

Урок 5. Обзор возможностей Airflow, установка и настройка

1. Установите AirFlow как показано на семинаре, соберите список команд установки, скриншоты GUI и работы AirFlow из мобы. Соберите все данные в один pdf файл.
2. Проверить гипотезу будет ли достаточно установки `sudo apt install -y python3 python3-pip python3-venv` без `sudo apt install python3.8`
3. Создайте мануал со списком команд для установки.

!!!ВАЖНО!!! Все команды из PowerShell выполняются с правами Администратора, ваша учётная запись тоже должна быть с правами администратора, у вас должен быть обеспечен доступ в интернет.

Все действия выполнялись под ЛИЦЕНЗИОННУЮ операционную систему Windows 10 Pro, разрядность x64, инструкция написана под версию x64, никакой совместимости с 32x не гарантирую, Windows 10 Home не поддерживает WSL.

Так же не гарантируется работоспособность на Windows 10 Repack by Васяна, выполняйте действия на свой страх и риск, поскольку репаки by Васян часто любят чудить с слётом регистрации и установкой компонентов, когда пытаешься сделать, что-то за гранью обычного пользователя.

Это же касается Linux, работоспособность репаков Linux от Васяна на wsl не гарантируется, благо пингвины бесплатны, нужно лишь поставить НОРМАЛЬНЫЙ Linux(к примеру Ubuntu) без всякого гурманства типа Astra Linux который впервые очередь написан для Товарища Майора.

1. Для начала установки AirFlow на платформу Windows 10 нужно включить встроенную виртуальную машину Linux.

1.1 Вы можете воспользоваться официальной инструкцией от Microsoft доступной по данным ссылкам:

<https://learn.microsoft.com/ru-ru/windows/wsl/install> - в случае если данная инструкция выводит ошибки то нужно воспользоваться более старой версией данной инструкции, про неё так же написано в данной инструкции но на крайний случай ссылку также прилагаю.

<https://learn.microsoft.com/ru-ru/windows/wsl/install-manual> - как написано более старая инструкция.

<https://learn.microsoft.com/ru-ru/windows/wsl/basic-commands#install> - список основных команд инструкция.

Краткая инструкция от меня для кому лень читать тонны текста от Microsoft.

1.2 Увы, нужно обновить Windows 10 до последней версии, в которой полно рекламы, очень нужный помощник Сири и куча другой нужной лабуды и шлака(да обновления безопасности процессоров, я о вас, обычному пользователю вы нахер не нужны, пожиратели производительности), но таков путь!

1.2.1 Для начала вам надо зайти в биос (кнопка зависит от вашего производителя материнской платы вот некоторые из них: F2, F10, F12, DEL) и включить виртуализацию вашего процессора, в случае если процессор не поддерживает виртуализацию то установка виртуальной машины не возможна.

1.2.2 В системе Windows 10 Зайти в "Панель управления" -> "Программы" или "Программы и компоненты" -> "Включение и отключение компонентов Windows".

1.2.3 Далее вам нужно либо включить либо удостовериться что включены следующие компоненты (поставьте галочки):

- Платформа для виртуальной машины;
- Подсистема Windows для Linux;

1.2.4 Перезагрузите компьютер.

1.2.5 Откройте PowerShell и введите следующую команду:

- `wsl --set-default-version 2` (Данной командой мы обозначим запуск второй версии wsl которая нам нужна)

1.2.6 Если в консоле не будет ошибки перезагрузите компьютер и далее перейдите к пункту запуска и настройки Linux.

В случае той или иной ошибки в консоле то Google вам в помощь и инструкция от Microsoft.

1.2.7 Список команд для работы с wsl:

- `wsl install` автоматическая установка wsl (лично у меня не сработало пришлось делать ручками)
- `wsl --update` обновление wsl (скачивание новой версии программы)
- `wsl --version` проверить версию установленной и используемой wsl
- `wsl -l` список установленных дистрибутивов Linux
- `wsl --unregister <name>` удаление дистрибутива Linux из wsl из списка установленных пишите имя вашего дистрибутива за место <name>
- `wsl --list --running` лист запущенных Linux на текущий момент в среде wsl
- `wsl -t <name>` команда для выключения Linux который запущен название можно взять с листа и вписать в за место поля <name>
- `wsl --shutdown` команда которая завершает все виртуальные машины wsl
- `Restart-Service LxssManager` для перезапуска службы Диспетчера подсистемы Windows для Linux (LxssManager), может потребоваться для решения проблем с WSL или виртуальными машинами Linux, а также для применения обновлений или изменений конфигурации.

1.2.8 Здесь я напишу ошибки с которыми, я встречался после выполнения всех действий выше, т.е они возникнут даже если вы все сделали правильно по моей инструкции:

Ошибка:

`WslRegisterDistribution failed with error: 0x800701bc ????????`
`????????` (Вопросики это кириллица или другой язык которую не может отобразить консоль, вопросиков будет много я не стал их полностью копировать)

Лечение:

Я смог обойти с помощью команды в PowerShell: `wsl --update`

Ниже оригинал с английской версии сайта:

"I typed `wsl --update` and the new version of wsl is installed. Then I installed Ubuntu by `wsl --install -d ubuntu` and it worked."

Ошибка:

Вы вдруг такой решили версию wsl, ввели команду `wsl --version` а консоль напишет, что такой команды нет.

Лечение:

Не переживайте если команда `wsl --set-default-version 2` прошла успешно то у вас всё хорошо и Linux должен запуститься после настройки и перезагрузки а после инсталляции Linux команда `wsl --version` у вас заработает.

Ошибка:

Error code: Wsl/Service/CreateInstance/MountVhd/ERROR_FILE_NOT_FOUND

Лечение:

У меня это вылезло когда я захотел переставить свой Linux лечиться командой `wsl -l` затем командой `wsl --unregister Ubuntu` всё можете приступать к работе с чистого листа.

Ниже оригинал с английской версии сайта:

List the distributions installed, by running following in PowerShell.

`wsl -l`

Unregister the distribution. Replace the "Ubuntu" below with your distribution name found in Step #1:

NOTE: THIS COMMAND WILL COMPLETELY UNINSTALL YOUR WSL DISTRO.

`wsl --unregister Ubuntu`

Launch the Ubuntu (or other distribution) which was installed using Microsoft Store

2. Клиент Linux можно установить на вашу wsl несколькими способами:

2.1 Заходим в магазин Microsoft Store, пишем в поиске Linux, выбираем который понравился и нажимаем скачать, после скачивания и установки, нажимаем запустить, начётся процесс инициализации и установки Linux куда мы вводим свой логин и пароль.

2.2 Скачиваем дистрибутив с официального сайта Linux который вам понравился, распаковываем и запускаем из папки, начнётся процесс инициализации и установки Linux куда мы вводим свой логин и пароль.

2.3 Так же образ можно скачать с помощью самой консоли PowerShell для этого:

2.3.1 `cd C:\` с помощью данной команды перейдите в каталог в котором вы хотите создать папку

2.3.2 `mkdir <name>` с помощью данной команды создайте папку, название папки за место <name>

2.3.3 `Invoke-WebRequest -Uri https://aka.ms/wslubuntu2004 -OutFile Ubuntu.appx -UseBasicParsing` для загрузки образа Linux предварительно не забудьте перейти в созданную папку с помощью команды `cd`

- Uri: Этот параметр указывает URL-адрес, с которого загружается файл.

- OutFile: Этот параметр указывает путь и имя файла, в который будет сохранен загруженный файл.

- UseBasicParsing: Этот параметр указывает PowerShell не использовать Internet Explorer для анализа веб-страницы при загрузке.

2.3.4 `move .\Ubuntu.appx .\Ubuntu.zip` данная команда используется для того, чтобы переименовать файл в `.zip`

2.3.5 `Expand-Archive .\Ubuntu.zip` данная команда нужна для распаковки `.zip` в папку.

3. Теперь обновим Linux, добавим репозиторий DeadSnakes PPA (Personal Package Archive), что позволит нам устанавливать различные версии Python, используя "apt" или "apt-get" и установим за одним Python

3.1 `sudo apt update` получаем пакеты обновлений для Linux

3.2 `sudo add-apt-repository ppa:deadsnakes/ppa` подключаем нужный нам репозиторий

3.3 `sudo apt install -y python3 python3-pip python3-venv` команда устанавливает Python, pip и venv последней версии доступной в подключённом репозитории.

- `python3`: Это интерпретатор языка программирования Python версии 3.x. Он позволяет запускать и выполнять Python-скрипты и приложения.

- `python3-pip`: Это инструмент управления пакетами для Python. Он используется для установки, обновления и управления сторонними библиотеками и модулями Python.

- `python3-venv`: Этот пакет предоставляет виртуальное окружение Python, которое позволяет изолировать проекты Python друг от друга. Виртуальные окружения полезны для управления зависимостями проекта и предотвращения конфликтов между версиями пакетов.

- `sudo apt install -y python3.8 python3.8-pip python3.8-venv` пример команды если нам нужна конкретная версия Python, в данном случае Python 3.8

3.4 `sudo apt update` получаем пакеты обновлений для Linux

3.5 `sudo apt upgrade` обновляем все до последних версий

4. Далее мы устанавливаем Mysql server и настроим его, для простоты понимания логов лучше запустить через программу MobaXterm_Personal, однако если у вас нет проблем ориентирования в командной строке можно продолжить работу в ней:

4.1 `sudo apt-get install mysql-server` устанавливаем Mysql server

4.2 `sudo /etc/init.d/mysql start` запускаем Mysql

4.3 `sudo mysql_secure_installation` теперь запускаем настройку сервиса везде ставим нет кроме последнего пункта:

- `VALIDATE PASSWORD COMPONENT...` ставим No

- `By default, a MySQL installation has an anonymous user...` ставим No

- `Normally, root should only be allowed to connect from`

`'localhost'...` ставим No

- `By default, MySQL comes with a database named 'test' that...` ставим No

- `Reloading the privilege tables will ensure that all changes made so far will take effect immediately anyone can access...` ставим Yes

4.5 `sudo mysql -u root -p` заходим в программу Mysql, если спросит пароль нажимаем Enter далее вводим данную строку:

```
ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY '1';
```

строка выше установит пароль "1", затем выходим командой exit

4.6 `sudo nano /etc/mysql/my.cnf` данный файл нужно отредактировать добавив в конец все строки:

```
[mysqld]
port = 33061
```

сохранить изменения в файле

4.7 `sudo service mysql restart` перезапускаем Mysql сервер

4.8 Теперь вам надо зайти в MySQL оболочку, которая у вас установлена на Windows 10 это может быть HeidiSQL или любая другая программа, вам нужно создать новое подключение к базе данных ввести логин: root(который мы установили), пароль: 1(который мы установили) и порт: 33061(который мы установили) сделать следующий SQL запрос:

```
CREATE DATABASE Airflow CHARACTER SET UTF8mb3 COLLATE utf8_general_ci;
CREATE USER 'Airflow'@'%' IDENTIFIED BY '1';
```

в результате у вас появится база данных Airflow, затем зайти в пользователей и выдать пользователю Airflow все права доступа.

4.9 Чтобы быть более уверенным в работе можно ввести команду:

```
SELECT NOW()
```

на выходе должно в запросе получиться "2024-04-24 20:02:06" будет ваше время и дата

4.10 Далее надо создать базу данных "spark"

5. Это последний шаг а именно установка программы AirFlow

5.1 `python3 -m venv airflow-venv && source airflow-venv/bin/activate` данной командой мы заходим в виртуальную среду airflow

5.2 `pip install apache-airflow==2.7.3 --constraint "https://raw.githubusercontent.com/apache/airflow/constraints-2.7.3/constraints-3.8.txt" --force-reinstall --upgrade` данной командой мы устанавливаем airflow

5.3 `export AIRFLOW_HOME=/home/airflow && airflow db init` теперь делаем пробный запуск airflow

если у вас возникли проблемы с правами доступа тогда сделайте следующие:

- `cd /home/` перейдите в данную директорию
- `sudo chmod -R 777 ./` и выполните данную команду, затем снова сделайте пробный запуск airflow

5.4 Теперь у вас по пути "/home/" появится папка "airflow" там есть файл airflow.cfg который, мы должны заменить нашим файлом airflow.cfg

5.5 Теперь нужно в каталоге "/home/airflow/" создать папку "dags"

5.6 `export AIRFLOW_HOME=/home/airflow && airflow db init` делаем повторную инициализацию

ModuleNotFoundError: No module named 'MySQLdb' - данная ошибка говорит о том, что нам нужно установить ещё один модуль, чтобы его установить введите команду:

```
sudo apt-get install python3.8-dev libmysqlclient-dev && sudo apt install pkg-config && pip install mysqlclient
```

- `pip install python3.8` - установит дополнительно питон нужной версии
- `pip install pkg-config` - установит дополнительную библиотеку
- `pip install mysqlclient` - установит нужный mysqlclient

5.7 Выключаем wsl и включаем его через команду `PowerShell wsl -shutdown`, затем запускаем заново.

5.8 Снова инициализируем виртуальную среду шаг 5.1 и инициализируем airflow шаг 5.3

5.9 airflow users create --role Admin --username admin --email admin --firstname admin --lastname admin --password admin данной командой создаём пользователя для airflow

5.10 export AIRFLOW_HOME=/home/airflow && (airflow scheduler & airflow webserver -p 8080) данной командой запускаем сервер airflow, чтобы к нему подключиться в адресной строке браузера нужно ввести localhost:8080 в результате чего появиться админ сайт airflow логин/пароль: admin

Поздравляю вы полностью установили и запустили AirFlow в Windows 10 с помощью WSL

```
rtorn@Alicorn-113-56-527-85f-13:~$ sudo apt update
This message is shown once a day. To disable it please create the
/home/rtorn/.ushellrc file.
[sudo] password for rtorn:
Hit:1 http://archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Hit:3 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Hit:4 http://archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Hit:5 http://archive.ubuntu.com/ubuntu focal/universe amd64 Packages [8628 kB]
Hit:6 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [2809 kB]
Hit:7 http://archive.ubuntu.com/ubuntu focal/universe Translation-en [5124 kB]
Hit:8 http://archive.ubuntu.com/ubuntu focal/universe amd64 c-n-f Metadata [205 kB]
Hit:9 http://archive.ubuntu.com/ubuntu focal/multiverse amd64 Packages [144 kB]
Hit:10 http://archive.ubuntu.com/ubuntu focal/multiverse Translation-en [184 kB]
Hit:11 http://archive.ubuntu.com/ubuntu focal/multiverse amd64 c-n-f Metadata [9136 B]
Hit:12 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [3266 kB]
Hit:13 http://security.ubuntu.com/ubuntu focal-security/main Translation-en [435 kB]
Hit:14 http://archive.ubuntu.com/ubuntu focal-updates/main Translation-en [1517 kB]
Hit:15 http://security.ubuntu.com/ubuntu focal-security/main amd64 c-n-f Metadata [131.2 kB]
Hit:16 http://security.ubuntu.com/ubuntu focal-security/restricted amd64 Packages [2772 kB]
Hit:17 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 c-n-f Metadata [17.2 kB]
Hit:18 http://security.ubuntu.com/ubuntu focal-security/restricted Translation-en [187 kB]
Hit:19 http://security.ubuntu.com/ubuntu focal-security/restricted amd64 c-n-f Metadata [552 B]
Hit:20 http://security.ubuntu.com/ubuntu focal-security/universe amd64 Packages [956 kB]
Hit:21 http://archive.ubuntu.com/ubuntu focal-updates/restricted amd64 Packages [2889 kB]
Hit:22 http://security.ubuntu.com/ubuntu focal-security/universe Translation-en [201 kB]
Hit:23 http://security.ubuntu.com/ubuntu focal-security/universe amd64 c-n-f Metadata [19.2 kB]
Hit:24 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 Packages [24.0 kB]
Hit:25 http://security.ubuntu.com/ubuntu focal-security/multiverse Translation-en [5084 B]
Hit:26 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 c-n-f Metadata [148 B]
Hit:27 http://archive.ubuntu.com/ubuntu focal-updates/restricted Translation-en [403 kB]
Hit:28 http://archive.ubuntu.com/ubuntu focal-updates/restricted amd64 c-n-f Metadata [552 B]
Hit:29 http://archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [1183 kB]
Hit:30 http://archive.ubuntu.com/ubuntu focal-updates/universe Translation-en [283 kB]
Hit:31 http://archive.ubuntu.com/ubuntu focal-updates/universe amd64 c-n-f Metadata [26.7 kB]
Hit:32 http://archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 Packages [26.2 kB]
Hit:33 http://archive.ubuntu.com/ubuntu focal-updates/multiverse Translation-en [7880 B]
Hit:34 http://archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 c-n-f Metadata [520 B]
Hit:35 http://archive.ubuntu.com/ubuntu focal-backports/main amd64 Packages [45.7 kB]
Hit:36 http://archive.ubuntu.com/ubuntu focal-backports/main Translation-en [16.3 kB]
Hit:37 http://archive.ubuntu.com/ubuntu focal-backports/main amd64 c-n-f Metadata [1420 B]
Hit:38 http://archive.ubuntu.com/ubuntu focal-backports/restricted amd64 c-n-f Metadata [116 B]
Hit:39 http://archive.ubuntu.com/ubuntu focal-backports/universe amd64 Packages [25.0 kB]
Hit:40 http://archive.ubuntu.com/ubuntu focal-backports/universe Translation-en [16.3 kB]
Hit:41 http://archive.ubuntu.com/ubuntu focal-backports/universe amd64 c-n-f Metadata [880 B]
Hit:42 http://archive.ubuntu.com/ubuntu focal-backports/multiverse amd64 c-n-f Metadata [116 B]
Fetched 31.0 MB in 4s (7846 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
176 packages can be upgraded. Run 'apt list --upgradable' to see them.
rtorn@Alicorn-113-56-527-85f-13:~$ sudo add-apt-repository ppa:deadsnakes/ppa
This PPA contains more recent Python versions packaged for Ubuntu.
Disclaimer: there's no guarantee of timely updates in case of security problems or other issues. If you want to use them in a security-or-otherwise-critical environment (say, on a production server), you do so at your own risk.

Update Note
-----
Please use this repository instead of ppa:kruul/deadsnakes.

Reporting Issues
-----
Issues can be reported in the master issue tracker at:
https://github.com/deadsnakes/issues/issues

Supported Ubuntu and Python Versions
-----
Ubuntu 20.04 (focal) Python3.5 - Python3.7, Python3.9 - Python3.11
Ubuntu 22.04 (jammy) Python3.7 - Python3.9, Python3.11 - Python3.13
Note: Python2.7 (all), Python 3.8 (focal), Python 3.10 (jammy) are not provided by deadsnakes as upstream ubuntu provides those packages.

Why some packages aren't built:
-----
```


[illegible][illegible]


```
mysql> exit
bye
ritort@allcorn-113-St-527-BSF-13:~$ sudo nano /etc/mysql/my.cnf
ritort@allcorn-113-St-527-BSF-13:~$ sudo nano /etc/mysql/my.cnf
* Stopping MySQL database server mysqld
* Starting MySQL database server mysqld
su: warning: cannot change directory to /nonexistent: No such file or directory
ritort@allcorn-113-St-527-BSF-13:~$ python3 -a venv airflow-venv --source airflow-venv/bin/activate
[airflow-venv] ritort@allcorn-113-St-527-BSF-13:~$ pip install apache-airflow==2.7.3 --constraint "https://raw.githubusercontent.com/apache/airflow/constraints-2.7.3/constraints-3.8.txt" --force-reinstall --upgrade
Collecting apache-airflow==2.7.3-py3-none-any.whl (12.9 MB)
  Downloading apache-airflow-2.7.3-py3-none-any.whl (12.9 MB)
Collecting Flask==2.2.5
  Downloading flask-2.2.5-py3-none-any.whl (101 kB)
Collecting tenacity==8.2.3
  Downloading tenacity-8.2.3-py3-none-any.whl (24 kB)
Collecting alembic==1.12.1
  Downloading alembic-1.12.1-py3-none-any.whl (226 kB)
Collecting apache-airflow-providers-wagtail==4.0
  Downloading apache-airflow-providers-wagtail-4.0-py3-none-any.whl (17 kB)
Collecting python-evtdb==15.9
  Downloading python-evtdb-15.9.0.tar.gz (31 kB)
Collecting Flask-Caching==2.1.0
  Downloading flask_caching-2.1.0-py3-none-any.whl (28 kB)
Collecting apache-airflow-providers-sqlite==3.5.0
  Downloading apache-airflow-providers-sqlite-3.5.0-py3-none-any.whl (13 kB)
Collecting cryptography==41.0.5
  Downloading cryptography-41.0.5-cp37-abi3-manylinux_2_17_x86_64_musllinux_2.17_x86_64.whl (4.4 MB)
Collecting python-datatables==2.8.2
  Downloading python_datatables-2.8.2-py3-none-any.whl (247 kB)
Collecting graphviz==0.20.1
  Downloading graphviz-0.20.1-py3-none-any.whl (47 kB)
Collecting Flask-Sessions==0.5.0
  Downloading flask_session-0.5.0-py3-none-any.whl (7.2 kB)
Collecting SQLAlchemy==1.4.50
  Downloading SQLAlchemy-1.4.50-cp38-cp38-manylinux_2_5_x86_64_musllinux_2.12_x86_64_musllinux_2018_x86_64_musllinux_2_17_x86_64_musllinux_2014_x86_64.whl (1.6 MB)
Collecting Jinja2==3.1.2
  Downloading Jinja2-3.1.2-py3-none-any.whl (133 kB)
Collecting python-slugify==8.0.1
  Downloading python_slugify-8.0.1-py2.py3-none-any.whl (9.7 kB)
Collecting cron-descriptor==1.4.0
  Downloading cron_descriptor-1.4.0.tar.gz (29 kB)
Collecting Pygments==2.16.1
  Downloading Pygments-2.16.1-py3-none-any.whl (1.2 MB)
Collecting ConfigUpdater==3.1.1
  Downloading ConfigUpdater-3.1.1-py3-none-any.whl (34 kB)
Collecting packaging==23.2
  Downloading packaging-23.2-py3-none-any.whl (53 kB)
Collecting itsdangerous==2.1.2
  Downloading itsdangerous-2.1.2-py3-none-any.whl (15 kB)
Collecting opentelemetry-exporter-otlp==1.29.0
  Downloading opentelemetry_exporter_otlp-1.29.0-py3-none-any.whl (7.0 kB)
Collecting argcomplete==1.12.3
  Downloading argcomplete-1.12.3-py3-none-any.whl (41 kB)
Collecting arconote==1.1.3
  Downloading arconote-1.1.3-py3-none-any.whl (41 kB)
Collecting catstrings==2.1.2
  Downloading catstrings-2.1.2-py3-none-any.whl (41 kB)
[airflow-venv] ritort@allcorn-113-St-527-BSF-13:~$
```

```
Traceback (most recent call last):
  File "/usr/lib/python3.8/pathlib.py", line 1288, in mkdir
    self._accessor.mkdir(self, mode)
FileNotFoundError: [Errno 2] No such file or directory: '/home/airflow/logs/scheduler'
During handling of the above exception, another exception occurred:
Traceback (most recent call last):
  File "/usr/lib/python3.8/pathlib.py", line 1288, in mkdir
    self._accessor.mkdir(self, mode)
FileNotFoundError: [Errno 2] No such file or directory: '/home/airflow/logs'
During handling of the above exception, another exception occurred:
Traceback (most recent call last):
  File "/usr/lib/python3.8/logging/config.py", line 563, in configure
    handler = self.configure_handler(handler_name)
  File "/usr/lib/python3.8/logging/config.py", line 744, in configure_handler
    result = factory(**kwargs)
  File "/home/ritort/airflow-venv/lib/python3.8/site-packages/airflow/utis/log/file_processor_handler.py", line 50, in __init__
    Path(self.get_log_directory()).mkdir(parents=True, exist_ok=True)
  File "/usr/lib/python3.8/pathlib.py", line 1292, in mkdir
    self.parent.mkdir(parents=True, exist_ok=True)
  File "/usr/lib/python3.8/pathlib.py", line 1292, in mkdir
    self.parent.mkdir(parents=True, exist_ok=True)
  File "/usr/lib/python3.8/pathlib.py", line 1288, in mkdir
    self._accessor.mkdir(self, mode)
PermissionError: [Errno 13] Permission denied: '/home/airflow'
The above exception was the direct cause of the following exception:
Traceback (most recent call last):
  File "/home/ritort/airflow-venv/bin/airflow", line 5, in <module>
    from airflow.__main__ import main
  File "/home/ritort/airflow-venv/lib/python3.8/site-packages/airflow/__init__.py", line 68, in <module>
    settings.initialize()
  File "/home/ritort/airflow-venv/lib/python3.8/site-packages/airflow/settings.py", line 541, in initialize
    LOGGING_CONFIG_PATH = configure_logging()
  File "/home/ritort/airflow-venv/lib/python3.8/site-packages/airflow/logging_config.py", line 74, in configure_logging
    raise ValueError("Unable to configure handler")
ValueError: Unable to configure handler 'processor'
[airflow-venv] ritort@allcorn-113-St-527-BSF-13:~$ cd ..
[airflow-venv] ritort@allcorn-113-St-527-BSF-13:~/home$ ls
ritort
[airflow-venv] ritort@allcorn-113-St-527-BSF-13:~/home$ sudo chmod -R 777 ./
[airflow-venv] ritort@allcorn-113-St-527-BSF-13:~/home$ ls
ritort
[airflow-venv] ritort@allcorn-113-St-527-BSF-13:~/home$ export AIRFLOW_HOME=/home/airflow && airflow db init
/home/ritort/airflow-venv/lib/python3.8/site-packages/airflow/cli/commands/db_command.py:43 DeprecationWarning: 'db init' is deprecated. Use 'db migrate' instead to migrate the db and/or airflow connections create-default-connections to create the default connections.
DB: sqlite:///home/airflow/airflow.db
[2024-04-24T20:13:45.450000] [migration.py:213] INFO - Context impl SQLiteImpl.
[2024-04-24T20:13:45.450000] [migration.py:216] INFO - Will assume non-transactional DDL.
INFO [alembic.runtime.migration] Context impl SQLiteImpl.
INFO [alembic.runtime.migration] Will assume non-transactional DDL.
INFO [alembic.runtime.migration] Running stamp revision -> 4b9de8318b3a
WARNING[airflow.models.crypto] empty cryptography key - values will not be stored encrypted.
Initialization done
[airflow-venv] ritort@allcorn-113-St-527-BSF-13:~/home$
```


