

# IIT Dharwad

## Dept. of CSE

Remote log-in to a Linux machine - instructions

# Logging-in

- Logging into remote Linux system requires you to use SSH (“Secure Shell Protocol”)
  - Login credentials are encrypted
  - **SSH server** must be running on the system that you are logging into; Happens on most Linux systems by default.
  - **SSH client**, another piece of software, is used to authenticate and connect to the SSH server
    - Client software is available for all platforms (OSs)

# Logging-in Windows

- Powershell on Windows 10
  - Press (Windows + R ) -> Type “powershell”
  - Type `ssh <username>@<remotenode_IP_address>`
  - OR Type `ssh <username>@<remotenode_hostname>`
  - Type Yes when prompted (only first time)
  - Provide log-in credentials

 Windows PowerShell

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/powershell

PS C:\Users\ndheg> ssh nikhilh@10.250.101.100_
```

# Logging-in Windows

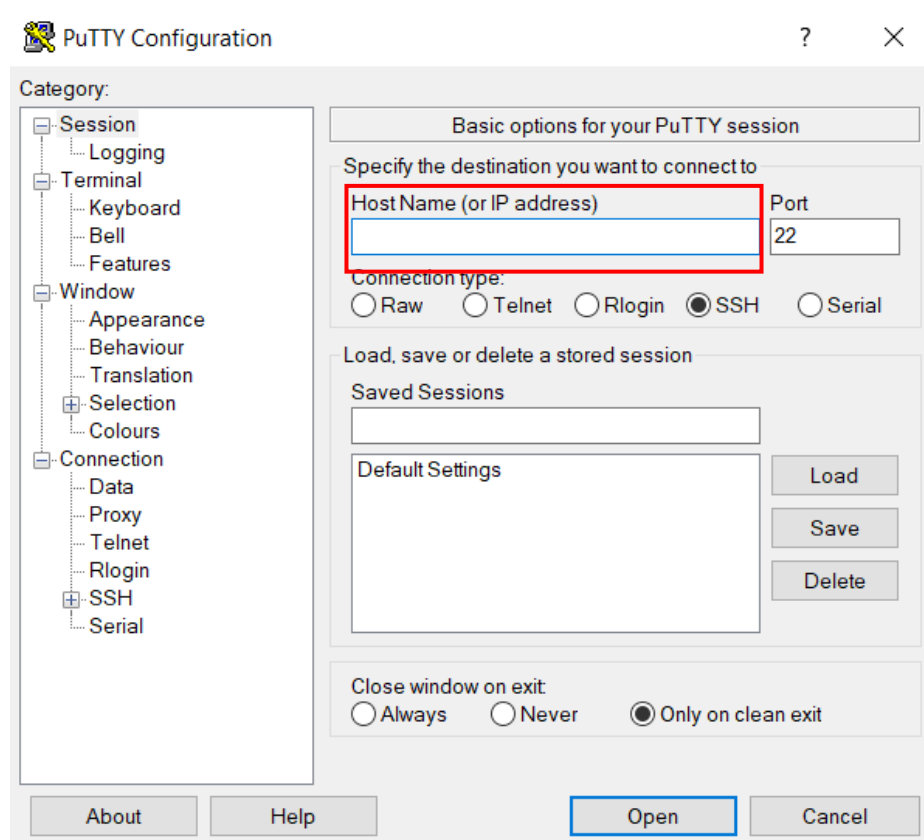
- PuTTY SSH client Windows
  - Download PuTTY from <https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html> (64-bit .exe)
  - Double click on the icon after downloading



putty

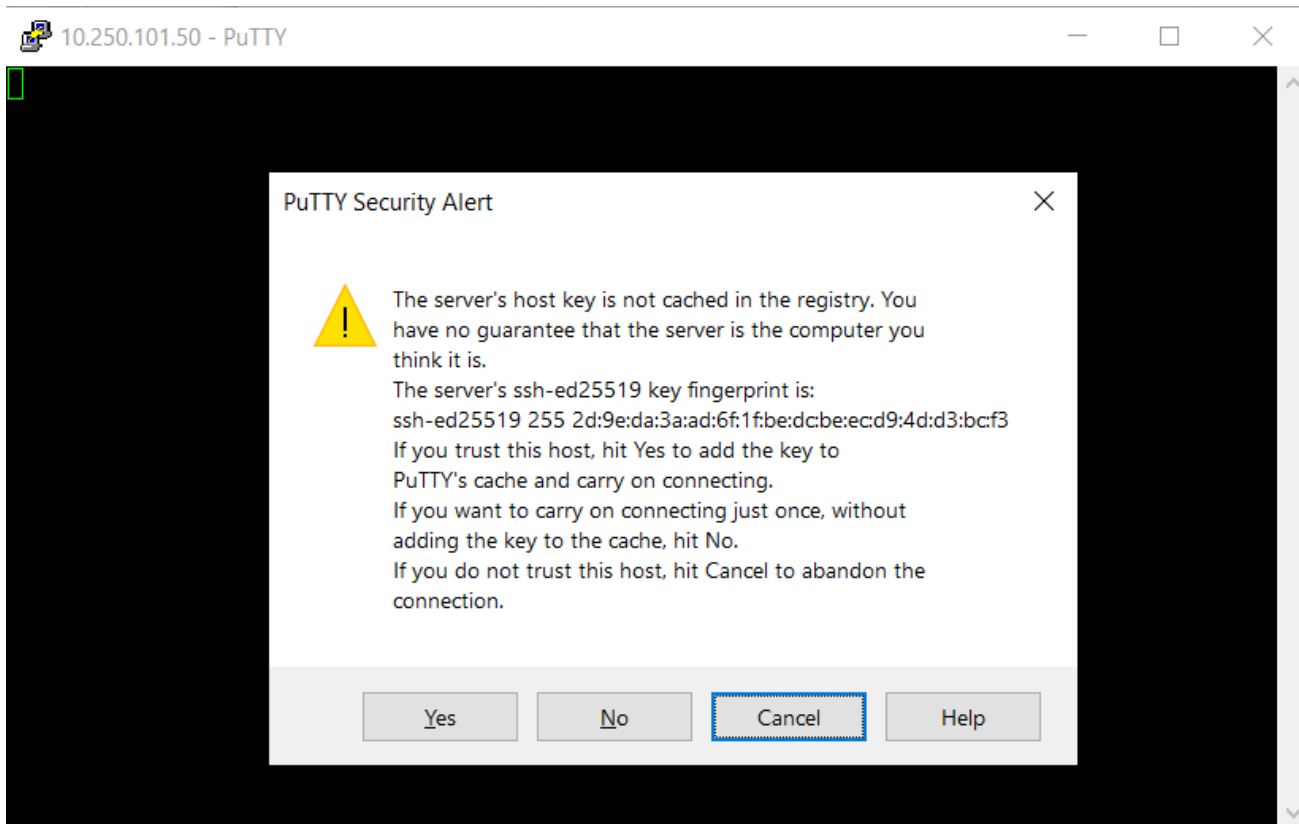
# Logging-in Windows

- Type in the Host Name / IP address and click 'Open'




# Logging-in Windows



- Click on 'Yes' (you are accepting the server host key)



# Logging-in Windows

- Enter log-in credentials

 nikhilh@iitdhmaster:~

```
 login as: nikhilh  
 nikhilh@10.250.101.100's password:  
Last login: Wed Mar 17 09:53:33 2021 from 10.196.7.237  
Intel(R) Parallel Studio XE 2020 Update 2 for Linux*  
Copyright 2009-2020 Intel Corporation.  
[nikhilh@iitdhmaster ~]$ █
```

# Logging-in MAC

- Open the 'Terminal' program on MAC (Go -> Applications -> Terminal)

```
Last login: Sun Mar  7 11:35:13 on ttys000
```

```
The default interactive shell is now zsh.
```

```
To update your account to use zsh, please run `chsh -s /bin/zsh`.
```

```
For more details, please visit https://support.apple.com/kb/HT208050.
```

```
apples-MacBook-Pro:~ apple$ █
```

- Type “ssh <username>@<masternode\_IP\_address>
- Type 'Yes' when prompted (only first time)
- Provide log-in credentials



# Logging-in Linux

- If you are a Linux user, you know what a 'Terminal' is 😊
- Type “ssh <username>@<remotenode\_IP\_address>
- Type 'Yes' when prompted (only first time)
- Provide log-in credentials

# After logging in...

- You have landed on the “command interpreter” / terminal.
  - The command interpreter is waiting for your commands..

# Useful Linux Commands

ls, ls -l

man

mkdir

cd

pwd

cp

mv

scp

rm //use with caution!

cat

less

head, tail

vi, vim, emacs, nano, pico

gzip, tar, zip

who

cut

wc

bc

echo

*Type “man <command\_name\_here>” on the Linux terminal to get help info*

# Useful Linux Commands - scp

scp - To move files back and forth between remote machine and your local system

scp file1 <user\_name>@<master\_node\_ip>

from your system to Master node



```
Windows PowerShell
PS C:\Temp> scp README.txt nikhilh@10.250.101.100:
nikhilh@10.250.101.100's password:
README.txt 100% 137 19.4KB/s 00:00
PS C:\Temp> scp nikhilh@10.250.101.100:pscp.exe .
nikhilh@10.250.101.100's password:
pscp.exe 100% 669KB 3.9MB/s 00:00
PS C:\Temp>
```

from Master node to your system

scp <user\_name>@<master\_node\_ip>:file1 .

*Note: if you have Windows-based systems, you can open Powershell and the scp command works. Otherwise, you can also download [pscp](#)*

# Useful Linux Commands – zip, unzip

zip, unzip - To compress/uncompress folders/directories

```
zip -r compressed.zip workshop_files/  
unzip compressed.zip
```

-r for recursively applying the compression to folders within

```
Windows PowerShell  
PS C:\Temp\Nikhil\Courses\Others\HPC> zip -r workshop_files.zip workshop_files  
updating: workshop_files/ (192 bytes security) (stored 0%)  
updating: workshop_files/HPC_101_1.pptx (172 bytes security) (deflated 3%)  
updating: workshop_files/README.txt (172 bytes security) (stored 0%)  
adding: workshop_files/test_combination.out (172 bytes security) (deflated 57%)  
adding: workshop_files/test_complex.out (172 bytes security) (deflated 75%)  
adding: workshop_files/test_expr.out (172 bytes security) (deflated 72%)  
adding: workshop_files/test_if.out (172 bytes security) (deflated 61%)  
adding: workshop_files/test_mult.out (172 bytes security) (deflated 53%)  
PS C:\Temp\Nikhil\Courses\Others\HPC> unzip workshop_files.zip  
Archive: workshop_files.zip  
replace workshop_files/HPC_101_1.pptx? [y]es, [n]o, [A]ll, [N]one, [r]ename: A  
  inflating: workshop_files/HPC_101_1.pptx  
  extracting: workshop_files/README.txt  
  inflating: workshop_files/test_combination.out  
  inflating: workshop_files/test_complex.out  
  inflating: workshop_files/test_expr.out  
  inflating: workshop_files/test_if.out  
  inflating: workshop_files/test_mult.out  
PS C:\Temp\Nikhil\Courses\Others\HPC>
```

# Useful Linux Commands – tar

tar – **T**ape **A**rchive to compress/uncompress folders/directories

```
tar -cvf workshop.tar workshop_files/  
tar -xvf workshop.tar
```

Type `man tar` to know about flags

tar followed by gzip compression:

```
tar -czvf workshop.tar.gz workshop_files/  
tar -xzvf workshop.tar.gz
```

# Useful Linux Commands - man

- type `man <command>` and hit Enter key to get help

```
➤ nikhilh@iitdhmaster:~
```

```
[nikhilh@iitdhmaster ~]$ man wc
```

- type `q` to quit. Use arrows to scroll

```
Ubuntu nikhilh@iitdhmaster:~
(cs406WC(1) User Comm

NAME
    wc - print newline, word, and byte counts for each file

SYNOPSIS
    wc [OPTION]... [FILE]...
    wc [OPTION]... --files0-from=F

DESCRIPTION
    Print newline, word, and byte counts for each FILE, and a total
    when FILE is -, read standard input. A word is a non-zero-le
    options below may be used to select which counts are pri
    byte maximum line length
```

# Useful Linux Commands

## Other utility commands

`cd HPC` → change directory to HPC

`cd ..` → change directory to parent

`vim hello.txt` → open a file `hello.txt` for editing. See vi commands

`ls` → list files in the current directory

`head hello.txt` → display the first few lines of `hello.txt`

`cat hello.txt` → display entire content of `hello.txt`

`pwd` → display the name of the present working directory

`who` → display the names of all users who have currently logged-in



# stdout, stdin, stderr in Linux

- stdout
  - Output that is printed to screen (terminal)
- stdin
  - Keyboard input
- stderr
  - Error messages printed to screen (terminal)

also called streams (input stream, output stream, error stream)

# redirects and pipes in Linux

//redirect standard output to file1.txt

- `echo "hello world" > file1.txt`

//feed input to the cat command from file1.txt rather than keyboard input.

- `cat < file1.txt`

//create a pipeline, where the output of echo command is fed to input of bc command.

- `echo "100+200" | bc`

# Editing a file

- Open any editor that is available on the remote machine. Opening a GUI based editor may not be possible at the moment.
  - Example editors:
    - pico
    - nano
    - gedit
    - emacs
    - vim