Nano Editor
Nano is a simple, user-friendly text editor for Unix-like operating systems.
Here are some of the basic commands and key bindings to help you get started with Nano:

Creating a new file using the `nano` text editor
To create and edit a new file named `myfile.txt`, for example, type:
nano myfile.txt

Starting Nano

To open a file with Nano: nano filename

Basic Navigation

- Arrow Keys: Move the cursor up, down, left, and right.
- Ctrl + A: Move to the beginning of the line.
- Ctrl + E: Move to the end of the line.
- Ctrl + Y: Move up one screen.
- Ctrl + V: Move down one screen.

Editing Text

- Ctrl + K: Cut (delete) the current line.
- Ctrl + U: Uncut (paste) the last cut line.
- Ctrl + ^ (Ctrl + Shift + 6): Set mark (begin selection).
- Ctrl + K: Cut selected text.
- Ctrl + U: Paste the text in the clipboard.
- Alt + 6: Copy selected text.

Searching and Replacing

- Ctrl + W: Search for text.
- **Ctrl + **: Search and replace text.

File Operations

- Ctrl + O: Write (save) the file.
- Ctrl + X: Exit Nano. If there are unsaved changes, Nano will prompt you to save them.

Other Commands

- Alt + U: Undo the last action.
- Alt + E: Redo the last undone action.

Saving and Exiting

To save a file, press Ctrl + O, then press Enter to confirm. To exit Nano, press Ctrl + X. If you've made changes to the file, Nano will ask if you want to save them.

Accessing the Terminal

Command	Description
ls	Lists files and directories in the current directory.
cd	Changes the current directory.
pwd	Prints the current working directory's path.
mkdir	Creates one or more directories.
touch	Creates empty files or updates file timestamps.
rm	Removes files and directories.
cp	Copies files and directories.
mv	Moves or renames files and directories.

PM2:

- PM2 is an advanced process manager for NodeJS applications that allows you quickly start, control, or stop your node processes. It runs as a daemon on the server and will make sure your app is available 24/7/365!
- PM2 is like a helpful robot for computers. Imagine you have a toy that you want to keep playing with all the time, but sometimes it stops working. PM2 is a special robot that watches your toy and makes sure it keeps working, even if it stops or has a problem. It's used by people who build websites and apps to make sure their work keeps running smoothly without any breaks.
- Maintain other info, such as PID of the process its current status, and memory usage
- Application that are running under PM2 will be restarted automatically if the application crashes or killed

Installation

```
npm install -g pm2
```

Now the `pm2` command should be available via command line and you can check the installed version via `pm2 -v`

How to start a node application with PM2

`pm2 start app.js --name my-api`	Start and name a process

Cluster mode

`pm2 start app.js -i 0`	Will start maximum processes with LB depending on available CPUs

Listing

`pm2 list`	Display all processes status
`pm2 jlist`	Print process list in raw JSON
`pm2 prettylist`	Print process list in beautified JSON
`pm2 describe 0`	Display all information about a specific process
`pm2 monit`	Monitor all processes

Logs

pm2 logs [--raw]	Display all processes logs in streaming
pm2 flush	Empty all log files
pm2 reloadLogs	Reload all logs

ctions

`pm2 stop all`	Stop all processes
`pm2 restart all`	Restart all processes
`pm2 reload all`	Will 0s downtime reload (for NETWORKED apps)
`pm2 stop 0`	Stop specific process id
`pm2 restart 0`	Restart specific process id
`pm2 delete 0`	Will remove process from pm2 list
`pm2 delete all`	Will remove all processes from pm2 list
`pm2 save`	Save processes list to respawn at reboot
`pm2 reset`	reset the restart counter
`pm2 reset all`	reset all restart counters

For more details : https://pm2.keymetrics.io/docs/usage/process-management

Connecting Linux server to local machine using .pem file

- PEM file (which stands for Privacy Enhanced Mail) to connect to your Linux server, it means you are using a key-based authentication instead of a password.

Step 1: Install an SSH Client

- Windows: Download and install an SSH client like PuTTY.
- Mac/Linux: You already have an SSH client built-in.

Step 2: Get Your Server's Information

- You need the IP address of your server.
- You need your username for the server.
- You need the PEM file provided by your server's administrator.

Step 3: Set Permissions for the PEM File

- Mac/Linux:
  - Open Terminal.
  - Navigate to the directory where your PEM file is located.
  - Run the command: chmod 400 yourfile.pem (replace yourfile.pem with the actual file name).
- Windows:
  - No need to set permissions in this way, skip to the conversion step if using PuTTY.

Step 4: Convert PEM File (Windows Only)

If you're using Windows with PuTTY, you need to convert the PEM file to a PPK file.

- Open PuTTYgen (installed with PuTTY).
- Click "Load" and select your PEM file.
- Click "Save private key" to save it as a PPK file.

Step 5: Open the SSH Client

- Windows (PuTTY): Open PuTTY.
- Mac/Linux: Open the Terminal application.

Step 6: Enter Server Information

- Windows (PuTTY):

    - In the "Host Name" field, type your server's IP address.
    - In the "Category" section, go to "Connection" > "SSH" > "Auth".
    - Click "Browse" and select your PPK file.
    - Go back to "Session" and click "Open".
- Mac/Linux:
    - In the Terminal, type: ssh -i /path/to/your/file.pem username@IP_address (replace /path/to/your/file.pem with the path to your PEM file, username with your actual username, and IP_address with the server's IP address).
    - Press Enter.

Step 7: Accept the Connection

- The first time you connect, you might see a message asking if you trust the server. Type "yes" and press Enter.

Step 8: You're Connected!

- If everything is correct, you should now be connected to your Linux server. You can now type commands to interact with the server.

Installing Nodejs in linux server

1) curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.7/install.sh | bash
2) loading nvm: source ~/.bashrc
3) installing: nvm install --lts

for installing git in linux:

1. sudo yum update
2. Install git in Amazon EC2 instance:sudo yum install git

setup git:

1. git config --global user.name "Your Name"
2. git config --global user.email "mailto:youremail@domain.com"

for ubuntu:

1. sudo apt update
2. sudo apt install git

Copy folder from ubuntu server/linux to local machine

To copy a folder from the server to your local machine, you can use the `scp` (secure copy) command.

Steps:

1. scp -i /path/to/yourfile.pem -r username@IP_address:/path/to/remote/folder /path/to/local/destination
    a. Replace `/path/to/yourfile.pem` with the path to your PEM file.
    b. Replace `username` with your server's username.
    c. Replace `IP_address` with your server's IP address.
    d. Replace `/path/to/remote/folder` with the path to the folder on the server you want to copy.
    e. Replace `/path/to/local/destination` with the path where you want to save the folder on your local machine.

To upload a file from your local machine to an Ubuntu or Linux server

1) scp -i /path/to/yourfile.pem /path/to/local/file username@IP_address:/path/to/remote/destination
    a) Replace `/path/to/yourfile.pem` with the path to your PEM file.
    b) Replace `/path/to/local/file` with the path to the file on your local machine that you want to upload.
    c) Replace `username` with your server's username.
    d) Replace `IP_address` with your server's IP address.
    e) Replace `/path/to/remote/destination` with the path on the server where you want to save the file.

To open a new port on an Ubuntu or Linux server
Using UFW (Uncomplicated Firewall)
check Current Firewall Status:
checking the current status of UFW to see if it's active: sudo ufw status

Allow the New Port:
If UFW is active and you want to open a new port (e.g., port 12345), run:
sudo ufw allow 12345/tcp

Note:

- After opening a new port, ensure that any applications or services you intend to use with that port are configured correctly to listen on it.
- Always be cautious when modifying firewall rules, especially on production systems, to avoid unintended security risks.

Closing port
Check Current Firewall Status: First, check the current status of UFW to see which ports are open: sudo ufw status numbered

Delete the Rule for the Port: Identify the rule number associated with the port you want to close from the output of the previous command. Then, delete the rule using its number: sudo ufw delete [rule_number]

Replace `[rule_number]` with the actual number of the rule you want to delete.

Verify the Change: Verify that the rule has been removed: sudo ufw status numbered

Note:

- Always be cautious when modifying firewall rules, especially on production systems, to avoid unintended security risks.
- After closing a port, ensure that any applications or services no longer need that port for communication.