Guide d'installation de JupyterLab sur Debian 12

Jupyter est une application Web gratuite et open source pour l'informatique interactive et la science des données. Jupyter prend en charge tout langages de programmation et fournit plusieurs logiciels, tels que JupyetrLab, qui fournit une interface riche en fonctionnalités et à onglets environnement d'édition multi-ordinateurs portables, Notebook en tant que création de notebook légère et simplifiée, Qtconsole et bien d'autres plus.

Dans ce guide, nous vous guiderons pas à pas dans l'installation de JupyterLab sur Debian 12. Vous allez installer JupyterLab sur l'environnement virtuel Python, en exécutant JupyterLab en tant que service systemd, vous installerez et configurerez Nginx en tant que service proxy inverse pour JupyterLab.

Conditions préalables

Avant de commencer, confirmez que vous disposez des éléments suivants :

- Un serveur Debian 12.
- Un utilisateur non root avec des privilèges d'administrateur.

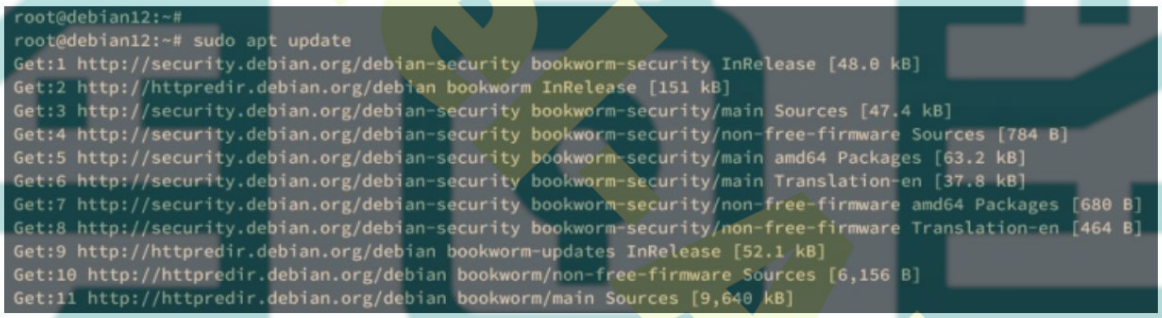
Installation des dépendances

Jupyter est un écosystème de logiciels Python gratuits et open source pour l'informatique interactive dans toute la programmation langues. Pour installer Jupyter, vous devez vous assurer que Python est installé sur votre système.

Dans cette section, vous installerez les dépendances du package pour Jupyter, qui incluent Python3, Pip package manager, Python venv virtual environment, et Node.js.

Avant de commencer, mettez à jour et actualisez l'index de votre paquet Debian à l'aide de la commande suivante.

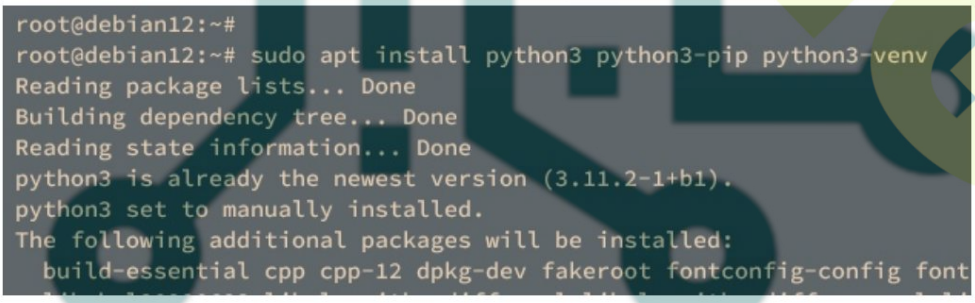
```
sudo apt update
```



Installez maintenant les dépendances en exécutant la commande 'apt install ' . Avec cela, vous installerez le Python3, Pip package manager, venv virtual environment management, et Node.js.

```
sudo apt install python3 python3-pip python3-venv nodejs
```

Tapez y pour confirmer et procéder à l'installation.



Une fois les dépendances installées, exécutez la commande suivante pour vérifier la version de chaque dépendance, y compris Python3, Pip et Node.js

```
python3 --version

pip3 --version nœud --version
```

Le résultat ci-dessous vous montre que Python 3.11, Pip 23 et Node.js 18.13 sont installés.

Installer Jupyter

Après avoir installé les dépendances, vous installerez ensuite les méta-packages Jupyter à l'aide de l'environnement virtuel Python sur la machine Debian 12. Pour y parvenir, vous devez effectuer les opérations suivantes :

- Création d'un environnement virtuel Python
- Installer Jupyter
- Configuration de JupyterLab

Commençons.

Création d'un environnement virtuel Python

Dans cette section, vous allez créer un nouvel environnement virtuel Python qui sera utilisé pour l'installation de Jupyter. Avec ça Dans ce scénario, vous disposerez d'un environnement de développement isolé qui n'affectera pas l'ensemble de votre système. Aussi, vous pouvez recréez cela au fil du temps.

Connectez-vous à votre utilisateur non root à l'aide de la commande suivante.

```
su - user
```

Créez maintenant un nouveau ~/projet et emménagez-y. Ensuite, créez un nouvel environnement virtuel appelé venv à l'aide du répertoire en suivant la commande. Après avoir exécuté la commande, le nouveau répertoire venv sera créé.

```
mkdir -p ~/project; cd ~/project
python3 -m venv venv
```

Ensuite, exécutez la commande suivante pour activer l'environnement virtuel venv. Une fois activé, l'invite de votre shell devient comme "(venv) user@hostname:#...".

```
source venv/bin/activate
```

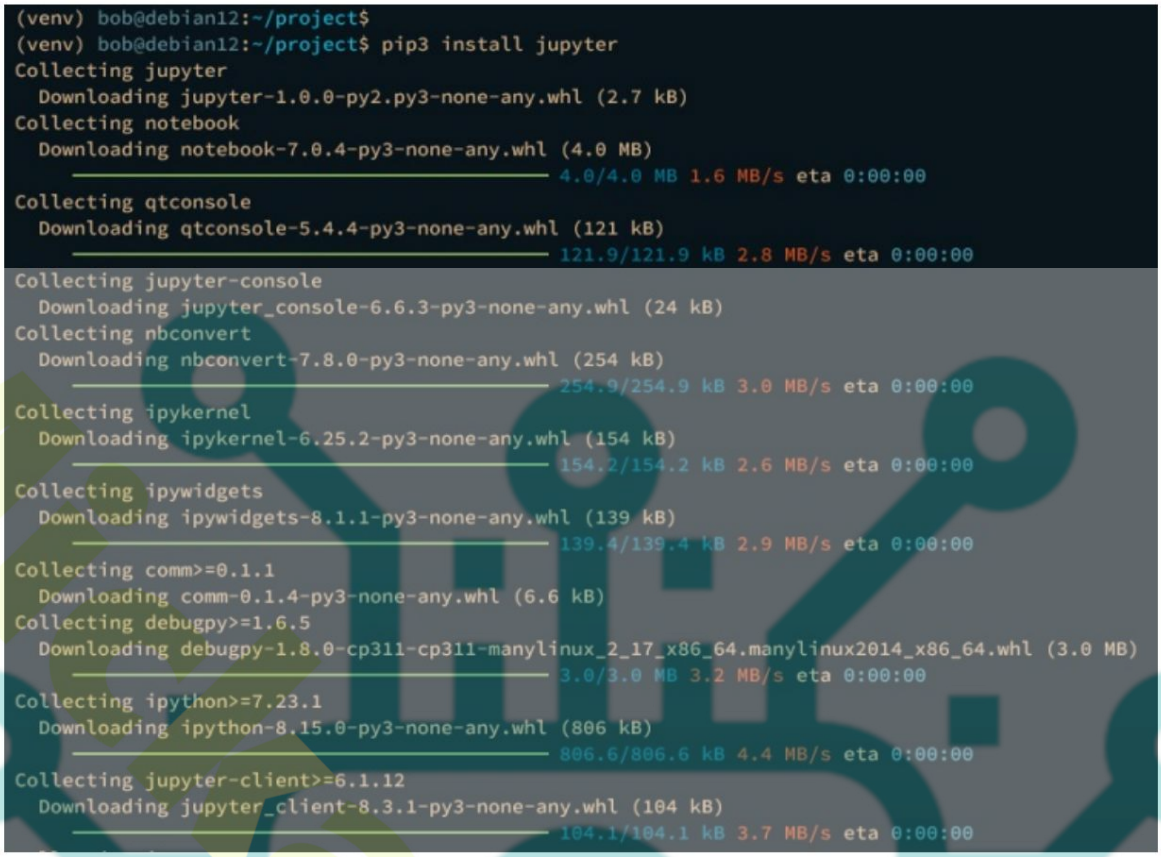
Installer Jupyter

Après avoir créé un environnement virtuel Python. vous installerez le package Jupyter via Pip. Le package Jupyter est une méta package d'écosystèmes Jupyter, qui comprend IPython, JupyterLab, Jupyter Server, Jupyter Notebook, qtconsole et beaucoup plus.

Exécutez la commande pip3 ci-dessous pour installer Jupyter sur votre système Debian.

```
pip3 install jupyter
```

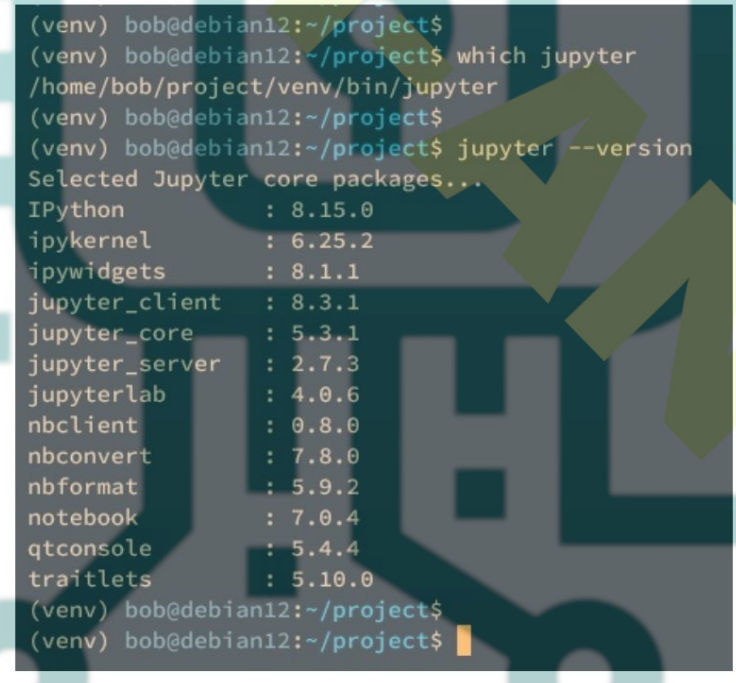
Lors de l'installation, vous trouverez ci-dessous la sortie qui s'affichera sur votre terminal :



Une fois l'installation terminée, vérifiez l'emplacement du fichier binaire Jupyter et sa version à l'aide de la commande suivante.

```
which jupyter
jupyter --version
```

Dans le résultat suivant, vous devriez voir que le fichier binaire Jupyter se trouve dans le répertoire venv. Vous devriez également voir la version du package Jupyter installée sur votre système.



Configuration de JupyterLab

JupyterLab est l'interface utilisateur Web de nouvelle génération pour l'écosystème Jupyter. Avec JupyterLab, vous pouvez travailler avec des documents et des activités telles que Jupyter Notebook, un éditeur de texte et un terminal dans une seule fenêtre. Le JupyterLab vous offre une expérience de type IDE pour gérer et développer des projets Jupyter.

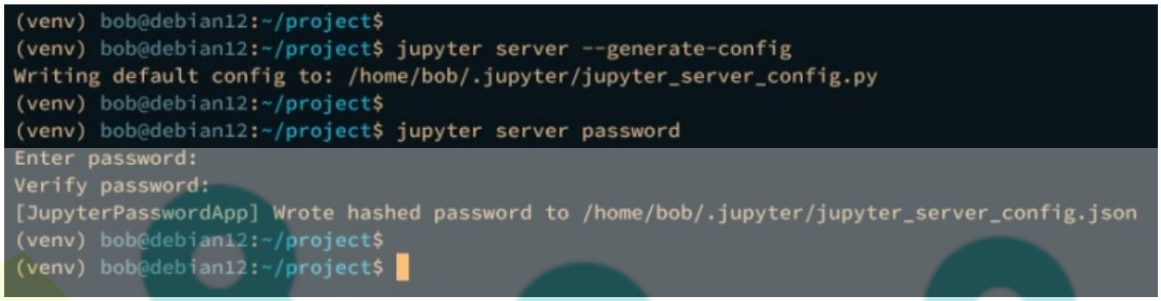
Dans cette section, vous allez configurer JupyterLab pour votre projet Jupyter. Mais avant cela, vous devez d'abord configurer le serveur Jupyter pour sécuriser votre installation.

Tout d'abord, exécutez la commande suivante pour générer le fichier de configuration du serveur Jupyter et configurer le mot de passe. Saisissez votre mot de passe et répétez.

```
jupyter server --generate-config
```

```
jupyter server password
```

Après avoir exécuté la commande, la configuration du serveur Jupyter sera générée à ~/.jupyter/ directory.



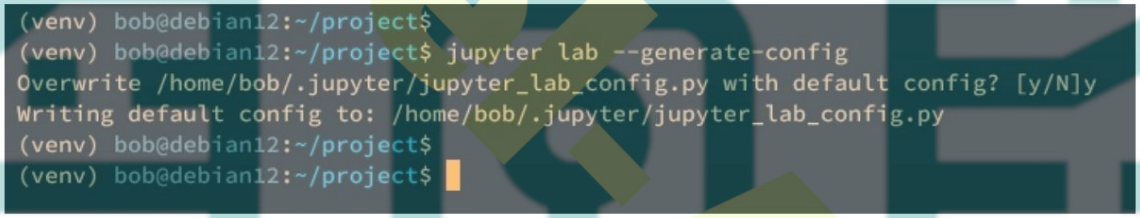
Vous pouvez vérifier le contenu de la configuration du serveur Jupyter à l'aide de la commande suivante.

```
jupyter server --show-config
```

La sortie similaire ci-dessous sera affichée :

Ensuite, exécutez la commande suivante pour générer une nouvelle configuration pour JupyterLab. La configuration de JupyterLab sera généré dans le répertoire ~/.jupyter/

```
jupyter lab --generate-config
```



Une fois la configuration générée, vérifiez la configuration JupyterLab à l'aide de la commande suivante.

```
jupyter lab --show-config
```

Vous trouverez ci-dessous un exemple de configuration JupyterLab :

Vous pouvez maintenant exécuter l'installation de JupyterLab à l'aide de la commande suivante. Assurez-vous de changer l'adresse IP avec votre Adresse IP du serveur.

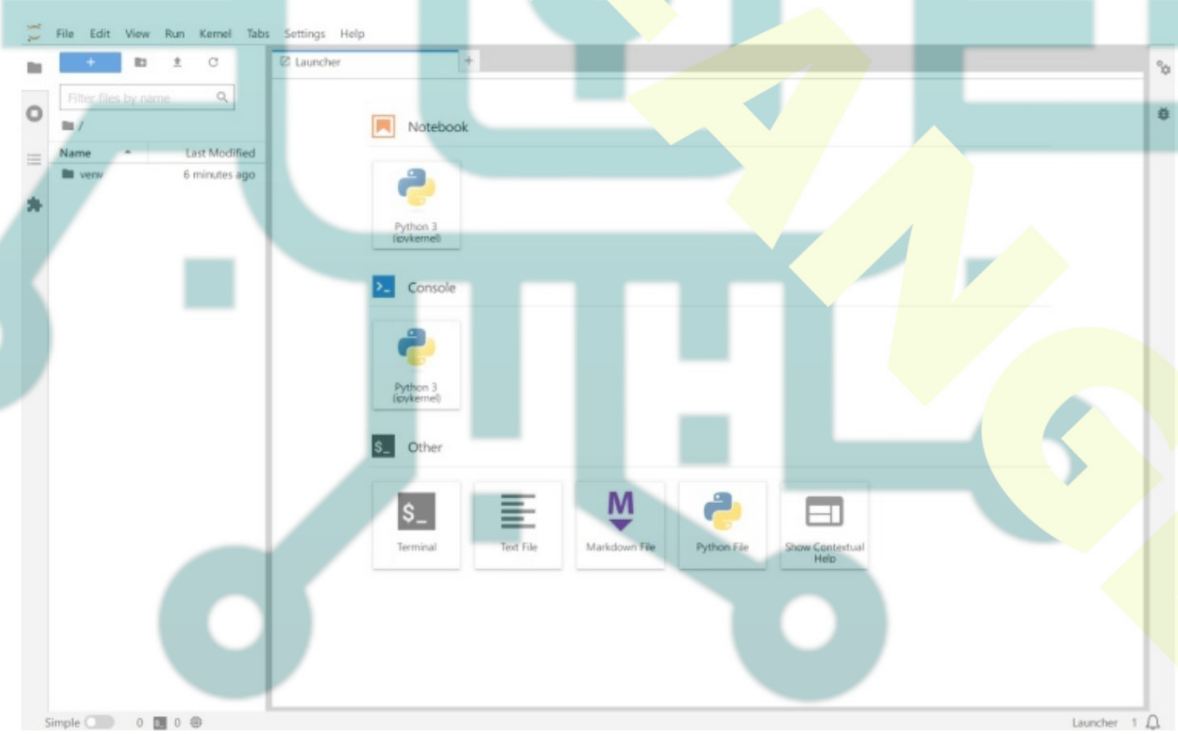
```
jupyter lab --ip 192.168.10.15
```

Le JupyterLab s'exécutera sur l'adresse IP de votre serveur avec le port par défaut 8888. Le JupyterLab est accessible via une URL chemin /lab.

```
(venv) bob@debian12:~/project$
(venv) bob@debian12:~/project$ jupyter lab --ip 192.168.10.15
                                Package jupyterlab took 0.0000s to import
                                Package jupyter_lsp took 0.0131s to import
                                A `_jupyter_server_extension_points` function was not found in jupyter_lsp. Instead, a `_jupyter
_server_extension_paths` function was found and will be used for now. This function name will be deprecated in future releases of Jupy
ter Server.
                                Package jupyter_server_terminals took 0.0059s to import
                                Package notebook took 0.0000s to import
                                Package notebook_shim took 0.0000s to import
                                A `_jupyter_server_extension_points` function was not found in notebook_shim. Instead, a `_jupyt
er_server_extension_paths` function was found and will be used for now. This function name will be deprecated in future releases of Ju
pyter Server.
                                jupyter_lsp | extension was successfully linked.
                                jupyter_server_terminals | extension was successfully linked.
                                jupyterlab | extension was successfully linked.
                                notebook | extension was successfully linked.
                                notebook_shim | extension was successfully linked.
                                notebook_shim | extension was successfully loaded.
                                jupyter_lsp | extension was successfully loaded.
                                jupyter_server_terminals | extension was successfully loaded.
                               pyterLab extension loaded from /home/bob/project/venv/lib/python3.11/site-packages/jupyterlab
                               pyterLab application directory is /home/bob/project/venv/share/jupyter/lab
                               tension Manager is 'pypi'.
                                jupyterlab | extension was successfully loaded.
                                notebook | extension was successfully loaded.
                                Serving notebooks from local directory: /home/bob/project
                                Jupyter Server 2.7.3 is running at:
                                http://192.168.10.15:8888/lab
                                    http://127.0.0.1:8888/lab
                                Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
                                No web browser found: Error('could not locate runnable browser').
                                Skipped non-installed server(s): bash-language-server, dockerfile-language-server-nodejs, javasc
ript-typescript-langserver, jedi-language-server, julia-language-server, pyright, python-language-server, python-lsp-server, r-languag
eserver, sql-language-server, texlab, typescript-language-server, unified-language-server, vscode-css-languageserver-bin, vscode-html-
languageserver-bin, vscode-json-languageserver-bin, yaml-language-server
```

Ouvrez votre navigateur Web et visitez l'adresse IP de votre serveur suivie du port 8888 et du chemin tel que path /lab, http://192.168.10.15:8888/lab. Saisissez le mot de passe de votre serveur Jupyter et cliquez sur Connexion pour continuer.

Si vous disposez du mot de passe approprié, vous devriez voir le tableau de bord utilisateur JupyterLab comme suit :



Vous pouvez maintenant appuyer deux fois sur Ctrl+c pour terminer le processus JupyterLab.

Exécuter JupyterLab en tant que service Systemd

À l'étape suivante, vous créerez un nouveau fichier de service systemd qui sera utilisé pour exécuter JupyterLab. De cette façon, votre L'installation de JupyterLab s'exécutera en arrière-plan en tant que service systemd et vous pourrez facilement gérer JupyterLab via l'utilitaire systemctl.

Créez un nouveau fichier de service systemd /etc/systemd/system/jupyterlab.service à l'aide de la commande suivante de l'éditeur nano.

```
sudo nano /etc/systemd/system/jupyterlab.service
```

Insérez la configuration suivante et assurez-vous de changer le nom d'utilisateur **bob.**

```
[Unité]
Description=Service JupyterLab

[Service]
Type=simple
PIDFile=/run/jupyter.pid
ExecStart=/home/bob/project/venv/bin/jupyter lab --config=/home/bob/.jupyter/jupyter_lab_config.py User=bob Group=bob


WorkingDirectory=/ accueil/bob/projet
Restart=toujours
RestartSec=10

[Installer]
WantedBy=multi-user.target
```

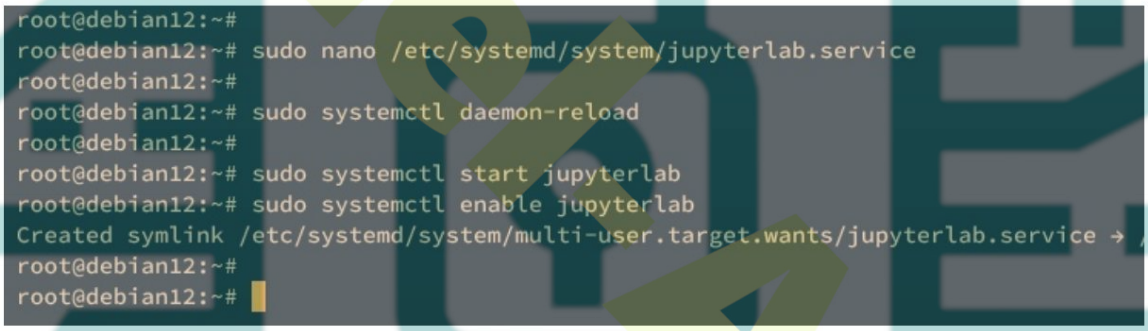
Une fois terminé, enregistrez et fermez le fichier.

Maintenant, exécutez ce qui suit systèmectl commande pour recharger le gestionnaire systemd et appliquer les modifications.

```
sudo systemctl daemon-reload
```

Ensuite, démarrez et activez le service jupyterlab à l'aide de la commande suivante.

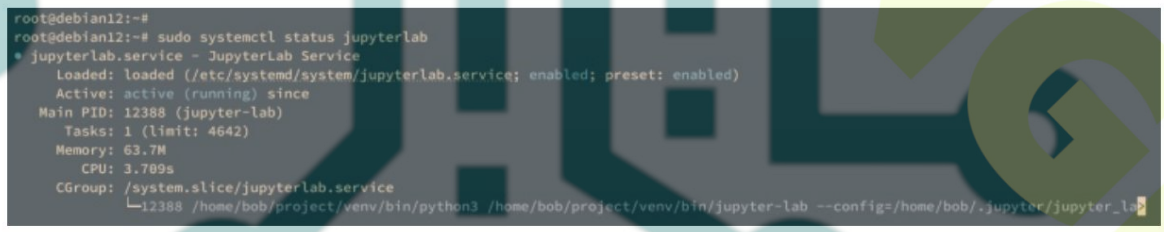
```
sudo systemctl start jupyterlab
sudo systemctl enable jupyterlab
```



Enfin, vérifiez le service jupyterlab pour vous assurer que le service est en cours d'exécution.

```
sudo systemctl status jupyterlab
```

Le résultat suivant confirme que le service jupyterlab est en cours d'exécution.



Configurer Nginx comme proxy inverse pour Jupyter

Dans ce guide, vous exécuterez JupyterLab avec Nginx comme proxy inverse. Pour y parvenir, vous devez effectuer les opérations suivantes :

- Autoriser l'accès à distance à JupyterLab
- Installation et configuration de Nginx en tant que proxy inverse

Autoriser l'accès à distance à JupyterLab

Par défaut, l'installation de JupyterLab n'est accessible que via une adresse IP locale. Pour autoriser la connexion à distance, vous devez modifier la configuration par défaut de JupyterLab.

Ouvrez la configuration de JupyterLab ~/.jupyter/jupyter_lab_config.py en utilisant la commande suivante de l'éditeur nano.

```
nano ~/.jupyter/jupyter_lab_config.py
```

Décommentez l'option c.ServerApp.allow_remote_access et modifiez la valeur sur True. Cela rendra JupyterLab accessible à partir d'une connexion à distance, ce qui inclut l'exécution de JupyterLab derrière un proxy inverse.

c.ServerApp.allow_remote_access = Vrai

Enregistrez le fichier et quittez l'éditeur lorsque vous avez terminé.

Maintenant, exécutez ce qui suit systèmectl commande pour redémarrer le service jupyterlab et appliquer les modifications.

```
sudo systemctl restart jupyterlab
```

Ensuite, vérifiez le service jupyterlab à l'aide de la commande suivante.

```
sudo systemctl status jupyterlab
```

Dans la sortie inférieure, vous devriez voir le jeton généré pour accéder à votre installation JupyterLab et assurez-vous de le copier pour la section suivante.

```
To access the server, open this file in a browser:
    file:///home/bob/.local/share/jupyter/runtime/jpserver-13839-open.html
Or copy and paste one of these URLs:
    http://localhost:8888/lab?token=3bbb64d9a83ea9c08972647df5d585ef512a6cbaa8e28740
    http://127.0.0.1:8888/lab?token=3bbb64d9a83ea9c08972647df5d585ef512a6cbaa8e28740
```

Installation et configuration de Nginx en tant que proxy inverse

Après avoir autorisé l'accès à distance à JupyterLab, vous allez ensuite installer Nginx et le configurer comme proxy inverse pour votre installation JupyterLab.

Installez Nginx en exécutant la commande apt install ci-dessous.

```
sudo apt install nginx -y
```

Vous trouverez ci-dessous le résultat lors de l'installation de Nginx.

```
root@debian12:~#
root@debian12:~# sudo apt install nginx -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  nginx-common
Suggested packages:
  fcgiwrap nginx-doc ssl-cert
The following NEW packages will be installed:
  nginx nginx-common
0 upgraded, 2 newly installed, 0 to remove and 30 not upgraded.
Need to get 640 kB of archives.
After this operation, 1,696 kB of additional disk space will be used.
```

Une fois Nginx installé, créez un nouvel éditeur nano de configuration de bloc serveur. /etc/nginx/sites-available/jupyterlab en utilisant ce qui suit

```
sudo nano /etc/nginx/sites-available/jupyterlab
```

Insérez la configuration suivante et assurez-vous de modifier le nom de domaine dans l' option server_name .

```
server {
        listen 80;
        server_name jupyterlab.hwdomain.io;
        access_log /var/log/nginx/hwdomain.io.access.log;  error_log /
       var/log/nginx/hwdomain.io.error.log;
        location / {
            proxy_pass http://127.0.0.1:8888;
            proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
           proxy_set_header X-Real-IP $remote_addr;
            proxy_set_header Host $http_host;
            proxy_http_version 1.1;
```

```
        proxy_redirect off;
        proxy_buffering off;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
        proxy_read_timeout 86400;

    }
}
```

Enregistrez et quittez le fichier lorsque vous avez terminé.

Exécutez maintenant la commande suivante pour activer la configuration Nginx du fichier de /etc/nginx/sites-available/jupyterlab et vérifier le bloc du serveur.

```
sudo ln -s /etc/nginx/sites-available/jupyterlab /etc/nginx/sites-enabled/
sudo nginx -t
```

Si vous disposez de la syntaxe Nginx appropriée, vous devriez obtenir le résultat "la syntaxe est correcte - le test est réussi".

Enfin, exécutez la commande systemctl ci-dessous pour redémarrer le service nginx et appliquer les modifications. Ensuite, vérifiez-le pour vous assurer que le service est en cours d'exécution.

```
sudo systemctl restart nginx
sudo systemctl status nginx
```

Si Nginx est en cours d'exécution, vous devriez obtenir la sortie active (en cours d'exécution).

Accéder à l'installation de JupyterLab via une machine locale

Pour accéder à JupyterLab via un nom de domaine, vous pouvez utiliser le /etc/hôtes fichier pour les clients Linux ou C:\Windows\System32\drivers\etc\hosts fichier pour les utilisateurs Windows.

Ouvrez le /etc/hôtes sur votre client Linux à l'aide de l'éditeur nano.

```
sudo nano /etc/hosts
```

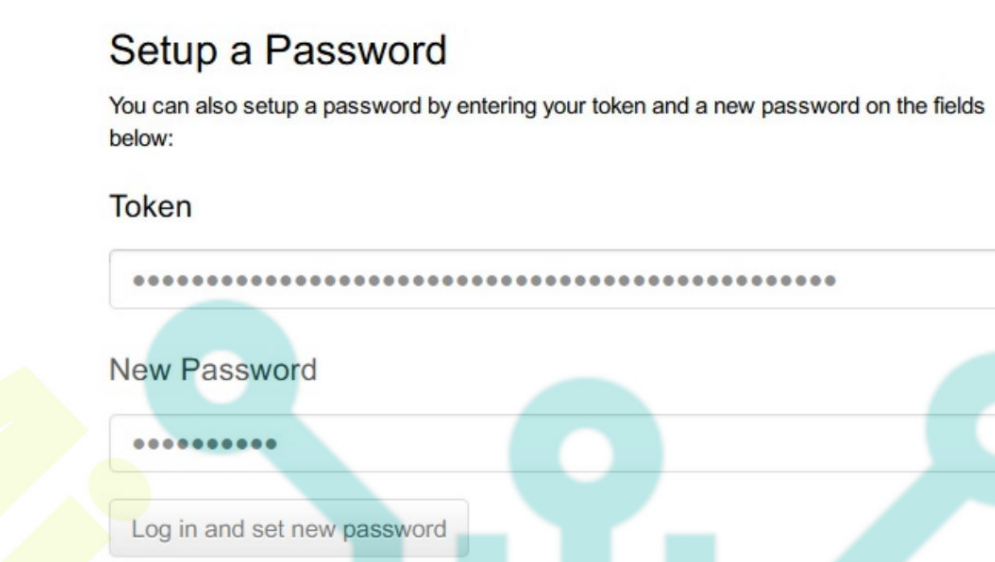
Insérez la configuration suivante dans le fichier et assurez-vous de modifier l'adresse IP et le nom de domaine avec votre information.

192.168.10.15 jupyterlab.hwdomain.io

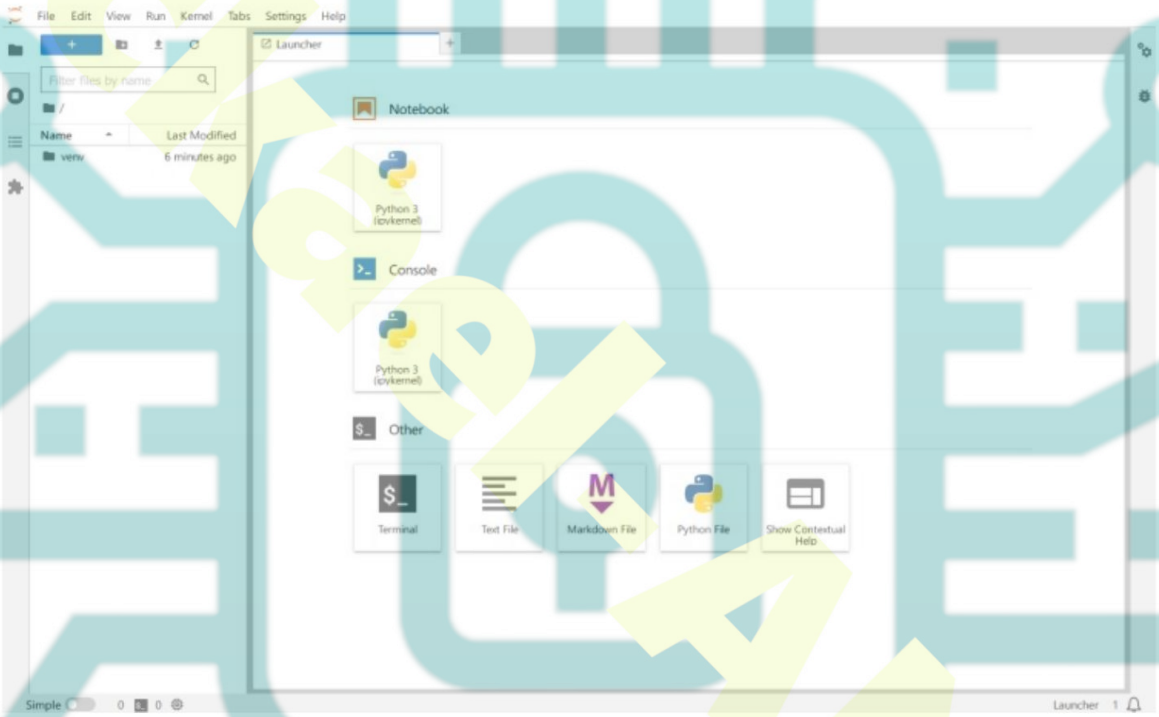
Enregistrez et quittez le fichier lorsque vous avez terminé.

Ensuite, ouvrez votre navigateur Web et visitez le nom de domaine de votre installation JupyterLab, tel que http://jupyterlab.hwdomain.io. Si votre installation réussit, vous devriez obtenir la page de connexion JupyterLab.

Sur la page inférieure, saisissez le mot de passe généré et le nouveau mot de passe pour votre installation JupyterLab. Cliquez ensuite sur le bouton Connectez-vous et définissez un nouveau mot de passe.



Si tout se passe bien, vous devriez être redirigé vers le tableau de bord JupyterLab comme suit.



Avec cela, votre installation JupyterLab est terminée.

Conclusion

En complétant ce guide étape par étape, vous avez terminé l'installation de JupyterLab sur le serveur Debian 12. Vous avez installé JupyterLab à l'aide de l'environnement virtuel Python, exécuté JupyterLab en arrière-plan en tant que service systemd et configuré Nginx comme proxy inverse pour JupyterLab. Vous pouvez désormais créer et gérer votre projet Jupyter.
