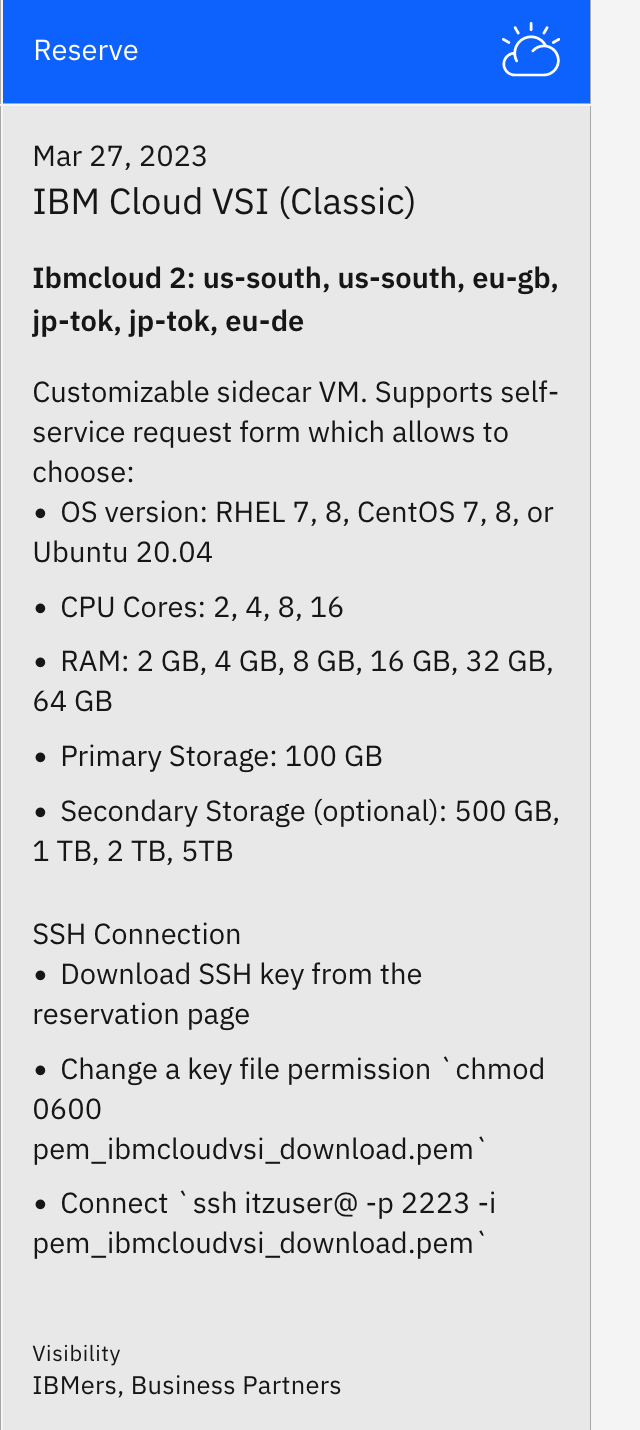
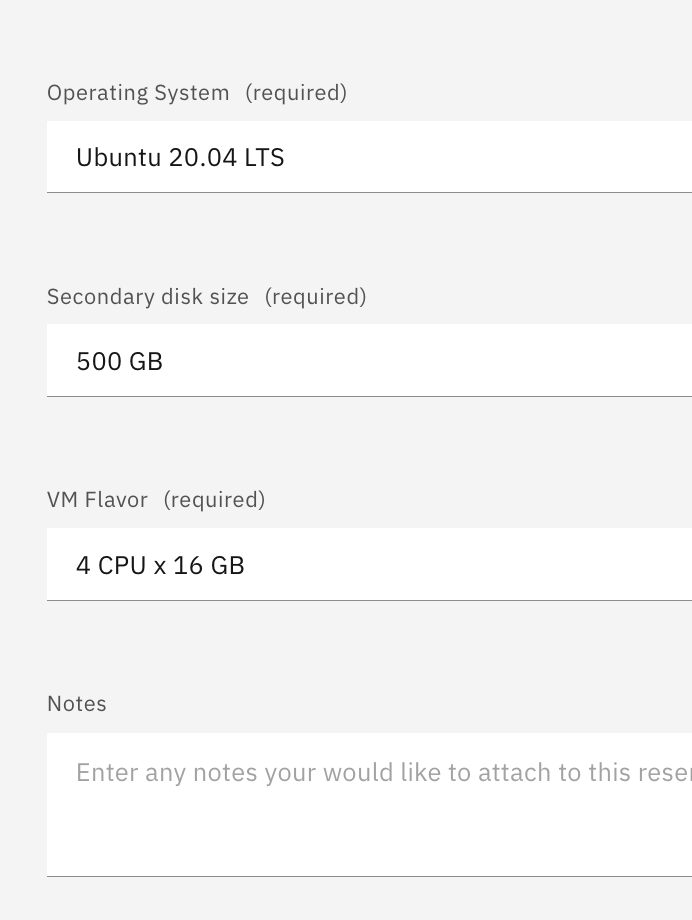
# Setting up a lab client with Ubuntu 22.10

## 1. Reserve a system

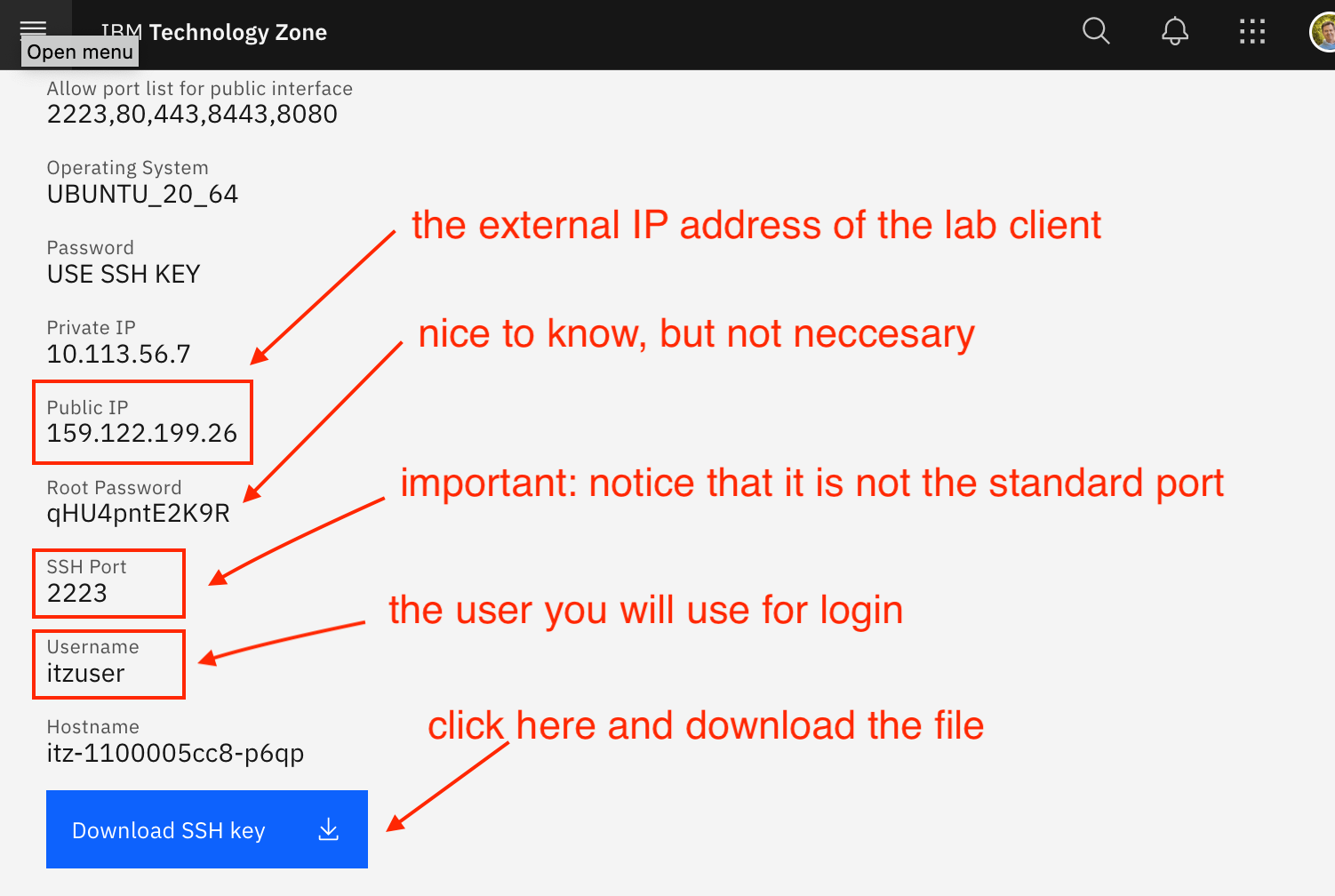
Access the [IBM Techzone](https://techzone.ibm.com/) and reserve a small linux server. This is the recommended option that we will use in the following steps:



Please enter the following parameters during your reservation:



The reservation usually takes about 15-20 minutes to complete and you will get an email indicating that your system is ready. Please, click on the link that points to your reservation because we will use some information for the next steps:



Now, assuming that your laptop runs MacOS, linux native or a linux VM on windows, issue the following commands in a terminal window of your local machine but change the external IP address and the PEM file with the data given at the end of the reservation page in the Techzone (see the picture above). You will be logged in your new system for the first time:

# LOGIN CELL  
export EXTERNAL\_IPADDR=\_\_\_replace\_with\_the\_external\_address\_ip\_address\_of\_the\_lab\_client\_\_  
export PEM\_FILE=\_\_\_replace\_with\_the\_file\_downloaded\_from\_the\_reservation\_button\_\_\_\_\_  
# Don't forget to change the file permissions  
chmod 600 $PEM\_FILE  
  
ssh itzuser@$EXTERNAL\_IPADDR -p 2223 -i $PEM\_FILE

The first thing you must do after logging is to clone the git repository of the workshop containing all the materials. Run this command on the lab client

# Clone the workshop repository in the lab client  
git clone https://github.com/angel-ibm/deployment-databand.git

## 2. Install packages

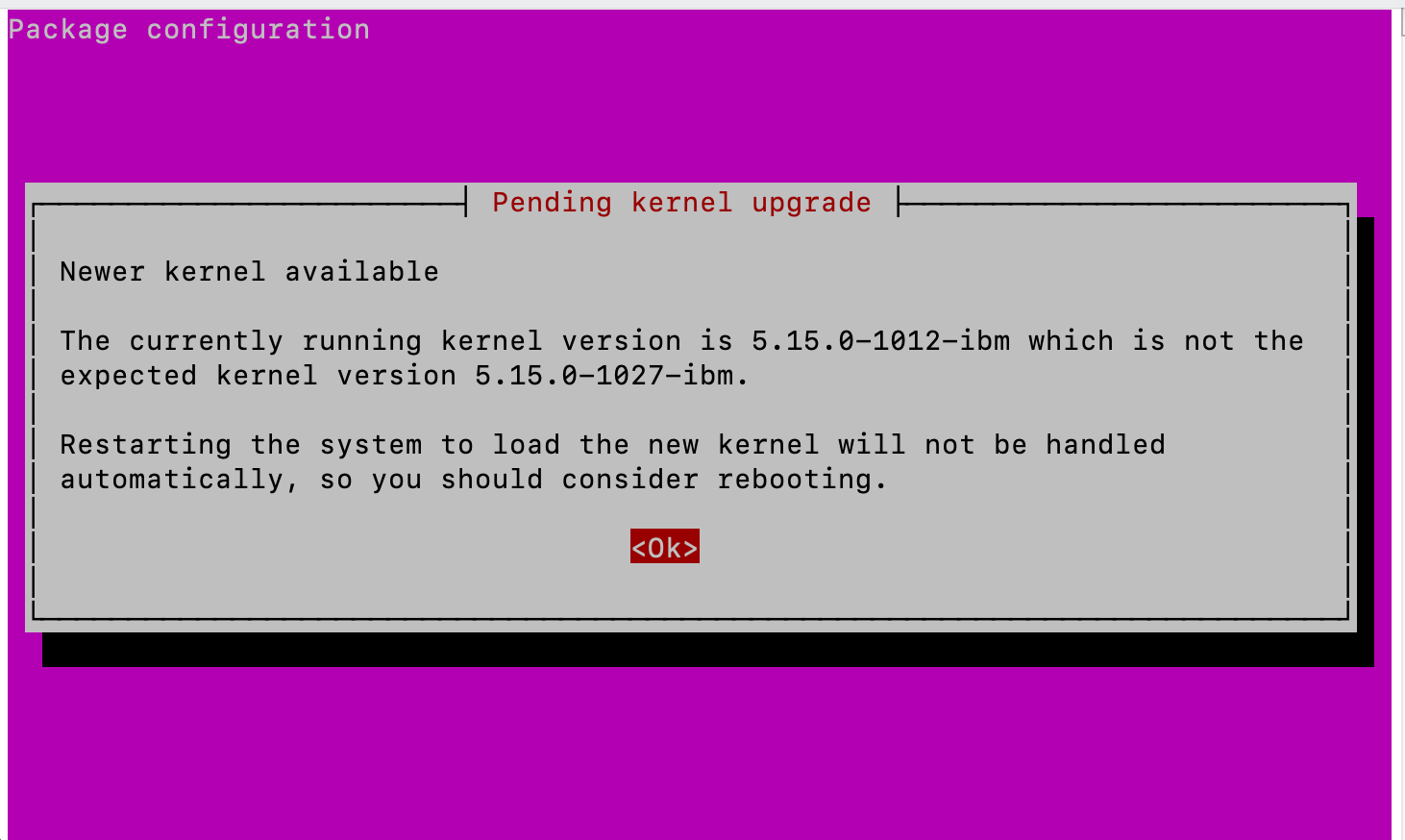
And now, you have two options:

* A: run a single script that installs all necessary packages
* B: run every single command step-by-step to do the same (only necessary if something goes wrong)

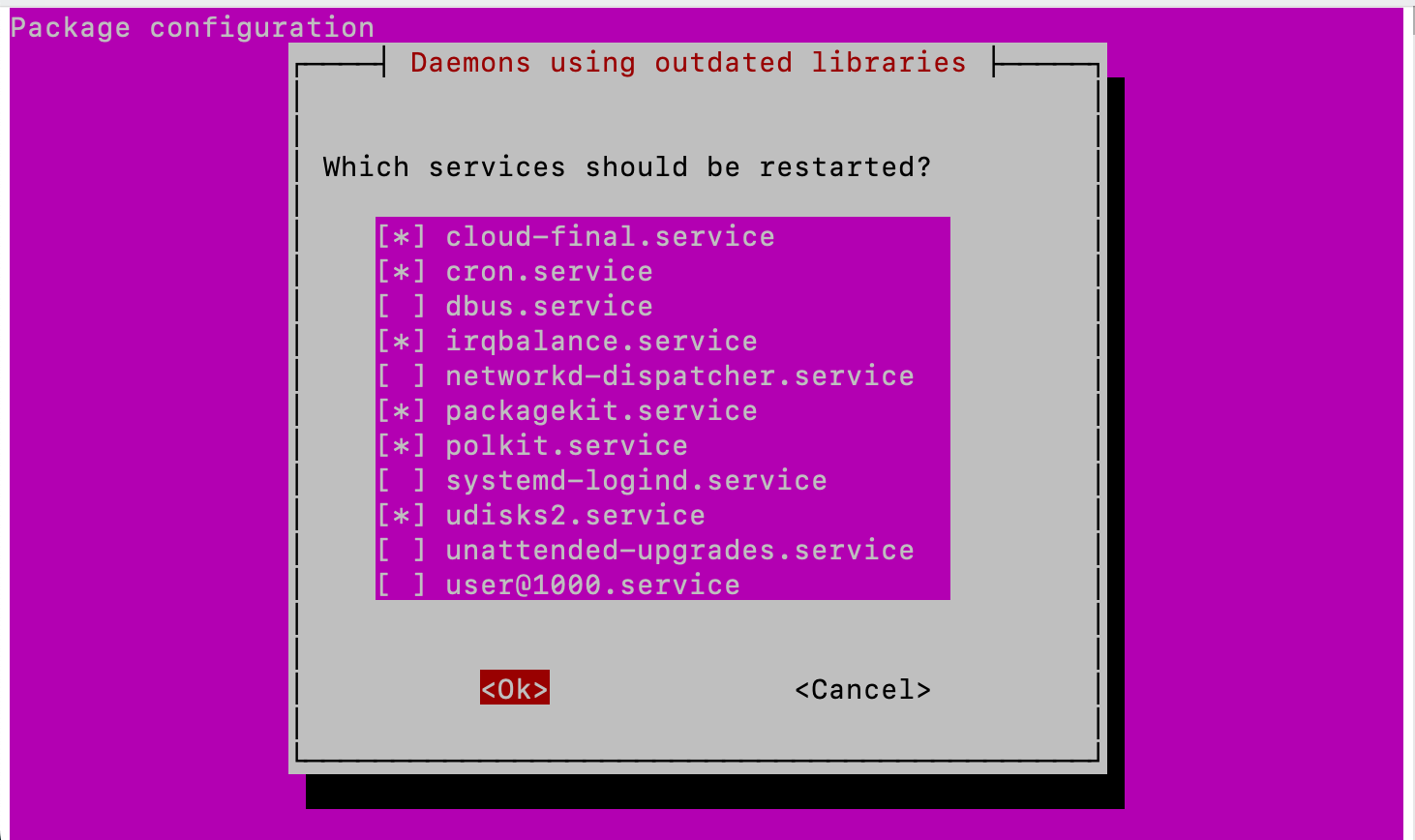
### A. Single install script

# Run this cell in the lab client to install automaticall all necessary packages  
cd ~/deployment-databand/client  
chmod +x ubuntu.sh  
./ubuntu.sh

The installation process will ask you to confirm some changes. Simply press the key if you see this prompt:



Or and then if you see a different prompt similar to this:



After approximately 15 minutesm if everything went well and you see jupyter lab starting, you are done. Skip the next section and go to the section 3 to start Jupyter.

### B. Step-by-step installation

You may want to run the following steps in the lab client if something went wrong with the single install script or if you just want to follow the process. Simply cut-and-paste each command in a terminal and execute them one by one, verifying the output:

# Create an alias for python3  
sudo apt update -y  
sudo apt upgrade -y  
  
echo "alias python=python3" >> .bash\_aliases  
. ./.bash\_aliases

# Install pip  
sudo apt install python3-pip -y

# Install Jupyter Notebooks  
sudo apt install jupyter -y  
pip install --upgrade jupyter\_core  
pip install jupyterlab  
pip install bash\_kernel  
python -m bash\_kernel.install  
pip install markupsafe==2.0.1  
. ./.profile

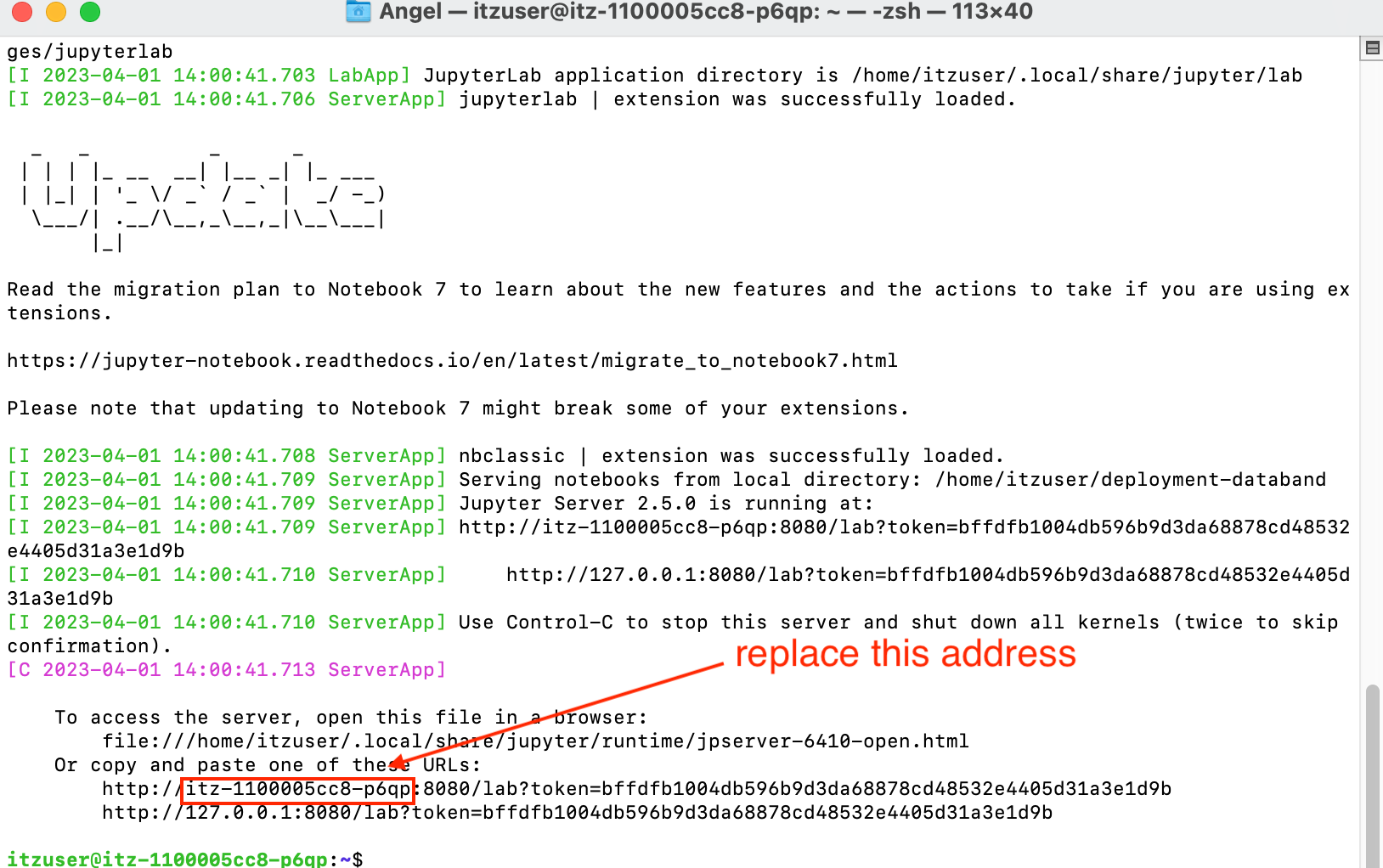
# Install snap  
sudo snap install yq  
  
# Install helm  
sudo snap install helm --classic  
  
# Install oc   
wget https://mirror.openshift.com/pub/openshift-v4/x86\_64/clients/ocp/4.10.54/openshift-client-linux-4.10.54.tar.gz  
tar -zxf openshift-client-linux-4.10.54.tar.gz  
sudo mv kubectl oc /usr/local/bin  
rm README.md openshift-client-linux-4.10.54.tar.gz

# Install docker  
sudo apt-get install ca-certificates curl gnupg  
sudo mkdir -m 0755 -p /etc/apt/keyrings  
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg  
echo \  
 "deb [arch="$(dpkg --print-architecture)" signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu \  
 "$(. /etc/os-release && echo "$VERSION\_CODENAME")" stable" | \  
 sudo tee /etc/apt/sources.list.d/docker.list > /dev/null  
sudo apt-get update  
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin -y  
sudo usermod -aG docker $USER

# Start Jupyter  
  
jupyter lab --ip 0.0.0.0 --port 8080 --notebook-dir ~/deployment-databand --no-browser & disown

## 3. Access Jupyter Lab from your local browser

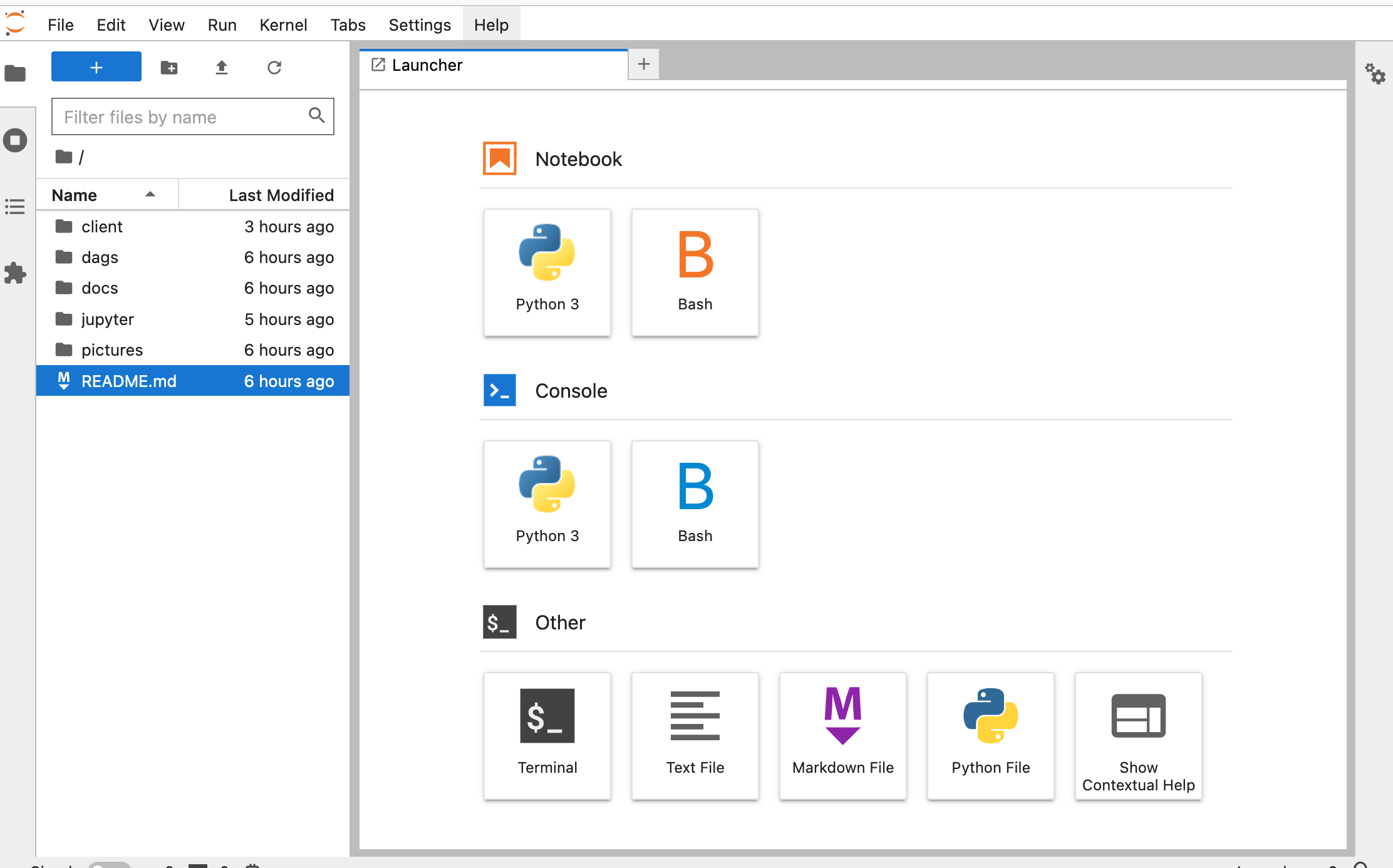
If everything went well in the previous sections, you will have seen the output of the jupyter lab server indicating the url for accessing it. However, this is not exactly what you must enter on your local browser. Please locate the following information and notice the address you will need to replace:



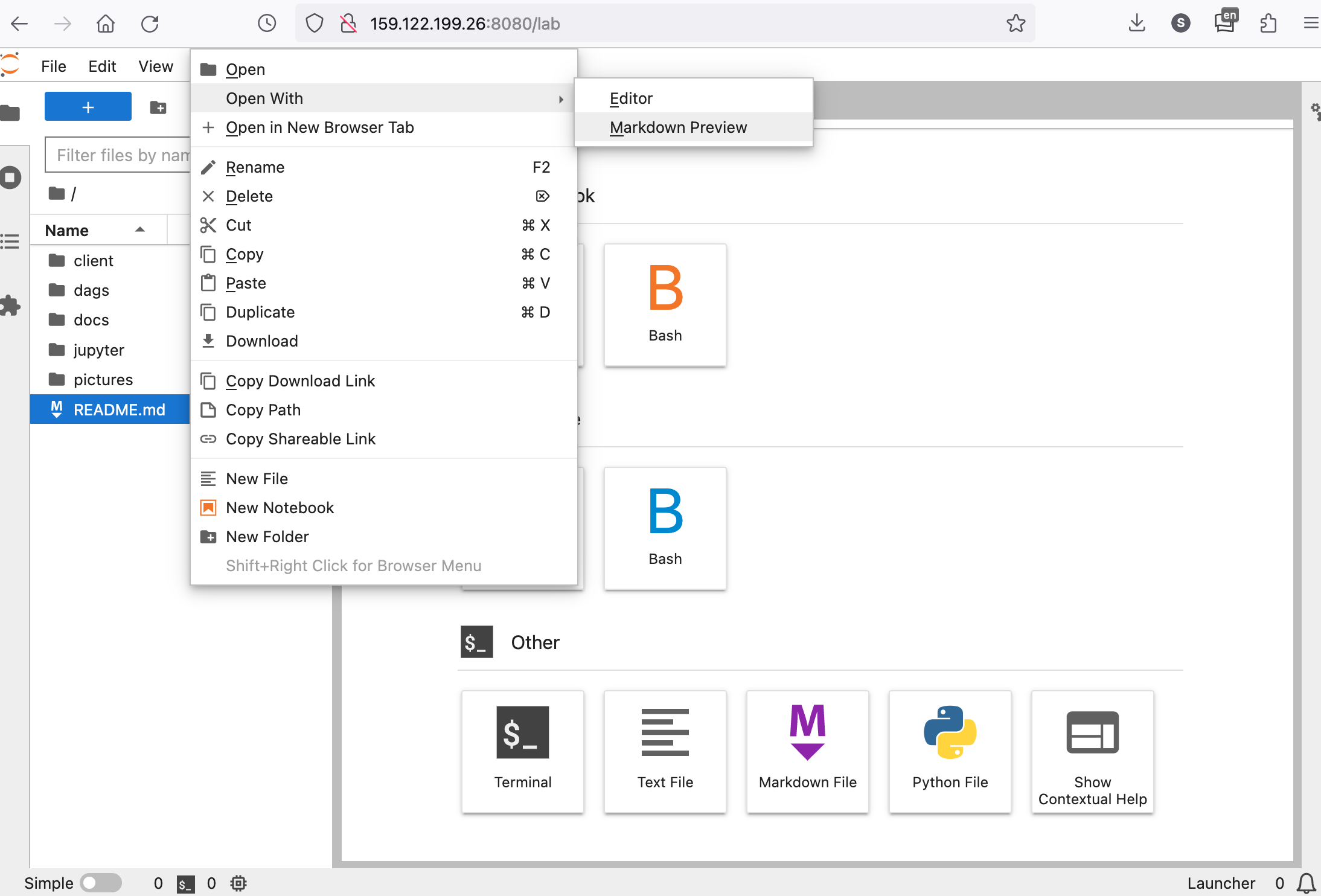
In the case displayed above, jupyter is reporting an external hostname but this name is for sure not known on your laptop. That is why we need to replace it with the external IP address of the lab client we obtained in the reservation page of the TechZone (see the screnshot of the section 1 above ).

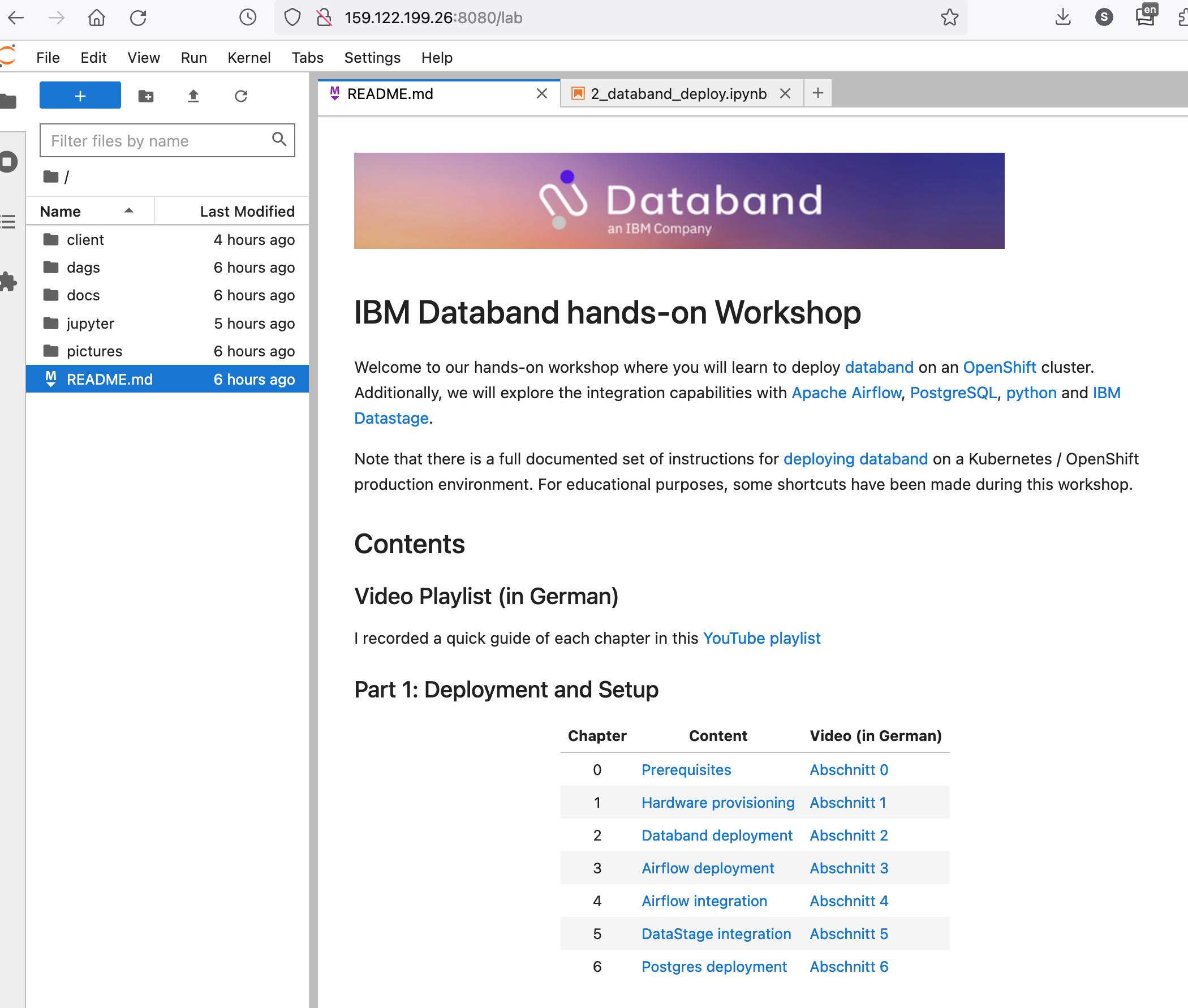
That means: this is the url we need to enter in our browser:

http://159.122.199.26:8080/lab?token=bffdfb1004db596b9d3da68878cd48532e4405d31a3e1d9b

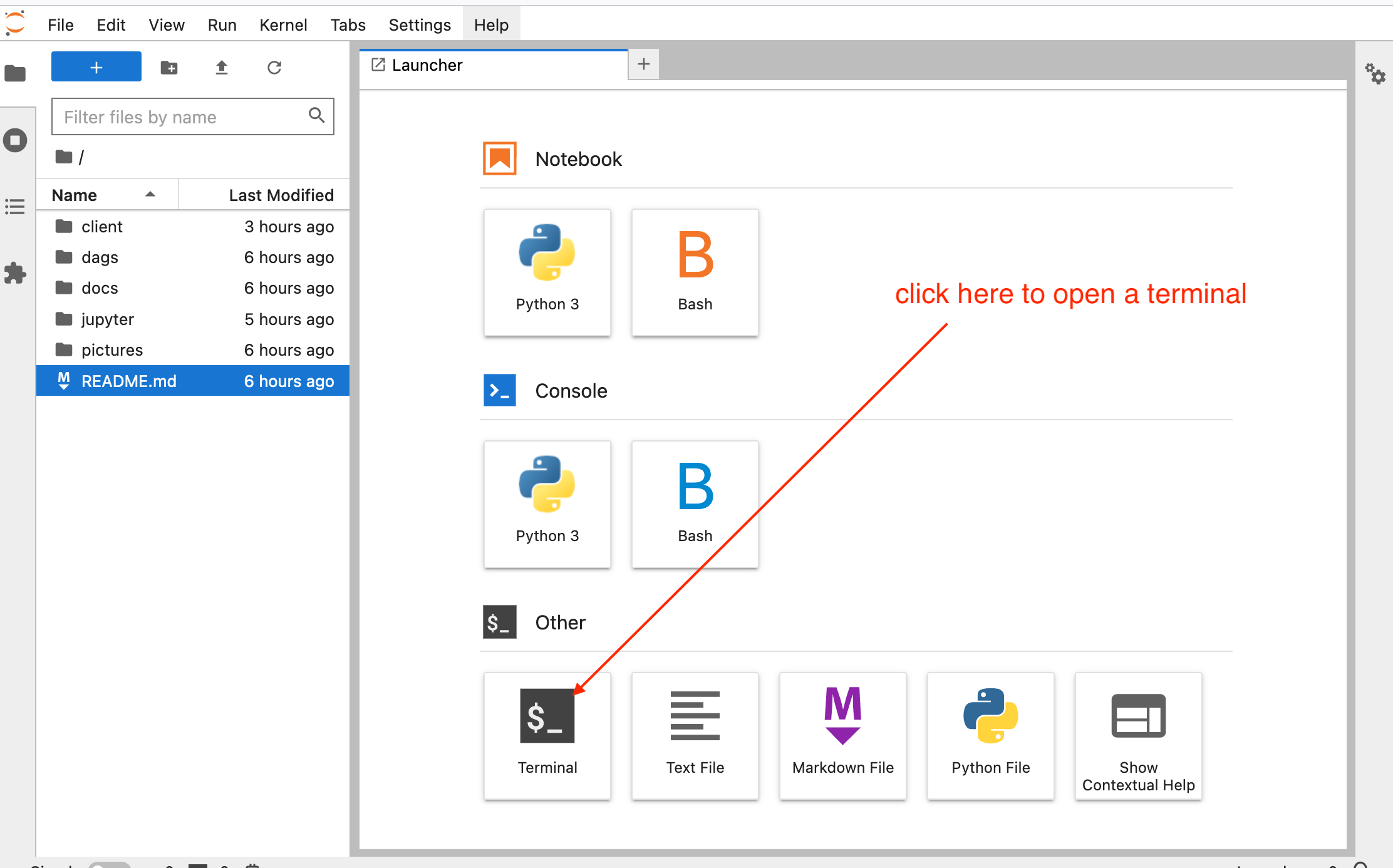


If you want to display the main page of the workshop (the Readme.md file), remember that Jupyter will not render a Markdown file unless you indicate it explictly. Click on the file with the right button of the mouse, select "open with" and click on "preview"





From this moment, you can start accessing the chapters of the workshop. Remember that you no longer need to open a terminal on your local laptop. Jupyter lab has an embedded terminal for your browser that can be used during this workshop (if needed at all)



Jupyter started in the background and it is safe to close the terminal on your local computer.

## 4. Considerations about Jupyter Lab

If the lab client ist rebooted, you must start jupyter once again with the command:

jupyter lab --ip 0.0.0.0 --port 8080 --notebook-dir ~/deployment-databand --no-browser & disown

You will get a new token that you need to cut and paste on your browser to access Jupyter. This may be somehow cumbersome and, if you want to avoid the token, you may execute:

jupyter server password

and type a password of your choice when prompted. Then, you may restart Jupyter again and access it from your browser without the token parameter, i.e:

http://159.122.199.26:8080/lab

You will need to enter the password directly on your browser when prompted.

Additionally, if you want Jupyter to start automatically after reboot, you can add an qucik entry to the cron. Follow this procedure:

1. create a start script named /home/itzuser/start\_jupyter.sh with the following content:

#!/usr/bin/bash  
  
 jupyter lab --ip 0.0.0.0 --port 8080 --notebook-dir ~/deployment-databand --no-browser & disown

Don't forget to give the execution permission:

chmod +x /home/itzuser/start\_jupyter.sh

1. as root (or sudo), edit the crontab of Ubuntu with the command:

crontab -e

1. add the following line at the end of the file:

@reboot sudo -i -u itzuser /home/itzuser/start\_jupyter.sh

1. save the file and Jupyter will start automatically after the reboot

## Important considerations

This document provided you with instructions for setting up a lab client with all necessary modules and packages in order that you don't have to use your own laptop, with the exception of the browser. However, there are things that we cannot (or may not) do automatically for you:

* obtaining the databand images (this is commercial software) and uploading them to the lab client
* setting up DataStage (that must be done with CloudPak for Data)
* provision an OpenShift cluster

Those activities are covered in other sections of the workshop.