

CO-1

K. Kishore.

Home Assignment

2000031667

- 1) Make a memory utilization check script on linux host, and if it's lower than 1GB throw an error "minimum of 1GB RAM is required."
Verify exit code with help of: echo \$

A: there is a bash script that performs a memory utilization check on a linux host and throws an error if the available memory is lower than 1GB.

```
#!/bin/bash
MEM_TOTAL=$(grep MemTotal /proc/meminfo | awk '{ print $2 }')
MEM_AVAIL=$(grep MemAvailable /proc/meminfo | awk '{ print $2 }')

if [ $((MEM_AVAIL/1024)) -lt 1024 ]; then
    echo "minimum of 1GB RAM is required."
    exit 1
else
    echo "Memory check passed."
    exit 0
fi
```

you can verify the exit code of the script by running it and checking the value of '\$?', after the script has completed.

bash

• /memory-check.sh

echo \$?

If the memory check passes, the exit code will be '0', indicating success. if the memory check fails, the exit code will be '1', indicating failure.

- 2> Bash script that performs backups of specified and a database, with the following features.
- configurable backup directory, list of directories to backup, database name, and database user.
 - Archives and compresses the files using 'tar'.
 - Dumps and archives the database using 'mysqldump' and 'tar'.
 - Adds a date tag to the backup files.
 - Makes separate daily and weekly backups.
 - Rotates the backups, keeping only the last 7 daily backups & 4 weekly backups.

configuration parameters.

BACKUP_DIR = (path / to / backup / directory

DIRECTORIES_TO_BACKUP = (/ path / to / dirs / path / to /
dir 2)

DATABASE_NAME = database_name

DATABASE_USER = database_user

Date.

DATE_TAG = \$(date +%Y-%m-%d)

weekly backup flag.

IS_WEEKLY = false

check if weekly backup.

if ["\$(date +%u)" -eq 7]; then

IS_WEEKLY = true

fi

backup function.

backup() {

local dir-to-backup = \$1

local target-file = \$2

creating tar.

tar -czf \$target-file \$dir-to-backup.


```
# check if the backup was successful.
if [ $? -ne 0 ]; then
    echo "Backup of $dir-to-backup failed."
    exit 1
fi
}
```

```
# Backup the directories.
```

```
for dir in "${DIRECTORIES_TO_BACKUP[@]}"; do
    backup $dir $BACKUP_DIR/$DATE_TAG-$
        (base name $dir).tar.gz
done.
```

```
# Backup database.
```

```
mysqldump -u $DATABASE_USER -p $DATABASE_NAME
gzip > $BACKUP_DIR/$DATE_TAG-$DATABASE.
```

```
# check if the backup was successful.
```

```
if [ $? -ne 0 ]; then
```

```
    echo "Backup of database $DATABASE_NAME fail
```

```
    exit 1
```

```
fi
```

Remove old backups

```
find $BACKUP_DIR -type f \(-name "*.tar.gz" -o  
-name "*.sql.gz" \) -mtime +7
```

Remove old weekly backups, if this is not a weekly backup.

if ["\$IS_WEEKLY" = false]; then

```
find $BACKUP_DIR -type f \(-name
```

```
"*-week*.tar.gz" -o -name "*-week*.sql.gz"
```

fi

Add the backup script to crontab

```
echo "0 0 * * * /path/to/backup-script.sh" |  
crontab -
```