1. **Services in Linux**
2. **Package Management and Process Monitoring.**
3. **Communication commands**

mail

configure (install) smtp server

mailid, subject, email body. ctrl+D

1. **Miscellaneous commands**

**Video-6**

Find / -type f -empty

What is Load average and interval time?

Who

W

Users

Uptime

Date

Whereis

Whoami

Man

Ifconfig

Info

Service

Systemctl

Service sshd start

Status sshd. service

Last

Ps => process id proc/ directory

Ps -ef

Kill

Top

Sar (System activity report): used to collect the CPU Memory and I/O usage

**Video-7**

Archive/Data backup commands

Zip

Unzip

Tar

User and group administration related commands: -

Uid

Useradd

RHEL 7.x onwards UID starts from 1000

Before that version starts from 500

Usermod L

Usermod l

Usermod g

Vi /etc/sudoers

Sudo groupadd dba

lid

**Video-8**

Cron: - internally uses cron service

**Service crond. service status**

How to give crontab access to ec2-user?

**Video-9**

scp - = secure copy (across servers)

Connect to the server using WinSCP and FileZilla?

Password less ssh between 2 AWS instances using SSH keys?

AWK and cut command?

For a file to transfer

Scpfriends.txt ec2-user@IP: /tmp/

For a directory  
scp -r DevOps/ ec2-user@IP: /tmp/

Free

Free – h

Free -k

Free -g

Swap memory

Hardware information

Dmidecode

Dmidecode

**Communication commands**

mail

configure (install) SMTP server

mailid, subject, email body. ctrl+D

**protocols port numbers**

POP3

HTTP-80

HTTPS -443

FTP-21

SSH-22

**Other commands**

Cal – calender

Cal -3 => displays prev current next month

Cal -1 => current month

Cal – 2021 => full year

Cal nov 2021

**Wget**

The non-interactive network downloader

Wget link

Curl -o maven.zip link

Ls | tee teeopt.txt

Cat teeopt.txt

Script => creates a file called typescript

Date

Hostname

Ifconfig

Cat typescript

Script vinod.txt

Script -a viond.txt (append)

Cat vinod.txt

Ping

telnet

(Sudo yum install telnet -y)

telnet google.com 80

telnet google.com 8080

history

**uname**

**os-release**

**uname -a**

**cat /etc/\*release**

**netstat -atunlp**

**if you wants to know which port using**

**a all**

**t acp**

**u udp**

**n port number**

**l listening port**

**p port**

**sudo netstat -atunlp**

**need to execute date command for every 5 seconds**

**watch date (by default every 5 secs).**

**Watch -n 5 date**

**Exit**

**Ctrl+d**

**Sudo shutdown**

**Date**

**cat - used to type out a file (or combine files)**

**head - display first few lines**

**tail - display last few lines**

**man - view documentation.**

**head**

**tail**

**less =>page by page**

**more => percentage wise.**

**cut => based on delimiter this will filter the content in file.**

**sort =>sort in alphabetical order (ascending to descending)**

**sed=> stream editor. can search and replace the content**

**p->print**

**n->no duplicates**

**-e => multiple search items.**

**odd numbered lines**

**even numbered lines**

**print 2 lines after some content**

**print 6 lines before some content**

**d => deletes lines**

**between line to line.**

**sed 's/linux/DevOps/g'test.txt => g-> global means all occurrences, if you don’t use it only first occurrences change.**

**-i => insert the content.**

**grep and awk:**

**-----------**

**grep=>Global regular expression**

**Crontab or Cron service**

**Crontab: work repetitively or schedule the job.**

**Cron: Cron is a deamon service that executes schedules commands. Cron also reads /etc/crontab**

**.**

**Crontab: Crontab is the program used to install, deinsatll or list the tables used to drive the cron daemon in Vixie cron .**

**# Minute Hour DayOfMonth Month DayOfWeek Commnad / Script**

**# (0-59) ( 0-23) (1-31) (1-12 or Jan-Dec) (0-6 or Sun-Sat) ( usr/bin/find)**

1. 0-59

2. 0-23

3. 1-31

4. 1-12

5. 0-7

// checks whether cron service is running or not .

**# service crond.service status**

# ls -l

**Hello.sh**

**Files Descriptors:**

**0 => std i/p**

**1 => std o/p**

**2 => std err**

**Crontab -l => list all the ejob in crontab.**

**Crontab -e =>**

**Crontab -r =>**

**/etc/cron.allow** =>

Hello.sh

Echo “Hello How are you guys ….”

Echo “GM/GA/GE”

Echo “Today date is : ”

date :

:wq

$ ./hello.sh

$ ls -l

Hello.sh

This file doesn’t have execute permissions.

$ chmod u+x hello.sh

$ ./hello.sh

$ clear

Crontab means crontable.

**What is the crontab and can you tell the format?**

**And what are the possible values for each field?**

MHDMDC

**video- 5 bhanu**

**RPM VS YUM**

****

Normally packages are available for Redhat in the form of **.rpm**

Download from internet and install rpm packages.

Repository means collection of packages.

1. Rpm and 2. Yum

These 2 are package managers.

Sudo – i

# ls

// if any rpm packages available then you can install or you go for yum,

// yum is advanced.

# rpm -qa // this command lists the installed packages in the system

# rpm -q wget // to verify the wget is installed or not

# rpm -q vim

Package vim is not installed

Where we have stored our Redhat package managers?

Download package manager

# yum install wget -y

# rpm -ivh vim-minimal-8.2.274.X.xxx.rpm –nodeps

# rm \*.rpm

# rpm -ivh nano \*.rpm

# rpm -evh nanao\*.rpm

YUM is advanced package manager in redhat.

APT-GET is advanced package manager in Ubuntu.

# cd /etc/yum.repos.d/

**yum.repos.d # ls**

radhat-rhui-client.repo redhat-rhui.repo

# yum repolist

# yum installed vim

# yum installed list

// this lists out the installled packages.

//yum automatically install dependencies.

# yum info tomcat

# yum info tomcat9

# yum info jenkins

// Third party tools not installed

# systemctl start httpd

# systemctl status httpd

// deployment path to httpd

# cp index.html /var/www/html

# yum install jenkins

**DNF instead yum command.**

# dnf remove httpd

# dnf groupinstall ‘development tools’

# dnf groupremove ‘development tools’

# yum groupinfo development tools

**Softwares available in the format of zip or tar.gz**

Binaries available in the form of zip , tar.gz

We can extract **tar.gz** using tar command.

# wget \*\*\*\*.tar.gz

# ls

# tar xzvf \*\*.tar.gz

# yum install unzip -y

**Create tar file**

# tar -cvf dir1.tar dir1

# rm -r dir1

# rm -rf dir1

Dir1.tar available now

**Extract tar files**

# tar xvf dir1.tar

* Not required while extracting.

**Create tar.gz**

# tar cvfz dir1.tar.gz dir1 // - not required

# ls

**COPY**

# cp dir1.tar.gz /home/ec2-user/dir1.tar

**Video 6 bhanu**

Create 2 redhat instances and learn how to interact with them.

Ssh

Ssh-keygen

**Consider we have 2 servers.**

**How to communicate from one server to another server.**

**there are 2 ways**

**1. keypair mechanism**

**2. password mechanism**

**1. keypair mechanism**

ssh is by default installed in every OS.

$ ssh

$ ssh-keygen

your identification (private key) stored here

/home/ec2-user/.ssh/id\_rsa

public key stored here

/home/ec2-user/.ssh/id\_rsa.pub

# share public key with another system

$ cat

$ mkdir .ssh

$ ls

.ssh $ vi authorized keys

in this file you can add as many as public keys.

$ ssh user@ip

copy my public key in your system, then i can communicate with you.

how to check memory?

$ ssh XXX free -mh

@ ssh XXX df -h

Using ssh we can communicate with remote servers.

Public key sharing is done through user level so u need to copy each and every user separately.

Create a new user at server2 and create a directory .ssh and file authorised\_keys

You have to update permissions at server2 of .ssh/authorised\_keys file.

now login to server2 from server1.

**2. password mechanism**

go to server2 open the sshd\_config file as root user.

# etc/ssh/sshd\_config

Enable PasswordAuthentication = yes.

Now you can login from server1

**How to setup password less mechanism?**

You can generate a ssh key to remote machine using the following command.

ssh-copy-id bhanu@18.188.230.232

scp => secure copy to copy from one server to another server

scp file100 ramesh@18.188.230.232:/home/ramesh

scp in forward and backward direction

====

Just replace scp with rsync

**What is the diff between scp and rsync?**

scp copies the data everytime from beginning, rsync is feasible when more data.

**Viewing Compressed Files**

**When working with compressed files, many standard commands cannot be used directly. For many commonly-used file and text manipulation programs, there is also a version especially designed to work directly with compressed files. These associated utilities have the letter "z" prefixed to their name. For example, we have utility programs such as zcat, zless, zdiff and zgrep.**

**Here is a table listing some z family commands:**

**$ zcat compressed-file.txt.gz** => To view a compressed file

**$ zless somefile.gz**  
or  
**$ zmore somefile.gz =>** To page through a compressed file

**$ zgrep -i less somefile.gz =>** To search inside a compressed file

**$ zdiff file1.txt.gz file2.txt.gz =>** To compare two compressed files

**Note that if you run zless on an uncompressed file, it will still work and ignore the decompression stage. There are also equivalent utility programs for other compression methods besides gzip, for example, we have bzcat and bzless associated with bzip2, and xzcat and xzless associated with xz.**