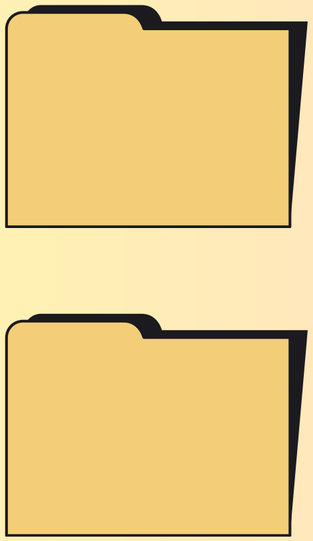
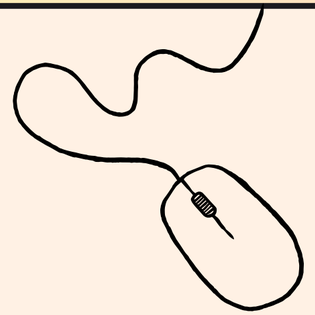


How to survive SSH and Bash

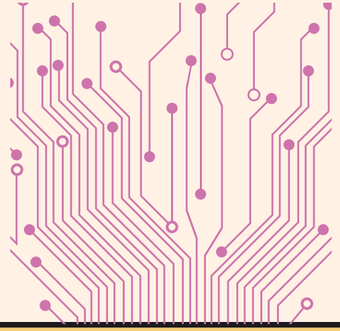
(without wanting to bash your head off a wall)

Todays Guide

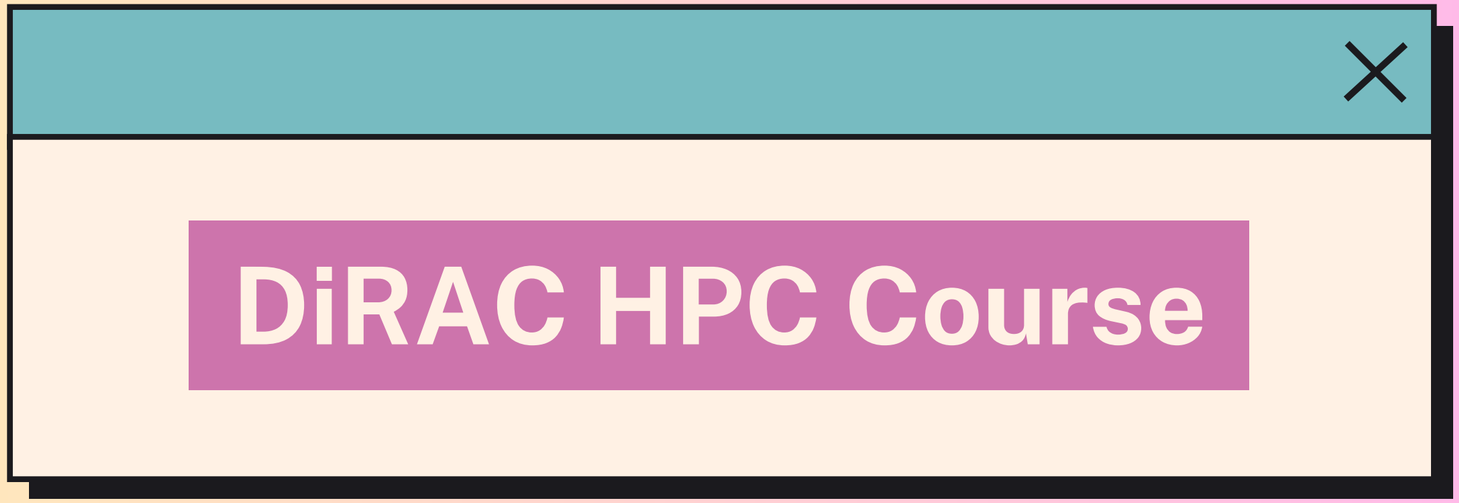


Intro to Bash

What is SSH



SSH Keys



https://dirac.ac.uk/foundation-hpc-skills-course/

SSH Access

getting on to remote machines

Connecting to HPC

We connect to HPC systems through command-line interfaces (CLI)

The shell is a program that runs other programs

Most popular Unix shell is Bash

Interact with the shell via the CLI

CLI - typing based interface, GUI - graphical based interface

Why Use SSH?

To connect a local machine to a remote system we use SSH

SSH - Secure SHell

SSH gives a secure way to access a computer over an unsecured network
Also provides password and public key authentication

SSH keys can enable single sign on (SSO) and allow file movement and transfers

SSH secure tunnels can be used for forwarding and encrypts actions through it
SCP - secure copy protocols

Terminals & Prompts

We connect using SSH through a terminal or prompt

Linux & Mac - Terminal
Windows - command prompt (but no SSH so needs to be installed)

Terminals and prompts are the CLI and allow us to interact with Bash

SSH keys

SSH keys are an alternative to using a password for authentication when logging on to remote systems, transferring files, or doing version control

SSH keys are by default stored in ~/.ssh/

You can check them using ls ~/.ssh/

When creating keys, you make a pair: a private one that is stored on your own machine and a public key which is stored on the remote machine

Creating SSH Keys

ssh-keygen

-t : type (default is RSA, can also use EdDSA ed25519)
-f : filename (use this if you don’t want to overwrite your default)
-o : OpenSSH key format
-a : number of rounds of passphrase derivation
-b : number of bits in the key

-p : change passphrase
-e : ‘export’ - reformat existing key

SSH Key Security

Passwords: be sensible

Private key: always only yours, NEVER share

2FA: may be utilised alongside some login methods

Logging On

ssh user@remotehost.web

plus any command line options

e.g. ssh user@login8.cosma.dur.ac.uk

SSH Shortcuts

SSH config:

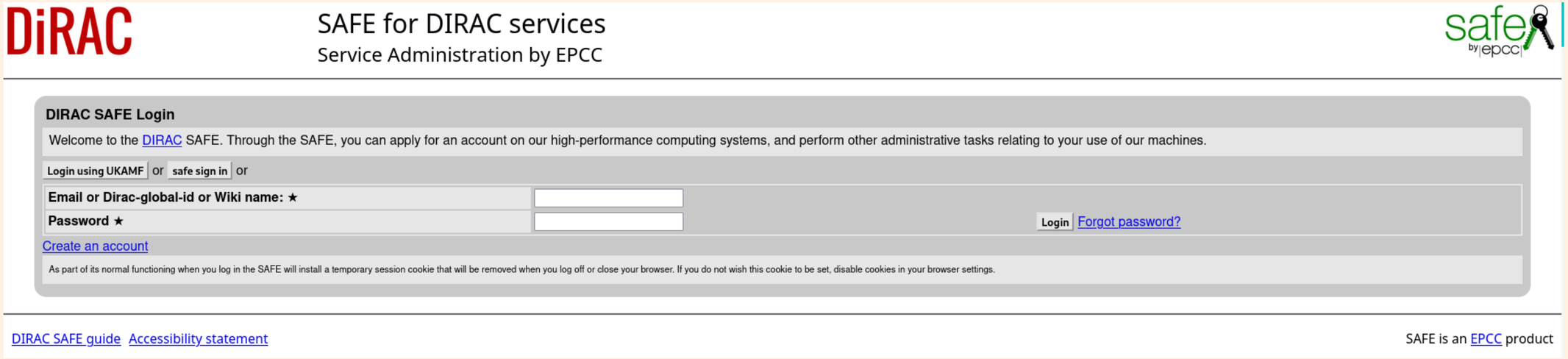
cd ~/.ssh
nano config

Host example
    HostName codecuppa
    User sarah
    Port 10000
    IdentityFile /path/to/privatekey

now can login with ssh example

DiRAC SAFE - SSH for COSMA

https://safe.epcc.ed.ac.uk/dirac/



DiRAC SAFE - SSH for COSMA

https://epcced.github.io/safe-docs/

Login account details

Username	dc-john7
Machines	**Name** cosma
Status	Active
Creation Date	07-Oct-2022
Projects	• do009 - hpcicc-excalibur • dp004 - VIRGO
Default Budget	dp004
do009 resources	
dp004 resources	**Resource Pool** — **Remaining Budget** mi-cosma8-cpu — 44,178,046.2 CPUhs mi-cosma7-cpu — 13,314,448.3 CPUhs

Account credentials

#	Type	Credential	Identifier	Status
4	[redacted]			
5	[redacted]			

PhD student - supervised by Sownak - Carlton - Alastair

Add Credential | Delete Credential | Request Password Reset | View Login Account Password | Set Default Budget | Join Project | Change primary Project

Bash Basics

bish, bash, bosh

Some Useful Commands

whoami - your username
pwd - print working directory
mkdir - make directory/new folder

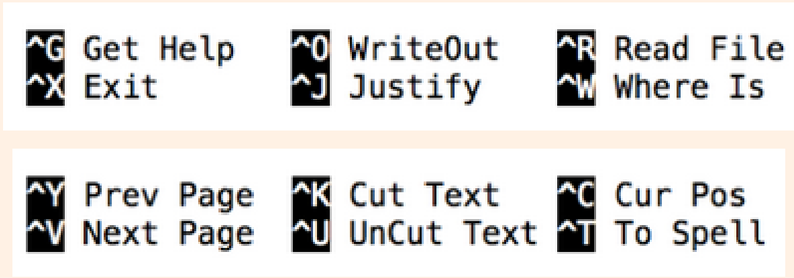
cd - change directory (cd ~ for home directory)
. - current directory
.. - parent directory

ls - print names of files (alphabetical, in columns)
ls -a : all files, including hidden files
ls -l: list in long format
ls -lh: list in format with human readable sizes
ls --help

Text Editors

vi, vim, emacs, nano

nano:
ctrl + o - save file
ctrl + x - exit editor
ctrl + k - delete a line or chunk of text
ctrl + u - paste



vim:
i - insert mode (typing)
esc - command mode
:q! - quit without saving
:wq - save and quit
:x - save and close
dd - delete a line
y - paste a line

File Handling

cat - read file (head, tail, less, can also use -f to get updates printed out)
mv - move file
cp - copy
rm - remove (use with caution) - can use rm - i (and set as alias)

wget/curl/aria2c/rsync - downloading files over HTTP(S)/FTP(S) connections

tar - extract files from .tar.gz

upload: scp local/destination user@remote.web:file
download: scp user@remote.web:file local/destination
overall: scp FROM TO

Other Useful Bash

*- wildcard - takes place of character to allow multiple operations

echo - prints back
> - pipes/redirects output e.g. into file
| - chain commands

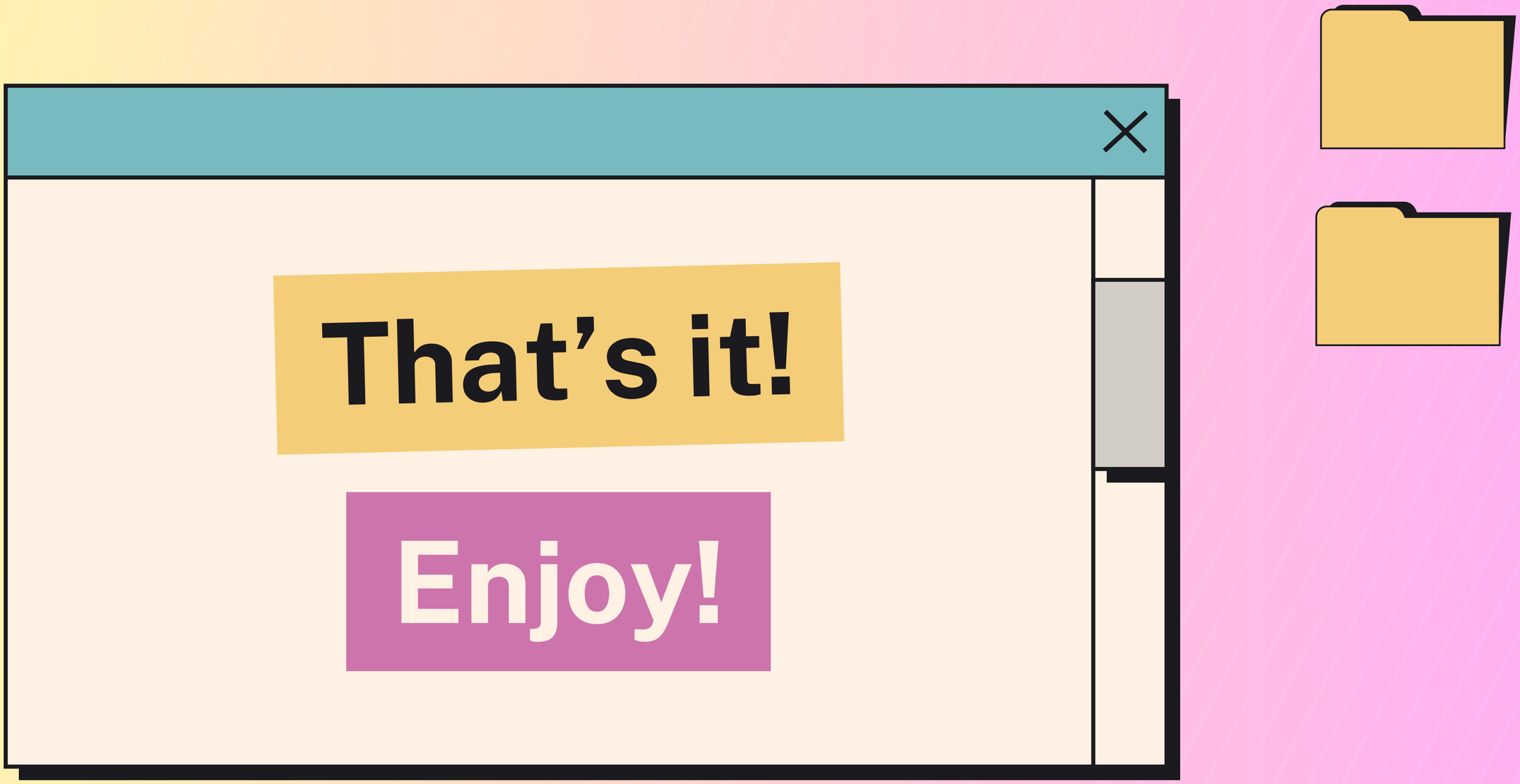
set variables using $VARNAME

Bash Scripts

.sh or .bash files

#!/bin/bash

Make sure to check your file permissions!
chmod +x filename


That’s it!
Enjoy!