**File Creation:**

* Create an empty file using the touch command:

touch file1

> file1

* Create a text file and add data using the cat command:

cat > file1.txt

Add data here

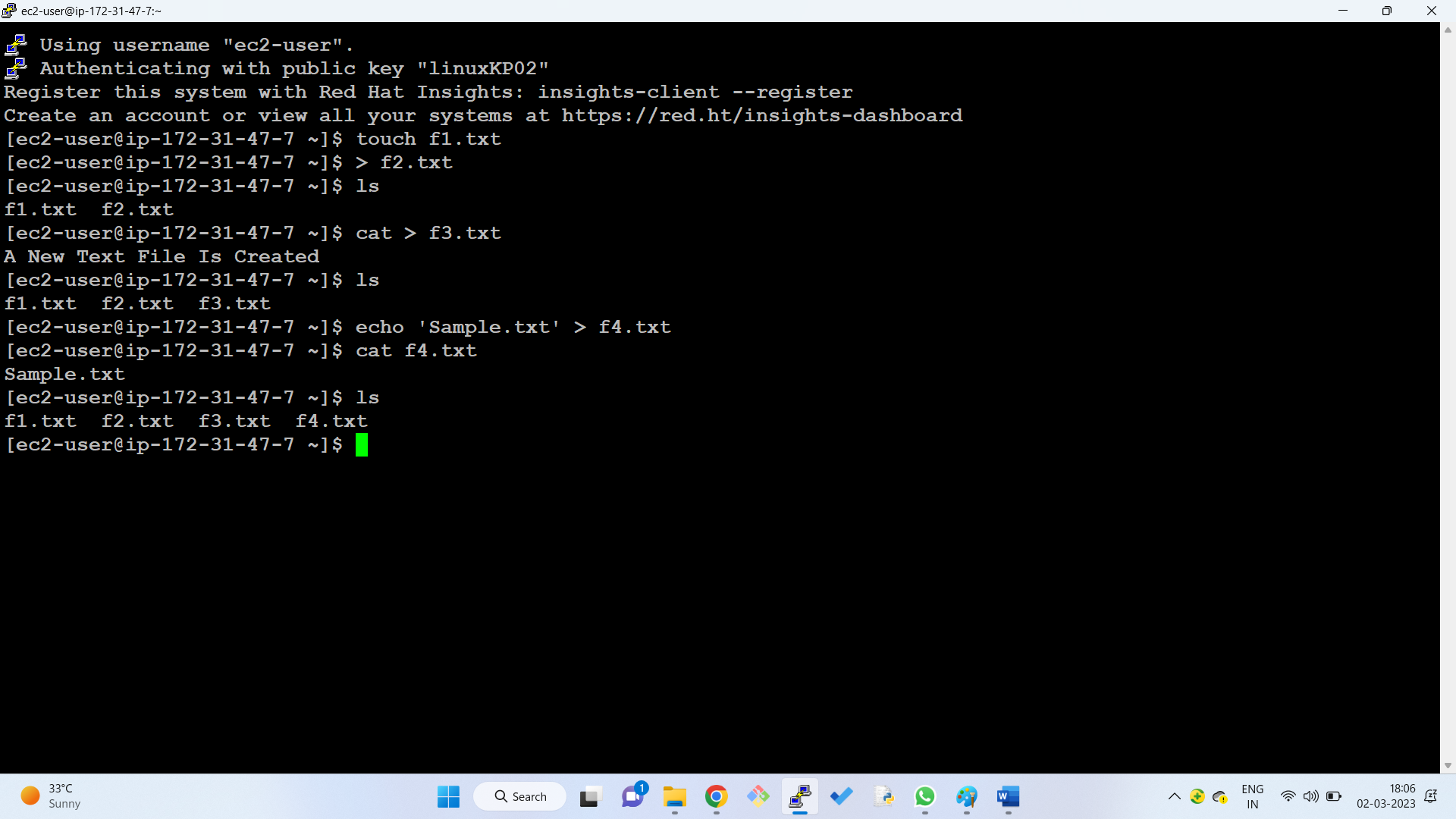
Ctrl+D to save

* Create a text file with data using the echo command:

echo 'Sample text' > file1.txt

cat file1.txt (to see content inside file)

* Create a file using a text editor like vi, vim, or nano.



**Various way to create a file:**

1. With touch command
2. With redirect operator
3. With cat command
4. With echo command
5. Use text editor(vi, vim, nano)

**File Removal:**

* Remove a single file using the rm command:

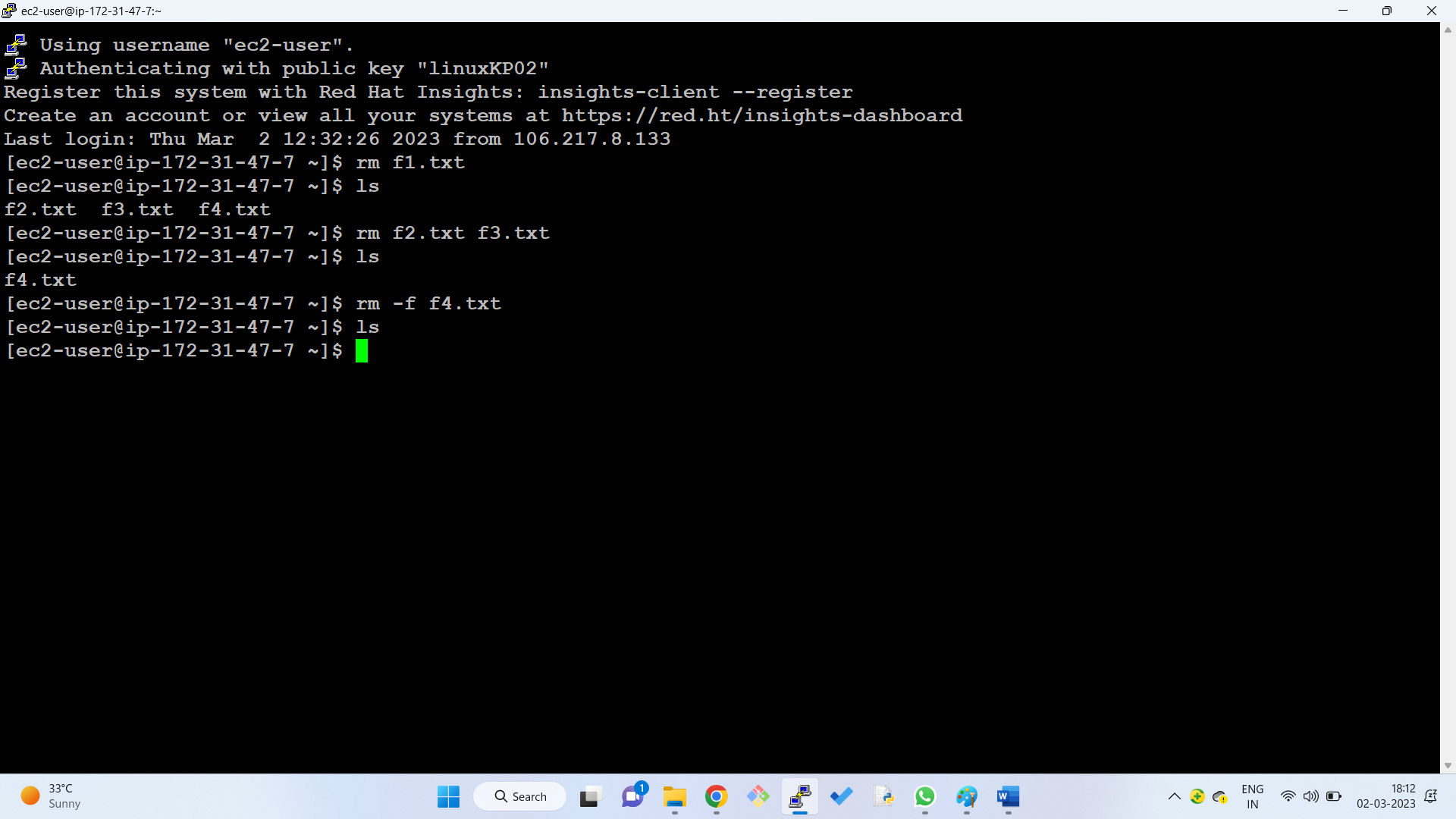
rm file1.txt

* Remove multiple files using the rm command:

rm file2.txt file3.txt

* Force the removal of a write-protected file using the -f flag:

rm -f f4.txt



**Directory Creation:**

* Create a directory using the mkdir command:

mkdir dir1

* Create multiple directories using the mkdir command:

mkdir {test1,test2,test3}

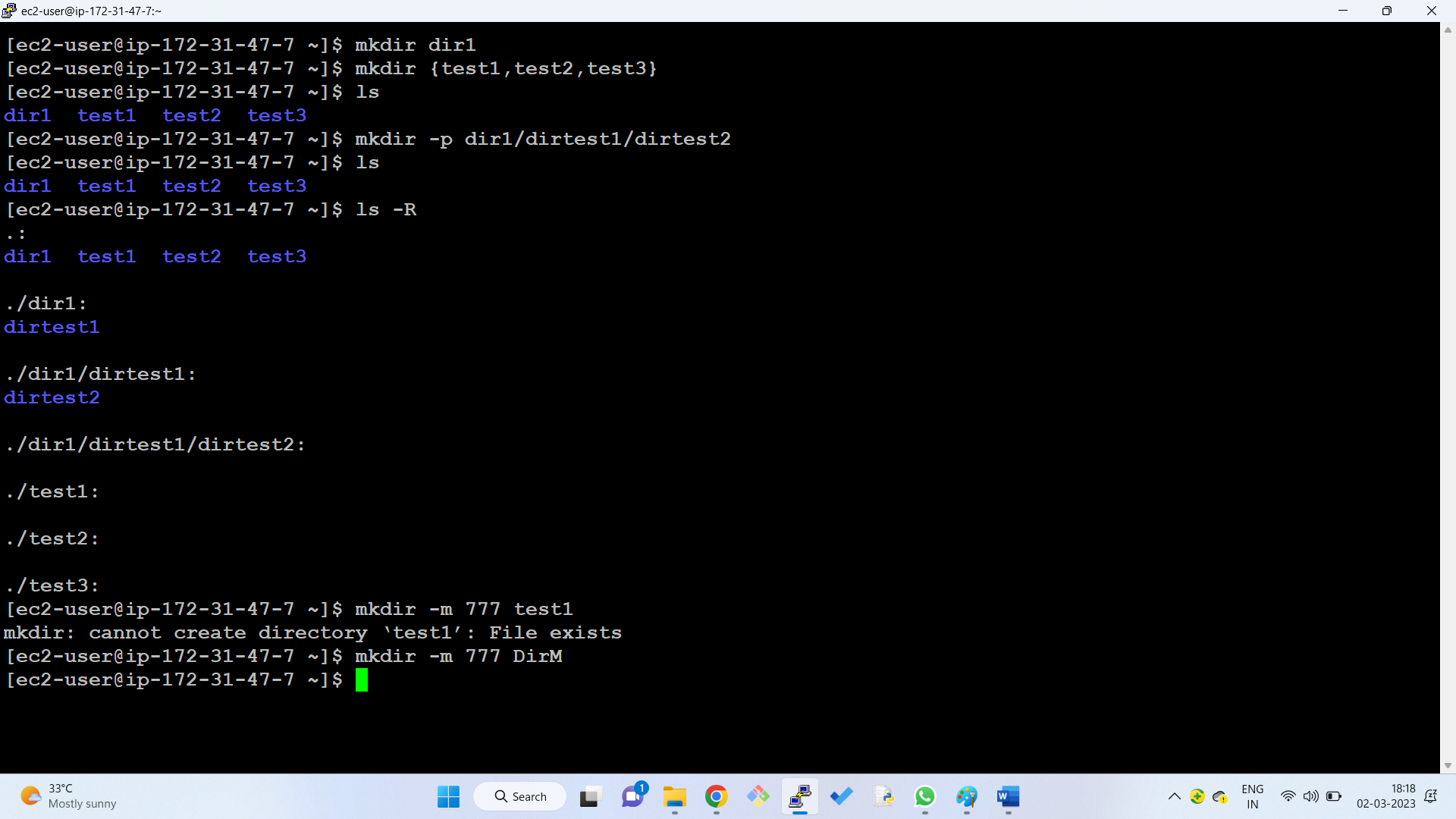
* Create parent directories using the -p flag:

mkdir -p dir1/dirtest1/dirtest2

ls -R (Shows recursive directory tree)

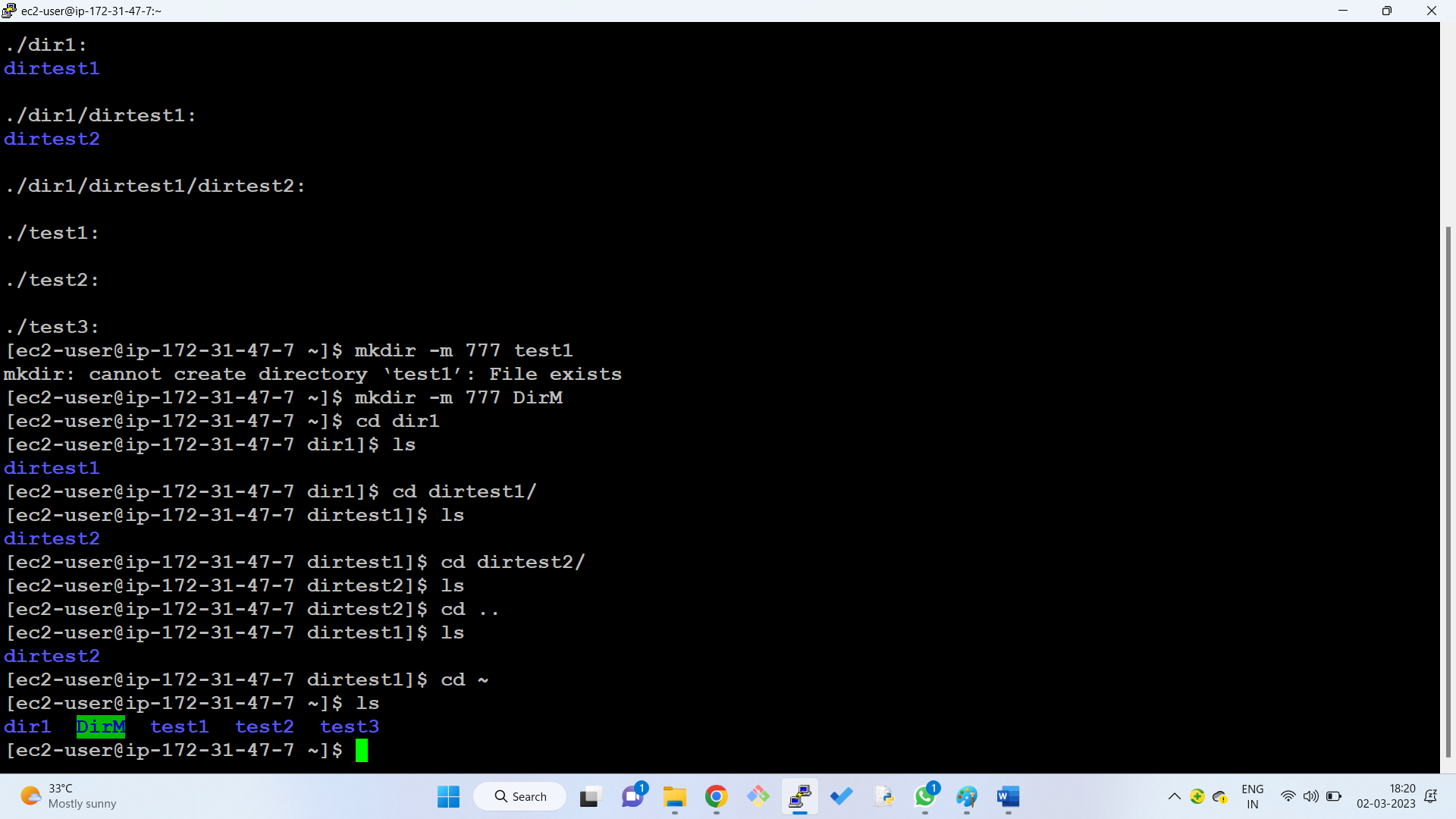
* Set permissions when creating a directory using the -m flag:

mkdir -m 777 dirM



**Navigating Directories:**

* Go to the home directory using the cd command:
* Go to the previous directory using the cd .. command:
* Go to the home directory using the cd ~ command:



**Directory Removal:**

* Remove a directory and its contents using the rm -r command:

rm -r dir1

* Remove a directory without confirmation using the -f flag:

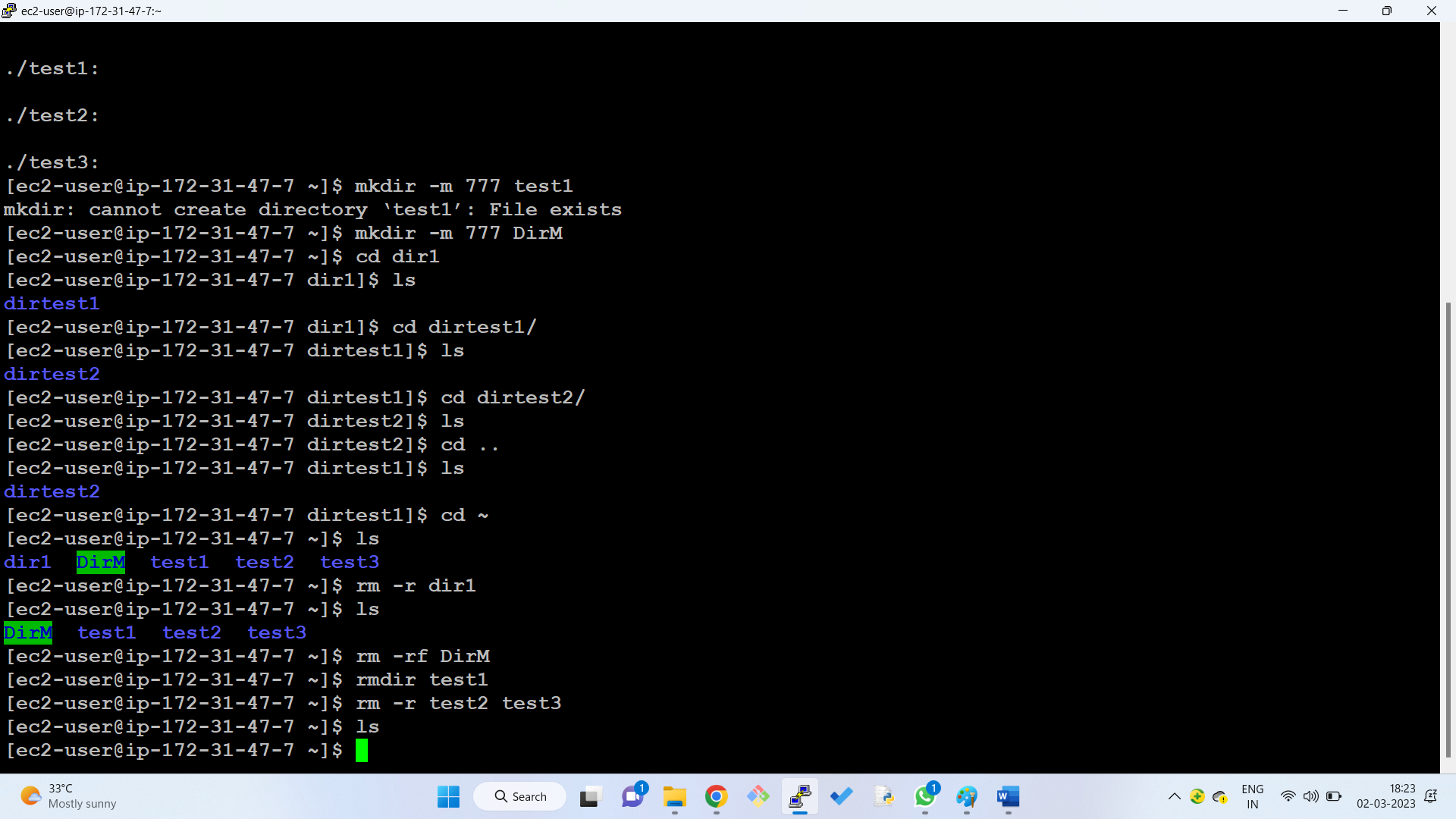
rm -rf dir1

* Remove multiple directories using the rm -r command:

rm -r dir1 dir2 dir3

* Remove empty directories using the rmdir command:

rmdir dir1



**Listing Files and Directories:**

* List files and directories using the ls command:

ls

* List files and directories in long format using the -l flag:

ls -l

* View hidden files using the -a flag:

ls -a

* List files in reverse order using the -r flag:

ls -r

* Recursively list subdirectories using the -R flag:

ls -R

* List files and directories in reverse order, with detailed information using the -ltr flag:

ls -ltr

* List files by file size using the -ls flag:

ls -ls

* Display inode number of file or directory using the -i flag:

ls -i

* List UID and GID of files using the -n flag:

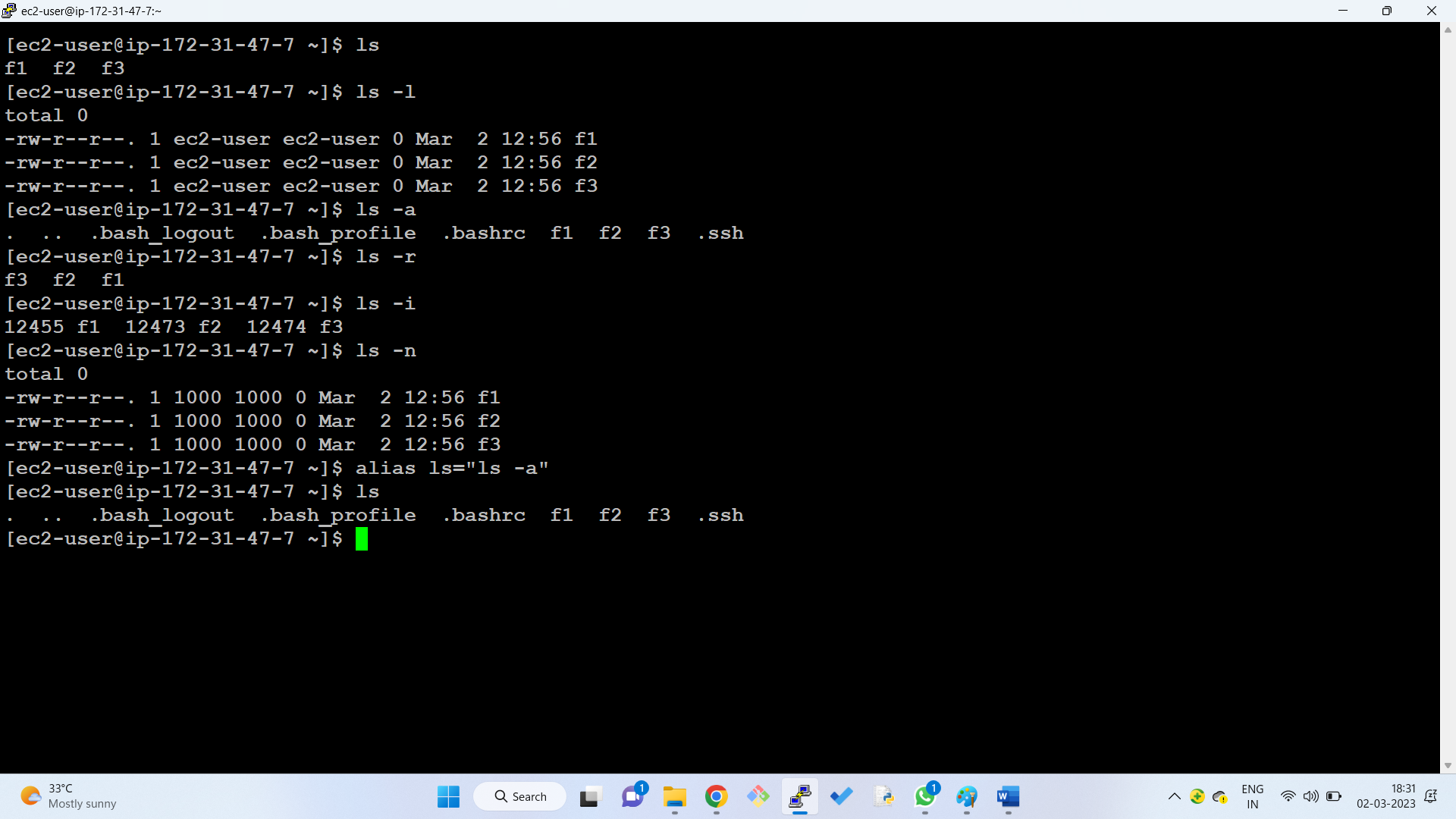
ls -n

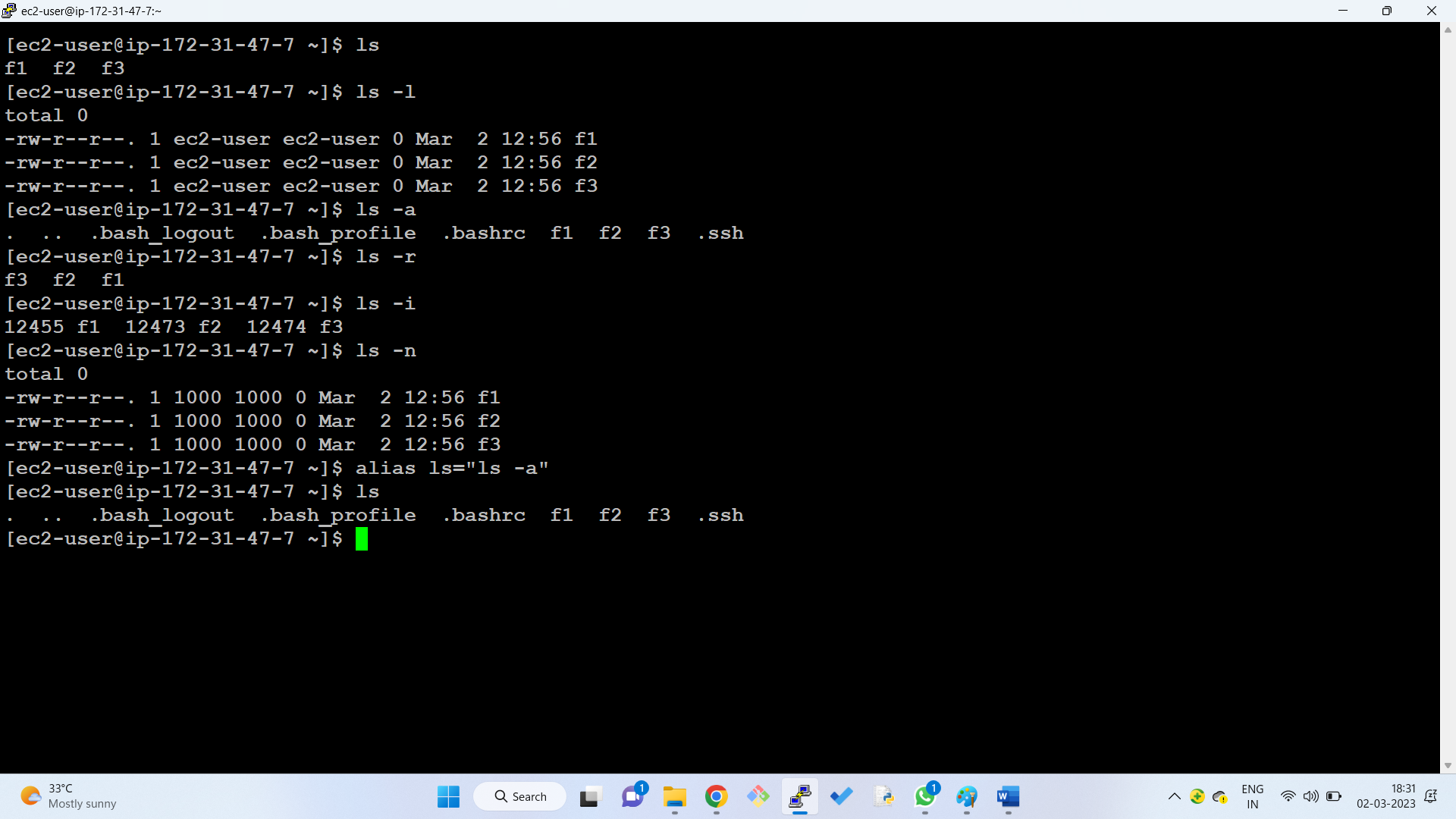
* Get help for the ls command using the --help flag:

ls --help

* Get version information for the ls command using the --version flag:

ls –version





**Copying Files and Directories:**

* Copy a file using the cp command:

cp f1.txt f2.txt

* Copy a file to another directory using the cp command:

cp f1.txt dir1

* Rename and copy a file to a different path using the cp command:

cp f1.txt dir1/f2.txt

* Copy multiple files to a directory using the cp command:

cp f1.txt f2.txt f3.txt dir1

* Copy an entire directory and its contents using the -R flag:

cp -R documents dir1

* Copy files and directories using the rysnc command:

rysnc -a f1.txt /dir1/f1\_backup.txt

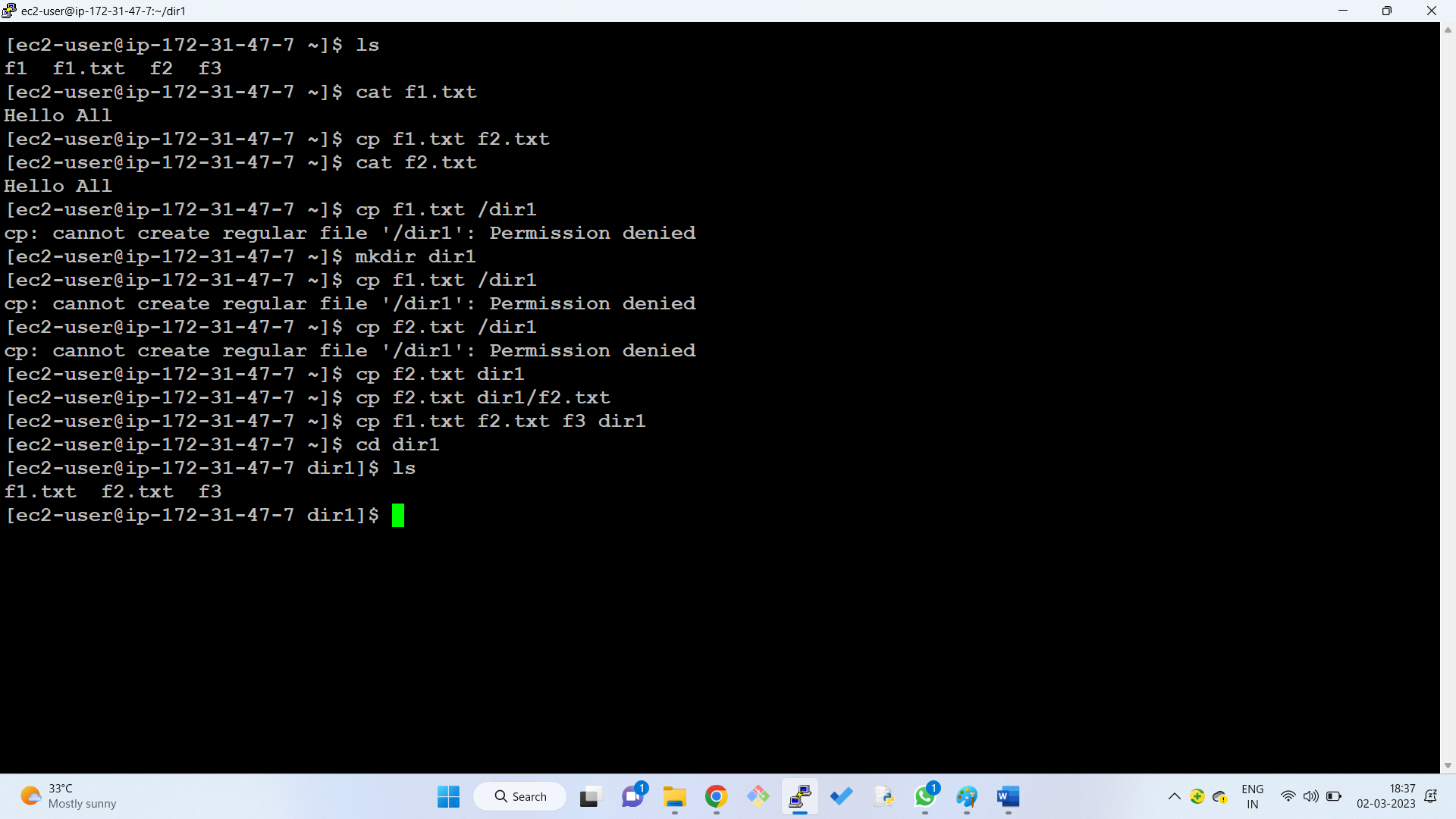
rysnc – synchronize or transfer data between two locations

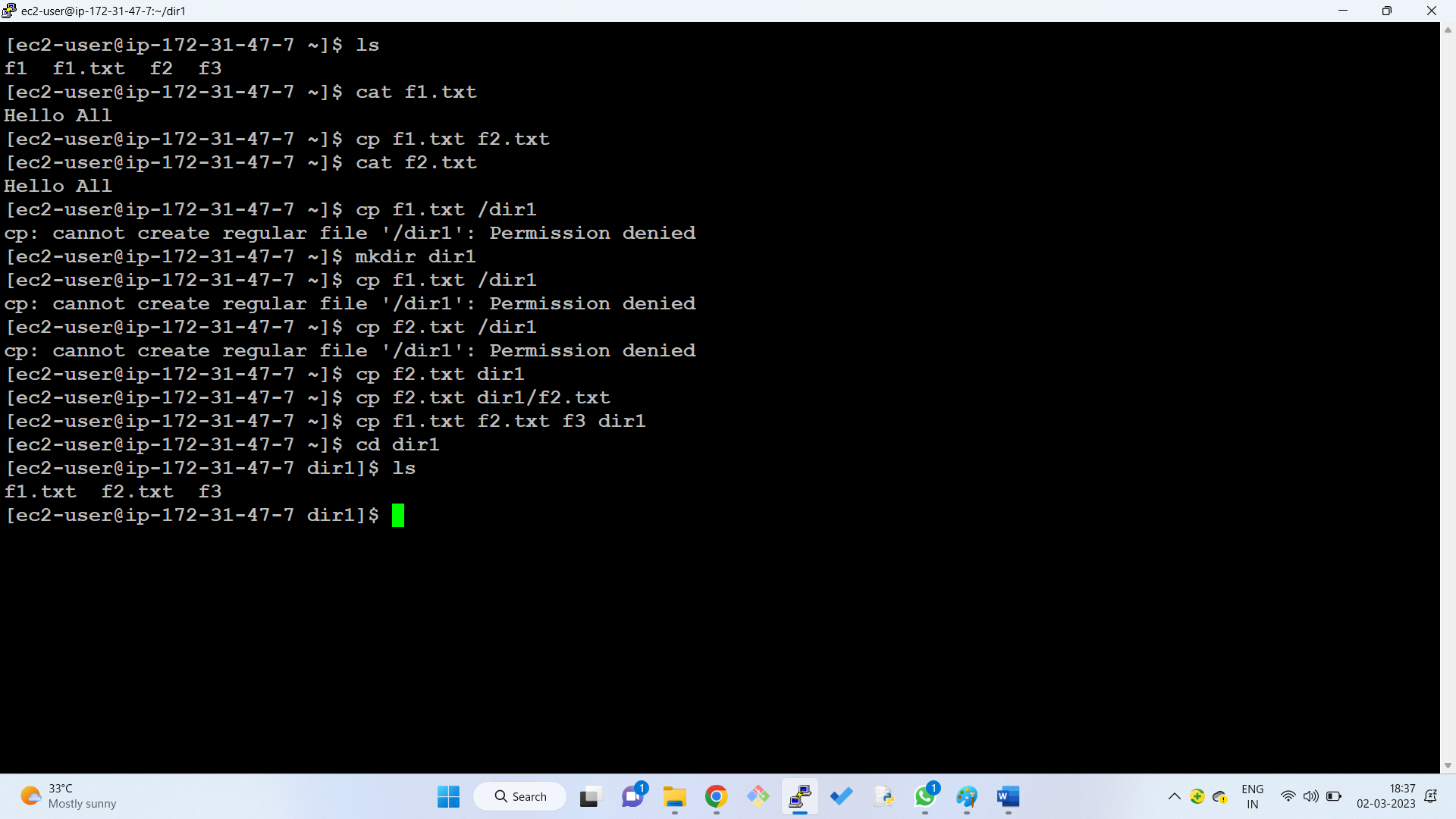
replace f1.txt in working directory, replace /dir1/ with the destination, f1\_backup.txt as the target indicates file will be renamed during copy.

* List contents of a directory using the ls command:

ls dir1

cd dir (change directory)





**Vi Editor: Basic Commands**

**Modes:**

* **Command Mode:** The default mode for Vi. In this mode, you can navigate the file, enter insert mode, and execute commands.
* **Insert Mode:** In this mode, you can type text into the file. Press Esc to exit insert mode.

**Saving and Exiting:**

* Esc:wq: Save the file and quit Vi.
* Esc:q!: Exit Vi without saving the file.

**Common Commands:**

* i: Enter insert mode.
* Esc: Exit insert mode.
* dd: Delete the current line.
* yy: Copy the current line.
* p: Paste the copied line below the current line.

**Navigation:**

* j: Move down one line.
* k: Move up one line.
* h: Move left one character.
* l: Move right one character.
* G: Go to the end of the file.
* gg: Go to the beginning of the file.

**Package Management in Linux**

* Package management plays a crucial role in Linux distributions, enabling users to easily install, update, and remove software packages.
* Two primary package management systems are widely used:

1. APT (Advanced Package Tool) for Debian-based distributions like Ubuntu
2. yum (Yellowdog Updater, Modified) for Red Hat-based distributions like CentOS and Fedora.

**Package Repositories**

* Package repositories serve as centralized locations where software packages are stored and organized.
* These repositories are maintained by the Linux community and contain thousands of packages, each with its dependencies and metadata.

### Installing Packages

To install a package using APT or yum, follow these steps:

1. **Update the repository index:** This ensures your system has the latest package information.

sudo apt-get update (for APT)

sudo yum update (for yum)

1. **Install the package:** Specify the package name using the install command.

sudo apt-get install package\_name (for APT)

sudo yum install package\_name (for yum)

### Removing Packages

To remove a package using APT or yum, follow these steps:

1. **Remove the package:** Specify the package name using the remove command.  
   sudo apt-get remove package\_name (for APT)  
   sudo yum remove package\_name (for yum)
2. **Remove configuration files:** For some packages, configuration files may remain after removing the package. To remove these files, use the purge command.

sudo apt-get purge package\_name (for APT)

sudo yum remove --purge package\_name (for yum)

### Upgrading Packages

To upgrade all installed packages to their latest versions, use the following commands:  
  
sudo apt-get upgrade (for APT)  
sudo yum upgrade (for yum)

For upgrading specific packages, use the install command with the -u flag:  
  
sudo apt-get install -u package\_name (for APT)  
sudo yum update -u package\_name (for yum)

**Note:**

* tree command can be used to visualize the directory structure of installed packages.

**How to create a linux server in AWS Cloud?**

**Step 1**: Create an AWS Account

1. Visit the AWS website at https://aws.amazon.com/ and click on the "Sign In to the Console" button.
2. If you already have an AWS account, enter your credentials and sign in. If you don't have an account, click on the "Create a free account" button and follow the instructions to sign up.

**Step 2**: Launch an EC2 Instance

1. Once you're signed in, navigate to the EC2 (Elastic Compute Cloud) service within the AWS Management Console.
2. Click on the "Launch Instance" button to begin the process of creating a new instance.

**Step 3**: Choose an Amazon Machine Image (AMI)

1. Select the Amazon Machine Image (AMI) for the Linux distribution you want to use. AMIs are pre-configured templates that provide the operating system and software for your EC2 instance. Popular Linux AMIs include Amazon Linux2, Ubuntu, and CentOS.

**Step 4**: Choose an Instance Type

1. Choose an instance type based on the requirements of your workload. Instance types define the hardware resources, such as CPU, memory, and storage, that will be allocated to your instance. Select an instance type that can adequately handle the expected traffic and processing demands of your application.

**Step 5**: Create or Select a Key Pair

1. A key pair is a set of cryptographic keys that allow you to securely connect to your Linux instance using SSH. You can either create a new key pair or select an existing one from your AWS account.

**Step 6**: Configure Network Settings (Optional)

1. You can customize the network settings for your instance, such as assigning a public IP address and configuring security groups. These settings determine how your instance can be accessed from the internet and other resources within your AWS VPC (Virtual Private Cloud).

**Step 7**: Choose the Number of Instances

1. Specify the number of instances you want to create. This is relevant if you need to deploy a scalable application or distribute workloads across multiple instances.

**Step 8**: Launch the Instance

1. Review your configuration and click on the "Launch Instances" button. AWS will begin provisioning the EC2 instance based on your selected parameters.

**Step 9**: Connect to the Instance

1. Once the instance is in the "Running" state, you can connect to it either by using AWS Console or using an SSH client. You will need to use the private key from your key pair to authenticate the connection.

**Note:**

* You can monitor the status and performance of your EC2 instance using the EC2 dashboard and CloudWatch metrics.
* AWS offers various tools and services to manage your Linux instances, including Elastic Beanstalk for application deployment, CloudFormation for infrastructure automation, and Lambda for serverless computing.