a supercomputing resource of MD Anderson's High Performance Computing (HPC), which is dedicated for use by the research community.

8.1 Login Seadragon

```
$ ssh seadragon
```

8.2 Interactive Running

```
# cpu interactive
$ busb -Is -q interactive -W 1:00 -M 64 -R rusage[mem=64] -n 4 /bin/bash
# gpu interactive
$ busb -Is -q gpu-medium -gpu num=1:gmem=16 -W 3:00 -M 64 -R rusage[mem=64] -n 10 /bin/
↪bash
```

8.3 Load Modules

```
$ module load singularity/3.5.2
$ module load cuda10.1/toolkit/10.1.243
```

8.4 Start Singularity Instance

```
$ singularity run --nv --bind <local_dir>:<container_dir> <singularity_path> <program>
```

8.5 More Useful Commands

```
$ bsub < <lsf_script> # submit job via script
$ bjobs -u all | more # check all running jobs
$ bjobs -u all | grep gpu # check all gpu jobs
$ bjobs -p # show all pending jobs
$ bjobs -l xxxxxxx # check the details of one specific job
$ bkill -l xxxxxxx # kill one specific job
```

CHAPTER

NINE

CONDA ENV

9.1 List available envs

```
$ conda info --envs
```

9.2 Create env

```
$ conda create --name <env_name> python=3.6
```

9.3 Activate env

```
$ source activate <env_name>
```

9.4 Deactivate env

```
$ source deactivate
```

9.5 Remove env

```
$ conda remove --name <env_name> --all
```

CHAPTER

TEN

VIRTUAL ENV

10.1 Installation

```
$ pip install virtualenv
```

10.2 Create virtual environment

Create with specific python version

```
$ virtualenv -p python3 myvenv
```

10.3 Activate environment

```
$ source myvenv/bin/activate
```

10.4 Deactivate the environment

```
$ deactivate
```

CHAPTER

ELEVEN

TMUX

tmux is a terminal multiplexer: it enables a number of terminals to be created, accessed, and controlled from a single screen. tmux may be detached from a screen and continue running in the background, then later reattached.

11.1 Installation

```
$ sudo apt-get install tmux
```

11.2 List available sessions

```
$ tmux ls
```

11.3 Start new session

```
$ tmux new -s <session_name>
```

11.4 Attach to session

```
$ tmux a -t <session_name>
```

11.5 Kill session

```
$ tmux kill-session -t <session_name>
```

CHAPTER

TWELVE

SFTP

SSH File Transfer Protocol, a network protocol used for secure file transfer over secure shell.

12.1 Login to server

Login to user home directory of the server.

```
$ sftp <user_name>@<server_ip>
```

Login to specific directory of the server.

```
$ sftp <user_name>@<server_ip>:<remote_dir>
```

12.2 File transfer

Download file from server.

```
$ get <server_path> <local_path>
```

Upload local file to server.

```
$ get <local_path> <server_path>
```

12.3 Command

Command for local host.

```
$ !<command>
```

Command for server.

```
$ <command>
```

12.4 Exit server

Exit from server.

```
$ bye
```

CHAPTER

THIRTEEN

APACHE

13.1 Restart

```
$ service apache2 restart
```

13.2 Check error log

```
$ tail -n 20 /var/log/apache2/error.log
```

CHAPTER

FOURTEEN

ABOUT FUC-CHEN

Frequently Used Commands for CHEN (FUC-CHEN) is a collection of most FUCs used in daily coding. Please consider *star* this repo if it helps you.