A Quick Intro to lxplus, bash & python

Starterkit 2022
28th November
Richard Williams & Dan Thompson
What is lxplus?

- **lxplus** is CERN’s interactive linux service for all CERN users
- It consists of ~100 of virtual machines each with 30GB RAM & 8 cpus, equipped with fairshare systems to ensure everyone can stay online!
- By logging in to **USERNAME@lxplus.cern.ch** the system will connect you to the machine **lxplus7NNN** with the most resources available.
- **Let’s log in!**

More Detailed Instructions in [Pre-Workshop Checklist](#)
What is lxplus?

- **lxplus** is CERN’s interactive linux service for all CERN users
- It consists of ~100 of virtual machines each with 30GB RAM & 8 cpus, equipped with fairshare systems to ensure everyone can stay online!
- By logging in to **USERNAME@lxplus.cern.ch** the system will connect you to the machine **lxplusNNN** with the most resources available.
- Let’s log in!

More Detailed Instructions in [Pre-Workshop Checklist](#)
What is lxplus?

- **lxplus** is CERN’s interactive linux service for all CERN users
- It consists of ~100 of virtual machines each with 30GB RAM & 8 cpus, equipped with fairshare systems to ensure everyone can stay online!
- By logging in to `USERNAME@lxplus.cern.ch` the system will connect you to the machine `lxplusNNN` with the most resources available.
- **Let’s log in!**

More Detailed Instructions in [Pre-Workshop Checklist](#)
What is lxplus?

- **lxplus** is CERN’s interactive linux service for all CERN users
- It consists of ~100 of virtual machines each with 30GB RAM & 8 cpus, equipped with fairshare systems to ensure everyone can stay online!
- By logging in to `USERNAME@lxplus.cern.ch` the system will connect you to the machine `lxplusNNN` with the most resources available.
- Let’s log in!

More Detailed Instructions in [Pre-Workshop Checklist](#)
Ixplus Tips and File Storage

- You can increase your `/afs/cern.ch/user/u/username/` disk space if you need to.
- In LHCb we also have access to an Ixplus work directory: `/afs/cern.ch/work/u/username/` with more space available.

- **CERN EOS**: You have two long term storage areas, [more info here](https://resources.web.cern.ch/)
  - `/eos/user/u/username/` Your private EOS space (also visible on [CERNBox](https://www.cernbox.ch/))
  - `/eos/lhcb/user/u/username/` For storing tuples
Bash Introduction

- **pwd**: Shows the current working directory path
- **ls**: List files/directories
  - `ls -a`: For hidden locations
  - `ls -l`: For extra details
  - `ls path/to/dir`: To peek at a specific directory
    - Can combine arguments (-alh etc.)
- **cd dir**: Move about directories
  - `cd ../`: To move up a directory
- **touch filename**: Create a new file
- **mkdir dir**: Create directory
- **cp source destination**: Copy
- **mv source destination**: Move
- **rm file**: Remove a file
  - `rm -d` for removing a directory
  - `rm -r` recursive, remove directory and its contents
Bash Introduction

- **pwd**: Shows the current working directory path
- **ls**: List files/directories
  - `ls -a`: For hidden locations
  - `ls -l`: For extra details
  - `ls path/to/dir`: to peek at a specific directory
  - Can combine arguments (`-alh` etc.)
- **cd dir**: Move about directories
  - `cd ../`: To move up a directory
- **touch filename**: Create a new file
- **mkdir dir**: Create directory
- **cp source destination**: Copy
- **mv source destination**: Move
- **rm file**: Remove a file
  - `rm -d` for removing a directory
  - `rm -r` recursive, remove directory and its contents

Use tab for autocomplete; Super useful! Try double tapping tab?

```
[dathomps@lxplus738 Starterkit22]$ touch testFile2.py
[dathomps@lxplus738 Starterkit22]$ ls -l
   total 4
-rw-r--r-- 1 dathomps z5 0 Nov 20 23:19 testFile2.py
-rw-r--r-- 1 dathomps z5 0 Nov 20 23:04 testFile.txt
drwxr-xr-x 2 dathomps z5 2048 Nov 20 23:04 testLesson
drwxr-xr-x 2 dathomps z5 2048 Nov 20 23:04 testLesson
   total 6
-rw-r--r-- 1 dathomps z5 0 Nov 20 23:19 testFile2.py
drwxr-xr-x 2 dathomps z5 2048 Nov 20 23:19 testFiles
-rw-r--r-- 1 dathomps z5 0 Nov 20 23:04 testFile.txt
drwxr-xr-x 2 dathomps z5 2048 Nov 20 23:04 testLesson
drwxr-xr-x 2 dathomps z5 2048 Nov 20 23:04 testLesson
   total 6
-rw-r--r-- 1 dathomps z5 0 Nov 20 23:19 testFile2.py
drwxr-xr-x 2 dathomps z5 2048 Nov 20 23:19 testFiles
drwxr-xr-x 2 dathomps z5 2048 Nov 20 23:04 testLesson
drwxr-xr-x 2 dathomps z5 2048 Nov 20 23:04 testLesson
[dathomps@lxplus738 Starterkit22]$ cp testScript.sh testFiles/.
[dathomps@lxplus738 Starterkit22]$
[dathomps@lxplus738 Starterkit22]$ ls testFiles
   testScript.sh
```

. Points to the “current” directory
Bash Introduction

- **pwd**: Shows the current working directory path
- **ls**: List files/directories
  - `ls -a`: For hidden locations
  - `ls -l`: For extra details
  - `ls path/to/dir`: to peek at a specific directory
  - Can combine arguments (-alh etc.)
- **cd dir**: Move about directories
  - `cd ../`: To move up a directory
- **touch filename**: create a new file
- **mkdir dir**: Create directory
- **cp source destination**: copy
- **mv source destination**: move
- **rm file**: remove a file
  - `rm -d` for removing a directory
  - `rm -r` recursive, remove directory and its contents

Wildcards * : the shell will fill these with any character(s)
Editing Files! A million ways available, choose your fave!

There are numerous ways to edit text files when connected via ssh, some lxlplus options:

- **Terminal Text Editors**: `vim`, `nano` + more - [Vim Guide](#), [nano Guide](#)
- **X-Forwarding Text Editors**: `emacs`, `gedit` - Allows editing in external window
- **Local Text Editors**: Mount your file system with `sshfs` or a remote scp file explorer (I use [winscp](#))
- **IDEs**: Integrated Development Environments can combine your shell, file explorer and text editor into one program! [Vscode](#)

For now I will use `nano`:

```bash
$ nano testFile.txt : Creates a new file, now we can create anything!
```

**Exercise!** Create and save a `testFile.txt` using one of the above methods. Use `$ cat testFile.txt` to quickly check the content!
Extra Info!

- A description of most commands can be accessed using `command --help`

- For further information on bash check the documentation: [https://devdocs.io/bash/](https://devdocs.io/bash/)

- For a more extensive tutorial follow the HEP Software Foundation “Introducing the Shell”
Bash Scripts

- Let’s take a look at `lbConda_Starterkit_Create.sh`
- This is a shell script, allowing us to take the commands we use frequently and save them in a chain that we can run with a one line command!

```
$ nano lbConda_Starterkit_Create.sh
```

```
#!/bin/bash
source $HOME/.bashrc
shopt -s expand_aliases
(source /cvmfs/lhcb.cern.ch/lib/LbEnv
lb-conda-dev virtual-env default/2022-11-21
$1
$1/run pip install zfit==0.10.1
$1/run python -m ipykernel install --user --name=$1
) &> starterkitWrapperInstall.txt
```

- Links for More Info!
  - cvmfs
  - LbEnv
  - lb-conda
  - pip
  - ipykernel
Preparing for Today’s Lesson!

- Let’s use these skills to prepare for today’s Python lesson.
- First we need to download the repository
  - `$ mkdir Starterkit2022; cd Starterkit2022`
- Take a look around, there are a number of notebooks, root files and csv’s!

- Now return to `/path/to/user/Starterkit2022/starterkit-2022-python-and-bash-tutorial/` And then...
  - `$ bash lbConda_Starterkit_Create.sh starterkitEnv &`

- What did this script do? Let’s investigate!
  - (Spoiler alert, it’s installing a python environment ready for the week ahead!)
- Let’s take a look at `lbConda_Starterkit_Create.sh`
- This is a shell script, allowing us to take the commands we use frequently and save them in a chain that we can run with a one line command!

```bash
$ nano lbConda_Starterkit_Create.sh
```

- `/bin/bash` — Tells the terminal to run the script in the bash shell
- `source $HOME/.bashrc` — Access our home environment + aliases
- `shopt -s expand_aliases` —
- `source /cvmfs/lhcb.cern.ch/lib/LbEnv` — Access LHCb Commands
- `lb-conda-dev virtual-env default/2022-11-21 $1` — install a new python env
- `$1/run pip install zfit==0.10.1` — install zfit (for tomorrow!)
- `Install python -m ipykernel install --user --name=$1` — install a kernel
- `starterkitWrapperInstall.txt` — Write the output to a .txt file

Let’s create our own!

Links for More Info!
- `cvmfs`
- `LbEnv`
- `lb-conda`
- `pip`
- `ipykernel`
Creating our own!

```bash
#!/bin/bash

echo "hello world" # Prints the statement to the terminal

echo "We are storing ${1} in 'myVariable'" # $1 (or $2, 3, 4 etc...) refers to the 1st, 2nd, 3rd, 4th arguments of the script

export myVariable=${1} # Creating a new environment variable

# $0 tells bash to look for the variable "name"
echo "myVariable is ${myVariable}" # Now use it!
```

What happens if you run: $ bash testScript.sh

How about: $ bash testScript.sh something

Try: $ bash testScript.sh something else

Maybe: $ bash testScript.sh "something else"

Exercise!

Have a go playing around with this script to get some practice!

Idea: What does $0 do?
Environment Variables & .bashrc

What did `export myVariable=$1` do? It made a new variable **within that script.**

Try `$ echo $myVariable` in your terminal...

We can use bash scripts to create variables within other environments:

- Instead use `source`: `$ source testScript.sh something`
- Now: `$ echo $myVariable` this variable is available in the terminal!

This can be very useful when you need to frequently access the same filepaths, commands or phrases -> write the variables to a bash script and `source`!

**.bashrc:** Located in your `$HOME/.bashrc` is a very convenient file for storing your favourite environment variables and commands. Every time you login to lxplus `$ source .bashrc` runs automatically! e.g I have `export WORK=/afs/cern.ch/work/d/dathomps/` in mine!
Links to Further Techniques and tools, Endless Possibilities!

- So much more to teach, so little time! More on HEP Software Foundation:
  - [Introducing the Shell](#)
  - [More about the UNIX Shell](#)
- Want to keep a session running? [Tmux/Screen guide](#), [tmux cheat sheet](#)
- [lxplus HTCondor](#), when your shell script needs more power!
- [pypi](#) the huge ecosystem of python modules
  - [lb-conda wrappers](#), an easy way to install and use new packages… more about this now!
How to Run Python!

- Your python environment installation should have completed now.
- Now we can begin coding with python!
- In the directory `starterkit-2022-python-and-bash-tutorial` take a look at `PythonExample.py` and try running it with

```bash
$ starterkitEnv/run python PythonExample.py
```

This launches the environment we created and runs python!

Alternatively we can launch a **new bash session** in the environment by

```bash
$ starterkitEnv/run bash and then simply $ python PythonExample.py
```

Try editing, saving and re-running this python file.
Jupyter - python in a Notebook!

- For this python session + zFit tomorrow we will need jupyter notebooks
- Jupyter provides a web based interactive python session which combines developing, documenting, and executing code into one package!

From your starterkitEnv bash session ($ starterkitEnv/run bash) run:

$ jupyter notebook --port=NNNN --no-browser

where NNNN is the port number you entered when logging in to lxplus!

Follow or copy the link that produced into your browser and an interactive web-based application should appear!

(Note: Jupyter will root itself to the directory you launched it from)
Summary

Thanks for listening to this very quick intro! Feel free to ask any questions throughout the week, no question is a bad question!

Links to more in-depth tutorials:

- [HSF Shell/Bash Tutorial](#)
- [HSF Introduction To Python](#)
- [bash, python cheat-sheets](#)

Remember, the internet is your friend, someone has always had the same issue/question!

A note for the future: Style and Documentation! [style-guides](#) exist to help you make software that is easy to understand for others and probably even you!
Preparing for Today’s Lesson! (copy from afs)

- Let’s use these skills to prepare for today’s Python lesson.
- First we need to download the repository
  - $ mkdir Starterkit2022; cd Starterkit2022
- Take a look around, there are a number of notebooks, root files and csv’s!

- Now return to /path/to/user/Starterkit2022/starterkit-2022-python-and-bash-tutorial/ And then...
  - $ ./lbConda_Starterkit_Create.sh starterkitEnv &

- What did this script do? Let’s investigate!
  - (Spoiler alert, it’s installing a python environment ready for the week ahead!)
Preparing for Today’s Lesson! (krb5)

- Let’s use these skills to prepare for today’s Python lesson.
- First we need to download the repository
  - $ mkdir Starterkit2022; cd Starterkit2022
- Take a look around, there are a number of notebooks, root files and csv’s!

- Now return to /path/to/user/Starterkit2022/starterkit-2022-python-and-bash-tutorial/ And then...
  - $ ./lbConda_Starterkit_Create.sh starterkitEnv &

- What did this script do? Let’s investigate!
  - (Spoiler alert, it’s installing a python environment ready for the week ahead!)