

Node.js deployment with pm2

For running **Node.js** applications in production, you need **process manager** who takes care of the node processes.

We explain how to use the [PM2](#) for setting up a **Node.js Application** for production on this site.

PM2

[PM2](#) is a daemon process manager that will help you manage and keep your application online.

Installation

1. Log in via [SSH](#) with your **Site User**:

```
ssh john-doe@instance-ip-address
```

2. Go to the **root directory** of your project:

```
cd htdocs/www.domain.com/
```

3. Install the latest **pm2** via **npm**.

```
npm install pm2@latest -g
```

Start the Application

Use the following command to start your **application** via **pm2**:

APP NAME

Replace the **app-name** variable with the name of your application.

```
pm2 start npm --name $app-name -- start
```

Your **application** is now running via **pm2**.

[illegible]

Save Configuration

To save the **pm2** configuration, execute the **save** command:

pm2 save

Adding a Cron Job

To ensure, that your **application** is running after a reboot of your instance, you need to configure a cron job.

1. First copy the output of the **PATH** variable:

```
echo $PATH
```

The output will look similar to this:

```
/home/john-doe/.nvm/versions/node/v14.19.3/bin:/usr/local/sbin:/usr/local/bin:/u
```

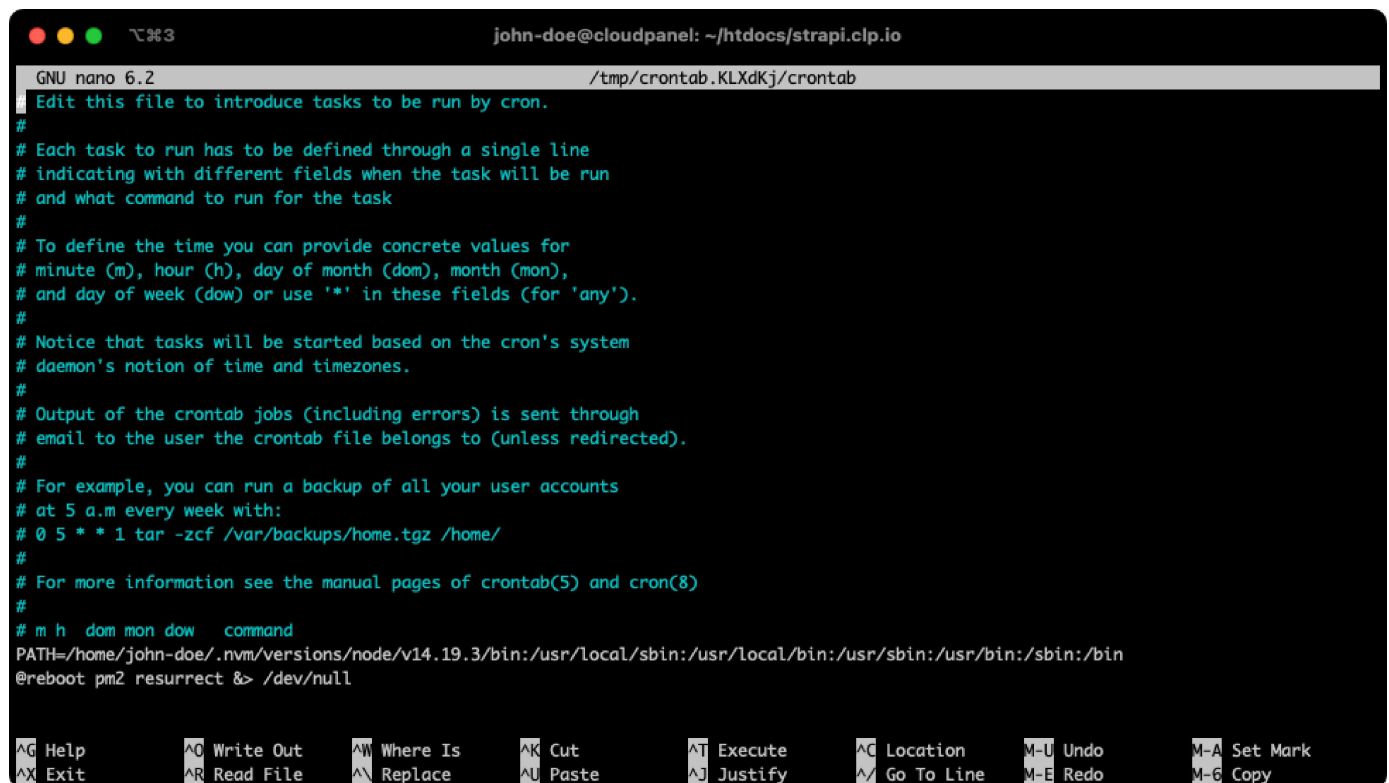
2. Edit the **crontab** for the **site user**.

```
crontab -e
```

3. Add the following lines to it:

```
PATH=$PASTE_THE_OUTPUT_OF_$PATH
@reboot pm2 resurrect &> /dev/null
```

Example configuration



```
john-doe@cloudpanel: ~/htdocs/strapi.clp.io
GNU nano 6.2 /tmp/crontab.KLXdKj/crontab
Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow  command
PATH=/home/john-doe/.nvm/versions/node/v14.19.3/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
@reboot pm2 resurrect &> /dev/null
```

4. Reboot your **instance** and check if the application is **running**:

```
pm2 status
```

The status should be **online** to confirm that your Application is running after reboot.

```
john-doe@cloudpanel: ~  


| id | name   | namespace | version | mode | pid  | uptime | u | status | cpu | mem    | user     | watching |
|----|--------|-----------|---------|------|------|--------|---|--------|-----|--------|----------|----------|
| 0  | my-app | default   | 0.39.1  | fork | 1112 | 20s    | 0 | online | 0%  | 53.2mb | john-doe | disabled |

  
john-doe@cloudpanel:~$
```

Troubleshooting

PM2 provides a [logs](#) command to see the application's output, which helps troubleshoot.

```
pm2 logs
```

 [Edit this page](#)