**Mini Project: Personal Linux Server Setup and Automation**

# Personal Linux Server Setup and Automation

## Overview

This project sets up a personal web server on an Amazon Linux EC2 instance, including automated backups. The process involves installing and configuring Apache, setting up a backup system, and automating these tasks using shell scripts.

## Prerequisites

* AWS account with permissions to create EC2 instances.
* SSH key pair for connecting to your EC2 instance.
* Basic knowledge of Linux command line and AWS management console.

## Steps to Set Up the Server

### 1. Launch an EC2 Instance

1. **Launch an Instance:**
   * Log in to your AWS Management Console.
   * Navigate to the EC2 Dashboard and click "Launch Instance".
   * Choose the Amazon Linux 2 AMI.
   * Select an instance type (e.g., t2.micro for free tier eligibility).
   * Configure the instance details (default settings are usually fine).
   * Add storage (default settings are usually fine).
   * Configure security groups:
     + Allow SSH (port 22) for your IP.
     + Allow HTTP (port 80) for web traffic.
   * Review and launch the instance. Save the private key file (.pem) when prompted.
2. **Connect to Your Instance:**
   * Open your terminal.
   * Connect to your instance using SSH:

ssh -i /path/to/your-key.pem ec2-user@your-ec2-public-dns

### 2. Install and Configure Apache

1. **Update the System:**

sudo yum update -y

1. **Install Apache:**

sudo yum install httpd -y

sudo systemctl start httpd

sudo systemctl enable httpd

1. **Host a Simple Website:**
   * Create a simple HTML file:

echo '<html><body><h1>Hello, World!</h1></body></html>' | sudo tee /var/www/html/index.html

* + Verify the website by navigating to your EC2 instance's public DNS in a web browser.

### 3. Set Up the Backup System

1. **Create a Backup Script:**
   * Create a script to back up your web server's content and configuration files:

sudo tee /usr/local/bin/backup\_script.sh << 'EOF'

#!/bin/bash

BACKUP\_DIR="/backup"

TIMESTAMP=$(date +"%F-%H%M")

LOGFILE="/var/log/backup\_script.log"

mkdir -p $BACKUP\_DIR/$TIMESTAMP

{

echo "Backup started at $(date)"

cp -r /var/www $BACKUP\_DIR/$TIMESTAMP

cp -r /etc/httpd $BACKUP\_DIR/$TIMESTAMP

echo "Backup completed at $(date)"

} >> $LOGFILE 2>&1

EOF

sudo chmod +x /usr/local/bin/backup\_script.sh

1. **Schedule the Backup:**
   * Install cronie if not already installed:

bash

Copy code

sudo yum install cronie -y

sudo systemctl start crond

sudo systemctl enable crond

* + Add a cron job to run the backup script daily at 2:00 AM:

(crontab -l 2>/dev/null; echo "0 2 \* \* \* /usr/local/bin/backup\_script.sh") | crontab -

### 4. Automate the Setup Process

1. **Create an Automation Script:**
   * Combine the setup and configuration into a single script:

sudo tee /usr/local/bin/setup\_script.sh << 'EOF'

#!/bin/bash

# Update system

sudo yum update -y

# Install Apache

sudo yum install httpd -y

sudo systemctl start httpd

sudo systemctl enable httpd

# Set up website

echo '<html><body><h1>Hello, World!</h1></body></html>' | sudo tee /var/www/html/index.html

# Set up backup script

sudo tee /usr/local/bin/backup\_script.sh << 'SCRIPT'

#!/bin/bash

BACKUP\_DIR="/backup"

TIMESTAMP=$(date +"%F-%H%M")

LOGFILE="/var/log/backup\_script.log"

mkdir -p $BACKUP\_DIR/$TIMESTAMP

{

echo "Backup started at $(date)"

cp -r /var/www $BACKUP\_DIR/$TIMESTAMP

cp -r /etc/httpd $BACKUP\_DIR/$TIMESTAMP

echo "Backup completed at $(date)"

} >> $LOGFILE 2>&1

SCRIPT

sudo chmod +x /usr/local/bin/backup\_script.sh

# Schedule backup script

(crontab -l 2>/dev/null; echo "0 2 \* \* \* /usr/local/bin/backup\_script.sh") | crontab -

EOF

sudo chmod +x /usr/local/bin/setup\_script.sh

1. **Run the Automation Script:**
   * Execute the setup script on your EC2 instance:

sudo /usr/local/bin/setup\_script.sh

### 5. Verification

1. **Verify the Web Server:**
   * Check the web server status:

sudo systemctl status httpd

* + Open a web browser and navigate to your EC2 instance's public DNS. You should see the "Hello, World!" message.

1. **Verify the Backup Script:**
   * Run the backup script manually:

sudo /usr/local/bin/backup\_script.sh

* + Check the backup directory:

sudo ls /backup/$(date +"%F-%H%M")

1. **Verify the Cron Job:**
   * Check the cron job schedule:

bash

Copy code

crontab -l

* + Check cron logs:

sudo cat /var/log/cron

### Additional Notes

* **Log Output:** The backup script logs its output to /var/log/backup\_script.log.
* **Security Groups:** Ensure your security group rules allow HTTP traffic on port 80 and SSH traffic on port 22.
* **Modify Backup Frequency:** Adjust the cron schedule as needed. For testing, you can set it to run every minute and revert it back to daily once verified.

## Conclusion

This project demonstrates setting up a web server on Amazon Linux, automating backups, and using cron for scheduling tasks. By following the steps and verifying each component, you ensure that the server is set up correctly and maintains regular backups.

**Instructions on how to test the setup and verify everything is working as expected**.

Below are detailed instructions on how to test the setup and verify that everything is working as expected.

## Testing and Verification Steps

### 1. Verify the Web Server Setup

#### Check Apache Service Status

1. **Log in to your EC2 instance:**

ssh -i /path/to/your-key.pem ec2-user@your-ec2-public-dns

1. **Check the status of the Apache service:**

sudo systemctl status httpd

* + The output should indicate that the service is active (running).

#### Access the Web Page

1. **Open a web browser.**
2. **Navigate to your EC2 instance's public DNS:**
   * Example: http://your-ec2-public-dns
   * You should see the "Hello, World!" message displayed on the page.

### 2. Verify the Backup Script

#### Run the Backup Script Manually

1. **Log in to your EC2 instance (if not already logged in).**
2. **Execute the backup script:**

sudo /usr/local/bin/backup\_script.sh

* + This should create a backup in the /backup directory.

#### Check the Backup Directory

1. **List the contents of the backup directory:**

sudo ls /backup/$(date +"%F-%H%M")

* + Replace $(date +"%F-%H%M") with the actual timestamp if needed.
  + You should see directories or files backed up from /var/www and /etc/httpd.

### 3. Verify the Cron Job

#### Check Cron Job Scheduling

1. **List the cron jobs for the current user:**

crontab -l

* + You should see the entry: 0 2 \* \* \* /usr/local/bin/backup\_script.sh

#### Check Cron Logs

1. **Check the cron logs to verify if the job ran:**

sudo cat /var/log/cron

* + Look for entries indicating that the backup\_script.sh was executed.

#### Test Cron Job by Modifying the Schedule

1. **Edit the crontab to run the script every minute for testing:**

(crontab -l 2>/dev/null; echo "\* \* \* \* \* /usr/local/bin/backup\_script.sh") | crontab -

1. **Wait a few minutes and then check the backup directory:**
2. sudo ls /backup/
   * You should see multiple backups created with different timestamps.
3. **Revert the cron job to its original schedule:**

(crontab -l 2>/dev/null; echo "0 2 \* \* \* /usr/local/bin/backup\_script.sh") | crontab -

### Additional Verification Steps

#### Check Backup Script Log

1. **Open the log file to review the backup script’s log entries:**

sudo cat /var/log/backup\_script.log

* + Ensure the log entries indicate successful backup operations.

#### Verify Script Permissions

1. **Ensure the setup and backup scripts are executable:**

sudo chmod +x /usr/local/bin/setup\_script.sh

sudo chmod +x /usr/local/bin/backup\_script.sh

### Summary

By following these steps, you can ensure that:

1. The Apache web server is installed and running.
2. The backup script is functioning correctly and creating backups.
3. The cron job is scheduled properly and executes the backup script at the specified times.

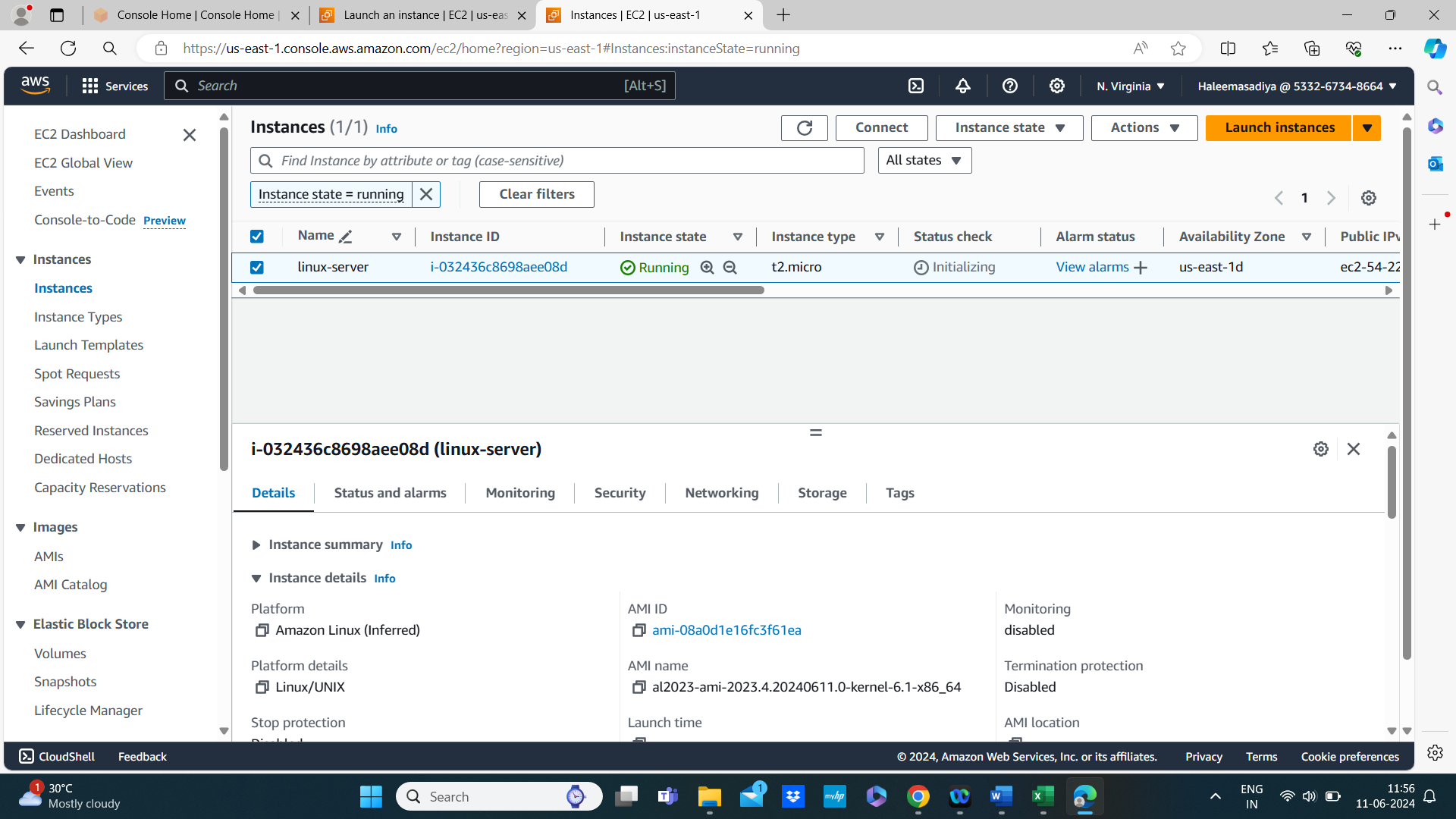
### Final Notes

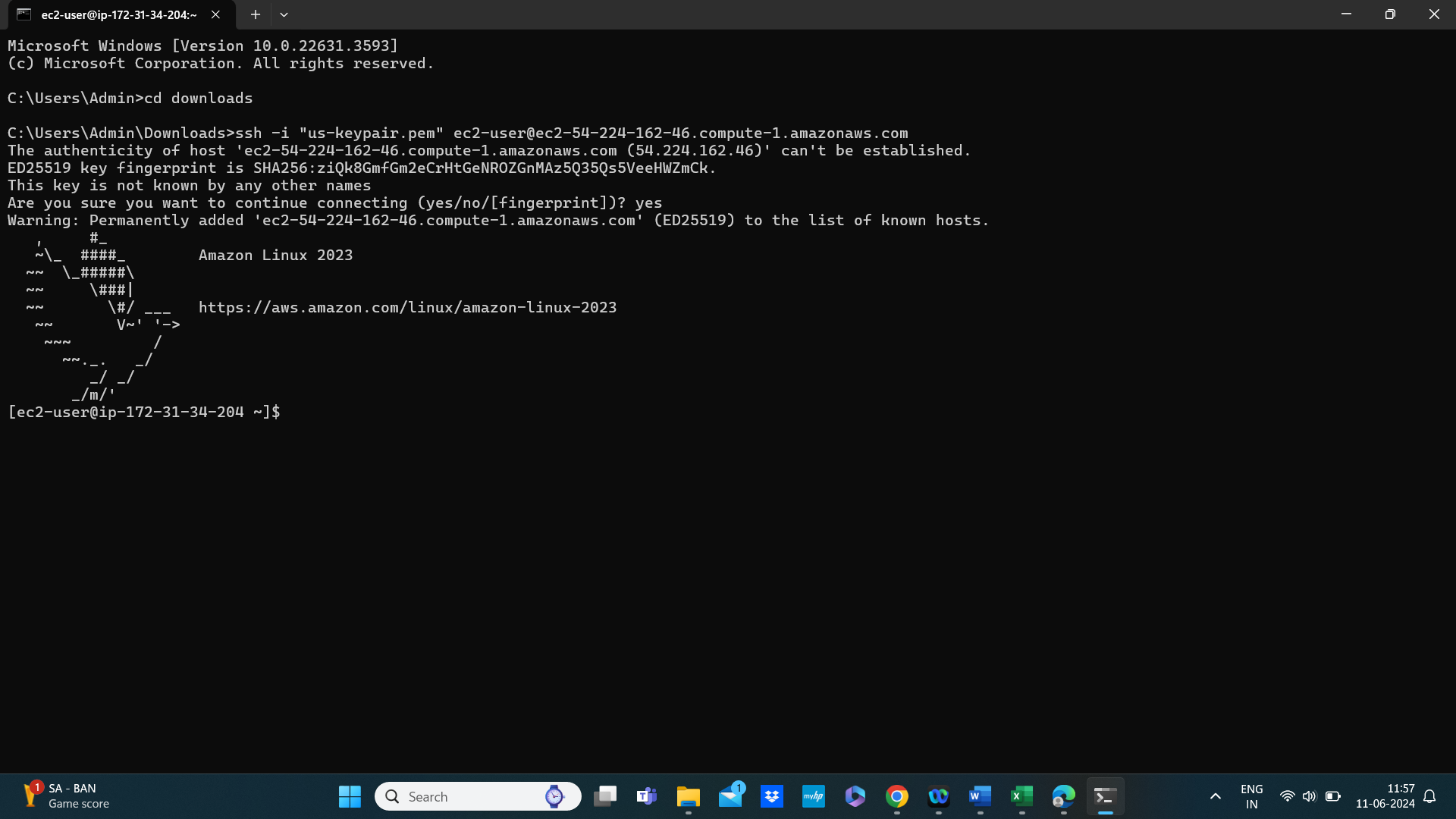
* **Security Groups:** Ensure your security group rules in AWS allow HTTP traffic on port 80 and SSH traffic on port 22.
* **Log Files:** Regularly check log files for any errors or issues.
* **Backup Directory Management:** Monitor the /backup directory to ensure it doesn't consume too much disk space. Implement a rotation policy if needed.

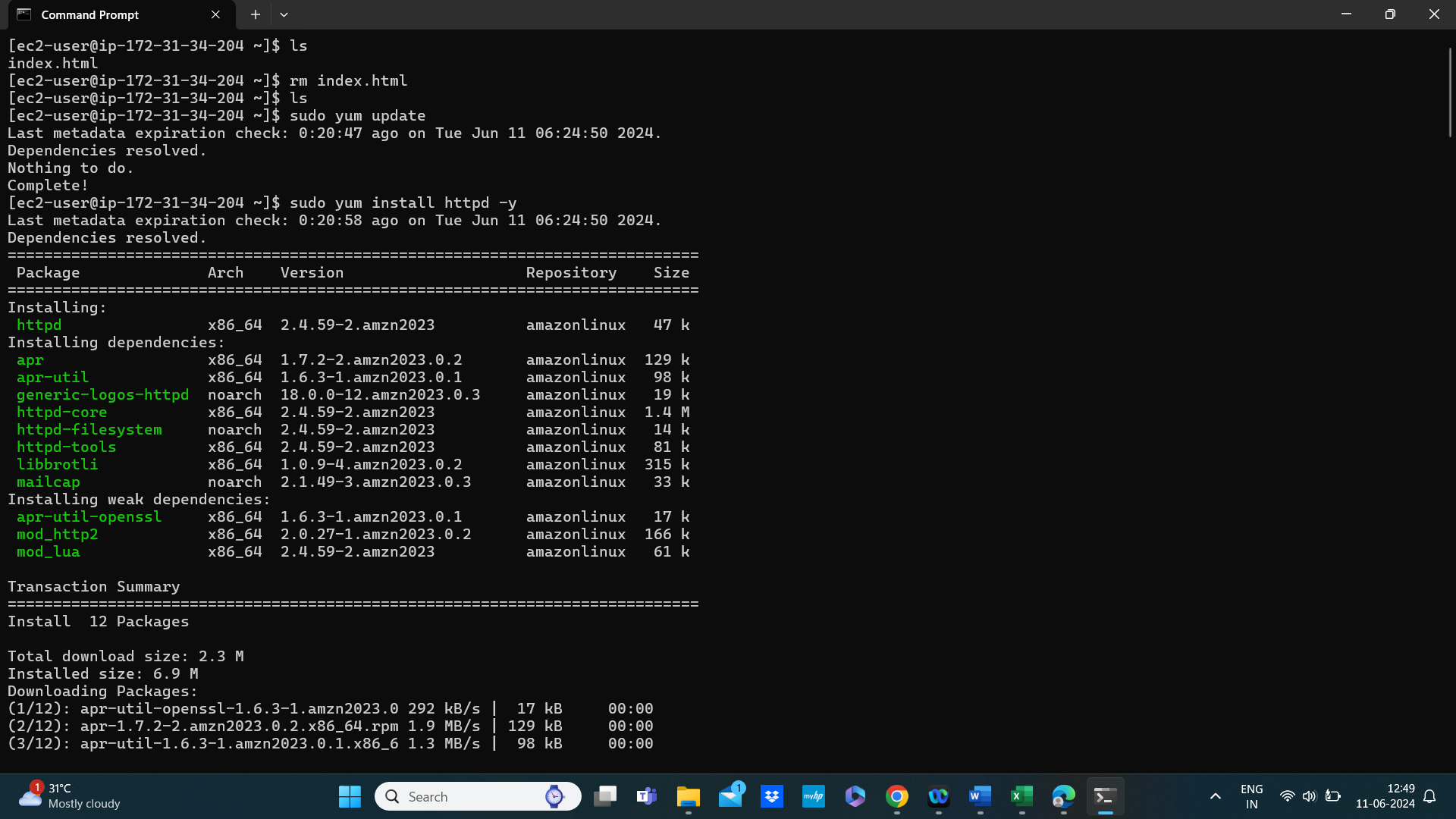
These steps will help you confirm that your setup is correct and functioning as intended.

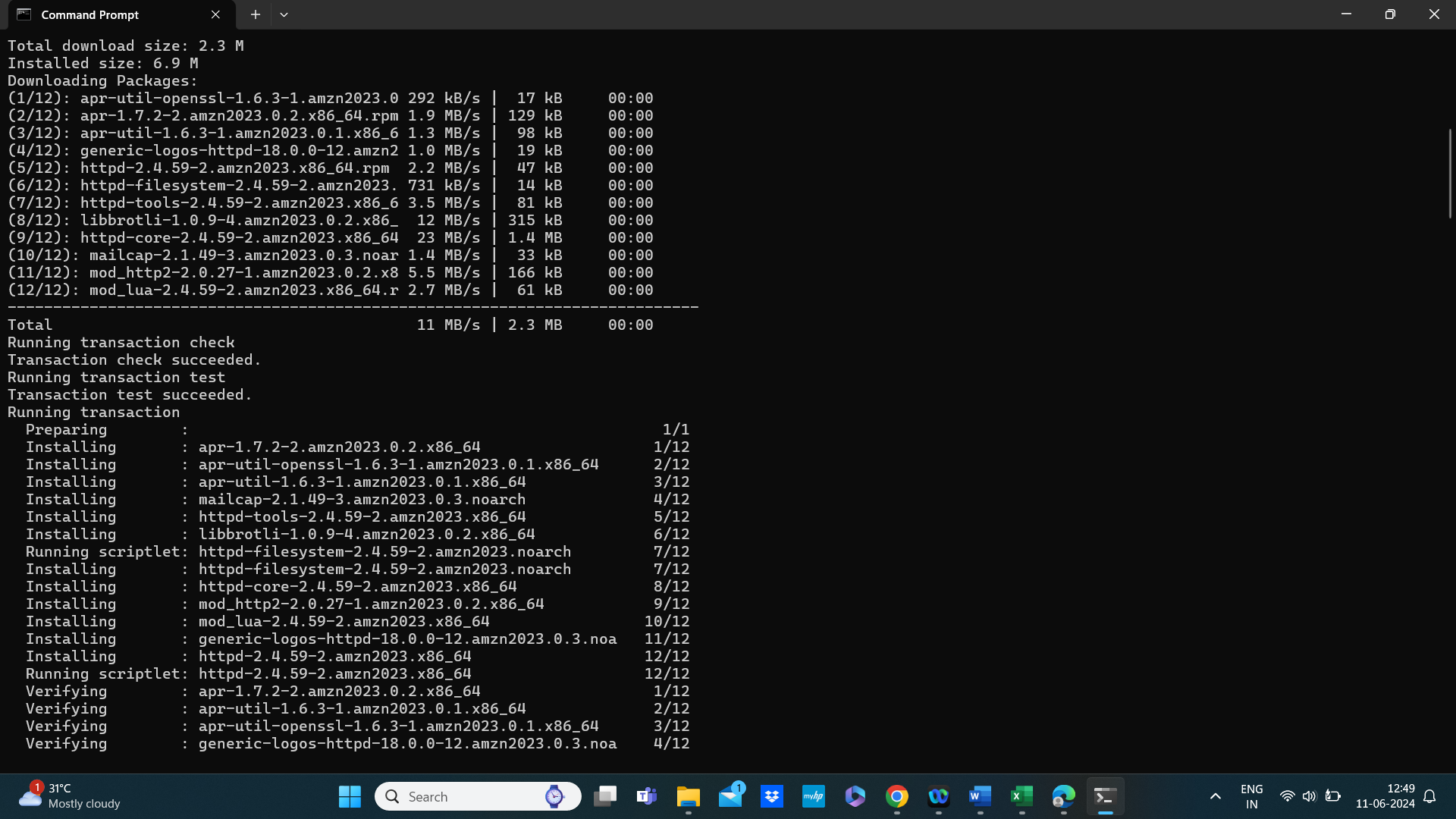
Top of Form

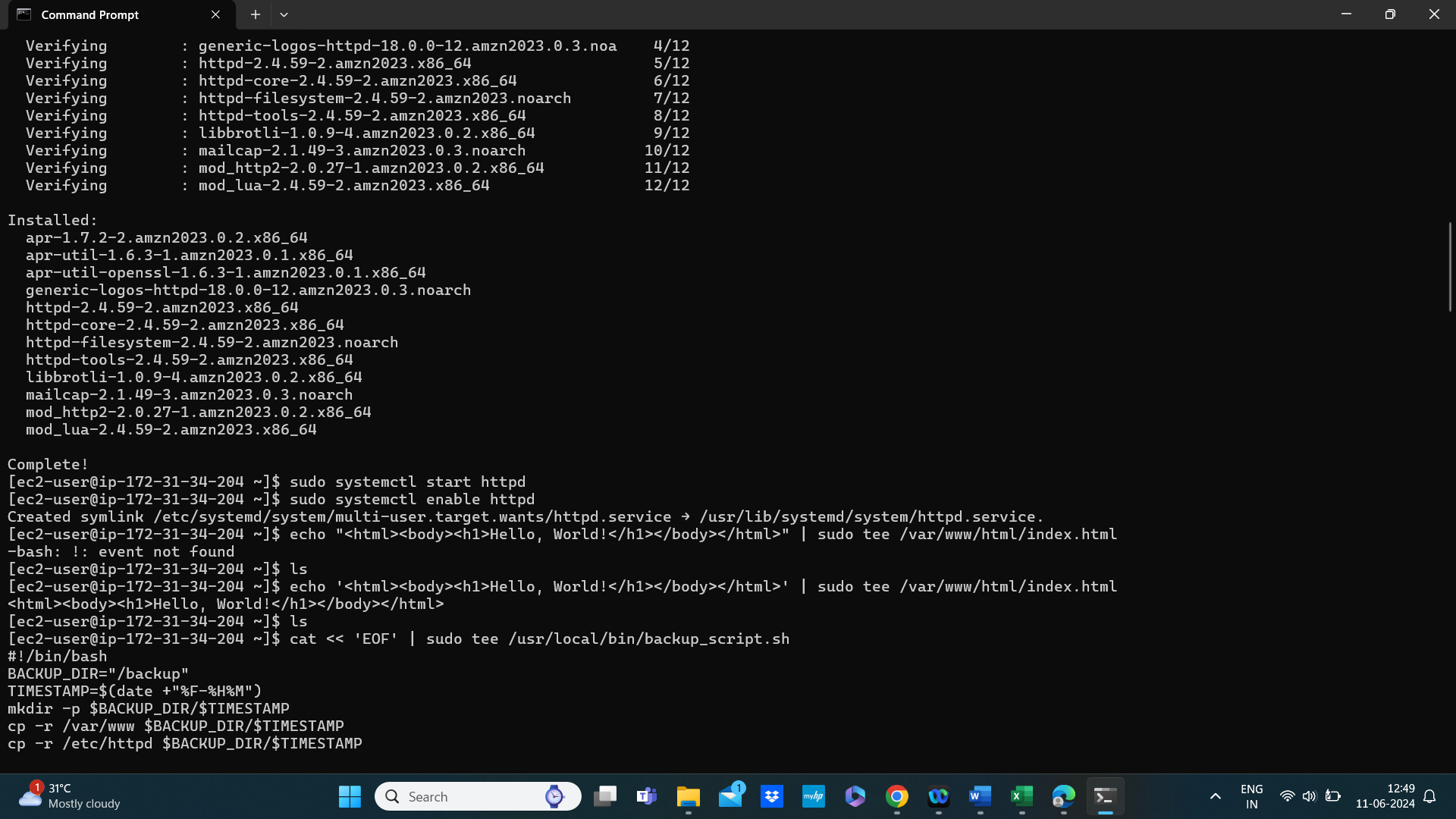
Bottom of Form

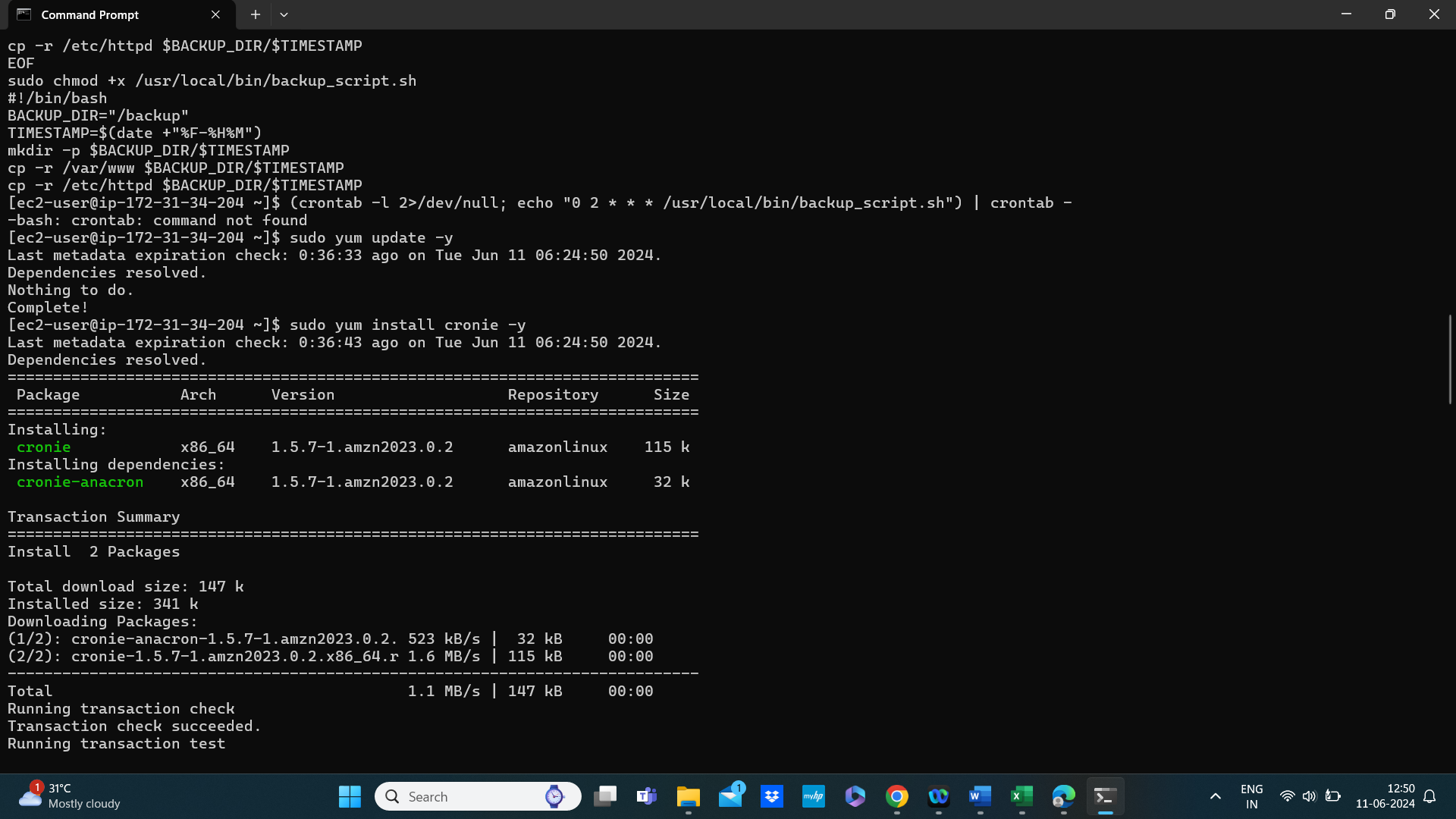


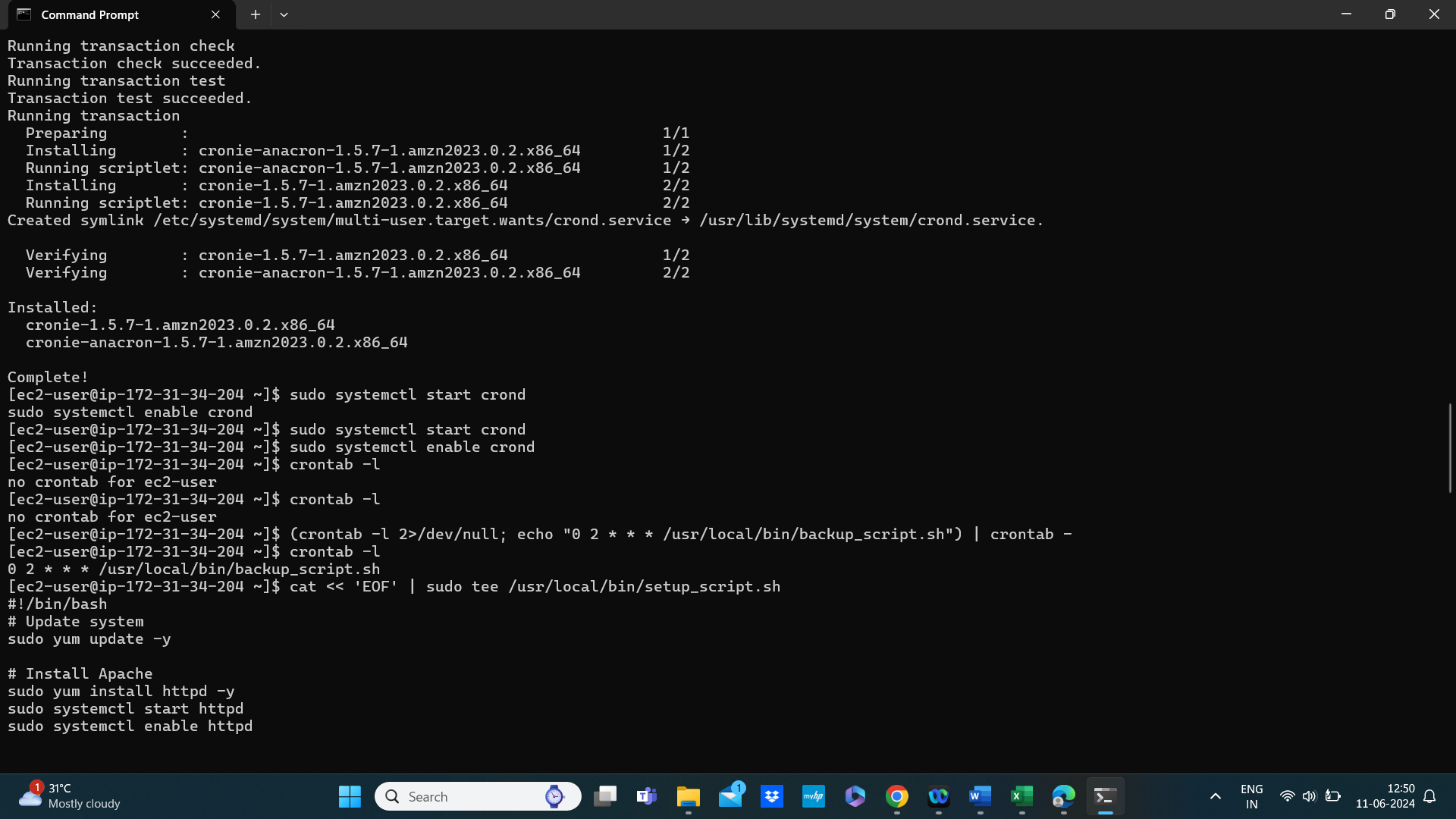


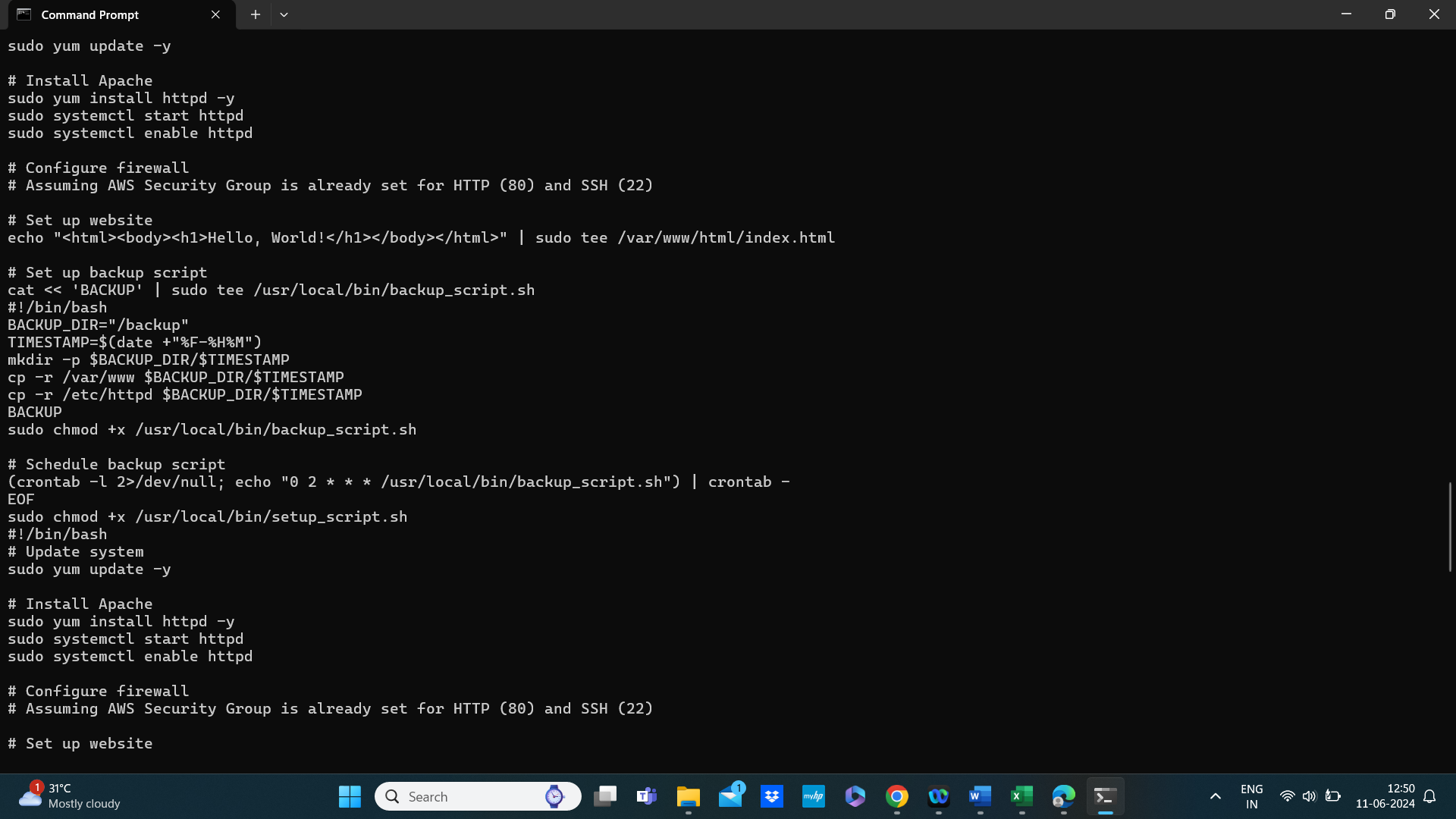


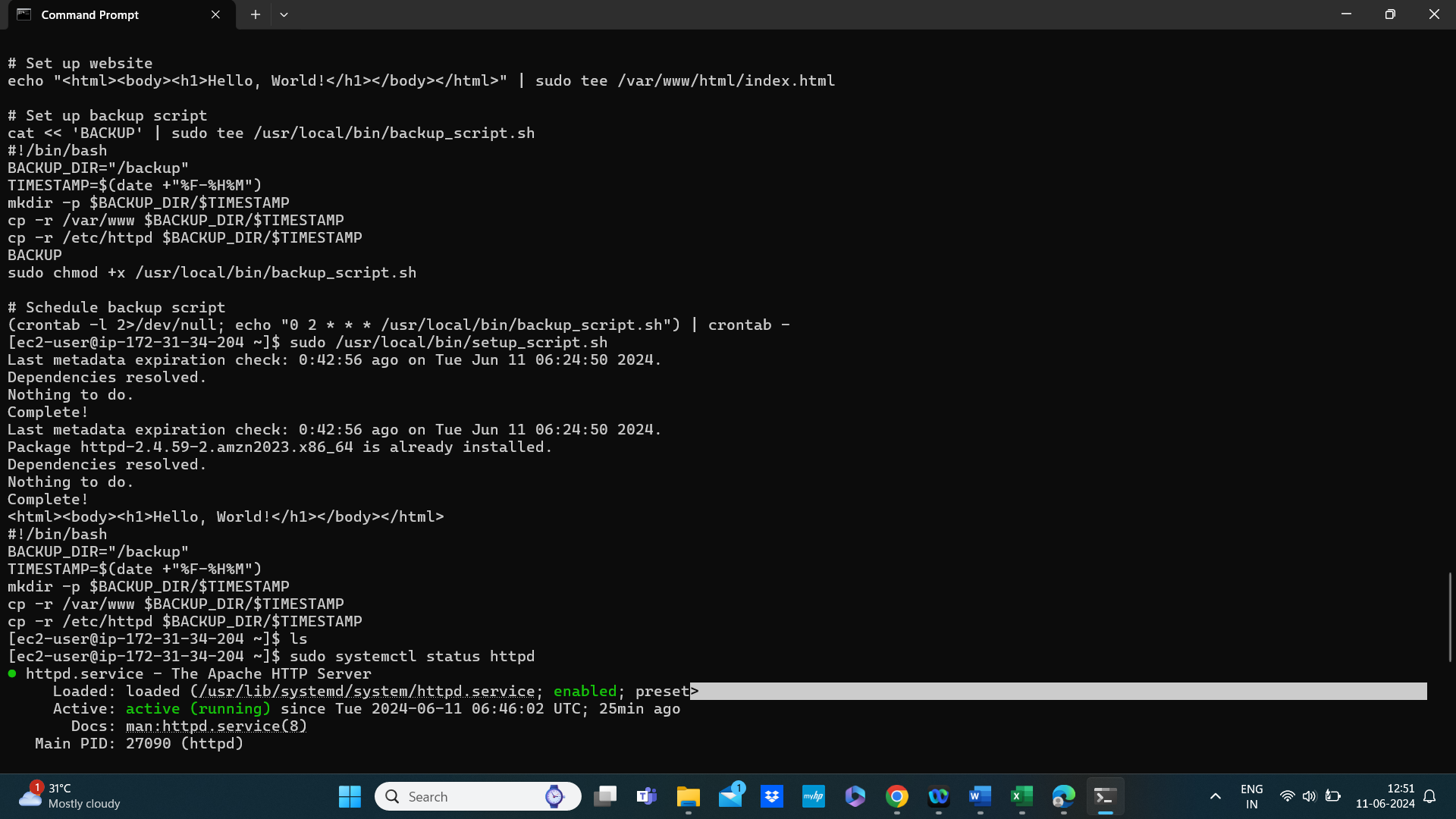


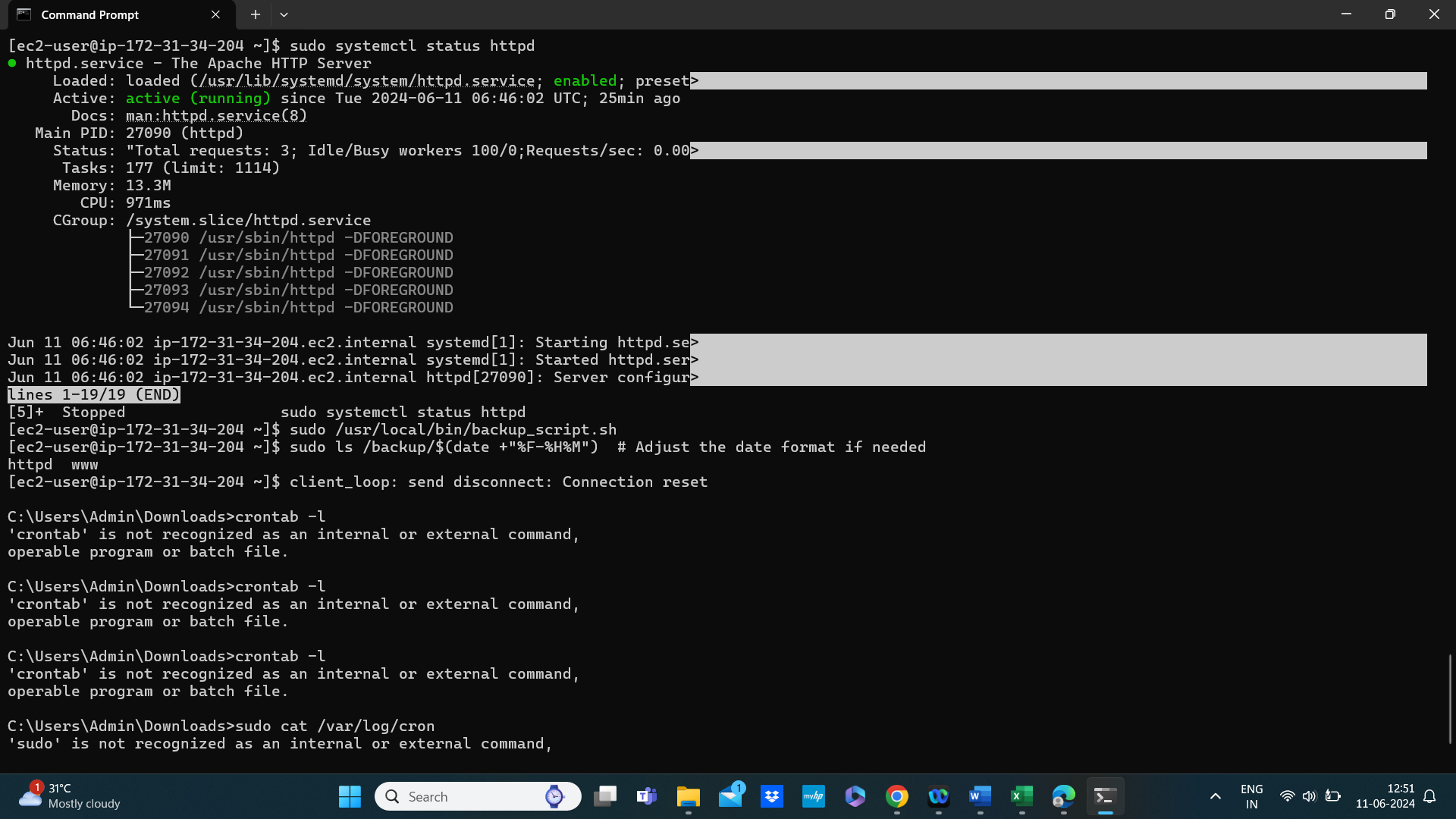


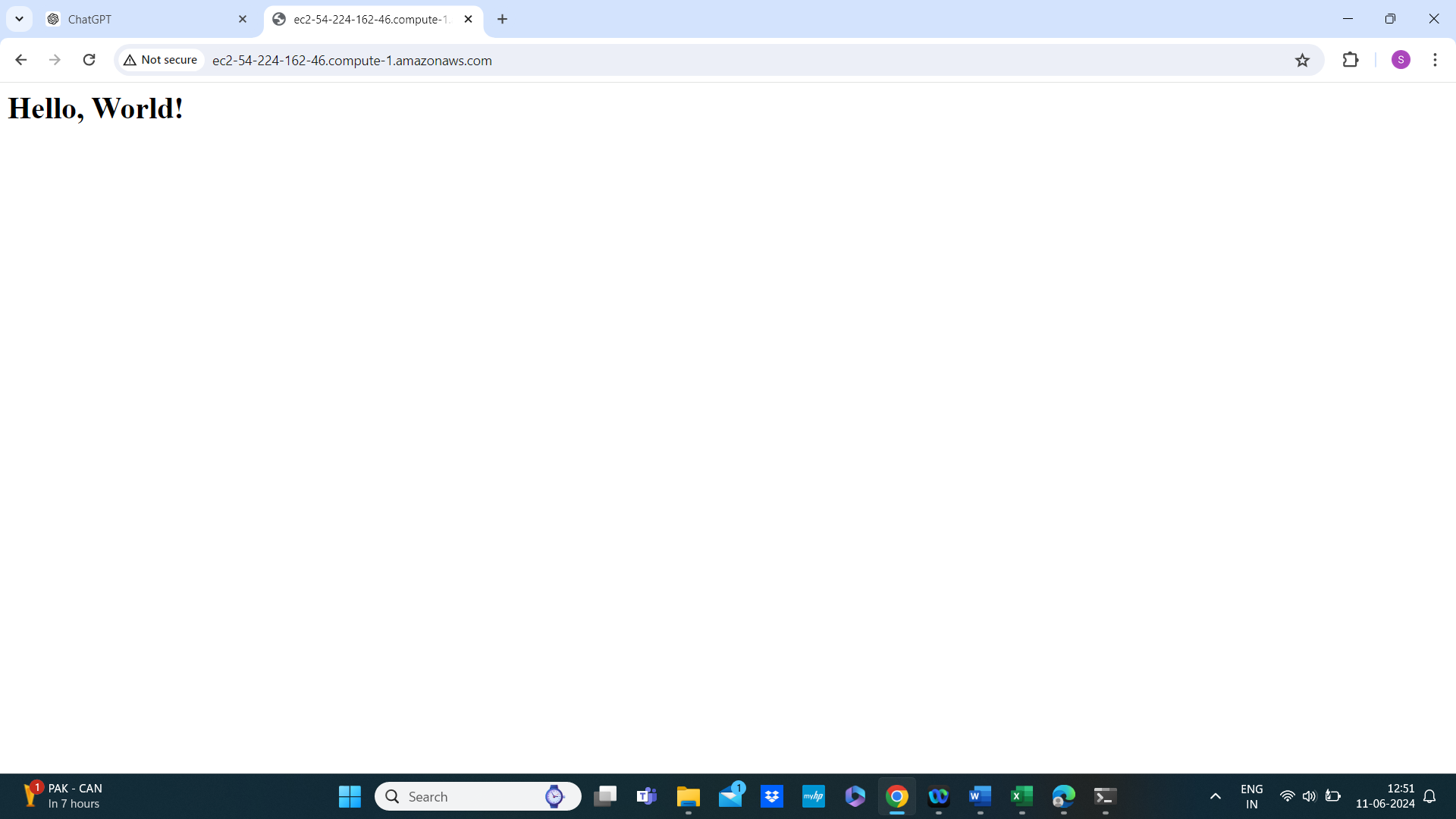












Terminate the instance to avoid charges

