**# TASK 3 Automated Backup and Verification for Apache and Nginx**

A simple solution to automate and verify backups for Apache and Nginx servers. This setup ensures secure and reliable backups of server configurations and document roots, scheduled weekly using cron jobs.

**Features**

* **Automated Backups:** Weekly backups using cron jobs.
* **Secure Storage:** Backup files stored in /backups/ with clear naming conventions.
* **Integrity Verification:** Ensures backups are complete and error-free.
* **Lightweight Scripts:** Minimal and efficient bash scripts for automation.

**Requirements**

* Linux-based operating system
* Apache and/or Nginx installed
* Write permissions for /backups/ directory
* Basic understanding of bash scripting and cron jobs

**Getting Started**

**1. Create Backup Scripts**

Write two bash scripts for automating backups (replace <path\_to\_script> with the actual script location):

* **backup\_apache.sh**
* **backup\_nginx.sh**

**Content of backup\_apache.sh (replace paths as needed):**

Bash

#!/bin/bash

# Get current date

DATE=$(date +%Y-%m-%d)

# Backup directory

BACKUP\_DIR=/backups

# Apache configuration directory

APACHE\_CONF\_DIR=/etc/apache2

# Apache document root directory (replace with your actual path)

APACHE\_DOC\_ROOT=/var/www/html

# Create backup archive

tar -czf ${BACKUP\_DIR}/apache\_backup\_${DATE}.tar.gz ${APACHE\_CONF\_DIR} ${APACHE\_DOC\_ROOT}

# Verify backup integrity

tar -tzf ${BACKUP\_DIR}/apache\_backup\_${DATE}.tar.gz

# Print completion message

echo "Apache backup completed and verified on $(date)"

**Content of backup\_nginx.sh (replace paths as needed):**

Bash

#!/bin/bash

# Get current date

DATE=$(date +%Y-%m-%d)

# Backup directory

BACKUP\_DIR=/backups

# Nginx configuration directory

NGINX\_CONF\_DIR=/etc/nginx

# Nginx document root directory (replace with your actual path)

NGINX\_DOC\_ROOT=/var/www/html

# Create backup archive

tar -czf ${BACKUP\_DIR}/nginx\_backup\_${DATE}.tar.gz ${NGINX\_CONF\_DIR} ${NGINX\_DOC\_ROOT}

# Verify backup integrity

tar -tzf ${BACKUP\_DIR}/nginx\_backup\_${DATE}.tar.gz

# Print completion message

echo "Nginx backup completed and verified on $(date)"

**Explanation:**

* The scripts create compressed backup archives (.tar.gz) with timestamps for easy identification.
* They verify the integrity of the backups using tar -tzf.

**2. Set Permissions**

Make the scripts executable:

Bash

chmod +x backup\_apache.sh backup\_nginx.sh

**3. Create Backup Directory**

Ensure the /backups/ directory exists and has write permissions for the user running the scripts:

Bash

sudo mkdir -p /backups

sudo chmod 766 /backups

**Note:** Adjust permissions (766) based on your specific needs.

**4. Schedule Cron Jobs**

Add these lines to your crontab to schedule backups every Tuesday at 12:00 AM (replace <path\_to\_script> with the actual script location):

Bash

crontab -e

Add these lines:

0 0 \* \* 2 <path\_to\_script>/backup\_apache.sh

0 0 \* \* 2 <path\_to\_script>/backup\_nginx.sh

**Explanation:**

* 0 0 \* \* 2: Runs the commands at 00:00 hours on Tuesdays (\* \* 2).

Save and exit the editor.

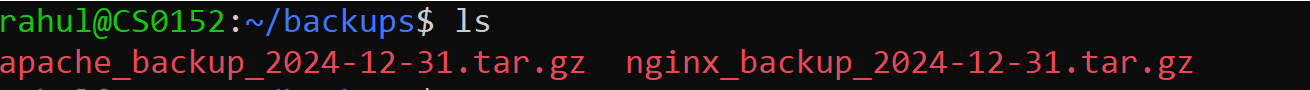
**Verify Backups**

**Manually verify the backups using these commands:**

List the backup files:

Bash

ls /backups/



Verify the integrity of specific backups (replace with the desired filename):

Bash

tar -tzf /backups/apache\_backup\_2024-12-31.tar.gz

tar -tzf /backups/nginx\_backup\_2024-12-31.tar.gz