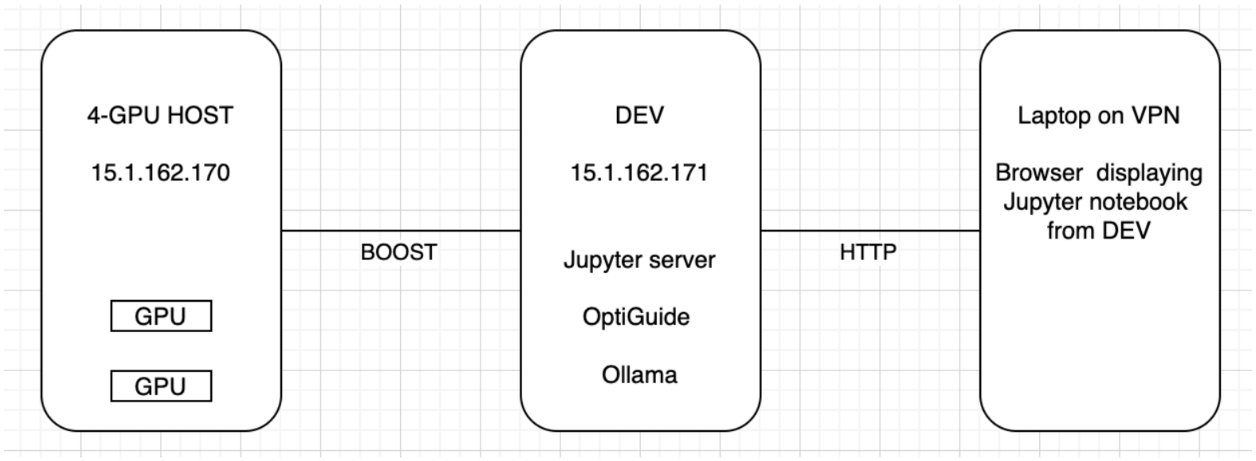
Installation Guide – Running Optiguide Supply Chain framework using Boost



Technical Requirements

For this setup you will need:

a HOST PC with GPUs and Boost
a DEV PC (with decent connection speed to the HOST) with Boost and additional software requitements listed below
and a laptop (for connecting to DEV and using Jupyter for remote connectivity instead of using HP Anyware – which can be very sluggish)

OS: Windows 10 or 11

Software Requirements

Ensure that you have the following software installed on DEV

Boost

Git https://git-scm.com/downloads/win

Python (3.11) - https://www.python.org/downloads/release/python-31111/

Pip - follow instructions from this website: https://pip.pypa.io/en/stable/installation/

Ollama - https://ollama.com/

Clone code from HP repo:

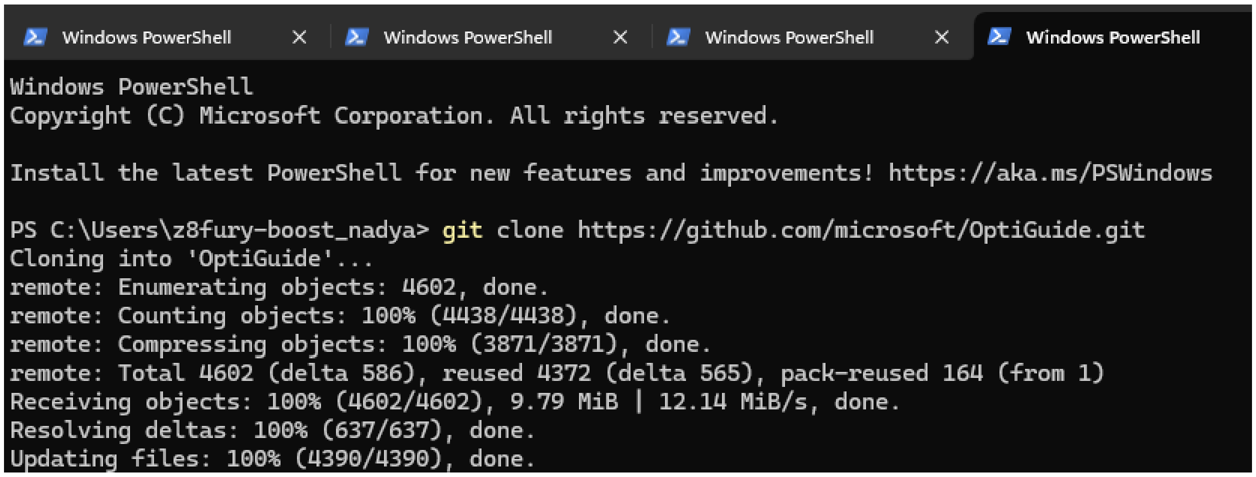
git clone https://github.azc.ext.hp.com/phoenix/Demos-for-Gartner-Amplify-and-GTC/tree/main/Nadya

<u>Install correct version for optiguide using requirements file:</u>

Go to terminal and change directory to:

Demos-for-Gartner-Amplify-and-GTC\Nadya\Optiguide (supply chain chatbot) on Boost

pip install -r requirements.txt



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

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\z8fury-boost_nadya> git clone https://github.com/microsoft/OptiGuide.git
Cloning into 'OptiGuide'...
remote: Enumerating objects: 4602, done.
remote: Counting objects: 100% (4438/4438), done.
remote: Compressing objects: 100% (3871/3871), done.
remote: Total 4602 (delta 586), reused 4372 (delta 565), pack-reused 164 (from 1)
Receiving objects: 100% (4602/4602), 9.79 MiB | 12.14 MiB/s, done.
Resolving deltas: 100% (637/637), done.
Updating files: 100% (4390/4390), done.
```

Jupyterlab installation steps

On DEV

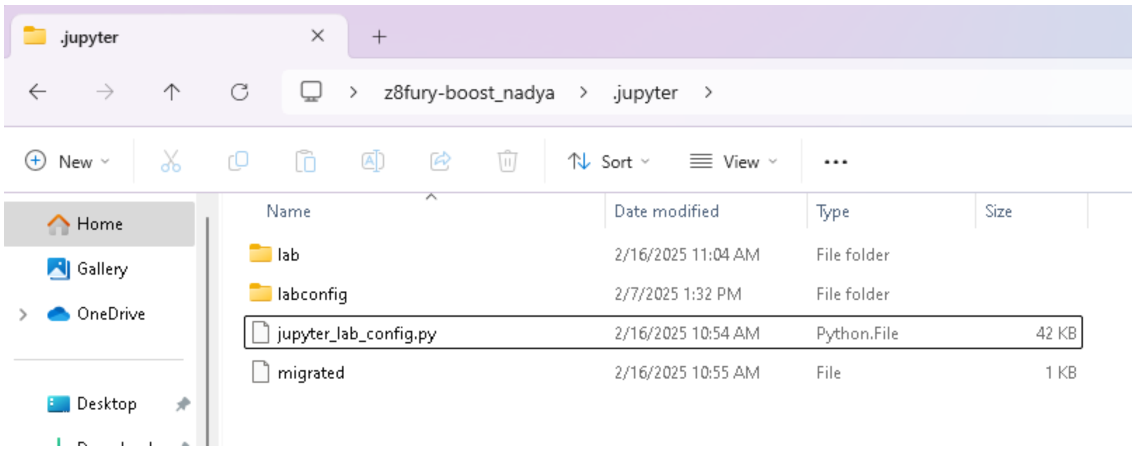
go to terminal, type:

pip install jupyterlab

To configure jupyterlab for remote access (so you can run notebooks remotely from your laptop), type:

jupyter lab –generate-config

Open jupyter_lab_config.py in ".jupyter" folder in your home directory:



Name	Date modified	Type	Size
lab	2/16/2025 11:04 AM	File folder	
labconfig	2/7/2025 1:32 PM	File folder	
jupyter_lab_config.py	2/16/2025 10:54 AM	Python.File	42 KB
migrated	2/16/2025 10:55 AM	File	1 KB

Open jupyter_lab_config.py with text editor,

1. find and replace:

# c.ServerApp.token = '<DEPRECATED>'

with a token, to make the remote login to work

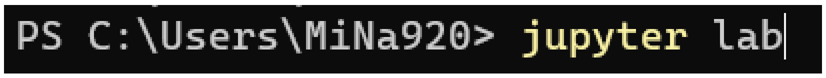
c.ServerApp.token = '314159'

2. find and replace

# c.ServerApp.ip = 'localhost'

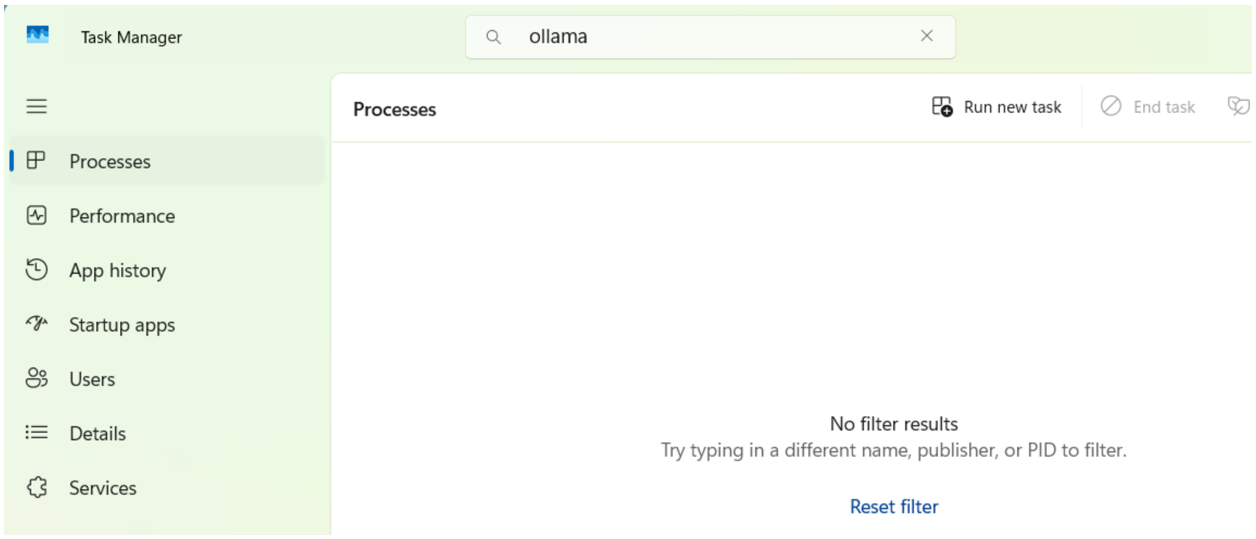
With: c.ServerApp.ip = '0.0.0.0'

Open terminal and start jupyter lab from your home folder



```
PS C:\Users\MiNa920> jupyter lab
```

Open another tab in Terminal, make sure Boost is connected to GPUs on the HOST and ollama is NOT running (open Task Manager and Search processes for ollama).



No filter results
Try typing in a different name, publisher, or PID to filter.

Pull the models:

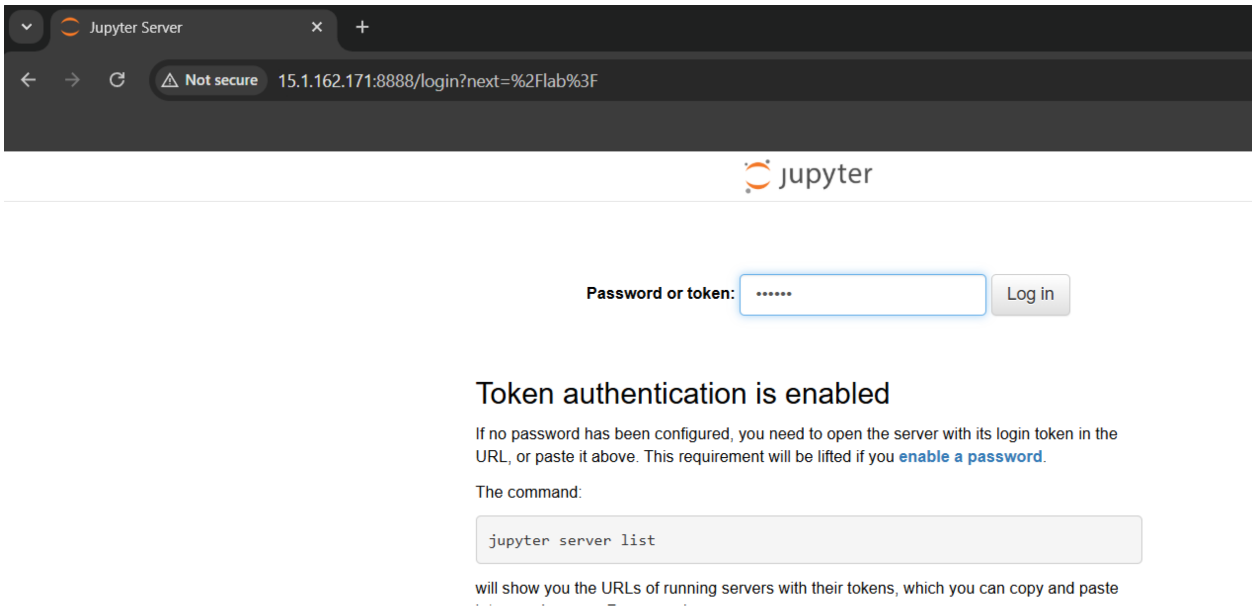
```
ollama pull llama3.2

ollama pull llama3.3
ollama serve
```

<u>On the laptop:</u>

Make sure your VPN stays connected.

Open browser and type: http://15.1.162.171:8888

Then type the token we created in jupyter-config file: **314159** in the prompt.


Jupyter Server

Not secure 15.1.162.171:8888/login?next=%2Flab%3F

Password or token: •••••• Log in

Token authentication is enabled

If no password has been configured, you need to open the server with its login token in the URL, or paste it above. This requirement will be lifted if you **enable a password**.

The command:

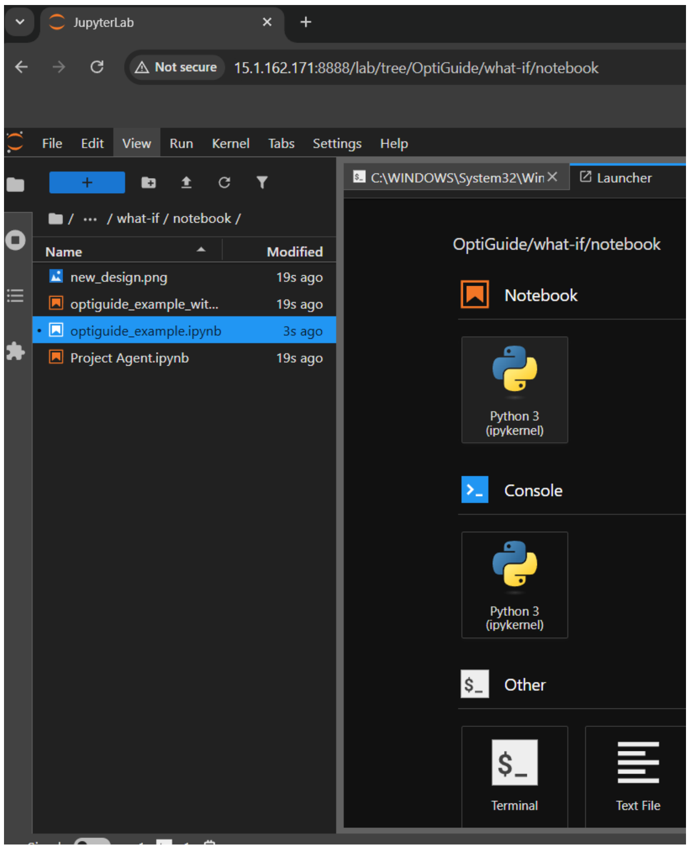
```
jupyter server list
```

will show you the URLs of running servers with their tokens, which you can copy and paste

Voila, now you can use JupyterNotebook and the terminal on DEV without remote desktop.

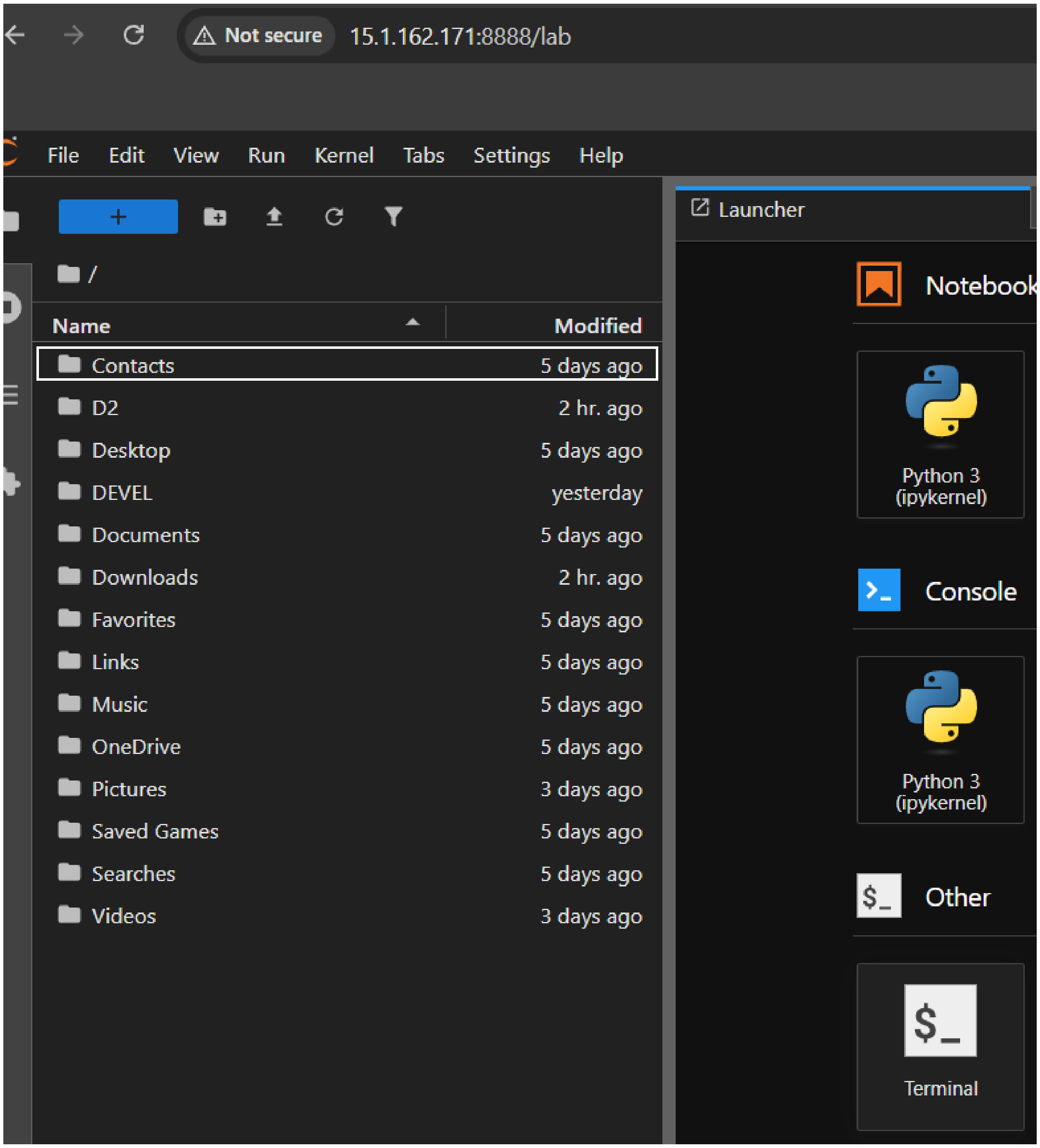
Now, let’s get the optiguide

In Jupyter Lab, open Launcher View by clicking the Plus sign in the upper left corner. This will open a Launcher View and list of available applications.



Optiguide setup

Pip



Name	Modified
Contacts	5 days ago
D2	2 hr. ago
Desktop	5 days ago
DEVEL	yesterday
Documents	5 days ago
Downloads	2 hr. ago
Favorites	5 days ago
Links	5 days ago
Music	5 days ago
OneDrive	5 days ago
Pictures	3 days ago
Saved Games	5 days ago
Searches	5 days ago
Videos	3 days ago

Now, open Notebook: optiguide_example.ipynb:

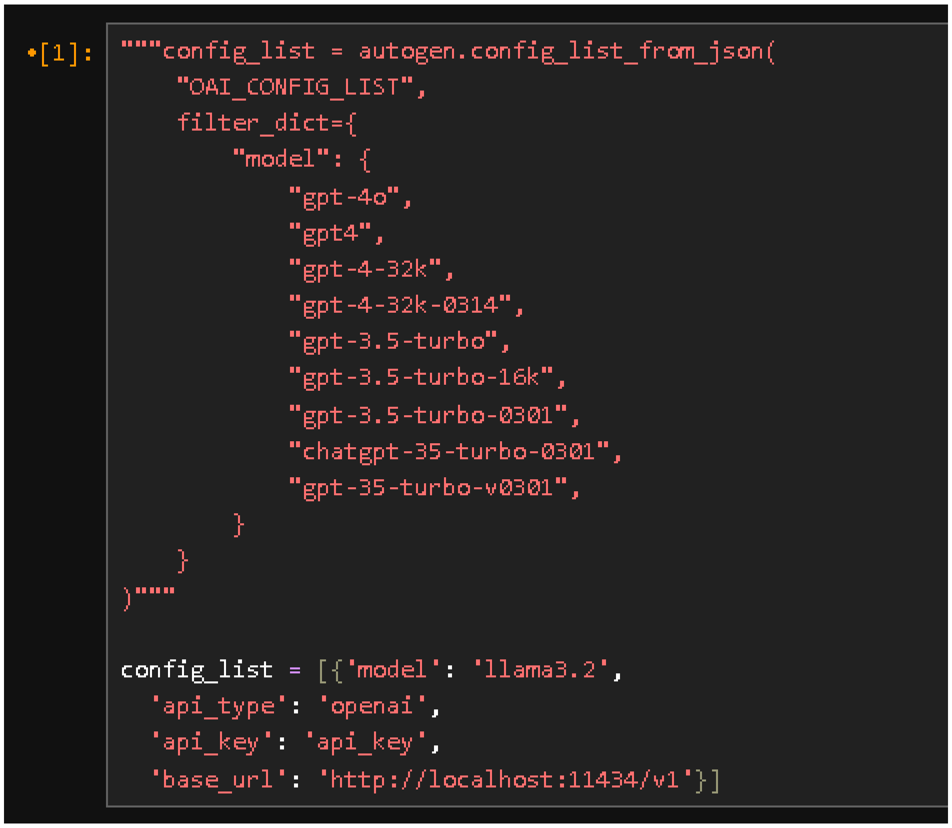
and change config_list variable to:

config_list = [{'model': 'llama3.2',

 'api_type': 'openai',

 'api_key': 'api_key',

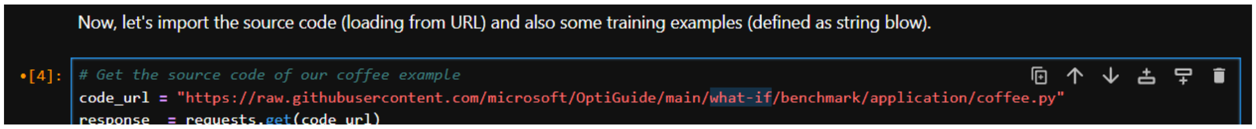
 'base_url': 'http://localhost:11434/v1'}]



```
•[1]: """config_list = autogen.config_list_from_json(
          "OAI_CONFIG_LIST",
          filter_dict={
              "model": {
                  "gpt-4o",
                  "gpt4",
                  "gpt-4-32k",
                  "gpt-4-32k-0314",
                  "gpt-3.5-turbo",
                  "gpt-3.5-turbo-16k",
                  "gpt-3.5-turbo-0301",
                  "chatgpt-35-turbo-0301",
                  "gpt-35-turbo-v0301",
              }
          }
      )"""

      config_list = [{'model': 'llama3.2',
        'api_type': 'openai',
        'api_key': 'api_key',
        'base_url': 'http://localhost:11434/v1'}]
```

Then change code_url in the notebook to this:

code_url = "https://raw.githubusercontent.com/microsoft/OptiGuide/main/what-if/benchmark/application/coffee.py"



Now, let's import the source code (loading from URL) and also some training examples (defined as string blow).

```
•[4]: # Get the source code of our coffee example
      code_url = "https://raw.githubusercontent.com/microsoft/OptiGuide/main/what-if/benchmark/application/coffee.py"
      response  = requests.get(code_url)
```

Now, execute code in the notebook cell by cell, and interact with the framework.