

How to run jupyter notebook in the background ? No need to keep one terminal for it

Asked 5 years ago Modified 5 months ago Viewed 111k times

75

▲▼

🔖


🔄

Often we run `jupyter notebook` to pop up a page in browser to use notebook. However, the terminal opening the server remains there. Is there a way that we can close that terminal with server running in the back ?


jupyter-notebook

Share Edit Follow

edited Jun 16, 2020 at 16:25

 [desertnaut](#)
55.3k 21 132 163

asked Nov 16, 2017 at 13:34

 [Xingdong](#)
1,215 2 13 17

13 Answers

Sorted by:

Highest score (default) ▾

62

▲▼

🔖

✓

🔄

You can put the process into the background by using `jupyter notebook --no-browser & disown`. You can close the terminal afterwards and the process will still be running.

If you're using `zsh` you can also use a shorter version that does the same: `jupyter notebook --no-browser &!`.

To kill the process you can use `pgrep jupyter` to find the PID of the process and then `kill 1234`, replacing 1234 with the PID you just found.

Explanation

The `--no-browser` flag makes jupyter not open the browser automatically, it also works without this flag.


The `&` puts it into the background of the currently running shell.

The `disown` then removes the job from the background of the currently running shell and makes it run independently of the shell so that you may close it.

In the `zsh` version the `&!` is a built-in function that does the same as `& disown`.

Share Edit Follow

edited Jun 15 at 15:48

 [pgrosslicht](#)
1,465 13 17

4 Can you please give an example? – [rainman](#) Oct 12, 2018 at 0:34

5 How to close it after? – [Pablo Ruiz Ruiz](#) Nov 21, 2019 at 10:32

1 @PabloRuizRuiz ps aux | grep jupyter and kill 1234 ? – [Jivan](#) Mar 2, 2020 at 8:51

2 Using this method, closing the terminal shuts down jupyter server for me. Any reasons as to why this might be happening? – [Nitin Siwach](#) Feb 6, 2021 at 6:32

1 This does not work, and the answer lacks details. Closing Jupyter closes the task. – [Arka Mukherjee](#) Dec 10, 2021 at 2:55



47



Tmux is a good option available to run your Jupyter Notebook in the background.

I am running my Jupyter Notebook on a google cloud platform's VM instance with OS: Ubuntu 16.0. Where

1. I have to start SSH terminal
2. Then run the command: `jupyter-notebook --no-browser --port=5000` . It will start the Jupyter Notebook on port number 5000 on that VM instance
3. Then I open my browser and type `ip_addresses_of_my_instance:port_number` which is 5000. It will open my Notebook in the browser.

Now up to this, all is good. But wait if the connection with my SSH terminal is terminated then the Jupyter Notebook stops immediately and hence I have to re-run it once again once the ssh terminal is restarted or from new ssh terminal.

To avoid this `tmux` is very good option.

Terminal Multiplexer (`tmux`) to Keep SSH Sessions Running in the background after ssh terminal is closed:

1. Start the ssh terminal
2. Type `tmux` . It will open a window in the same terminal.
3. Give command to start Jupyter Notebook here. Open Notebook.

Now if SSH terminal is closed/terminated it will keep running your notebook on the instance.

If the connection terminated then:

4. reconnect or open new ssh terminal. To see this Jupyter Server(which is kept running in the background) type: `tmux attach` command.

(Edited: changed "notebook" to "Jupyter Server")

Want to terminate tmux session:

5. Close the notebook. Then type `exit` in tmux-terminal-window. (update: If we close the notebook and use `tmux detach` command: it will exit from tmux session window/terminal without terminating/stopping the tmux sessions)

For more details please refer to this article: <https://www.tecmint.com/keep-remote-ssh-sessions-running-after-disconnection/>



When I run the code `jupyter notebook notebook.ipynb` in the tmux terminal, it does give the link to run the notebook in the browser. But browser displays the following message: "This site can't be reached localhost refused to connect." – [rainman](#) Oct 12, 2018 at 23:31

try this: close tmux session (detach tmux), then restart the server and try once again in a new tmux terminal to run this command. let me know if you face the same problem once again. – [Yogesh Awdhut Gadade](#) Oct 14, 2018 at 13:27

- 1 [@Yogesh](#) From the answer what I understand is after reconnecting "tmux attach" will open the running notebook automatically. After attaching, I can see the notebook is running but it doesn't open on the browser. What am I missing? – [shaurov2253](#) Dec 1, 2020 at 21:55

Hi Shaurov, the command after running this command: "tmux attach" from the terminal it will show Jupyter Server is running. To see the running notebook you need to open the Jupyter notebook in the Browser. (I have edited the text from "Notebook" to "Jupyter Notebook". Thanks) – [Yogesh Awdhut Gadade](#) Dec 3, 2020 at 4:24

How does one see the output from this? Does it get logged each couple of minutes on the file? I can see that the file is still running but not sure where the outputs are – [Kurt](#) Apr 2, 2021 at 10:05



25



Under *nix, the best way to run a program avoiding to be terminated by closing the terminal is to use *nohup* (no Hang up).

To start browser after running the server use the command:

```
nohup jupyter notebook &
```

And to start the server without opening the browser use the command:

```
nohup jupyter notebook --no-browser &
```

Note that you can shut down the `jupyter` server by using Quit in the upper right of the page of `jupyter`.

`nohup` puts as a parent of the process `init(0)`, so it will not receive the "Hang Up" signal when the terminal is closed. All the output (standard output and standard error) are redirected to the file `nohup.out` `nohup` exists both as program and shell command, so if you have `bash`, check `man` page of `bash` to read more details.



- 1 Can confirm it works on Windows Subsystem for Linux (tested on Ubuntu 18.04 and Windows 10 1803) – [x__x](#) Aug 15, 2019 at 8:26

This is the real best answer for the *nix systems. It works great – [Nicholas Gooding Rios](#) Aug 5, 2021 at 2:16

Since I need the URL with port and token from this executed on remote server, I wrap this command like so: `nohup bash -c 'jupyter lab --no-browser' &> server.out &` and get the url to paste in my local machine's web



This works for me when running a jupyter notebook server in the background.

22

```
$> nohup jupyter notebook --allow-root > error.log &
```



Stop the nohup jupyter notebook is simple.



First, find the pid of jupyter:



```
$> ps -ef| grep jupyter
```

e.g output like:

```
root      11417   2897  2 16:00 pts/0  00:04:29 /path/to/jupyter-notebook
```

Then kill the process:

```
$> kill -9 11417
```

You can also simplify this by storing the pid with:

```
$> nohup jupyter notebook --allow-root > error.log & echo $!> pid.txt
```

i.e, you can stop the notebook with:

```
$> kill -9 $(cat pid.txt)
```

An alternative way to stop the jupyter notebook is quit from the notebook page.

Share Edit Follow

edited Mar 1, 2020 at 9:12

answered Dec 26, 2018 at 7:38



dan1st

10.7k 7 32 62



TizeeU0U

464 3 8

do not use `kill -9` by default, it's too harsh. – Adrian Oct 18, 2020 at 10:35

jupyter notebook has `list` and `stop` subcommands. If you've launched only one server you can just `jupyter notebook stop` instad of `ps aux | grep && kill`. It also stops notebook gracefully. – heyzling Feb 2 at 18:28 ✎

@Adrian what do you mean by "too harsh?" In my experience other kill commands don't always get the job done.
– Wassadamo May 20 at 19:30



You can use `screen` to run it.

7

```
screen -A -m -d -S anyscreenname jupyter notebook --no-browser
```



This will start jupyter in a screen and you can access screen using screen commands.



Share Edit Follow

answered Jan 31, 2019 at 6:30



redcode

71 1 1



Actually, jupyter notebook & alone is not enough, the backend will still log to your screen. What you need is, cited from this [issue](#)

5



```
jupyter notebook > /path/to/somefileforlogging 2>&1 &
```



Share Edit Follow

answered Apr 1, 2019 at 11:23



limtete

89 1 5



You can start up the notebook in a `screen` or `tmux` session. Makes it easy to check error messages, etc.

3



Share Edit Follow

answered Jun 25, 2018 at 19:40



Nick Crawford

5,036 2 21 20



For remote machines `jupyter notebook &` works fine. However, it does not work on local machines when you close the terminal. For local machines use `tmux`.

2



Share Edit Follow

answered Sep 25, 2019 at 3:51



GT GT

73 4



This doesn't work when I try executing on RHEL remote machine. The URL with token it returns to stout doesn't work.

– [Wassadamo](#) May 20 at 19:45



Not real sophisticated but it gets the job done:

2



```
#!/bin/bash
```

```
#probably should change this to a case switch
```



```
if [ "$1" == "end" ]
then
```

```
    echo
    echo "Shutting Down jupyter-notebook"
    killall jupyter-notebook
    echo
    exit 0
fi
```

```

if [ "$1" == "-h" ]
then
    echo
    echo "To start   : jnote <port> [default 8888]"
    echo "To end     : jnote end"
    echo "This help  : jnote -h"
    echo
    exit 0
fi

#cast from string
PORT=$(( $1 ))

RETURN=0
PID=0

if [ "$PORT" == "0" ] || [ "$PORT" == "" ]; then PORT=8888; fi

echo
echo "Starting jupyter-notebook"

#background and headless, set port, allow colab access, capture log, don't open browser yet

nohup jupyter notebook \

```

Share Edit Follow

edited Jan 3, 2020 at 0:34

answered Jan 2, 2020 at 23:51



DJ Swarm
21 3



jupyter notebook & >> disown

2



put the process into the background by using `jupyter notebook &`
then type `disown` OR `disown <the process id>`
you can close the terminal now



Share Edit Follow

edited Jun 16, 2020 at 16:28

answered Mar 1, 2020 at 16:02



desertnaut
55.3k 21 132 163



hossein hayati
980 2 14 29



Detach Jupyter process from the controlling terminal and send all its input and output data to `/dev/null` which is a special device file that writes-off any data written to it.

1



`jupyter notebook </dev/null &>/dev/null &`



Lazy people like me would prefer to edit `~/.bash_aliases` and create an alias:



`alias jnote='jupyter notebook </dev/null &>/dev/null &'`

Reference: <https://www.tecmint.com/run-linux-command-process-in-background-detach-process/>

Share Edit Follow

edited Jul 17, 2020 at 21:51

answered Jul 17, 2020 at 21:25



▲ If you are using iTerm2 software, first you need to set:

0

```
brew shellenv
```



Then start jupyter in nohup:



```
eval $(/usr/local/bin/brew shellenv)
```

```
nohup jupyter notebook &
```

Share Edit Follow

edited Mar 7 at 13:12



ouflak

2,438 10 43 49

answered Mar 7 at 10:23



saikumardaram

1 4



0



Share Edit Follow

As suggested by one of the users, using `jupyter notebook &` solves the issue. Regarding the comments stating that it kills the kernel after closing the terminal, probably you are using `jupyter-notebook &`.



answered Feb 3 at 10:04



schaftler

1