

Date: 25 July 2025

Data Backup

--Santosh Kyadari, Computer Centre

Backup

A copy of data created and stored separately from the original to protect against data loss

- Backup the system before doing major changes
- Ensure recovery of major losses
- Take backups periodically
- Mock drills of restore points
- Adhere to the backup policies of the organization
- Should prevent Ransomware attacks

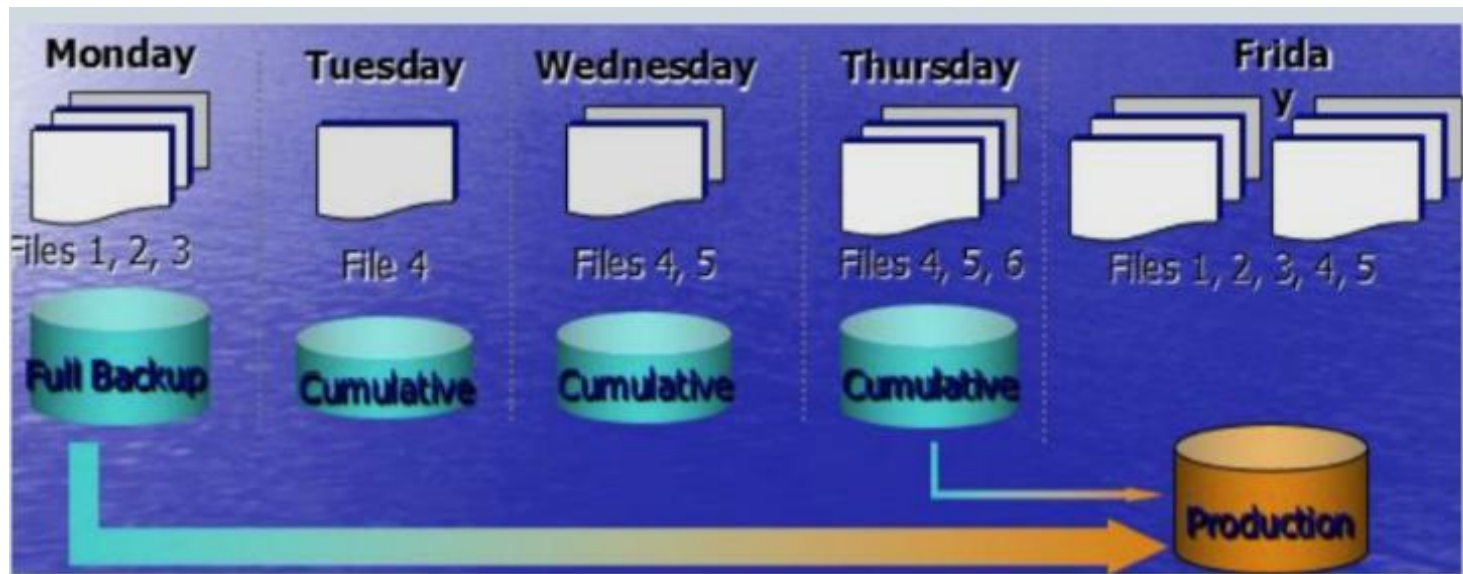
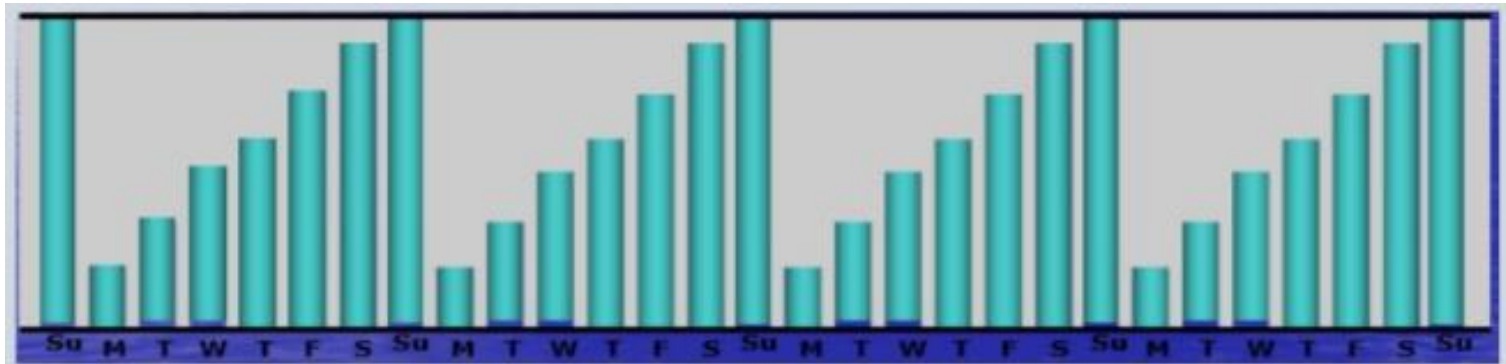
Different Kinds of backup

Full backup

Copies all selected data, regardless of whether it has changed since the last backup.

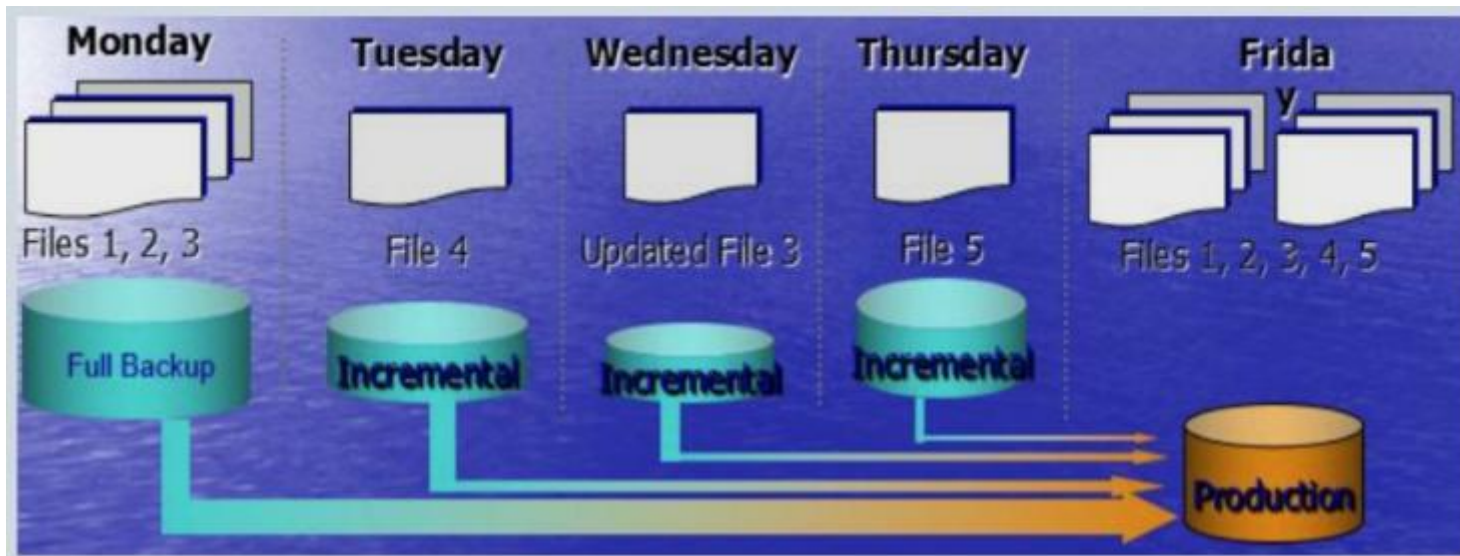
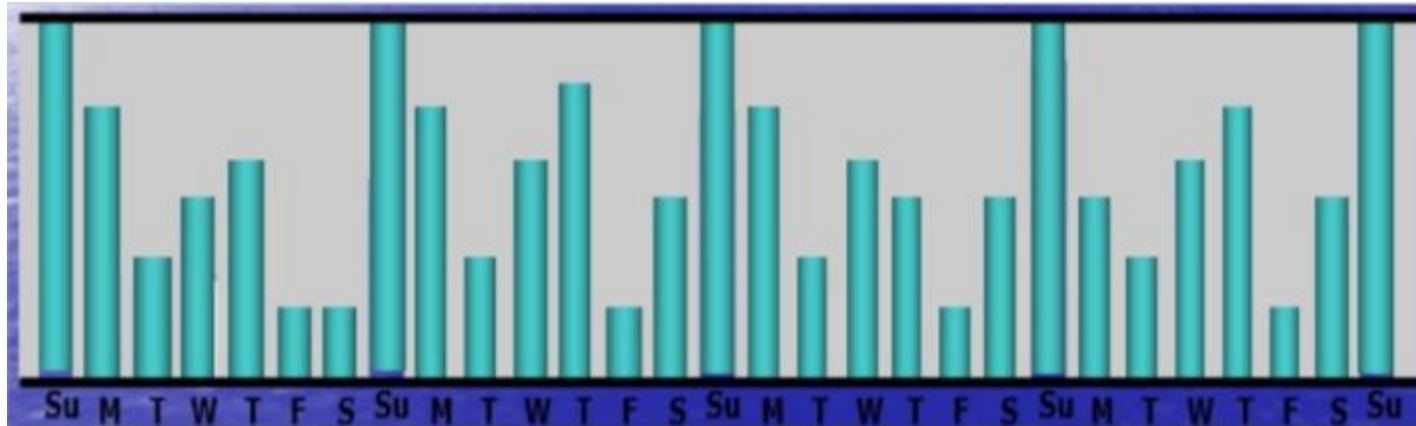
Differential Backup

Backs up data that has changed since the last full backup. It takes longer than incremental backups but requires fewer restore steps.

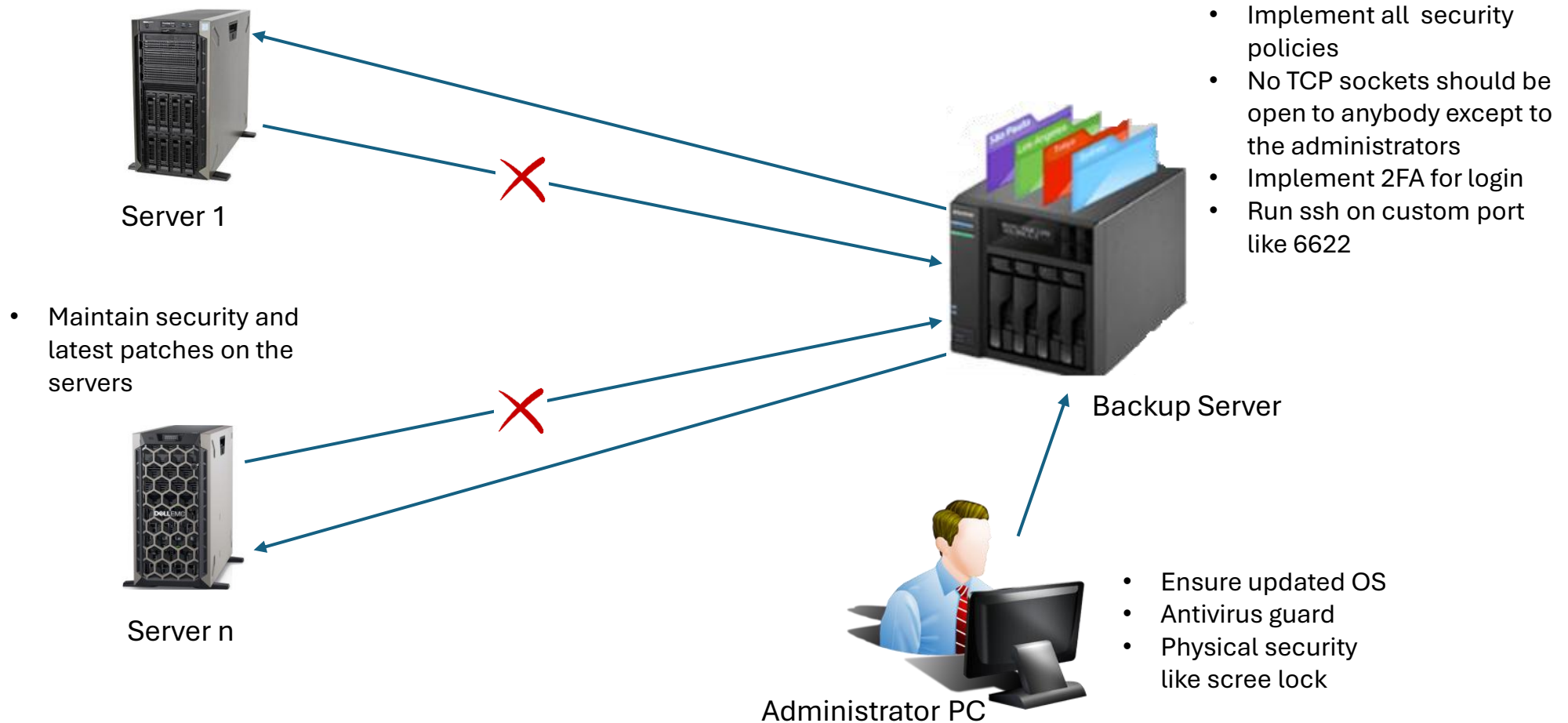


Incremental Backup

Backs up only the data that has changed since the last full or incremental backup, saving time and storage space.



Basic Block diagram of Backup network



ssh configuration for client server

In /etc/ssh/**sshd_config** of server

#To login from backup server

Match Address <IP address of Backup server>

PermitRootLogin yes

Password less login from backup server

ssh-keygen -t rsa # It will create id_rsa and id_rsa.pub keys in ~/.ssh

ssh-copy-id root@server.tifr.res.in

Comparison

Commercial Backup Solutions

- Veritas NetBackup
- Veem
- Comvault
- Acronis

Open Source Backup Solutions

- Tar
- Dump
- Rsync
- Rsnapshot
- Bacula
- Amenda

Commercial	Open Source
Costly	Free
Data migration painful	Easy
If OEM changes, needs to learn	Once learned
AMC / Subscription renewals	No Pain
Limit of data size	Unlimited

Conventional backup commands in linux

```
tar -czvf backup.tar.gz /etc
```

```
tar -xzvf backup.tar.gz
```

```
ssh root@host "cd /etc && tar czf - ." > /backup/backup.tar.gz
```

```
ssh root@host "tar czf - /etc " > /backup/backup.tar.gz
```

```
dump -0vf backup.dump /etc
```

```
restore -i -f backup.dump
```

```
restore -x -f backup.dump /tmp
```

```
rsync -av /home -e ssh root@test.tifr.res.in:/backup/.
```

```
rsync -av --delete --rsh=/usr/bin/ssh root@test.tifr.res.in:/home/ /backup/
```


Mysql backup commands in linux

```
mysqldump --all-databases > all_databases.sql
```

```
mysqldump -u root -p[root_password] [database_name] > dumpfilename.sql
```

```
mysql -u root -p[root_password] [database_name] < dumpfilename.sql
```

Rsnapshot

rsnapshot creates incremental backups using rsync and hard links. This ensures that only changes are transferred, and the hard links allow for space-saving, multi-level backups

```
rsnapshot -v -c rsnapshot.conf daily|weekly|monthly
```

Contents of `rsnapshot.conf`

```
snapshot_root /ccdata/backup/mh/
```

```
retain daily 6
```

```
retain weekly 4
```

```
retain monthly 12
```

```
retain yearly 3
```

```
Logfile      /ccdata/backup/mh/rsnapshot.log
```

```
lockfile    /ccdata/backup/mh/rsnapshot.pid
```

```
backup root@mh.tifr.res.in:/etc/      mh/
```

```
backup root@mh.tifr.res.in:/var/log/   mh/
```

```
backup root@mh.tifr.res.in:/home/     mh/
```

```
backup root@mh.tifr.res.in:/uploads/  mh/
```

Consumed space on the backup server

3.0T	/ccdata/backup/mh/daily.0	drwxr-xr-x. 3 root root	Jul 18 06:16 daily.0
8.9G	/ccdata/backup/mh/daily.1	drwxr-xr-x. 3 root root	Jul 17 06:20 daily.1
16G	/ccdata/backup/mh/daily.2	drwxr-xr-x. 3 root root	Jul 16 06:20 daily.2
9.6G	/ccdata/backup/mh/daily.3	drwxr-xr-x. 3 root root	Jul 15 06:20 daily.3
13G	/ccdata/backup/mh/daily.4	drwxr-xr-x. 3 root root	Jul 14 06:03 daily.4
23G	/ccdata/backup/mh/daily.5	drwxr-xr-x. 3 root root	Jul 12 06:11 daily.5
21G	/ccdata/backup/mh/weekly.0	drwxr-xr-x. 3 root root	Jul 7 05:59 weekly.0
33G	/ccdata/backup/mh/weekly.1	drwxr-xr-x. 3 root root	Jun 28 06:01 weekly.1
11G	/ccdata/backup/mh/weekly.2	drwxr-xr-x. 3 root root	Jun 23 05:39 weekly.2
20G	/ccdata/backup/mh/weekly.3	drwxr-xr-x. 3 root root	Jun 16 05:39 weekly.3
128G	/ccdata/backup/mh/monthly.0	drwxr-xr-x. 3 root root	Jun 1 13:40 monthly.0

sample backup.sh script

```
#!/bin/bash
```

```
#Usage /ccdata/backup/test/backup.sh
```

```
#Contab 30 0 * * * /ccdata/backup/test/backup.sh
```

```
# \ >>/ccdata/backup/test/errors.log 2>&1
```

```
#set -e
```

```
#SCDIR=dirname ${echo $0}
```

```
#SCDIR=`dirname "$0"`
```

```
SCDIR=$(dirname "$0")
```

```
cd "$SCDIR"
```

```
#echo $SCDIR
```

```
SCNAME=$(echo "$SCDIR" | awk -F '/' '{print $NF}')
```

```
LOG="$SCDIR"/"$SCNAME".log
```

```
DATE=$(date +%Y-%m-%d)
```

```
# 0 is Sunday, 1 is Monday, ..., 6 is Saturday
```

```
DAY_OF_WEEK=$(date +%w)
```

```
DAY_OF_MONTH=$(date +%d)
```

```
# Function to create a backup
```

```
backup_create()
```

```
{
```

```
    local backup_type="$1"
```

```
    echo "***** $1 Backup started `date`" >> "$LOG"
```

```
    /usr/bin/rsnapshot -v -c "$SCDIR"/rsnapshot.conf "$1" >> "$LOG" 2>&1
```

```
    echo "***** $1 Backup Ended `date`" >> "$LOG"
```

```
    /usr/bin/du -sh "$SCDIR"/* >> "$LOG"
```

```
    echo "DU -sh END time `date`" >> "$LOG"
```

```
}
```

```
# Yearly backup (on January 1st)
```

```
if [ "$DAY_OF_MONTH" -eq 01 ] && [ "$(date +%m)" -eq 01 ]; then
```

```
    backup_create "yearly"
```

```
# Monthly backup (on the 1st of the month)
```

```
elif [ "$DAY_OF_MONTH" -eq 01 ]; then
```

```
    backup_create "monthly"
```

```
# Weekly backup (every Sunday)
```

```
elif [ "$DAY_OF_WEEK" -eq 0 ]; then
```

```
    backup_create "weekly"
```

```
# Daily backup
```

```
else
```

```
    backup_create "daily"
```

```
fi
```