**📘 Linux Backup & Recovery – Step by Step Notes**

**🔹 1. What is Backup & Recovery?**

* **Backup**: Copying data to a secure location for protection against data loss.
* **Recovery**: Restoring that data in case of failure, corruption, or accidental deletion.

**Types of Backup:**

1. **Full Backup** → Entire system/data copy.
2. **Incremental Backup** → Only changes since last backup.
3. **Differential Backup** → Changes since last full backup.
4. **Snapshot** → Point-in-time image (e.g., LVM snapshot, VM snapshot).

**🔹 2. Important Backup Tools in Linux**

* **rsync** → Sync files/directories.
* **tar** → Archive & compress data.
* **dd** → Disk/partition cloning.
* **dump/restore** → Traditional FS backup.
* **scp/rsync over SSH** → Remote backups.
* **Bacula / Amanda / Duplicity / Timeshift** → Enterprise solutions.
* **Cloud backup** → AWS S3, Azure Blob, GCP Storage.

**🔹 3. Backup Location**

* **Local** → External HDD, USB, other partitions.
* **Remote** → NFS, NAS, SAN, or SSH servers.
* **Cloud** → AWS, GCP, Azure.

**🔹 4. Pre-Backup Checklist**

✅ Check disk space with df -h.  
✅ Verify important mount points (/home, /var, /etc).  
✅ Stop critical services if consistent backup required (DB, Web apps).  
✅ Exclude unnecessary directories (/proc, /sys, /tmp).

**🔹 5. Backup Methods – Step by Step**

**(A) Using tar (Archive & Compress)**

# Full backup

tar -cvpzf /backup/full-backup-$(date +%F).tar.gz --exclude=/proc --exclude=/sys --exclude=/tmp /

# Extract backup

tar -xvpzf /backup/full-backup-2025-08-21.tar.gz -C /

**(B) Using rsync (Efficient backup)**

# Sync /home to backup server

rsync -avz /home/ user@backupserver:/backups/home/

# With progress & delete

rsync -avz --progress --delete /home/ user@backupserver:/backups/home/

**(C) Using dd (Disk/Partition Clone)**

# Clone disk

dd if=/dev/sda of=/dev/sdb bs=64K conv=noerror,sync

# Create image file

dd if=/dev/sda of=/backup/sda.img bs=64K

**(D) Database Backup (MySQL Example)**

# Backup all databases

mysqldump -u root -p --all-databases > /backup/db\_backup.sql

# Restore database

mysql -u root -p < /backup/db\_backup.sql

**(E) LVM Snapshot (Before risky operation)**

# Create snapshot

lvcreate --size 2G --snapshot --name my\_snap /dev/vg0/my\_lv

# Restore snapshot

lvconvert --merge /dev/vg0/my\_snap

**🔹 6. Recovery Process – Step by Step**

1. **Identify backup type** (full, incremental, snapshot).
2. **Check backup integrity** (e.g., tar -tzf backup.tar.gz).
3. **Unmount target partition** (to avoid conflicts).
4. **Restore files using tar/rsync/dd/mysql restore**.
5. **Verify restored files** (diff, md5sum).

**🔹 7. Automating Backups with Cron**

# Edit crontab

crontab -e

# Run daily backup at 2 AM

0 2 \* \* \* rsync -avz /home/ /backup/home-$(date +\%F)/

**🔹 8. Post-Backup Verification**

* Compare checksum:

md5sum original\_file backup\_file

* Ensure backup file size is consistent.
* Test restore on staging server (best practice).

**🔹 9. Interview Questions & Answers**

**Q1: Difference between Incremental & Differential backup?**

* Incremental → Only changes since last backup (faster, smaller).
* Differential → Changes since last full backup (larger, but easier restore).

**Q2: How do you backup a live database?**

* Use mysqldump --single-transaction, LVM snapshots, or DB replication.

**Q3: How do you exclude directories during backup?**

* Use --exclude with tar or rsync.

**Q4: How do you ensure backup integrity?**

* Verify with md5sum / sha256sum / tar -tzf.

**Q5: What’s the difference between rsync and scp?**

* rsync → Syncs only changes, efficient.
* scp → Copies everything every time.

**Q6: If backup failed, how do you troubleshoot?**

* Check logs (/var/log/), verify disk space, network connectivity, cron logs.