

# Shell Scripting Projects – Learning & DevOps

This README contains **all shell scripting project problem statements**, organized by **difficulty level** and **DevOps relevance**. These projects are designed for learning, practice, and resume-building.

---

## PART 1: Shell Scripting Projects (Learning Purpose)

### Easy Level (Basics)

#### 1. File Organizer Script

##### **Problem Statement:**

Write a shell script that takes a directory path as input and organizes files into separate folders based on their file extensions (e.g., `.txt`, `.pdf`, `.jpg`). If a folder does not exist, the script should create it automatically.

---

#### 2. System Information Script

##### **Problem Statement:**

Create a shell script that displays system-related information such as current username, operating system details, kernel version, disk usage, memory usage, and system uptime in a readable format.

---

#### 3. Simple Backup Script

##### **Problem Statement:**

Develop a script that takes a source directory and creates a compressed backup (`.tar.gz`) with the current date and time in the filename. The backup should be stored in a specified backup directory.

---

### Medium Level (Logic + Functions)

## 4. User Management Script

### Problem Statement:

Write a menu-driven shell script that allows an administrator to create a user, delete a user, and list all users on the system. The script should check for root privileges before execution.

---

## 5. Log Analyzer Script

### Problem Statement:

Create a shell script that reads a log file and reports the number of error messages, warning messages, and unique IP addresses found in the log file.

---

## 6. Disk Space Monitoring Script

### Problem Statement:

Develop a script that monitors disk usage and alerts the user if usage exceeds a defined threshold percentage. The script should be suitable for scheduling via cron.

---

## Hard Level (Advanced Automation)

## 7. Menu-Based Utility Script

### Problem Statement:

Build an interactive shell script with a menu that allows users to perform multiple operations such as backup, disk cleanup, and system monitoring from a single interface.

---

## 8. Process Monitoring Script

### Problem Statement:

Write a script that monitors CPU and memory usage of running processes and automatically terminates processes that exceed predefined resource limits.

---

## 9. Mini Package Management Script

**Problem Statement:**

Create a shell script that can install and remove software packages by detecting the operating system (Ubuntu/CentOS) and using the appropriate package manager.

---

## PART 2: Shell Scripting Projects (DevOps Perspective)

### Easy Level (DevOps Basics)

#### 10. Environment Setup Script

**Problem Statement:**

Develop a shell script that automatically installs commonly used development and DevOps tools such as Git, Docker, and a programming language runtime on a fresh Linux system.

---

#### 11. Application Health Check Script

**Problem Statement:**

Write a script that checks whether an application process is running, verifies if a specific port is open, and confirms that the application returns a valid HTTP status code.

---

#### 12. Log Cleanup Automation Script

**Problem Statement:**

Create a shell script that deletes log files older than a specified number of days to prevent disk space issues. The script should support automation using cron.

---

### Medium Level (CI/CD & Reliability)

#### 13. Automated Backup and Cloud Upload Script

**Problem Statement:**

Develop a script that creates backups of application data and uploads them to a cloud storage service (e.g., AWS S3 or Google Cloud Storage) using command-

line tools.

---

## 14. Service Auto-Restart Script

### Problem Statement:

Write a shell script that continuously checks the health of a system service and automatically restarts it if it becomes unavailable.

---

## 15. Application Deployment Script

### Problem Statement:

Create a deployment automation script that pulls the latest code from a Git repository, builds the application, and restarts the service with minimal downtime.

---

## Hard Level (Production-Grade DevOps)

## 16. Blue-Green Deployment Script

### Problem Statement:

Build a shell script that implements a blue-green deployment strategy by maintaining two environments and switching traffic between them without downtime.

---

## 17. Server Hardening Automation Script

### Problem Statement:

Develop a script that performs basic server hardening tasks such as disabling root SSH login, configuring a firewall, and enabling intrusion prevention tools.

---

## 18. Infrastructure Validation Script

### Problem Statement:

Write a shell script that validates infrastructure readiness by checking CPU, memory, disk space, and network connectivity, and generates a summarized report for deployment readiness.

---

## Usage

- Each project can be implemented as a separate script.
  - Scripts should include proper error handling and logging.
  - Projects can be version-controlled and documented individually inside this repository.
- 

## Goal

The goal of this collection is to strengthen **Linux shell scripting skills** and apply them to **real-world DevOps automation scenarios**.