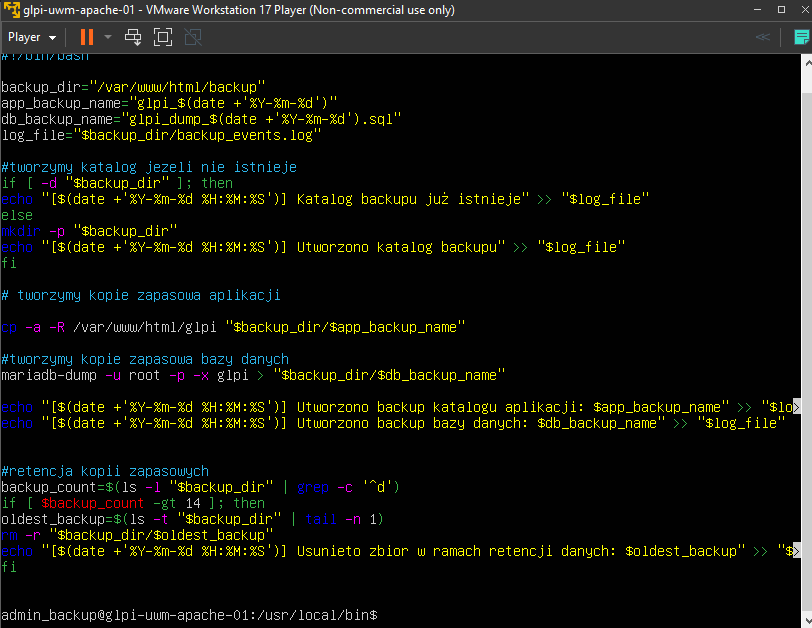
|  |  |  |  |
| --- | --- | --- | --- |
|  | |  | |
| Organizator | PKP Informatyka Sp. z o.o. | Organizator | Uniwersytet Warmińsko-Mazurski w Olsztynie |
| Produkty | Ćwiczenia | Wersja dok. | 2.0 |
| Spotkanie | Projektowe | Temat | Zarządzanie zasobami IT (ITAM) |
| Zespół | Zespół A |  |  |

|  |  |
| --- | --- |
| ***Backup i Restore*** | |
| **I** | **Zadania** |
| 1. | **Zadanie grupowe - Zespół A**  **Zadanie 1: Procedura Backup i Restore Procedura Backup**   1. Tworzymy nowego użytkownika admin\_backup sudo adduser admin\_backup      1. Zmieniamy jego uprawnienia   sudo usermod -aG sudo admin\_backup     1. Przełączamy bieżącego użytkownika na użytkownika admin\_backup   sudo su admin\_backup     1. Instalujemy menadżera zadań crone sudo apt-get update   sudo apt-get install cron |

|  |  |
| --- | --- |
|  | 5. Tworzymy skrypt cd //usr/local/bin  sudo nano crone\_backup.sh    #!/bin/bash backup\_dir="/var/www/html/backup"  app\_backup\_name="glpi\_$(date +'%Y-%m-%d')" db\_backup\_name="glpi\_dump\_$(date +'%Y-%m-%d').sql" log\_file="$backup\_dir/backup\_events.log"  # Sprawdzamy, czy katalog już istnieje  if [ -d "$backup\_dir" ]; then  echo "[$(date +'%Y-%m-%d %H:%M:%S')] Katalog backupu już istnieje: $backup\_dir" >> "$log\_file" else  mkdir -p "$backup\_dir"  echo "[$(date +'%Y-%m-%d %H:%M:%S')] Utworzono katalog backupu: $backup\_dir" >> "$log\_file"  fi  # Tworzymy kopię zapasową aplikacji  cp -a -R /var/www/html/glpi "$backup\_dir/$app\_backup\_name"  # Tworzymy kopię zapasową bazy danych  mariadb-dump -u root -p -x glpi > "$backup\_dir/$db\_backup\_name"  # Logowanie zdarzeń  echo "[$(date +'%Y-%m-%d %H:%M:%S')] Utworzono backup katalogu aplikacji: $app\_backup\_name" >> "$log\_file" echo "[$(date +'%Y-%m-%d %H:%M:%S')] Utworzono backup bazy danych: $db\_backup\_name" >> "$log\_file"  # Retencja kopii zapasowych - utrzymanie 14 ostatnich kopii backup\_count=$(ls -l "$backup\_dir" | grep -c '^d')  if [ $backup\_count -gt 14 ]; then oldest\_backup=$(ls -t "$backup\_dir" | tail -n 1) rm -r "$backup\_dir/$oldest\_backup"  echo "[$(date +'%Y-%m-%d %H:%M:%S')] Usunięto zbiór w ramach retencji danych: $oldest\_backup" >> "$log\_file"  fi  **Poniżej screnshot powyższego skryptu:** |



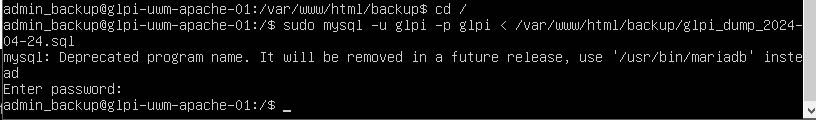
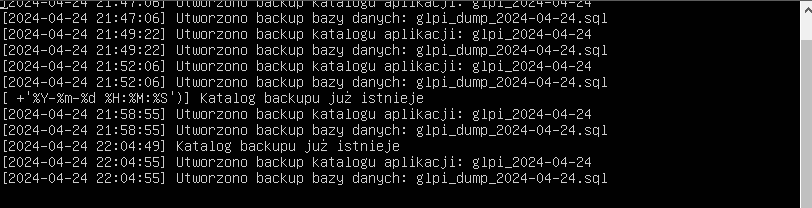
1. Zmieniamy uprawnienia dla pliku crone\_backup.sh sudo chmod +x crone\_backup.sh
2. Uruchamiamy skrypt, aby potwierdzić, że działa.

cd //usr/local/bin

sudo ./crone\_backup.sh

Wszystko działa poprawnie

Tak wyglądają logi:



1. Otwieramy edytor pliku crontab crontab -e
2. Ustawienie harmonogramu codziennie o 21 i przekierowujemy zarówno standardowe wyjście, jak i błędy tego skryptu do pliku dziennika crone\_backup.log.

0 21 \* \* \* /usr/local/bin/crone\_backup.sh >> usr/local/bin/crone\_backup.log 2>&1

**Restore**

Przywrócenie bazy danych:

sudo mysql -u glpi -p glpi < /var/www/html/backup/glpi\_dump\_2024-04-24.sql

Przywrócenie plików GLPI:

sudo cp -a -R /var/www/html /backup/glpi\_2024-04-24/\* /var/www/html/glpi/