

AdminGuide

Администрирование для самых малen

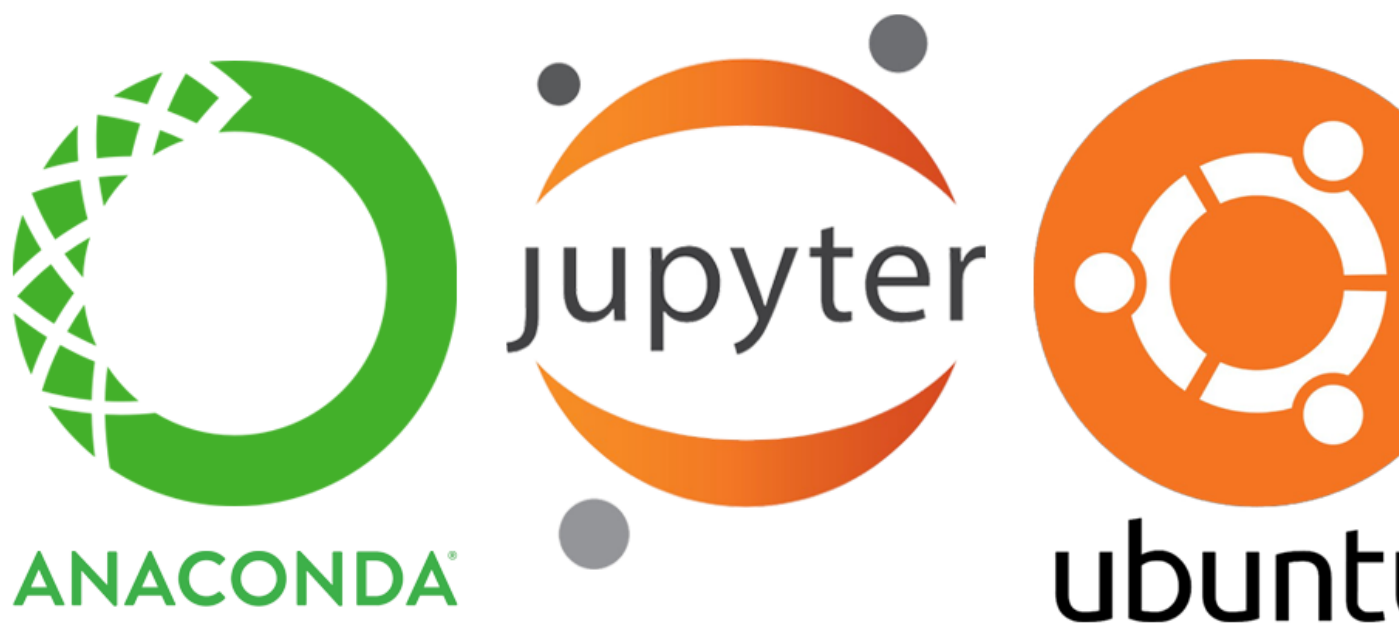
Главная страница » Jupyter Notebook и Anaconda. Удаленная разработка на Ubuntu Server

[ANACONDA](#) ♦ [DEVELOPMENT](#) ♦ [JUPYTER NOTEBOOK](#) ♦ [UBUNTU](#) ♦ [UBUNTU SERVER](#)

Jupyter Notebook и Anaconda. Удаленная разработка на Ubuntu Server

written by Belfigor | 05.08.2019 | 2223 views





0 comment

Jupyter Notebook и Anaconda для удаленной разработки на отдельном Ubuntu Server, нужны для облегчения себе жизни. А так же чтобы не таскать с места на место свои проектики или не работать с jupyter notebook через удаленный рабочий стол. Еще распространена ситуация, когда у вас дома, в офисе, на даче или датацентре стоит мощная вычислительная нода, многопроцессорная, с профильными для машинного



Можно конечно всё поставить на масдай, но 1) нестабильно, 2) в случае работы в виртуальной среде нарушает лицензию использования масдая. Ведь 99.9% желающих захотят запилить в виртуальную среду какую-нибудь обычную прошку да? А она даже с лицензионным валидным ключем – будет нарушать её. Собственно зачем жертвовать стабильностью работы, и нарушать лицензию, если этого можно избежать?

В данной статье не рассматриваются вопросы относительно проброса трафика из интернета до вашей машины. Подразумевается что этот вопрос у вас уже решен и ваш трафик из интернета спокойно доходит до вашей машины, хотя бы по одному TCP порту.

1. Что мы имеем?

Некий сервер, внутри нашей локальной сети, куда мы хотим запилить Ubuntu Server, на неё установить Anaconda и использовать идущей в комплекте Jupyter Notebook. Почему не голый Jupyter Notebook? Потому что у анаконды в комплекте куча еще всяких прочих плюшек.

2. Что нам надо?

Собственно только сервер (В моём случае виртуальная машина внутри бесплатного гипервизора). Чем мощнее тем лучше, но всё зависит от того чем именно вы занимаетесь в Jupyter Notebook. Я буду использовать виртмашину где:

1. 4 vCPU от процессора Intel
2. 8Gb Ram
3. 16Gb Hdd
4. IP 192.168.220.135

Вы свои параметры должны определить для себя самостоятельно

С чего стоит начать:

- [Установка Ubuntu Server 18.04](#)

Что может пригодиться:

- [Ubuntu Server 18.04 — Установка статического IP](#)
- [Ubuntu Server 18.04 — Изменение имени сервера](#)

1. Обновляем сервер для Jupyter Notebook

```
1. sudo apt update && sudo apt dist-upgrade -y
```

Raw

Copy

Extern

 liq

```
1. wget https://repo.anaconda.com/archive/Anaconda3-2019.07-Linux-x86_64.sh
```

Raw Copy Extern Enli

3. Проверяем хеш скачанного файла

Открываем [страницу с хешами](#), выбираем тип дистрибутива который качали, и там смотрим конкретную версию файла

Hashes for Anaconda3-2019.07-Linux-x86_64.sh

All Installer files are available at <https://repo.anaconda.com/archive/>.

You can verify the data integrity of the Anaconda Installer files by [running a local program to generate their MD5 or SHA-256 cryptographic hashes](#) and checking the output to be sure it matches the hashes (or “checksums”) below.

If the MD5 or SHA-256 hash that you generate does not match the one here, the file may not have downloaded completely. Please download it again and re-check. If repeated downloads produce the same result, please [contact us](#) to report the problem, including the file name, whether you used MD5 or SHA-256, the hash you generated, and the hash on the site.

exact time file was last modified, as Unix time stamp	1564065380.5078816
time file was last modified, in human readable format	2019-07-25 09:36:20
exact file size, in bytes	541906131
file size, in human friendly format	516.8 MiB
md5	ec6a6bf96d75274c2176223e8584d2da

Проверяем хеш на сайте

```

ux-x86_64.sh
Resolving repo.anaconda.com (repo.anaconda.com)... 104.16.131.3, 104.16.130.3, 2
606:4700::6810:8203, ...
Connecting to repo.anaconda.com (repo.anaconda.com)|104.16.131.3|:443... connect
ed.
HTTP request sent, awaiting response... 200 OK
Length: 541906131 (517M) [application/x-sh]
Saving to: 'Anaconda3-2019.07-Linux-x86_64.sh'

Anaconda3-2019.07-L 100%[=====>] 516.80M  1.14MB/s   in 7m 32s

2019-08-02 18:01:21 (1.14 MB/s) - 'Anaconda3-2019.07-Linux-x86_64.sh' saved [541
906131/541906131]

```

Проверяем хеш скачанного файла

В моём случае я скачивал в корень своей папки, поэтому у меня команда будет выглядеть так:

```
1. md5sum ~/Anaconda3-2019.07-Linux-x86_64.sh
```

Raw Copy Extern Enli

4. Устанавливаем Anaconda

Вводим команду `bash ~/Anaconda3-2019.07-Linux-x86_64.sh`, принимаем лицензионное соглашение введя `yes` и ожидаем окончания установки

В конце инсталлятор спросит

```

belfigor@adminguide-ru: ~
wrappt      pkgs/main/linux-64::wrappt-1.11.2-py37h7b6447c_0
wurlitzer   pkgs/main/linux-64::wurlitzer-1.0.2-py37_0
xlrd         pkgs/main/linux-64::xlrd-1.2.0-py37_0
xlsxwriter  pkgs/main/noarch::xlsxwriter-1.1.8-py_0
xlwt        pkgs/main/linux-64::xlwt-1.3.0-py37_0
xz           pkgs/main/linux-64::xz-5.2.4-h14c3975_4
yaml         pkgs/main/linux-64::yaml-0.1.7-had09818_2
zeromq       pkgs/main/linux-64::zeromq-4.3.1-he6710b0_3
zict         pkgs/main/noarch::zict-1.0.0-py_0
zipp         pkgs/main/noarch::zipp-0.5.1-py_0
zlib         pkgs/main/linux-64::zlib-1.2.11-h7b6447c_3
zstd         pkgs/main/linux-64::zstd-1.3.7-h0b5b093_0

Preparing transaction: done
Executing transaction: - WARNING conda.core.envs_manager:register_env(46): Unable to re
environment. Path not writable or missing.
environment location: /home/belfigor/anaconda3
registry file: /home/belfigor/.conda/environments.txt
ne
installation finished.
Do you wish the installer to initialize Anaconda?

```

Anaconda, Jupyter Notebook для удаленной разработки на Ubuntu Server – Завершение у

`exit` или просто перезагрузить сервер с помощью `sudo reboot -h now`

5. Настраиваем Jupyter Notebook

1. Генерируем конфиг файл

```
1. jupyter notebook --generate-config
```

[Raw](#) [Copy](#) [Extern](#) [Enliq](#)

Эта команда сгенерирует конфиг файл, в моём случае по следующему пути:

```
/home/belfigor/.jupyter/jupyter_notebook_config.py
```

2. Генерируем ssl сертификат

```
1. openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout mykey.key
```

[Raw](#) [Copy](#) [Extern](#) [Enliq](#)

Данная команда сгенерирует файлы сертификата в той же папке, которая была открыта в мом запуска. В моём случае это домашняя моя домашняя папка `~/`, полный путь выглядит как

```
/home/belfigor
```

3. Изменяем конфигурацию Jupyter Notebook

Открываем на редактирование наш конфиг и ищем указанные ниже параметры

```
1. nano /home/belfigor/.jupyter/jupyter_notebook_config.py
```

[Raw](#) [Copy](#) [Extern](#) [Enliq](#)

Искать параметры в nano можно с помощью комбинации клавишь `Ctrl+W`. Сохранить изменен помощью `Ctrl+O`, а завершить редактирование с помощью `Ctrl+X`.

1. Находим следующие строки и прописываем там путь до файлов сертификатов

```
1. c.NotebookApp.keyfile = '/home/belfigor/mykey.key'
2. c.NotebookApp.certfile = '/home/belfigor/mycert.pem'
```

[Raw](#) [Copy](#) [Extern](#) [Enliq](#)

```
1. c.NotebookApp.ip = '192.168.220.135'
```

Raw

Copy

Extern

Enliq

Я указываю ip интерфейса, которым сервер смотрит в локальную сеть, откуда из интернета прокинут порт 35344, для доступа из вне.

Если у сервера больше одного интерфейса и мы хотим чтобы Jupyter обрабатывал входящие подключения со всех, можно указать “*”

3. Задаем порт на котором будут обрабатываться подключения

```
1. c.NotebookApp.port = 35344
```

Raw

Copy

Extern

Enliq

4. Отключаем автозапуск браузера

Так как у нас Ubuntu Server, обладающий лишь текстовой консолью, нам нет необходимости каждый раз пытаться запустить браузер.

```
1. c.NotebookApp.open_browser = False
```

Raw

Copy

Extern

Enliq

5. Дополнительные настройки

Так же внутри конфиг файла есть огромная куча настроек, ненужных для личного использования которые могут оказаться полезными при более глубоком изучении инфраструктуры Jupyter построении какой-то “своей особенной не такой как у других” инфраструктуры 😊

4. Задаем пароль от веб интерфейса

```
1. jupyter notebook password
```

Raw

Copy

Extern

Enliq

Этой же командой его можно сбросить если он забыт.

5. Запускаем Jupyter Notebook

Т.к. в пункте 5.2 мы сгенерировали файла сертификата, а в пункте 5.3 указали путь к ним, чтобы запустить Jupyter Notebook с подключением по https, нам достаточно воспользоваться командой

```
1. jupyter notebook
```

Raw

Copy

Extern

Enliq



1. `jupyter notebook --certfile=mycert.pem --keyfile=mykey.key` Raw Copy Extern Enliq

6. Подключаемся к Jupyter Notebook

После запуска, в консоли мы увидим на каком ip и порту стартовал сервер:

```

1. (base) belfigor@adminguide-ru:~$ jupyter notebook
2. [I 10:19:50.263 NotebookApp] JupyterLab extension loaded from /home/k
3. [I 10:19:50.263 NotebookApp] JupyterLab application directory is /hor
4. [I 10:19:50.265 NotebookApp] Serving notebooks from local directory:
5. [I 10:19:50.265 NotebookApp] The Jupyter Notebook is running at:
6. [I 10:19:50.265 NotebookApp] https://192.168.220.135:35344/
7. [I 10:19:50.266 NotebookApp] Use Control-C to stop this server and sh

```

Нам осталось только открыть браузер и подключиться по указанным реквизитам

`https://192.168.220.135:35344/` с другой машины внутри локальной сети и ввести пароль заданный в пункте 5.4 и уже можно приступать к работе.

Собственно если не терпится начать что-то записывать, то можно всё бросать и набигать на jupyter notebook. Но если же есть еще лишние 5 минут, предлагаю сделать свою жизнь чуть лучше ус менеджер расширений для jupyter.

6. Устанавливаем Nbextensions

Если Jupyter Notebook сервер работает, завершаем его работу с помощью Ctrl+C => y и последо вводим команды:

```

1. conda install -c conda-forge jupyter_contrib_nbextensions
2. conda install -c conda-forge jupyter_nbextensions_configurator
3. jupyter contrib nbextension install --user

```

Когда запрашивает подтверждение, вводим [y]

По окончании установки, зайдя в Jupyter Notebook мы увидим следующее:



[Files](#)
[Running](#)
[Clusters](#)
[Nbextensions](#)

Configurable nbextensions

☐ disable configuration for nbextensions without explicit compatibility (they may break your notebook environment, but can be useful to show for nbextension development)

filter: by description, section, or tags

- | | | |
|--|---|--|
| <input type="checkbox"/> (some) LaTeX environments for Jupyter | <input type="checkbox"/> 2to3 Converter | <input type="checkbox"/> AddBefore |
| <input type="checkbox"/> Autopep8 | <input type="checkbox"/> AutoSaveTime | <input type="checkbox"/> Autoscroll |
| <input type="checkbox"/> Cell Filter | <input type="checkbox"/> Code Font Size | <input type="checkbox"/> Code prettify |
| <input type="checkbox"/> Codefolding | <input type="checkbox"/> Codefolding in Editor | <input type="checkbox"/> CodeMirror mode extensions |
| <input type="checkbox"/> Collapsible Headings | <input type="checkbox"/> Comment/Uncomment Hotkey | <input checked="" type="checkbox"/> contrib_nbextensions_help_item |
| <input type="checkbox"/> daterstamp | <input type="checkbox"/> Equation Auto Numbering | <input type="checkbox"/> ExecuteTime |
| <input type="checkbox"/> Execution Dependencies | <input type="checkbox"/> Exercise | <input type="checkbox"/> Exercise2 |
| <input type="checkbox"/> Export Embedded HTML | <input type="checkbox"/> Freeze | <input type="checkbox"/> Gist-it |
| <input type="checkbox"/> Help panel | <input type="checkbox"/> Hide Header | <input type="checkbox"/> Hide input |
| <input type="checkbox"/> Hide input all | <input type="checkbox"/> Highlight selected word | <input type="checkbox"/> highlighter |
| <input type="checkbox"/> Hinterland | <input type="checkbox"/> Initialization cells | <input type="checkbox"/> isort formatter |
| <input checked="" type="checkbox"/> jupyter-js-widgets/extension | <input type="checkbox"/> Keyboard shortcut editor | <input type="checkbox"/> Launch QTConsole |
| <input type="checkbox"/> Limit Output | <input type="checkbox"/> Live Markdown Preview | <input type="checkbox"/> Load TeX macros |
| <input type="checkbox"/> Move selected cells | <input type="checkbox"/> Navigation-Hotkeys | <input checked="" type="checkbox"/> Nbextensions dashboard tab |
| <input checked="" type="checkbox"/> Nbextensions edit menu item | <input type="checkbox"/> nbTranslate | <input type="checkbox"/> Notify |
| <input type="checkbox"/> Printview | <input type="checkbox"/> Python Markdown | <input type="checkbox"/> Rubberband |
| <input type="checkbox"/> Ruler | <input type="checkbox"/> Ruler in Editor | <input type="checkbox"/> Runtools |
| <input type="checkbox"/> Scratchpad | <input type="checkbox"/> ScrollDown | <input type="checkbox"/> Select CodeMirror Keymap |
| <input type="checkbox"/> SKILL Syntax | <input type="checkbox"/> Skip-Traceback | <input type="checkbox"/> Snippets |
| <input type="checkbox"/> Snippets Menu | <input type="checkbox"/> spellchecker | <input type="checkbox"/> Split Cells Notebook |

Менеджер расширений Nbextensions

Скорее всего галочка disable configuration for... у вас стоит, снимаем её.

Теперь мы можем ставить представленные в списке расширения. Моё самое любимое – Table of (2). Оно формирует из Markdown ячеек содержание, которое удобно использовать для быстрой на по проекту, когда счёт ячеек идёт на сотни.

TEXT.RU

ANACONDA

JUPYTER NOTEBOOK

MACHINE LEARNING

UBUNTU

UBUNTU SERVER

0 comment

BELFIGOR



[previous post](#)**PostgreSQL 10 – Поточная репликация с
отработкой фейловера**[next post](#)**Zabbix Helper Kit и автоматизация
обновления Zabbix Agent**

YOU MAY ALSO LIKE

**Контроллер домена Linux –
Видеоурок**

24.05.2020

**Сервер 1С 8.3 на Ubuntu 20.04.
Установка**

23.05.2020

**Контроллер домена Ubuntu
– Ubuntu 20.04 AD-D**

10.05.2020

ДОБАВИТЬ КОММЕНТАРИЙ

[f FACEBOOK](#)[t TWITTER](#)[p PINTEREST](#)[t TUMBLR](#)[v YOUTUBE](#)

