

**SSH:**

Install Powershell first:   
You may have it already installed, search for Windows PowerShell in search bar. If not install from:  
[Installing PowerShell on Windows - PowerShell | Microsoft Learn](https://learn.microsoft.com/en-us/powershell/scripting/install/installing-powershell-on-windows?view=powershell-7.4)  
<https://learn.microsoft.com/en-us/powershell/scripting/install/installing-powershell-on-windows?view=powershell-7.4>

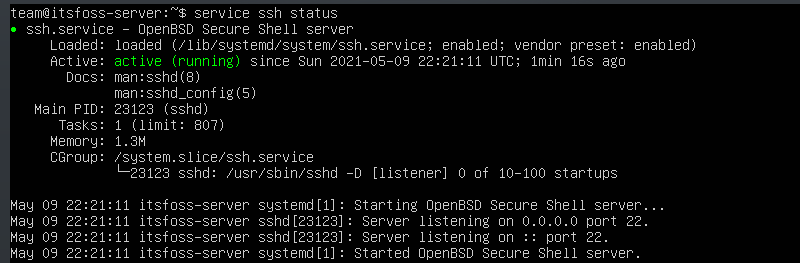
Follow the steps mentioned in this site:  
for installing ssh client and server.  
[Get started with OpenSSH for Windows | Microsoft Learn](https://learn.microsoft.com/en-us/windows-server/administration/openssh/openssh_install_firstuse?tabs=powershell)  
<https://learn.microsoft.com/en-us/windows-server/administration/openssh/openssh_install_firstuse?tabs=powershell>

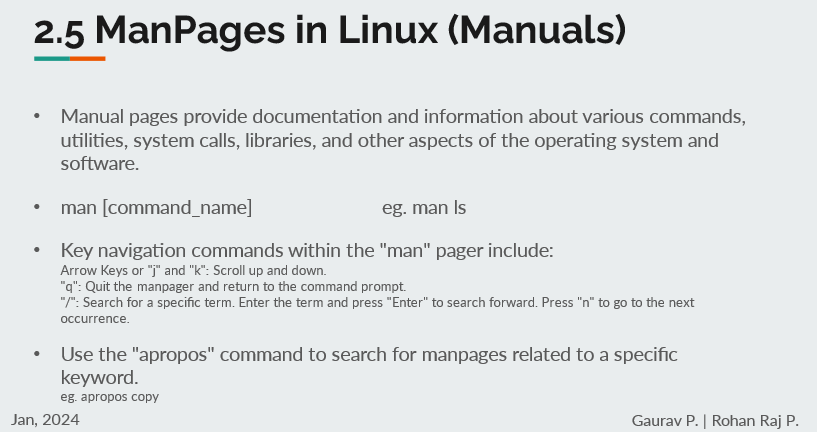
For starting ssh in powershell:

Start-Service sshd

For connecting with client:

ssh [user\_name]@[ip-address]  
example: ssh [student1@192.168.1.224](mailto:student1@192.168.1.224)  
put the password of that user (student1) the you may proceed to operating the server.

For making the host server:  
1. sudo apt update && sudo apt upgrade  
2. sudo apt install openssh-server  
3. service ssh status  
You should see something like this:  
  
if not try doing:  
sudo systemctl enable --now ssh  
4. sudo ufw allow ssh   
this is done to allow ssh through firewall  
5. sudo service ssh start  
this is done to start the service

**Man Pages:  
**

* Use normal scroll (of mouse/trackpad) or arrows key to move around
* Use ‘Space Key’ to move a page down
* Use ‘B’ key to move a page up
* Use ‘Q’ key to quit (as written in the last line of Man pages)
* Type ‘/ [keyword]’ to search for something in the man pages
* Normal searching of keyword in man pages in terminal  
  man -k [keyword]
* Try man man to know more about man pages

**Terminal Multiplexing:**

* Tmux and Screen
* Install tmux using – sudo apt install tmux
* New terminal – (control+b) and type ‘%’
* Switch terminal – (control+b) and use arrow keys
* Detach terminal – (control+b) and type ‘d’
* **HW: Differences between screen and tmux**

**Shell Scripting:**

1. Shell = command interpreter
2. Shell script = file containing set of valid linux commands
3. We needed touch command to create a new file, let’s directly create new file this time:  
   vi file.ext  
   vim file\_name.sh  
   nano test.py
4. For vi and vim text editor  
   - we need to go to insert mode by pressing ‘i’ key then we may be able to type in the text within it.  
   - # to write some command  
   - after completing the code, to write and exit the text editor we press ‘Esc’ and then type ‘:wq!’ then press enter
5. First script  
   >> nano first.sh  
   # This is my first script  
   echo This is my first shell script  
   >> sh first.sh :: to run shell in sh  
   >> bash first.sh :: to run shell in bash  
   **HW: Differences between bash and sh and their basic functionalities**  
   >> cat first.sh :: to view code written in the script file
6. Let’s take some input from user and then display them back  
   >> nano second.sh  
   # Ask user for their name and their favorite number and display it  
   echo Can I get your name?  
   read nme  
   echo What is your favorite number?  
   read num  
   echo Hello $nme , I guess your fav number is $num .  
   >> sh second.sh
7. Expression handling  
   - Operators  
   plus (+) is +  
   minus (-) is -  
   multiply (\* or x) is \\* [backward slash and asterisk]  
   divide (/) is /  
   modulus division (%) is %  
   - evaluation of operation  
   `expr $var1 [operator] $var2` eg. `expr $a \\* $b`  
   $((var1 [operator] var2)) eg. $((a + b))  
   -keep in mind the blank space while writing code  
   >>nano third.sh  
   # A simple calculator for two digits  
   echo Enter two numbers  
   read num1 num2  
   echo Sum = `expr $num1 + $num2`  
   echo Difference = `expr $num1 - $num2`  
   echo Product = `expr $num1 \\* $num2`  
   echo Quotient = `expr $num1 / $num2` and reminder = `expr $num1 % $num2`  
   >> sh third.sh  
   - ` is Acend, below Esc key, left side of number 1
8. Conditionals  
   equal to : -eq  
   greater than : -gt  
   lesser than: -lt  
   greater than or equal to : -ge  
   lesser than or equal to : -le  
   and: -a  
   or: -o  
   not: -n  
   - Normal if statement:  
   if [condition]  
   then  
   /Statements/  
   fi  
   eg:  
   read income  
   if [ $income -ge 100000 ]  
   then  
   echo You are rich  
   fi  
   - if then else   
   if [condition]  
   then  
   /Statements/  
   else  
   /Statements/  
   fi  
   eg:  
   read number  
   if [ $number -ge 0 ]  
   then  
   echo The given number is positive  
   else  
   echo The given number is negative  
   fi  
   - if - else if - else if - ……. – else  
   -need to close if statement with fi (with as many if written in the code)  
   >> nano largest  
   # Find largest of three numbers  
   read n1 n2 n3  
   if [ $n1 -ge $n2 -a $n1 -ge $n3 ]  
   then  
   echo $n1 is largest among all  
   else if [ $n2 -ge $n1 -a $n2 -ge $n3 ]  
   echo $n2 is largest among all  
   else if [ $n3 -ge $n1 -a $n3 -ge $n2 ]  
   echo $n3 is largest among all  
   else  
   echo Wrong Input  
   fi  
   fi  
   fi  
   >> sh largest.sh