Part 8

Real-World Automation With Shell Scripts

Big Idea First (Mental Model)

Automation means:

> Do once manually → do forever automatically

If a task is:

- Repetitive
- Predictable
- Rule-based

It should be scripted.

SECTION 1: Automated Backup Script

Problem

Back up important data daily with a timestamp.

Script

```
#!/bin/bash

SOURCE="/home/user/data"
DEST="/home/user/backups"
DATE=$(date +%F)

tar -czf $DEST/backup_$DATE.tar.gz $SOURCE
```

What this script does

- Compresses data

- Names backup by date
- Prevents overwriting

Used in servers, DevOps, production systems.

SECTION 2: Log Cleanup Automation

Problem

Logs grow forever and eat disk space.

Script

```
#!/bin/bash

LOG_DIR="/var/log"
DAYS=7

find $LOG_DIR -type f -name "*.log" -mtime +$DAYS -delete
```

Key idea:

> Old files + rules = automation

SECTION 3: Disk Usage Monitoring

Problem

Alert when disk is almost full.

Script

```
#!/bin/bash

THRESHOLD=80
USAGE=$(df / | tail -1 | awk '{print $5}' | tr -d '%')

if [ $USAGE -ge $THRESHOLD ]; then
```

```
  echo "Disk usage is high: $USAGE%"
fi
```

Common in monitoring systems.

SECTION 4: Process Monitoring Script

Problem

Restart a service if it stops.

Script

```
#!/bin/bash

SERVICE="nginx"

if ! pgrep $SERVICE > /dev/null; then
  echo "$SERVICE is not running. Restarting..."
  systemctl restart $SERVICE
fi
```

Used in high-availability systems.

SECTION 5: User Activity Logging

Problem

Track who logs in and when.

Script

```
#!/bin/bash

echo "Login report on $(date)" >> logins.txt
who >> logins.txt
```

SECTION 6: Batch File Processing

Problem

Rename many files at once.

Script

```
#!/bin/bash

for file in *.txt; do
  mv "$file" "old_$file"
done
```

SECTION 7: Automating with `cron` (Scheduling)

What is `cron`?

> Linux scheduler that runs scripts automatically.

Edit cron jobs

```
crontab -e
```

Example: Daily backup at 2 AM

```
0 2 * * * /home/user/backup.sh
```

Field	Meaning
0	Minute
2	Hour
*	Day
*	Month
*	Weekday

SECTION 8: Automation Best Practices

Use variables

Add comments

Log output

Test manually first

Avoid hard-coded paths

SECTION 9: Common Production Mistakes

No logging

No error checking

Scripts without execute permission

Running as root unnecessarily

Professional rule:

> Automate carefully — scripts can break systems fast.