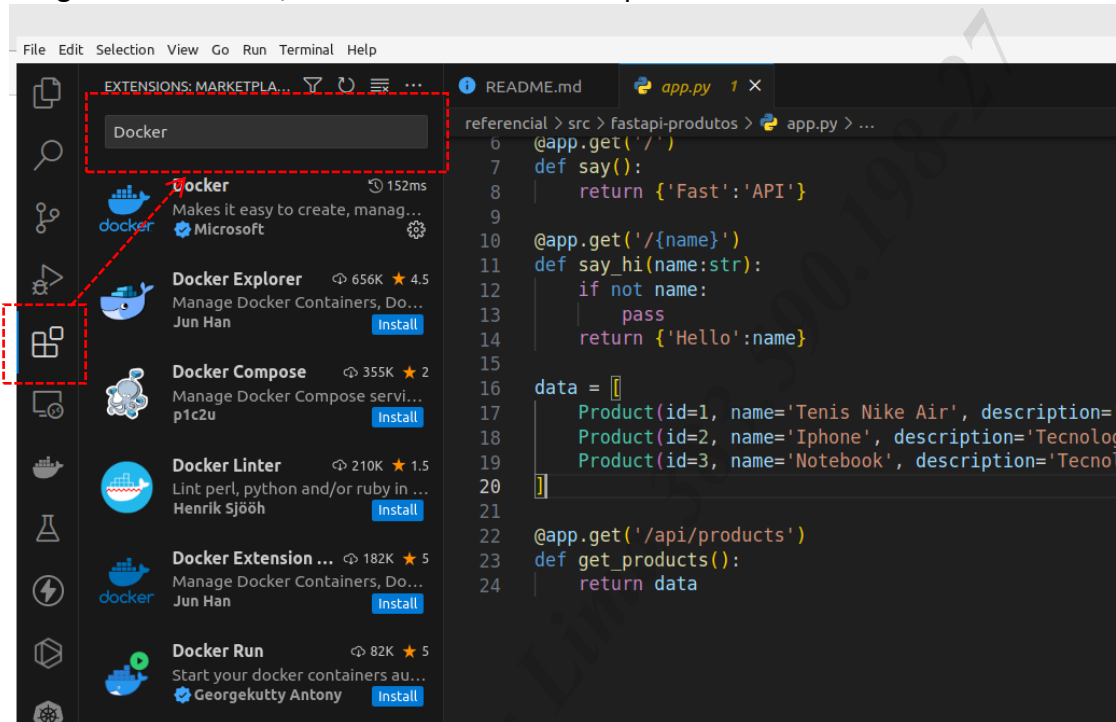


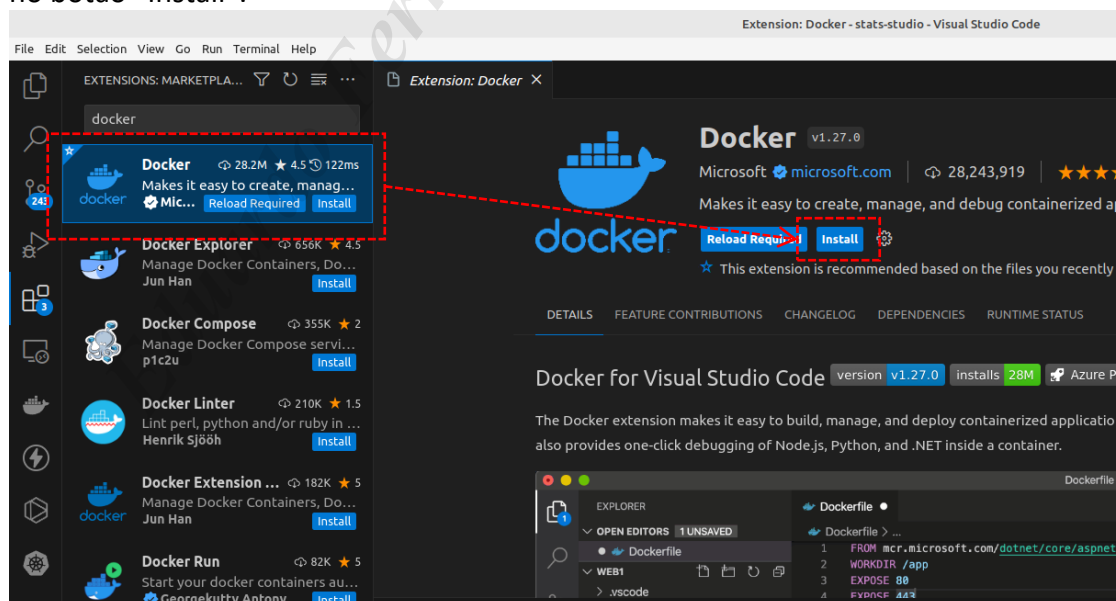
Prof. Guilherme Bezerra de Lima

Tutorial extensão Docker do VS Code e acesso ao terminal do container

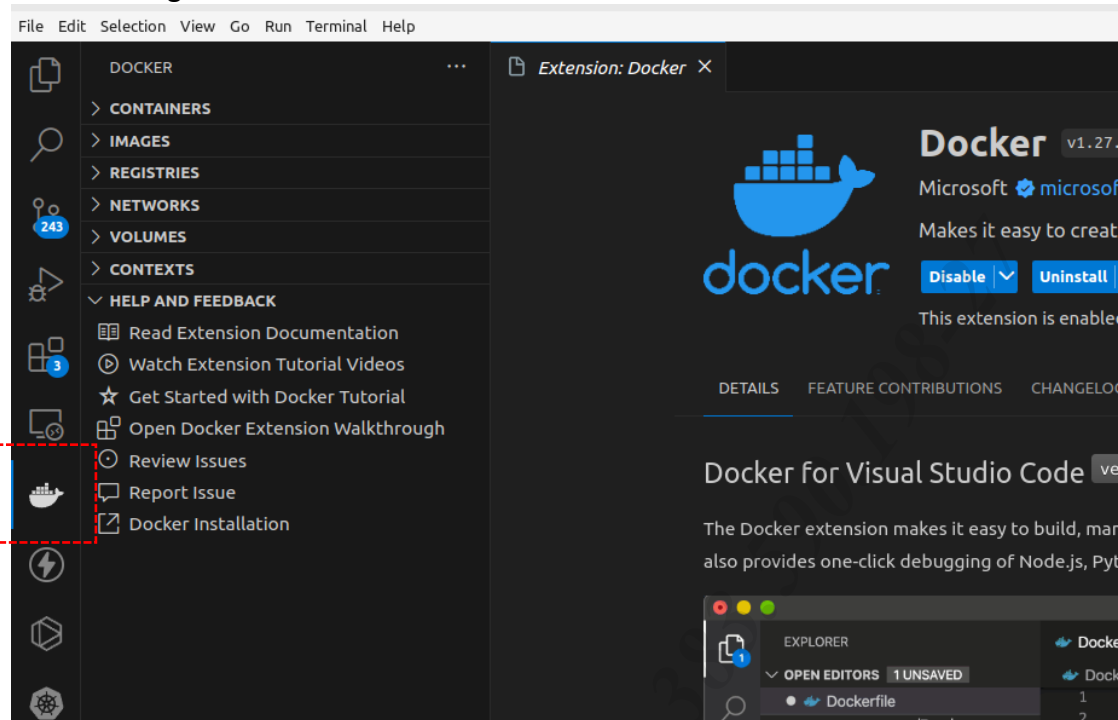
1. Acessar a aba extensões do VS Code clicando no botão sinalizado na figura abaixo e digitando “Docker”, como está sinalizado no quadrado em vermelho.



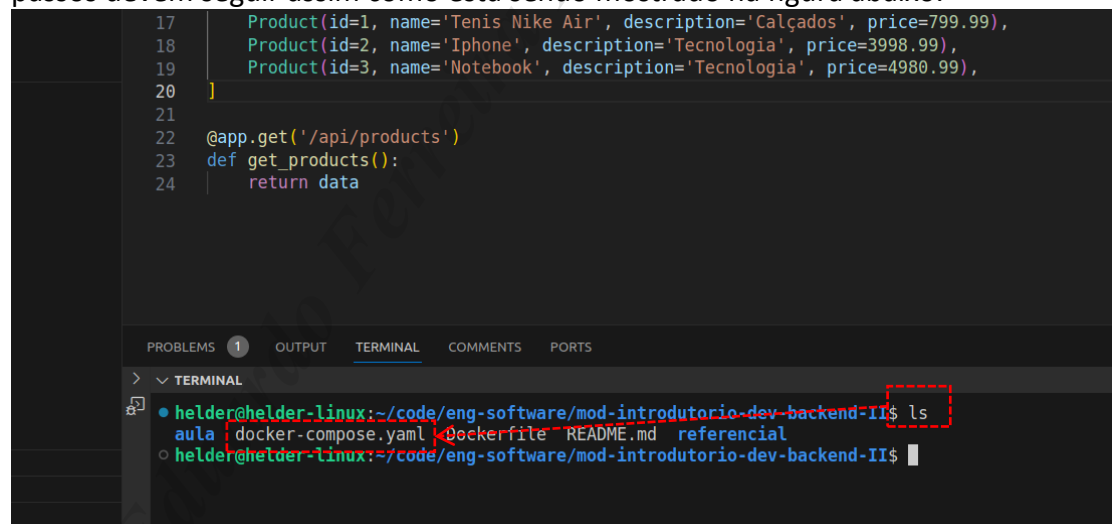
2. A extensão do Docker vai aparecer na primeira posição e os cliques devem ser feitos como aparecem na figura, primeiro na janela da extensão do Docker e depois no botão “Install”.



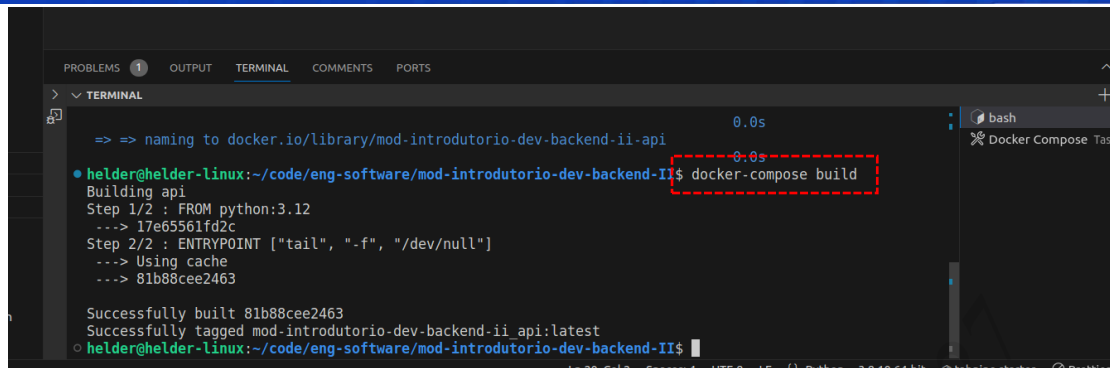
3. O ícone do Docker vai aparecer como uma nova aba do seu VS Code, assim como mostra na figura abaixo.



4. Nesse momento é importante identificar se você está no caminho correto ou não. No terminal do VS Code, digite o comando `ls` e aperte enter, se aparecer um arquivo chamado `docker-compose.yaml`, quer dizer que você está no caminho correto. Os passos devem seguir assim como está sendo mostrado na figura abaixo.



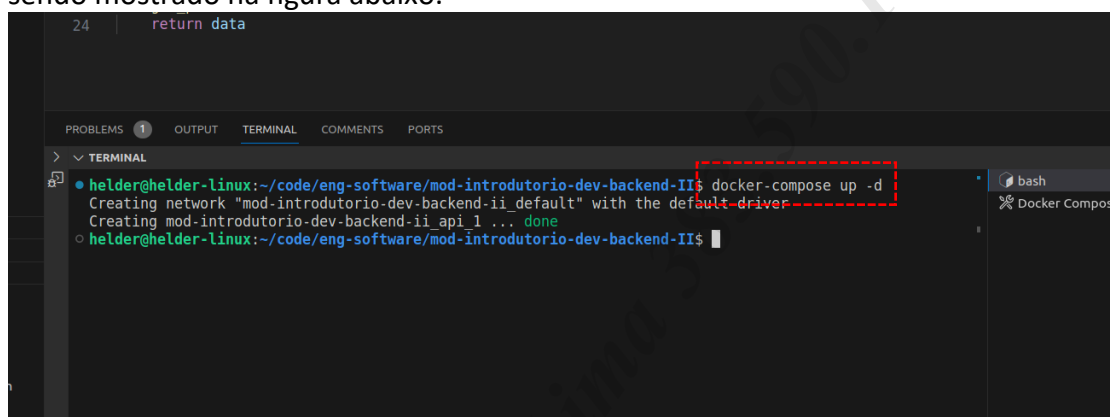
5. Para criar o seu ambiente de desenvolvimento é necessário gerar as imagens do container com o comando `docker-compose build`, assim como mostra a figura abaixo.



```
=> => naming to docker.io/library/mod-introdutorio-dev-backend-ii-api
0.0s
• helder@helder-linux:~/code/eng-software/mod-introdutorio-dev-backend-ii$ docker-compose build
Building api
Step 1/2 : FROM python:3.12
--> 17e65561fd2c
Step 2/2 : ENTRYPOINT ["tail", "-f", "/dev/null"]
--> Using cache
--> 81b88cee2463

Successfully built 81b88cee2463
Successfully tagged mod-introdutorio-dev-backend-ii_api:latest
o helder@helder-linux:~/code/eng-software/mod-introdutorio-dev-backend-ii$
```

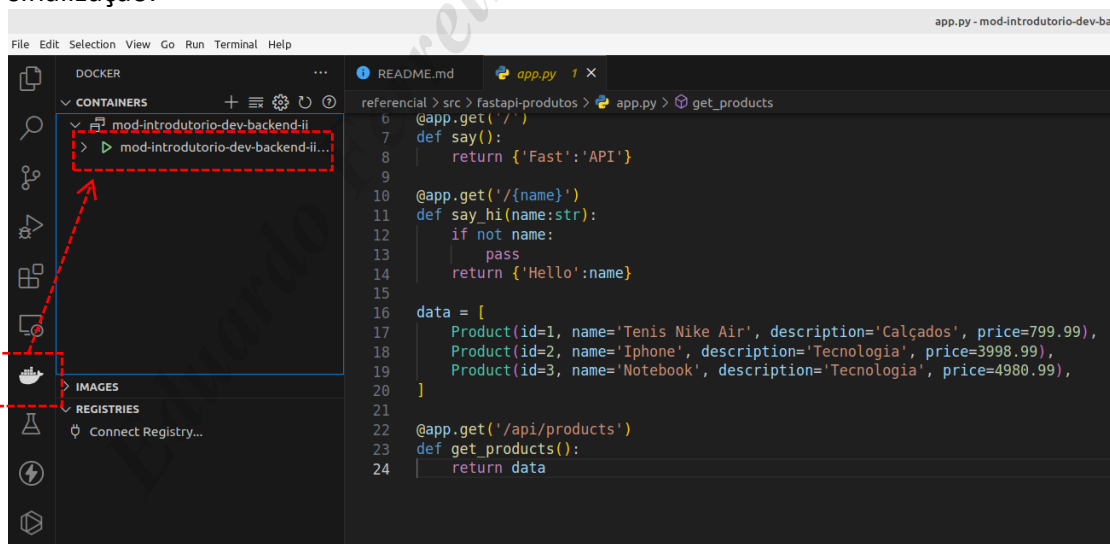
6. Agora que as imagens dos containers já estão geradas, é possível subir e acessar os containers utilizando o com o comando `docker-compose up -d`, assim como está sendo mostrado na figura abaixo.



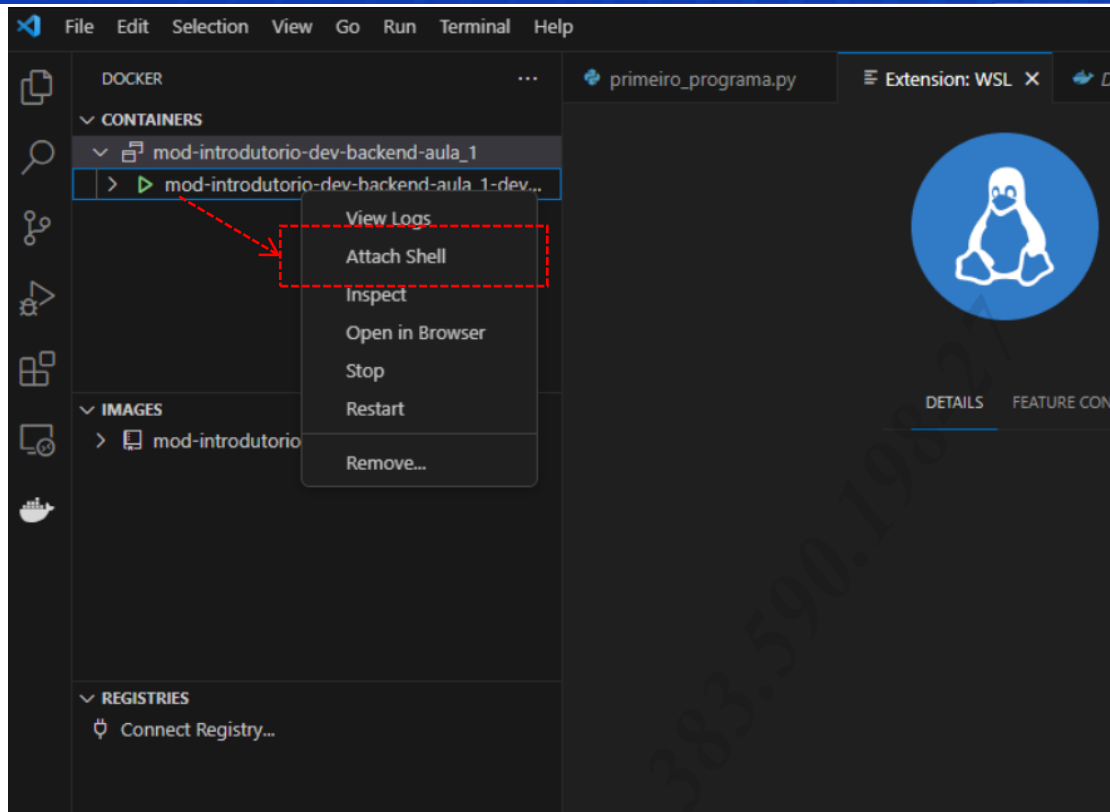
```
24 | return data

• helder@helder-linux:~/code/eng-software/mod-introdutorio-dev-backend-ii$ docker-compose up -d
Creating network "mod-introdutorio-dev-backend-ii_default" with the default driver
Creating mod-introdutorio-dev-backend-ii_api_1 ... done
o helder@helder-linux:~/code/eng-software/mod-introdutorio-dev-backend-ii$
```

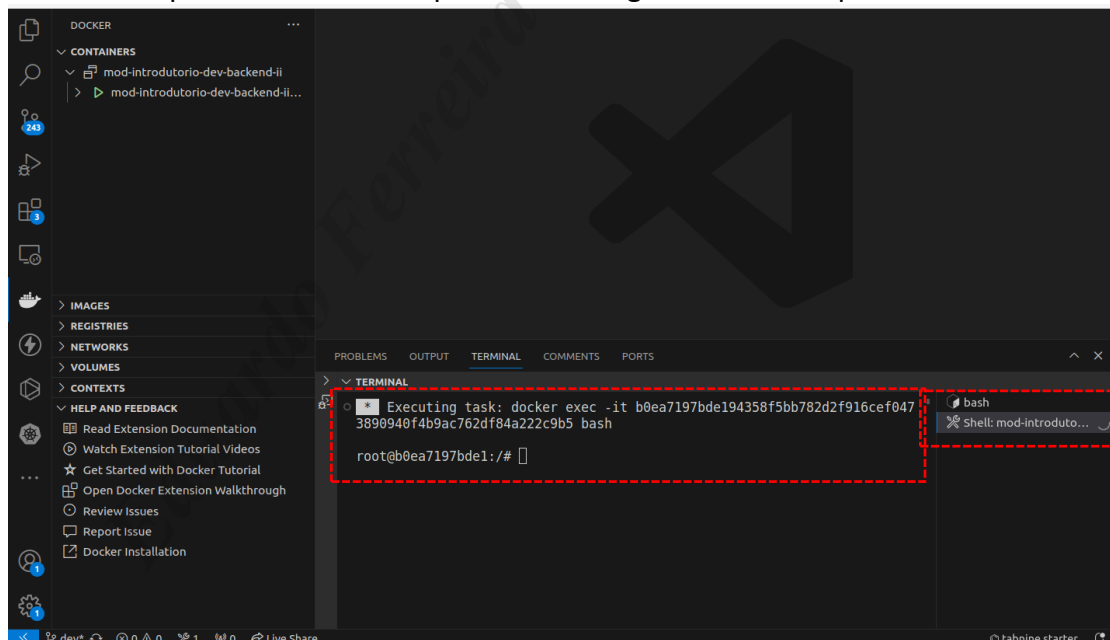
7. Depois que os containers forem criados, eles irão aparecer na aba da extensão do Docker no VS Code, assim como mostra a figura e os quadrados vermelhos de sinalização.



8. Para acessar o terminal do container é só clicar com o botão direito do mouse e selecionar a opção `Attach Shell` que vai aparecer através de uma janela abaixo do containers. A imagem abaixo mostra o que deverá aparecer nesse momento.

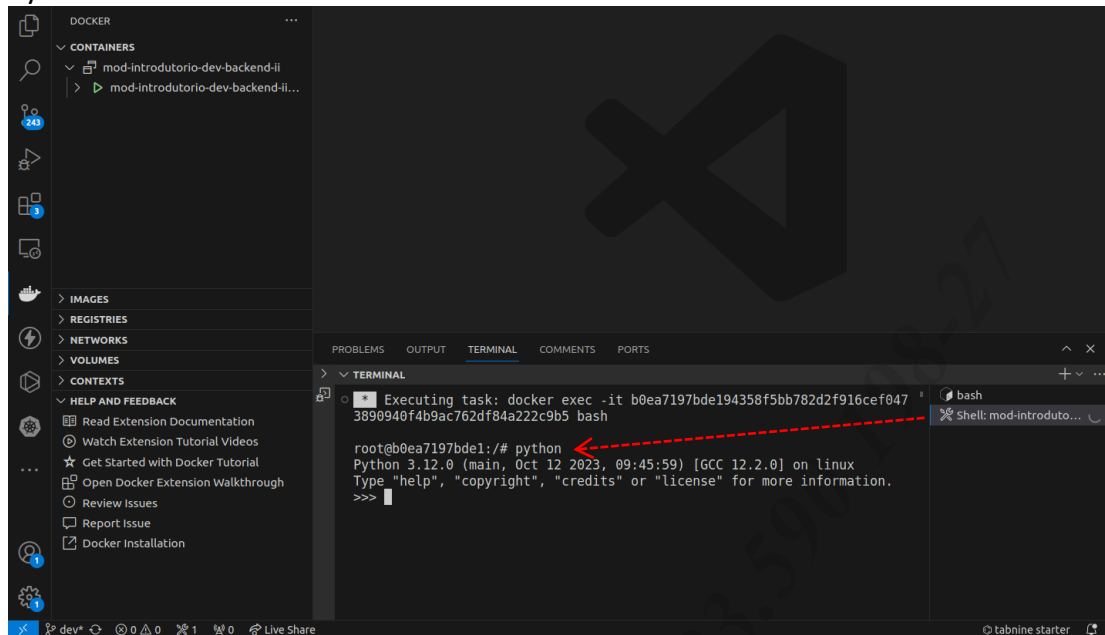


9. Após essa operação, um novo terminal irá aparecer na parte inferior do VS Code e o nome do terminal vai ser diferente do que é utilizado pelo seu usuário, isso quer dizer que você está acessando o terminal do container. Os quadrados vermelhos sinalizam o que deve acontecer quando você seguir os mesmos passos.



10. Para identificar se tudo foi feito da forma correta, você pode utilizar a função de alguma coisa que foi instalada apenas no container. A imagem abaixo mostra o exemplo do acesso do console da linguagem Python através do terminal do

container com o comando python, isso só funcionou porque o container tinha o Python instalado.



The screenshot shows the Docker extension interface in VS Code. The left sidebar lists various Docker-related actions under 'HELP AND FEEDBACK'. The main area displays the 'TERMINAL' tab, which shows the execution of a task: `docker exec -it b0ea7197bde194358f5bb782d2f916cef047 bash`. The terminal output shows the prompt `root@b0ea7197bde1:/#` and the command `python` being executed. The output of the command is: `Python 3.12.0 (main, Oct 12 2023, 09:45:59) [GCC 12.2.0] on linux`. A red dashed arrow points from the `python` command in the terminal to the output text. The status bar at the bottom indicates the active workspace is `dev*` and the terminal is using the `bash` shell.

```
root@b0ea7197bde1:/# python
Python 3.12.0 (main, Oct 12 2023, 09:45:59) [GCC 12.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```