🛡️ Backup, Recovery & Automation

📘 Scenario Overview

After a server glitch caused by a power surge, leadership identified the absence of a formal backup strategy as a serious risk. As the **Backup Sentinel**, I was tasked with designing and implementing a **robust backup, verification, recovery, and automation process** to protect Project Phoenix's critical data.

This lab simulates **real-world disaster preparedness**, emphasizing reliability, validation, and unattended execution.

🎯 Objectives

- Identify and document critical data for backup
- Create compressed system backups using `tar`
- Verify backup integrity without extraction
- Restore a specific critical file from an archive
- Automate backups using `cron` with timestamped archives

🧾 Step 1: Identify Critical Data for Backup

Purpose

Limit backups to **high-value directories** to keep archives efficient and manageable.

Command Used

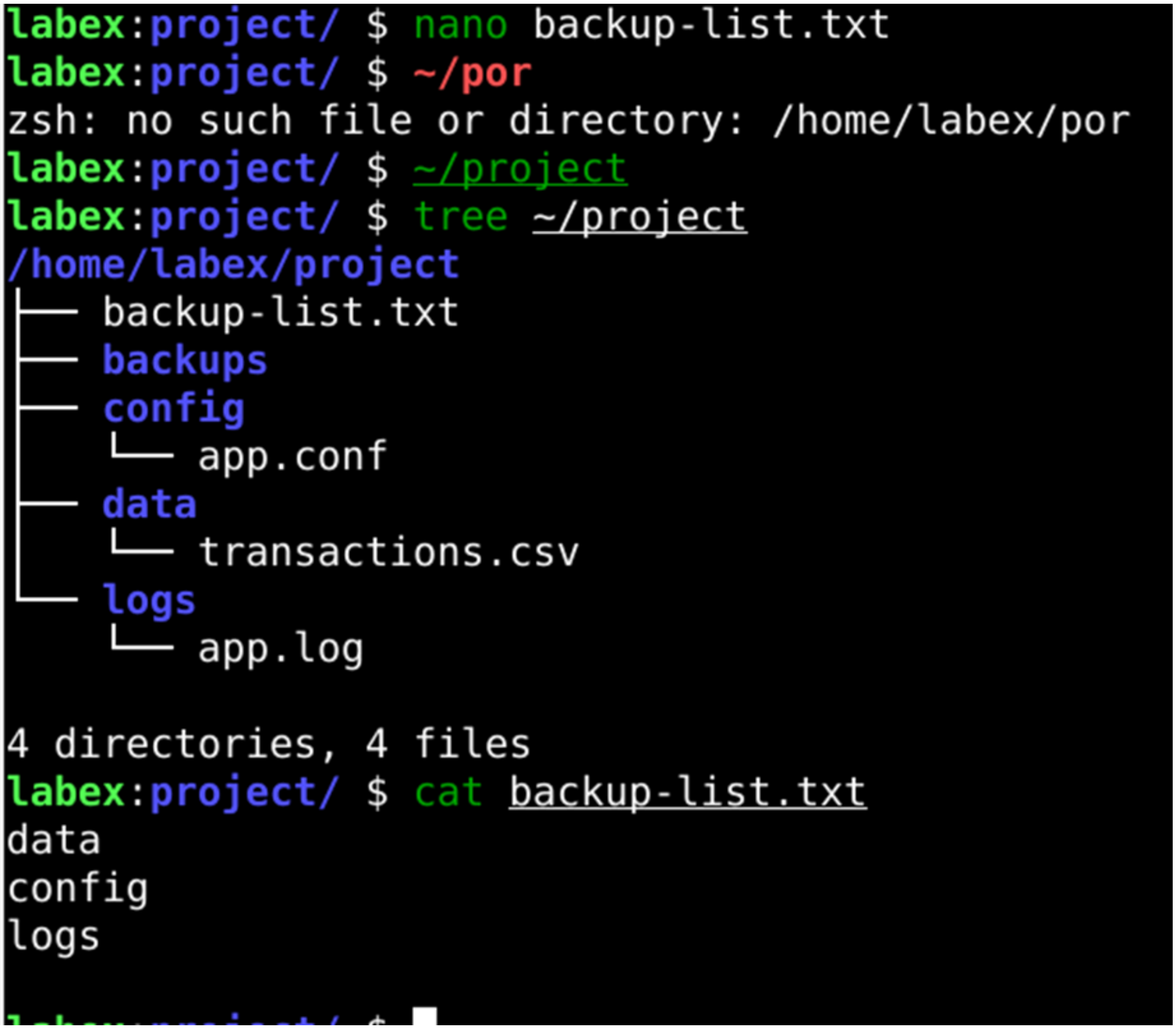
```
cd ~/project
printf "data\nconfig\nlogs\n" > backup-list.txt
```

Result

`backup-list.txt` contains:

```
data
config
logs
```

This file serves as a **backup manifest** for repeatable and auditable backups.



```
labex:project/ $ nano backup-list.txt
labex:project/ $ ~/por
zsh: no such file or directory: /home/labex/por
labex:project/ $ ~/project
labex:project/ $ tree ~/project
/home/labex/project
├── backup-list.txt
├── backups
├── config
│   └── app.conf
├── data
│   └── transactions.csv
└── logs
    └── app.log

4 directories, 4 files
labex:project/ $ cat backup-list.txt
data
config
logs

labex:project/ $ 
```

📦 Step 2: Create a Full Backup Archive

Purpose

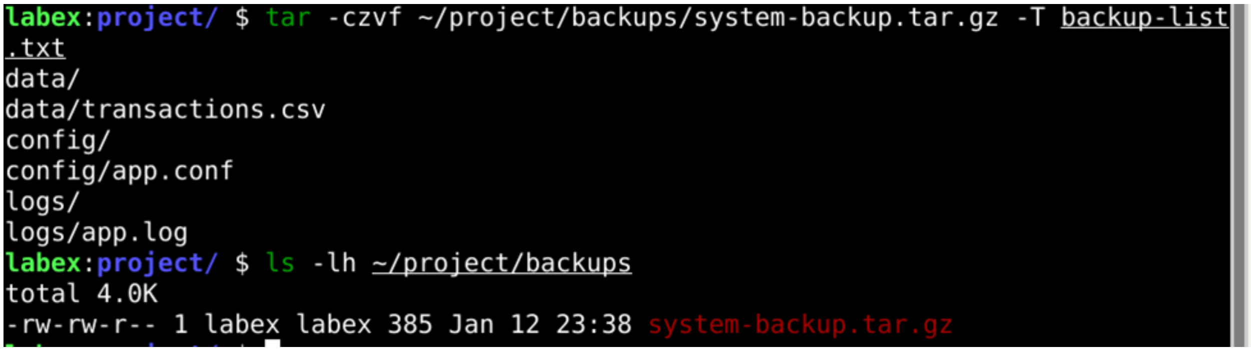
Bundle and compress all critical directories into a single archive.

Command Used

```
tar -czf ~/project/backups/system-backup.tar.gz -T
~/project/backup-list.txt
```

Outcome

Archive created at:
`~/project/backups/system-backup.tar.gz`

-
- Used `-T` to read sources from `backup-list.txt`
- Ensures consistency across future backups
- 

```
labex:project/ $ tar -czvf ~/project/backups/system-backup.tar.gz -T backup-list
.txt
data/
data/transactions.csv
config/
config/app.conf
logs/
logs/app.log
labex:project/ $ ls -lh ~/project/backups
total 4.0K
-rw-rw-r-- 1 labex labex 385 Jan 12 23:38 system-backup.tar.gz
```

🔍 Step 3: Verify Backup Integrity

Purpose

Confirm the backup is readable and contains the expected files.

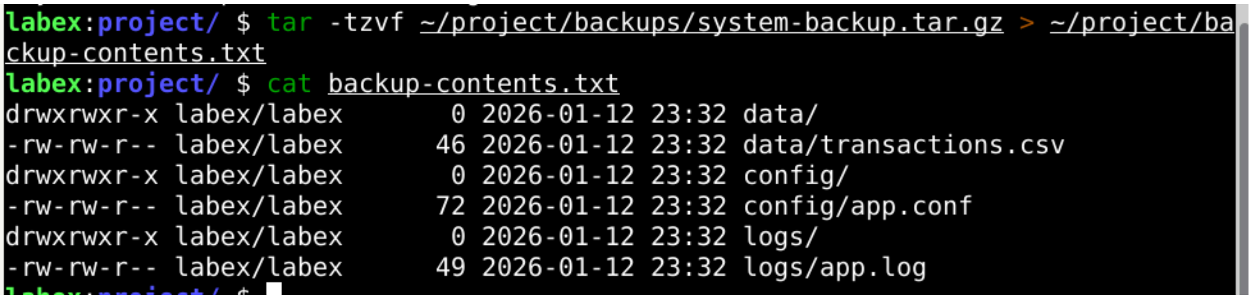
Command Used

```
tar -tzf ~/project/backups/system-backup.tar.gz >
~/project/backup-contents.txt
```

Result

- `backup-contents.txt` lists all archived files
- No extraction performed

- Confirms backup completeness and integrity



```
labex:project/ $ tar -tzvf ~/project/backups/system-backup.tar.gz > ~/project/backup-contents.txt
labex:project/ $ cat backup-contents.txt
drwxrwxr-x labex/labex       0 2026-01-12 23:32 data/
-rw-rw-r-- labex/labex      46 2026-01-12 23:32 data/transactions.csv
drwxrwxr-x labex/labex       0 2026-01-12 23:32 config/
-rw-rw-r-- labex/labex      72 2026-01-12 23:32 config/app.conf
drwxrwxr-x labex/labex       0 2026-01-12 23:32 logs/
-rw-rw-r-- labex/labex      49 2026-01-12 23:32 logs/app.log
labex:project/ $
```

♻️ Step 4: Restore a Critical File from Backup

Scenario

The application configuration file `config/app.conf` was accidentally deleted.

Simulate the Incident

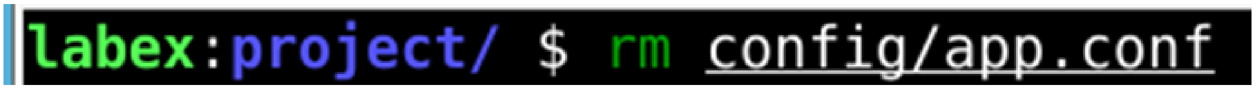
```
rm ~/project/config/app.conf
```

Restore Only the Required File

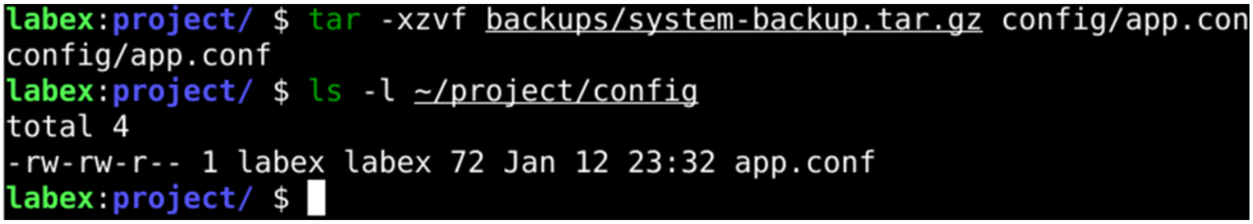
```
tar -xzf ~/project/backups/system-backup.tar.gz config/app.conf
```

Result

- `~/project/config/app.conf` successfully restored
- No unnecessary files extracted
- Application configuration recovered quickly



```
labex:project/ $ rm config/app.conf
```



```
labex:project/ $ tar -xzvf backups/system-backup.tar.gz config/app.con
config/app.conf
labex:project/ $ ls -l ~/project/config
total 4
-rw-rw-r-- 1 labex labex 72 Jan 12 23:32 app.conf
labex:project/ $ 
```

⏰ Step 5: Automate Backups with Cron

Purpose

Eliminate reliance on manual backups by scheduling automatic runs.

Edit Cron Jobs

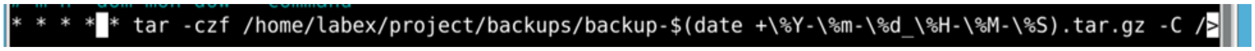
```
crontab -e
```

Cron Entry (Runs Every Minute)

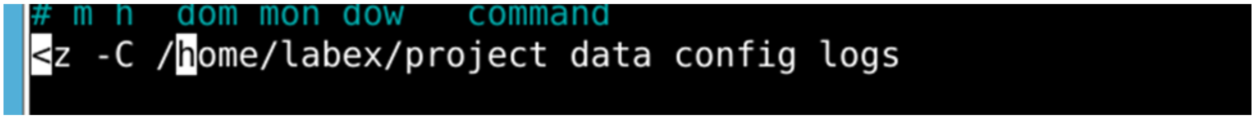
```
* * * * * tar -czf /home/labex/project/backups/backup-$(date
+\%F_\%H-\%M-\%S).tar.gz -T /home/labex/project/backup-list.txt
```

Key Features

- Runs automatically every minute
- Uses **absolute paths**
- Timestamped filenames prevent overwriting
- Fully unattended backup process



```
* * * * * tar -czf /home/labex/project/backups/backup-$(date +\%Y-\%m-\%d_\%H-\%M-\%S).tar.gz -C />
```



```
# m h  dom mon dow   command
<z -C /home/labex/project data config logs
```

🧠 Skills Demonstrated

- Backup planning and data prioritization

- Archive creation with `tar` and `gzip`
- Backup verification without extraction
- Targeted file restoration
- Cron job scheduling and automation
- Disaster recovery preparedness
- Operational resilience design