🧵 Bash Scripting & Automation

📘 Scenario Overview

As a Junior DevOps Engineer, I was tasked with addressing uncontrolled log growth on a production server. Manual cleanup was inefficient and error-prone, so the solution was to **design and implement a reusable Bash script** capable of validating directories, accepting user input, and automating file operations safely.

This lab simulates **real-world infrastructure automation**, where scripting replaces repetitive manual tasks.

🎯 Objectives

- Create and execute a Bash shell script
- Use variables to improve script flexibility
- Capture and process user input
- Implement error handling with conditional logic
- Automate file operations using loops

🧱 Step 1: Create a Basic Shell Script

Purpose

Establish a working script foundation and verify execution.

Commands Used

```
cd ~/project
nano log_manager.sh
```

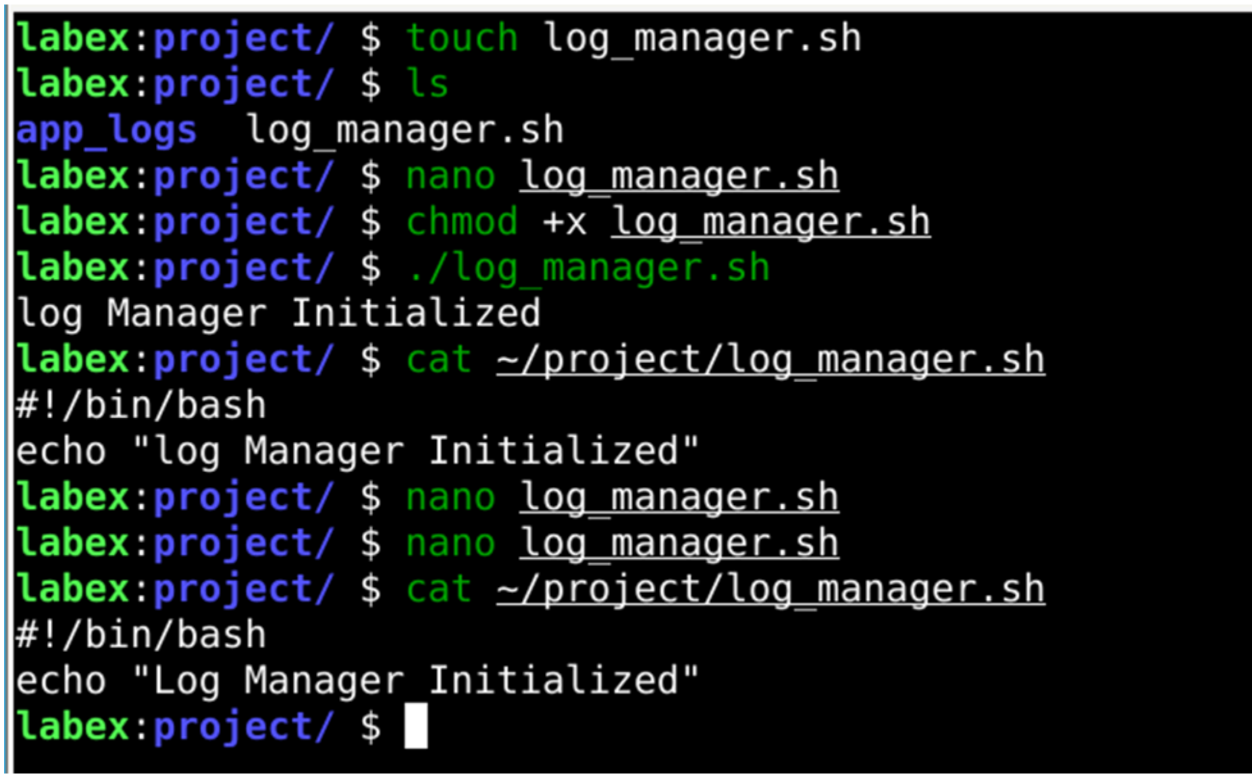
Script Contents

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

```
echo "Log Manager Initialized."
```

Make Script Executable

```
chmod +x log_manager.sh
```



```
labex:project/ $ touch log_manager.sh
labex:project/ $ ls
app_logs  log_manager.sh
labex:project/ $ nano log_manager.sh
labex:project/ $ chmod +x log_manager.sh
labex:project/ $ ./log_manager.sh
log Manager Initialized
labex:project/ $ cat ~/project/log_manager.sh
#!/bin/bash
echo "log Manager Initialized"
labex:project/ $ nano log_manager.sh
labex:project/ $ nano log_manager.sh
labex:project/ $ cat ~/project/log_manager.sh
#!/bin/bash
echo "Log Manager Initialized"
labex:project/ $ 
```

🔧 Step 2: Add Variables and User Input

Purpose

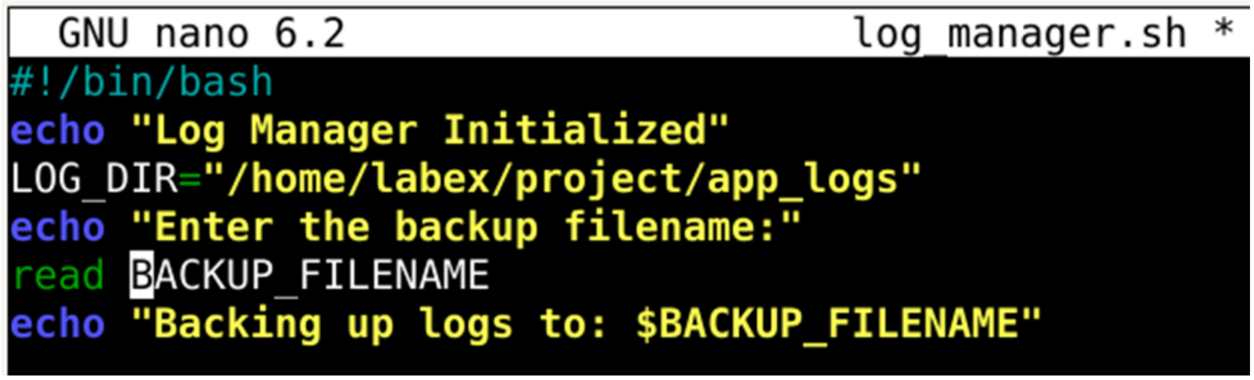
Make the script dynamic and reusable.

Script Enhancements

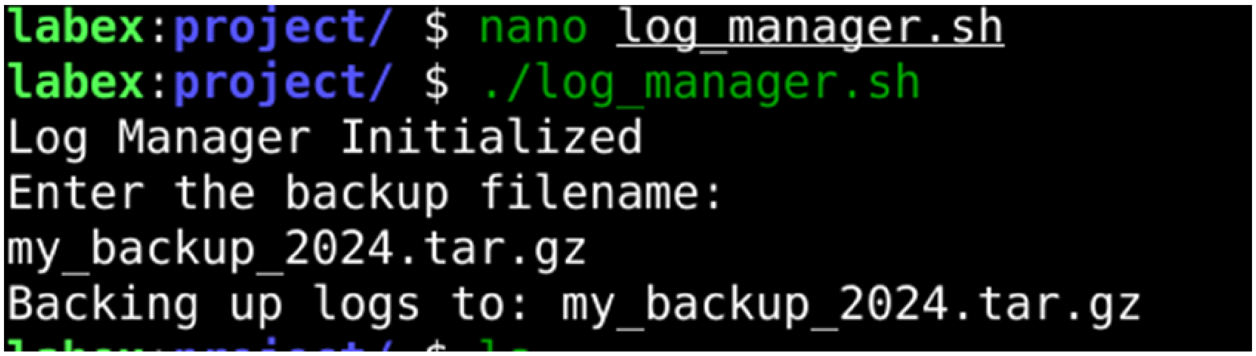
```
LOG_DIR="/home/labex/project/app_logs"

echo "Enter the backup filename: "
read BACKUP_FILENAME

echo "Backing up logs to: $BACKUP_FILENAME"
```

Key Concepts

- Variables (`LOG_DIR`, `BACKUP_FILENAME`)
- Interactive user input with `read`
- Variable interpolation using `$`



```
  GNU nano 6.2                      log_manager.sh *
#!/bin/bash
echo "Log Manager Initialized"
LOG_DIR="/home/labex/project/app_logs"
echo "Enter the backup filename:"
read BACKUP_FILENAME
echo "Backing up logs to: $BACKUP_FILENAME"
```



```
labex:project/ $ nano log_manager.sh
labex:project/ $ ./log_manager.sh
Log Manager Initialized
Enter the backup filename:
my_backup_2024.tar.gz
Backing up logs to: my_backup_2024.tar.gz
```

🛑 Step 3: Implement Conditional Logic

Purpose

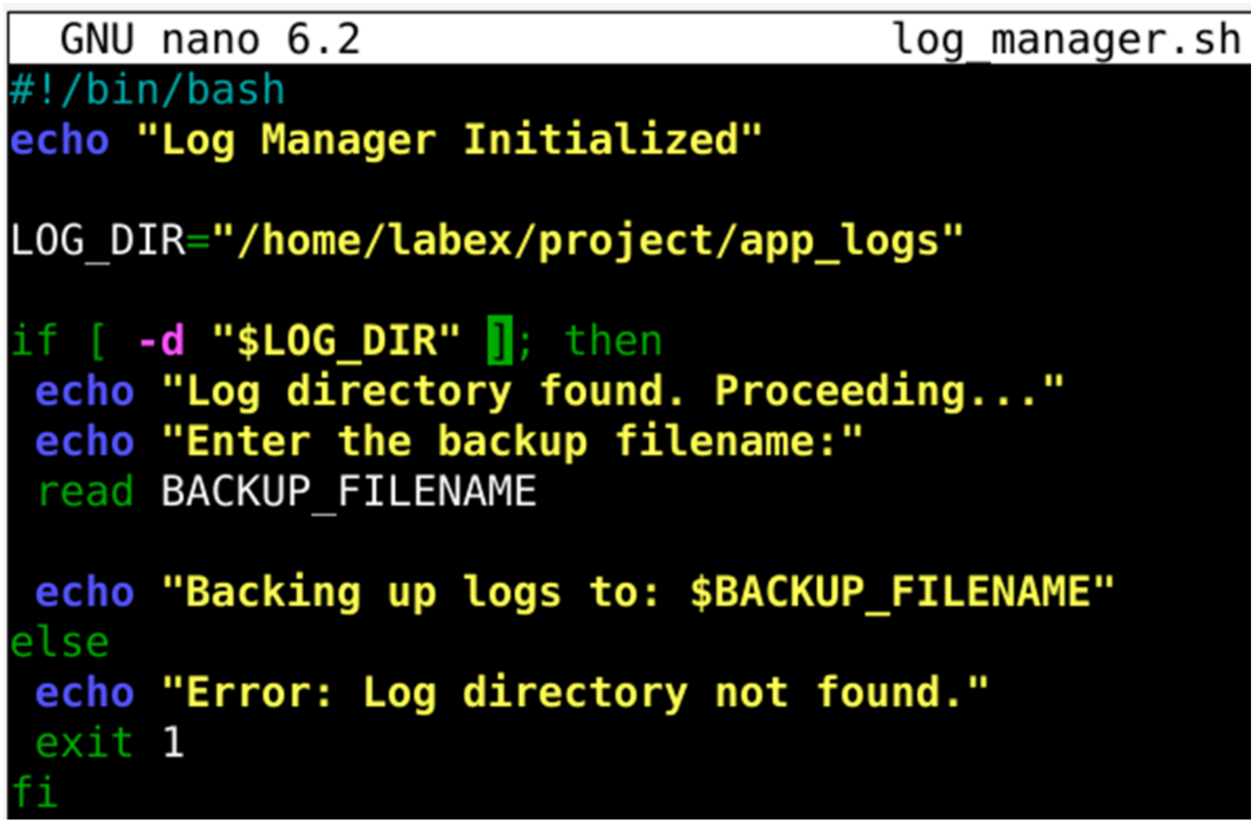
Prevent script failure if the log directory does not exist.

Script Logic

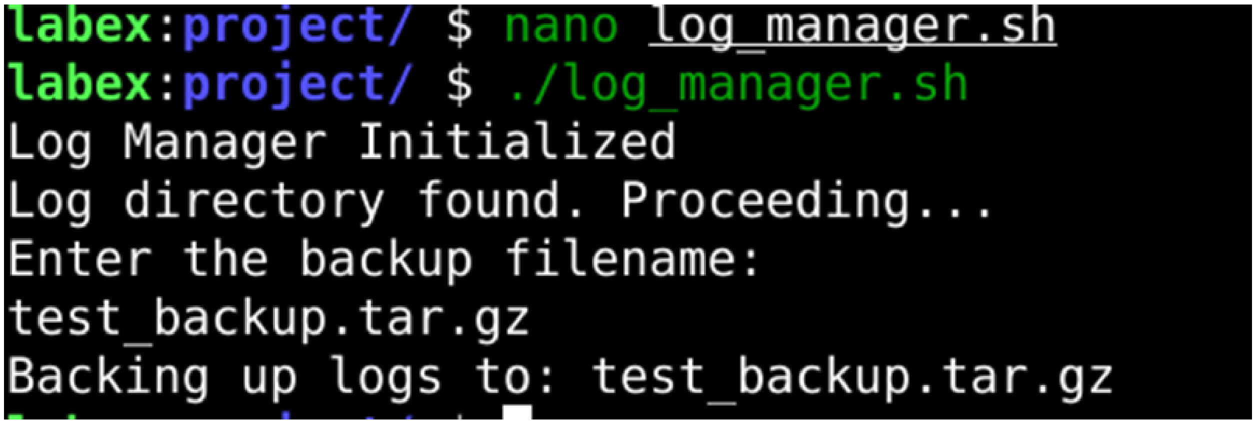
```
if [ -d "$LOG_DIR" ]; then
    echo "Log directory found."
else
    echo "Error: Log directory not found."
    exit 1
fi
```

Outcome

- Script exits safely if prerequisites are not met
- Prevents unintended behavior or errors



```
  GNU nano 6.2                           log_manager.sh
#!/bin/bash
echo "Log Manager Initialized"

LOG_DIR="/home/labex/project/app_logs"

if [ -d "$LOG_DIR" ]; then
 echo "Log directory found. Proceeding..."
 echo "Enter the backup filename:"
 read BACKUP_FILENAME

 echo "Backing up logs to: $BACKUP_FILENAME"
else
 echo "Error: Log directory not found."
 exit 1
fi
```



```
labex:project/ $ nano log_manager.sh
labex:project/ $ ./log_manager.sh
Log Manager Initialized
Log directory found. Proceeding...
Enter the backup filename:
test_backup.tar.gz
Backing up logs to: test_backup.tar.gz
```

🔁 Step 4: Automate File Operations with a Loop

Prepare Backup Directory

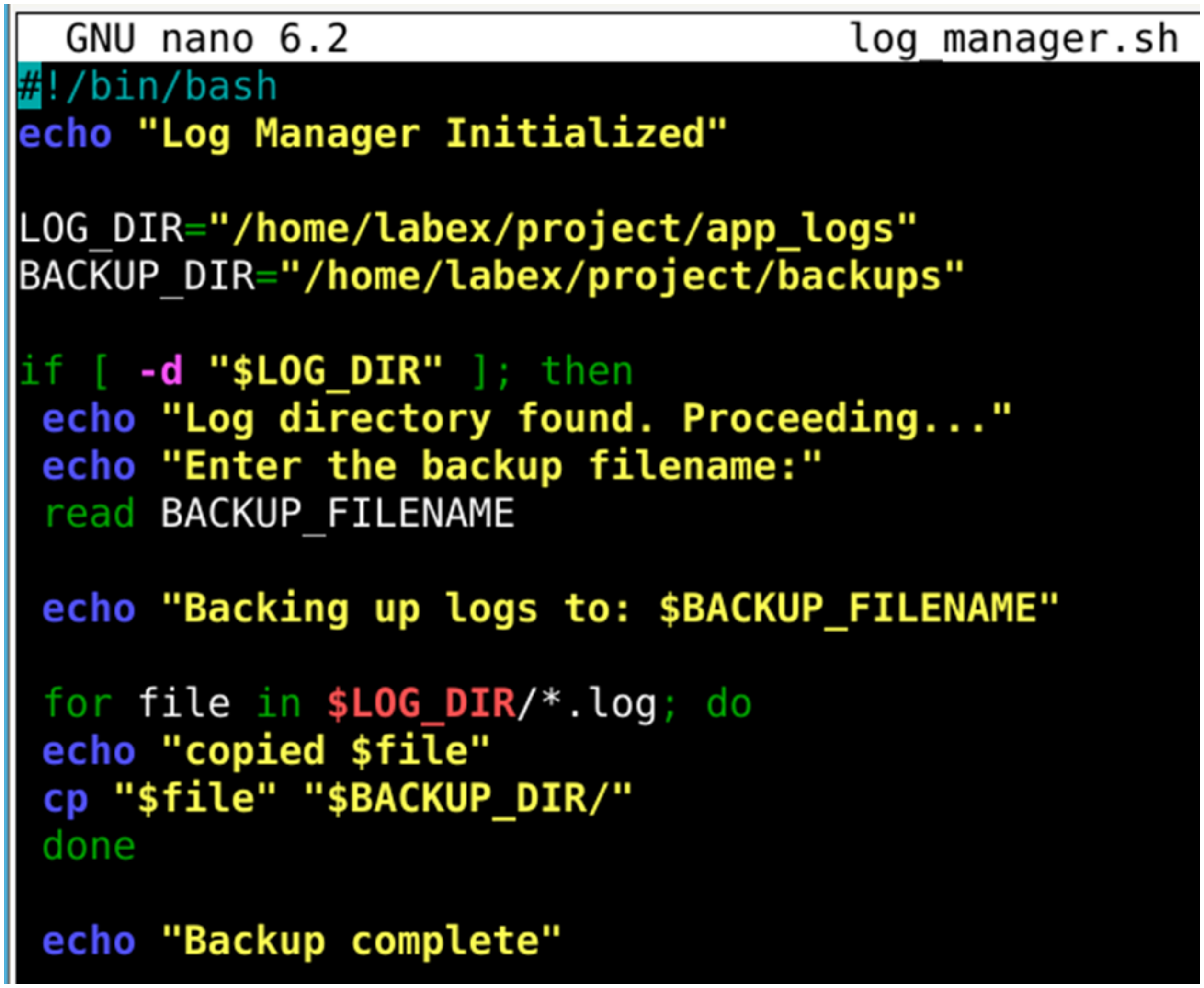
```
mkdir -p ~/project/backups
```

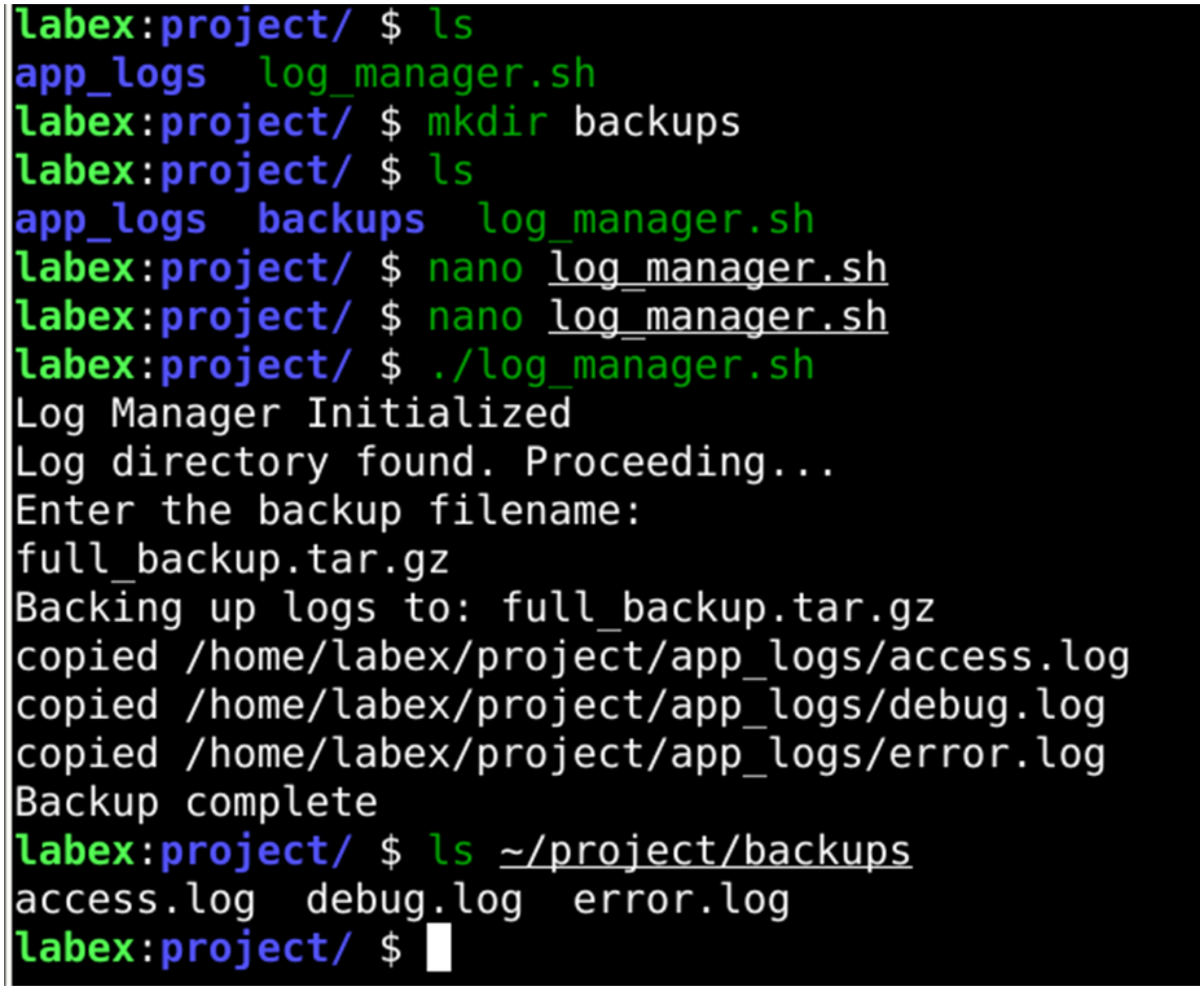
Script Loop

```
for file in "$LOG_DIR"/*.log; do
    cp "$file" ~/project/backups
    echo "Copied $(basename "$file")"
done
```

Result

- Iterates through all `.log` files
- Copies each file to a backup location
- Prints confirmation for every operation


```
  GNU nano 6.2                              log_manager.sh
#!/bin/bash
echo "Log Manager Initialized"

LOG_DIR="/home/labex/project/app_logs"
BACKUP_DIR="/home/labex/project/backups"

if [ -d "$LOG_DIR" ]; then
 echo "Log directory found. Proceeding..."
 echo "Enter the backup filename:"
 read BACKUP_FILENAME

 echo "Backing up logs to: $BACKUP_FILENAME"

 for file in $LOG_DIR/*.log; do
 echo "copied $file"
 cp "$file" "$BACKUP_DIR/"
 done

 echo "Backup complete"
```



```
labex:project/ $ ls
app_logs  log_manager.sh
labex:project/ $ mkdir backups
labex:project/ $ ls
app_logs  backups  log_manager.sh
labex:project/ $ nano log_manager.sh
labex:project/ $ nano log_manager.sh
labex:project/ $ ./log_manager.sh
Log Manager Initialized
Log directory found. Proceeding...
Enter the backup filename:
full_backup.tar.gz
Backing up logs to: full_backup.tar.gz
copied /home/labex/project/app_logs/access.log
copied /home/labex/project/app_logs/debug.log
copied /home/labex/project/app_logs/error.log
Backup complete
labex:project/ $ ls ~/project/backups
access.log  debug.log  error.log
labex:project/ $ 
```

🧠 Skills Demonstrated

- Bash scripting fundamentals
- Shebang usage and script execution
- Variables and user input handling
- Conditional error checking
- Looping and file automation
- Defensive scripting practices