Academic Project - FINAL REPORT

By Kunjan Sharma

Objective

The objective of this lab was to practice Linux system administration tasks by ensuring multiple Debian virtual machines were fully updated and by creating and executing an automated backup script. The goal was to demonstrate the ability to manage system updates, write and understand shell scripts, and verify successful backups across multiple systems in a controlled lab environment.

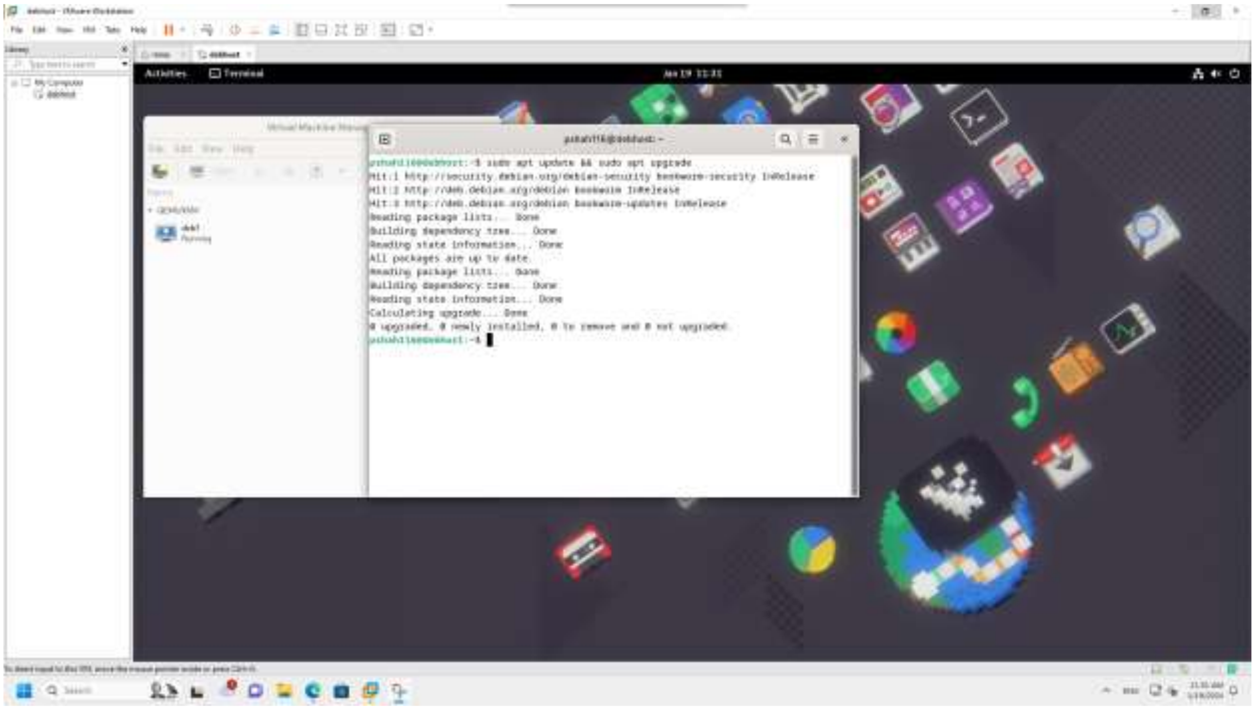
Environment Overview

The lab environment consisted of the following systems:

- **DEB1** – Debian virtual machine
- **DEB2** – Debian virtual machine
- **DEB3** – Debian virtual machine
- **DEBHOST** – Host machine used to run the backup script

All systems were managed using Linux command-line tools and shell scripting.

Explanation of Screenshots and Steps



```
[illegible]@debhost:~$ sudo apt update && sudo apt upgrade
Hit:1 http://security.debian.org/debian-security bookworm-security InRelease
Hit:2 http://deb.debian.org/debian bookworm InRelease
Hit:3 http://deb.debian.org/debian bookworm-updates InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
[illegible]@debhost:~$
```

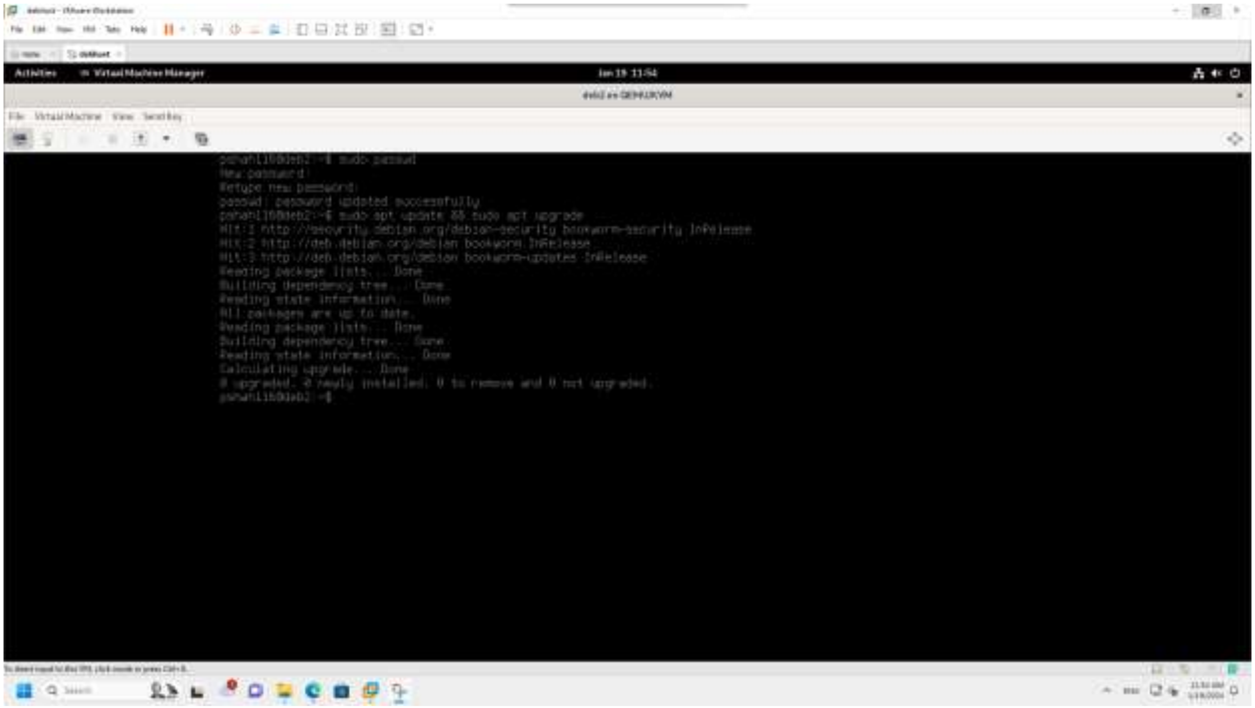
Screenshot 1: DEB1 Update and Upgrade

This screenshot shows the Debian machine **DEB1** being updated and upgraded using the following commands:

- apt update
- apt upgrade

The apt update command refreshes the package list from the repositories, while apt upgrade installs the latest available versions of installed packages. This confirms that DEB1 is fully up to date.

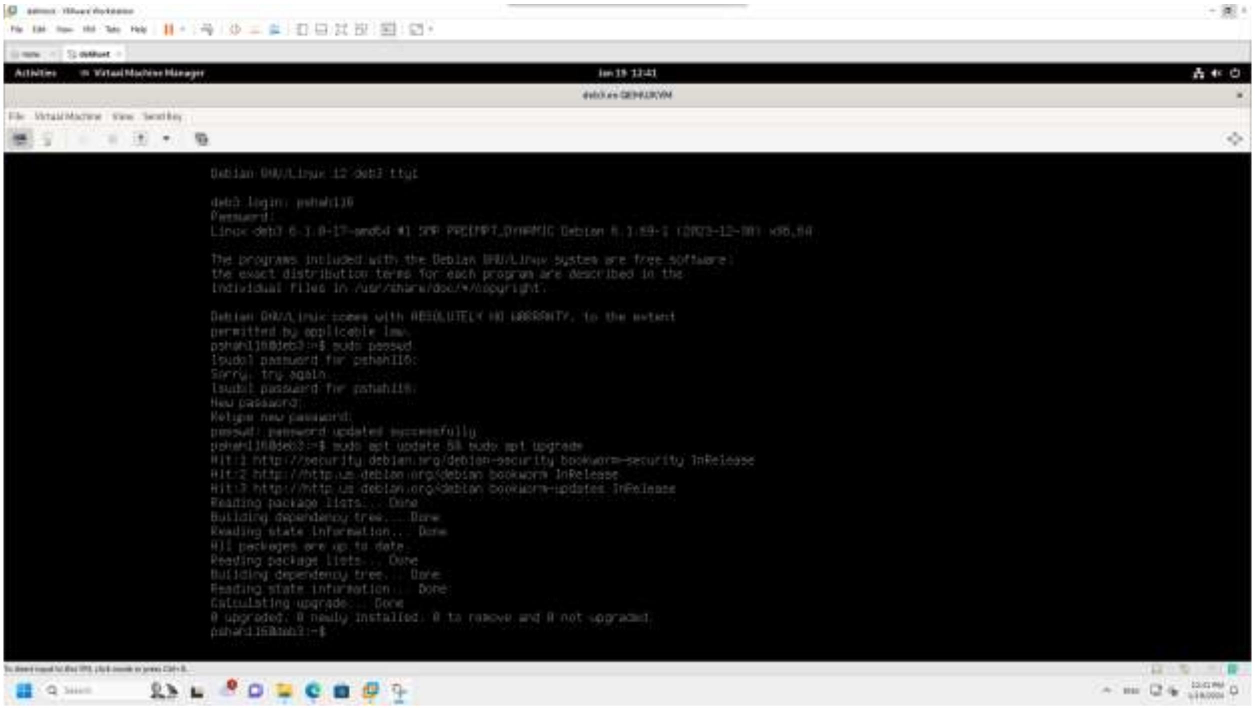
Screenshot 2: DEB2 Update and Upgrade



```
pshah1150@deb2:~$ sudo passwd
New password:
Retype new password:
passwd: password updated successfully
pshah1150@deb2:~$ sudo apt update && sudo apt upgrade
Hit:1 http://security.debian.org/debian-security bookworm-security InRelease
Hit:2 http://deb.debian.org/debian bookworm InRelease
Hit:3 http://deb.debian.org/debian bookworm-updates InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
pshah1150@deb2:~$
```

This screenshot shows the same update and upgrade process performed on **DEB2** using apt update and apt upgrade. This ensures consistency across systems and confirms that DEB2 is running the latest package versions.

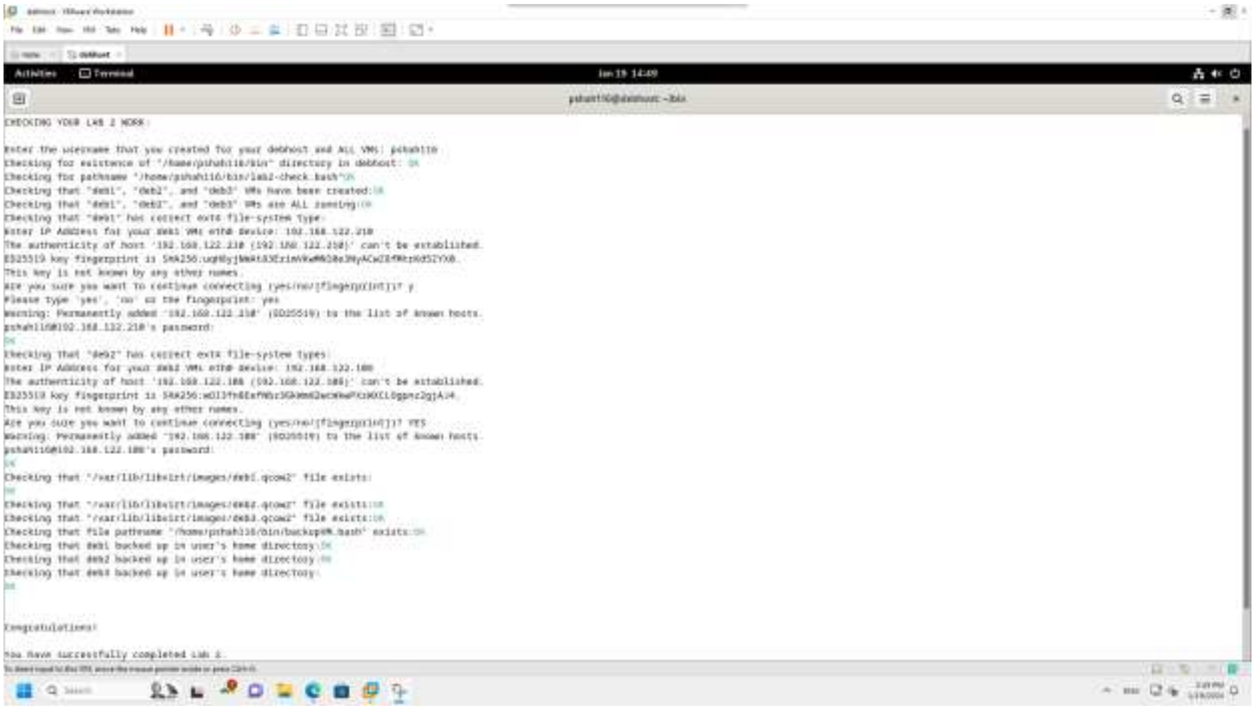
Screenshot 3: DEB3 Update and Upgrade


debhost - VMware Workstation
File Edit View VM Tabs Help
Home
debhost
Activities
Virtual Machine Manager
Jan 19 12:41
deb3 on QEMU/KVM
File Virtual Machine View Send Key

```
Debian GNU/Linux 12 deb3 tty1

deb3 login: pshah116
Password:
Linux deb3 6.1.0-17-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.69-1 (2023-12-30) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
pshah116@deb3:~$ sudo passwd
[sudo] password for pshah116:
Sorry, try again.
[sudo] password for pshah116:
New password:
Retype new password:
passwd: password updated successfully
pshah116@deb3:~$ sudo apt update && sudo apt upgrade
Hit:1 http://security.debian.org/debian-security bookworm-security InRelease
Hit:2 http://http.us.debian.org/debian bookworm InRelease
Hit:3 http://http.us.debian.org/debian bookworm-updates InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
pshah116@deb3:~$
```

This screenshot demonstrates that **DEB3** has also been successfully updated and upgraded. All three Debian machines are now synchronized in terms of system updates, which is important before performing backups or automation tasks.

Screenshot 4: Backup Script Execution on DEBHOST

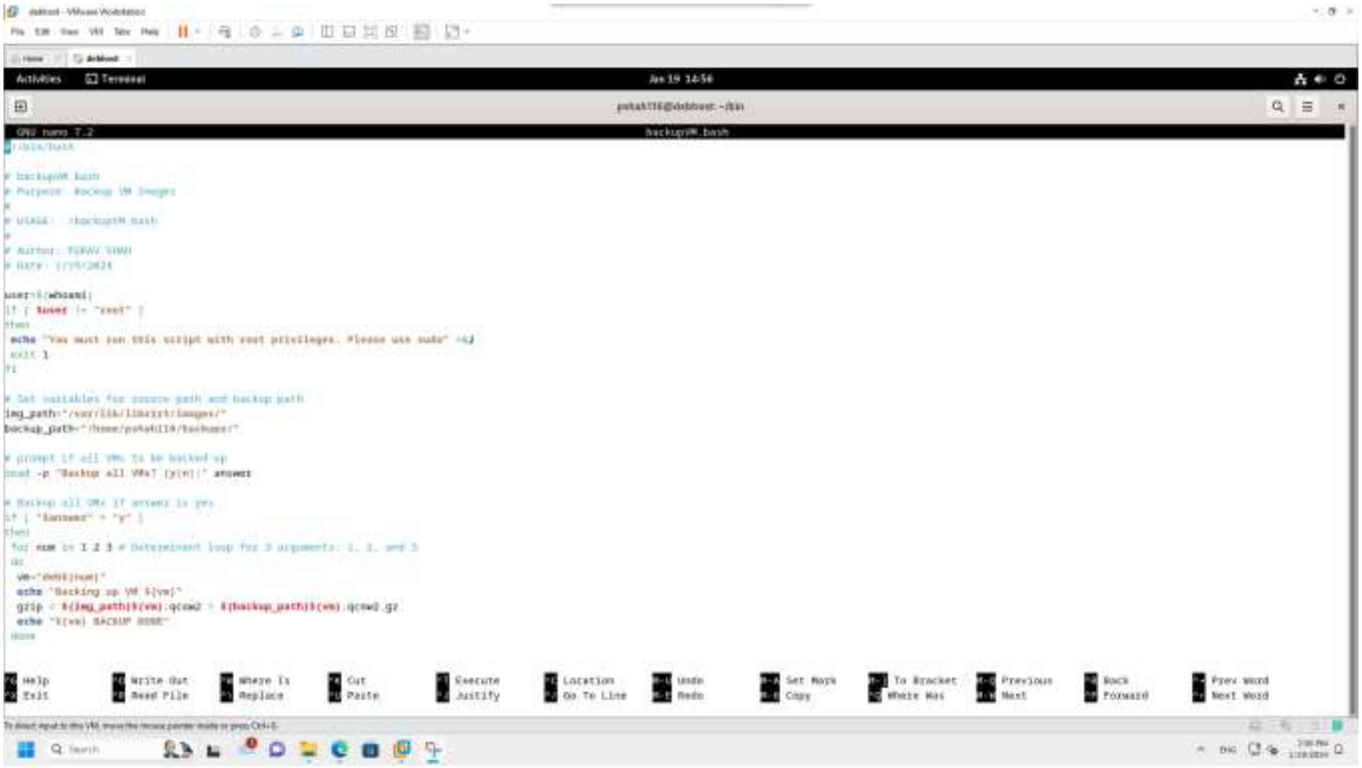


```
CHECKING YOUR LAB 2 WORK:

Enter the username that you created for your debhost and ALL VMs: pshah116
Checking for existence of "/home/pshah116/bin" directory in debhost: OK
Checking for pathname "/home/pshah116/bin/lab2-check.bash":OK
Checking that "deb1", "deb2", and "deb3" VMs have been created:OK
Checking that "deb1", "deb2", and "deb3" VMs are ALL running:OK
Checking that "deb1" has correct ext4 file-system type:
Enter IP Address for your deb1 VMs eth0 device: 192.168.122.210
The authenticity of host '192.168.122.210 (192.168.122.210)' can't be established.
ED25519 key fingerprint is SHA256:ugH8yjNWAt83ErimVKwMN38e3NyACwZ8fMtzKd52YX8.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '192.168.122.210' (ED25519) to the list of known hosts.
pshah116@192.168.122.210's password:
OK
Checking that "deb2" has correct ext4 file-system types:
Enter IP Address for your deb2 VMs eth0 device: 192.168.122.188
The authenticity of host '192.168.122.188 (192.168.122.188)' can't be established.
ED25519 key fingerprint is SHA256:wOl3fn8EefWbz3GkWmQwcWwPKsWXCL0gpnz2gjA24.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? YES
Warning: Permanently added '192.168.122.188' (ED25519) to the list of known hosts.
pshah116@192.168.122.188's password:
OK
Checking that "/var/lib/libvirt/images/deb1.qcow2" file exists:
OK
Checking that "/var/lib/libvirt/images/deb2.qcow2" file exists:OK
Checking that "/var/lib/libvirt/images/deb3.qcow2" file exists:OK
Checking that file pathname "/home/pshah116/bin/backupVM.bash" exists:OK
Checking that deb1 backed up in user's home directory:OK
Checking that deb2 backed up in user's home directory:OK
Checking that deb3 backed up in user's home directory:
OK


Congratulations!

You have successfully completed Lab 2.
```

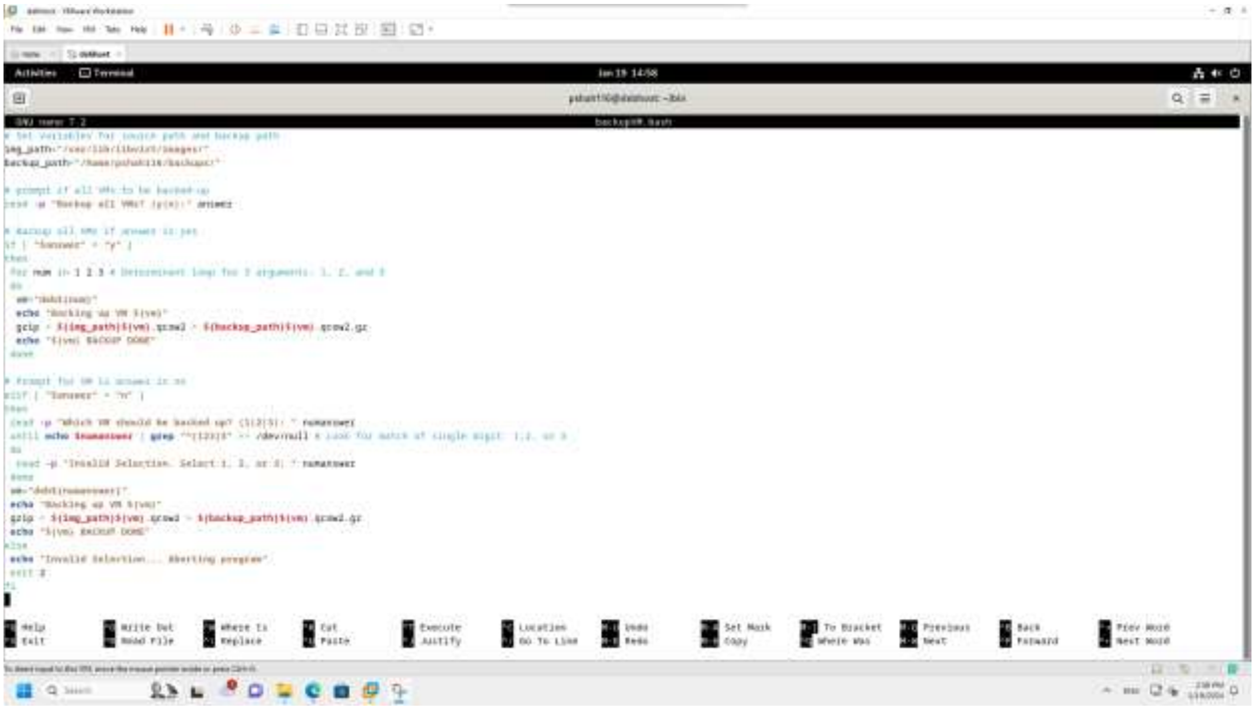
This screenshot shows the backup script running on **DEBHOST**. The output includes confirmation messages such as "OK" and "Congratulations," indicating that the script executed successfully without errors.

This step verifies that the script logic works as intended and that the backup process was initiated correctly.

Screenshot 5: Contents of backupVM.bash



```
#!/bin/bash

# backupVM.bash
# Purpose: Backup VM Images
#
# USAGE: ./backupVM.bash
#
# Author: PURAV SHAH
# Date: 1/19/2024

user=$(whoami)
if [ $user != "root" ]
then
 echo "You must run this script with root privileges. Please use sudo" >&2
 exit 1
fi

# Set variables for source path and backup path
img_path="/var/lib/libvirt/images/"
backup_path="/home/pshah116/backups/"

# prompt if all VMs to be backed up
read -p "Backup all VMs? (y|n):" answer

# Backup all VMs if answer is yes
if [ "$answer" = "y" ]
then
 for num in 1 2 3 # Determinant loop for 3 arguments: 1, 2, and 3
 do
  vm="deb${num}"
  echo "Backing up VM ${vm}"
  gzip < ${img_path}${vm}.qcow2 > ${backup_path}${vm}.qcow2.gz
  echo "${vm} BACKUP DONE"
 done
```



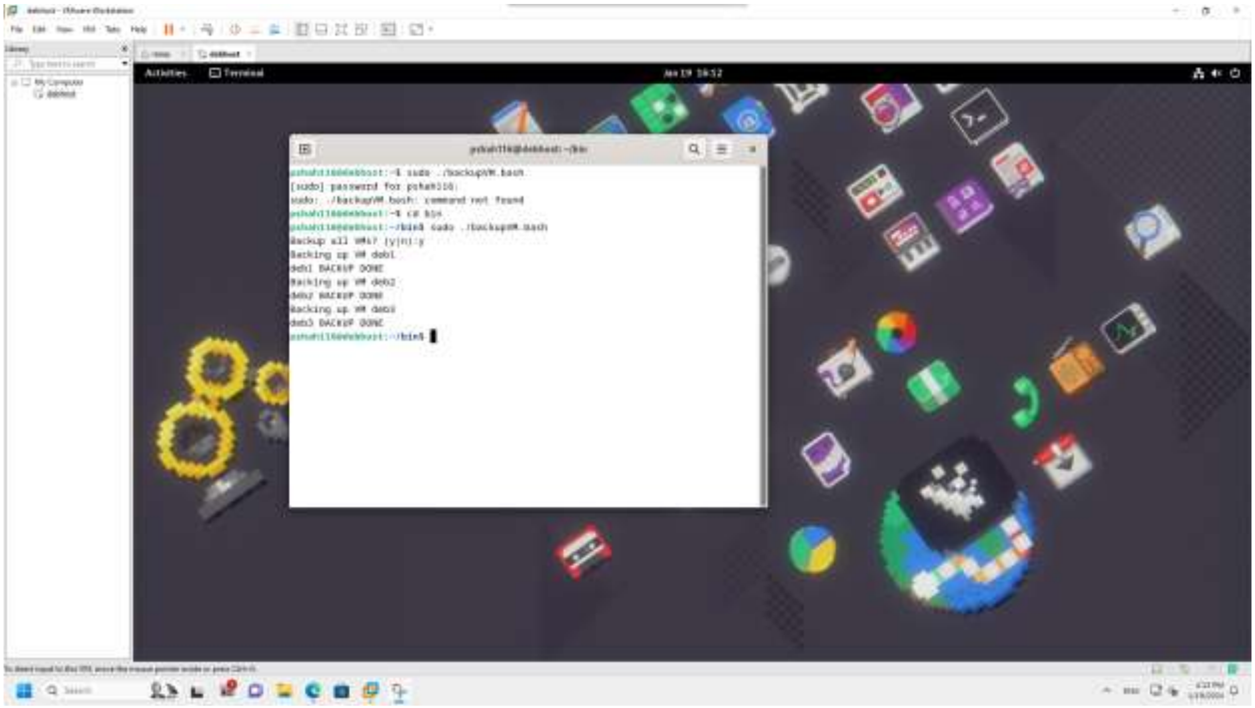
```
# Set variables for source path and backup path
img_path="/var/lib/libvirt/images/"
backup_path="/home/pshah116/backups/"

# prompt if all VMs to be backed up
read -p "Backup all VMs? (y|n):" answer

# Backup all VMs if answer is yes
if [ "$answer" = "y" ]
then
 for num in 1 2 3 # Determinant loop for 3 arguments: 1, 2, and 3
 do
  vm="deb${num}"
  echo "Backing up VM ${vm}"
  gzip < ${img_path}${vm}.qcow2 > ${backup_path}${vm}.qcow2.gz
  echo "${vm} BACKUP DONE"
 done

# Prompt for VM is answer is no
elif [ "$answer" = "n" ]
then
 read -p "Which VM should be backed up? (1|2|3): " numanswer
 until echo $numanswer | grep "^[123]$" >> /dev/null # Look for match of single digit: 1,2, or 3
 do
  read -p "Invalid Selection. Select 1, 2, or 3: " numanswer
 done
 vm="deb${numanswer}"
 echo "Backing up VM ${vm}"
 gzip < ${img_path}${vm}.qcow2 > ${backup_path}${vm}.qcow2.gz
 echo "${vm} BACKUP DONE"
else
 echo "Invalid Selection... Aborting program"
 exit 1
fi
```

These screenshots display the contents of the backupVM.bash script. The script is responsible for:

- Identifying the target virtual machines (DEB1, DEB2, and DEB3)
- Performing automated backup operations
- Displaying status messages to confirm successful execution

This script demonstrates the use of Bash scripting to automate repetitive system administration tasks.

Screenshot 6: Successful Backup of DEB1, DEB2, and DEB3



```
pshah116@debhost:~$ sudo ./backupVM.bash
[sudo] password for pshah116:
sudo: ./backupVM.bash: command not found
pshah116@debhost:~$ cd bin
pshah116@debhost:~/bin$ sudo ./backupVM.bash
Backup all VMs? (y|n):y
Backing up VM deb1
deb1 BACKUP DONE
Backing up VM deb2
deb2 BACKUP DONE
Backing up VM deb3
deb3 BACKUP DONE
pshah116@debhost:~/bin$
```

This screenshot confirms that the backup script successfully backed up **DEB1**, **DEB2**, and **DEB3**. The output verifies that each system was processed correctly by the script.

This step confirms the effectiveness of the automation and validates the backup process.

Conclusion

In this lab, multiple Debian virtual machines were successfully updated, upgraded, and backed up using a custom Bash script. The lab demonstrated practical skills in Linux system maintenance, scripting, and automation. By ensuring systems were fully updated before performing backups, the lab followed best practices for system administration.

Overall, this lab reinforced the importance of automation in managing multiple systems and showcased the ability to write, execute, and verify scripts in a Linux environment.