

A Guide to the terminal multiplexer tmux

Ina Krapp
SAFE Research Datacenter*

Last Update: October 15, 2024

tmux¹ is a terminal multiplexer. It lets you switch between several programs in one terminal, detach them (they keep running in the background), and reattach them to a different terminal. This guide shows how to run code on the SAFE Research Infrastructure using tmux.

Contents

1	Contact	1
2	Prerequisite	2
3	Create a new tmux session - Visual Studio Code	2
4	Run Code in a tmux session - Visual Studio Code	3
5	Detach and attach a tmux session - Visual Studio Code	4
6	End a tmux session - Visual Studio Code	6
7	Working with tmux - Windows Powershell	8

1 Contact

If you encounter any difficulties or just want general information, do not hesitate to contact us.

SAFE Research Datacenter: datacenter@safe-frankfurt.de

More information about the SAFE Research Datacenter, and further guides can be found [here](#), and [here](#).

*krapp@safe-frankfurt.de

¹<https://github.com/tmux/tmux/wiki>

2 Prerequisite

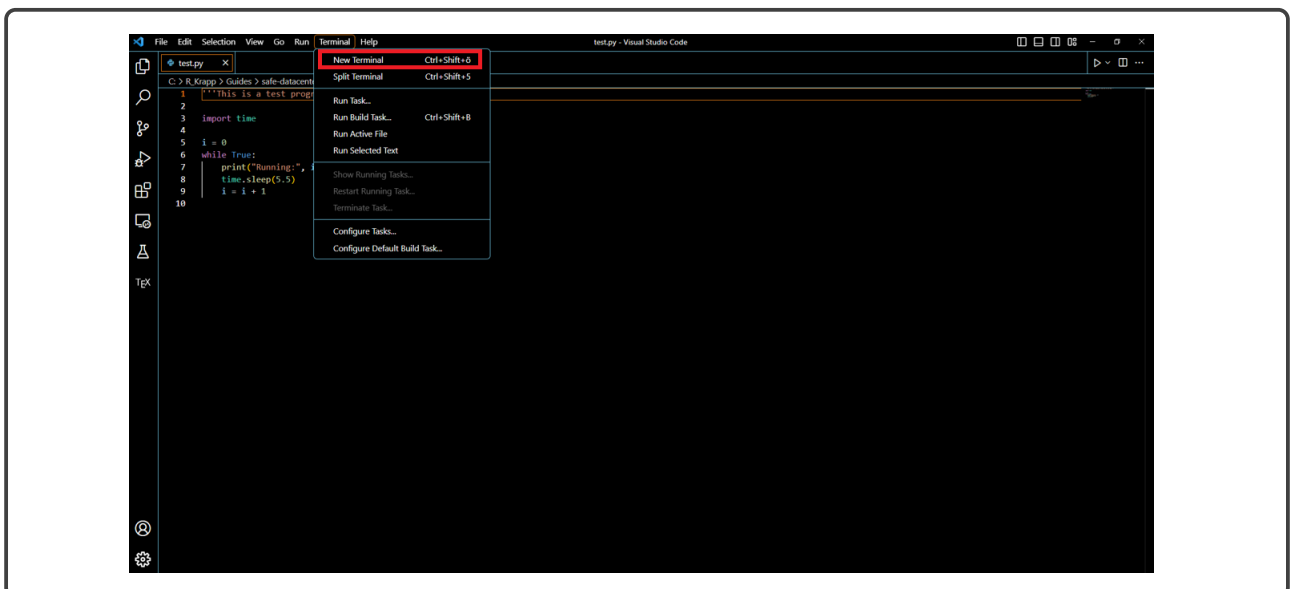
For security reasons, a connection to the SAFE Research Infrastructure can only be established from within the network of Goethe University. You must use Cisco Anyconnect VPN from outside the Goethe University network to connect to the SAFE Research Infrastructure. Instructions on how to set up the VPN connection can be found [here](#). For VPN authentication, please use your HRZ credentials. This instruction primarily uses Visual Studio Code. If you wish to use it but have not installed it yet, you can install it from [here](#):

- Installation of [Visual Studio Code](#) (VSCoDe)

Alternatively, it is possible to run tmux using the Windows Powershell, which is pre-installed on all Windows computers.

3 Create a new tmux session - Visual Studio Code

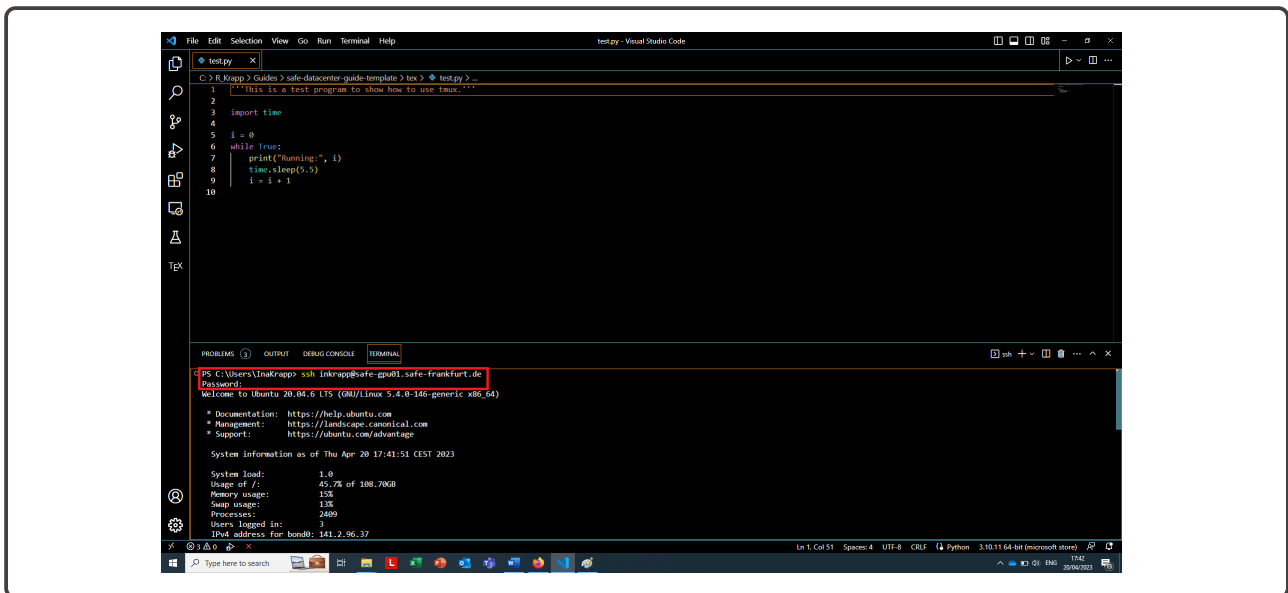
Before opening a new tmux session in Visual Studio Code, you need to open a terminal. This can be done by clicking on [Terminal](#) and selecting 'New Terminal'.



Then, if not already done, you need to establish a ssh connection. This is done using your HRZ credentials. So when using the code below, replace hrz-username with the hrz username assigned to you through SAFE (not your student username!)

```
1 ssh hrz-username@safe-gpu01.safe-frankfurt.de
```

When you executed the code, you will be asked for a password. Use your hrz password to log in. The result should look (with another username) close to this:



In the next step, open a tmux session. A tmux session can be given any name. In this example, a tmux session with the name session1 is created. To open a tmux session with this name, the code is:

```
1 tmux new -s session1
```

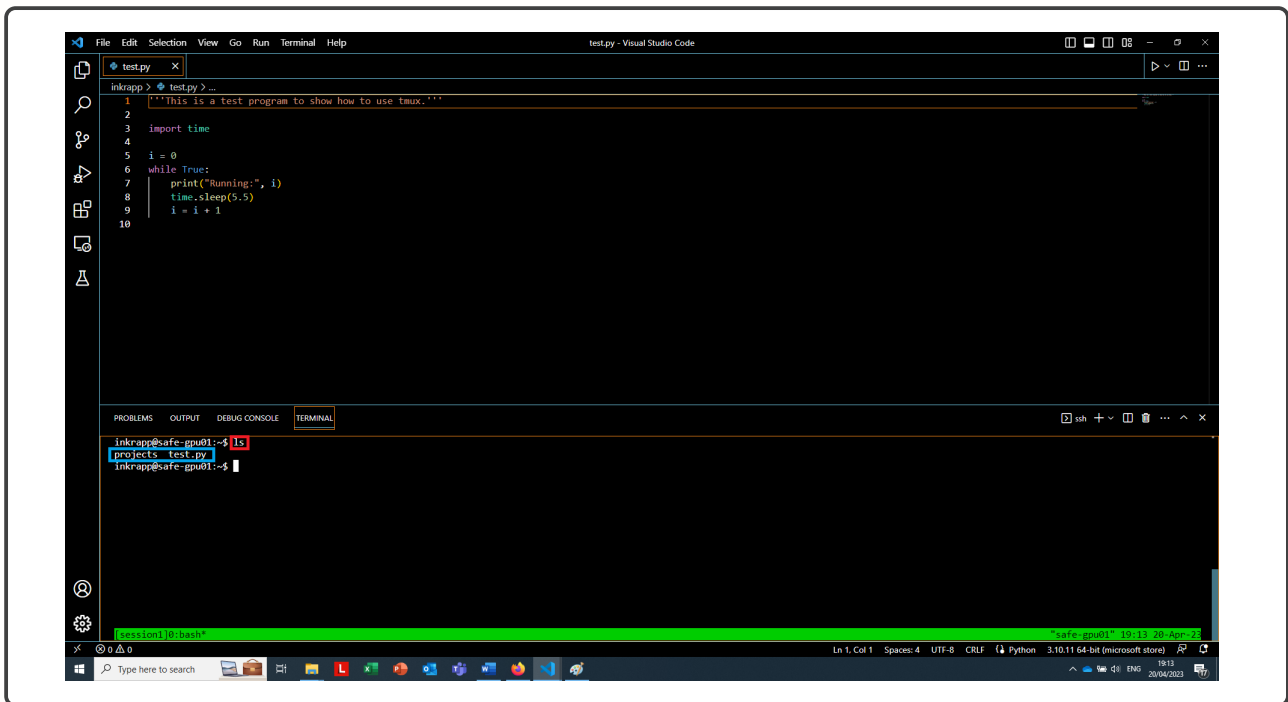
For other sessions, session1 in the code can be replaced by other session names. This allows the user to run multiple sessions at once.

4 Run Code in a tmux session - Visual Studio Code

Once a tmux session has been opened, code can be run in it. For demonstration purposes, a file test.py will be run here. It prints a number every 5.5 seconds. Jupyter lab and other development environments are often not supported by tmux. Code should be in basic formats such as .py to run it in tmux. Before running the code, make sure the correct working directory is provided in the terminal. You can check if it is correct by listing all files available in the directory. The file you want to execute should be among the listened files. The code for this is:

```
1 ls
```

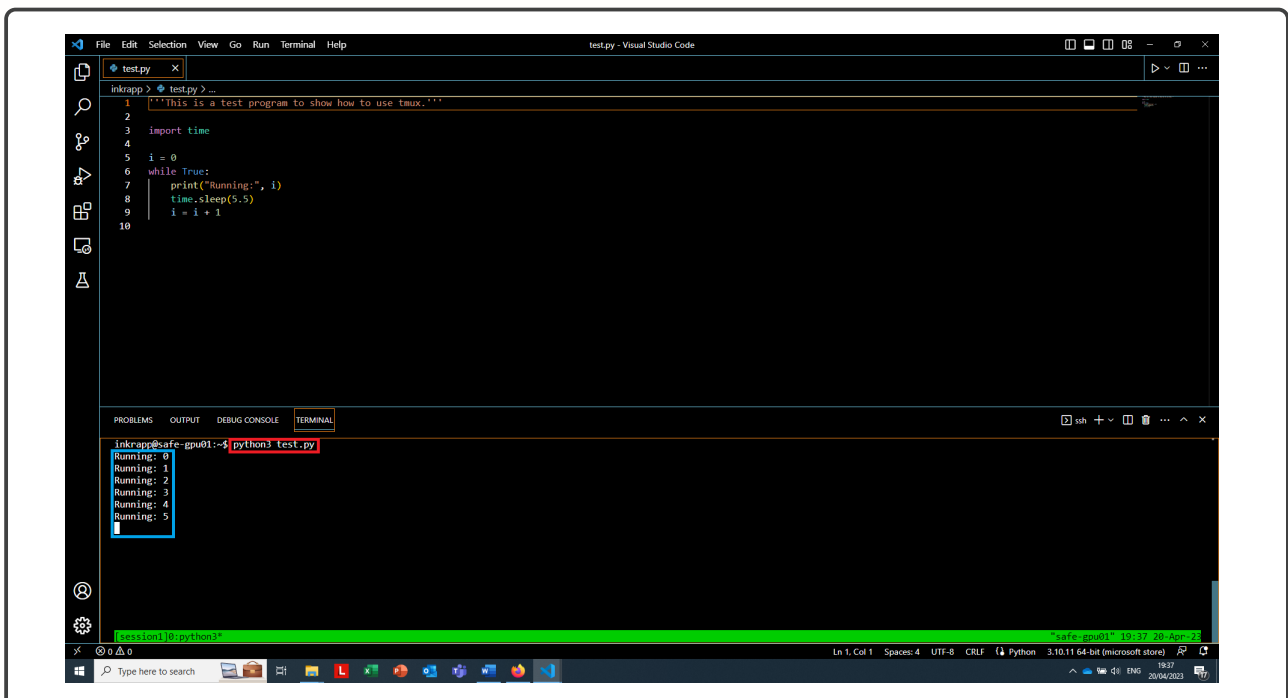
When the code is executed, it lists files and folders. In the example below, test.py is the program I want to execute. I can execute it now, since it is in my current working directory.



To run code, the basic syntax is usually the program name you want to execute, followed by the codefile. So, for python and the test.py-file, it is:

```
1 python3 test.py
```

The result should look like this, with the results of the code that get printed being depicted below the command to execute the code.



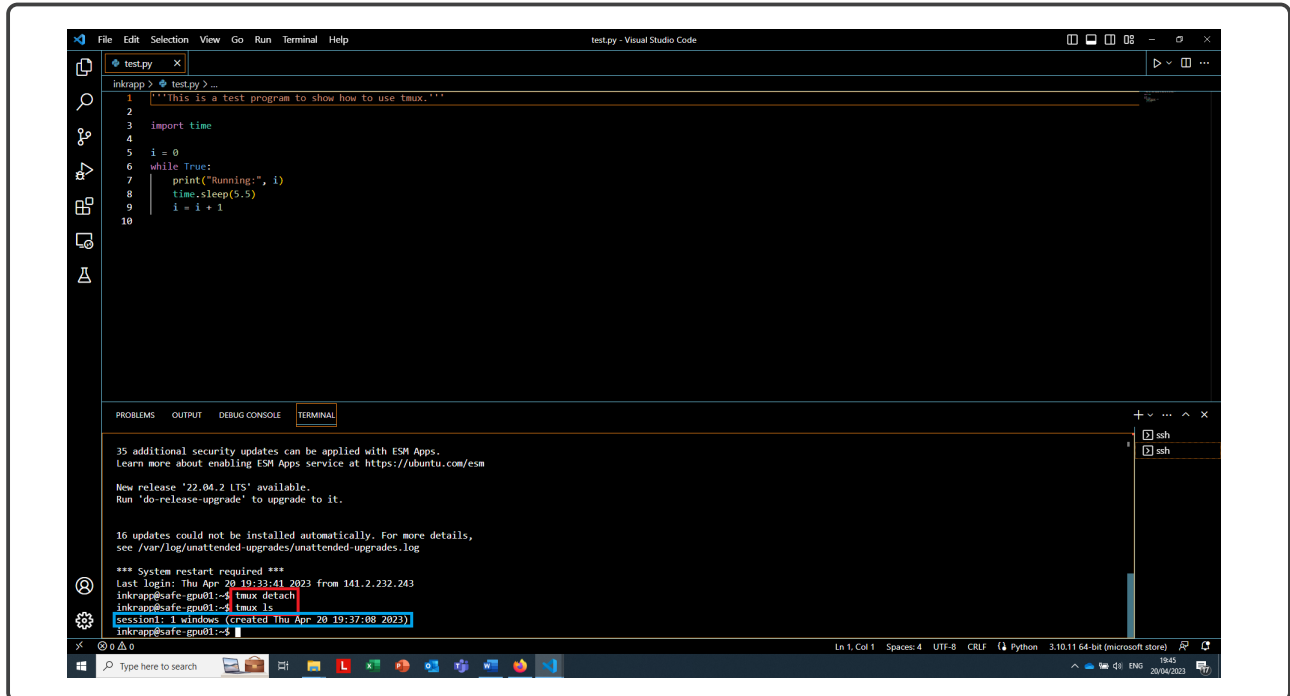
5 Detach and attach a tmux session - Visual Studio Code

As mentioned above, tmux sessions can be detached. This is useful because detached sessions continue to run regardless of if you switch off your computer, lose your internet connection or

connection to the server. To detach a tmux session, open a second terminal and log in again to the ssh if necessary. The command to detach a tmux session is:

```
1 tmux detach
```

You can use the `tmux ls` command to list all tmux sessions. Then, you see which tmux sessions you have currently running and if any of them are attached.

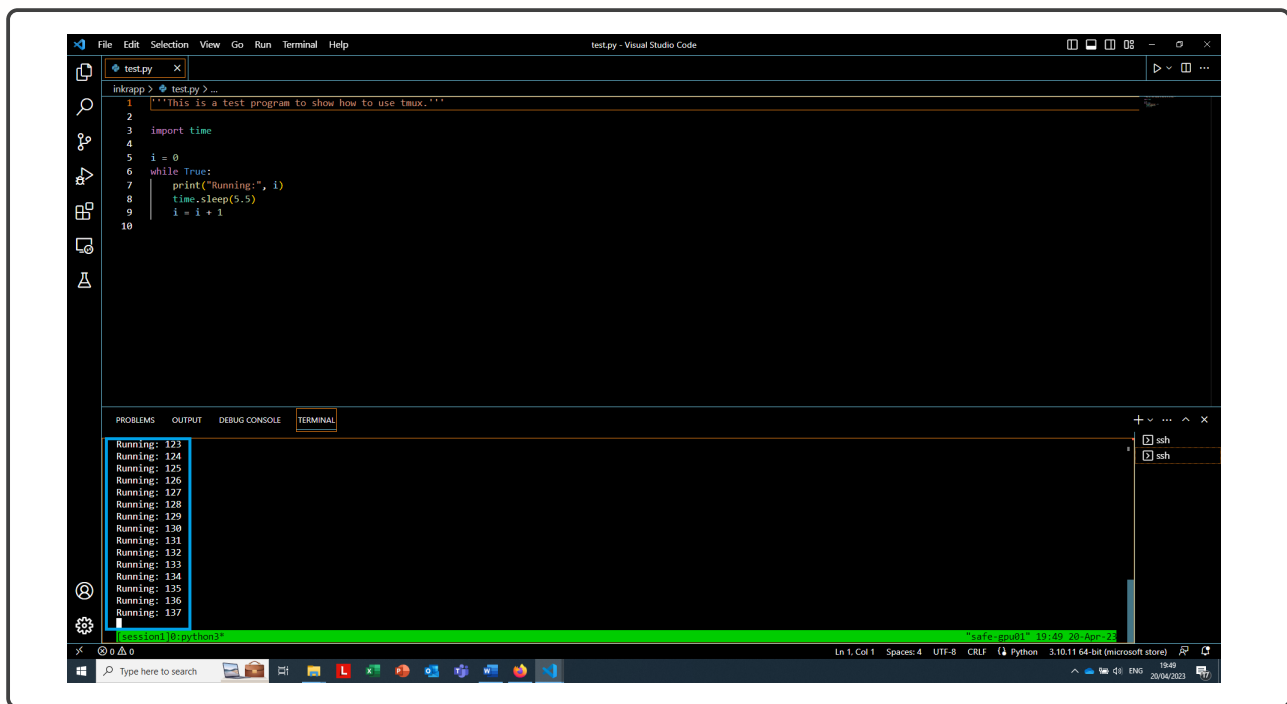


The screenshot shows a Visual Studio Code editor window with a Python file named `test.py`. The code in the file is a simple loop that prints "Running:" and sleeps for 5.5 seconds. Below the editor, the TERMINAL panel is open, showing the output of the script. The output includes system updates and a message indicating that a system restart is required. The terminal also shows the user's login history and the command `tmux ls` being executed, which lists the current tmux session as `session1`.

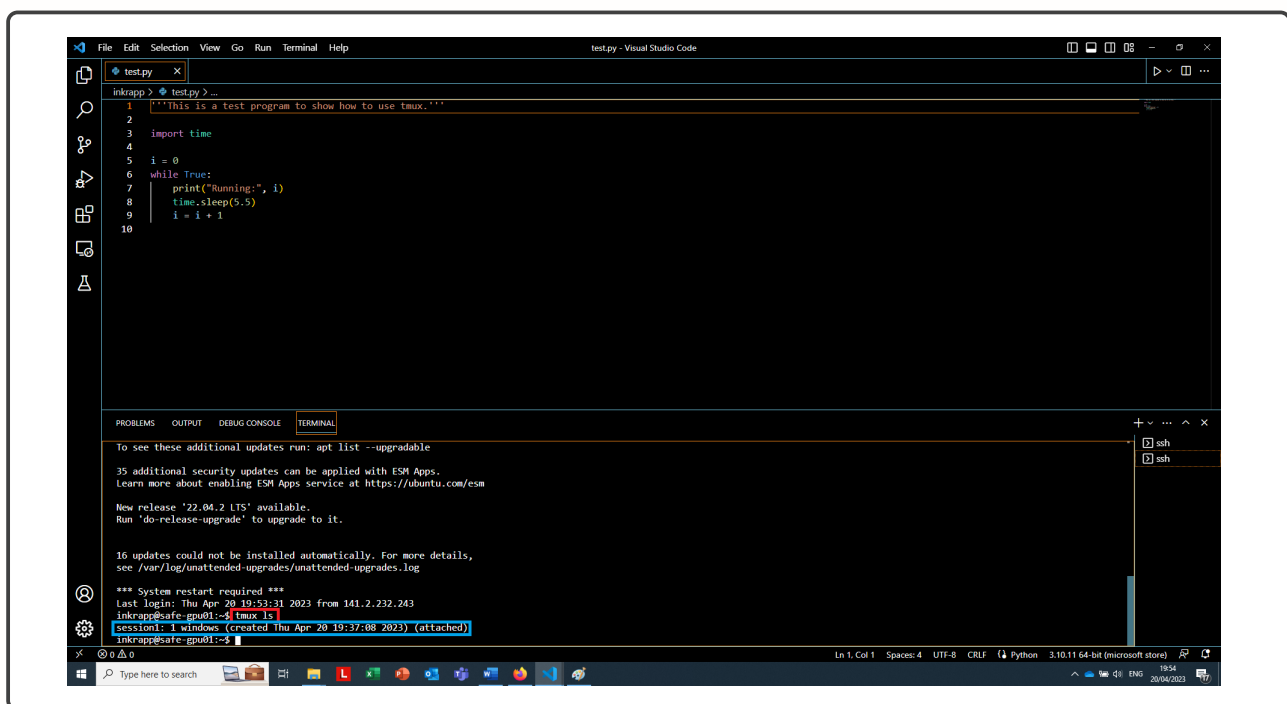
If you want to check, for example, after having detached a session in the previous evening, if the code has finished, you can reattach the tmux session. For this, you need to name the session again. So, in this example, where the session is named `session1`, the command is:

```
1 tmux attach -t session1
```

After you typed this in, you are back in the tmux session and can obtain the results your code printed out so far. In the test case:



If you run `tmux ls` now in a second terminal, you'll see that the session is labeled as attached.

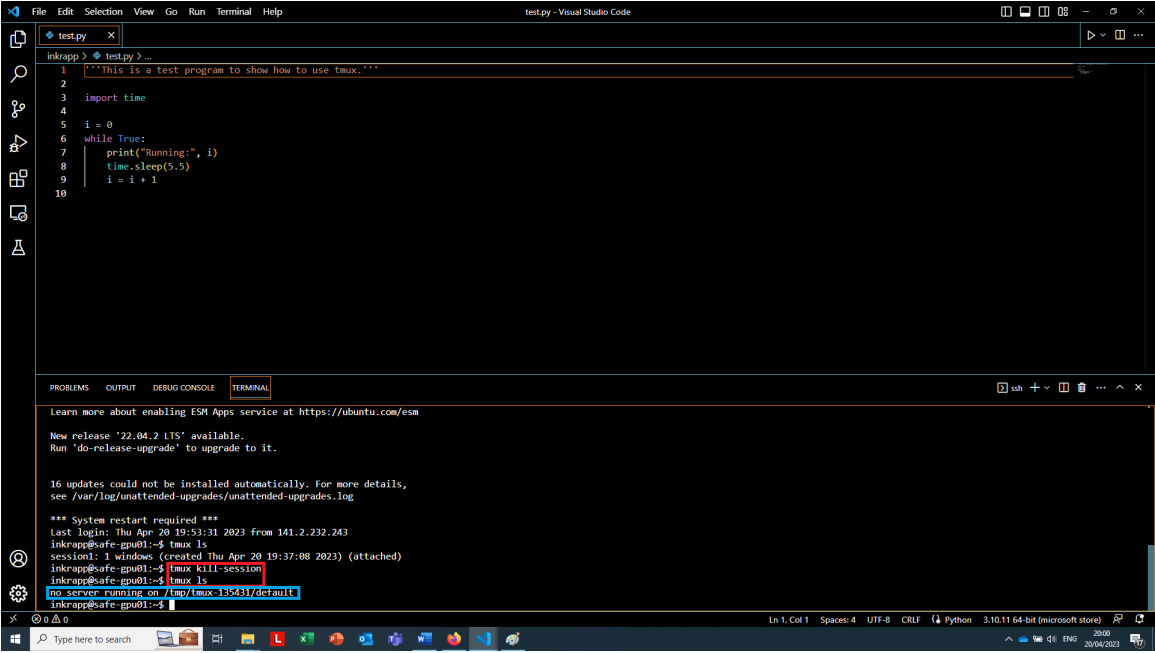


6 End a tmux session - Visual Studio Code

Since `test.py` contains an endless loop, it will not terminate itself at a given point in time. To terminate it, a command can be used, running it, again, in the second terminal. The command to end a tmux session even when the code is still running is:

```
1 tmux kill-session
```

By default, it ends the last used tmux session. -t session-name can be supplied to end a specific session. kill-server ends all tmux sessions. If you run tmux ls afterwards again, you can see no tmux session is active anymore:



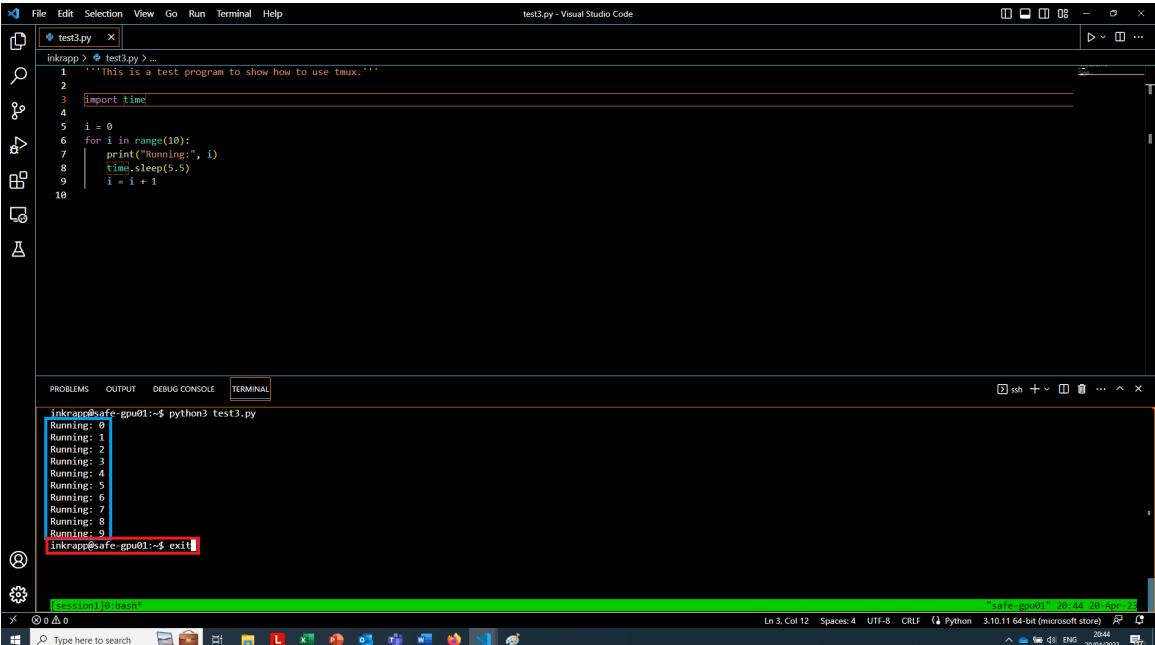
```
inkrapp> test.py > ...
1 '''This is a test program to show how to use tmux.'''
2
3 import time
4
5 i = 0
6 while True:
7     print("Running:", i)
8     time.sleep(5.5)
9     i = i + 1
10

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Learn more about enabling ESM Apps service at https://ubuntu.com/esm
New release '22.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

16 updates could not be installed automatically. For more details,
see /var/log/unattended-upgrades/unattended-upgrades.log

*** System restart required ***
Last login: Thu Apr 20 19:53:31 2023 from 141.2.232.243
inkrapp@safe-gpu01:~$ tmux ls
session1: 1 windows (created Thu Apr 20 19:37:08 2023) (attached)
inkrapp@safe-gpu01:~$ tmux kill-session
inkrapp@safe-gpu01:~$ tmux ls
no server running on /tmp/tmux-1392431/default
inkrapp@safe-gpu01:~$
```

If the code you ran in your tmux session has finished, the finished session looks like this:



```
inkrapp> test3.py > ...
1 '''This is a test program to show how to use tmux.'''
2
3 import time
4
5 i = 0
6 for i in range(10):
7     print("Running:", i)
8     time.sleep(5.5)
9     i = i + 1
10

inkrapp@safe-gpu01:~$ python3 test3.py
Running: 0
Running: 1
Running: 2
Running: 3
Running: 4
Running: 5
Running: 6
Running: 7
Running: 8
Running: 9
inkrapp@safe-gpu01:~$ exit

[session] 10: bash$
```

You can continue working with the same session by commanding to run another program with the same syntax used above to start the first program. Alternatively, you can simply exit the tmux session by typing exit, as it is shown above.

7 Working with tmux - Windows Powershell

tmux sessions can also be run in the Windows Powershell. The commands are exactly the same like in the Terminal that is integrated into Visual Studio Code. Here is an example how opening a tmux session in the Powershell can look like. The arrows point at the code entered by the user: Connecting to the ssh, looking at code in the current directory and opening a tmux session.

```
OpenSSH SSH client
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\InaKrapp> ssh inkraapp@safe-gpu01.safe-frankfurt.de
Password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-146-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Thu Apr 20 20:52:14 CEST 2023

System load:        1.08
Usage of /:          45.7% of 108.70GB
Memory usage:       19%
Swap usage:         13%
Processes:          2647
Users logged in:    4
IPv4 address for bond0: 141.2.96.37
IPv6 address for bond0: 2001:a38:1001:962:141:2:96:37

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

 * Introducing Expanded Security Maintenance for Applications.
   Receive updates to over 25,000 software packages with your
   Ubuntu Pro subscription. Free for personal use.

   https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

3 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

35 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/

New release '22.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

16 updates could not be installed automatically. For more details,
see /var/log/unattended-upgrades/unattended-upgrades.log

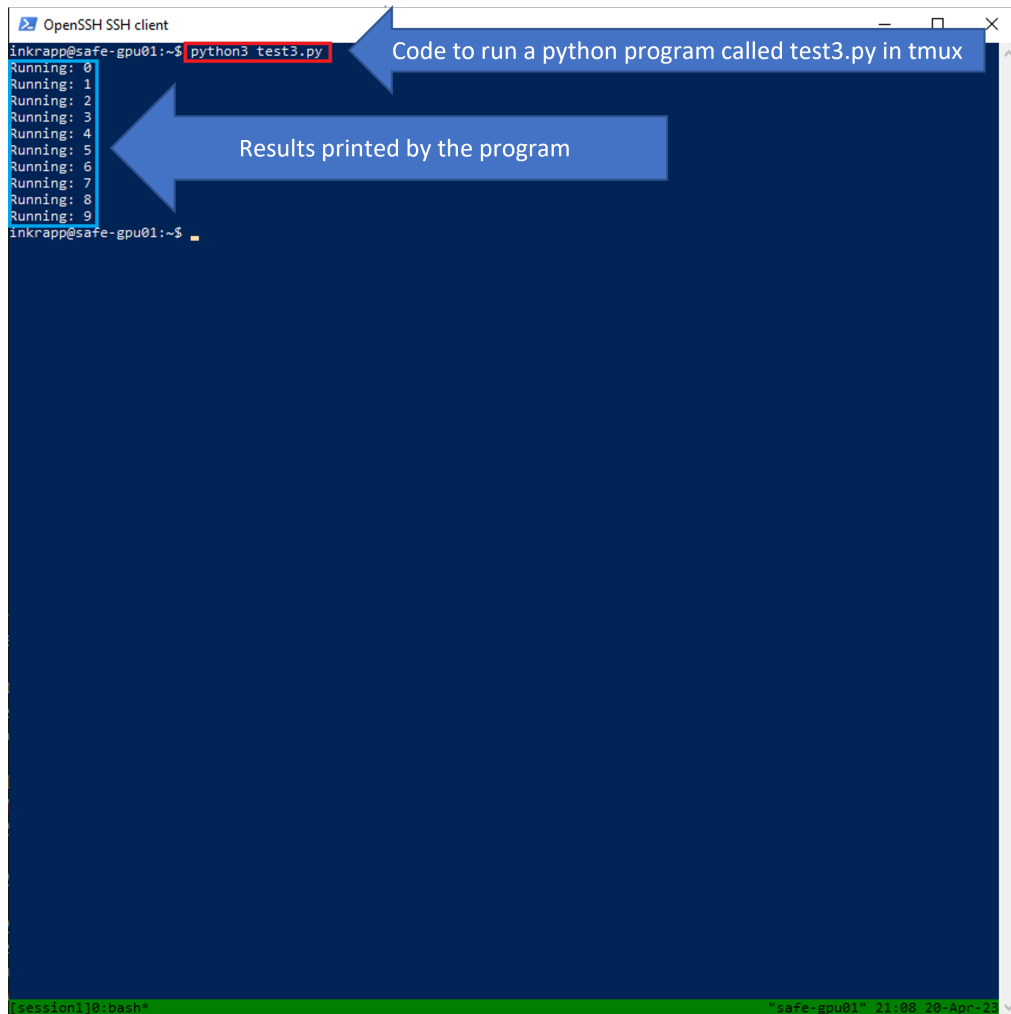
*** System restart required ***
Last login: Thu Apr 20 20:51:17 from 141.2.232.243
inkraapp@safe-gpu01:~$ ls
projects_test.py test3.py
inkraapp@safe-gpu01:~$ tmux new -s session1
```

Code to establish the ssh connection

Code to show everything in the working directory

Code to open a new tmux session

And here is an example how a tmux session which just ran some code can look like:



```
OpenSSH SSH client
inkrapp@safe-gpu01:~$ python3 test3.py
Running: 0
Running: 1
Running: 2
Running: 3
Running: 4
Running: 5
Running: 6
Running: 7
Running: 8
Running: 9
inkrapp@safe-gpu01:~$
```

Code to run a python program called test3.py in tmux

Results printed by the program

[session1]0:bash* "safe-gpu01" 21:08 20-Apr-23

>>

Like before, the user could now start the next program or close the session with 'exit'. Finally, like in Visual Studio Code, a tmux session can be attached, detached and ended using commands in a second terminal.

```
OpenSSH SSH client
System information as of Thu Apr 20 21:16:57 CEST 2023
System load:          2.2
Usage of /:           45.7% of 108.70GB
Memory usage:         19%
Swap usage:           13%
Processes:            2651
Users logged in:       4
IPv4 address for bond0: 141.2.96.37
IPv6 address for bond0: 2001:a38:1001:962:141:2:96:37

* Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
  just raised the bar for easy, resilient and secure K8s cluster deployment.

  https://ubuntu.com/engage/secure-kubernetes-at-the-edge

* Introducing Expanded Security Maintenance for Applications.
  Receive updates to over 25,000 software packages with your
  Ubuntu Pro subscription. Free for personal use.

  https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

3 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

35 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

New release '22.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

16 updates could not be installed automatically. For more details,
see /var/log/unattended-upgrades/unattended-upgrades.log

*** System restart required ***
Last login: Thu Apr 20 21:14:50 2023 from 141.2.232.243
inkrapp@safe-gpu01:~$ tmux ls
session1: 1 windows (created Thu Apr 20 21:16:06 2023) (attached)
session2: 1 windows (created Thu Apr 20 21:17:22 2023) (attached)
inkrapp@safe-gpu01:~$ tmux detach
inkrapp@safe-gpu01:~$ tmux ls
session1: 1 windows (created Thu Apr 20 21:16:06 2023) (attached)
session2: 1 windows (created Thu Apr 20 21:17:22 2023)
inkrapp@safe-gpu01:~$ tmux kill-session
inkrapp@safe-gpu01:~$ tmux ls
session1: 1 windows (created Thu Apr 20 21:16:06 2023) (attached)
inkrapp@safe-gpu01:~$
```

Note that by default, commands like detach, attach and kill-session are applied to the most recent session the user worked with. If the user has several different sessions open and would like to apply any of these commands, the command has to be followed by -t session-name. For example, the code to reattach session1 when the user most recently worked with session2 is:

```
1 tmux attach -t session1
```

Without the last part, the command would be valid, but would refer to session2 instead of session1.

In the Windows Powershell, alternatively to the detach command, the current tmux session can also be detached by a keyboard shortcut: "CTRL+b" and then "d".

I hope this guide was helpful. For any more questions, feel free to contact the data center: datacenter@safe-frankfurt.de