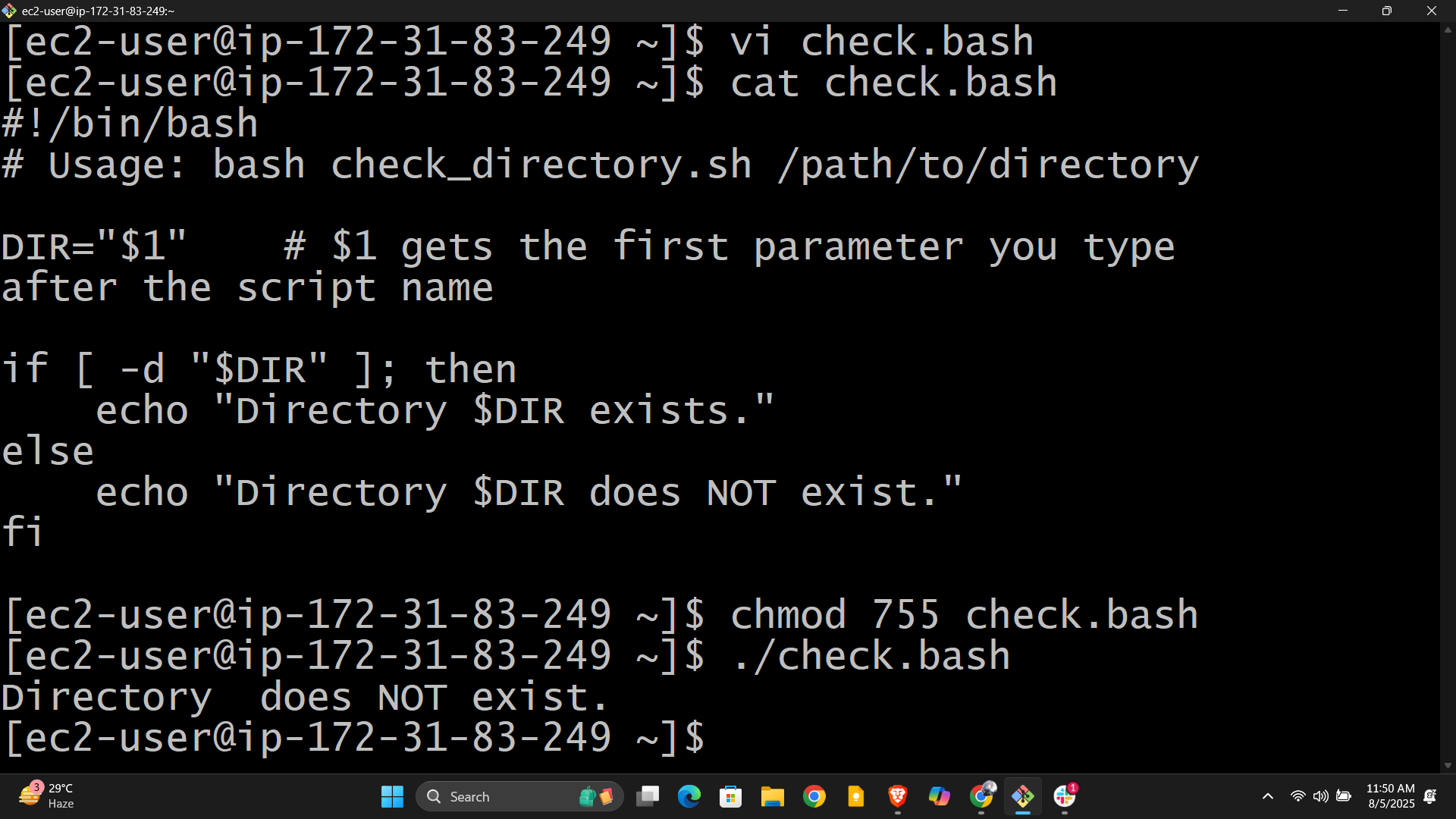
1) Create a Bash script to check if a directory is available or not.



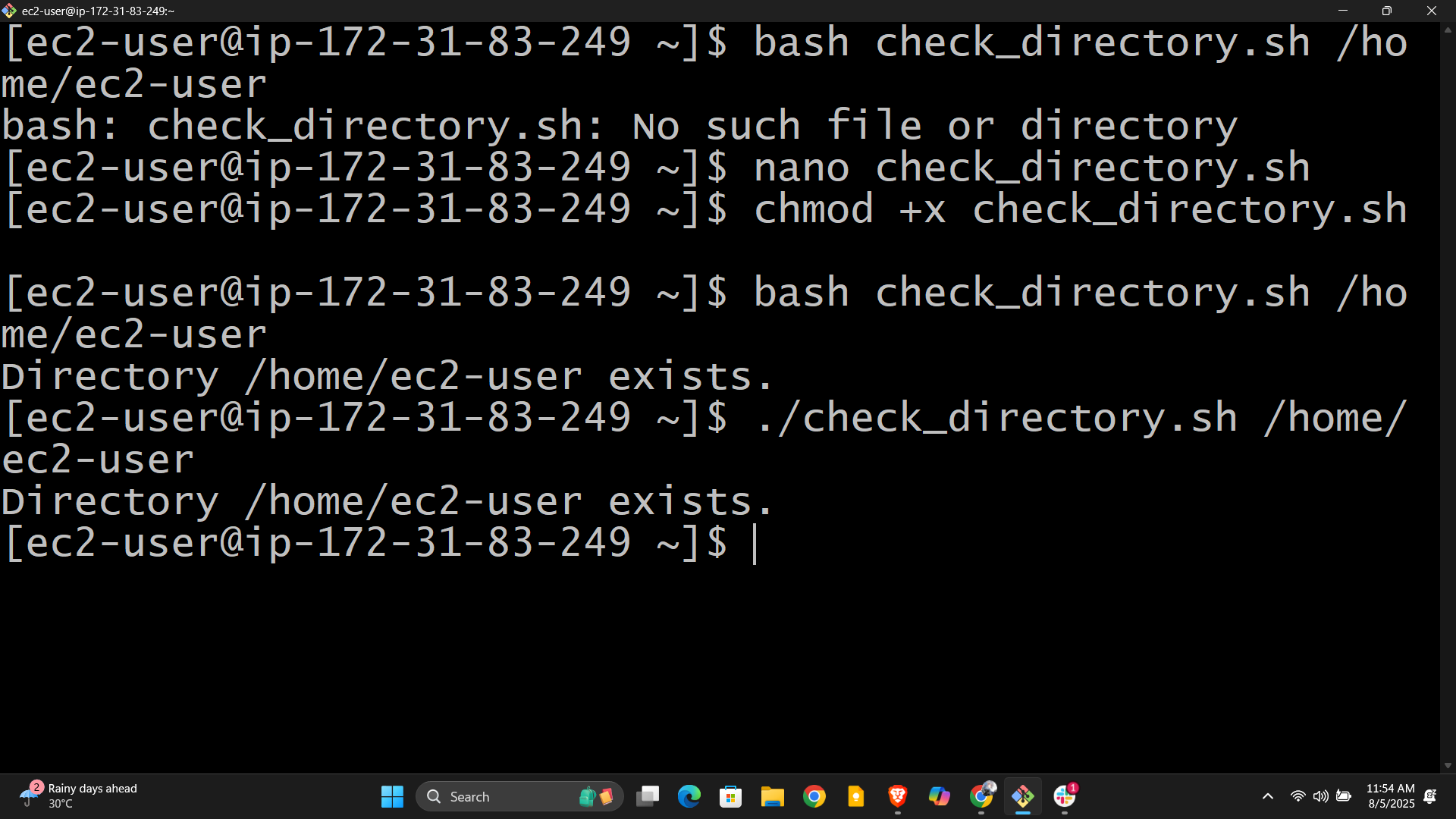
**1. Create the Script File**

**2. Copy & Paste the Script**

**3. Save and Exit nano**

**4. Make the Script Executable (optional but recommended)**

**5. Run the Script**

Or, if you made it executable:

**2) Create a bash script which will create multiple files.**

**1. Create the File**

**2. Copy and Paste this Script into nano**

#!/bin/bash

# This script creates files you list after the script name.

# Usage: bash create\_files.sh file1.txt file2.txt file3.txt

for file in "$@"

do

touch "$file"

echo "Created $file"

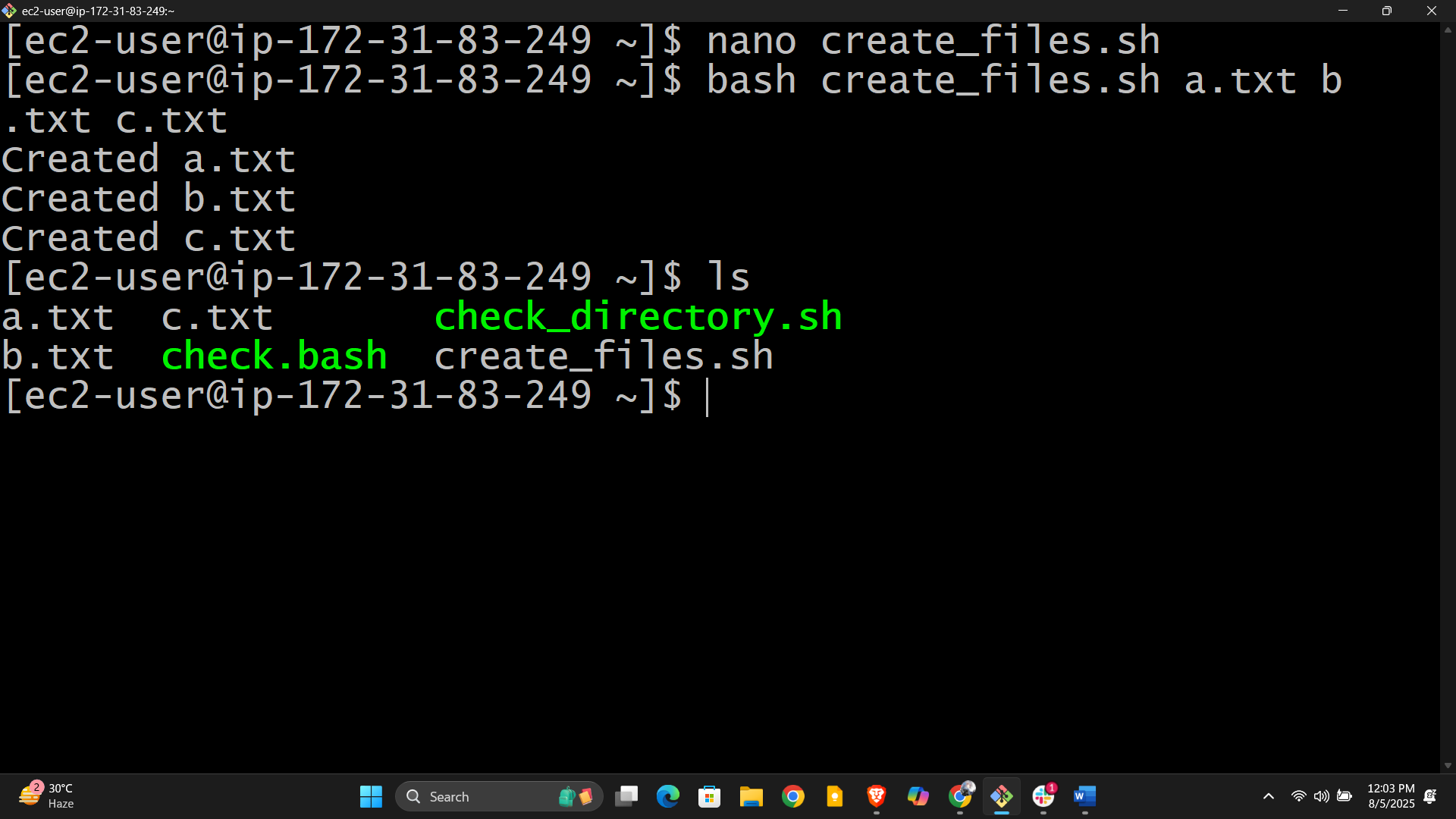
done

**3. Save and Exit nano**

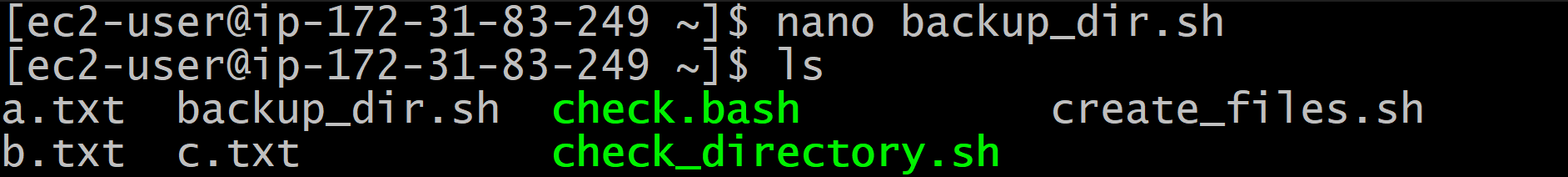
**4. Make the Script Executable (recommended)**

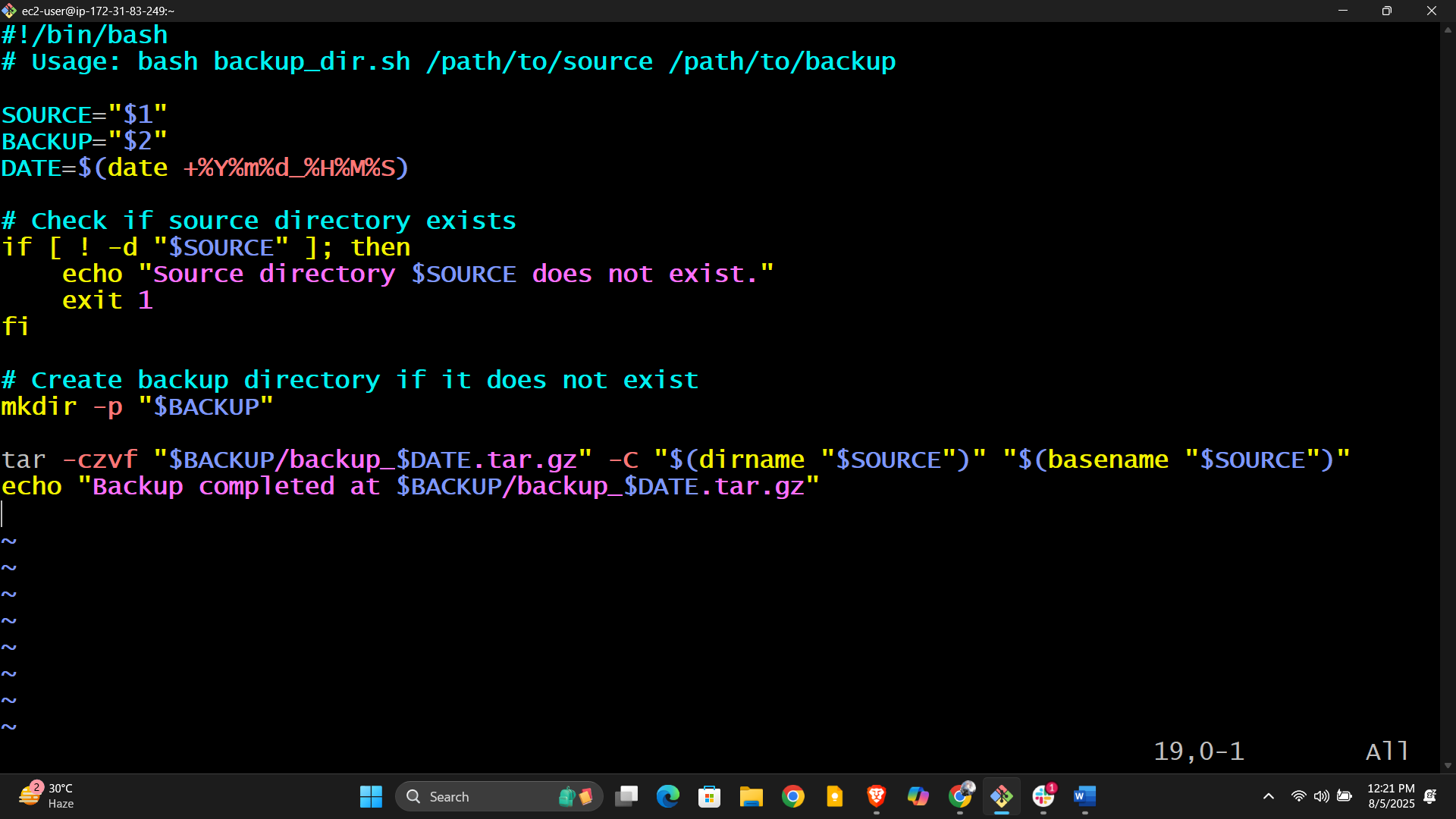
**5. Run Your Script**

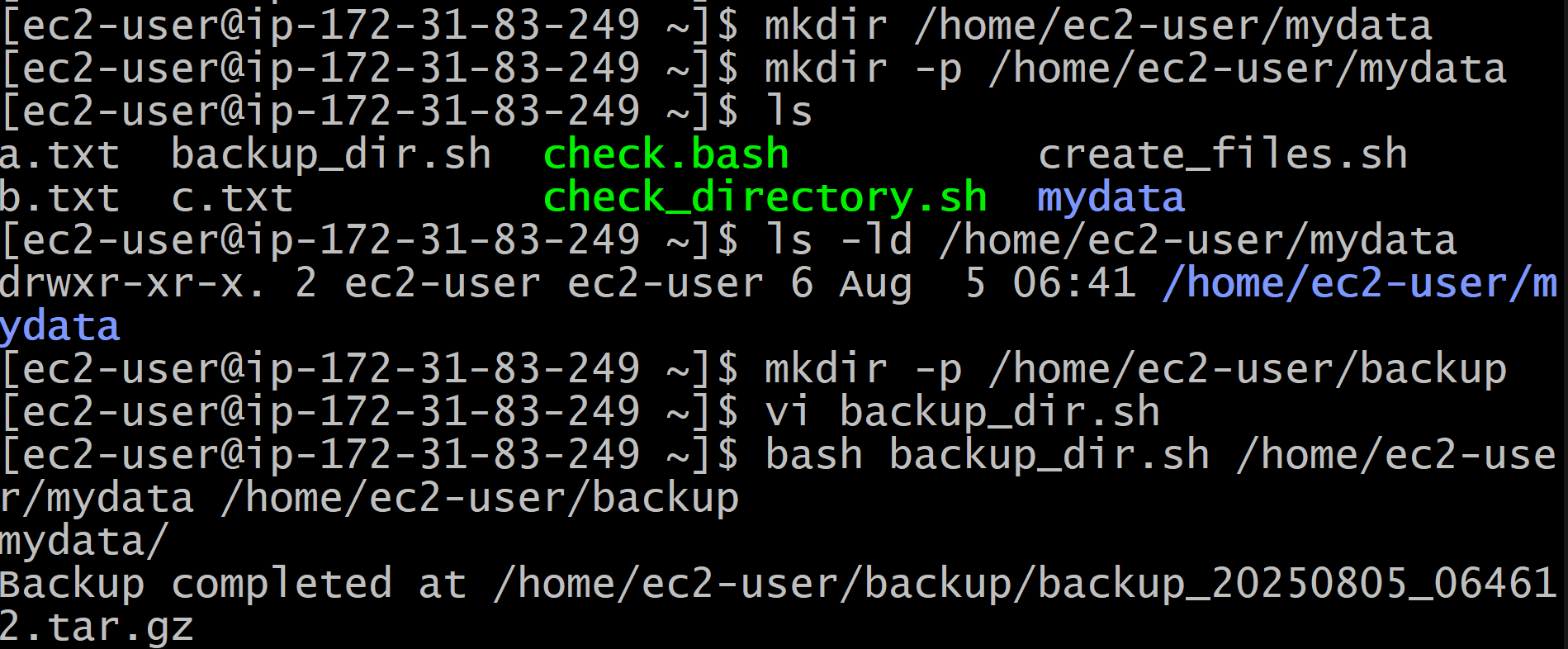
**or, if you made it executable:**

****

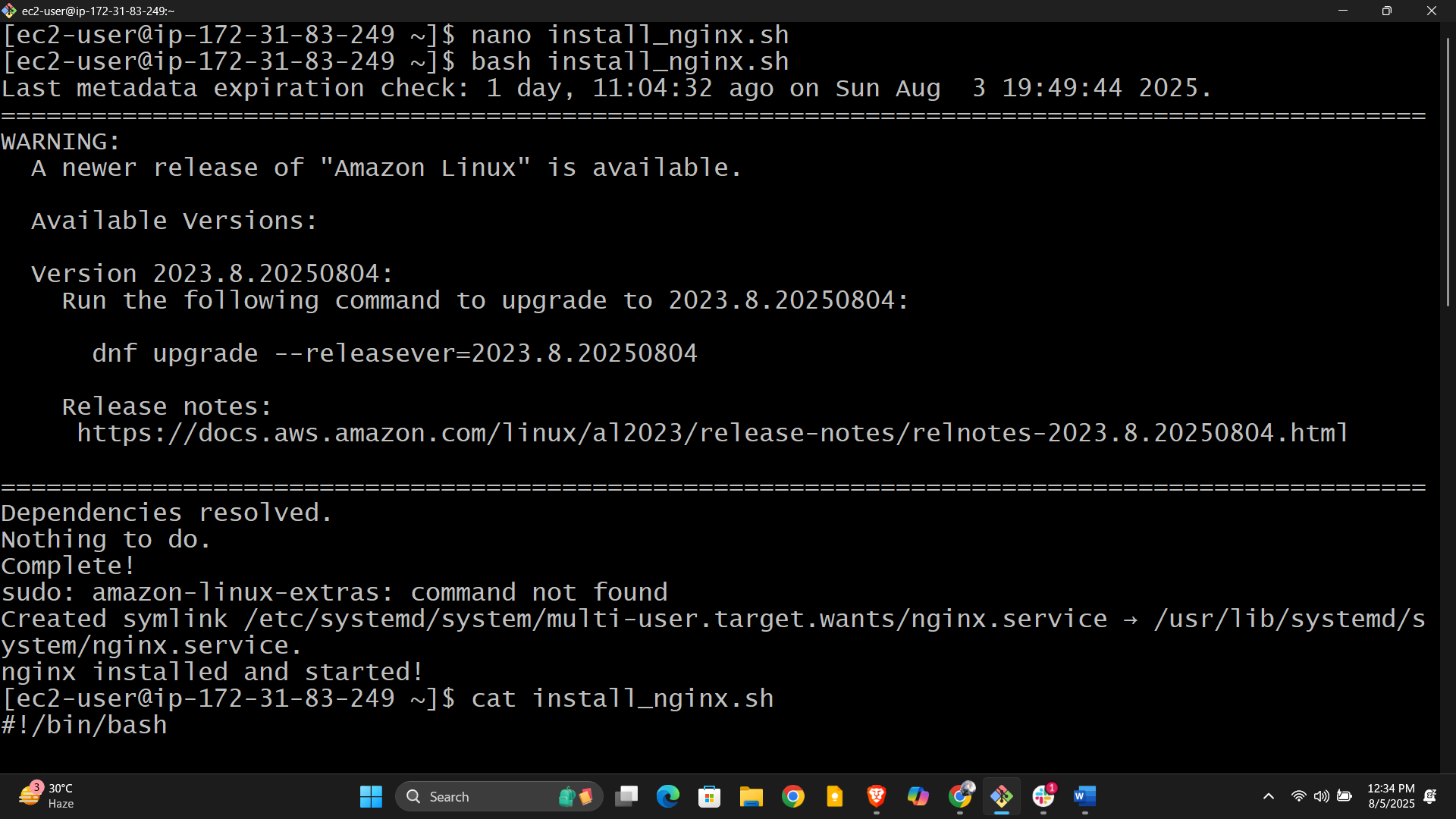
**3) Create a bash script to take backup of a directory.**

****

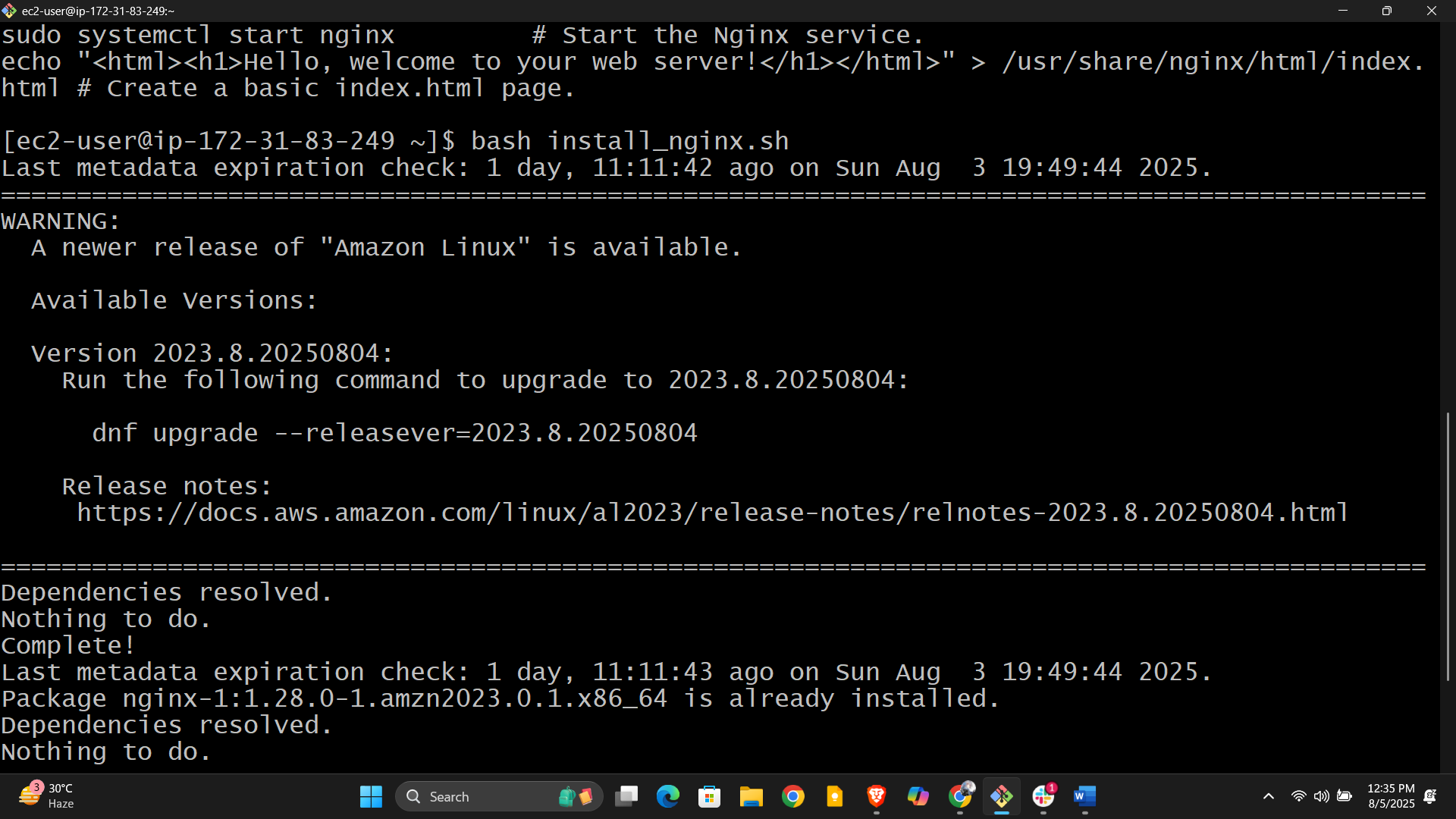
****

****

**4) Create a bash script to install nginx in ec2 server.**

****

****

****

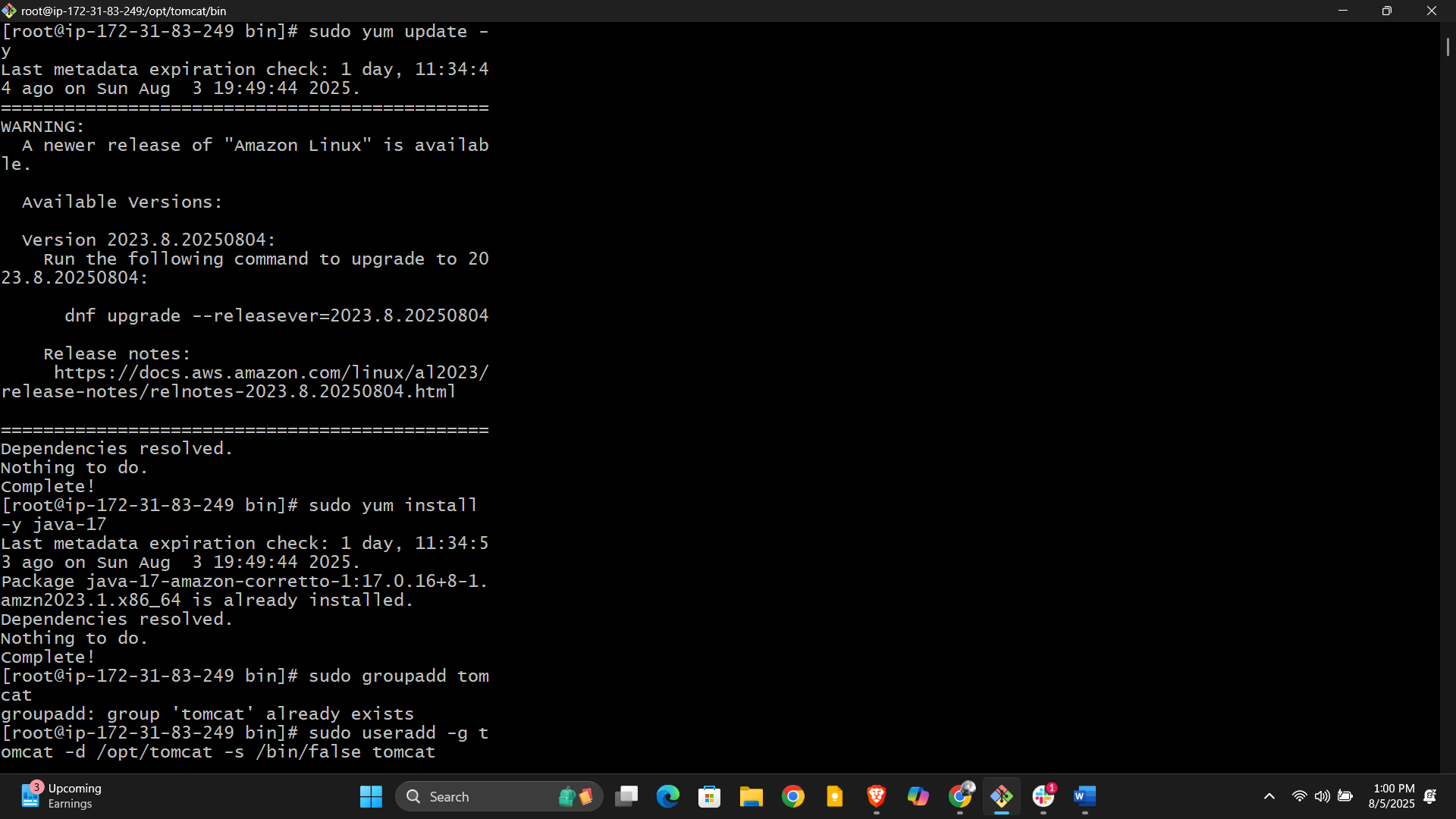
**5) Create a bash script to install ApacheTomcat in ec2 server.**

**1. Update the system  
2. Install Java**

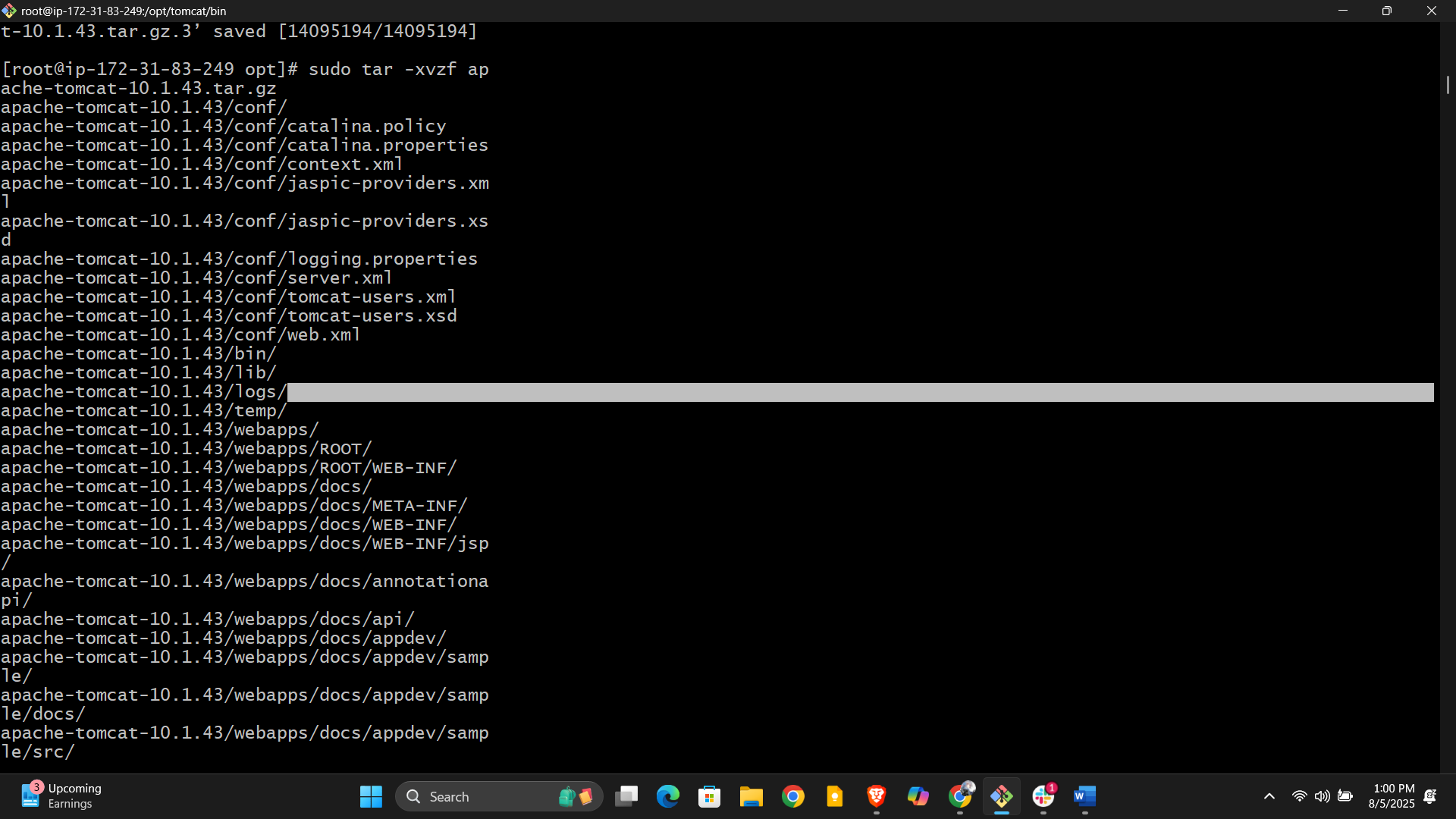
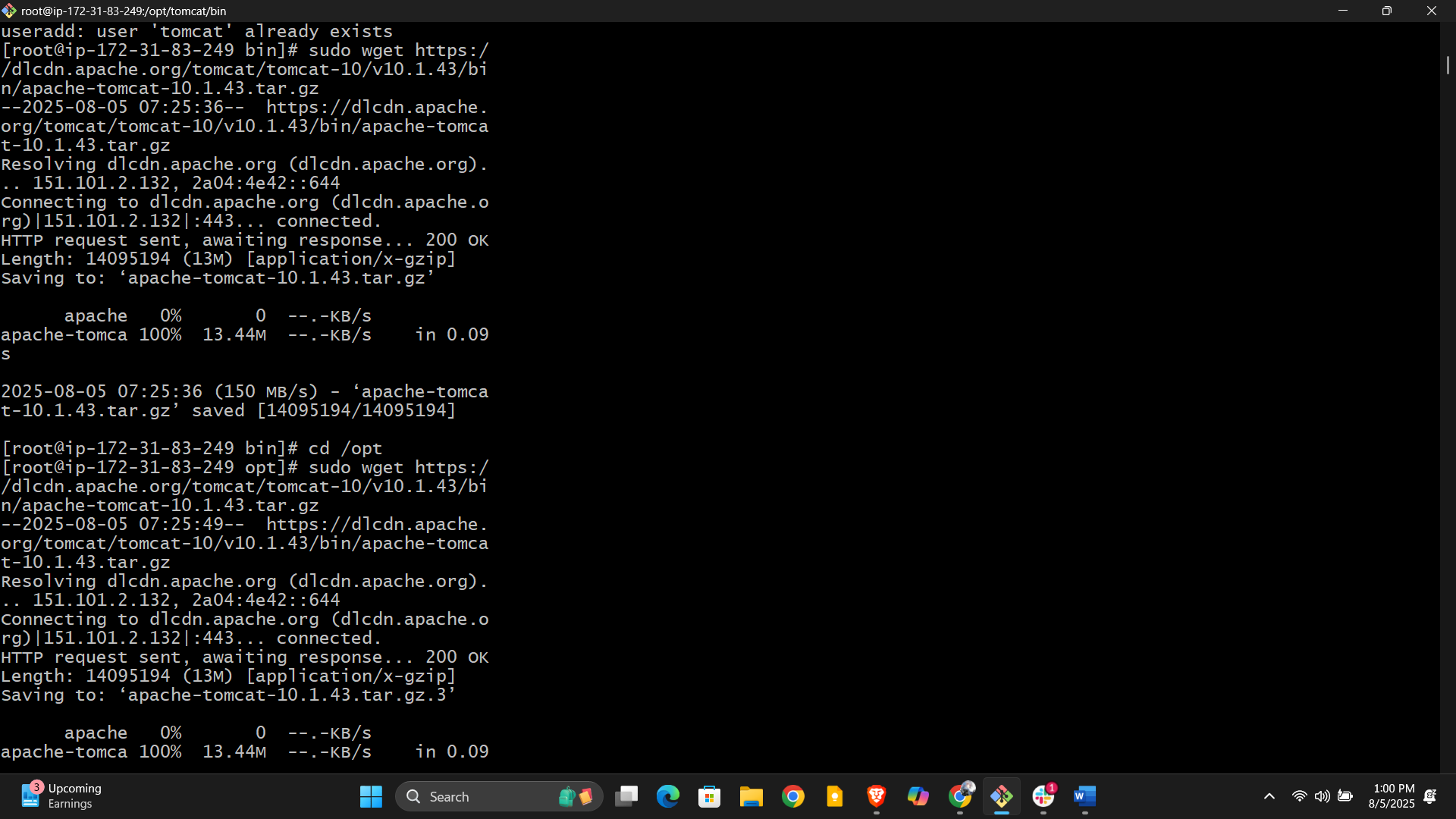
**3. Create a Tomcat user (Optional but recommended)**

**4. Download Apache Tomcat**

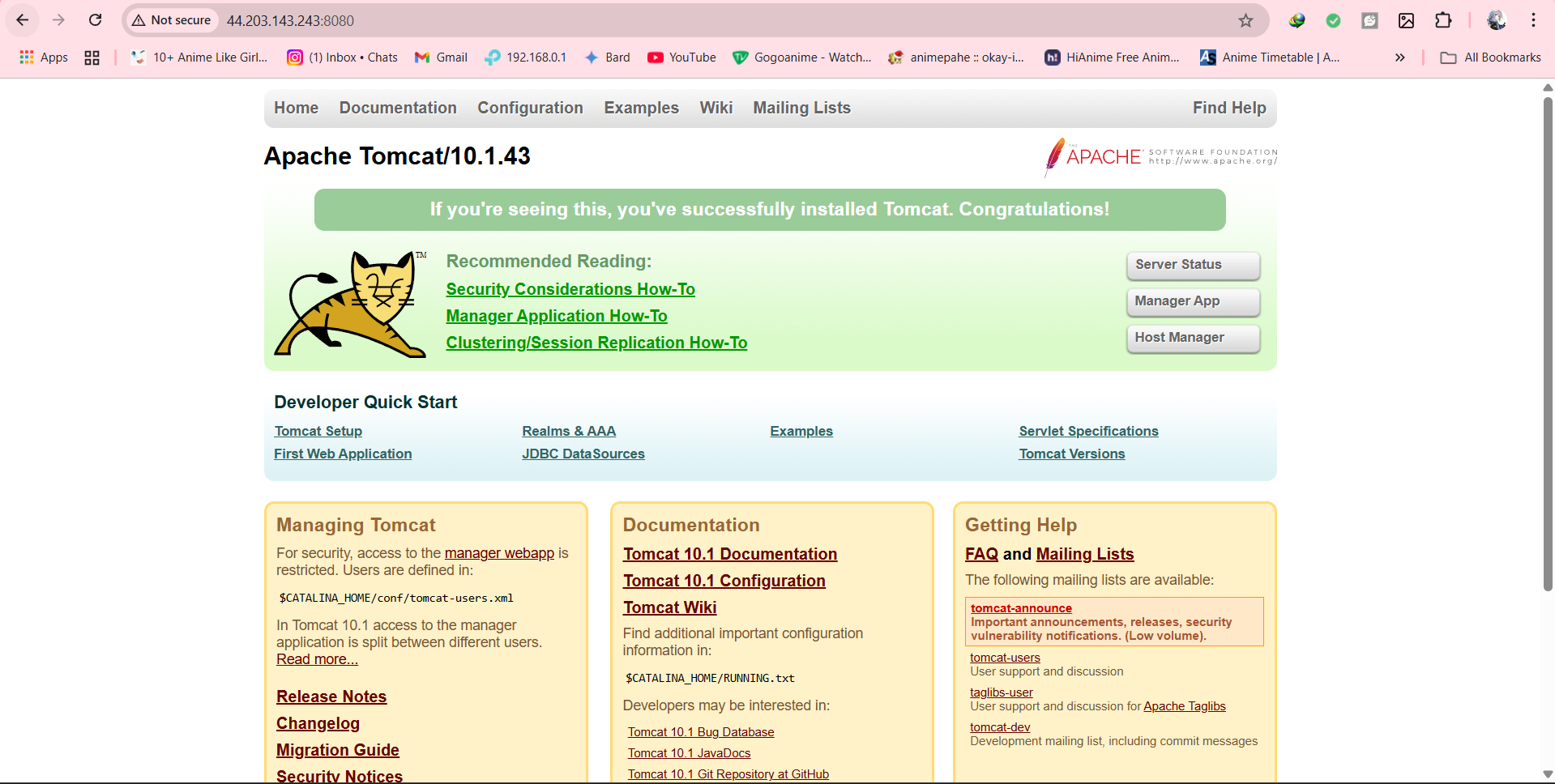
**5. Set Permissions**

****

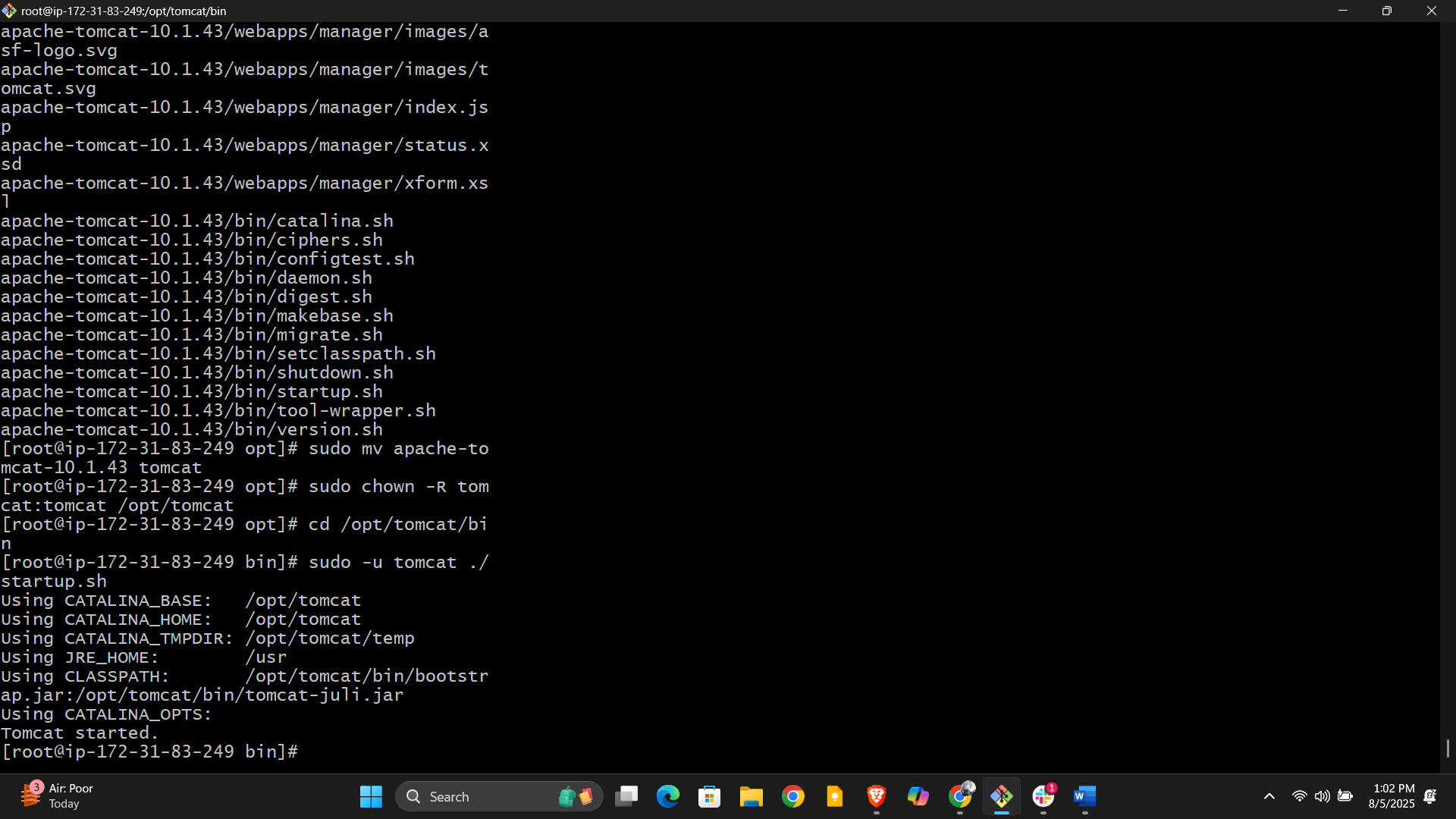
**6. Start Tomcat**

****

**7. Enable Access to Port 8080**

****

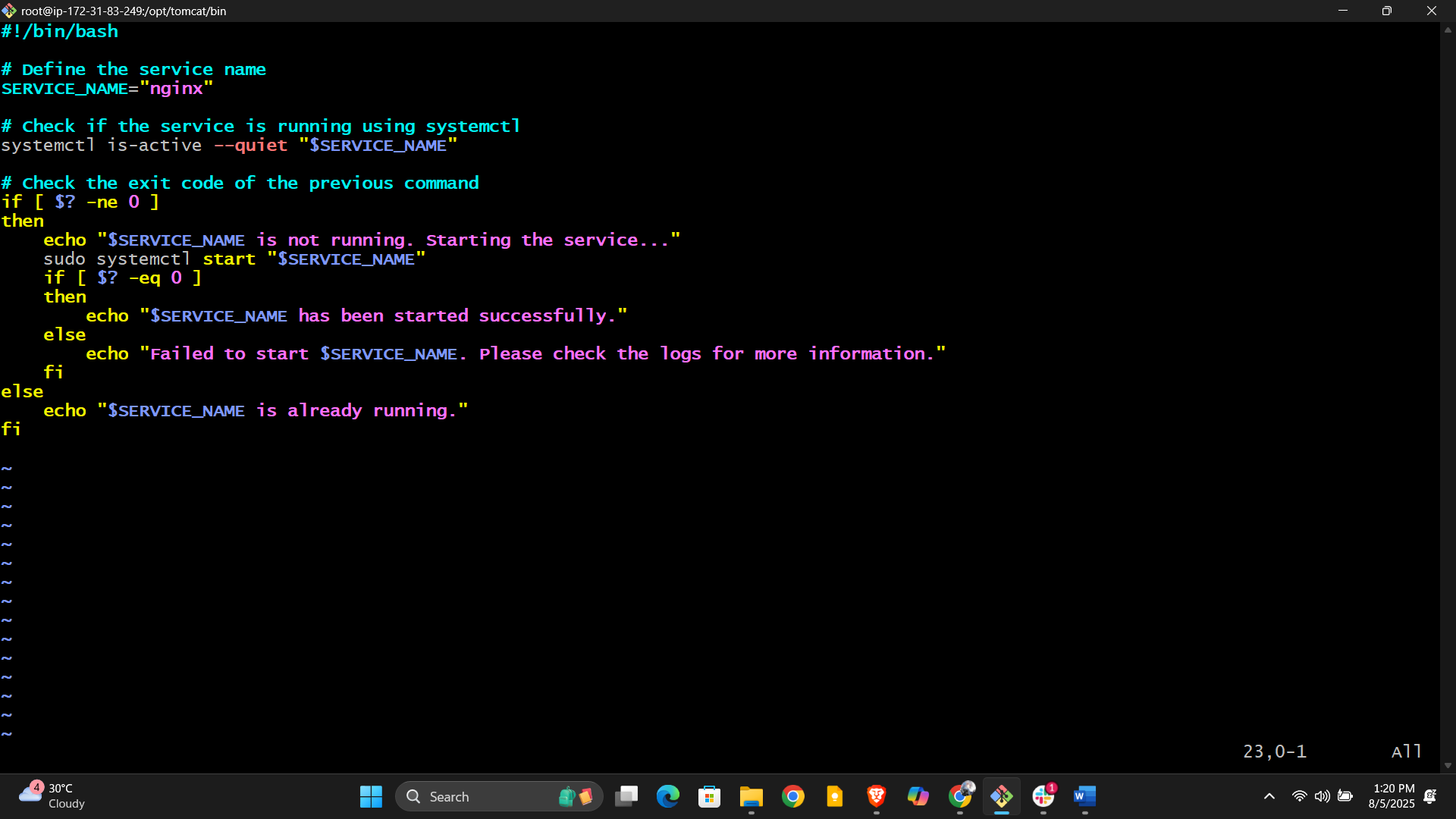
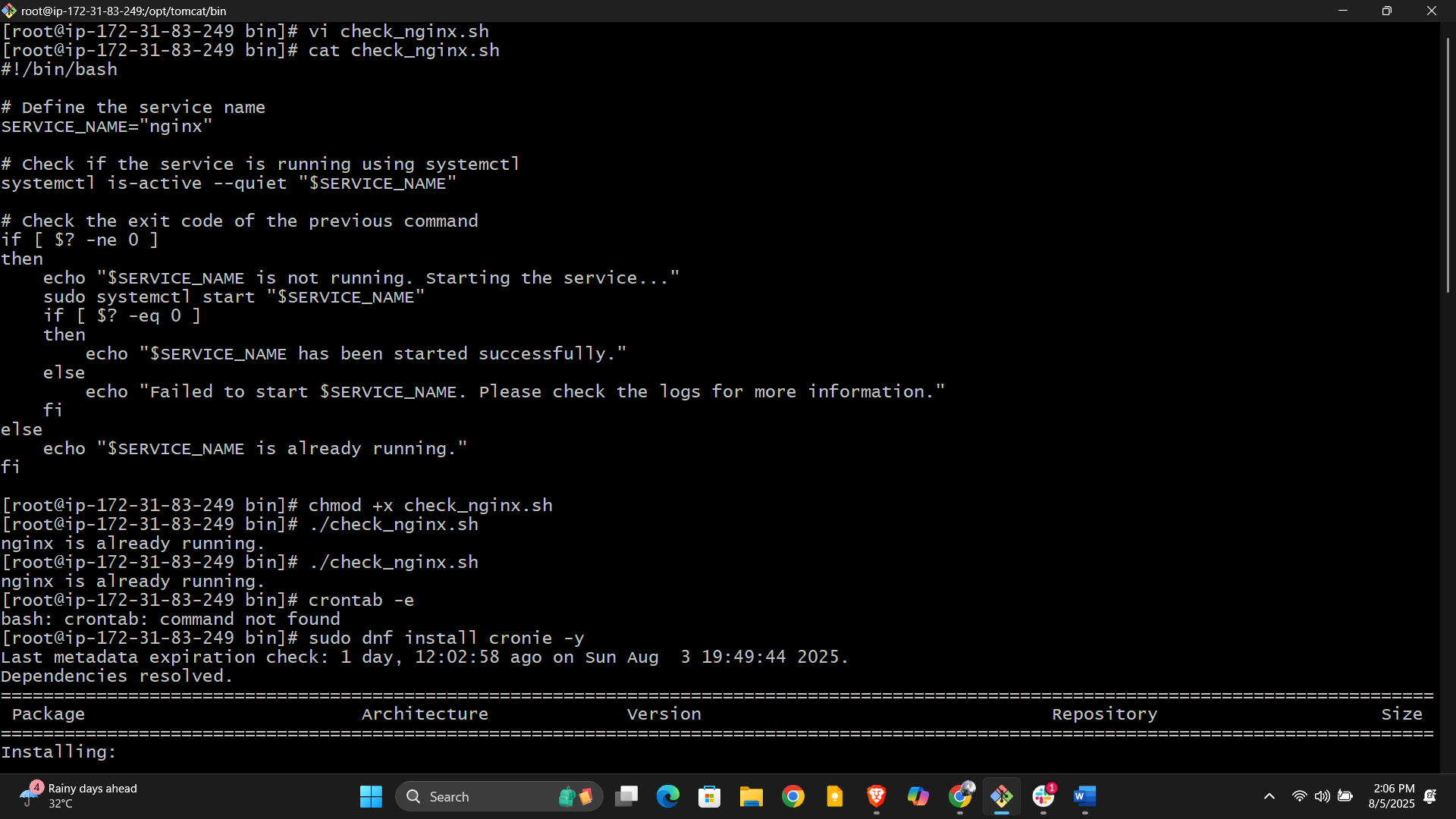
**8. Access Tomcat**

****

**6) Create a bash script to check list if nginx service is running or not,if not running then   
script should start the service.**

**1. Save the Script**

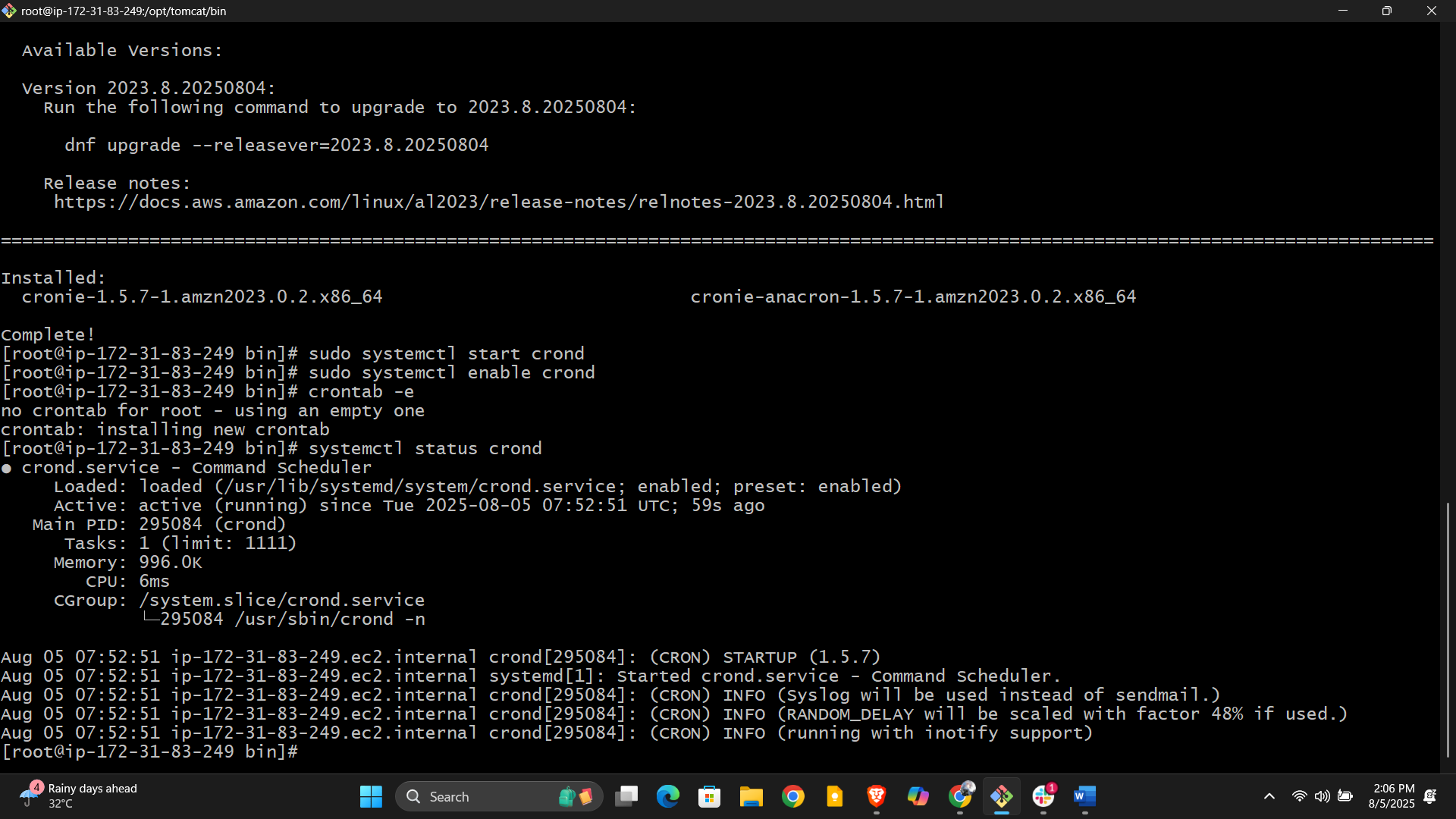
**2. Make the Script Executable**

**3. Run the Script Manually  
  
**

**4. Automate the Check with a Cron Job (Optional but Recommended)**

* **Add a line to run the script every 5 minutes (adjust as needed):**

**5. Verify Cron Job is Running**

****

**7) Create a bash script for calculator.**

1. **Save the script into a file, for example calculator.sh.**
2. **Make it executable:**

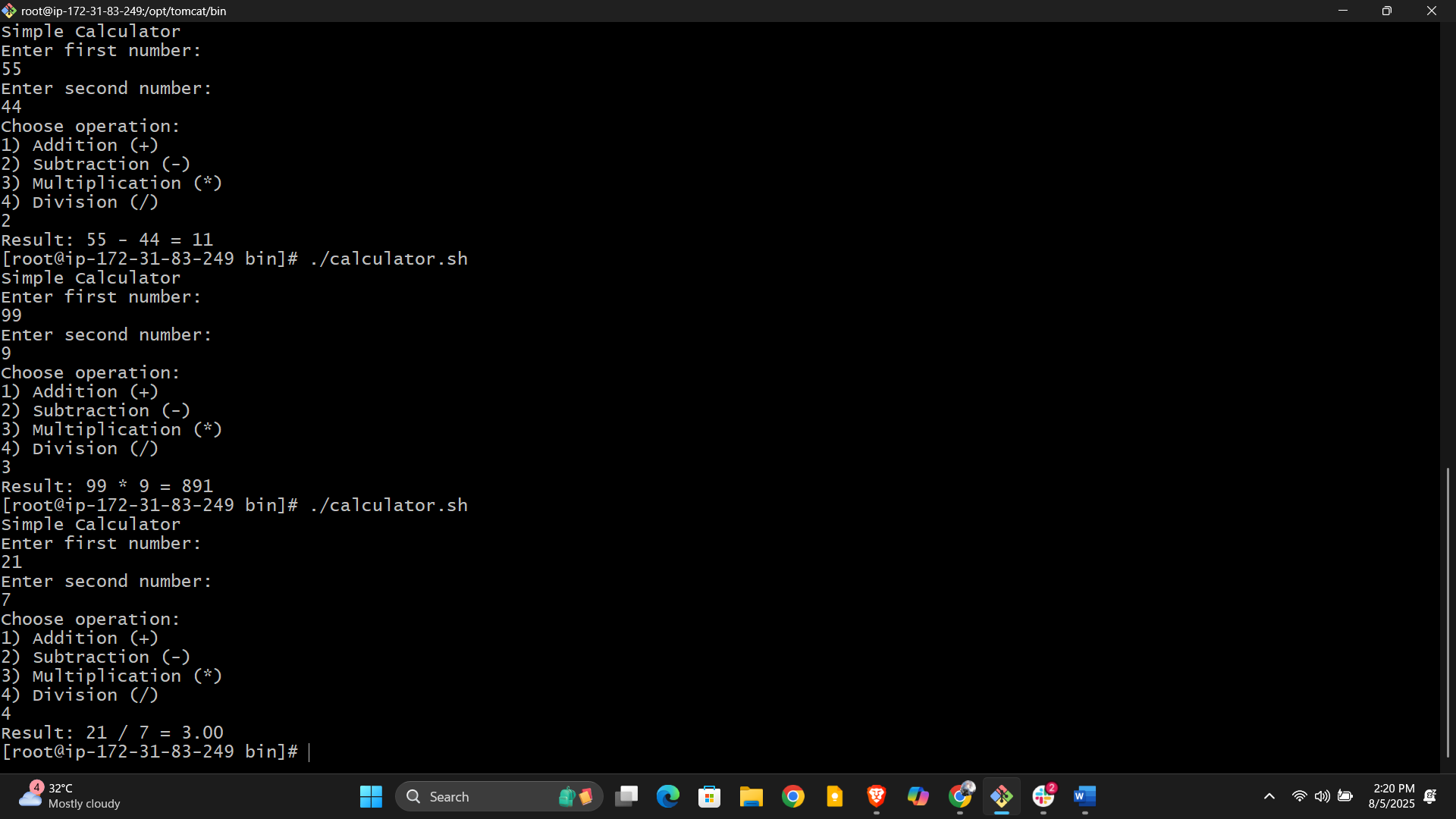
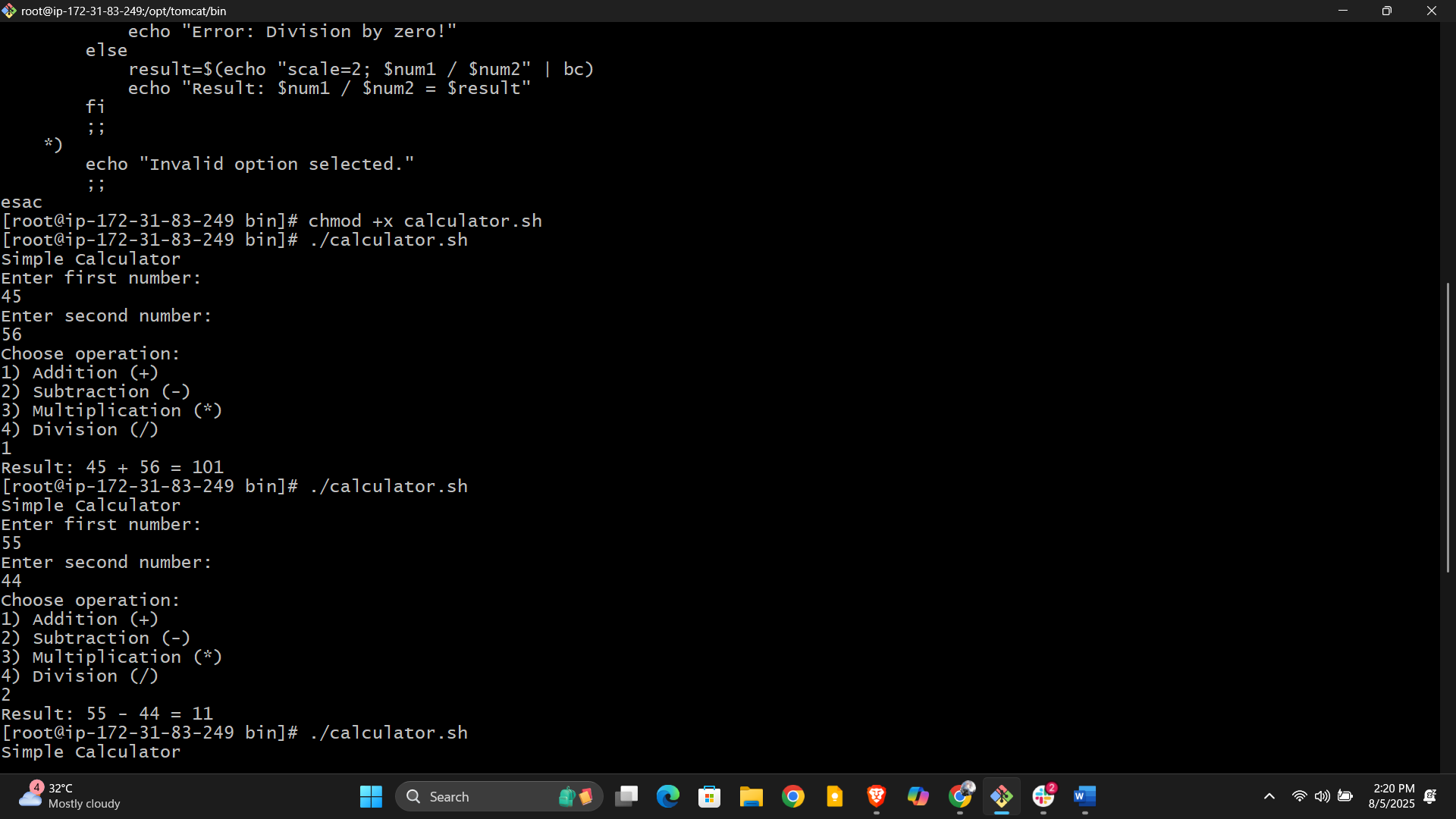
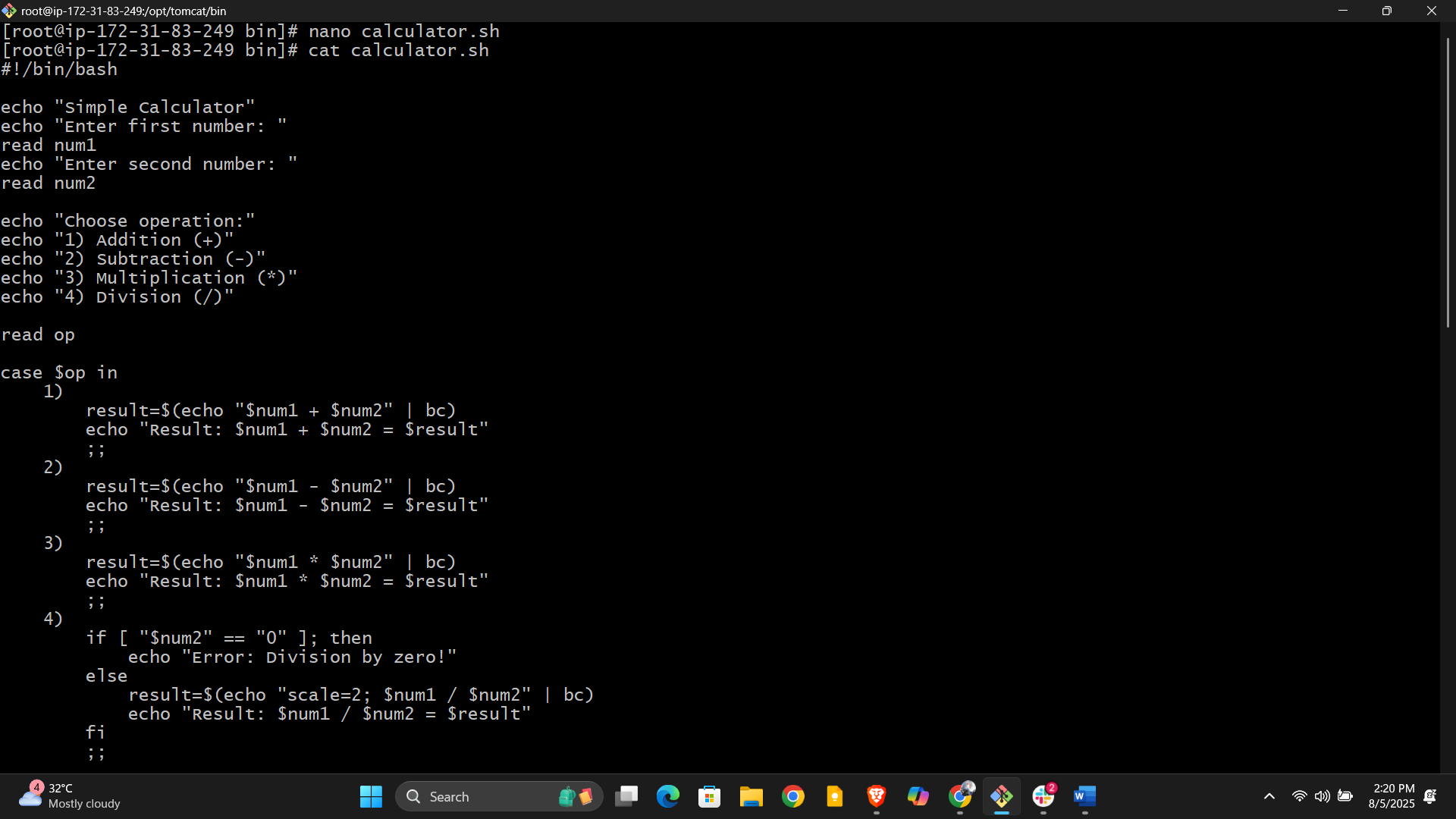
**bash**

**chmod +x calculator.sh**

1. **Run it:**

**bash**

**./calculator.sh**



**8) Create a bash script to check if directory is avaialble or not,if not then create a   
directory.**

How to use:

1. Save this script as check\_create\_dir.sh.
2. Make it executable:

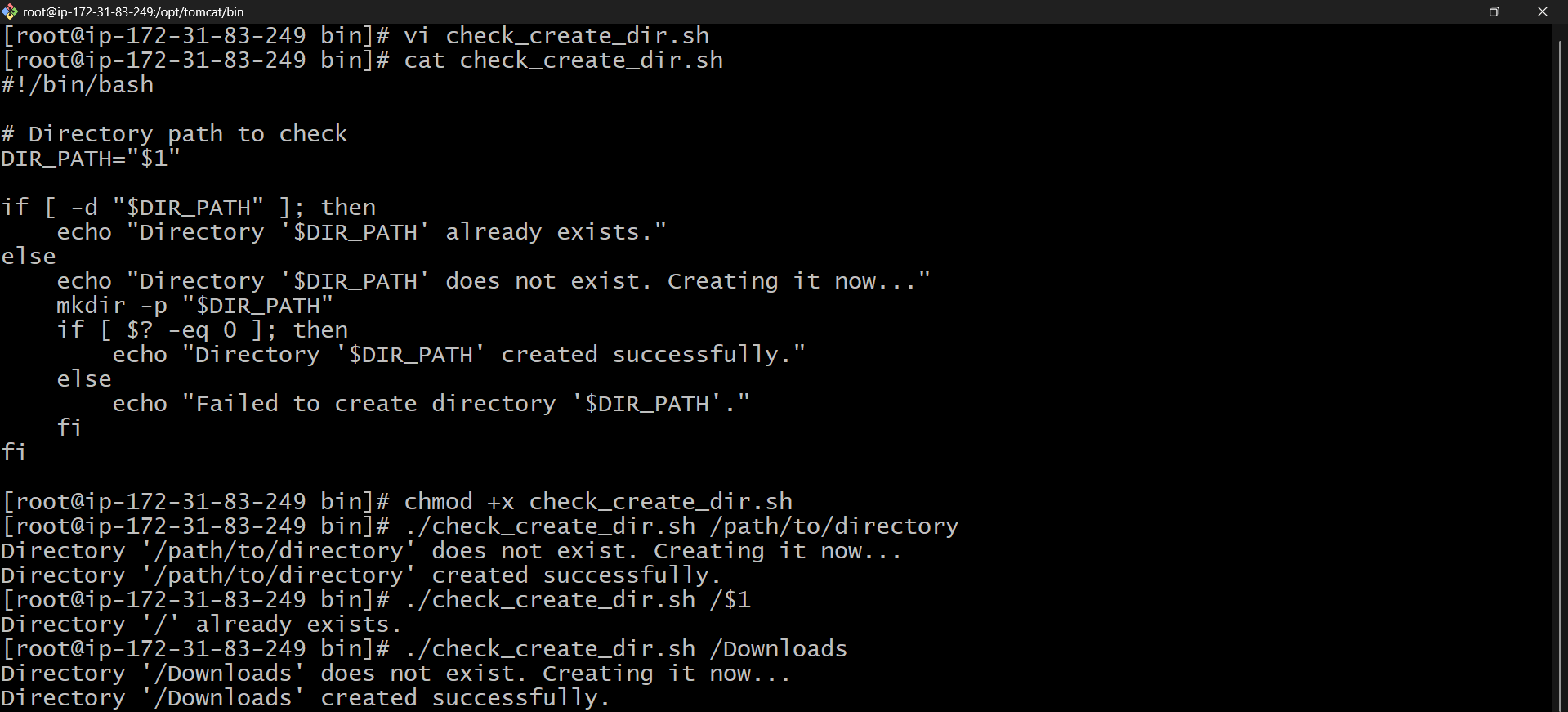
bash

chmod +x check\_create\_dir.sh

1. Run the script with the directory path as an argument:

bash

./check\_create\_dir.sh /path/to/directory

****

**9) Create bash script to delete last 3 lines for a file.**

How to use:

1. Save the script as delete\_last3.sh.
2. Make it executable:

bash

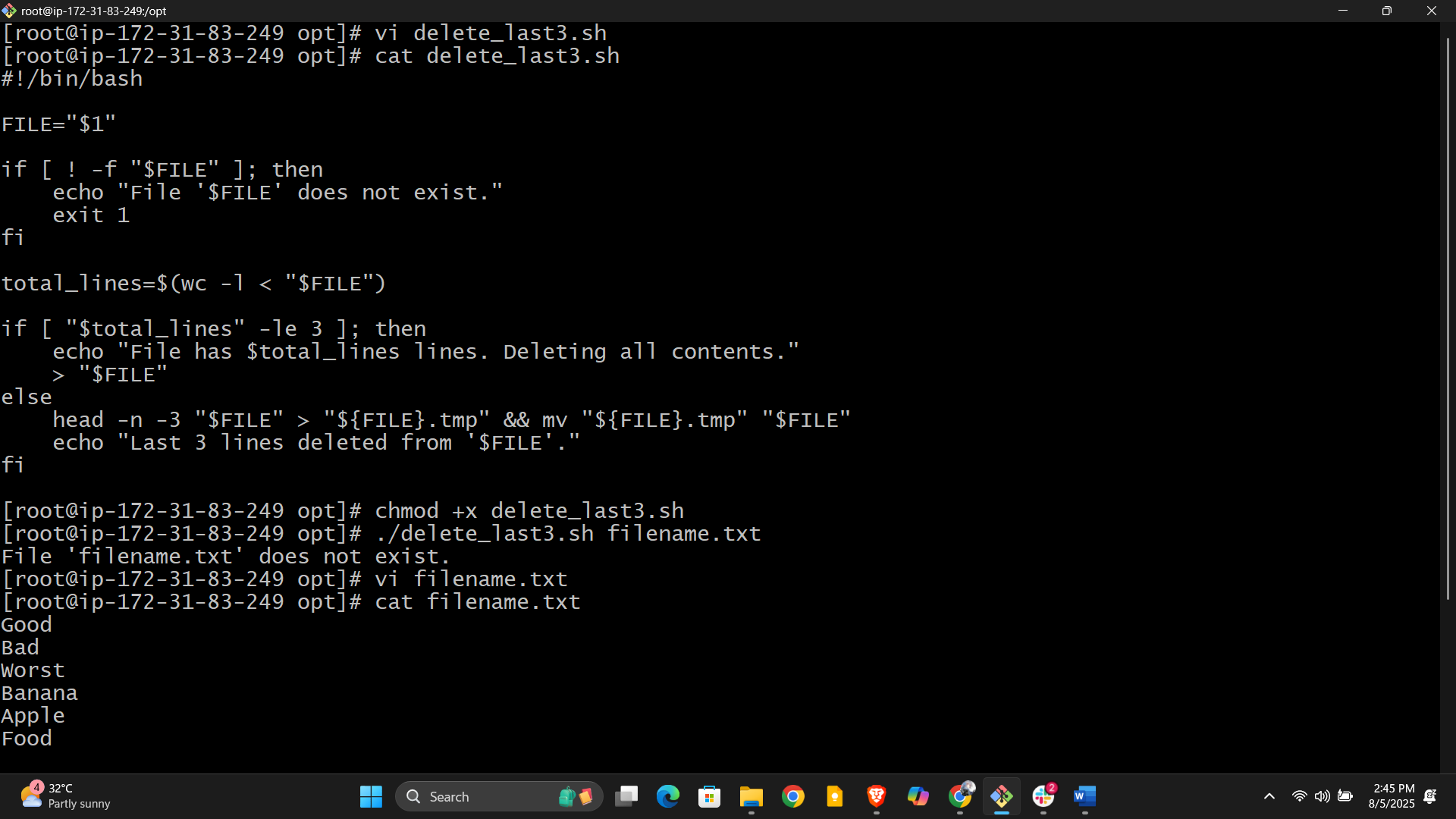
chmod +x delete\_last3.sh

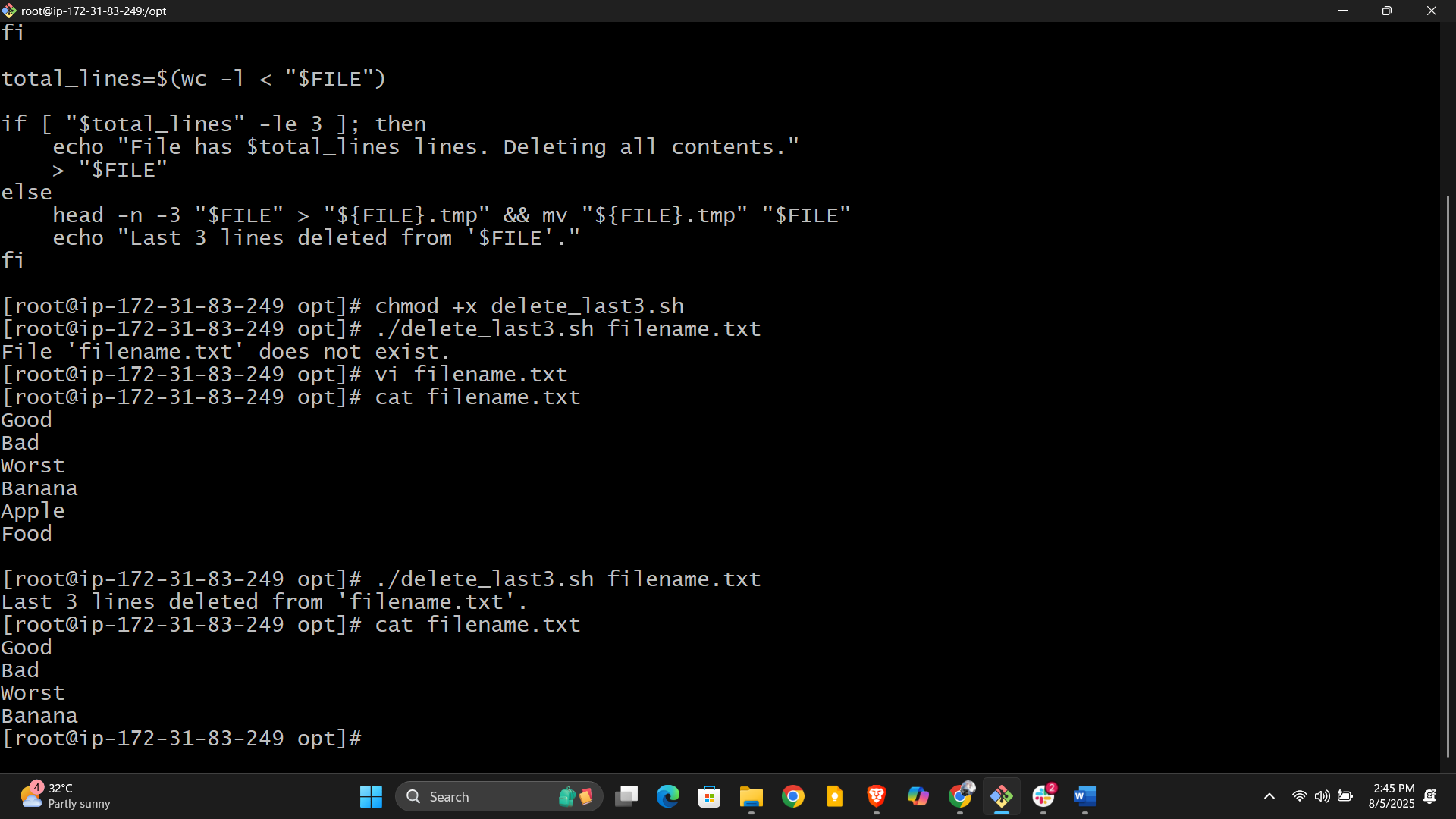
1. Run the script with a filename argument:

bash

./delete\_last3.sh filename.txt

This script safely removes the last 3 lines or clears the file if it has 3 or fewer lines.

****

****

**10) Bash script to monitor cpu and if it is more than 80% then send email notification.**

1. Replace your\_email@example.com with your real email address.
2. Save the script, e.g., cpu\_monitor.sh.
3. Make it executable:

bash

chmod +x cpu\_monitor.sh

1. Test your mail setup (mail command). If mail is not installed, install it using:

bash

sudo dnf install mailx -y *# or yum install mailx -y*

and configure your system to send emails (e.g., using ssmtp or another MTA).

5. Install and configure Postfix (full MTA)

bash

sudo dnf install postfix -y

sudo systemctl start postfix

sudo systemctl enable postfix

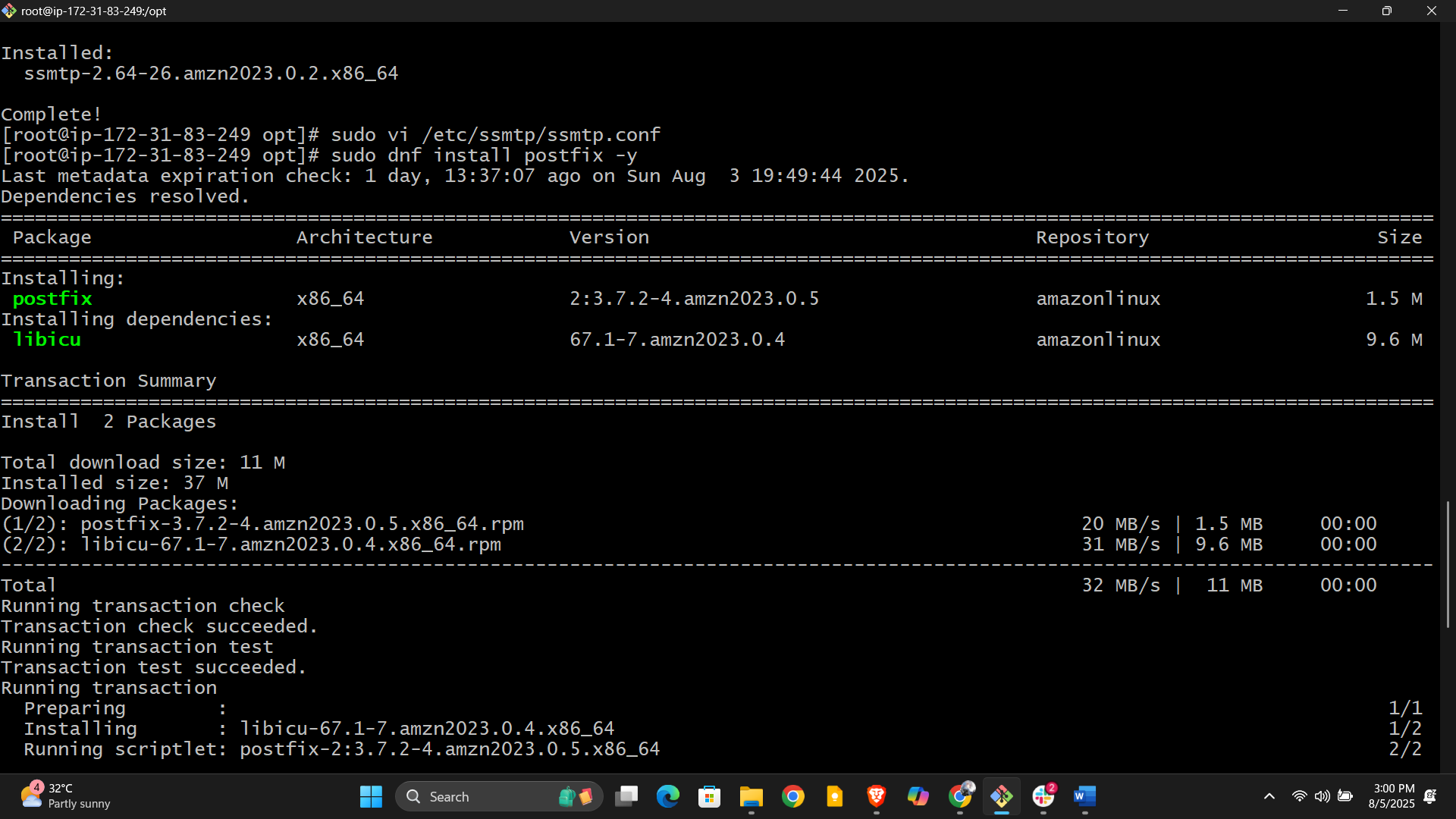
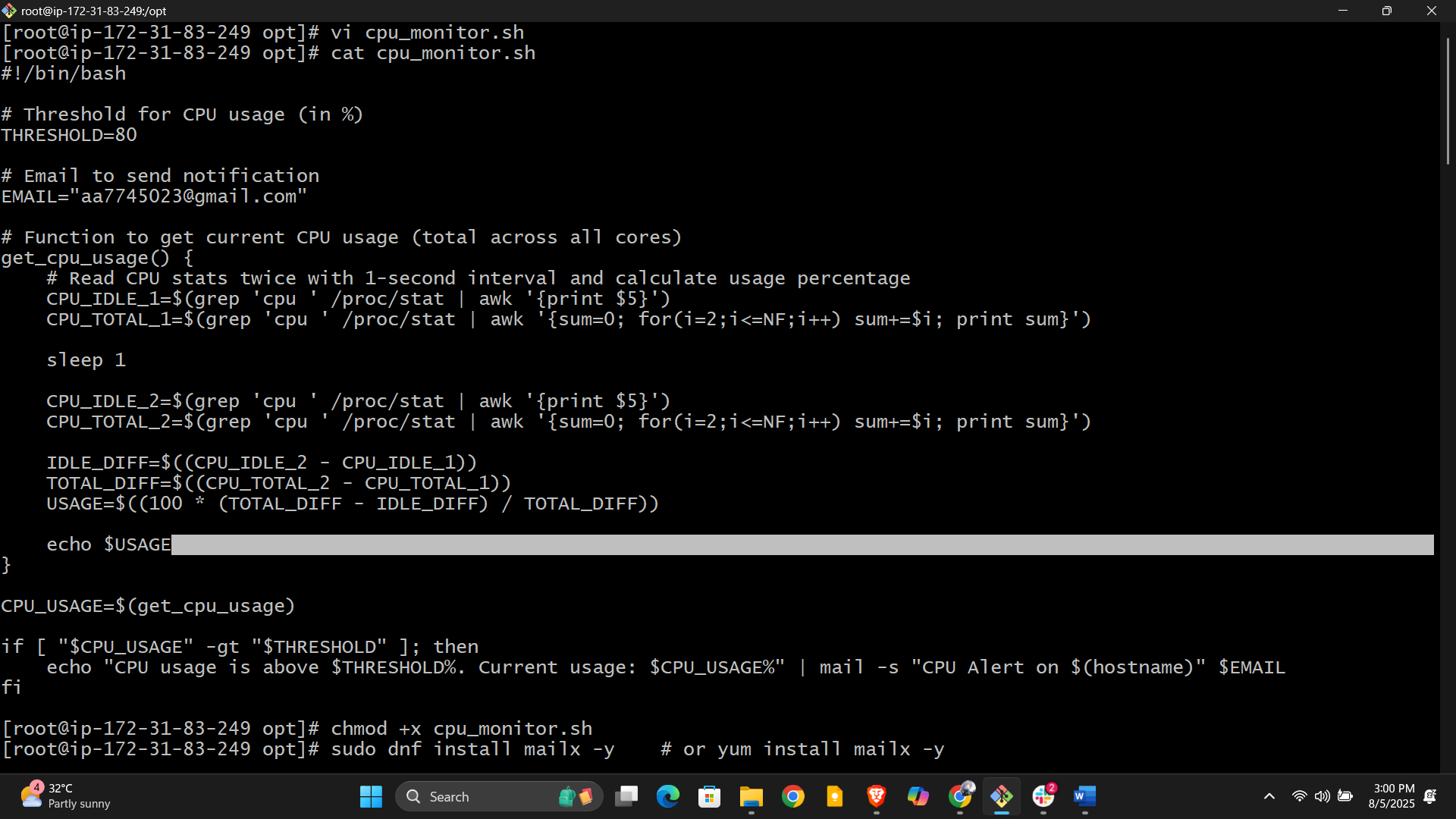
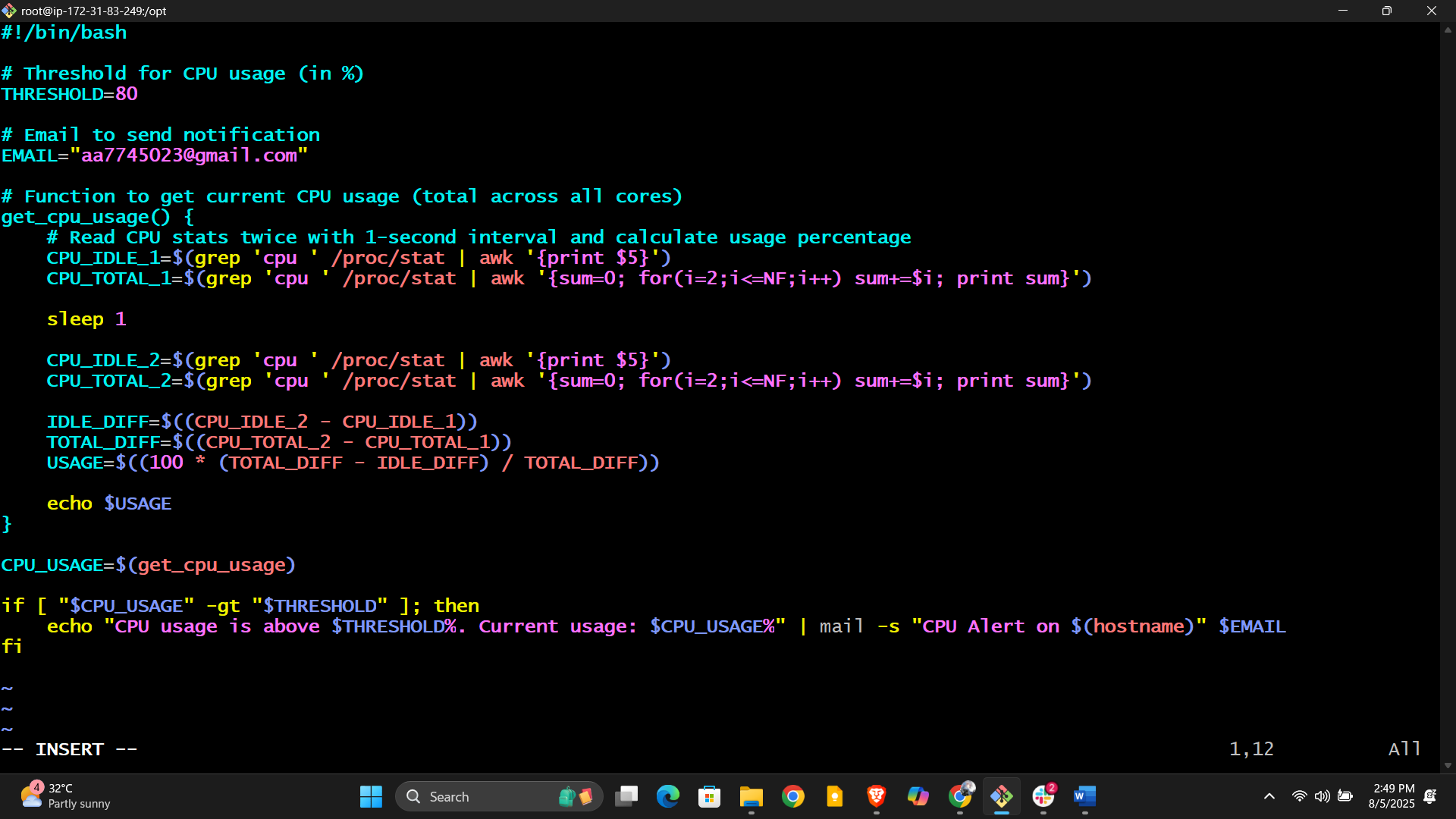
Postfix by default sends emails via local MTA. For external SMTP, additional configuration is needed.

**Then try sending email again:**

bash

echo "Test email from EC2" | mail -s "Test Email" your\_email@gmail.com

**(Note: This local setup may not successfully deliver emails outside without further SMTP relay setup.)**

****

**11) Bash script to monitor disk space and if it is more than 80% then send email   
notification.**

How to use:

1. Replace your\_email@example.com with the actual email address to receive alerts.
2. Save the script as disk\_monitor.sh.
3. Make it executable:

bash

chmod +x disk\_monitor.sh

1. Test your mail setup and run the script manually:

bash

./disk\_monitor.sh

1. Schedule it with a cron job to run periodically (e.g., every hour):

bash

crontab -e

* To see your current crontab entries, use:

bash

crontab -l

Step 6: Verify Email Setup

* Ensure your server’s mail system (e.g., Postfix) is configured and able to send emails.
* Test sending email manually:

bash

echo "Test mail" | mail -s "Test Email" your\_email@example.com

* Check your inbox and spam folder.

Step 5: Automate With Cron

Schedule regular execution:

1. Open crontab:

bash

crontab -e

1. Add a line to run the script every hour (adjust path as needed):

bash

0 \* \* \* \* /full/path/to/disk\_monitor.sh

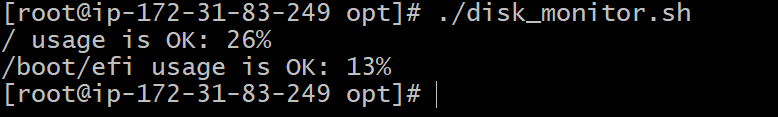
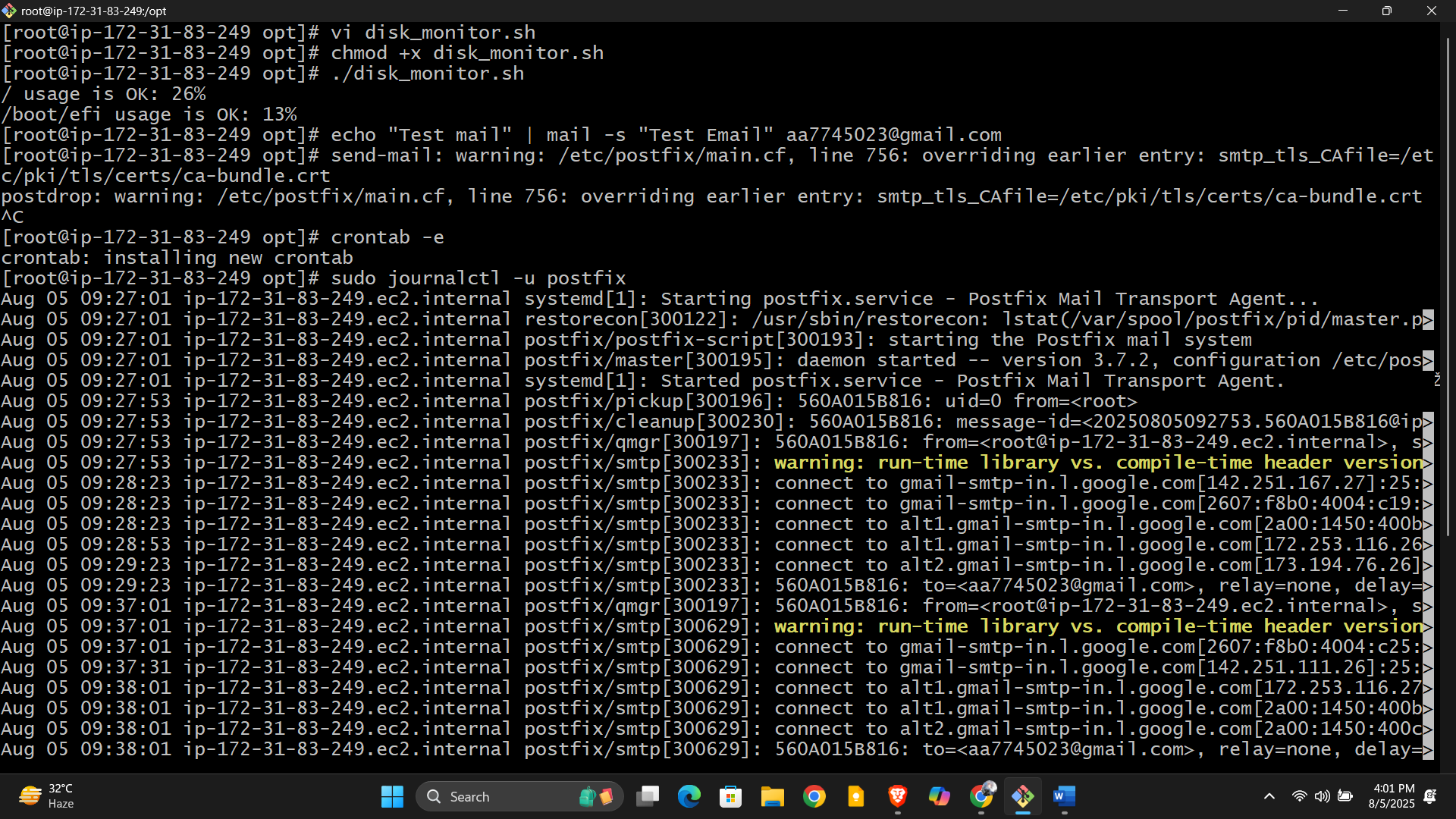
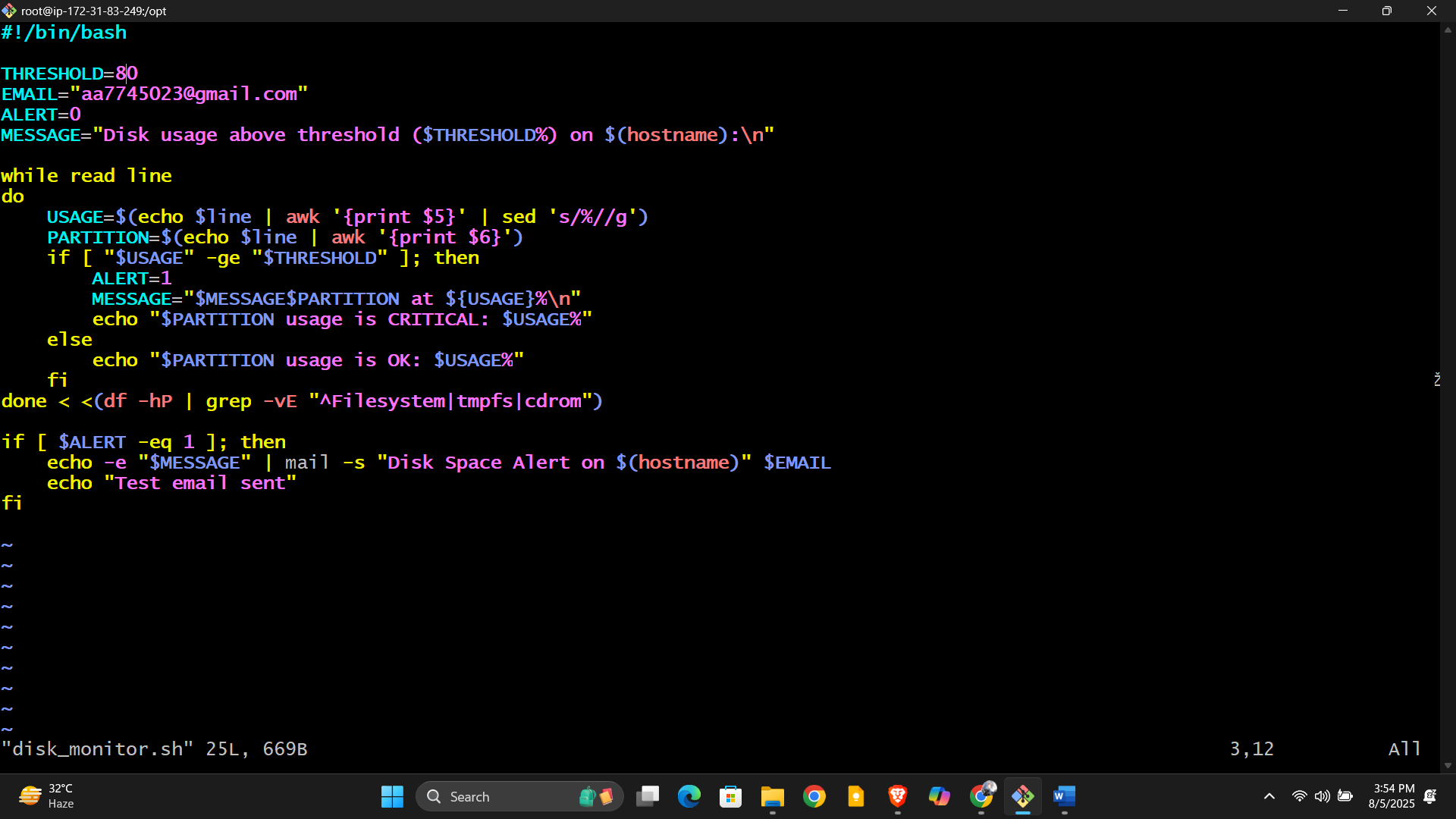
1. Save and exit.

Step 6: Monitor Logs and Adjust

* To debug mail delivery, check:

bash

sudo journalctl -u postfix

****

**12) Bash script to monitor memory and if it is more than 80% then send email notification.**

How to use:

1. Replace your\_email@example.com with your email address.
2. Save as memory\_monitor.sh.
3. Make executable:

bash

chmod +x memory\_monitor.sh

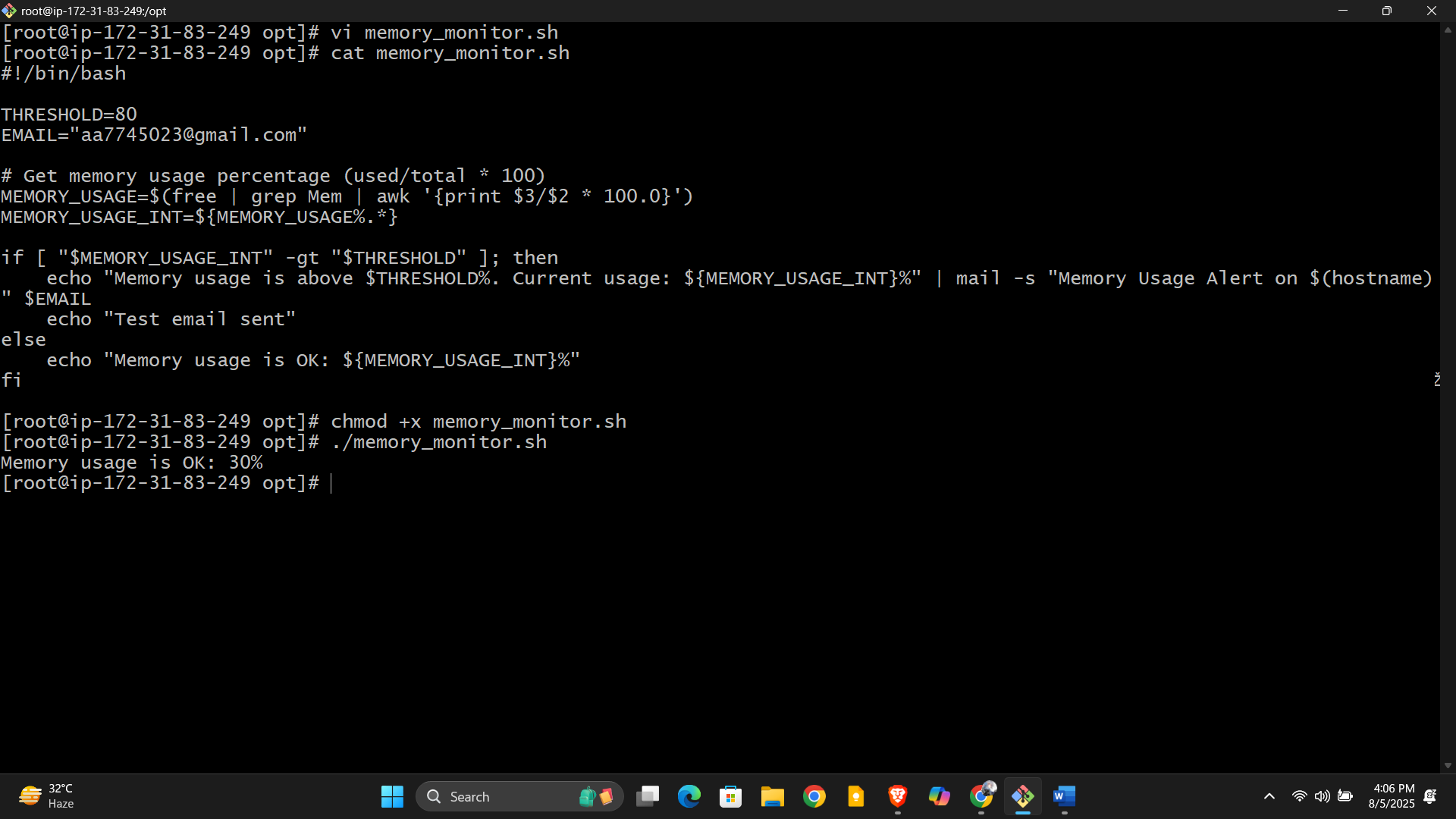
1. Run the script:

bash

./memory\_monitor.sh

1. To run automatically, add it to crontab as described previously.

This script will check RAM usage and send an alert email if usage crosses the 80% threshold.

****