Lesson: Popular Linux Scripts in Production Environments

Use cases showing the most popular Linux scripts in production environments. Let me know if you'd like this in a PowerPoint presentation format.

Lesson: Popular Linux Scripts in Production Environments

1. Introduction

- **Why Linux Scripts are Crucial in Production**
  - Automate repetitive tasks.
  - Ensure consistent workflows.
  - Enhance system performance and reliability.

2. Use Case 1: Log Management

- **Objective:** Automate the rotation, archiving, and cleanup of large log files.
- **Common Script Features:**
  - Compress old logs using `gzip` or `tar`.
  - Delete logs older than a specific number of days.
  - Monitor log size and alert if it exceeds thresholds.
- **Sample Script:**

```
#!/bin/bash
LOG_DIR="/var/log/app/"
ARCHIVE_DIR="/var/log/app/archives/"
find $LOG_DIR -name "*.log" -mtime +7 -exec gzip {} \;
find $ARCHIVE_DIR -name "*.gz" -mtime +30 -delete

```

- **Use in Production:**
  - Prevent disk space exhaustion.
  - Enable efficient debugging.

---

3. Use Case 2: Monitoring System Health

- **Objective:** Alert administrators when CPU, memory, or disk usage exceeds thresholds.
- **Common Script Features:**
  - Use `top`, `df`, and `free` commands to monitor usage.
  - Send email or Slack alerts.

- **Sample Script:**

```
#!/bin/bash
THRESHOLD=90
USAGE=$(df / | grep / | awk '{ print $5 }' | sed 's/%//')
if [ $USAGE -gt $THRESHOLD ]; then
    echo "Disk usage is above $THRESHOLD%" | mail -s "Disk Usage Alert" admin@example.com
fi

```

- **Use in Production:**
  - Proactive issue resolution before outages.

---

4. Use Case 3: Database Backups

- **Objective:** Schedule regular backups of databases.
- **Common Script Features:**
  - Use `mysqldump` or `pg_dump` for backups.
  - Encrypt and upload to a remote server or cloud storage.
- **Sample Script:**

```
#!/bin/bash
DB_NAME="prod_db"
BACKUP_DIR="/backups/"
TIMESTAMP=$(date +"%Y%m%d%H%M")
mysqldump -u root -pPassword $DB_NAME > ${BACKUP_DIR}${DB_NAME}_${TIMESTAMP}.sql
gzip ${BACKUP_DIR}${DB_NAME}_${TIMESTAMP}.sql

```

- **Use in Production:**
  - Disaster recovery and compliance with data retention policies.

---

5. Use Case 4: Deployment Automation

- **Objective:** Automate the deployment of new application versions.
- **Common Script Features:**
  - Stop current service.
  - Deploy updated code from Git or artifact repository.
  - Restart the service and perform health checks.
- **Sample Script:**

```
#!/bin/bash
APP_DIR="/opt/app/"
REPO="https://github.com/org/app.git"
SERVICE_NAME="app_service"
```

```
cd $APP_DIR
git pull $REPO
systemctl restart $SERVICE_NAME

```

- **Use in Production:**
  - Minimize downtime and ensure consistent deployments.

---

6. Use Case 5: User Management

- **Objective:** Automate user account creation and cleanup.
- **Common Script Features:**
  - Add new users with specific permissions.
  - Remove unused accounts to enhance security.
- **Sample Script:**

```
#!/bin/bash
USERNAME="new_user"
PASSWORD="password123"
useradd -m -s /bin/bash $USERNAME
echo "$USERNAME:$PASSWORD" | chpasswd

```

- **Use in Production:**
  - Efficient onboarding/offboarding processes.

---

7. Use Case 6: Application Uptime Monitoring

- **Objective:** Restart services if they fail unexpectedly.
- **Common Script Features:**
  - Check service status using `systemctl` or `ps`.
  - Restart service if it's not running.
- **Sample Script:**

```
#!/bin/bash
SERVICE="nginx"
if ! pgrep $SERVICE > /dev/null; then
    systemctl restart $SERVICE
fi

```

- **Use in Production:**
  - Reduce downtime without manual intervention.

---

8. Use Case 7: File Transfer Automation

- **Objective:** Automate file synchronization between servers.
- **Common Script Features:**
  - Use `rsync` or `scp` for secure transfers.
  - Schedule scripts using `cron`.
- **Sample Script:**

```
#!/bin/bash
SOURCE="/var/data/"
DEST="user@remote:/backup/"
rsync -avz $SOURCE $DEST

```

- **Use in Production:**
  - Ensure timely backups and data consistency.

---

9. Summary

- Linux scripts are invaluable in production for:
  - Reducing manual effort.
  - Enhancing system reliability.
  - Improving operational efficiency.

10. Exercises

1. Write a script to monitor free memory and alert when it drops below a threshold.
2. Modify the database backup script to upload backups to AWS S3.
3. Create a deployment script that rolls back changes if the application health check fails.
