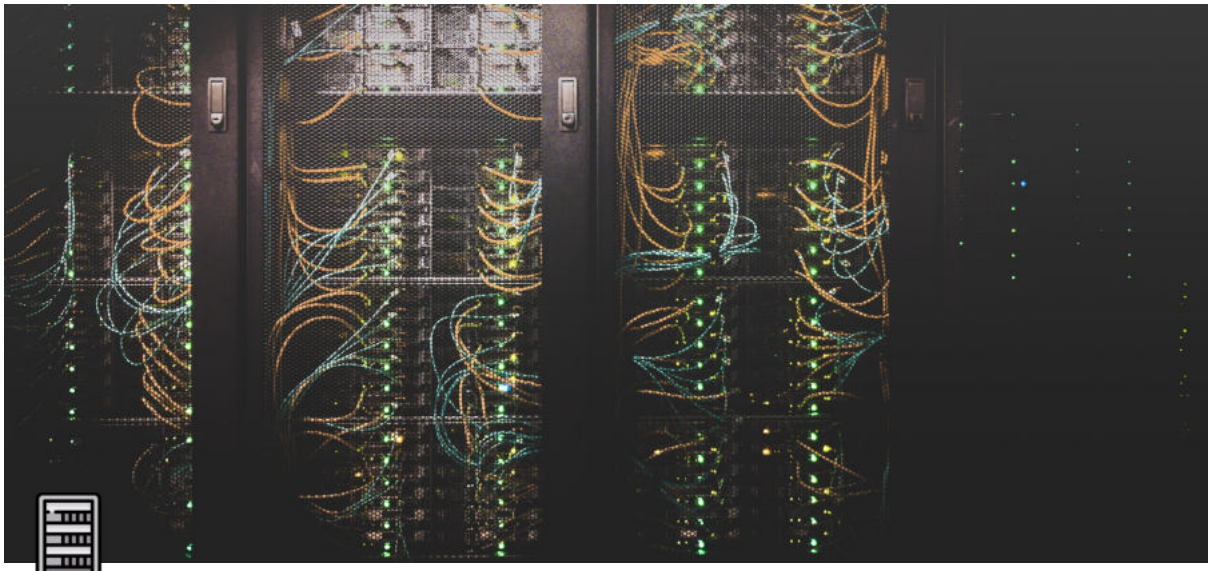


Knowing your Hardware

Type	📒 Lecture
Date	@March 14, 2022
Lecture #	1
Lecture URL	https://youtu.be/pbb5YQQhqXU
Notion URL	https://21f1003586.notion.site/Knowing-your-Hardware-5b54e4c1a93d457cbbe4be4a76f3d7fc
Week #	8

Packages to know hardware

```
clinfo, coreutils, dmidecode, fdisk, hardinfo, hdparm, hwinfo, lshw
memtester, net-tools, pciutils, procps, sysstat, upower, util-linux
```

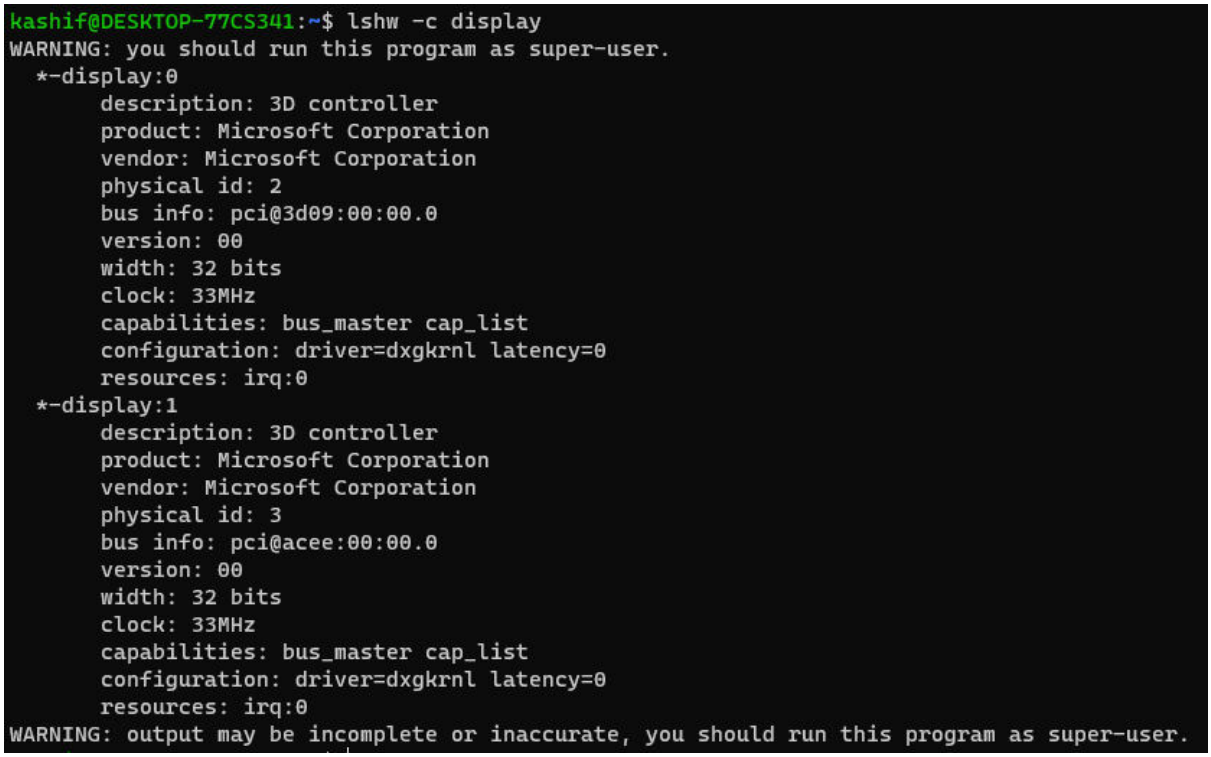
`hwinfo`

Gives us the hardware info

`lshw`

Lists the hardware

Example



To get details about the CPU

`cat /proc/cpuinfo`

Get details about the partitions

`cat /proc/partitions`

```
kashif@DESKTOP-77CS341:~$ cat /proc/partitions
major minor  #blocks  name

   1        0      65536 ram0
   1        1      65536 ram1
   1        2      65536 ram2
   1        3      65536 ram3
   1        4      65536 ram4
   1        5      65536 ram5
   1        6      65536 ram6
   1        7      65536 ram7
   1        8      65536 ram8
   1        9      65536 ram9
   1       10      65536 ram10
   1       11      65536 ram11
   1       12      65536 ram12
   1       13      65536 ram13
   1       14      65536 ram14
   1       15      65536 ram15
   8        0  268435456 sda
   8       16  268435456 sdb
```

How many block type devices (storage devices)?

`lsblk`

```
kashif@DESKTOP-77CS341:~$ lsblk
NAME MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda    8:0    0  256G  0 disk
sdb    8:16   0  256G  0 disk /
```

Gives us just the NAME and the SIZE

```
kashif@DESKTOP-77CS341:~$ lsblk -o NAME,SIZE
NAME  SIZE
sda   256G
sdb   256G
```

List all the PCIe devices

`lspci`

```
kashif@DESKTOP-77CS341:~$ lspci
3d09:00:00.0 3D controller: Microsoft Corporation Device 008e
acee:00:00.0 3D controller: Microsoft Corporation Device 008e
```

To know the # of DIMM modules installed

`sudo dmidecode --type memory`

Requires super user

Look at system details in the GUI

`hardinfo`

Make sure to install it first,

Might not work in WSL (Use WSLg for the GUI)

To know details about the GPU

`clinfo`

To know the battery status

`upower -e`

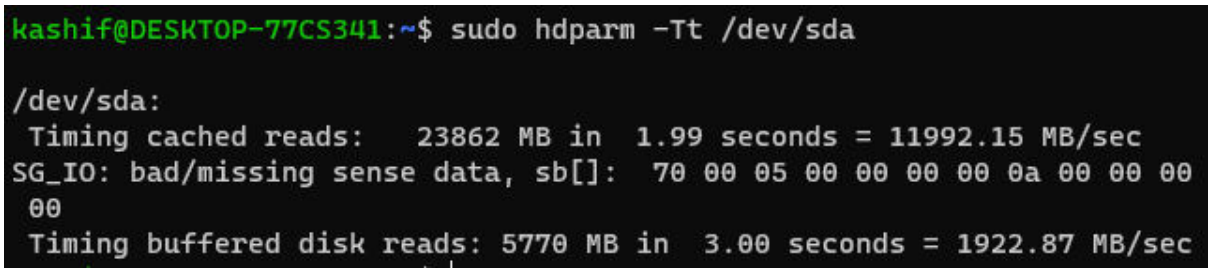
It will list a bunch of names, copy the entire path of your battery

`upower -i <battery-path>`

Storage benchmark?

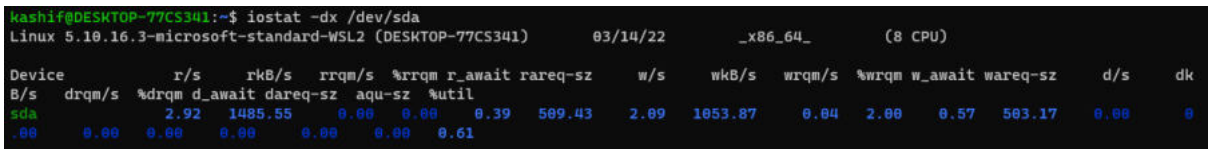
`sudo hdparm -Tt /dev/sda`

Requires super user



To know CPU and Storage usage (realtime?)

`iostat -dx /dev/sda`



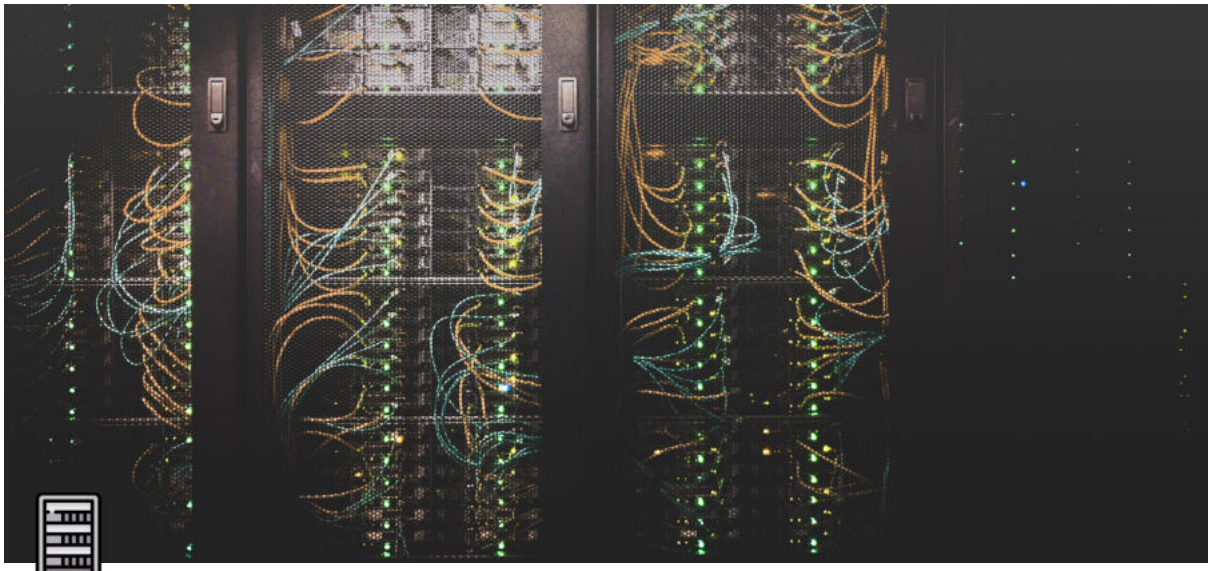
The numbers, Mason, what do they mean?

To know the network config

`ifconfig`

```
kashif@DESKTOP-77CS341:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.29.102.13  netmask 255.255.240.0  broadcast 172.29.111.255
        inet6 fe80::215:5dff:fe17:9306  prefixlen 64  scopeid 0x20<link>
        ether 00:15:5d:17:93:06  txqueuelen 1000  (Ethernet)
        RX packets 6241  bytes 6710652 (6.7 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1592  bytes 111257 (111.2 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 6  bytes 340 (340.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 6  bytes 340 (340.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```



Prompt Strings

Type	📒 Lecture
Date	@March 14, 2022
Lecture #	2
Lecture URL	https://youtu.be/UBjENhxwpcU
Notion URL	https://21f1003586.notion.site/Prompt-Strings-3cb3d692ac084eb7bd2d1c3fbe22209f
Week #	8

`bash` prompts

- PS1 → primary prompt string: `$`
- PS2 → secondary prompt for multi-line input: `>`
- PS3 → prompt string in select loops: `#?`
- PS4 → prompt string for execution trace: `+`

Escape Sequences

\A	Current time in 24-hour as hh:mm	\u	Current user's username
\d	Date in "weekday month day" format	\w	Current directory
\h	Hostname upto first period	\W	Basename of current directory
\H	Complete hostname	\#	Current command number
\s	Name of the shell	\$	If uid is 0, # else $
\t	Current time in 24-hour as hh:mm:ss	\@	Current time in 12-hour a.m/p.m
\T	Current time in 12-hour as hh:mm:ss	\\	A literal \ character

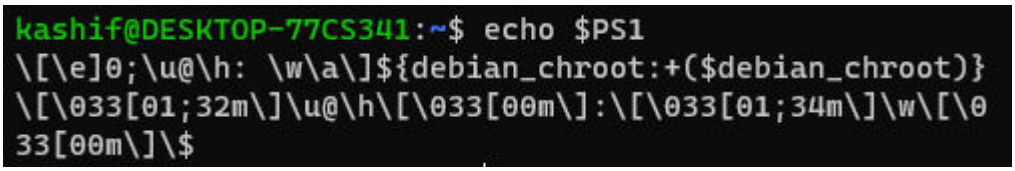
`\u@\h:\w\$` 👇

Username @ Machine name : **Current Directory** *Prompt Symbol*

Python command line

- `ps1` and `ps2` are defined in the module `sys`
- Change `sys.ps1` and `sys.ps2` if needed
- Override `__str__` method to have dynamic prompt

`>>>`



Modify the PS1 prompt by

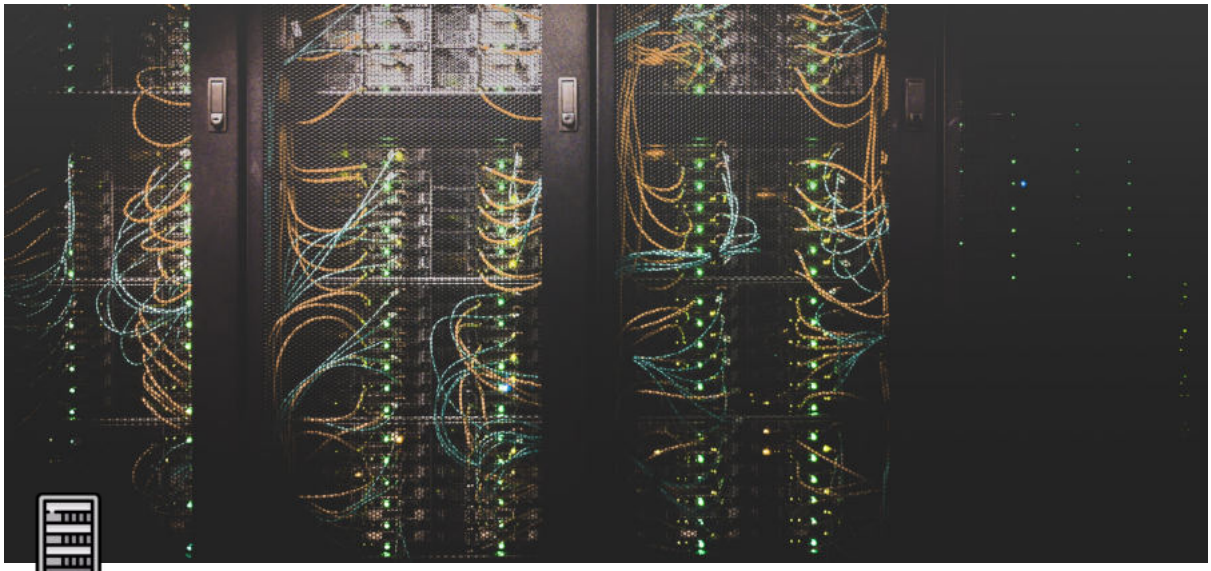
`PS1="\#:\$"`

If we want to go back to the default prompt → `source .bashrc`

Modify the PS2 prompt

`PS2='Hey \u, close the string: '`

Similar for PS3 and PS4



Some command line utilities

Type	📒 Lecture
Date	@March 14, 2022
Lecture #	3
Lecture URL	https://youtu.be/fNf74ycgD9w
Notion URL	https://21f1003586.notion.site/Some-command-line-utilities-acc242cefc3e4312a522a61e2befebe6
Week #	8

Utilities

- `find` → locating files and processing them
- `tar, gzip` etc. → packaging collections of files
- `make` → conditional actions

`find`

Usage: `find [pathnames] [conditions]`

`-name`	*pattern* to match filenames
`-type`	File type code eg., c for character file, d for directory, l for symbolic link etc.,
`-atime`	Files accessed +n (more than n), -n (less than n) days ago
`-ctime`	File changed +n (more than n), -n (less than n) days ago
`-regex`	Regular expression for *pattern* of filenames Combine with `-regextype` posix-basic, posix-egrep etc.,
`-exec`	Command to run using { } as place holder for filename
`-print`	Print the full path name of matching files

file packaging

- Deep file hierarchies
- Large number of tiny files
- `tar` → collect a file hierarchy into a single file
- `gzip` → compress a file
- **Applications** → backup, file sharing, reduce disc utilization

Possibilities

- `tar, zip`
- `compress` (ncompress), `gzip` (ncompress), `bzip2` (bzip2), `xz` (xz-utils), `7z` (p7zip-full)
- Tarballs like `bundle.tgz` for package + compress
- Time & memory required to shrink / expand versus size ratio
- Portability
- Unique names using timestamp, process ID etc., for backup tarballs

`make`

Usage: `make -f make.file`

```
# comments

TMP_FILES = *.o *.aux

.PHONY : clean

target : prerequisites
    recipe $(OPTION_NAME)

clean:
    rm -f $(TMP_FILES)
```

To compress using `tar`

```
tar -cvf filename.tar foldername/
```

To compress `.tar` file using `gzip`

```
gzip filename.tar
```

To uncompress a `.gz` file

```
gunzip filename.tar.gz
```

To compress a `.tar` file using `bzip2`

```
bzip2 filename.tar
```

To uncompress a `.bz2` file

```
bzip2 -d filename.tar.bz2
```

To compress a `.tar` file using `compress`

```
compress filename.tar
```

To uncompress a `.Z` file

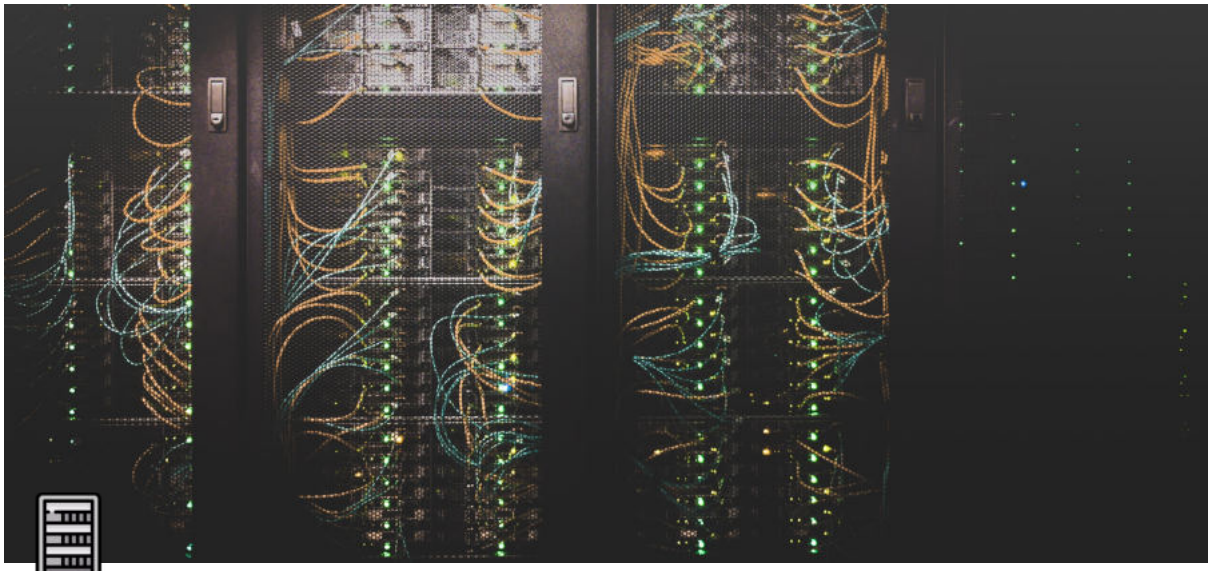
```
uncompress filename.tar.Z
```

To un-tar a file

```
tar -xzvf filename.tar
```

-xzvf → How to remember?

eXtract Ze Vucking Files



Networking commands and SSH

Type	📒 Lecture
Date	@March 14, 2022
Lecture #	4
Lecture URL	https://youtu.be/SxDIXtxR33c
Notion URL	https://21f1003586.notion.site/Networking-commands-and-SSH-9593fb49c2374246b525faff24f19091
Week #	8

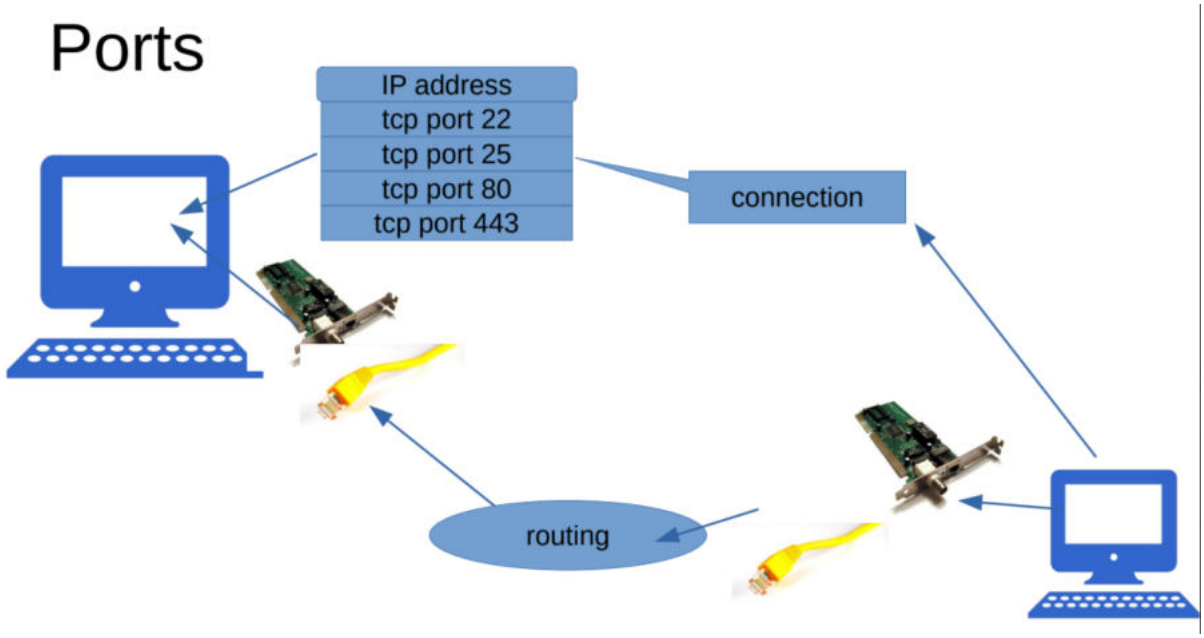
Network & SSH

Accessing remote machines on command line

IPv4 address range

- localhost

    - 127.0.0.0/8
- Private network
    - Class A: 10.0.0.0/8
        - 16,777,216
    - Class B: 172.16.0.0/12
        - 1,048,576
    - Class C: 192.168.0.0/16
        - 65,536
- Public network



Ways to gain remote access

- VPN access
- SSH tunnelling
- Remote desktop: x2go, rdp, pcoip
- Desktop over browser: Apache Guacomole
- Commercial, over internet: Teamviewer, AnyDesk, Zoho assist, ...

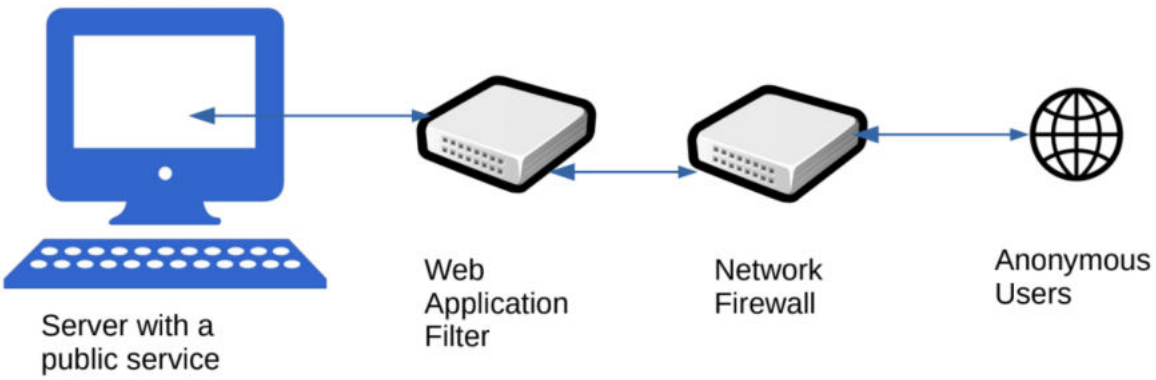
Some important ports

21	ftp	File transfer
22	ssh	Secure Shell
25	smtp	Simple Mail Transfer Protocol
80	http	Hypertext Tranfer Protocol
443	https	Secure Hypertext Tranfer Protocol
631	cups	Common Unix Printing System
3306	mysql	MySQL database

Firewall

- Ports open on my machine
- Ports needed to be accessed on remote machine
- Network routing over the port
- Firewall controls at each hop

Protecting a server



SELinux

- Security Enhanced Linux mode available on Ubuntu too, apart from server grade flavours like CentOS, Fedora, RHEL, SuSE Linux, etc.
- Additional layer of access control on files to services
- Role Based Access Control
- Process sandboxing, least privilege access for subjects
- Check using `ls -lZ` and `ps -eZ`
- RBAC items:
  - user (unconfined_u)
  - role (object_r)
  - type (user_home_t)
  - level (s0)
- Modes:
  - disabled
  - enforcing
  - permissive
- Tools:
  - semanage
  - restorecon

SELinux is recommended for all publicly visible servers

Network tools

ping	To see if the remote machine is up
traceroute	Diagnostics the hop timings to the remote machine
nslookup	Ask for conversion of IP address to name
dig	DNS lookup utility
netstat	Print network connections
mxtoolbox.com	For help with accessibility from public network
whois lookup	Who owns which domain name
nmap	(careful !) Network port scanner
wireshark	(careful !) Network protocol analyzer

High Performance Computing

- Look at www.top500.org for statistics
- Accessing a remote HPC machine is usually over SSH
- Long durations jobs are submitted to a job scheduler for execution
- Raw data if large needs to be processed remotely before being transferred to your machine
- Comfort with the command line is a must

```
kashif@DESKTOP-77CS341:~$ nslookup www.iitm.ac.in
Server:         172.29.96.1
Address:        172.29.96.1#53

Non-authoritative answer:
www.iitm.ac.in  canonical name = waf6.iitm.ac.in.
Name:   waf6.iitm.ac.in
Address: 103.158.42.57
```

Using 3rd party DNS lookup tool

8.8.8.8 (Google)

```
www.iitm.ac.in.     21600   IN  CNAME   waf6.iitm.ac.in.
waf6.iitm.ac.in.    21600   IN  A   103.158.42.57
```

208.67.222.222 (OpenDNS)

```
www.iitm.ac.in.     86400   IN  CNAME   waf6.iitm.ac.in.
waf6.iitm.ac.in.    86400   IN  A   103.158.42.57
```

1.1.1.1 (Cloudflare)

```
www.iitm.ac.in.     86400   IN  CNAME   waf6.iitm.ac.in.
waf6.iitm.ac.in.    86400   IN  A   103.158.42.57
```

9.9.9.9 (Quad9)

```
www.iitm.ac.in.     43200   IN  CNAME   waf6.iitm.ac.in.
waf6.iitm.ac.in.    43200   IN  A   103.158.42.57
```

Tool used here: https://tools.keycdn.com/dig

```
kashif@DESKTOP-77CS341:~$ nslookup www.mit.edu
Server:         172.29.96.1
Address:        172.29.96.1#53

Non-authoritative answer:
www.mit.edu     canonical name = www.mit.edu.edgekey.net.
www.mit.edu.edgekey.net canonical name = e9566.dscb.akamaiedge.net.
Name:   e9566.dscb.akamaiedge.net
Address: 72.247.54.42
Name:   e9566.dscb.akamaiedge.net
Address: 2600:140f:7800:182::255e
Name:   e9566.dscb.akamaiedge.net
Address: 2600:140f:7800:19e::255e
```

```
kashif@DESKTOP-77CS341:~$ dig www.google.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 60384
;; flags: qr rd ad; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;www.google.com.                        IN      A

;; ANSWER SECTION:
www.google.com.         0       IN      A       216.58.200.132

;; Query time: 0 msec
;; SERVER: 172.29.96.1#53(172.29.96.1)
;; WHEN: Mon Mar 14 14:56:20 IST 2022
;; MSG SIZE  rcvd: 62
```

To do a reverse lookup

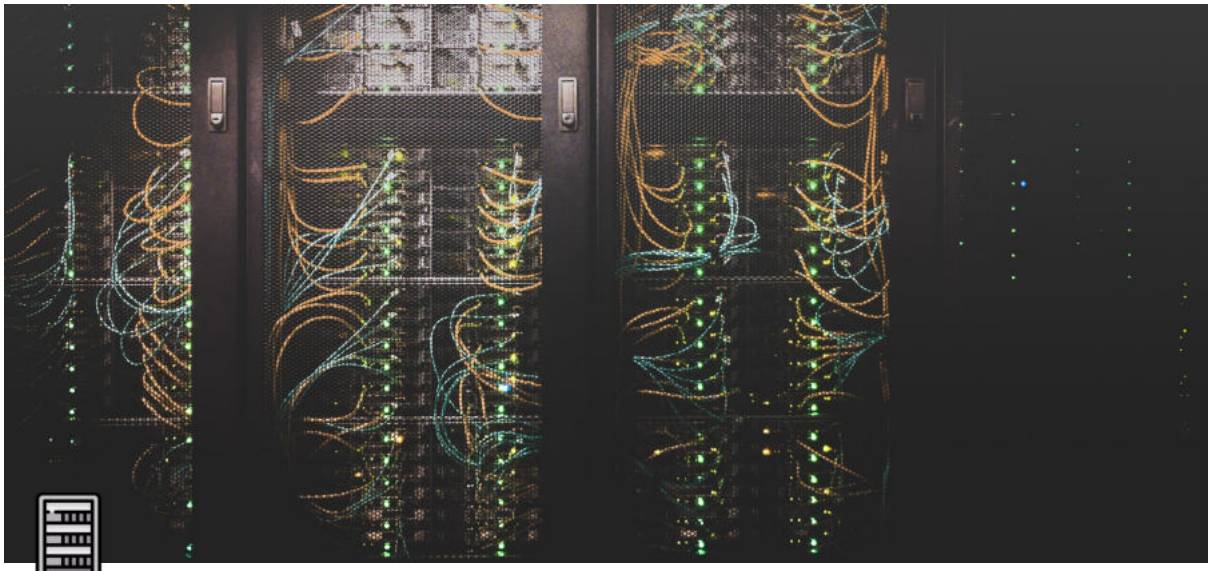
```
kashif@DESKTOP-77CS341:~$ dig -x 216.58.200.132

; <<>> DiG 9.16.1-Ubuntu <<>> -x 216.58.200.132
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 19691
;; flags: qr rd ad; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;132.200.58.216.in-addr.arpa.   IN      PTR

;; ANSWER SECTION:
132.200.58.216.in-addr.arpa. 0  IN      PTR     maa05s10-in-f4.1e100.net.

;; Query time: 10 msec
;; SERVER: 172.29.96.1#53(172.29.96.1)
;; WHEN: Mon Mar 14 14:57:07 IST 2022
;; MSG SIZE  rcvd: 110
```



Automating Scripts

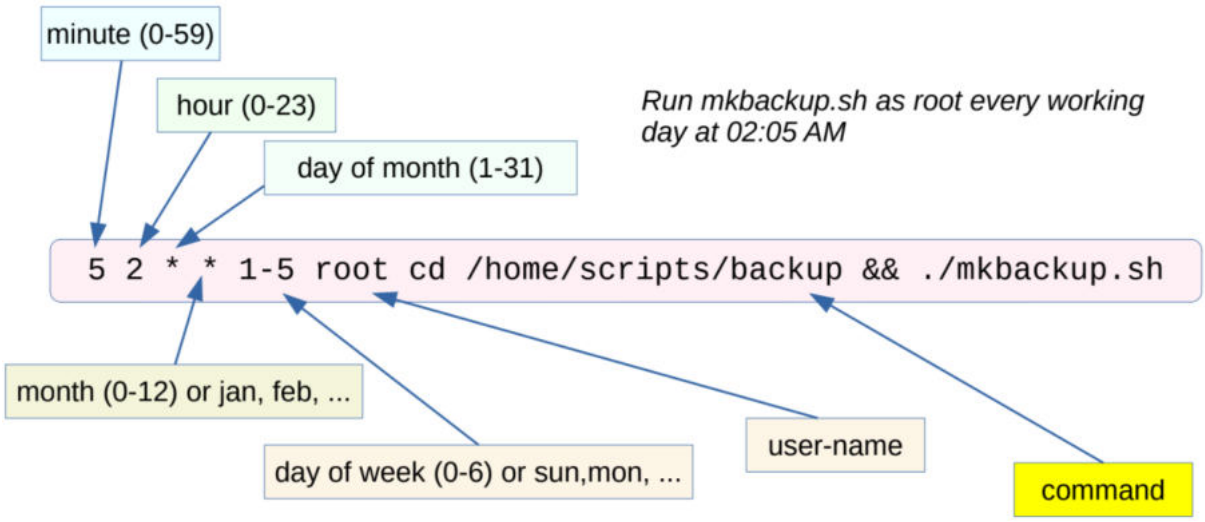
Type	📒 Lecture
Date	@March 14, 2022
Lecture #	5
Lecture URL	https://youtu.be/kWid87j6qIE
Notion URL	https://21f1003586.notion.site/Automating-Scripts-e44cc42c2e594a1aa2b77d6c176175e5
Week #	8

`cron`

- Service to run scripts automatically at scheduled times
- Tools:
  - `at`
  - `crontab`
  - `anacron`
  - `logrotate`

- Script locations:
  - `/etc/crontab`
  - `/etc/cron.d`
  - `/etc/cron.hourly`
  - `/etc/cron.daily`
  - `/etc/cron.weekly`
  - `/etc/cron.monthly`

Job definition



Startup scripts

- `/etc/init/`
- `/etc/init.d/`

Runlevel scripts

0	/etc/rc0.d/	Shutdown and power off
1	/etc/rc1.d/	Single user mode
2	/etc/rc2.d/	Non GUI multi-user mode w/o networking
3	/etc/rc3.d/	Non GUI multi-user mode with networking
4	/etc/rc4.d/	Non GUI multi-user mode for special purposes
5	**/etc/rc5.d/**	**GUI multi-user mode with networking**
6	/etc/rc6.d/	Shutdown and reboot

To create a `cron` job

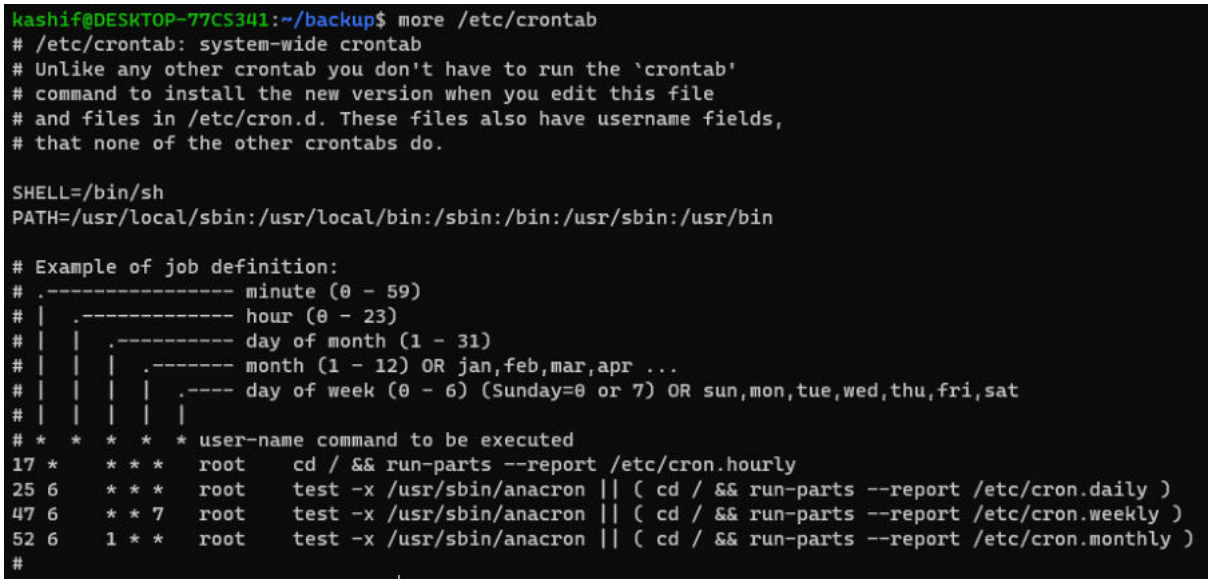
`crontab -e`

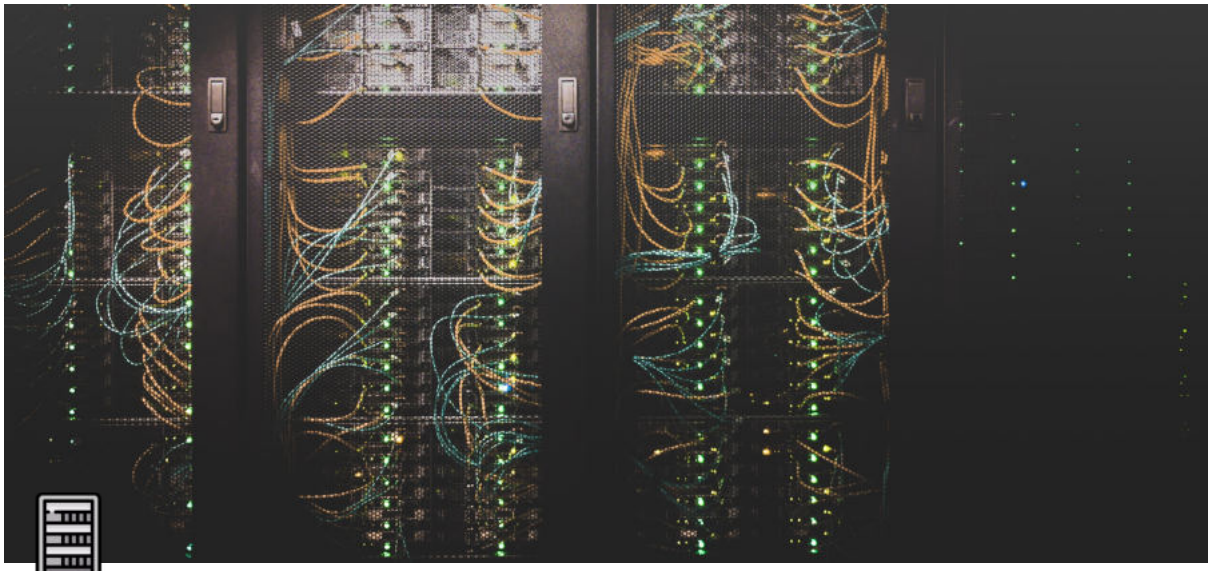
Add your cron job at the end of the file

```
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
27 * * * * cd /home/kashif/backup && ./mkbackup.sh
```

It should execute as per your mentioned cron timings

System-wide crontab


kashif@DESKTOP-77CS341:~/backup$ more /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.
SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
# Example of job definition:
# .---------------- minute (0 - 59)
# | .------------- hour (0 - 23)
# | | .---------- day of month (1 - 31)
# | | | .------- month (1 - 12) OR jan,feb,mar,apr ...
# | | | | .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# | | | | |
# * * * * * user-name command to be executed
17 * * * * root cd / && run-parts --report /etc/cron.hourly
25 6 * * * root test -x /usr/sbin/anacron || (cd / && run-parts --report /etc/cron.daily)
47 6 * * 7 root test -x /usr/sbin/anacron || (cd / && run-parts --report /etc/cron.weekly)
52 6 1 * * root test -x /usr/sbin/anacron || (cd / && run-parts --report /etc/cron.monthly)
#



Managing Storage

Type	📒 Lecture
Date	@March 14, 2022
Lecture #	6
Lecture URL	https://youtu.be/BvAtNUX7da4
Notion URL	https://21f1003586.notion.site/Managing-Storage-6105020474934a0fbdb51c66def065b8
Week #	8

LVM

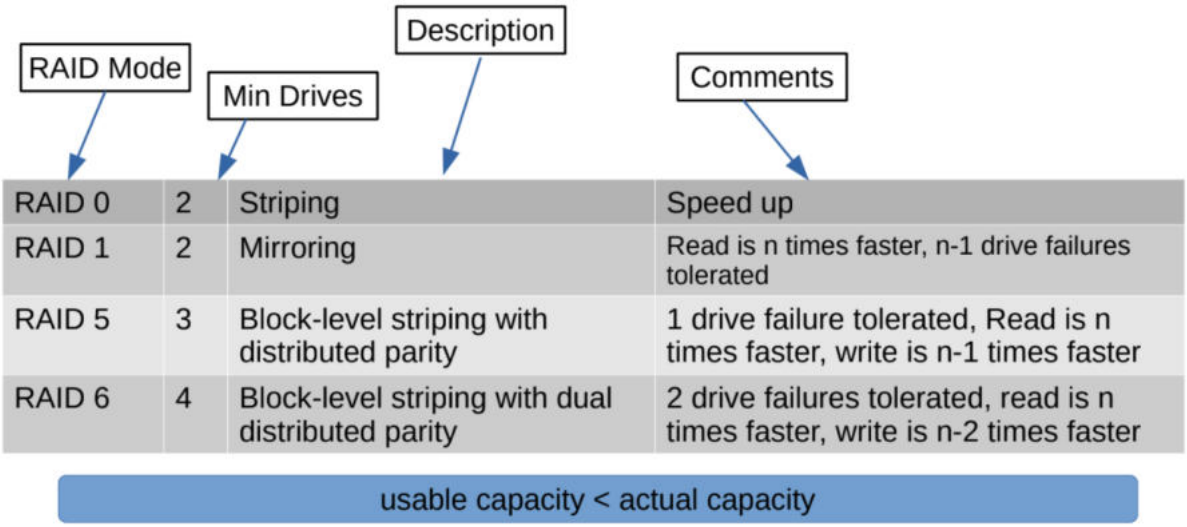
- Logical Volume Management
- Pooling multiple storage devices as a single logical volume
- lvm2 tools: create and manage virtual block devices from physical devices

RAID

- Redundant Array of Independent Disks

- Distributing data over multiple disks for redundancy/speed/increased capacity
- Raid controller: software or hardware

RAID modes



RAID Mode	Min Drives	Description	Comments
RAID 0	2	Striping	Speed up
RAID 1	2	Mirroring	Read is n times faster, n-1 drive failures tolerated
RAID 5	3	Block-level striping with distributed parity	1 drive failure tolerated, Read is n times faster, write is n-1 times faster
RAID 6	4	Block-level striping with dual distributed parity	2 drive failures tolerated, read is n times faster, write is n-2 times faster

usable capacity < actual capacity