DevOps Practical Report

# Introduction

This report outlines the steps taken to configure and manage a secure, monitored, and well-maintained development environment for two developers, Sarah and Mike, at TechCorp. The tasks cover system monitoring setup, user management and access control, and backup configuration for web servers. Each task is detailed with commands used, implementation steps, and sample outputs.

# Task 1: System Monitoring Setup

Objective: Configure a monitoring system to ensure the development environment’s health, performance, and capacity planning.  
Scenario: The development server is reporting intermittent performance issues. New developers need visibility into system resource usage for their tasks. System metrics must be consistently tracked for effective capacity planning.  
Requirements:  
- Install and configure monitoring tools (htop or nmon).  
- Monitor disk usage using df and du.  
- Identify resource-intensive processes.  
- Create basic reports saved to log files.

* Implementation Steps:

1. Install htop and nmon:

sudo apt update  
sudo apt install htop nmon -y

2. Monitor CPU, Memory, and Processes using htop or nmon.

3. Disk Usage Monitoring:

df -h > /var/log/disk\_usage\_report.log  
du -sh /home/\* >> /var/log/disk\_usage\_report.log

4. Identify resource-intensive processes:

ps aux --sort=-%mem | head -n 10 >> /var/log/top\_mem\_processes.log  
ps aux --sort=-%cpu | head -n 10 >> /var/log/top\_cpu\_processes.log

5. Reporting: Logs were saved in /var/log for future review and analysis.

# Task 2: User Management and Access Control

Objective: Set up user accounts and configure secure access controls for the new developers.  
Scenario: Sarah and Mike require isolated, secure environments for development tasks. Proper user management and password policies must be enforced.  
Requirements:  
- Create user accounts for Sarah and Mike.  
- Set up dedicated workspace directories.  
- Restrict access permissions.  
- Enforce password expiration and complexity policies.

* Implementation Steps:

1. Create Users:  
sudo useradd -m Sarah  
sudo useradd -m mike  
sudo passwd Sarah  
sudo passwd mike

2. Setup Workspace Directories:  
sudo mkdir /home/Sarah/workspace  
sudo mkdir /home/mike/workspace

3. Set Ownership and Permissions:  
sudo chown Sarah:Sarah /home/Sarah/workspace  
sudo chmod 700 /home/Sarah/workspace  
sudo chown mike:mike /home/mike/workspace  
sudo chmod 700 /home/mike/workspace

4. Enforce Password Policy:  
Edit /etc/login.defs and set:  
PASS\_MAX\_DAYS 30  
PASS\_MIN\_LEN 8

Then apply:  
sudo chage -M 30 Sarah  
sudo chage -M 30 mike

# Task 3: Backup Configuration for Web Servers

Objective: Configure automated backups for Apache and Nginx web servers to ensure data integrity and disaster recovery.  
Scenario: Sarah manages the Apache server, while Mike manages the Nginx server. Each must automate and verify backups weekly.  
Requirements:  
- Backup configurations and document roots.  
- Automate backups via cron jobs.  
- Name backups with date stamps.  
- Verify backup integrity.

* Implementation Steps:

1. Create Backup Directory:  
sudo mkdir -p /backups

* 2. Create Backup Scripts:  
  For Sarah:

#!/bin/bash  
DATE=$(date +%F)  
tar -czvf /backups/apache\_backup\_$DATE.tar.gz /etc/httpd/ /var/www/html/  
ls -lh /backups/apache\_backup\_$DATE.tar.gz > /backups/apache\_backup\_verify\_$DATE.log

* For Mike:

#!/bin/bash  
DATE=$(date +%F)  
tar -czvf /backups/nginx\_backup\_$DATE.tar.gz /etc/nginx/ /usr/share/nginx/html/  
ls -lh /backups/nginx\_backup\_$DATE.tar.gz > /backups/nginx\_backup\_verify\_$DATE.log

* 3. Setup Cron Jobs:  
  sudo crontab -e  
  Add the following lines:

0 0 \* \* 2 /home/Sarah/apache\_backup.sh  
0 0 \* \* 2 /home/mike/nginx\_backup.sh

* 4. Backup Verification:  
  Verify using `ls -lh` to confirm files and log contents.

# Conclusion

The development environment setup was successfully implemented with secure user access, real-time monitoring, and automated backups. Challenges encountered included adjusting file permissions and testing cron job execution. Logs and backups confirm the implementation was effective.