Linux Practice Project for DevOps Engineers

Project Overview

This project simulates real-time server maintenance and deployment tasks using Linux and Shell scripting. It covers health monitoring, log management, backups, deployments, user management, real-time alerting, and automated cleanup.

Project Objectives

- Automate server health checks.
- Implement log rotation and archival.
- Schedule automated backups.
- Automate application deployment.
- Manage users through shell scripts.
- Real-time log monitoring and alerts.
- Cleanup of temporary files and Docker images.
- Automate all processes using cron jobs.

Scripts

1. Health Check Script: `health_check.sh`

```
#!/bin/bash

# CPU, Memory, Disk Health Check
cpu_load=$(top -bn1 | grep "Cpu(s)" | awk '{print $2 + $4}')
memory_usage=$(free | grep Mem | awk '{print $3/$2 * 100.0}')
disk_usage=$(df -h / | awk 'NR==2 {print $5}' | sed 's/%//')

# Thresholds
cpu_threshold=80
memory_threshold=80
disk_threshold=80

if (( ${cpu_load%.*} > cpu_threshold )); then
    echo "CPU Load is high: ${cpu_load}%" | mail -s "CPU Alert"
mindcircuit@gmail.com
fi
```

```
if (( ${memory_usage%.*} > memory_threshold )); then
    echo "Memory usage is high: ${memory_usage}%" | mail -s "Memory Alert"

mindcircuit@gmail.com
fi

if (( disk_usage > disk_threshold )); then
    echo "Disk usage is high: ${disk_usage}%" | mail -s "Disk Alert"

mindcircuit@gmail.com
fi
```

---

2. Log Management Script: `log_rotate.sh`

```
#!/bin/bash

# Compress logs older than 7 days
find /var/log/app/ -name "*.log" -mtime +7 -exec gzip {} \;

# Move compressed logs to backup folder
mkdir -p /var/log/backup
find /var/log/app/ -name "*.gz" -exec mv {} /var/log/backup/ \;

# Delete backups older than 30 days
find /var/log/backup/ -name "*.gz" -mtime +30 -exec rm {} \;
```

---

3. Backup Automation Script: `backup_script.sh`

```
#!/bin/bash

backup_dir="/backup"
date=$(date +%F)
mkdir -p $backup_dir

# Backup /etc and /var/www
tar -czf $backup_dir/etc_backup_$date.tar.gz /etc
tar -czf $backup_dir/www_backup_$date.tar.gz /var/www

# Simulate copying to remote (can use rsync for real cases)
cp $backup_dir/* /remote_backup/
```

---

4. Deployment Automation Script: `deploy_app.sh`

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

```
# Pull latest code
git -C /var/www/html pull origin main

# Set permissions
chmod -R 755 /var/www/html

# Restart service
systemctl restart nginx

# Validate service status
if systemctl status nginx | grep -q running; then
    echo "Deployment successful and nginx is running."
else
    echo "Deployment failed or nginx is not running!" | mail -s "Deployment 
Issue" mindcircuit@gmail.com
fi
```

---

5. User Management Script: `user_management.sh`

```
#!/bin/bash

input="users.csv"
while IFS=, read -r username group

do
    useradd -m -s /bin/bash $username
    mkdir -p /home/$username/.ssh
    ssh-keygen -f /home/$username/.ssh/id_rsa -N ""
    chown -R $username:$username /home/$username/.ssh
    usermod -aG $group $username
    echo "User $username created and added to $group"
done < "$input"
```

---

6. Real-Time Log Monitoring Script: `log_monitor.sh`

```
#!/bin/bash

log_file="/var/log/app/app.log"

tail -F $log_file | while read line; do
    echo "$line" | grep -q "ERROR"
    if [ $? -eq 0 ]; then
        echo "Error detected: $line" | mail -s "Application Error Alert" 
mindcircuit@gmail.com
    fi
done
```

7. Cleanup Script: `cleanup.sh`

```
#!/bin/bash

# Remove temp files older than 7 days
find /tmp -type f -mtime +7 -exec rm {} \;

# Prune unused Docker images
docker image prune -f
```

8. Cron Job Setup

```
# Edit crontab: crontab -e

# Health check every 15 minutes
*/15 * * * * /scripts/health_check.sh

# Log rotation daily at midnight
0 0 * * * /scripts/log_rotate.sh

# Backup daily at midnight
0 0 * * * /scripts/backup_script.sh

# Cleanup every Sunday at 1 AM
0 1 * * 0 /scripts/cleanup.sh

# Log monitoring runs at startup as a background process
@reboot /scripts/log_monitor.sh &
```

Summary

This project covers key Linux and DevOps automation scenarios: - Health monitoring - Log and backup management - Automated deployments - User provisioning - Real-time alerting - Scheduled system maintenance