

Instalar Vapor API en un entorno de mambaforge en JupyterLab.

1. Debemos descargar MAMBAFORGE en nuestra carpeta de usuario. Usando los siguientes comandos:

```
curl -L -O https://github.com/conda-forge/miniforge/releases/latest/download/Mambaforge-Linux-x86\_64.sh
```

Esto descargará el archivo.sh directamente del GitHub original de MambaForge.

2. Para que Ubuntu lea y realice la instalación de este archivo en nuestra carpeta de usuario ejecutamos el siguiente comando:

```
bash Mambaforge-Linux-x86_64.sh
```

3. Una vez instalado aparecerá una carpeta llamada “mambaforge”. Para ingresar a cualquier entorno creado con esta instalación de mambaforge se deberá llamar el archivo “conda.sh” con el siguiente comando y su correspondiente ruta.

```
source mambaforge/etc/profile.d/conda.sh
```

4. Activamos el entorno base:

```
conda activate base
```

5. Creamos el entorno:

```
conda create -n vapor
```

6. Activamos este nuevo entorno para instalar las paqueterías necesarias:

```
conda activate vapor
```

7. Instalamos los paquetes siguientes:

```
mamba install Python=3.9.5
```

```
mamba install -c conda-forge -c ncar-vapor vapor
```

```
mamba install nb_conda_kernels
```

```
mamba install ipykernel
```

```
mamba install -c anaconda pyopengl
```

```
mamba install -c conda-forge netcdf4
```

```
mamba install -c anaconda ipywidgets=7.7.2
```

8. Revisemos la lista de kernels que tenemos actualmente creados con nuestra instalación de mambaforge.

`python -m nb_conda_kernels list`

Veremos uno nombrado de la siguiente forma y se indicara la ruta donde esta guardado el archivo .json (la ruta puede variar ligeramente solo por los dos primeros directorios):

`conda-env-vapor-py /home/a.7023/mambaforge/envs/vapor/share/jupyter/kernels/python3`

Deberemos verificar que tenga el nombre que le dimos al entorno.

9. Yendo a la ruta proporcionada encontraremos el archivo .json y 3 imágenes con formato .png que caracterizan a nuestro kernel en el “Launcher”.
10. Podemos ir a la ruta ya sea con la consola, o usando la interfaz de JupyterLab. El objetivo es editar el archivo kernel para CAMBIAR EL NOMBRE de este (figura 1 y 2).

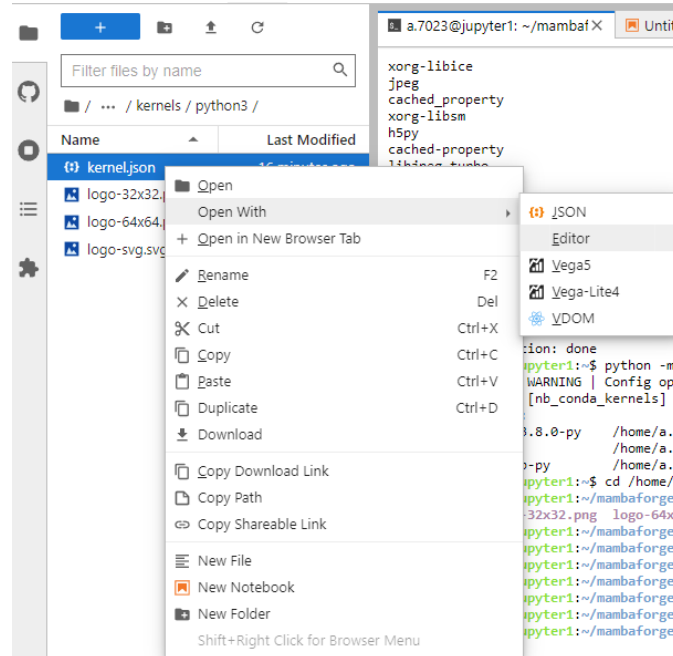
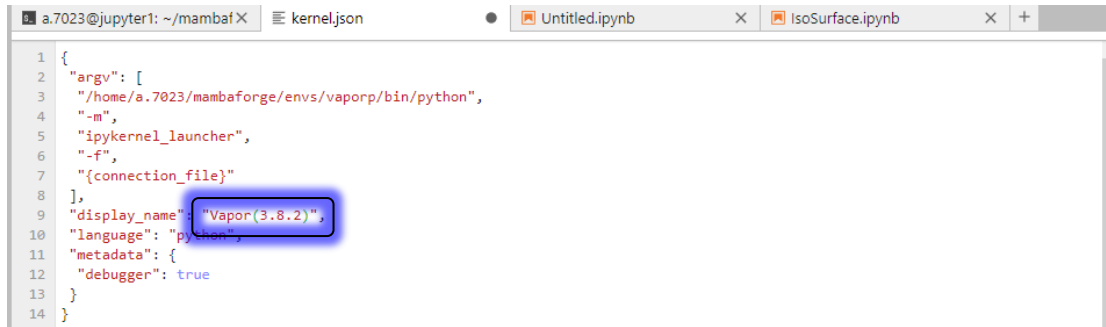


Figura 1



Figura 2

11. En la figura 2 se indica la opción a modificar y es donde ingresaremos el nombre que queremos que nuestro kernel tenga (figura 3).



```
1 {
2   "argv": [
3     "/home/a.7023/mambaforge/envs/vaporp/bin/python",
4     "-m",
5     "ipykernel_launcher",
6     "-f",
7     "{connection_file}"
8   ],
9   "display_name": "Vapor(3.8.2)",
10  "language": "python",
11  "metadata": {
12    "debugger": true
13  }
14 }
```

Figura 3

12. Para cambiar la imagen que aparece en el Launcher, que por defecto es el logo de Python deberemos sustituir los archivos .png marcados en la figura 4,

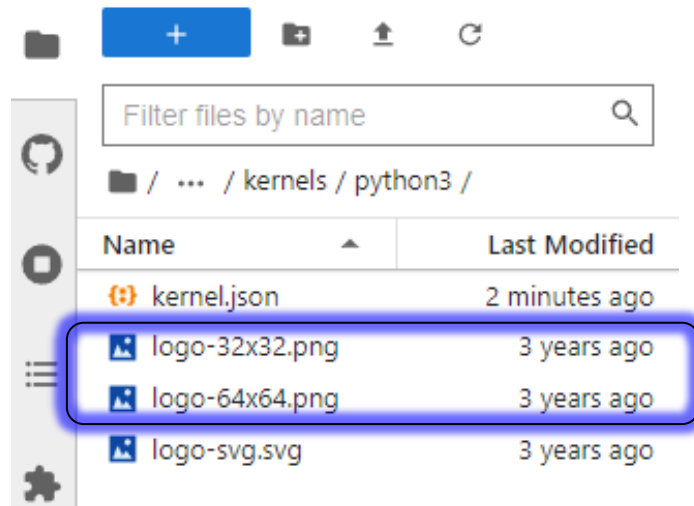


Figura 4

por otros archivos con el mismo nombre pero que representan el logo de Vapor (figura 5).

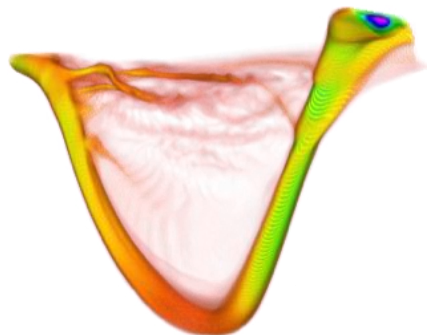


Figura 5

NOTA: Estas imágenes deben tener obligatoriamente formato .png, y ser de un tamaño de 64x64 y 32x32 respectivamente.

13. Al realizar estos últimos pasos, finalmente nuestro kernel se vera de esta forma:

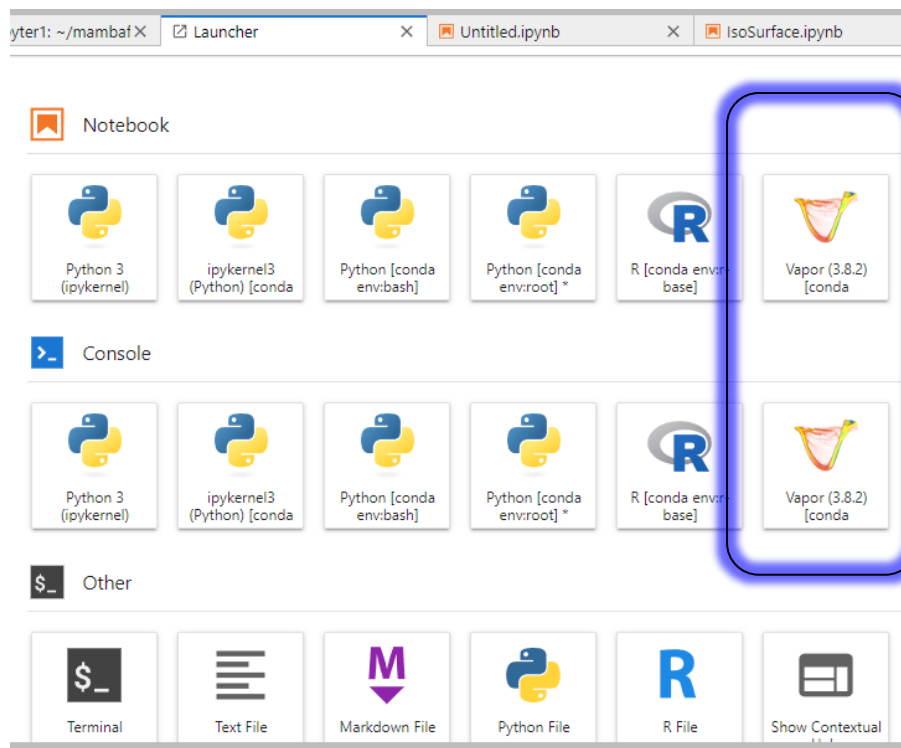


Figura 6

14. Algo que se debe considerar es lo siguiente: Vapor cuenta con un módulo de widgets que son extremadamente útiles. Sin embargo, estos deben ser instalados como un paquete adicional y podrán ser usados sin inconvenientes.
15. Si eres usuario root y tienes permitido hacer cualquier cambio en toda la instalación de jupyterlab y Ubuntu, bastará con instalar los siguientes paquetes en el entorno donde tienes instalado JupyterLab (probablemente sea el mismo donde instalaste Vapor)

```
mamba install jupyterlab
```

```
mamba install -c conda-forge jupyterlab_widgets
```

```
jupyter labextension install @jupyter-widgets/jupyterlab-manager
```

```
pip install jupyter_vapor_widget
```

16. Para comprobar que los paquetes `@jupyter-widgets/jupyterlab-manager` y `jupyter_vapor_widget` se instalaron correctamente y que están habilitados, usamos el siguiente comando:

`jupyter labextension list`

`jupyter nbextension list` (para las extensiones de notebook. Si no aparecen, cambiar una versión anterior de notebook: `pip install notebook== 6.1.5` o `mamba install notebook=6.5.2`)

17. Y habilitamos las extensiones desde la interfaz de JupyterLab.

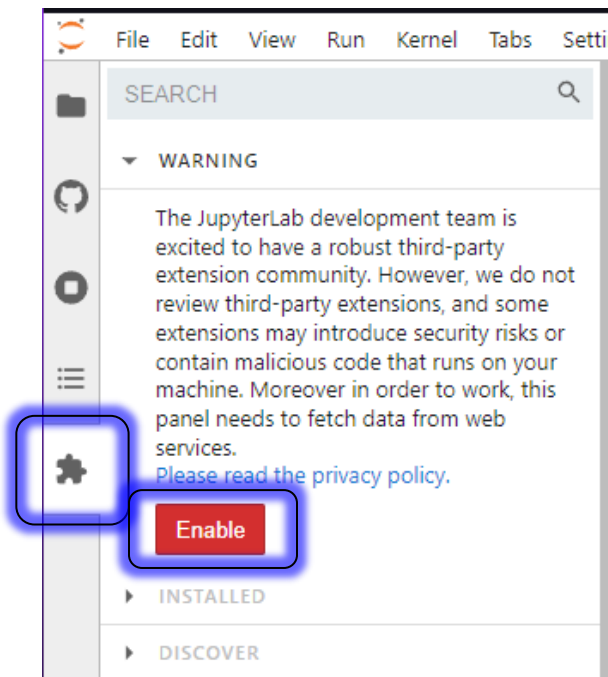


Figura 7

18. Si no eres usuario root y estas restringido en la instalación de Ubuntu y JupyterLab, quien deberá instalar los paquetes debe ser el administrador. Y lo hará en el entorno donde instaló JupyterLab.