



CONTENTS

Conditions préalables

Installation d'Anaconda

Configuration des environnements d'Anaconda

Mettre à jour Anaconda

Désinstallation d'Anaconda

Conclusion

TUTORIAL

Comment installer la distribution Anaconda pour Python sur Ubuntu 20.04

Python Development Data Analysis Machine Learning Ubuntu 20.04



By Lisa Tagliaferri

Published on May 28, 2020 9.1k

Français

Introduction

Anaconda est un gestionnaire de paquets open-source, un gestionnaire d'environnement et une distribution des langages de programmation Python et R. Il est couramment utilisé pour la science des données, l'apprentissage machine, le traitement des données à grande échelle, le calcul scientifique et l'analyse prédictive.

Proposant une collection de plus de 1 000 packages de science des données, Anaconda est disponible en version entreprise gratuite et payante. La distribution d'Anaconda est livrée avec l'utilitaire de ligne de commande `conda`. Vous pouvez en apprendre davantage sur Anaconda et `conda` en lisant [la documentation officielle d'Anaconda](#).

Ce tutoriel vous guidera dans l'installation de la version Python 3 d'Anaconda sur un serveur Ubuntu 20.04.

Conditions préalables

Avant de commencer ce guide, vous devez avoir un non root user avec des priviléges sudo configurés sur votre serveur.

Vous pouvez remplir cette condition préalable en suivant notre [guide de configuration initiale du serveur Ubuntu 20.04](#).

Installation d'Anaconda

La meilleure façon d'installer Anaconda est de télécharger le dernier script bash de l'installateur Anaconda, de le vérifier, puis de l'exécuter.

Recherchez la dernière version d'Anaconda pour Python 3 sur la [page des téléchargements d'Anaconda](#). Au moment où nous écrivons ces lignes, la dernière version est la 2020.02, mais vous devriez utiliser une version stable ultérieure si elle est disponible.

Ensuite, allez dans le répertoire `/tmp` sur votre serveur. C'est un bon répertoire pour télécharger des éléments éphémères, comme le script Anaconda bash, dont nous n'aurons pas besoin après l'avoir exécuté.

```
$ cd /tmp
```

Utilisez `curl` pour télécharger le lien que vous avez copié du site web d'Anaconda. Nous le transférerons dans un fichier appelé `anaconda.sh` pour une utilisation plus rapide.

```
$ curl https://repo.anaconda.com/archive/Anaconda3-2020.02-Linux-x86_64.sh --output anaconda.sh
```

Nous pouvons maintenant vérifier l'intégrité des données de l'installateur grâce à la vérification du hachage cryptographique, par le biais de la somme de contrôle SHA-256. Nous utiliserons la commande `sha256sum`, ainsi que le nom de fichier du script :

```
$ sha256sum anaconda.sh
```

Vous obtiendrez un résultat qui ressemble à ceci :

```
Output  
2b9f088b2022edb474915d9f69a803d6449d5fdb4c303041f60ac4aefcc208bb anaconda.sh
```

Vous devez vérifier la sortie par rapport aux hachages disponibles sur la page [Anaconda avec Python 3 sur Linux 64 bits](#) pour votre version d'Anaconda. Tant que votre sortie correspond au hachage affiché dans la ligne `sha256`, tout est bon.

Nous pouvons maintenant exécuter le script :

```
$ bash anaconda.sh
```

Vous recevrez la sortie suivante :

```
Output  
  
Welcome to Anaconda3 2020.02  
  
In order to continue the installation process, please review the license  
agreement.  
Please, press ENTER to continue  
>>>
```

Appuyez sur `ENTER` pour continuer et appuyez ensuite sur `ENTER` pour lire la licence. Une fois que vous aurez fini de lire la licence, vous serez invité à approuver les termes de la licence :

```
Output  
Do you approve the license terms? [yes|no]
```

Si vous acceptez, tapez `yes`.

À ce stade, vous serez invité à choisir l'emplacement de l'installation. Vous pouvez appuyer sur `ENTER` pour accepter l'emplacement par défaut, ou spécifier un autre emplacement pour le modifier.

```
Output  
Anaconda3 will now be installed into this location:  
/home/sammy/anaconda3  
  
- Press ENTER to confirm the location  
- Press CTRL-C to abort the installation  
- Or specify a different location below  
  
[/home/sammy/anaconda3] >>>
```

Le processus d'installation se poursuivra. Notez qu'il peut prendre du temps.

Une fois l'installation terminée, vous recevrez le résultat suivant :

```
Output  
...  
Preparing transaction: done  
Executing transaction: done  
installation finished.  
Do you wish the installer to initialize Anaconda3  
by running conda init? [yes|no]  
[no] >>>
```

Tapez `oui` pour pouvoir initialiser Anaconda3. Vous recevrez un résultat qui indique les modifications apportées dans les différents répertoires. Une des lignes que vous recevrez vous remerciera d'avoir installé Anaconda.

```
Output  
...
```

```
Thank you for installing Anaconda3!
```

```
...
```

Vous pouvez maintenant activer l'installation en vous récupérant le fichier `~/.bashrc` :

```
$ source ~/.bashrc
```

Une fois que vous aurez fait cela, vous serez placé dans l'environnement de programmation de `base` par défaut d'Anaconda, et votre invite de commande changera comme suit :

```
(base) sammy@ubuntu:~$
```

Bien qu'Anaconda soit livré avec cet environnement de programmation de `base` par défaut, vous devez créer des environnements séparés pour vos programmes et les garder isolés les uns des autres.

Vous pouvez également vérifier votre installation en utilisant la commande `conda`, par exemple avec `list` :

```
(base) sammy@ubuntu:~$ conda list
```

Vous recevrez une sortie de tous les packages disponibles dans l'installation d'Anaconda :

```
Output
# packages in environment at /home/sammy/anaconda3:
#
#           Name          Version      Build Channel
_ipyw_jlab_nb_ext_conf    0.1.0        py37_0
_libgcc_mutex                0.1            main
alabaster                   0.7.12       py37_0
anaconda                    2020.02      py37_0
...
...
```

Maintenant qu'Anaconda est installé, nous pouvons passer à la configuration des environnements d'Anaconda.

Configuration des environnements d'Anaconda

Les environnements virtuels d'Anaconda vous permettent de garder les projets organisés par versions Python et les paquets nécessaires. Pour chaque environnement Anaconda que vous créez, vous pouvez spécifier la version de Python à utiliser et vous pouvez conserver tous vos fichiers de programmation connexes dans ce répertoire.

Tout d'abord, nous pouvons vérifier quelles versions de Python sont disponibles pour notre utilisation :

```
(base) sammy@ubuntu:~$ conda search "python$"
```

Vous recevrez une sortie avec les différentes versions de Python que vous pouvez cibler, y compris les versions Python 3 et Python 2. Puisque nous utilisons Anaconda avec Python 3 dans ce tutoriel, vous n'aurez accès qu'aux versions de Python 3 des packages.

Créons un environnement en utilisant la version la plus récente de Python 3. Nous pouvons le faire en assignant la version 3 à l'argument `python`. Nous appellerons l'environnement `my_env`, mais vous préférerez sans doute utiliser un nom plus explicite pour votre environnement, surtout si vous utilisez des environnements qui vous permettent d'accéder à plusieurs version de Python.

```
(base) sammy@ubuntu:~$ conda create --name my_env python=3
```

Nous recevrons une sortie avec des informations sur ce qui est téléchargé et sur les paquets qui seront installés, puis nous serons invités à poursuivre avec `y` ou `n`. Si vous êtes d'accord, tapez `y`.

L'utilitaire `conda` va maintenant récupérer les paquets pour l'environnement et vous faire savoir

quand ce sera terminé.

Vous pouvez activer votre nouvel environnement en tapant ce qui suit :

```
(base) sammy@ubuntu:~$ conda activate my_env
```

Une fois votre environnement activé, le préfixe de votre invite de commande indiquera que vous n'êtes plus dans l'environnement de `base`, mais dans le nouvel environnement que vous venez de créer.

```
(my_env) sammy@ubuntu:~$
```

Dans l'environnement, vous pouvez vérifier que vous utilisez la version de Python que vous aviez prévue pour utiliser :

```
(my_env) sammy@ubuntu:~$ python --version
```

Output

```
Python 3.8.2
```

Lorsque vous êtes prêt à désactiver votre environnement Anaconda, vous pouvez le faire en tapant :

```
(my_env) sammy@ubuntu:~$ conda deactivate
```

Notez que vous pouvez remplacer le mot `source` par `.` pour obtenir les mêmes résultats.

Pour cibler une version plus spécifique de Python, vous pouvez passer une version spécifique à l'argument `python`, comme `3.5`, par exemple :

```
(base) sammy@ubuntu:~$ conda create -n my_env35 python=3.5
```

Vous pouvez inspecter tous les environnements que vous avez configurés avec cette commande :

```
(base) sammy@ubuntu:~$ conda info --envs
```

Output

```
# conda environments:
#
base          * /home/sammy/anaconda3
my_env        /home/sammy/anaconda3/envs/my_env
my_env35      /home/sammy/anaconda3/envs/my_env35
```

L'astérisque indique l'environnement actif actuel.

Chaque environnement que vous créez avec `conda create` sera livré avec plusieurs paquets par défaut :

- `_libgcc_mutex`
- `ca-certificates`
- `certifi`
- `libedit`
- `libffi`
- `libgcc-ng`
- `libstdcxx-ng`
- `nurses`
- `openssl`
- `pip`

- `python`
- `readline`
- `setuptools`
- `sqlite`
- `tk`
- `wheel`
- `xz`
- `zlib`

Vous pouvez ajouter des paquets supplémentaires, tels que `numpy` par exemple, avec la commande suivante :

```
(base) sammy@ubuntu:~$ conda install --name my_env35 numpy
```

Si vous savez, vous souhaitez un environnement `numpy` lors de la création, vous pouvez le cibler dans votre commande `conda create` :

```
(base) sammy@ubuntu:~$ conda create --name my_env python=3 numpy
```

Si vous ne travaillez plus sur un projet spécifique et si vous n'avez plus besoin de l'environnement associé, vous pouvez le supprimer. Pour ce faire, tapez ce qui suit :

```
(base) sammy@ubuntu:~$ conda remove --name my_env35 --all
```

Maintenant, lorsque vous tapez la commande `conda info --envs`, l'environnement que vous avez supprimé ne sera plus repertorié.

Mettre à jour Anaconda

Vous devriez régulièrement vous assurer qu'Anaconda est à jour afin de travailler avec toutes les dernières versions du package.

Pour ce faire, vous devez d'abord mettre à jour l'utilitaire `conda` :

```
(base) sammy@ubuntu:~$ conda update conda
```

Lorsque vous y êtes invité, tapez `y` pour procéder à la mise à jour.

Une fois que la mise à jour de `conda` est terminée, vous pouvez mettre à jour la distribution Anaconda :

```
(base) sammy@ubuntu:~$ conda update anaconda
```

Une fois encore, lorsque vous y êtes invité, tapez `y` pour continuer.

Cela vous garantira que vous utilisez les dernières versions de `conda` et d'Anaconda.

Désinstallation d'Anaconda

Si vous n'utilisez plus Anaconda et que vous devez le désinstaller, vous devez commencer par le module `anaconda-clean`, qui supprimera les fichiers de configuration pour la désinstallation d'Anaconda.

```
$ conda install anaconda-clean
```

Tapez `y` lorsque vous y êtes invité.

Une fois installé, vous pouvez exécuter la commande suivante. Vous serez invité à répondre à `y` avant de supprimer chacun d'entre eux. Si vous préférez ne pas être invité à répondre, ajoutez

--yes à la fin de votre commande :

```
$ anaconda-clean
```

Cela créera également un dossier de sauvegarde appelé `.anaconda_backup` dans votre répertoire d'origine :

```
Output  
Backup directory: /home/sammy/.anaconda_backup/2020-05-06T024432
```

Vous pouvez maintenant supprimer tout votre répertoire Anaconda en entrant la commande suivante :

```
$ rm -rf ~/anaconda3
```

Pour finir, vous pouvez supprimer la ligne PATH de votre fichier `.bashrc` qu'Anaconda a ajouté.

Pour ce faire, ouvrez d'abord un éditeur de texte comme nano :

```
$ nano ~/.bashrc
```

Ensuite, faites défiler vers le bas jusqu'à la fin du fichier (s'il s'agit d'une installation récente) ou tapez `CTRL + W` pour rechercher Anaconda. Supprimez ou commentez ce bloc Anaconda :

```
/home/sammy/.bashrc  
  
...  
# >>> conda initialize >>>  
# !! Contents within this block are managed by 'conda init' !!  
__conda_setup="$('/home/sammy/anaconda3/bin/conda' 'shell.bash' 'hook' 2> /dev/null)"  
if [ $? -eq 0 ]; then  
    eval "$__conda_setup"  
else  
    if [ -f "/home/sammy/anaconda3/etc/profile.d/conda.sh" ]; then  
        . "/home/sammy/anaconda3/etc/profile.d/conda.sh"  
    else  
        export PATH="/home/sammy/anaconda3/bin:$PATH"  
    fi  
fi  
unset __conda_setup  
# <<< conda initialize <<<
```

Lorsque vous avez fini de modifier le fichier, tapez `CTRL + X` pour quitter et `y` pour enregistrer les modifications.

Anaconda est maintenant retiré de votre serveur. Si vous n'avez pas désactivé l'environnement de programmation de `base`, vous pouvez quitter le serveur et y revenir pour le supprimer.

Conclusion

Ce tutoriel vous a guidé à travers l'installation d'Anaconda, le travail avec l'utilitaire en ligne de commande `conda`, la mise en place d'environnements, la mise à jour d'Anaconda et la suppression d'Anaconda si vous n'en avez plus besoin.

Vous pouvez utiliser Anaconda pour vous aider à gérer les charges de travail pour la science des données, le calcul scientifique, l'analyse et le traitement des données à grande échelle. De là, vous pouvez consulter nos tutoriels sur [l'analyse des données](#) et [l'apprentissage automatique](#) pour en savoir plus sur les différents outils disponibles et les projets que vous pouvez mener à bien.

Nous avons également un livre électronique gratuit sur [l'apprentissage automatique](#), [Python Machine Learning Projects](#), que vous pouvez télécharger.

Qu'avez-vous pensé de cette traduction?



Was this helpful?

Yes

No



[Report an issue](#)

About the authors



Lisa Tagliaferri

Community and Developer Education

Still looking for an answer?



Ask a question



Search for more help

RELATED

Join the DigitalOcean Community



Join 1M+ other developers and:

- Get help and share knowledge in Q&A
- Subscribe to topics of interest
- Get courses & tools that help you grow as a developer or small business owner

[Join Now](#)

[Comment installer MongoDB à partir des référentiels APT par défaut sur Ubuntu 20.04](#)

[Tutorial](#)

[Comment importer et exporter une base de données MongoDB sur Ubuntu 20.04](#)

[Tutorial](#)

Comments

0 Comments

Leave a comment...

[Sign In to Comment](#)



This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.



GET OUR BIWEEKLY NEWSLETTER

Sign up for Infrastructure as a Newsletter.



HOLLIE'S HUB FOR GOOD

Working on improving health and education, reducing inequality, and spurring economic growth? We'd like to help.



BECOME A CONTRIBUTOR

You get paid; we donate to tech nonprofits.

Featured on Community Kubernetes Course Learn Python 3 Machine Learning in Python Getting started with Go Intro to Kubernetes

DigitalOcean Products Virtual Machines Managed Databases Managed Kubernetes Block Storage Object Storage Marketplace VPC Load Balancers

Welcome to the developer cloud

DigitalOcean makes it simple to launch in the cloud and scale up as you grow – whether you're running one virtual machine or ten thousand.

[Learn More](#)

© 2021 DigitalOcean, LLC. All rights reserved.

Company	Products	Community	Contact
About	Pricing	Tutorials	Get Support
Leadership	Products Overview	Q&A	Trouble Signing In?
Blog	Droplets	Tools and Integrations	Sales
Careers	Kubernetes	Tags	Report Abuse
Partners	Managed Databases	Product Ideas	System Status
Referral Program	Spaces	Write for DigitalOcean	
Press	Marketplace	Presentation Grants	
Legal	Load Balancers	Hatch Startup Program	
Security & Trust Center	Block Storage	Shop Swag	

Sign up for our newsletter X

Get the latest tutorials on SysAdmin and open source topics.

Enter your email address

Sign Up

By continuing to use our website, you agree to our [Terms of Service](#), [Privacy Policy](#) and [Cookie and Tracking Notice](#). By continuing to browse our website, you agree to our use of cookies.

I understand

SCROLL TO TOP