

# PRIME COLLEGE

Khusibun, Kathmandu



## LAB REPORT OF INTRODUCTION TO CLOUD COMPUTING

**Submitted by:**

Name: Suyash Shrestha

Faculty: BSc.CSIT

Semester: 8<sup>th</sup>

**Submitted to:**

Sudan Prajapati

# CONTENTS

Lab No	Title/Question	Submission Date	Signature
1	Introduction to MapReduce		
2	Configuring RSA Authentication in VMware with CentOS		
3	Understanding File Permissions: chmod in Linux		
4	User and Group Management in Linux		
5	Automating Tasks with Cron Jobs in Linux		
6	Installing and Configuring Apache and MySQL Server on Linux		
7	Setting Up and Managing Databases with phpMyAdmin		
8	Laravel Installation and Setup		
9	Replication of VM		

# Lab 1

## Title: Introduction to MapReduce Programming

### 1. Objectives:

- To understand the concept of MapReduce and its role in distributed computing.
- To implement a basic MapReduce program for word count.
- To explore parallel processing using Python's multiprocessing module

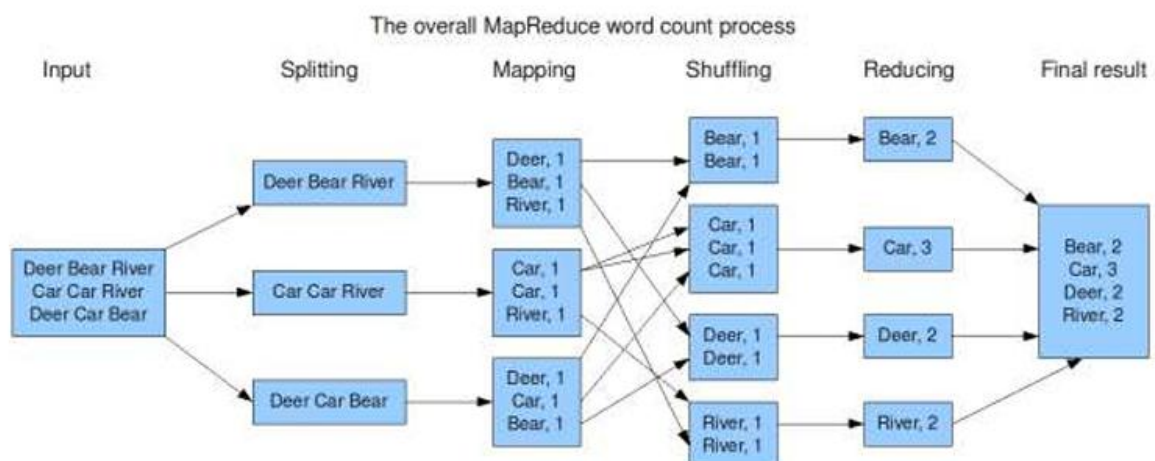
### 2. Theory:

#### MapReduce:

MapReduce is a distributed computing framework for processing large datasets in parallel across a cluster. It consists of three main stages: Mapping (converting input into key value pairs), Shuffling/Sorting (grouping keys together) and Reducing (aggregating results), making it efficient for big data processing in systems like Hadoop.

#### Steps in the MapReduce Process:

1. Input Splitting: The input text is divided into chunks across multiple nodes.
2. Mapping: Each chunk is processed by a Mapper, producing (word, 1) key-value pairs.
3. Shuffling & Sorting: The key-value pairs are grouped by word.
4. Reducing: The counts for each word are summed up.
5. Final Output: The result is stored in a distributed file system (like HDFS in Hadoop).



### 3. Source Code:

```
import multiprocessing
from collections import defaultdict

class MapReduce:
    def __init__(self, num_workers=2):
        self.num_workers = num_workers

    def map(self, chunk):
        word_counts = defaultdict(int)
        for word in chunk.split():
            word_counts[word] += 1
        return list(word_counts.items())

    def shuffle_sort(self, mapped_data):
        shuffled_data = defaultdict(list)
        for sublist in mapped_data:
            for word, count in sublist:
                shuffled_data[word].append(count)
        return shuffled_data

    def reduce(self, shuffled_data):
        return {word: sum(counts) for word, counts in shuffled_data.items()}

    def execute(self, text):
        chunks = text.split("\n")
        # Step 1: Map Phase (Parallel Processing)
        with multiprocessing.Pool(self.num_workers) as pool:
            mapped_data = pool.map(self.map, chunks)
        # Step 2: Shuffle and Sort Phase
        shuffled_data = self.shuffle_sort(mapped_data)
        # Step 3: Reduce Phase
        final_result = self.reduce(shuffled_data)
        return final_result

if __name__ == "__main__":
    text_corpus = """"Deer Bear River Car Car River Deer Car Bear""""
    mr = MapReduce(num_workers=3)
    result = mr.execute(text_corpus)
    print("Final Word Count:", result)
```

## **Output:**

```
■ ~\Personal\CSIT_Labs\8th_Semester\Advanced_Database  
> python .\map_reduce.py  
Final Word Count: {'Deer': 2, 'Bear': 2, 'River': 2, 'Car': 3}  
  
[16:05] Shell main
```

## **4. Outcome:**

To gain hands-on practical knowledge in implementing the MapReduce programming model, understanding its phases and executing a word count program using parallel processing in Python.

## **5. Conclusion:**

By following these steps, we successfully implemented a basic MapReduce program using Python. We executed the map, shuffle and reduce phases to process data in parallel, demonstrating how MapReduce enhances efficiency in large-scale data processing. This approach is widely used in distributed computing frameworks like Apache Hadoop.

## Lab 2

### **Title: Configuring RSA Authentication in VMware with CentOS**

#### **1. Objectives:**

- To understand the concept of VMware in virtualization.
- To learn how to install CentOS in a virtualized environment using VMware
- To understand the concept of RSA authentication and its importance in secure remote access.
- To configure RSA-based SSH authentication in a CentOS virtual machine running in VMware by implementing key-based authentication instead of password-based.

#### **2. Theory:**

##### **VMware:**

VMware is a leading software provider for virtualization technology, enabling multiple operating systems to run on a single physical machine. It provides hypervisors such as VMware Workstation, which create and manage virtual machines.

##### **Virtualization:**

Virtualization is the process of creating a virtual instance of a computer system, including hardware, storage, and networking, using a hypervisor. This allows multiple virtual machines to operate independently on the same physical server, improving resource utilization, flexibility and scalability.

##### **RSA Authentication:**

RSA (Rivest-Shamir-Adleman) is a cryptographic algorithm used for secure communication and authentication. In SSH (Secure Shell) authentication, RSA key pairs (public and private keys) replace traditional password-based authentication, enhancing security by preventing brute-force attacks and unauthorized access.

##### **Public-Key Cryptography:**

Public-key cryptography is a method where a key pair is generated:

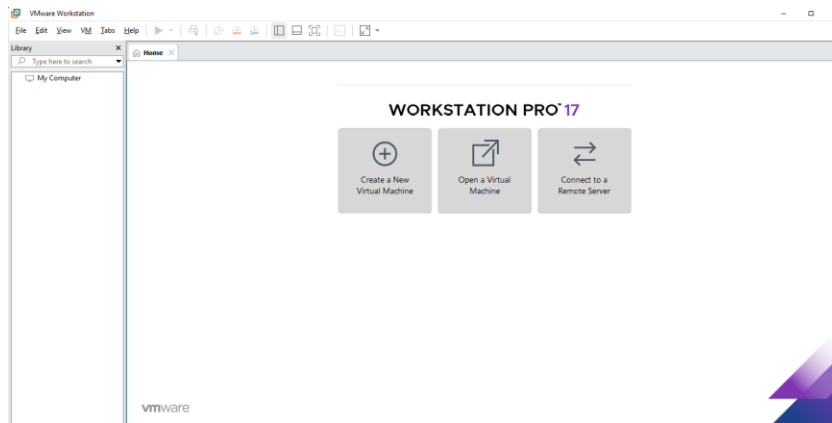
- **Public Key:** Shared with the server to allow authentication.
- **Private Key:** Kept secure on the client machine and never shared.

When a user tries to log in, the server verifies the user's identity by checking if they own the correct private key corresponding to the stored public key.

### 3. Steps:

#### A. Setting up CentOS in VMware:

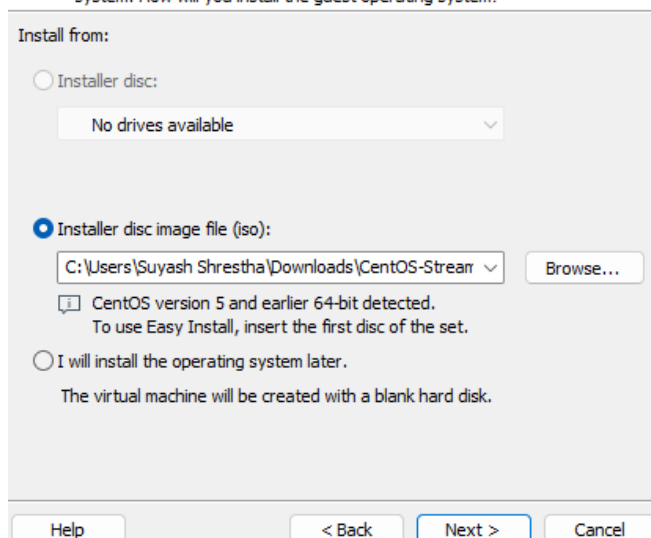
1. Install VMware Workstation Pro on your system.



2. Download the CentOS ISO image from the official website.
3. Create a new virtual machine in VMware and attach the CentOS ISO.

##### Guest Operating System Installation

A virtual machine is like a physical computer; it needs an operating system. How will you install the guest operating system?



4. Follow the installation steps to install CentOS.
5. Verify the network configuration.

```
suyash@localhost:~$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.49.128 netmask 255.255.255.0 broadcast 192.168.49.255
    inet6 fe80::20c:29ff:feef:8637 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:ef:86:37 txqueuelen 1000 (Ethernet)
    RX packets 665 bytes 159207 (155.4 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 550 bytes 59935 (58.5 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

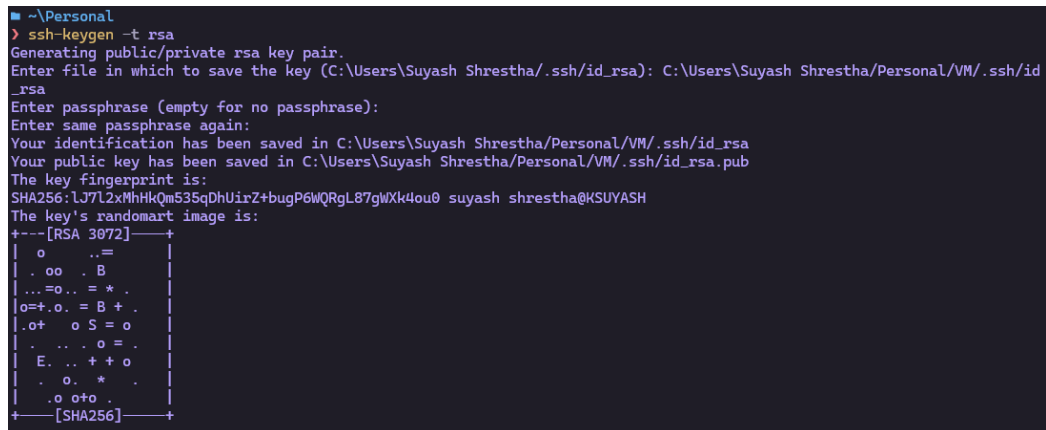
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 18 bytes 2112 (2.0 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 18 bytes 2112 (2.0 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

## B. Installing OpenSSH Server on CentOS:

1. Open the terminal in CentOS.
2. Install SSH server (if not already installed) using:
  - `sudo yum install -y openssh-server`
3. Start and enable the SSH service:
  - `Sudo systemctl start sshd`
  - `sudo systemctl enable sshd`
4. Check the status to ensure SSH is running:
  - `sudo systemctl status sshd`

## C. Generating an RSA Key Pair on the Client Machine:

1. On the client system (local machine), generate an RSA key pair.
  - `ssh-keygen -t rsa`
2. Save the key pair in the default location (`~/.ssh/id_rsa`).



```
~\Personal
> ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (C:\Users\Suyash Shrestha\.ssh\id_rsa): C:\Users\Suyash Shrestha\Personal\VM\.ssh\id_rsa
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in C:\Users\Suyash Shrestha\Personal\VM\.ssh\id_rsa
Your public key has been saved in C:\Users\Suyash Shrestha\Personal\VM\.ssh\id_rsa.pub
The key fingerprint is:
SHA256:1J7L2xMhHkQm535qDhUirZ+bugP6wQRgL87gWk4ou0 suyash shrestha@KSUYASH
The key's randomart image is:
+---[RSA 3072]---+
|  o  ..=
| . oo . B
| ...o.. = *
|o=+.o. = B +
|.ot o S = o
| . . . o =
| E. . . + o
| . o. *
|.o o+o
+---[SHA256]---
```

## D. Copying the Public key to the CentOS Server:

1. Copy the key using:
  - `cat ~/.ssh/id_rsa.pub | ssh root@192.168.49.128 "mkdir -p ~/.ssh && cat >> ~/.ssh/authorized_keys"`
2. Ensure correct permissions on the server:
  - `chmod 700 ~/.ssh`
  - `chmod 600 ~/.ssh/authorized_keys`

## E. Verifying RSA Authentication:

1. Now, try logging into the CentOS server from the local machine:
  - `ssh root@192.168.49.128`
2. If configured correctly, you can properly login without entering a password.



#### **4. Outcome:**

To gain hands-on practical knowledge in configuring RSA authentication in a virtualized CentOS environment using VMware. This involves setting up SSH key-based authentication, improving security and verifying authentication without passwords.

#### **5. Conclusion:**

By following these steps, we successfully configured RSA authentication on a CentOS virtual machine. This setup enhances security by replacing password-based authentication with key-based access. The implementation of public-key cryptography ensures a secure and efficient method for authenticating users in remote access environments.

## Lab 3

### Title: Understanding File Permissions: chmod in Linux

#### 1. Objectives:

- To understand the concept of file and directory permissions in Linux.
- To learn how to read, interpret and modify file permissions using chmod.
- To explore symbolic and numeric methods of setting permissions.

#### 2. Theory:

##### File Permissions in Linux:

Linux is a multi-user operating system that ensures security and access control through file permissions. Each file or directory has a set of permissions that determine who can read, write, or execute it.

There are three types of users for each file:

1. Owner (User) - usually the creator of the file
2. Group - users belonging to the same group as the file
3. Others - all other users

Each user type has three types of permissions:

PERMISSION	SYMBOL	NUMERIC
READ	r	4
WRITE	w	2
EXECUTE	x	1
NO ACCESS	-	0

Combining the values gives total permission. For example:

- $rwX = 4+2+1 = 7$  (full permission)
- $r-x = 4+0+1 = 5$  (read and execute)

Permissions for a file may look like this:

- `-rwxr-x---`

This means:

- Owner: **rwX** → full access (7)
- Group: **r-x** → read and execute (5)
- Others: **---** → no access (0)

This can be represented numerically as: 750

## chmod Command:

The chmod (change mode) command is used to change permissions of files and directories.

It supports **two methods**:

### Method I: Symbolic Notation

- u = user/owner
- g = group
- o = others
- a = all (user + group + others)
- Operators:
  - + → add permission
  - - → remove permission
  - = → assign permission

### Examples:

```
chmod u+x file1      # Give execute permission to user
chmod g-w file1      # Remove write permission from group
chmod o=r file1      # Set others to read-only
chmod a+rw file1     # Give read and write to all
```

### Method II: Numeric Notation

Each user's permission is represented by a single digit:

```
chmod 750 suyash.txt # Owner: rwx (7), Group: r-x (5), Others: --- (0)
chmod 741 suyash.txt # Owner: rwx (7), Group: r-- (4), Others: --x (1)
```

## Changing Ownership

Apart from permissions, ownership can also be modified:

```
chown <user>:<group> <file>
```

## 3. Steps:

### 1. Check Current Permissions

- `ls -l suyash.txt`  
-rw-r--r--. 1 suyash suyash 7 Apr 4 11:29 suyash.txt

### 2. Change Permissions Using Numeric Method

- `chmod 750 suyash.txt`  
-rwxr-x---. 1 suyash suyash 7 Apr 4 11:29 **suyash.txt**

Breakdown of 750:

USER TYPE	PERMISSION	EXPLANATION
OWNER	7 → rwx	Full Access
GROUP	5 → r-x	Read and execute
OTHERS	0 → ---	No access

### 3. Change Permissions Using Symbolic Method

- `chmod u+r newuser`
- `chmod g+r newuser`
- `chmod o-r newuser`
- `chmod a+x newuser`

### 4. Assign Mixed Permissions

- `chmod 741 runscript.sh` #Owner: `rwX(7)`, Group: `r--(4)`, Others: `--X(1)`

### 5. Grant Execute Permission for Shell Script

- `chmod +x myscript.sh`
- `./myscript.sh`

### 6. Practice Permission Combinations

- `chmod 700 secret.txt` # Only owner can access
- `chmod 755 script.sh` # Everyone can read/execute, only owner writes
- `chmod 000 restricted.log` # No one has access

## 4. Outcome:

To gain practical experience in reading and modifying file permissions in Linux using both symbolic and numeric modes of the `chmod` command, understanding how access is granted to different user roles.

## 5. Conclusion:

This lab provided hands-on practice with Linux file permissions. We learned how to set and modify permissions for users, groups, and others using symbolic and numeric notation. Understanding file permissions is fundamental to securing files and managing access control in a multi-user Linux environment.

## Lab 4

### Title: User and Group Management in Linux

#### 1. Objectives:

- To understand user and group management in Linux.
- To learn how to create, modify and delete user accounts and groups.
- To manage file permissions and ownership using chown.

#### 2. Theory:

##### User and Group Management in Linux:

Linux is a multi-user operating system where users and groups help in managing access and security. Each user has a unique User ID and belongs to at least one group. Groups allow administrators to define access control more efficiently.

- **User Management Commands:** useradd, usermod, and userdel help create, modify, and remove users.
- **Group Management Commands:** groupadd and groupdel are used to manage groups.

Proper user and group management is crucial for maintaining system security and organization.

##### File Permissions in Linux:

File permissions determine who can read, write or execute a file. They are represented as:

- **Read (r)**
- **Write (w)**
- **Execute (x)**

These permissions apply to three categories:

1. **Owner (u)** – The user who owns the file.
2. **Group (g)** – Other users in the same group.
3. **Others (o)** – All other users.

##### Changing File Ownership (chown) in Linux:

The chown command is used to change the ownership of a file or directory. Ownership can be changed for:

- **User ownership** (sudo chown user file)
- **Group ownership** (sudo chown :group file)
- **Both user and group ownership** (sudo chown user:group file)

### 3. Steps:

#### 1. Create Groups

- a. To create new groups, use:
  - `sudo groupadd staff`
  - `sudo groupadd sales`
  - `sudo groupadd marketing`

#### 2. View Existing Groups

- a. To check all groups available in the system:
  - `cat /etc/group`  
`staff:x:1001:`  
`sales:x:1002:`  
`marketing:x:1003:`

#### 3. Create a New User

- a. To create a new user named suyashshrestha:
  - `sudo useradd suyashshrestha`
- b. Set a password for the user:
  - `sudo passwd suyashshrestha`  

```
suyash@localhost:~$ sudo useradd suyashshrestha
suyash@localhost:~$ sudo passwd suyashshrestha
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: password updated successfully
```
- c. Switch to the new user:
  - `su suyashshrestha`
- d. Verify the current working directory:
  - `pwd`  

```
suyashshrestha@localhost:/home/suyash$ pwd
/home/suyash
```

#### 4. Add User to Groups

- a. Add user to a primary group and multiple secondary groups during creation:
  - `sudo useradd -g staff -G sales,marketing newuser`
- b. To add an existing user to additional groups:
  - `sudo usermod -aG sales,marketing newuser`
- c. Verify the user's group membership:
  - `grep newuser /etc/passwd`  

```
suyash@localhost:~$ grep newuser /etc/passwd
newuser:x:1002:1001:./home/newuser:/bin/bash
```

- `grep sales /etc/group`  
`suyash@localhost:~$ grep sales /etc/group`  
`sales:x:1002:newuser`

## 5. Modify User Details

- Change the primary group of a user:
  - `sudo usermod -g sales suyashshrestha`
- Check the updated group:
  - `id suyashshrestha`  
`suyash@localhost:~$ id suyashshrestha`  
`uid=1001(suyashshrestha) gid=1002(sales) groups=1002(sales)`

## 6. Change File Ownership with chown

- Create a test file:
  - `touch suyash.txt`
- Check the current owner of the file:
  - `ls -l suyash.txt`
- Change the owner of the file to suyashshrestha:
  - `sudo chown suyashshrestha suyash.txt`
- Change the group ownership to sales:
  - `sudo chown :sales suyash.txt`
- Change both the user and group ownership:
  - `sudo chown suyashshrestha:sales suyash.txt`
- Verify the changes:
  - `ls -l suyash.txt`

## 7. Delete a User

- To remove a user and their home directory:
  - `sudo userdel -r suyashshrestha`

## 4. Outcome:

To gain hands-on experience in creating and managing user accounts, groups, file permissions and file ownership in Linux, improving system administration skills.

## 5. Conclusion:

By following these steps, we successfully managed user accounts and groups in Linux, assigned appropriate permissions and changed file ownership. This ensures proper access control and enhances system security in a multi-user environment.

## Lab 5

### Title: Automating Tasks with Cron Jobs in Linux

#### 1. Objectives:

- To understand the purpose and usage of cron jobs in Linux.
- To learn how to create, schedule and manage automated tasks using cron service.
- To configure different scheduling options for periodic task execution.

#### 2. Theory:

##### Cron Jobs:

Cron is a time-based job scheduler in Unix-like operating systems that automates repetitive tasks. The cron service allows users to schedule scripts or commands to run at specified intervals (minutes, hours, daily, weekly, etc.).

##### Crontab:

Crontab (Cron table) is a configuration file where scheduled tasks are defined. Each user can have a separate crontab file, and system-level tasks can also be managed through global cron jobs.

##### Cron Syntax:

A cron job follows the format:

```
* * * * * <command_to_execute>
```

-	-	-	-	-	
					Day of the week (0 - 6) [Sunday = 0]
					Month (1 - 12)
					Day of the month (1 - 31)
					Hour (0 - 23)
					Minute (0 - 59)

##### Example:

```
30 2 * * * /home/user/backup.sh
```

This runs the backup.sh script every day at 2:30 AM.

#### 3. Steps:

1. Check if Cron is Installed and Running
  - a. Verify cron installation:
    - `crontab -l`
    - If cron is not installed, install it using:
      - `sudo yum install cronie -y`
  - b. Start and enable the cron service:
    - `sudo systemctl start crond`
    - `sudo systemctl enable crond`



## 2. Create a Basic Cron Job

- a. To schedule a cron job, edit the crontab file:
  - `crontab -e`
- b. Add a job to execute a script every day at 3 AM:
  - `0 3 * * * /home/suyash/runscript.sh`
- c. Save and exit the editor.

## 3. List and Remove Cron Jobs

- a. View all scheduled cron jobs:
  - `crontab -l`  

```
suyash@localhost:~/Desktop$ crontab -l
0 3 * * * /home/suyash/runscript.sh
```

- b. Remove a cron job:

- `crontab -r`

## 4. Schedule Different Types of Tasks

- a. Run a script every hour:
  - `0 * * * * /home/user/hourly_task.sh`
- b. Clear logs every Sunday at midnight:
  - `0 0 * * 0 echo "" > /var/log/syslog`
- c. Send a reminder email every Monday at 9 AM:
  - `0 9 * * 1 echo "Weekly Report Reminder" | mail -s "Reminder" suyash@gmail.com`

## 5. Verify Cron Job Execution

- a. Check cron logs to confirm job execution:
  - `cat /var/log/cron | grep CRON`
- b. Run the job manually to test if the script works:
  - `bash /home/suyash/runscript.sh`

## 4. Outcome:

To successfully automate tasks using cron jobs in Linux, understanding the crontab syntax, scheduling tasks and verifying their execution.

## 5. Conclusion:

By following these steps, we successfully automated tasks using cron jobs in Linux. This setup helps in scheduling system maintenance, backups, and periodic job execution, improving system efficiency and reducing manual intervention.

## Lab 6

### **Title: Installing and Configuring Apache and MySQL Server on Linux**

#### **1. Objectives:**

- To learn how to install and configure Apache web server. And MySQL Server.
- To understand the basic functionalities of web servers and database servers.
- To deploy and manage a basic web server environment and database system.

#### **2. Theory:**

##### **Web Servers:**

Web servers are software applications that serve web pages to users over the internet or a local network. They process HTTP requests and deliver HTML, CSS, JavaScript, and other files to web browsers.

##### **Apache Web Server:**

Apache is one of the most widely used open-source web servers. It follows a process-driven architecture, handling multiple connections using a multi-threaded approach. Apache is known for its extensive module support, flexibility, and compatibility with various operating systems.

##### **MySQL Server:**

MySQL is a widely used open-source relational database management system that utilizes Structured Query Language (SQL) for managing and manipulating data. It has grown in popularity due to its reliability, speed and ease of use. MySQL supports various OS, making it a versatile choice for many applications.

#### **3. Installation Steps:**

##### **A. Installing Apache Web Server:**

1. Install Apache:
  - `sudo yum install httpd`
2. Start Apache:
  - `sudo systemctl start httpd`
3. Enable Apache to start on boot:
  - `sudo systemctl enable httpd`
4. Modify the web page:
  - `sudo nano /var/www/html/index.html`

- Insert the following HTML content:

```
<!DOCTYPE html>
<html>
<head>
  <title>Welcome to Suyash's site</title>
</head>
<body>
  <h1>I am Suyash Shrestha</h1>
  <p>Hello!!! </p>
</body>
</html>
```

5. Set permissions for Apache:

- `sudo chown -R apache:apache /var/www/html`  
`sudo chmod -R 555 /var/www/html`

6. Restart Apache:

- `sudo systemctl restart httpd`

7. Allow Apache through the firewall:

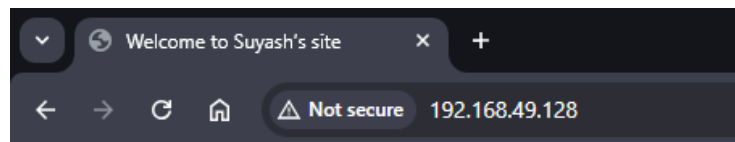
- `sudo firewall-cmd --zone=public --add-service=http --permanent`  
`sudo firewall-cmd --reload`

8. Open a browser outside the VM and test Apache by entering the VM's IP:

- `http://192.168.49.128`

Or, test inside the VM:

- `http://localhost`



**I am Suyash Shrestha**

Hello!!!

## B. Installing MySQL Server:

1. Update the system:

- `sudo yum update`

2. Install MySQL Server:

- `sudo yum install mysql-server`

3. Start the MySQL service:

- `sudo systemctl start mysqld`

4. Enable MySQL to start on boot:

- `sudo systemctl enable mysqld`

5. Secure the MySQL installation:

- `sudo mysql_secure_installation`

Follow the prompts to set up a root password, remove anonymous users, disable remote root login and remove the test database.

6. Log in to MySQL:

- `mysql -u root -p`

```
suyashshrestha@localhost:~/Desktop$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 13
Server version: 8.4.2 Source distribution

Copyright (c) 2000, 2024, Oracle and/or its affiliates.
```

7. Exit MySQL:

- `exit`

#### 4. Outcome:

To successfully install and configure Apache web server and MySQL Server on a Linux system, understanding how web servers and database servers' function. This lab provides hands-on experience in setting up a web server, hosting a simple webpage and managing a database system.

#### 5. Conclusion:

By following these instructions, we successfully installed, configured and tested Apache and MySQL Server on a Linux-based system. We deployed a simple webpage using Apache and set up a MySQL database.

## Lab 7

### Title: Setting Up and Managing Databases with phpMyAdmin

#### 1. Objectives:

- To understand the installation and configuration of phpMyAdmin on CentOS.
- To learn how to manage MySQL/MariaDB databases using phpMyAdmin.
- To enhance database administration skills through a web-based interface.

#### 2. Theory:

##### phpMyAdmin:

phpMyAdmin is a free and open-source web-based application written in PHP that provides a user-friendly interface for managing MySQL, MariaDB databases. It allows database administrators and users to perform various tasks such as creating, modifying, deleting databases, tables and records, as well as executing SQL queries.

##### Web Server Authentication:

Implementing authentication at the web server level adds an extra layer of security by requiring users to authenticate before accessing web applications. This is typically achieved using HTTP Basic Authentication, which prompts users for a username and password before granting access to the application.

#### 3. Steps:

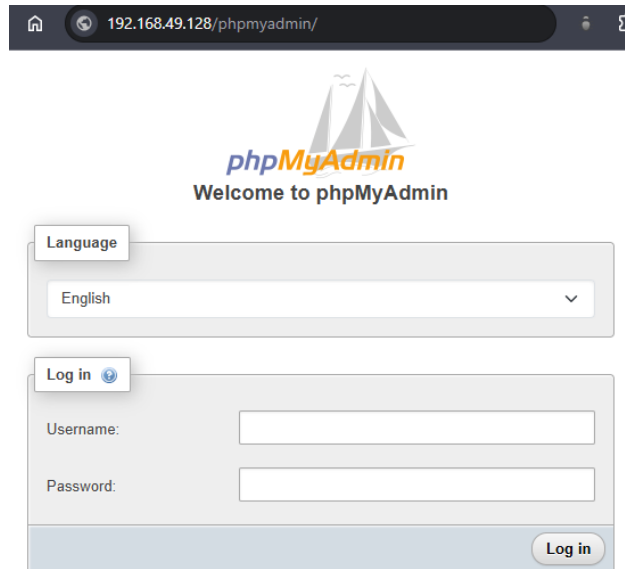
1. Install the EPEL Repository:
  - a. Enable Extra Packages for Enterprise Linux repo to access additional packages:
    - `sudo yum install epel-release`
2. Install phpMyAdmin
  - a. Install phpMyAdmin along with its dependencies:
    - `sudo yum install phpMyAdmin`
3. Configure phpMyAdmin
  - a. Open the phpMyAdmin configuration file for Apache:
    - `sudo nano /etc/httpd/conf.d/phpMyAdmin.conf`
  - b. By default, phpMyAdmin is configured to allow access only from the local machine. To allow access from specific IP addresses, locate the <RequireAny> section and add the following line:
    - `<RequireAny>Require ip 192.168.49.128</RequireAny>`
  - c. Save and exit the editor.
4. Restart Apache Web Server

a. Restart the Apache service to apply the changes:

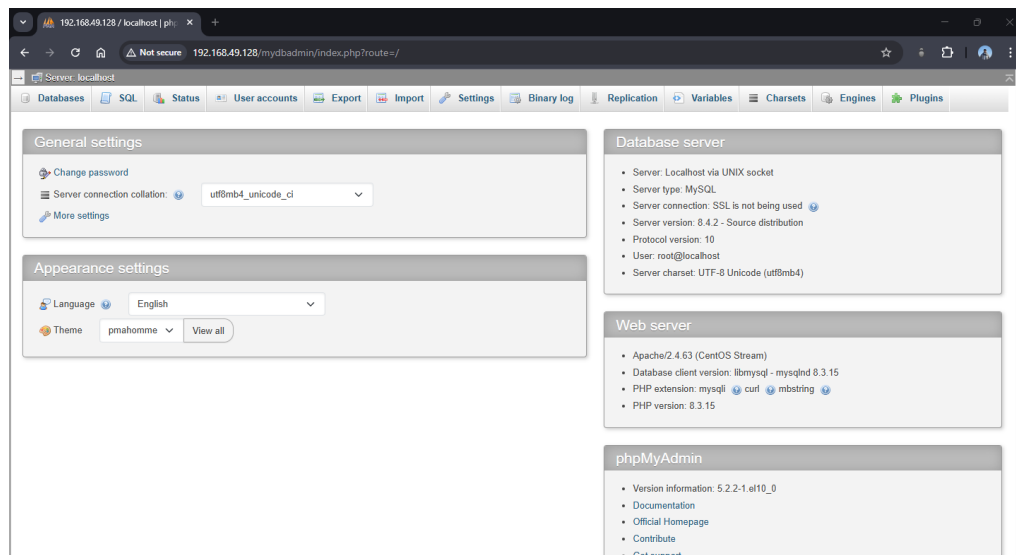
- `sudo systemctl restart httpd`

## 5. Access phpMyAdmin

a. Open a web browser and navigate to `http://192.168.49.128/phpmyadmin/`.



b. Log in using your MySQL/MariaDB username and password.



## 6. Secure phpMyAdmin

a. For enhanced security, consider changing the alias of the phpMyAdmin interface. Open the phpMyAdmin Apache configuration file:

- `sudo nano /etc/httpd/conf.d/phpMyAdmin.conf`

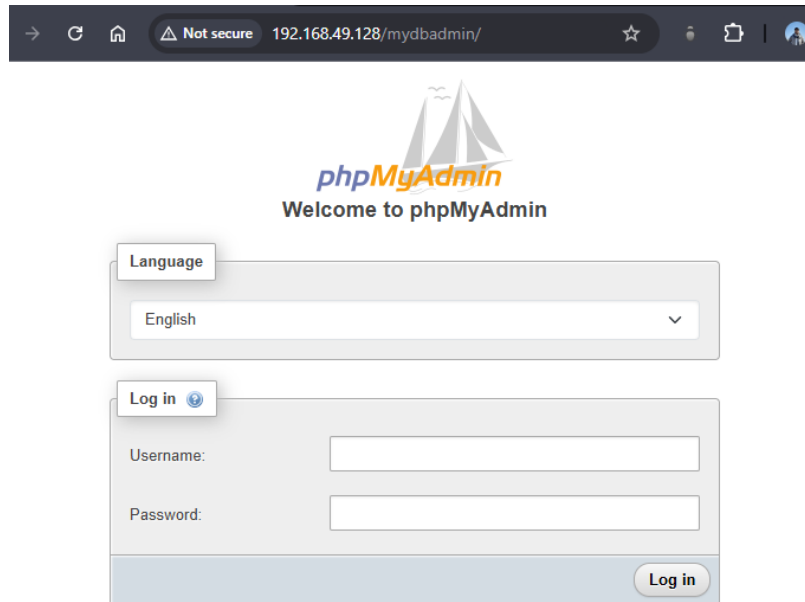
b. Locate the line:

- `Alias /phpMyAdmin /usr/share/phpMyAdmin`

c. Change /phpMyAdmin to a custom alias, for example:

- `Alias /mydbadmin /usr/share/phpMyAdmin`

- d. Save and exit the editor.
- e. Restart Apache to apply the changes:
  - `sudo systemctl restart httpd`
- f. Access phpMyAdmin using the new alias: `http://192.168.48.126/mydbadmin`.



#### 7. Set Up Web Server Authentication:

- a. Modify Apache Configuration: In the same phpMyAdmin Apache configuration file (`/etc/httpd/conf.d/phpMyAdmin.conf`), within the `<Directory /usr/share/phpMyAdmin/>` block, add the following directive:
  - `AllowOverride All`
- b. Save and close the file.
- c. Restart Apache:
  - `sudo systemctl restart httpd`
- d. Create .htaccess File: Create a .htaccess file in the phpMyAdmin directory:
  - `sudo nano /usr/share/phpMyAdmin/.htaccess`Add the following content:

```
AuthType Basic
AuthName "phpMyAdmin Login"
AuthUserFile /etc/httpd/pma_pass
Require valid-user
```
- e. Save and close the file.
- f. Create Password File: Use the `htpasswd` utility to create a password file and add a user:
  - `sudo htpasswd -c /etc/httpd/pma_pass suyash_p`

You'll be prompted to enter and confirm a password for the user. To add additional users:

- `sudo htpasswd /etc/httpd/pma_pass anotheruser`

g. Set Permissions: Ensure the password file has appropriate permissions.

#### **4. Outcome:**

To gain hands-on experience in installing and configuring phpMyAdmin, managing MySQL databases through a web-based interface and implement basic security measures.

#### **5. Conclusion:**

By following these steps, we successfully installed and configured phpMyAdmin, allowing for efficient and user-friendly management of MySQL databases. Additionally, we implemented basic security practices to restrict access and customize the phpMyAdmin interface, enhancing the overall security of the database management system.



## Lab 8

### Title: Laravel Installation and Setup

#### 1. Objectives:

- To learn how to install and configure Laravel on a CentOS system.
- To understand the prerequisites required for Laravel installation.
- To successfully deploy a Laravel application using Apache.

#### 2. Theory:

##### Laravel Framework:

Laravel is a popular open-source PHP framework used for building web applications. It follows the Model-View-Controller architectural pattern and provides built-in features like authentication, routing and database management, making development efficient. Laravel utilizes Composer, a dependency manager, to handle package installations and updates.

##### LAMP Stack:

Laravel requires a working LAMP (Linux, Apache, MySQL, PHP) environment. The LAMP stack enables PHP-based applications to run on a Linux server with Apache handling HTTP requests and MySQL managing the database.

##### Composer:

Composer is a dependency manager for PHP that allows users to install and manage Laravel and its required packages efficiently.

#### 3. Installation Steps:

1. Update the System
  - a. Before installing Laravel, update the CentOS system repositories:
    - `sudo yum update -y`
2. Install Apache, MySQL and PHP (LAMP)
  - a. To run Laravel, install the LAMP stack using:
    - `sudo yum install httpd mysql-server php -y`
  - b. Start and enable services:
    - `sudo systemctl start httpd mysqld`
    - `sudo systemctl enable httpd mysqld`
  - c. Secure MySQL:
    - `sudo mysql_secure_installation`
3. Install Composer
  - a. Laravel requires Composer for managing dependencies. Install it using:

- `curl -sS https://getcomposer.org/installer | php`
- `sudo mv composer.phar /usr/local/bin/composer`

```
suyash@localhost:~$ curl -sS https://getcomposer.org/installer | php
All settings correct for using Composer
Downloading...

Composer (version 2.8.7) successfully installed to: /home/suyash/composer.phar
Use it: php composer.phar
```

#### 4. Install Laravel Using Composer

##### a. Install:

- `composer global require laravel/installer`

##### b. Ensure ~/.composer/vendor/bin is added to the PATH:

- `echo 'export PATH="$HOME/.composer/vendor/bin:$PATH"' >> ~/.bashrc`
- `source ~/.bashrc`

#### 5. Create a Laravel Project

##### a. Navigate to the web directory and create a new Laravel application:

- `cd /var/www`
  - `laravel new portfolio`
- ```
suyash@localhost:/var/www$ laravel new portfolio
```



Which starter kit would you like to install?

None

Which testing framework do you prefer?

Pest

```
Creating a "laravel/laravel" project at "./portfolio"
Installing laravel/laravel (v12.0.4)
```

- `cd portfolio`

#### 6. Set Permissions:

##### a. To ensure Laravel has the necessary access rights, set permissions:

- `sudo chown -R apache:apache /var/www/portfolio`
- `sudo chmod -R 775 /var/www/portfolio/storage`
- `sudo chmod -R 775 /var/www/portfolio/bootstrap/cache`

#### 7. Configure Apache Virtual Host

##### a. Edit the Apache configuration file:

- `sudo nano /etc/httpd/conf.d/laravel.conf`

Add the following configuration:

```
<VirtualHost *:80>
```

```
DocumentRoot /var/www/portfolio/public
<Directory /var/www/portfolio >
    AllowOverride All
    Require all granted
</Directory>
</VirtualHost>
```

b. Save and exit, then restart Apache:

- `sudo systemctl restart httpd`

## 8. Set Laravel Application Key

- `php artisan key:generate`

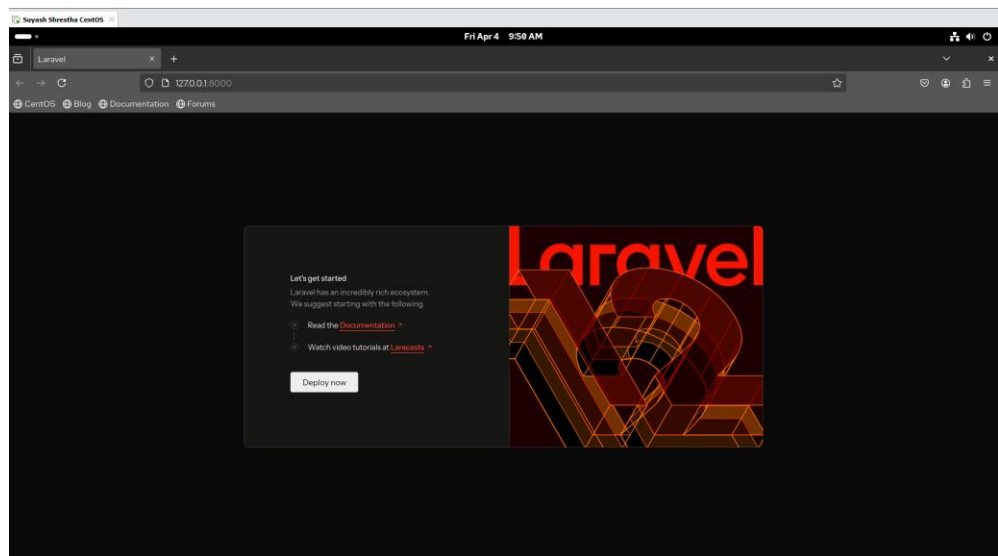
```
suyash@localhost:/var/www/portfolio$ php artisan key:generate
```

```
INFO Application key set successfully.
```

## 9. Test Laravel Installation

a. Access the Laravel application by entering the server's name

- `http://localhost:8000` in a web browser.



## 4. Outcome:

To successfully install, configure Laravel, understanding its prerequisites and environment setup, ensuring proper deployment on an Apache server and verifying the installation.

## 5. Conclusion:

By following these steps, Laravel was successfully installed and configured. A functional Laravel application was set up, demonstrating the use of Composer for dependency management, Apache for web hosting and MySQL for database management. This setup enables developers to build and deploy web applications efficiently within a structured PHP framework.

## Lab 9

### Title: Replication of Virtual Machine

#### 1. Objectives:

- To understand the concept and importance of Virtual Machine replication.
- To explore different types of VM replication methods.
- To perform basic replication of a VM using available virtualization tools.

#### 2. Theory:

##### VM Replication:

Virtual Machine replication is the process of creating and maintaining an exact copy of a virtual machine in a different physical or virtual environment. It is a critical aspect of disaster recovery and high availability in cloud and virtualized environments.

##### Types of VM Replication:

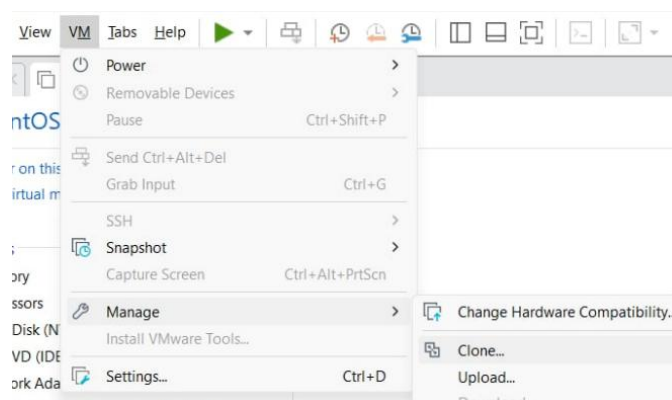
1. **Linked Clone:** A linked clone is a copy of a virtual machine that shares virtual disks with the parent VM in a read-only mode. It relies on the original VM's disk files and stores only the differences (delta data), making it faster to create and more space-efficient. However, it depends on the parent VM — if the parent is deleted or corrupted, the linked clone will not function properly.
2. **Full Clone:** A full clone is an independent, complete copy of a virtual machine, including all its virtual disk files. It operates entirely on its own, with no dependency on the source VM. While it takes longer to create and uses more storage space, it offers better performance and stability.

##### Use Cases:

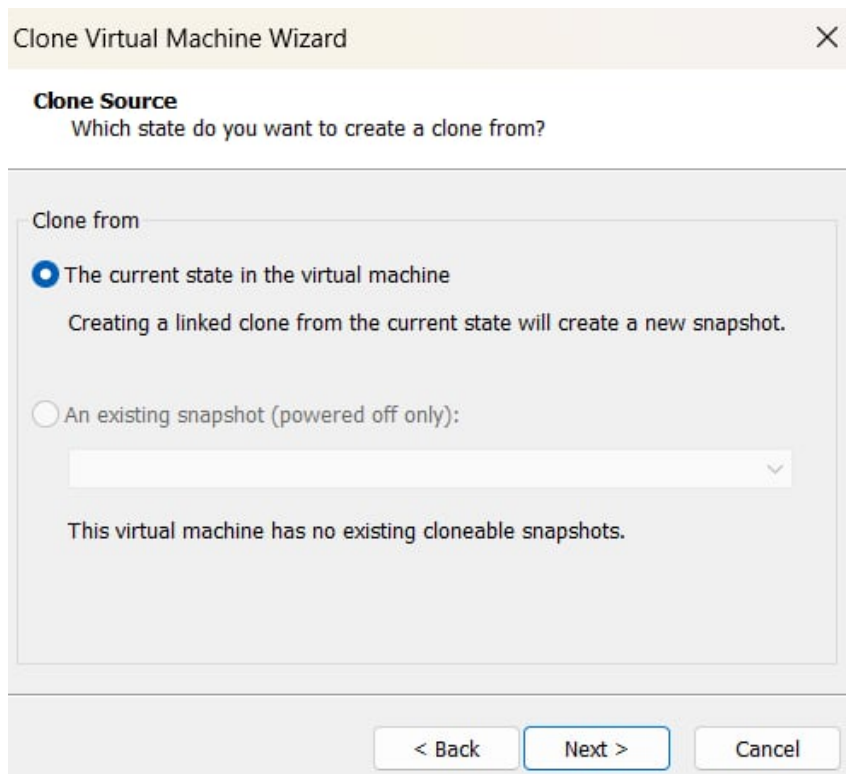
Disaster Recovery, Load Balancing, Migration, Backup & Restore

#### 3. Steps:

1. **Export VM:** Go to VM tab → Select Manage → Choose Clone.



2. A wizard is shown, click next and go on.



Clone Virtual Machine Wizard

**Clone Source**  
Which state do you want to create a clone from?

Clone from

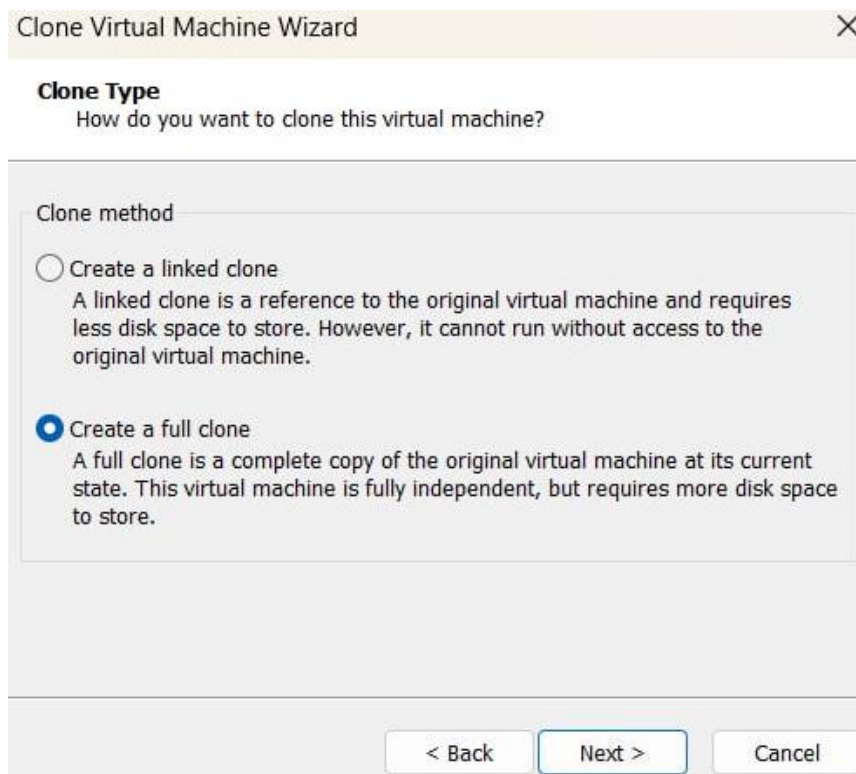
☒ The current state in the virtual machine  
Creating a linked clone from the current state will create a new snapshot.

☐ An existing snapshot (powered off only):  
[Empty dropdown menu]

This virtual machine has no existing cloneable snapshots.

< Back   Next >   Cancel

3. Create a full clone or linked clone:



Clone Virtual Machine Wizard

**Clone Type**  
How do you want to clone this virtual machine?

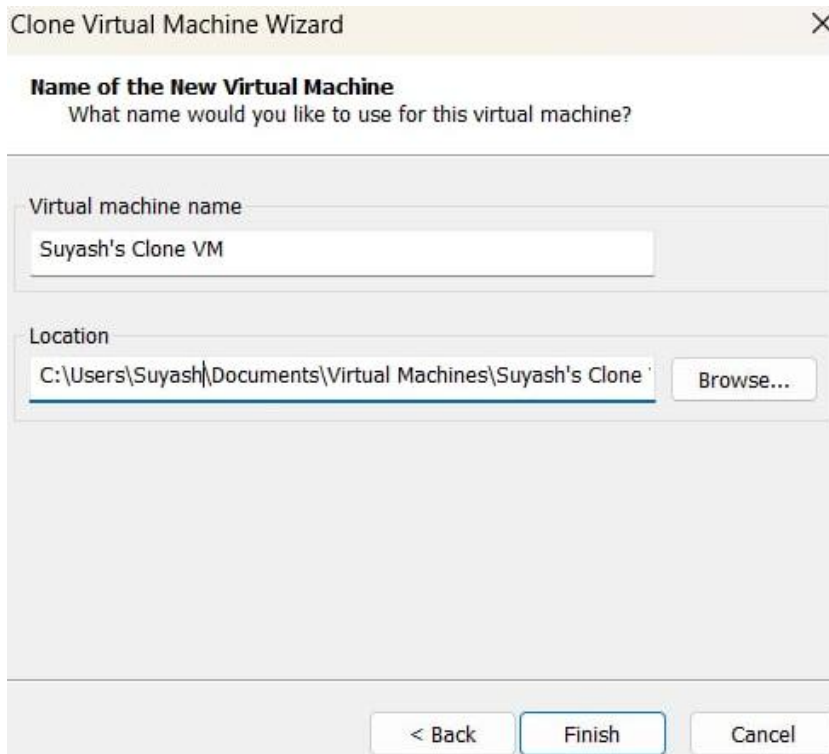
Clone method

☐ Create a linked clone  
A linked clone is a reference to the original virtual machine and requires less disk space to store. However, it cannot run without access to the original virtual machine.

☒ Create a full clone  
A full clone is a complete copy of the original virtual machine at its current state. This virtual machine is fully independent, but requires more disk space to store.

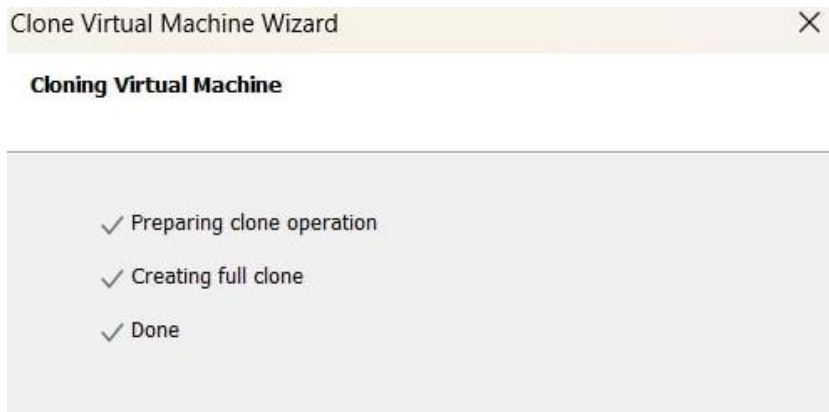
< Back   Next >   Cancel

4. Name the new clone VM:



The screenshot shows the 'Clone Virtual Machine Wizard' dialog box. The title bar says 'Clone Virtual Machine Wizard' with a close button. The main heading is 'Name of the New Virtual Machine' with a subtitle 'What name would you like to use for this virtual machine?'. There are two input fields: 'Virtual machine name' containing 'Suyash's Clone VM' and 'Location' containing 'C:\Users\Suyash\Documents\Virtual Machines\Suyash's Clone'. A 'Browse...' button is next to the location field. At the bottom are three buttons: '< Back', 'Finish', and 'Cancel'.

5. The cloning/replication process has been completed.



The screenshot shows the 'Clone Virtual Machine Wizard' dialog box. The title bar says 'Clone Virtual Machine Wizard' with a close button. The main heading is 'Cloning Virtual Machine'. Below it, there is a list of three steps, each with a checkmark: 'Preparing clone operation', 'Creating full clone', and 'Done'.

#### 4. Outcome:

To gain hands-on practical knowledge about concept of VM replication and perform a simple replication task using virtualization tools, reinforcing the importance of this process in ensuring system availability and disaster recovery.

#### 5. Conclusion:

By following these steps, we successfully replicated a virtual machine using manual export and import techniques. This process illustrated how VM replication ensures data safety, supports recovery strategies and enables operational continuity in virtualized environments. The knowledge gained here is fundamental in modern data center and cloud operations.