Page No. | Teacher's Sign / Remarks

PAGE NO.:
DATE: / /

Linux and python

You should have Accounts on there →
1. AWS cloud
2. GITHUB
3. Twitter

Techno → operating system
language

(Yes, we need to use bc command to use calculator)

Basic → 2+3 - calculator (app/ Software) - Interpreter - Kernel (program) - CPU - RAM - Harddisk

(Software - Interpreter - Kernel) → Operating System

Windows (99% users works on it)
Linux (only 1% users work on it)

Windows is also known as close source

Linux

1. open source Software
2. Terminal
3. command

PAGE NO.:
DATE: / /

In Micro Soft, untitled document is termed as Note pad and in Linux it is termed as Gedit (Graphical edit)

In Microsoft to open any icon we need to click on it then only its run. But in linux we can type and open any icon and run it through terminal

Linux is an open-source operating system.

Kernel is an computer program which takes input from user gives output and stores in Hard drive further.

Kernel is termed as NT in Microsoft. Kernel is termed as Darwin in MACBOOK. And Kernel is termed as Kernel in linux (unix or linux in other softwares)

when we download any software

In Microsoft its .exe

In MacBook its .Dmg

PAGE NO.:
DATE: / /

In linux its -Rpm (Redhat) or -Deb (ubuntu and mint)

1. Firefox
2. cd Desktop
3. Mkdir Hello
4. For (i=1; i<20; i++))
   { do
   mkdir adhocnw$i
   Done }

Change monitor - Desktop - Downloads
Date (To change the location of a file creating)
concept of Recycle**

j stands for julian

Pwd ⇒ present working directory

If anything starting with d its directory and if its starting with something else its file

★ Task 1 → To show download softwares or file on Desktop directy.
Task 2 → whatever we type on terminal must show error.
Task 3 → Concept of recycle

Task 4 → In calendar sept 1752 why 2nd is missing 3 to 13 days

Windows →

① Admin (The user who ran as previously)
② Notepad (untitled document)
③ CMD (command line)
④ Internet explorer
⑤ .exe
⑥ Program files goes automatically in C-drive (for eg → c:\users\adhoc)

Redhat (linux) →

① Root
② Credit
③ Terminal
④ Firefox
⑤ Rpm
⑥ Program files goes automatically in /drive linux is known as directory structure (for eg → /:\Home\adhoc)

mkdir means → make directory
rmdir means → Remove directory

Python

PAGE NO.:
DATE: / /

Sept. 17 52
3 to 13 days

u permission

iary
adhoc)

y

PAGE NO.:
DATE: / /

All tech integrity →
1. Pi (+python)
2. web app (Django flask)
3. Database
4. operating systems - linux, Microsoft and MacBook.
5. Network
6. Big Data
7. cloud
8. .Net, Iron python
9. Java, Jython

Python version → Python 3.6
SDLC → Software development life cycle

Python → code → run
C language → code → compiler → run

To give values to variables in python →
Directly x = 10 or y = "Hi"

Q1 why there is 4 bytes for integer in python?

PAGE NO.:
DATE: / /

Data types

① Immutable → int, float, string, byte, tuple
② Mutable → list, set, dictionary

rmdir → deletes any directory
rm → deletes a single file
rm -rf → deletes all directories as well as files.

code can be linked and run, type and run and as well as you run too through kernel.

SHELL

Date → Shell → Kernel → Hardware
(converting human language in machine language)

Hardware → Kernel → Shell → Date
(converting machine language in human language)

Shell basically is a gateway

which parses human language to kernel by converting it into machine language and machine language through kernel to shell by converting it into human language which is further displayed as an output on the screen for the user.

Date → Shell → Hard disk

Not found found

If the Hard disk shows not found its directly send to shell and then shell replies answer (command not found)

If the Hard disk shows found it sends to Kernel → Hardware then to shell and then answer.

→ If we type sudo as prefix with any command then it shows all the process and codes running in BackHand. (which is not shown until we type sudo) to open a file or a folder

Shells name in linux is Bash (Default)

More shells name →
1. Zsh
2. Bash
3. Sh
4. TCSh
5. CSh

Operating system → change → etc etc → extra configuration)

Eg. 1 ← Cedit/etc/shells → shows all shells we have and we can change/edit anything also

Eg. 2 ← cat/etc/shells → shows all shells but we cant change/edit anything in it.

Features of Shells →

① History (can be stored up to 1-1000, exceding 1000 will delete one from starting that is 1 and adds one more History 1001, after that it looks like (2-1001)

PAGE NO.
DATE / /

2. variables (x, y) → variables are used to stores values or some varchar value.

3. alias (changes the name of the command but the command remains same.

4. script (records all commands happening on screen in terminal)

tput makes font Bold

script reply shell class.txt (filename)
(shows reply of the script

script -t 2>shell class.txt → saves with time

Shell script / pro

Task5 → How to show History with Date and time?

Task 6 → ① How to update history if entered manually in /Home/adhoc/.bash

② History (without reboot or logging out the present terminal)

PS1 = "changes the terminal user name"

gnome-terminal → Brings back old or default terminal name

env → shows all pre-defined variables

If I want to change date name but the funtion should remain the same → use alias

>>> alias d="date" (Then typing d will show date)

tput setab 0,7,4 → changes Background color

Touch command is used to create file

Software Installer →

Windows → MSI
RedHat → yum / DNF
Ubuntu → apt / apt-get
MAC → Brew

...ges the terminal
...x → Brings back old
...minal name
...e pre-defined
...change date name
...should remain
...alias
...(Then typing
...anges
...ed to

sudo converts to root

To install → (software in)
python → pip
os → brut
Ruby → gem

Python →
① Interpreter - Each
② File
③ IDE

vulture and fisefly shows all dead code in a programe in % form

How to create python →
① create file → write code
② Assign execute permission
③ Run

(.) → current
./filename → This will show current file you working in.

PAGE NO.:
DATE: / /

chmod → change mode

Q2 How will OS come to know that the code we are running is of python.

#! → she Bang or hash Bang

Q3 Install VLC in windows with any programming?

docs.python.org → website of python (any doubts related to it)

t = ()
use case of tuple

data can be added in list but cant in tuple

X.append(data) → will be inserted at last

X.Insert(data) → (2, data) → can insert data at particular location

X.remove(data) → removes the value.

X.pop(data) → removes data of that particular position

Replace data → x(0) exclude old data

x(0) → new data name
x (new data name will be replaced by old one)

Set → unique collection of Homogeneous and Heterogeneous data.

GSOC → google Summer of code
SCM → Source code Management
.Bash|rc → Makes the change permanent in terminal though its closed else its Never Saved.

Q4 The file should not run by clicking icon instead run with typing in terminal?

Input Output Redirection

Output Redirection → Showing output at another place instead of present terminal.

1) file
2) Ram
3) E-Mail
4) Printer
5) Another system
6) whatsapp

PAGE NO.:
DATE: / /

Q5 How to write "Hello World" in a directory?

date > filename.txt → displays result in a file.

date > filename.txt → Typing the command again in the same filename will replace the old answer/output with another one.

↳ But if the command is wrong it will display in the terminal itself "command not found"

date >> filename.txt → This command adds the new data in the existing data/output but at last (appends it) or we can say suffix to the existing data

date 2> filename.txt → this command displays the error in a file (name) you entered in the command (putting 2 in front of >)

But if
display
only.

date
adds
existi

ITS
that
or n

#

#

#

PAGE NO.
DATE : / /

But if the code is right it will display output in the terminal only.

date 2>> filename.txt → It also adds the new output in the existing output as a suffix

Its in the hands of the developer that the code should show error or not.

code → run → exit code

If the code shows 0 as exit code then its success and if the exit code is non-zero its failure

Exit code → echo $?

&> filename.txt → & sign sends the success code as well as failure code in the selected file.

&>> filename.txt → this will append success and failure in the file as a suffix.

wc → word count

→ wc -l filename.txt → helps you to count the number of line in a particular file.

→ wc -w filename.txt → helps you to count the number of words in a particular file.

→ wc -c/-m filename.txt → helps you to count the number of characters/bytes in a particular fi

→ wc - filename.txt → helps you to count the number of line, word and characters in a particular fil

grep → helps to word search in a file.

→ grep -i wordname filename.txt → only a particular word will be printed and it is not case-sensitive that means it will not only print capital letter word but also small letter word. It will print both. For eg → If wordname

is HELLO it will print all HELLO in a file whether it HELLO or hello.

grep stands for → general regular expression patter.

→ grep -w wordname filename → This also prints a particular word from the file but it is case-sensitive that means if the wordname is starting with capital letter it will print only none starting with capital letter, it will not print same word starting with small letter. For eg → If wordname is Hello it will print only Hello words from file not hello.

Ansible

Step-1 → Pick 2 Machines

OS1 OS2

Step-2 → OS1 → Install Ansible

Step-3 → Assign Password to ec2-user in OS2

Step-4 → Change /etc/ssh/sshd_config to accept password in OS2

Step-5 → Restart Sshd service (SystemCtl Restart Sshd)

Step-6 → In OS1

In ansible Inventory file

→ etc/ansible/hosts

In hosts file add IP addsess of OS2 either at the Starting or Last of the file.

PAGE NO.:
DATE: / /

ansible basically helps to work on multiple pc's together and also helps us to wipe out all the data from all the pc's at the same point of time.

ansible a -u ec2-user -m ping

(-u → user; a → group name; -m → module)

ansible a -u ec2-user -m ping -K

↓

will let you enter the password for the OS/ec2-user's

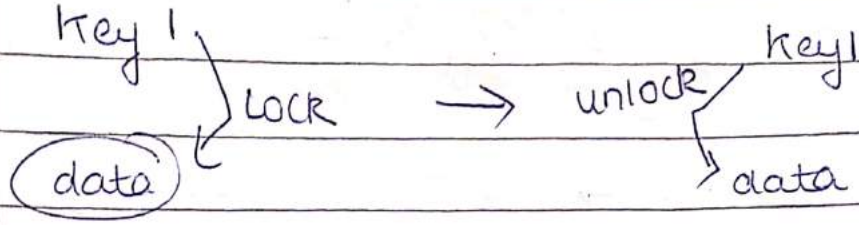
Authentication

RSA → makes key's for encryption and decryption

PAGE NO.:
DATE: / /

ssh-keygen → To make a key
And press only enter 2-3 times.

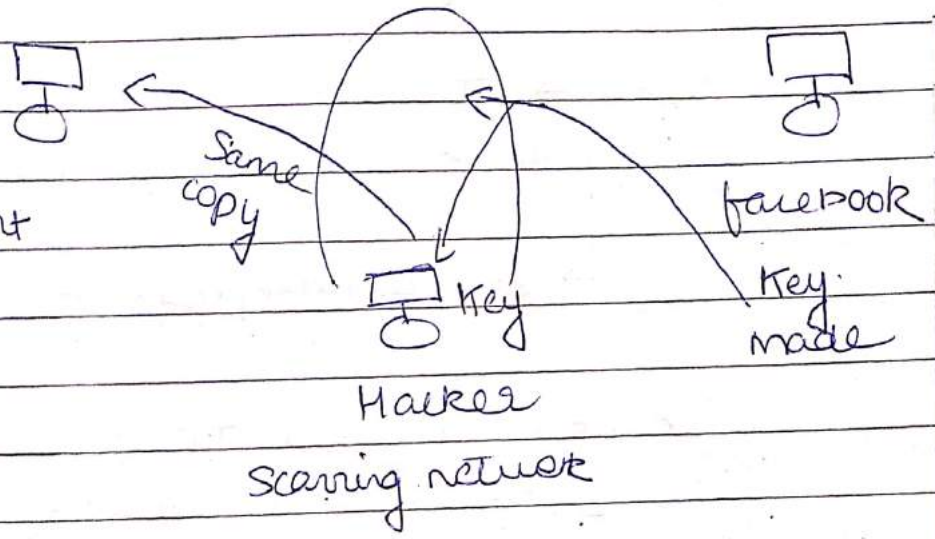
ssh-copy-id ec2-user@IPaddr



Sender receiver

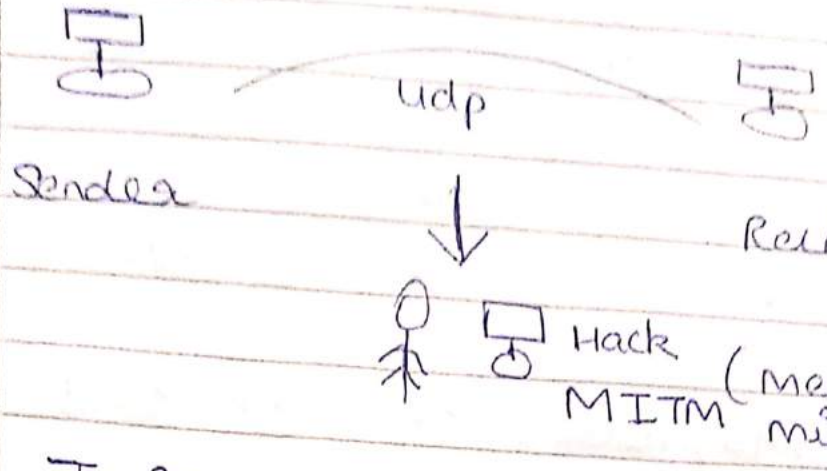
To make symetric key we AES or DES

Symetric → same key encryption and same key decryption.

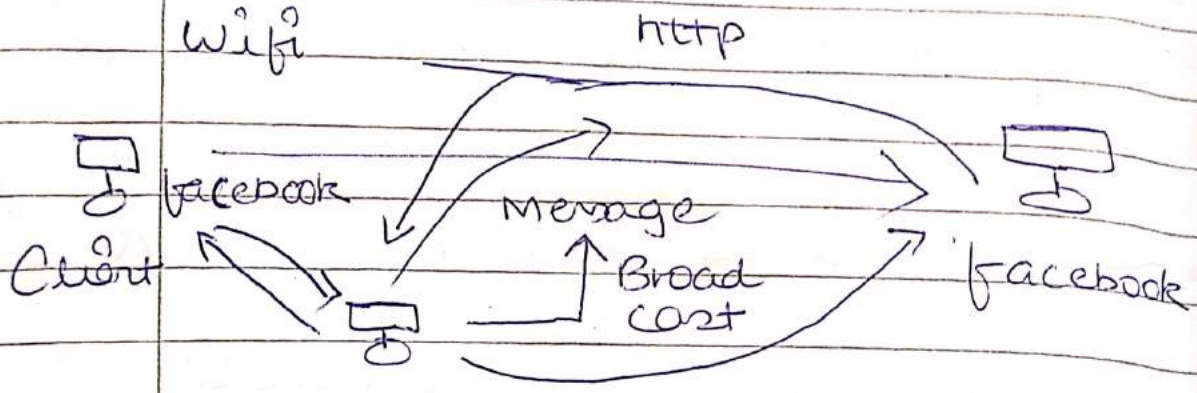


PAGE NO :
DATE : / /

Python



In Both codes sender.py and reciver.py, IP of only reciver is shared



Foreg → Cryptography → Secret writing
↳ used for sending message in a secret code.

Hello → Olleh | Or writing alphabets in 1-26 numbers.

encrypt
↙ ↓ ↘
Symetric | Asymetric

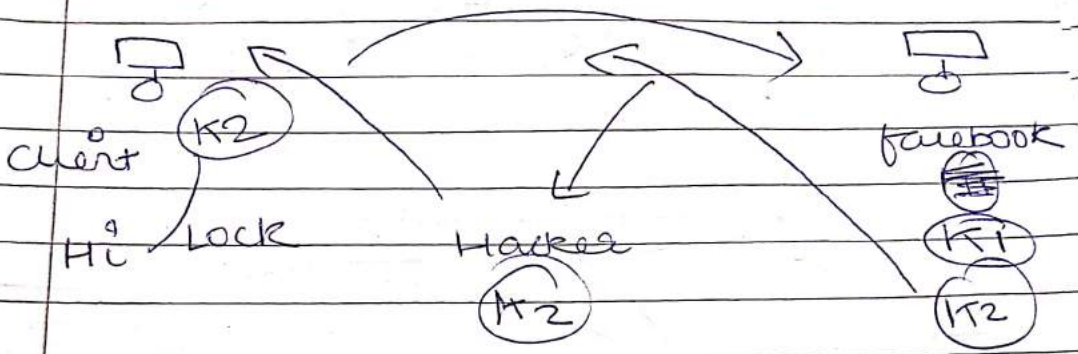
PAGE NO.:
DATE: / /

Asymetric

K1 K2

This means 2 keys one for lock and second for unlock.

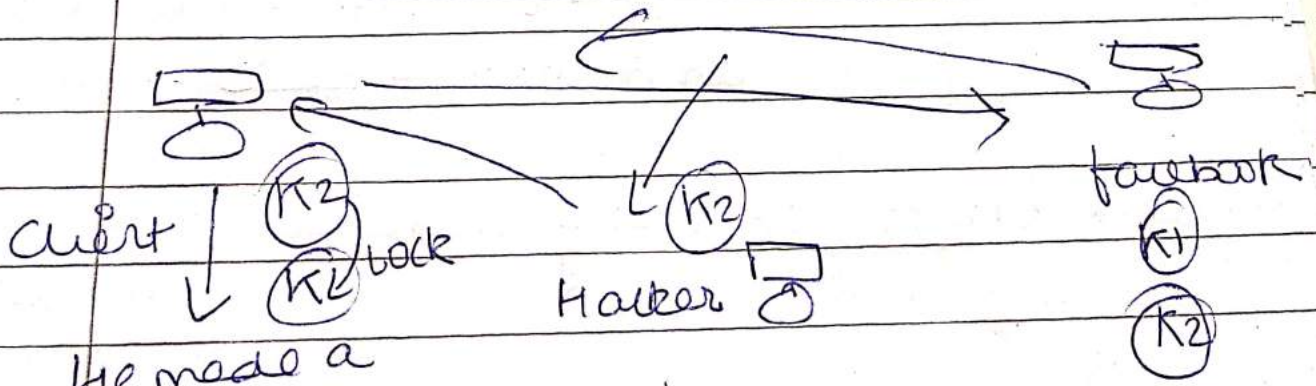
Asymetric → RSA | DSA
Symetric → AES | DES



New can be opened only with K1.

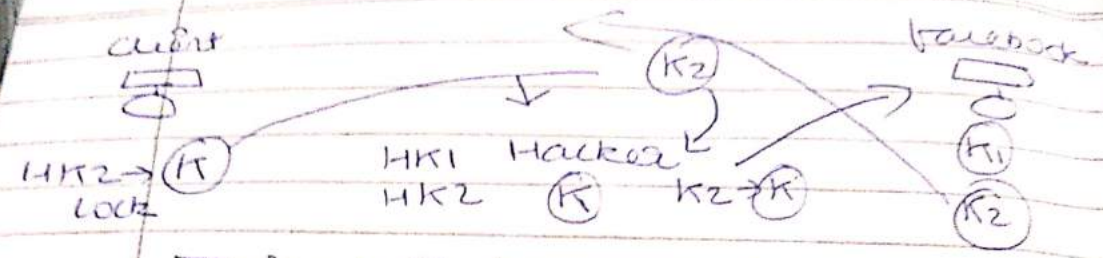
Security hybrid Model

Symetric + Asymetric



He made a new symetric key named K

He send K is locking K2



This time he don't send K2 to further and made his ~~K~~ own HK1 and HK2 → Hacker key

Hacker sent HK2 → client

client → K → HK2 → facebook

Hacker unlocked → HK2 with HK1 and got K

Then he sent K in K2 to facebook

By this all 3 got the data

SSL HeartBleed

PAGE NO.:
DATE: / /

OS1 → OS2

Ansible — ec2-user

Ansible ↓ ping → Target 35.153.127.151

SSh ec2-user@35.153.127.151 col

vim /etc/ansible/HOSTS

In top of the file insert

[e]

35.153.127.151

:wq (save file)

cat /etc/ansible/HOSTS

ansible a -u ec2-user -m ping -k

password → space space 321

Command → | > >>

Idempotency

Ansible module →

contribution ← Python

HTML code → EC2 → Apache HOST

PAGE NO.:
DATE: / /

yum
service
shell
copy

→ Store them in a file

↓

In ansible this file is called PlayBook

Language is YAML

→ ansible localhost -m shell -a "yum install httpd -y"

→ ansible localhost -m yum -a "name=httpd state=present"

↓

same only change name = service

→ ansible localhost -m service -a "name=httpd state=started"

echo hello ansible > index.html

cat index.html

ansible localhost -m copy -a

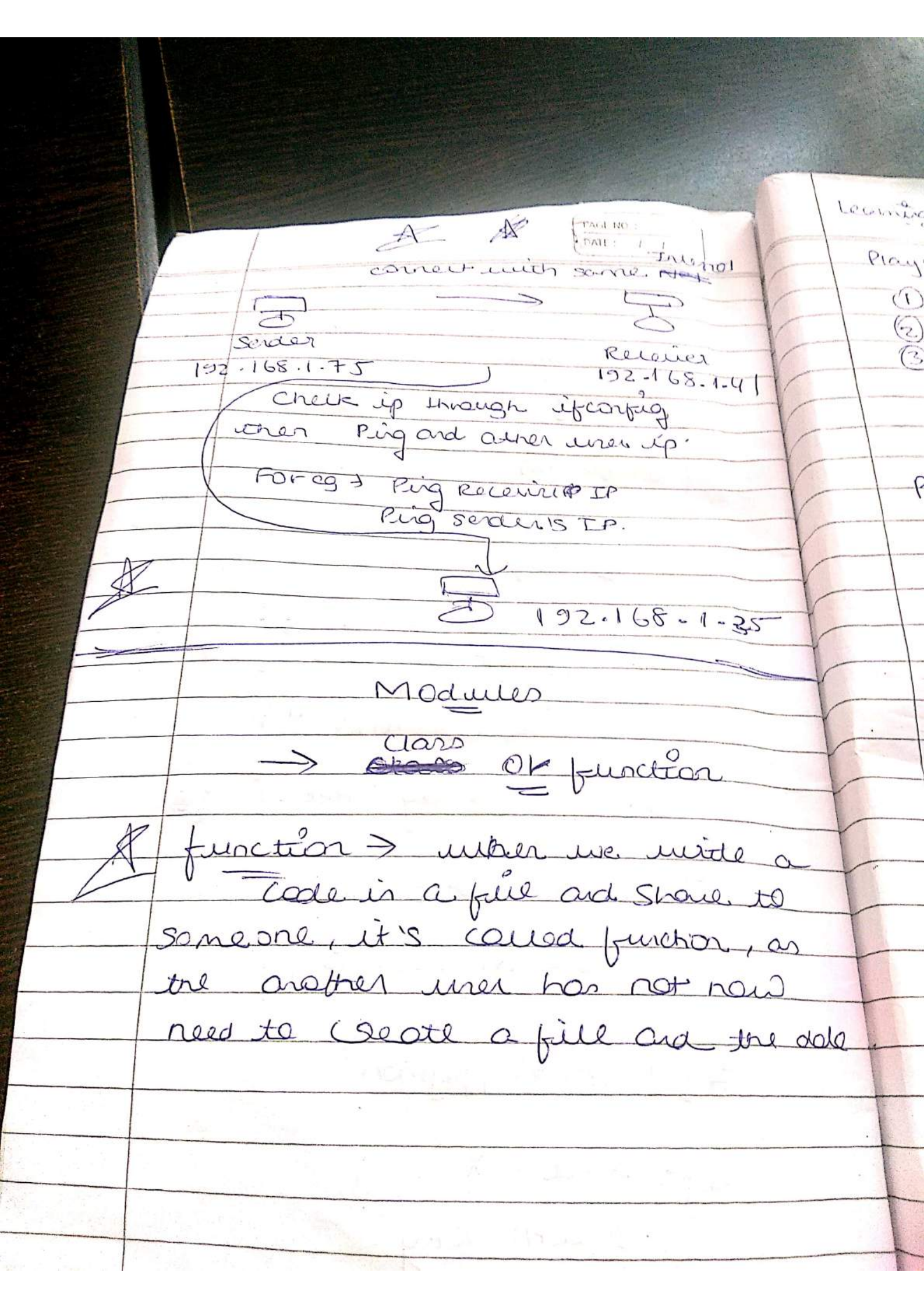
Learning YAML Language

PAGE NO.:
DATE: / /

Playbook as 3 sections

1. Target (group)
2. variable (optional)
3. Task (module which we want to execute)

playbook is started with 3hyper



Modules

→ Class or function

function → when we write a code in a file and share to someone, it's called function, as the another user has not now need to create a file and the code

PAGE NO.:
DATE: / /

Python function

without name

Lambda function

with name

In python function define is done by def command

Eg → def Hello():
print ("Hello python")
print(3+5)
print("Hiii")

then run it by Hello()

To make our own module

vi mymodule.py

#!/usr/bin/python3

def add(x,y);
return x+y

def two();

PAGE NO.:
DATE: / /

```
print("Hello World")

def clearScr():
    print("\n"*50)
```

cat mymodule.py

python3

```
import mymodule
dir(mymodule)

mymodule.hw()
mymodule.add(4,8)
```

PAGE NO.:
DATE: / /

python3 → Python 2
↳ urllib2

Module →

→ requests

google Backhand → right click → view page source
↳ To read the data of http page

~~cat~~ Getcode.py ↴ file

#!/usr/bin/python3

https
Response code →

404
505
302
200

→ mostly found errors when the webpage is not available

www.g.c

Public IP

(1) http
DNS

(2) g - IP → Apache

(3) https | http

Cloud

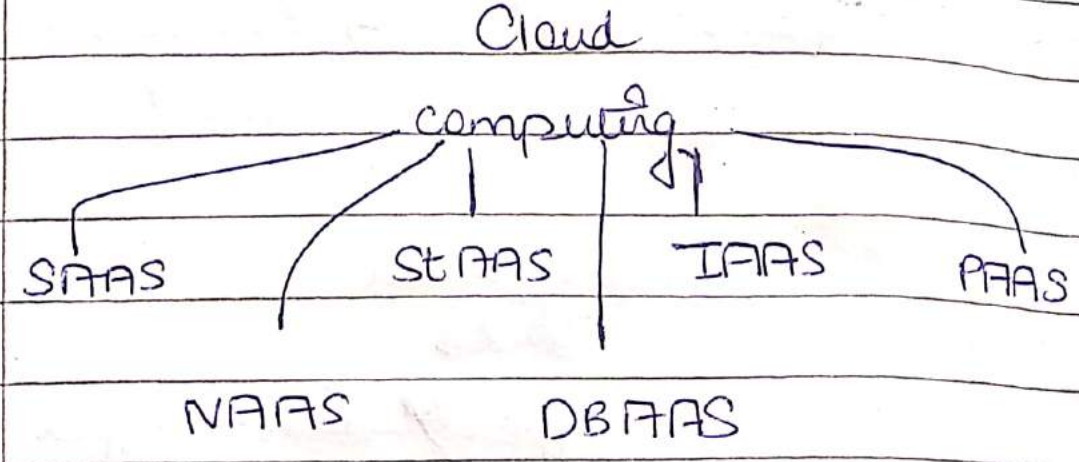
PAGE NO.:
DATE: / /

SAAS → Software as a Service
eg → Google

StAAS → Storage as a Service
eg → Harddisk, onedrive

IAAS → Infrastructure as a Service
eg → RAM, CPU, GPU

PAAS → Platform as a Service
eg →



DBAAS → Database as a Service
eg →

NAAS → Network as a service.

PAGE NO.:
DATE: / /

Cloud Provider →

AWS → 61% users
Azure
google
Openstack
IBM
Rackspace

AWS → Amazon web services

IAAS → EC2
(Infrastructure) ↳ elastic compute cloud

Data center
Racks
~~hypervisor~~ → Hypervisor
Hardware
↳ (Ram, CPU, HDD)

Hypervisor → helps to use multiple virtual BOX in a single OS.

Hardware → Hypervisor → Racks
↓
Data center

PAGE NO.:
DATE: / /

Hypervisor

Type 1 / Baremetal

Type 2 / Hosted

VMware → Top most Hypervisor

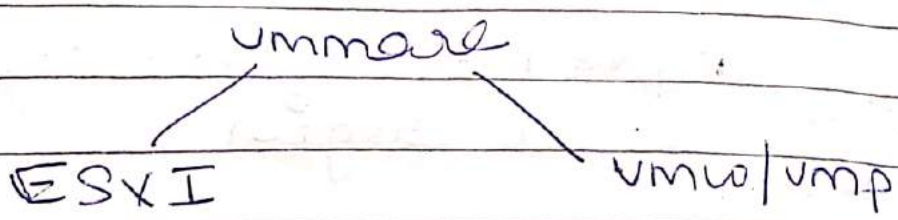
~~Server~~ Server
or Native

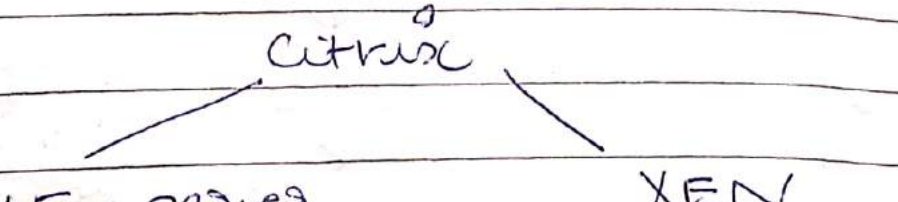
Citrix → Second most

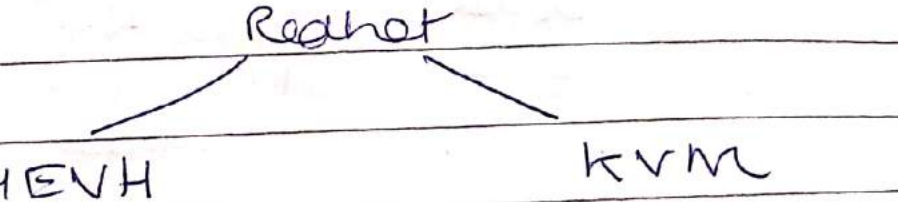
Redhat → Third most

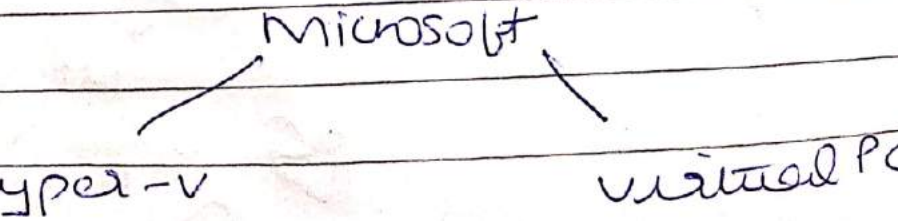
Microsoft → 4th most

Oracle → 5th most.







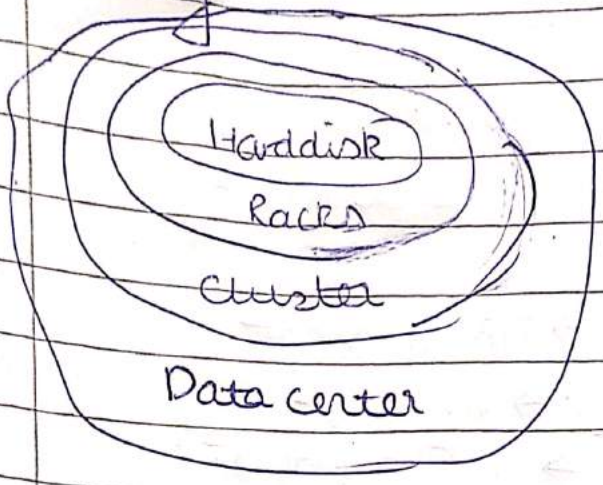


PAGE NO.:
DATE: / /

Oracle

VM | virtual BOX

Amazon uses citrix company for hypervisor and it type 1 only



More than one Data center is called region.

AWS
↓
EC2
↓
Region
Data center | DC
↓
Cluster
↓
Racks
↓
Hardware
↓
Hypervisor
OS

In Amazon Data center is termed as Availability zone.

Ram, cpu → Hypervisor
ECU → [redacted] (AWS)

In AWS Harddisk is termed as
EBS

In AWS Ram, cpu is termed as
ECU

" " OS Image " "
AMI

" " Network " "
VPC

Q How many Instances we can launch at once?

Q Can we change availability zones after launching Instance?

Boot [redacted] strapping

Security group name
→ awsjunebatchsecgroup

PAGE NO.:
DATE: / /

1024
1024
1048576 KB

EBS → elastic BLOCK storage

VPC → helps you to create network

AMI →

ECU →

Bigdata Hadoop

↳ huge Amount of Data

→ Store → where? How?

The price of perdrive depends upon its read and write speed.

↓

This is known as

IO Rate.

Ram and CPU never plays there role in coping, cutting or pasting, it is the work of Hard disk.

while Sending data or receiving data. always minimum speed is considered

Approach →

Foreg → 1 user (His HDD has 10MBps speed) sending → receiving ← 2 user (His HDD has 100MBPS speed)

If the user 1 is sending is sending data then only 10MBPS speed will only be considered and vice-versa.

Bigdata
↳ Problem:
↳ ~~Storage~~
Store (Speed, size)
↓
hadoop (Techno) → process
↳ Framework → Business.

Hadoop companies

Apache (hadoop) ↓ Open Source hadoop.

cloudera (hadoop) ↓ Not free charges are their

Hortonworks (hadoop) ↓ Not free charges are their

PAGE NO.:
DATE: / /

Apache hadoop

Version 1 — V2 — V3 (Latest)

We would be working on this of

↓

hadoop 2.7

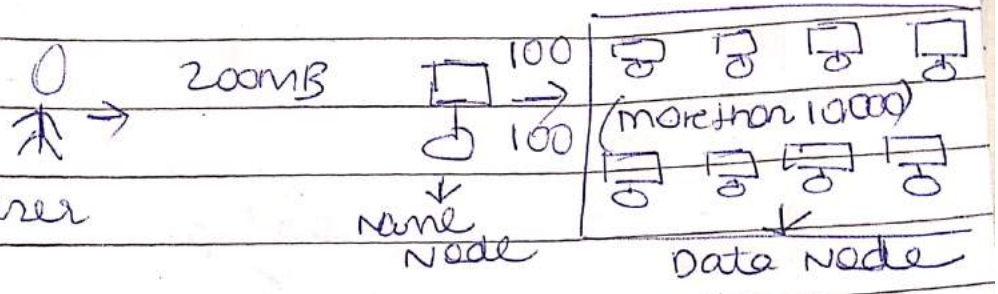
+

Redhat 7.5

hadoop is a framework/platform which gives solutions related to Big data in different ways.

hadoop distributed File system (HDFS) → It is a component to store Big data.

HDFS Architecture



Eg → we sent 200 MB data then it divides into small parts like 100 MB data and one more 100 MB data and store into many machines.

PAGE NO.:
DATE: / /

and each part is stored on min. three pc's so that if by chance one system fails in which are one part of data is stored, the other 2 systems can be used to retrieve the data.

The user always interacts with name node but actually it is stored in Data node.

Name node should have
Ram, CPU ↑ and HDD ↓

Data node should have
Ram, CPU ↓ and HDD ↑

TO start HDFS we need
Java 1.8

yum → To Download Software
rpm → To Install Downloaded Software.

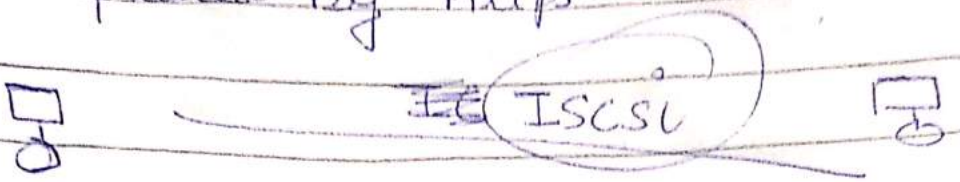
rpm -ql jdk → Software name
↳ tells all the files in the Software

PAGE NO.:
DATE: / /

pip3 install awscli
pip3 install boto3
aws
aws help
aws configure
Access key
Access ID → from Instance
default region → json
yum install ansible
ansible-doc
cd ansible/
aws ec2 help
aws ec2 describe-images
aws ec2 describe-images > image

PAGE NO.:
DATE: / /

But now NFS and CIFS are replaced by https.



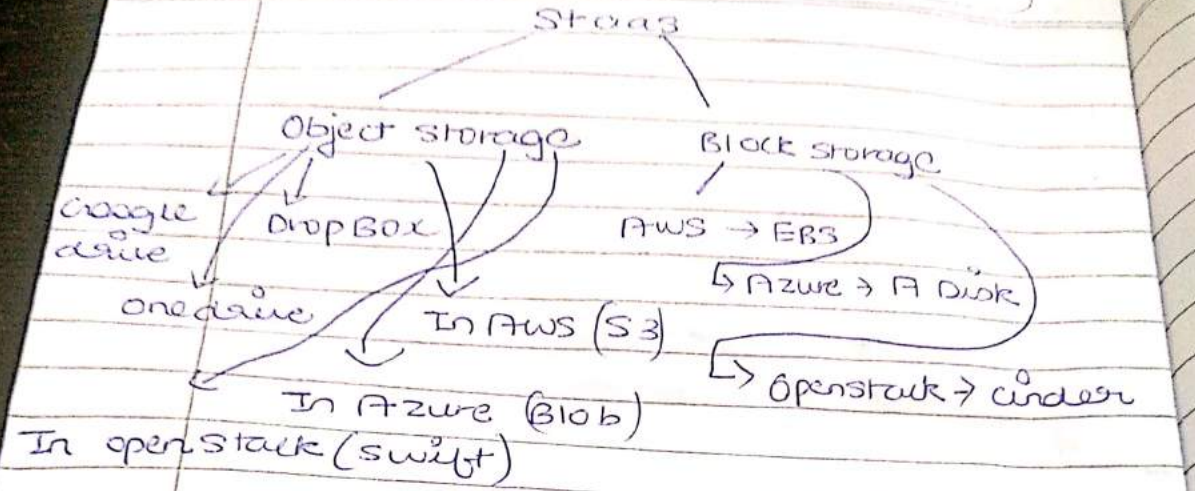
How will OS come to know that we connected the Harddisk or any perdrive to the system

It is done by SCSI
(Small computer system Interface)

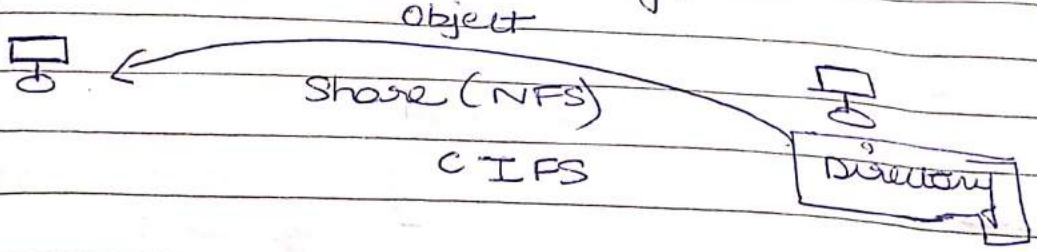
Basically SCSI is a protocol which helps to detect the pd or HDD of the same OS system.

ISCSI → Internet Small computer System Interface
↳ when we connect pd to one system and want to use on another system. (connect over Internet)

FCOE → Fiber channel over ethernet
↳ This also same as ~~IOS~~ ISCSI but it does through a ethernet

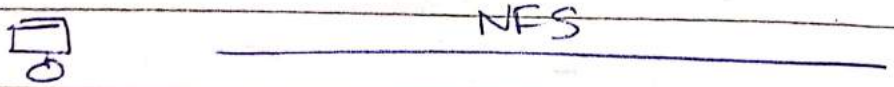


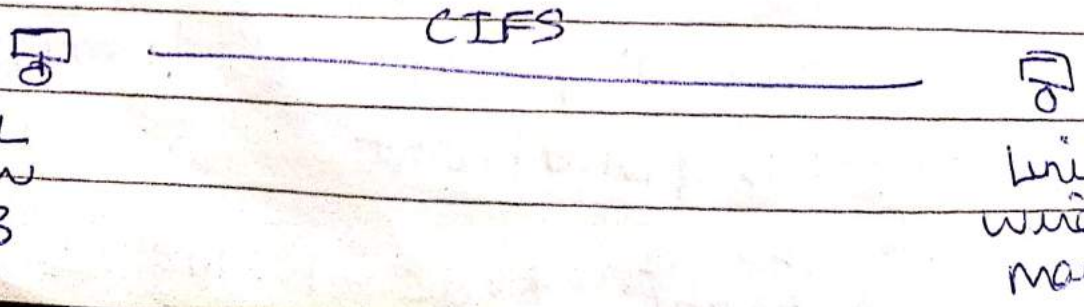
EBS (Elastic Block storage) →



NFS → Only able to share file when both the systems have same OS.

CIFS → able to share file when both the systems have any OS.





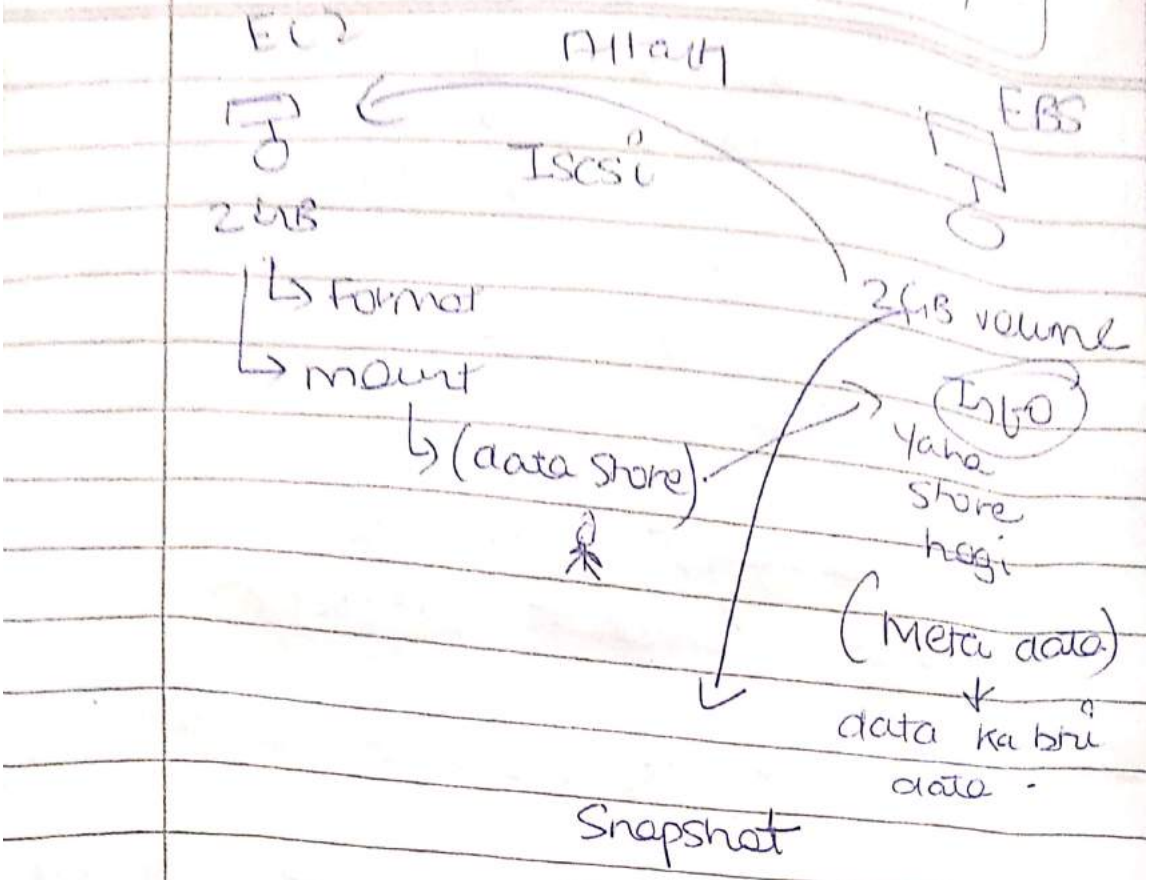
launch an instance
Install ansible
ansible localhost -m ping
cd /etc/ansible
ls
mkdir playbooks
ls
cd playbooks/
pwd

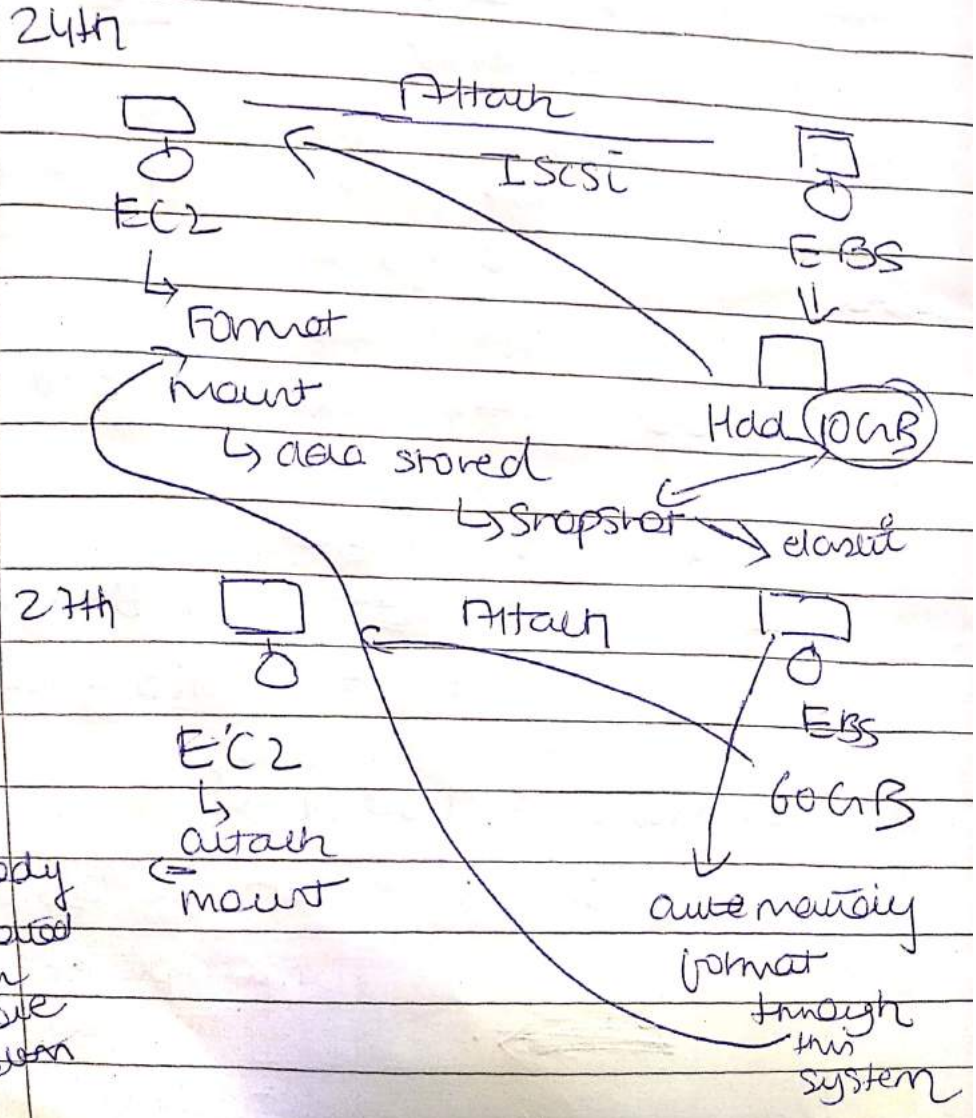
vi list_hdd.yml
↳ ---
- hosts: localhost
tasks:
- name: listing all
attached hdd
command: lsblk
register: x

- name: printing value of x
debug: var=x

ansible-playBooks list_hdd.yml
fdisk -l (select hdd)
Mkfs.xfs hdd name (/dev/xvdf)
Mkdir /mnt/part
mount /dev/xvdf /mnt/part
cd /mnt/part
ls
Mkdir hello {1..10} → ls

PAGE NO.:
DATE: / /


EC2
Attach
EBS
Iscsi
2GB
Format
mount
(data store)
2GB volume
Info
Yaha store hogi
(Meta data)
data ka bhi data.
Snapshot


24th
Attach
Iscsi
EC2
EBS
Format
mount
Hdd
10GB
data stored
Snapshot
elastic
27th
Attach
EBS
EC2
60GB
attach
mount
already formated from above system
automatically format through this system

PAGE NO.:
DATE: / /

Create snapshot
↳ 4GiB volume
↳ availability zone same

attach volume
↳

unmount old hdd
unmount /mnt/par

Then ~~mount /dev/xvdg~~

Then mount /dev/xvdg /mnt/new
LS /mnt/new
↳ The data in earlier hdd is transferred to new one)

EBS → Exam | Interview tips

→ Block storage
→ min → ~~1GiB~~ 1MB = 1000KB
1GiB 1mib = 1024KiB
→ max = 16384GiB
→ can we attach one EBS in multiple ec2
~~16TB~~

→ NO need to do

If we need one EBS more than 16 TB then while creating volume go on supports.

OR

limits → Request limit increase → automatically it will convert to aws support → choose second option → personal Information → choose call.

→ Encryption → if you want to put password on your pd or had, so that no one can read data..

→ KMS → Key management System

cd /etc/ansible
ls
cd playbooks/
ls
& yum install python3*

pip3 install awscli Boto3

PAGE NO.:
DATE: / /

To start a instance

```
---
- hosts: Localhost
  tasks:
   -ec2:
     image: "ami id" (found near instance)
     region: ap-south-1
     key_name: ansible
     instance_type: t2.micro
     wait: no
     count: 1
```

To close/stop a instance

```
---
- hosts: Localhost
  tasks:
   -ec2:
     instance ids:
     state: stopped
     region: ap-south-1
```

ansible-playBOOK launchinstance.yml

yum install python2-pip

rpm -ivh python2 (+tab)

wget http:// ___url___ python2-pip

pip install Boto

ansible-doc ec2

aws ec2 describe-key-pairs.

pip3 install awscli
pip3 install boto3
aws help
aws configure
↳ Acess Key
↳ Secret acess Key
↳ Refaunt region
↳ json / text / table

yum install ansible
ansible-doc

cd ansible/
aws ec2 help
aws ec2 describe-images
aws ec2 describe-instances

← vim launch-instance-yml
no gap

file will open

- hosts: localhost
  tasks:
  - ec2:

space

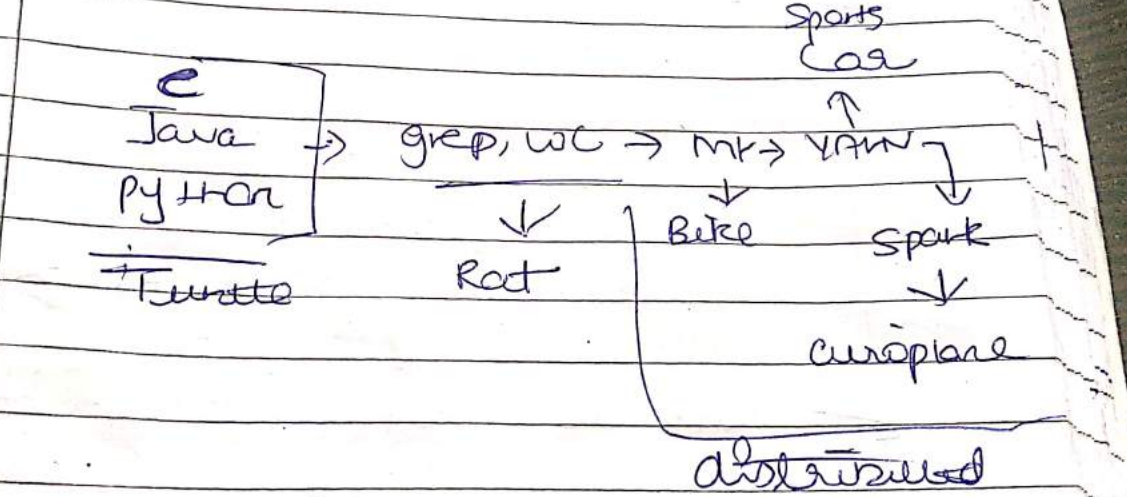
PAGE NO.:
DATE: / /

YARN → Yet another Resource Negotiator

Map reduce → distributed processing engine

↳ Bahut sare systems p ek file k bahar sare parts ko merge karega same code run karke then unki sum it and shows as one -

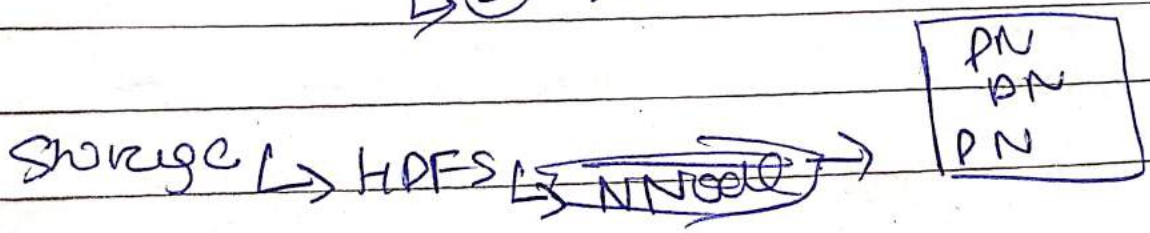
Pro-speed



Map reduce components

↳ ① Job Tracker

↳ ② Task Tracker



Yarn components →

1. Resource manager
2. Node manager

cd /etc/sysconfig/network-scripts/
vi

ifcfg-ens33 or wlp--- ← press Tab and open the file

BOOTPROTO → static
ONBOOT → yes
IPADDR → 192.168.10.55
↳ the IP which we want to be static

GATEWAY →
~~Gateway~~ → 192.168.10.1
DNS → 192.168.10.254
NETMASK → 255.255.255.0
↳ 2^8

route -n → to calculate

gateway server name

vi /etc/chrony.conf → server (adhac.example.com)

systemctl restart chronyd

chronyc

chronyc serverstats

chronyc servers

chronyc tracking

chronyc sources/ sourcestats

S3

Create a Bucket.

free -m → to see RAM

rhel 7 → yum install NTP

lsattr → to check if chattr is being used or not.

NTP → Network time protocol

rhel 8 only →
yum install chrpath.X86_64
yum install ccony
↓
systemctl restart chronyd-
" enable " service

rpm -qc chrony

PAGE NO.
DATE : / /

RHCSA Exam

Shell
Input and output redirection
tar and compression
NTP client
Partition, LVM
permissions
stratis
vdo → Virtual Disk Optimiser
Tuning (user-management)
Yum and Rpm
Password Break
IP static
crontab

vi /etc/resolv.conf
↳ name server 192.168.10.254 rhel7

chattr +i (filename)
chattr +i /etc/resolv.conf
↳ just an eg.
↳ we can't change anything in the file after running this command

man chattr
↳ defines the meaning of command

PAGE NO.:
DATE: / /

To display any file on Browser
first → security group →
Inbound → edit → http
→ anywhere

```
<Html>
<head>
<Title> ADHOC </Title>
</Head>
<body>
<h1> hello world </H1>
<p> Hii Html this is <b> Apana </b>
</p>
<p> Hello all </p>
<marquee> send email </marquee>
</Body>
</Html>
```

To insert image

```
<Html>

<h2> this is page 2

<a href = "https://www.google.
com"> click here, </a>
<a href = "index.html"> town here
</a>
```

PAGE NO.
DATE : / /

Agenda
↳ HTML → version 5
↳ CSS → version 3
↳ python → CGI
↳ Interface → web → Desktop
Server

HTML → It is a documentation type language.

HTML is a webpage → (MS office Document)
↓ ↓
• Html files • DOCS
• XLS

www.g.c
↳ DNS

DVWA → some concept of hacking

yum install httpd -y

systemctl start httpd
" enable httpd
" status "

cd /var/www/html
ls
pwd
↳ vi index.html

PAGE NO.:
DATE: / /

<meta> → auto refresh
go to next page

<!-- comment -->

google.com

KYO or Kaine
↳ framework

<input> → takes input from user.

user → input → data → backend → process
↑ output

telnet gogle.com 80

Data visualisation →

matplotlib
(SeaBorn) → now on this
gplot
mpld3 as this can be easily integrated with pandas

instance right click
↓
Image
↓
create image
↓
Image name
↓
create

check is AMIs

Image is used to transfer data from one instance to new one if we have lost our key.

Data Analytics / visualisation

→ CVS, XLS, Text → common type
→ Data visualize

↓

Pandas → Converts different types of data files is a common file so that we can access it easily.

Pandas is python Based library

PAGE NO.:
DATE: / /

Cloud
=
→ EC2, AMI, EBS, ELB, ECS

mkfs ki jagah mkswap

mkswap /dev/vde

mount ki jagah swapon /dev/vde

swapoff to off swap

blkid /dev/vde
lsblk --output=UUID /dev/vde
↓ ↳ To know UUID
Then in vi /etc/fstab

put UUID swap swap
defaults 0 0

swapon -a

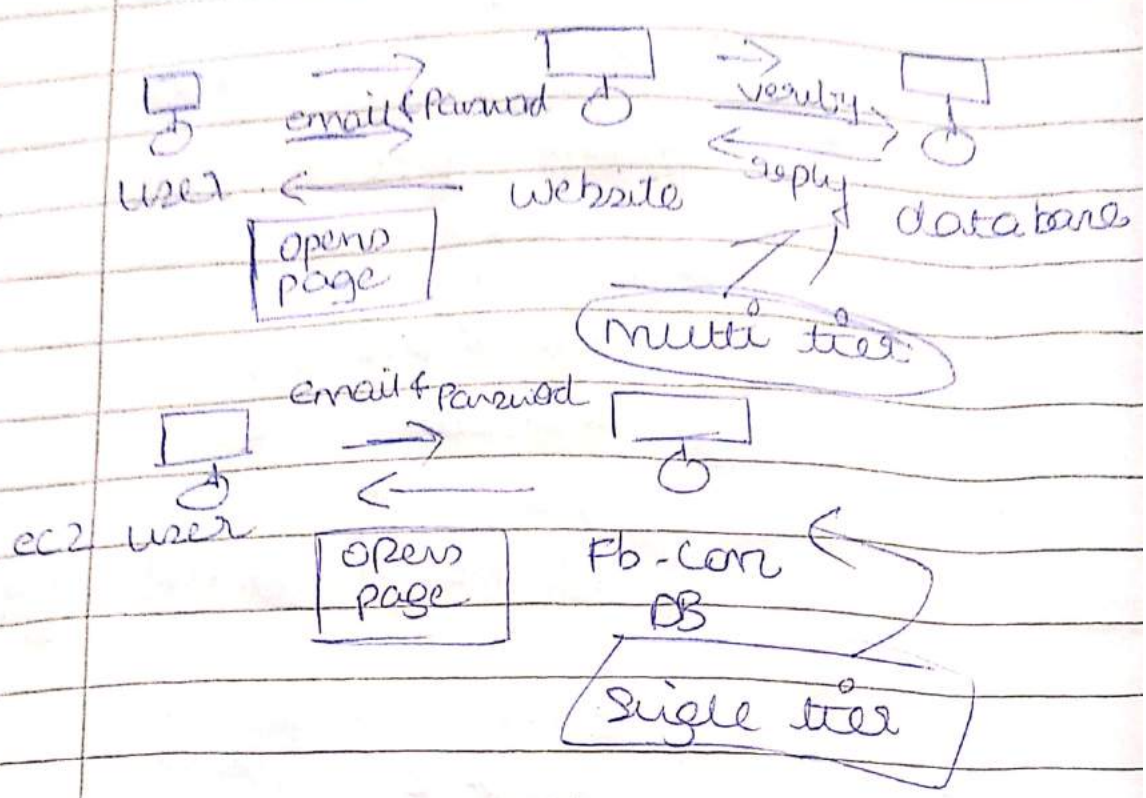
To extend LV
Forg → we gave LV 512M & want to add 100M more

↩ lvextend --size +100M /dev/ ——
XFS — growfs
↓
To format extended LV

↓
will format itself also and extend also

PAGE NO. :
DATE : / /

SQL



yum install mysql

mysql -u root -h ——— copy from database
-p

Python → mysql-connector
Java → JDBC / ODBC
Php → mysqli

mysql
oracle

vi dbconn.py

```
#!/usr/bin/python3
import mysql.connector as mysql
# RDS info
username = 'root'
password = '____'
db = 'Adhoc'
host = 'Endpoint url done hai'
# Now connecting
mysql.connect()
conn =
mysql.connect(user=u, password=p,
database=db, host=host)
# now generating a SQL lang cursor
cur = conn.cursor()
# Now we can write SQL query
cur.execute("show tables;")
#Now printing result
for i in cur.fetchall():
    print(i)
# closing conn
conn.close()
```

PAGE NO :
DATE : / /

1. Create the stratis pool labpool containing the block device /dev/vdb
2. Expand the capacity of labpool using the disk /dev/vdc available in the system.
3. Create a thinly provisioned file system named labfs in the labpool pool. mount this file system on /labStratisvol so that it persists across reboots. create a file named labfile1 that contains the text Hello world! on the labfs file system.
4. Verify that the thinly provisioned file system labfs dynamically grows as the data on the file system grows.
5. Create a snapshot named labfs-snap of the labfs file system. The snapshot allows you to open any file that is deleted from labfs.

PAGE NO.:
DATE: / /

Tuning

yum install tuned -y
systemctl start tuned
systemctl enable tuned
tuned-adm list
tuned-adm active
tuned-adm profile
tuned-adm profile balanced
tuned-adm active
tuned-adm recommend
tuned-adm off
tuned-adm active

systemctl start cockpit.socket
systemctl enable --now cockpit.socket

tuned-adm verify

ps -xl → Shows the performances of different services

nice → -20 or 19
↓ ↓
~~highest~~ Lowest
highest performance
performance → to change current nice
service → ~~current performance~~
next page

PAGE NO.
DATE : / /

① yum install stratis
② yum install stratis-cli

systemctl stratisd.service
systemctl enable stratisd.service

stratis --help

stratis pool create test1 /dev/vdc
stratis pool list

stratis filesystem create test1 eqtest1
pool name
↳ file system that we want to make

stratis filesystem list.

mkdir /fs1

mount /stratis/test1/eqtest1 /fs1

df-hT

---

ps -xu | grep vim
service -n 10 6689
ps -o pid, comm, nice

PAGE NO. :
DATE : / /

ps - aux --sort=pcpu

nice -n -5 vim .bashrc &

ps - xl | grep vim

kill 6689

kill -9 6689

↳ forcefully

~~vdovdm~~

~~vdevdm~~ settle

↳ used to edit the existing partition

vdo

yum install vdo kmod-kvdo

vdo

vdo create --name=vdo1

--device=/dev/vdd1

--vdoLogicalSize=50G