Cloud Data Warehouse for PostgreSQL

Data Warehouse Development

Product Documentation

Contents

Data Warehouse Development
Creating Airflow in Cloud

Last updated：2024-02-02 15:36:51

Apache Airflow is an open-source workflow management system that integrates orchestration, scheduling, monitoring, and graphical display. It can manage ETL tasks for data warehouses. This document describes how to build Airflow in a CVM instance.

Default Airflow Installation

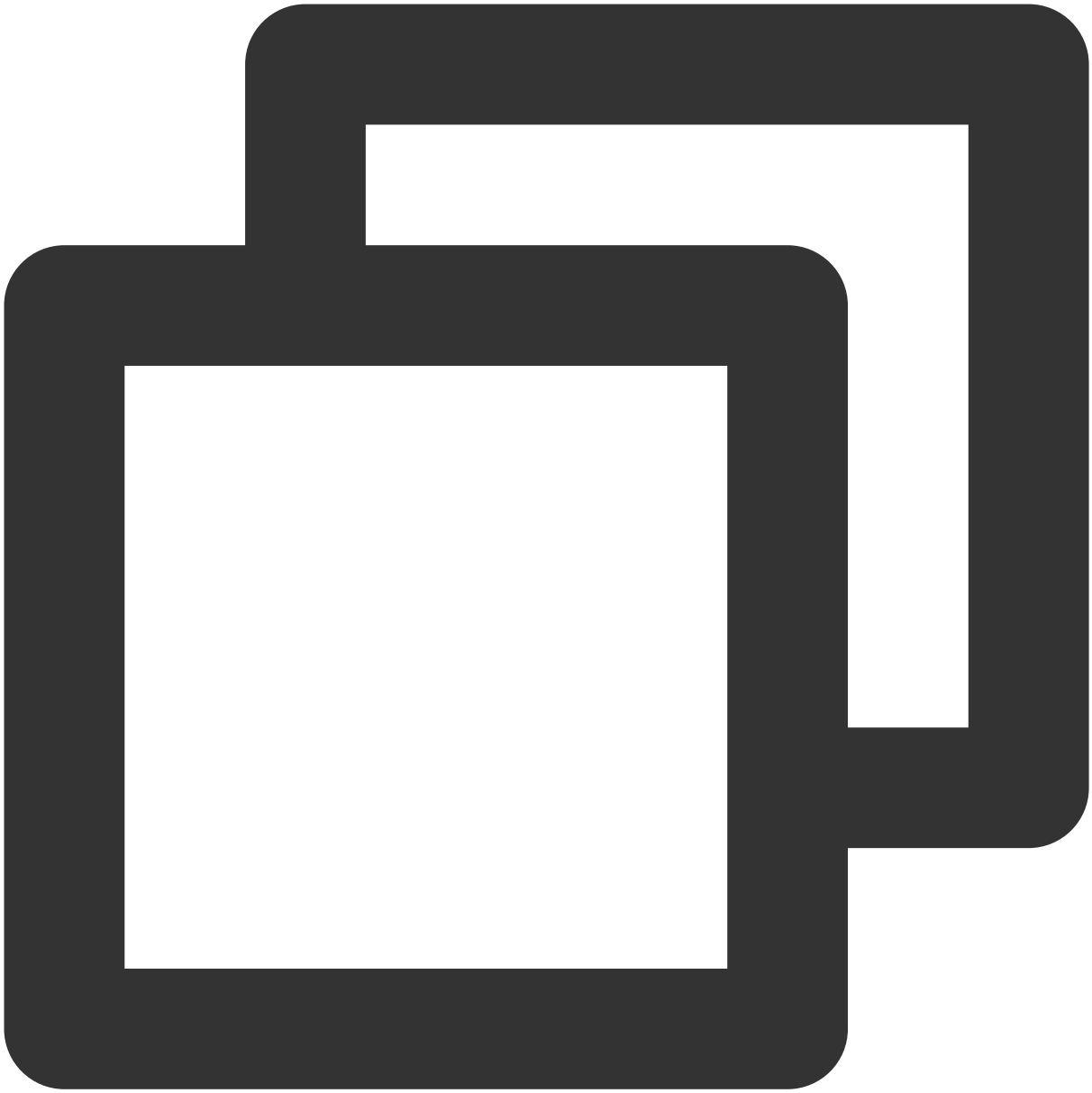
1. Purchase a CVM instance.

Note:

This document uses CentOS 8.0 as an example.

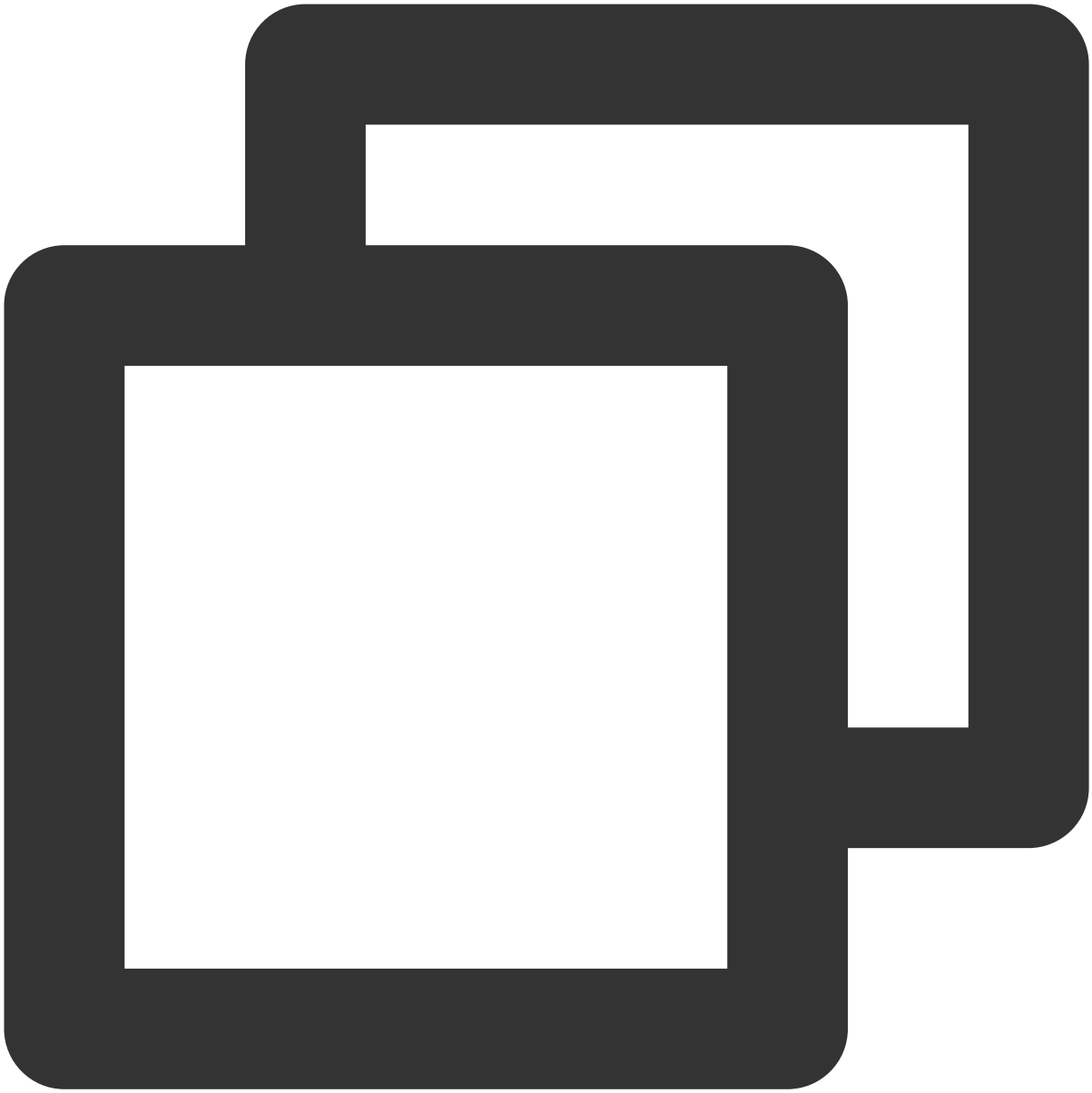
2. Install the dependent software.

Before installing Airflow, you need to install the following dependencies.



```
yum install redhat-rpm-config -y
yum install mysql-devel -y
yum install python3-devel -y
dnf update gcc annobin -y
```

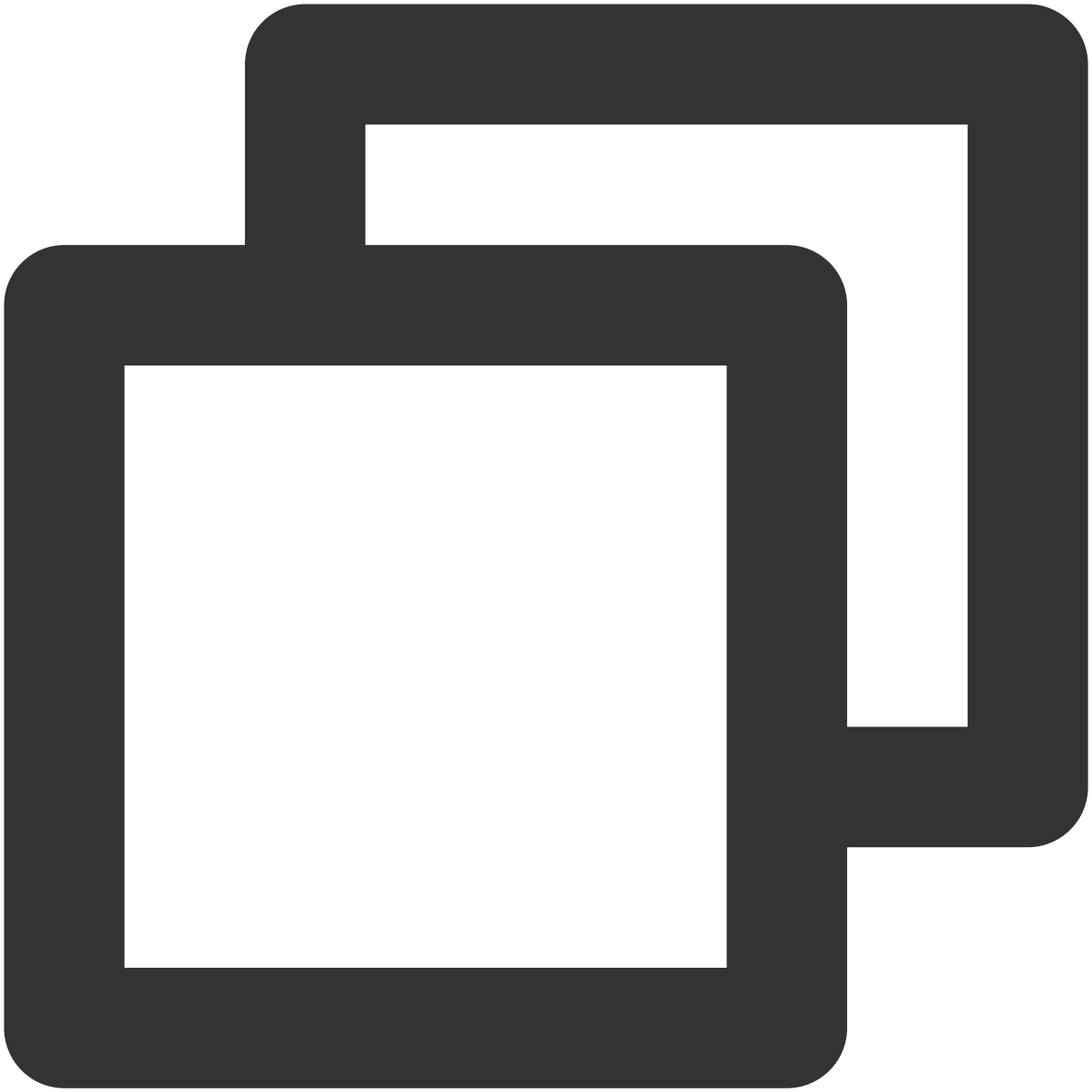
3. Create the `Home` directory.



```
mkdir -p /usr/local/services/airflow
export AIRFLOW_HOME=/usr/local/services/airflow
```

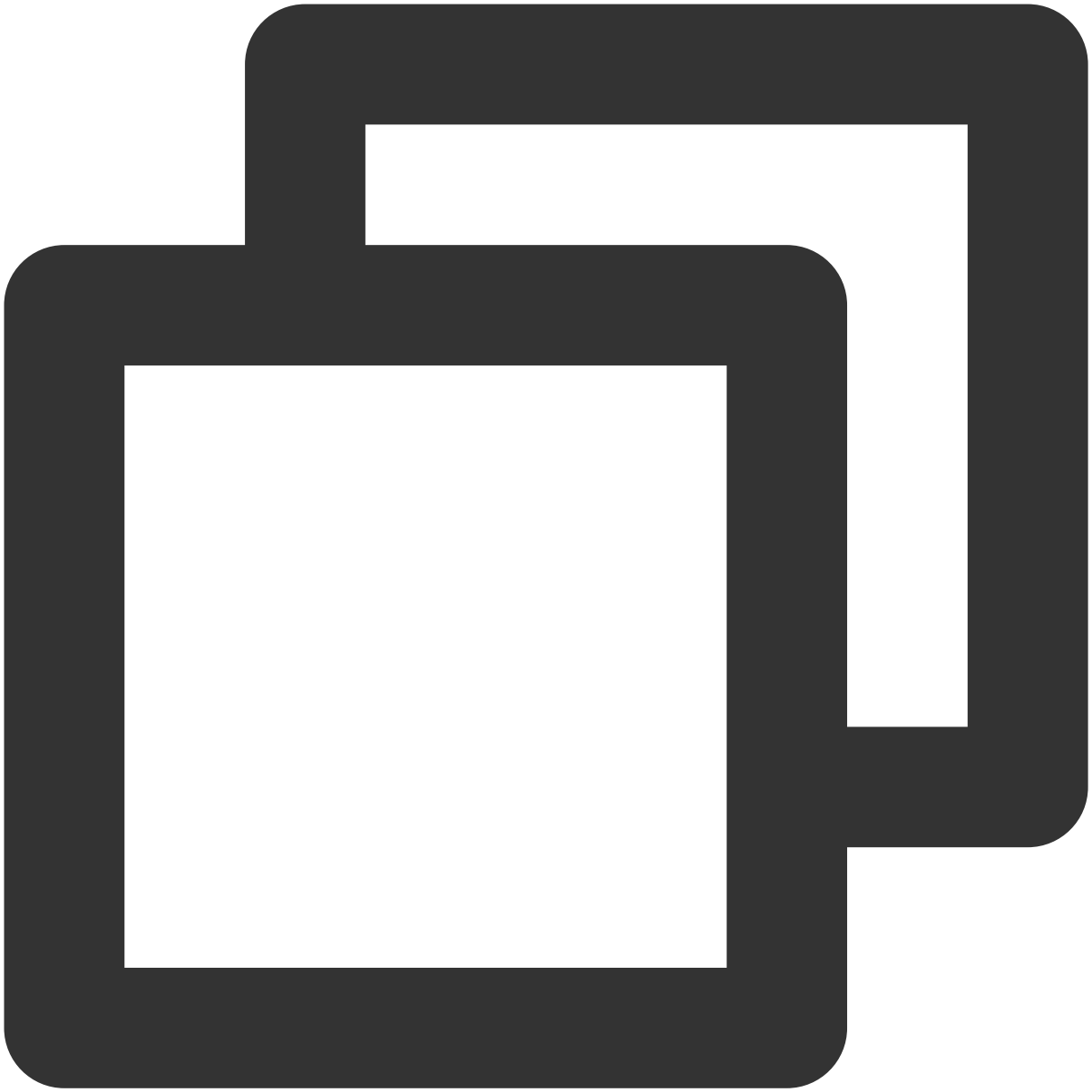
The `AIRFLOW_HOME` variable can be configured in the `/etc/profile` file.

4. Install Airflow.



```
pip install apache-airflow[mysql]
```

5. Initialize the database.



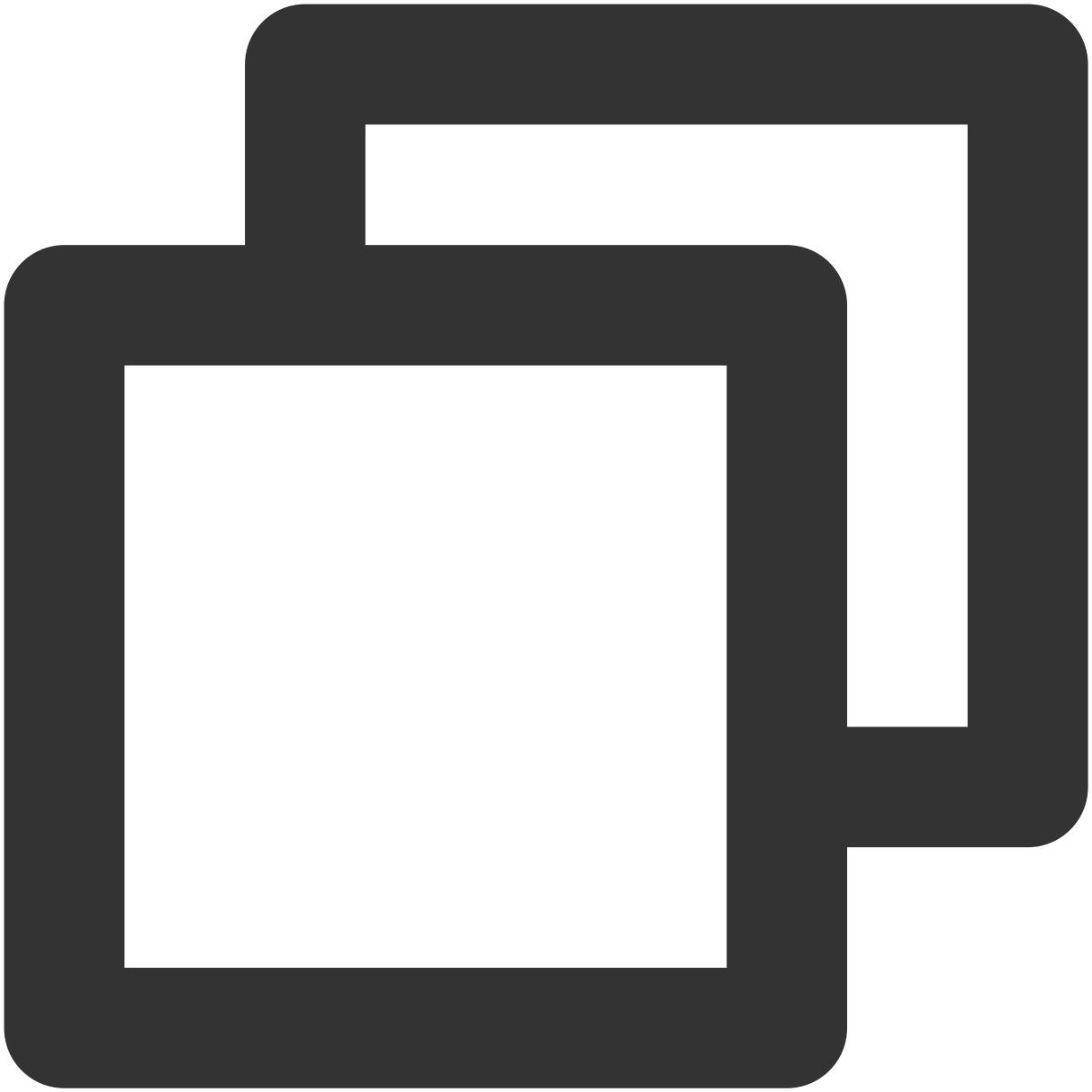
```
airflow initdb
```

6. Configure the security group.

The default port number for Airflow's web server is 8080. If you want to access it over the public network, you need to open the 8080 port in the security group.

7. Enable the web UI.

Use the following command:



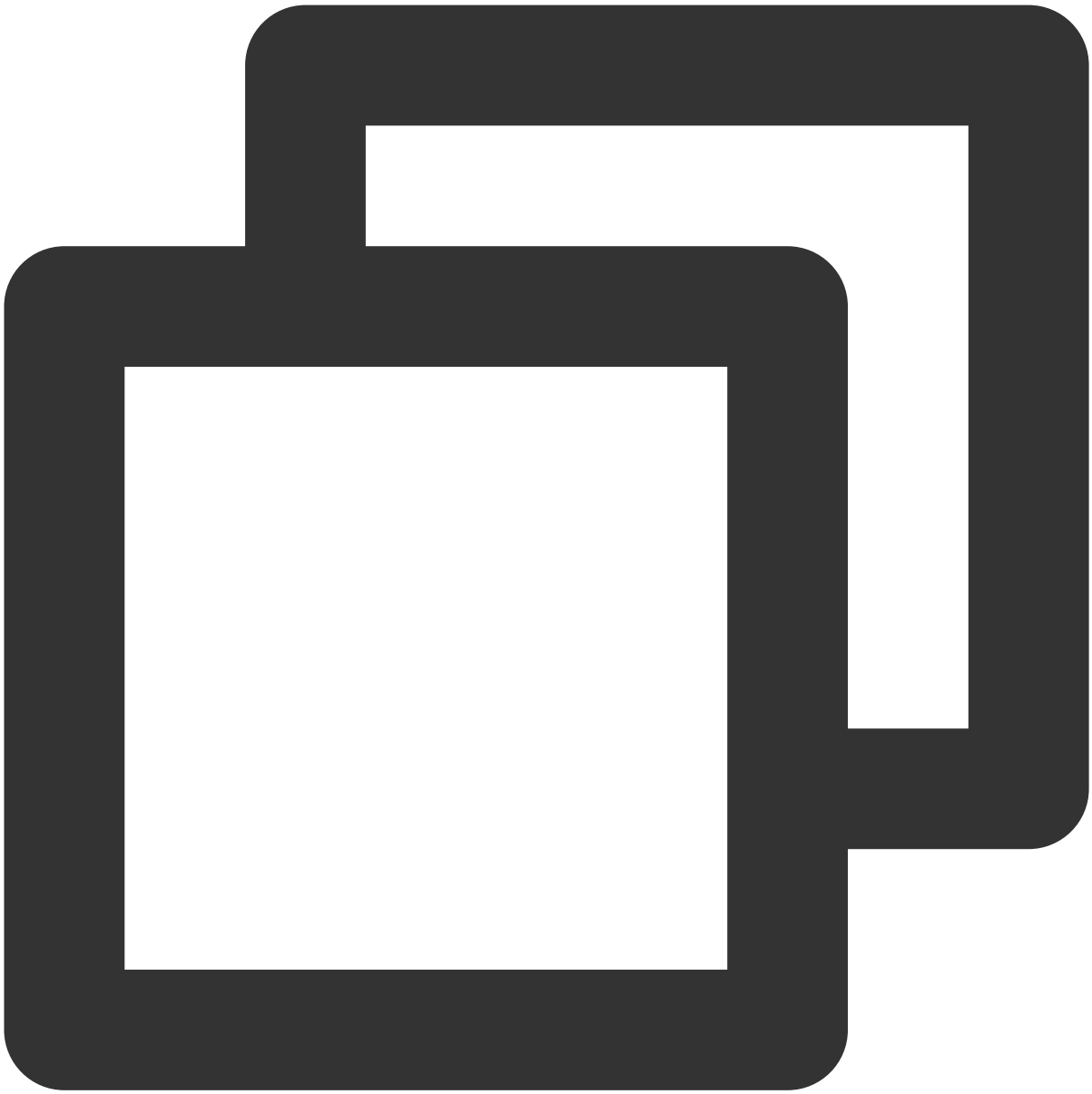
```
airflow webserver -D
```

If you can access the UI at `url http://{ip}:8080/admin/` , the configuration is successful.

Processing Time Zone

Airflow uses the UTC time zone, which is eight hours behind Beijing time. As Airflow writes some fixed code, you need to modify the source code in addition to the configuration files in the following steps:

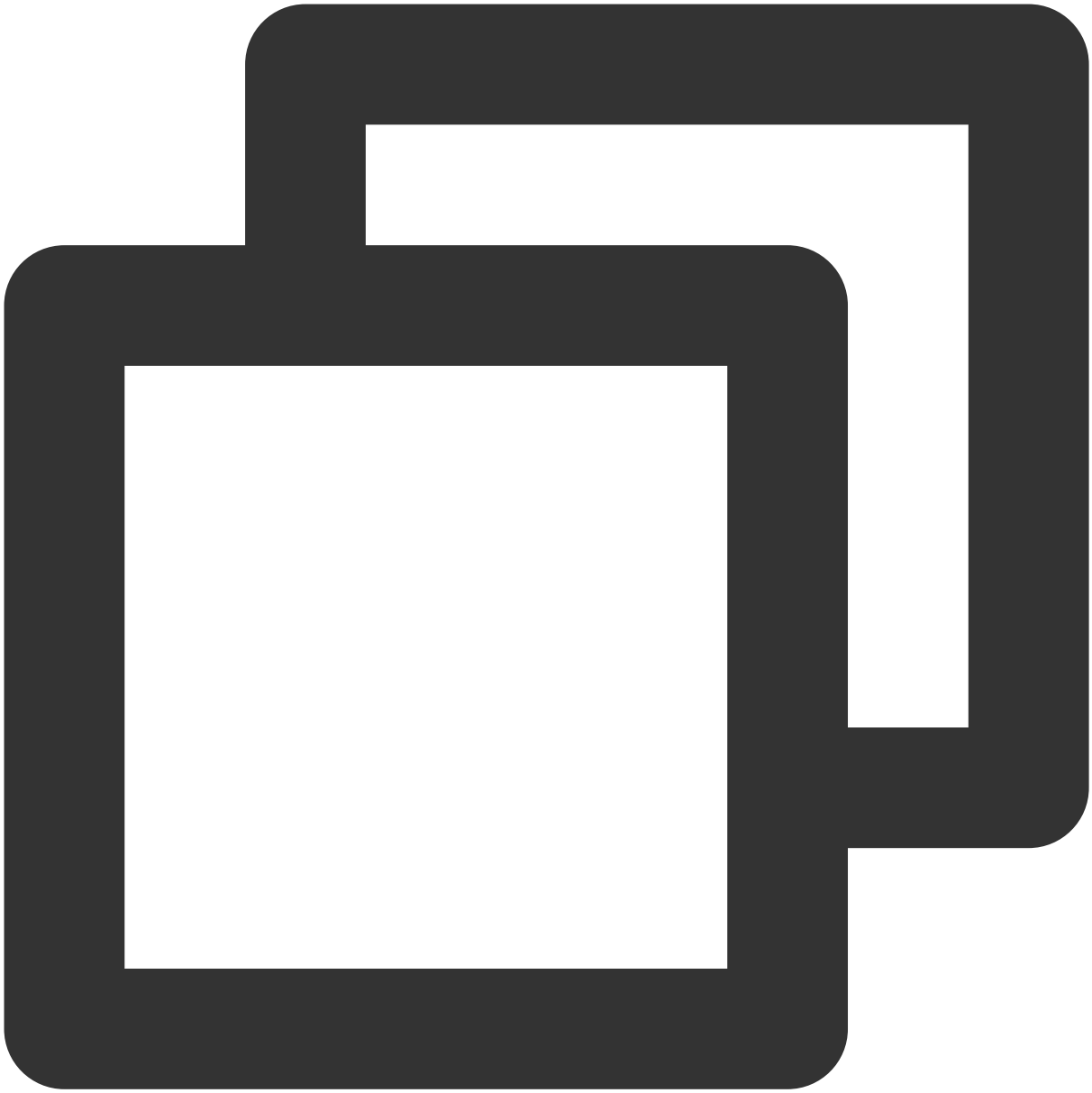
1. Modify `airflow.cfg` in `AIRFLOW_HOME` .



```
Change `default_timezone = utc` to `default_timezone = Asia/Shanghai`
Change `default_ui_timezone = UTC` to `default_ui_timezone = Asia/Shanghai`
```

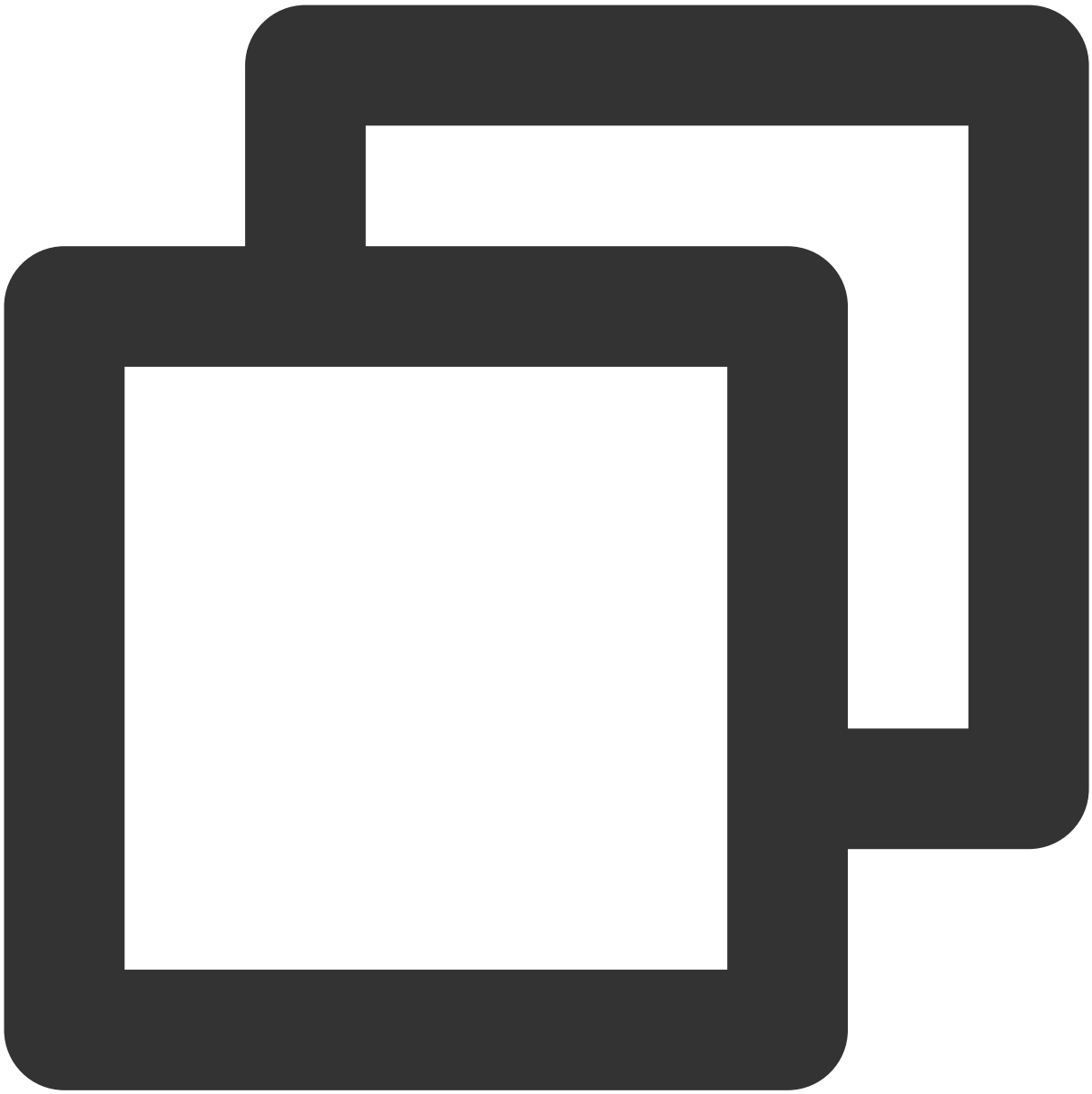
2. Modify the `/usr/local/lib/python3.6/site-packages/airflow/utils/timezone.py` file.

Add the following statement below the `utc = pendulum.timezone('UTC')` statement:



```
from airflow.configuration import conf
try:
 tz = conf.get("core", "default_timezone")
 if tz == "system":
     utc = pendulum.local_timezone()
 else:
     utc = pendulum.timezone(tz)
except Exception:
 pass
```

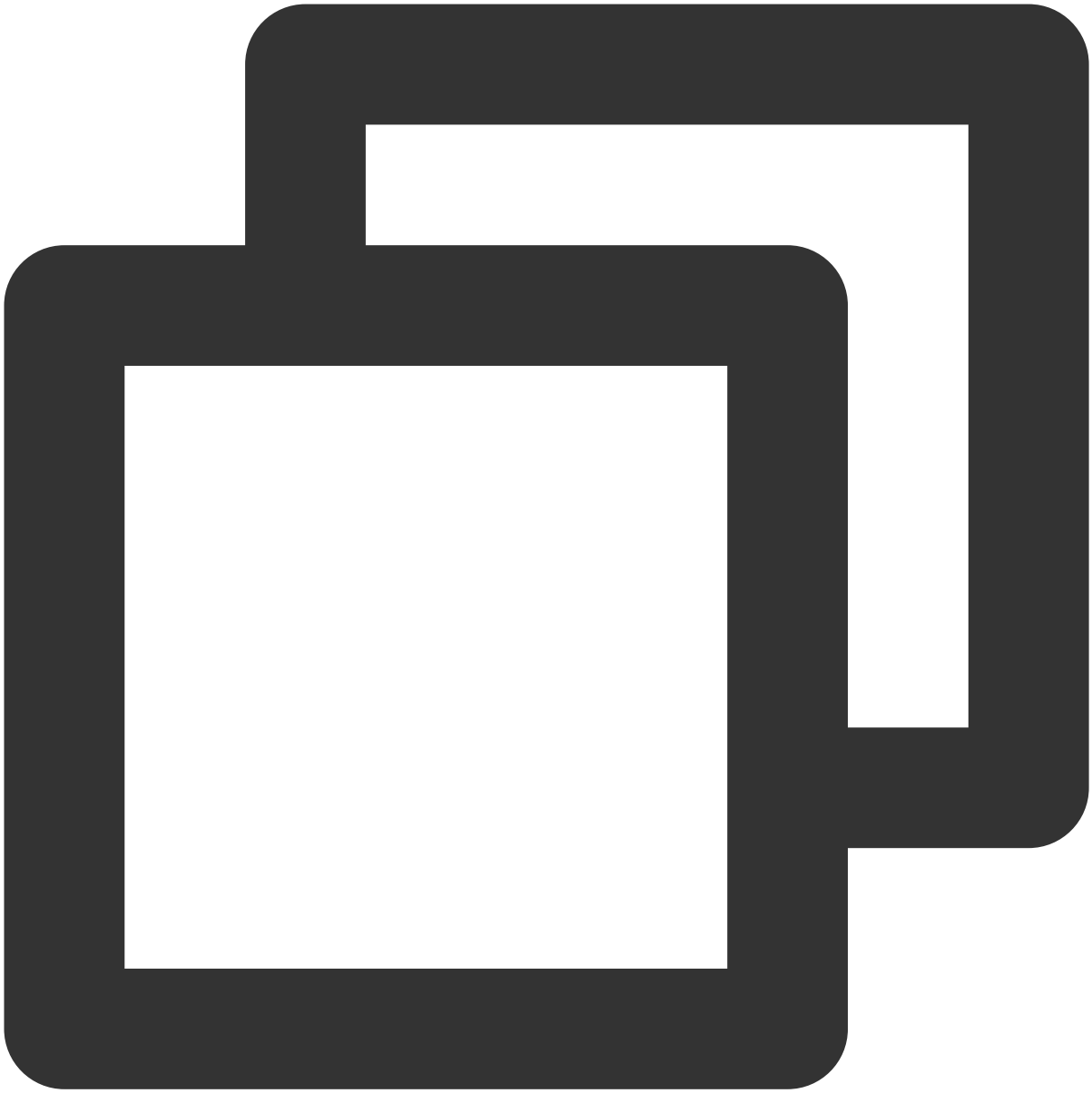
Modify the `utcnow()` function:



```
Change `d = dt.datetime.utcnow()` to `d = dt.datetime.now()`
```

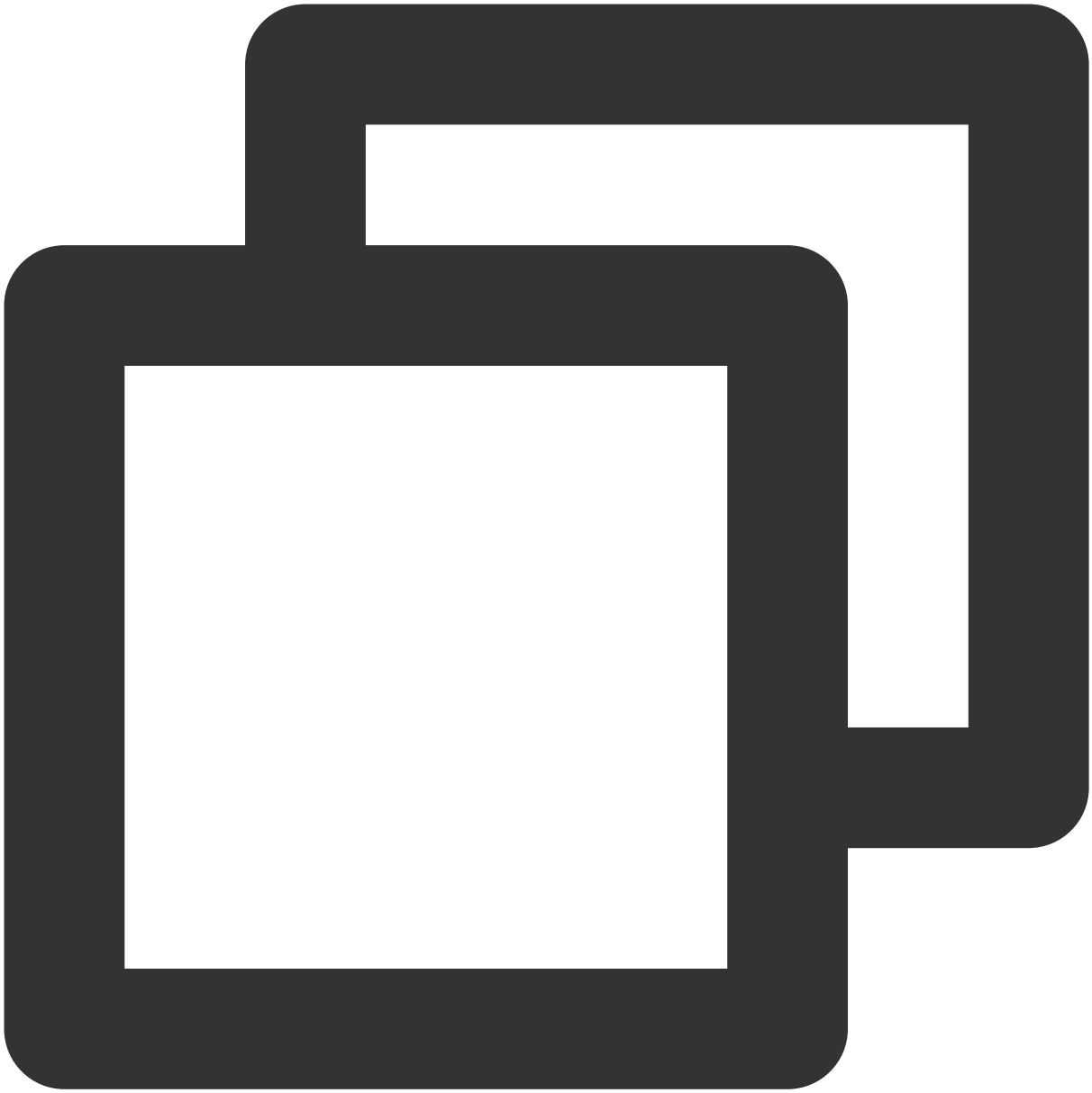
3. Modify the `/usr/local/lib/python3.6/site-packages/airflow/utils/sqlalchemy.py` file.

Add the following content below the `utc = pendulum.timezone('UTC')` statement:



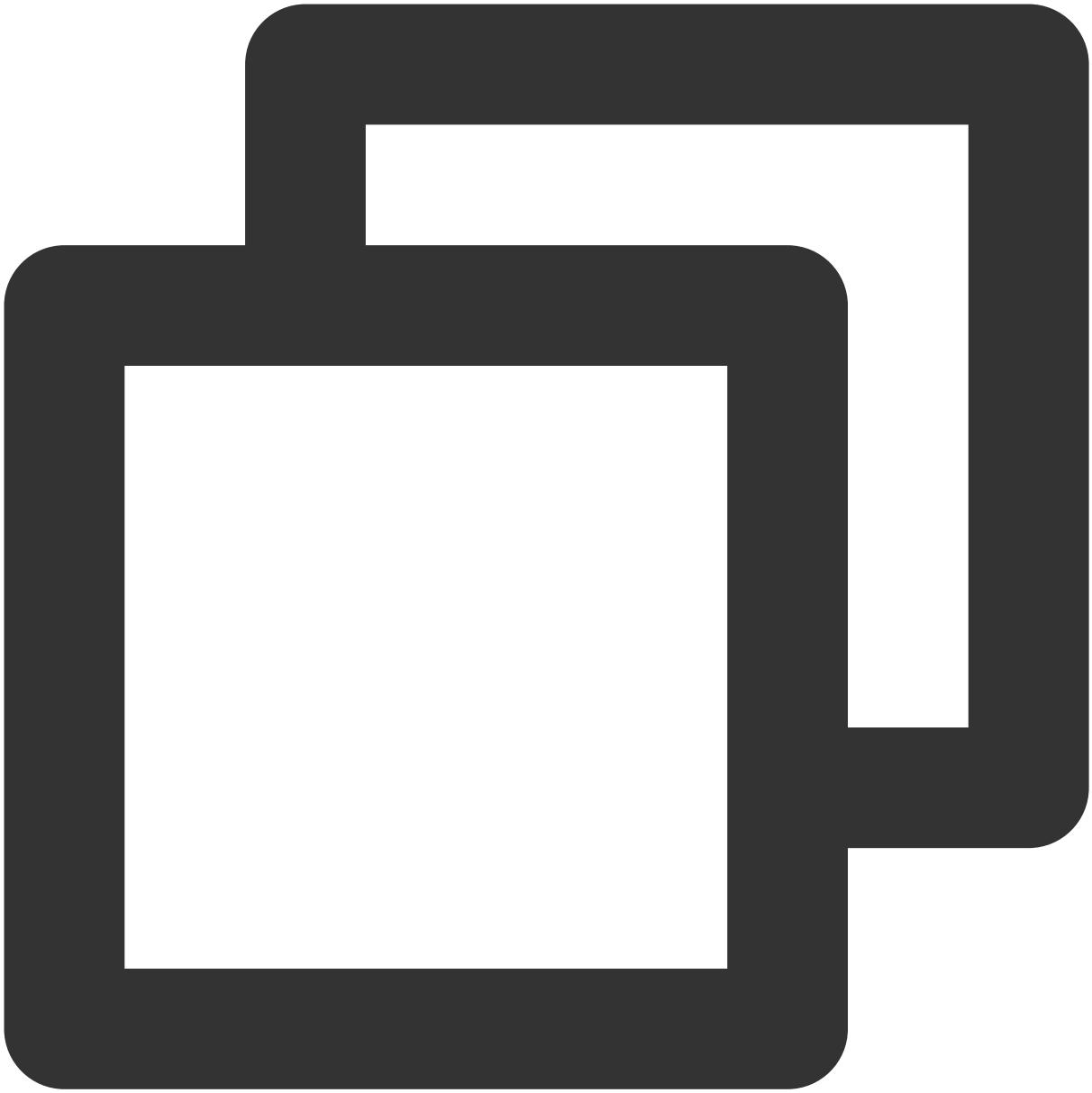
```
from airflow.configuration import conf
try:
 tz = conf.get("core", "default_timezone")
 if tz == "system":
     utc = pendulum.local_timezone()
 else:
     utc = pendulum.timezone(tz)
except Exception:
 pass
```

Comment the statement:

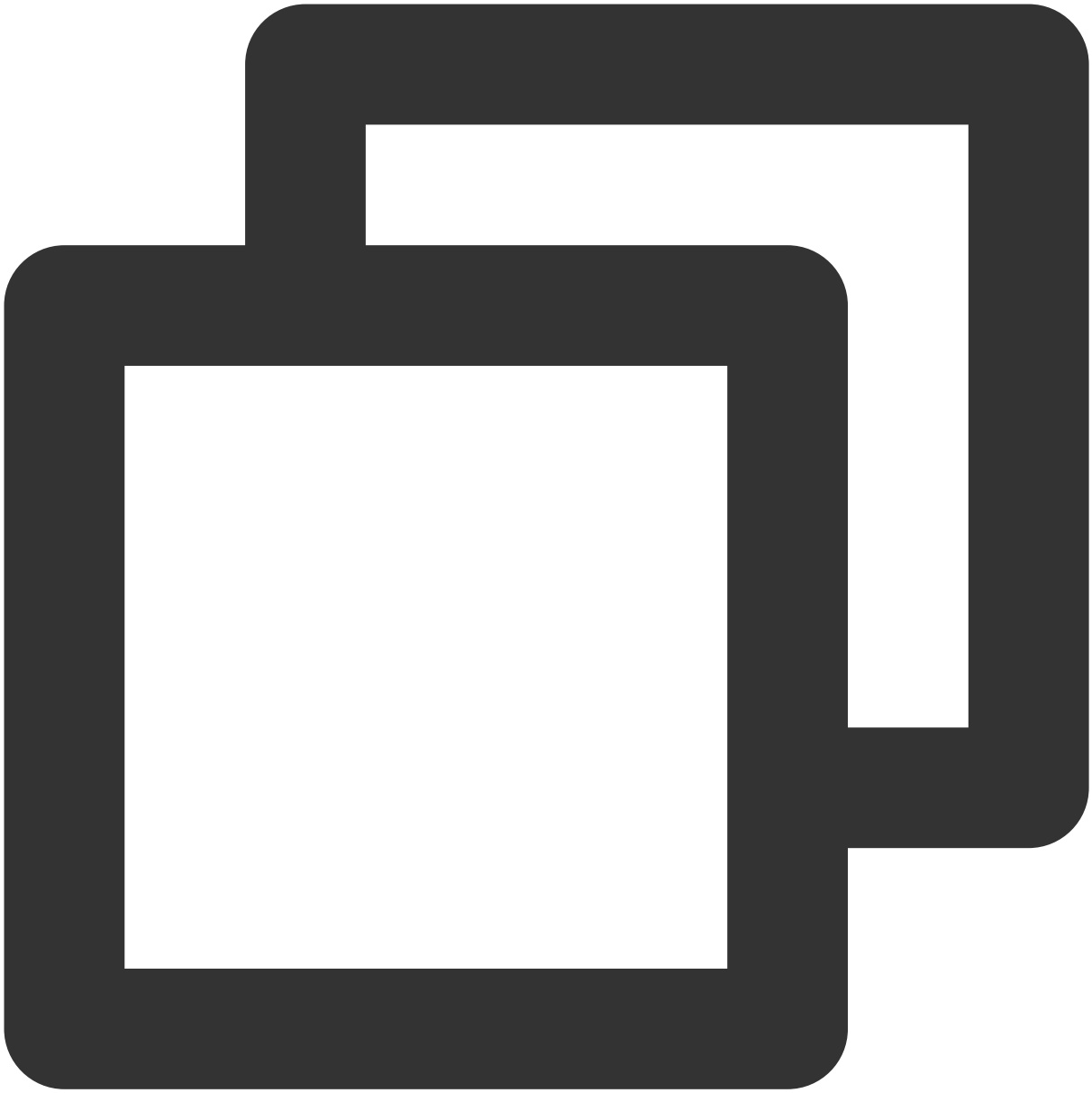


```
cursor.execute("SET time_zone = '+00:00'")
```

4. Modify the `/usr/local/lib/python3.6/site-packages/airflow/www/templates/admin/master.html` file.

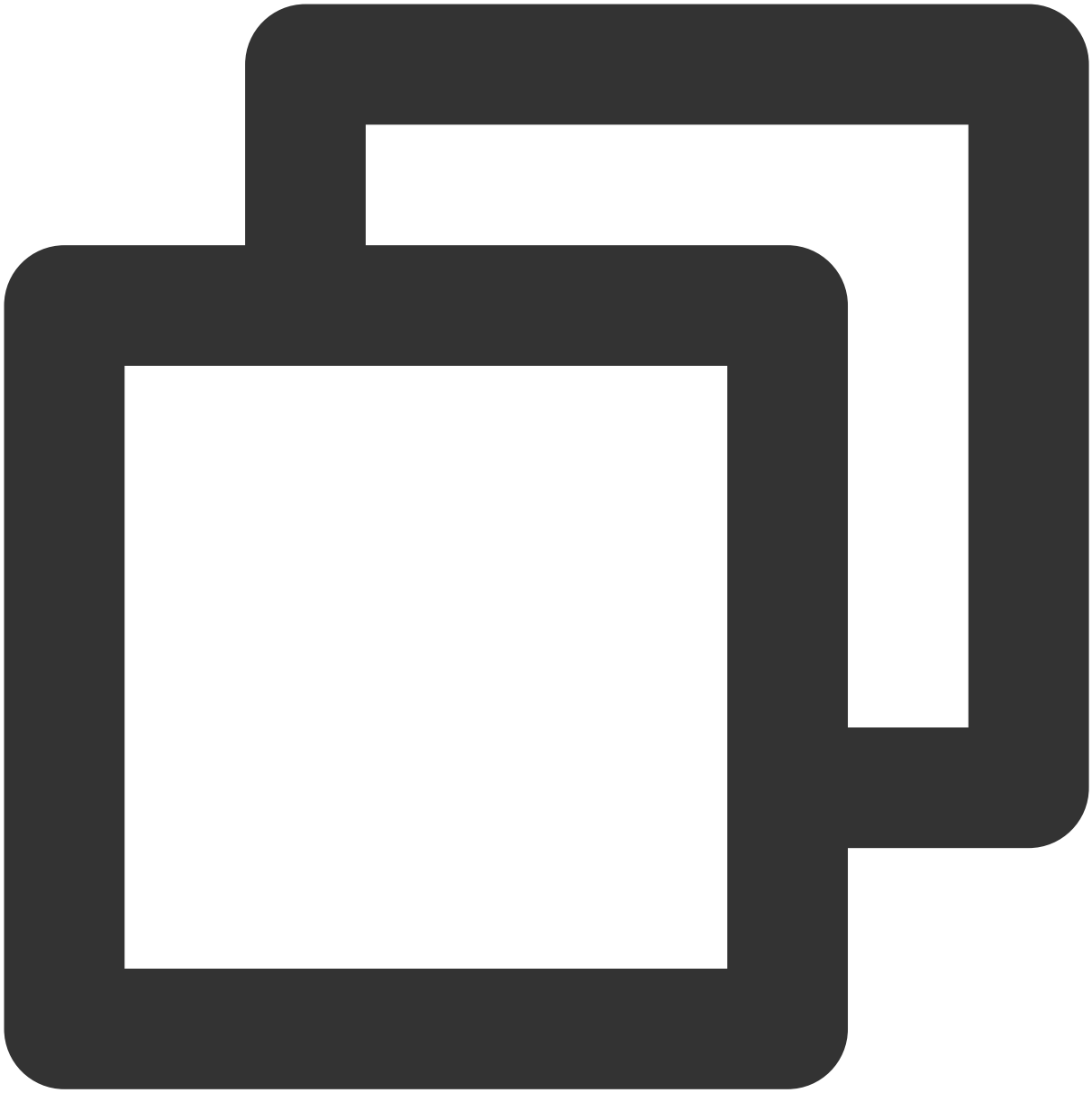


```
var UTCseconds = (x.getTime() + x.getTimezoneOffset()*60*1000);
Change to
var UTCseconds = x.getTime();
```



```
"timeFormat":"H:i:s %UTC%",
Change to
"timeFormat":"H:i:s",
```

5. Restart the web server.



```
cat {AIRFLOW_HOME}/airflow-webserver.pid
kill {pid}
airflow webserver -D
```

Using TencentDB for MySQL to Store Data

Airflow uses SQLite to store data by default. If you want to launch it in the production environment, you must ensure the high availability. In the following steps, TencentDB for MySQL is used as an example:

1. Purchase a TencentDB for MySQL instance.

Note:

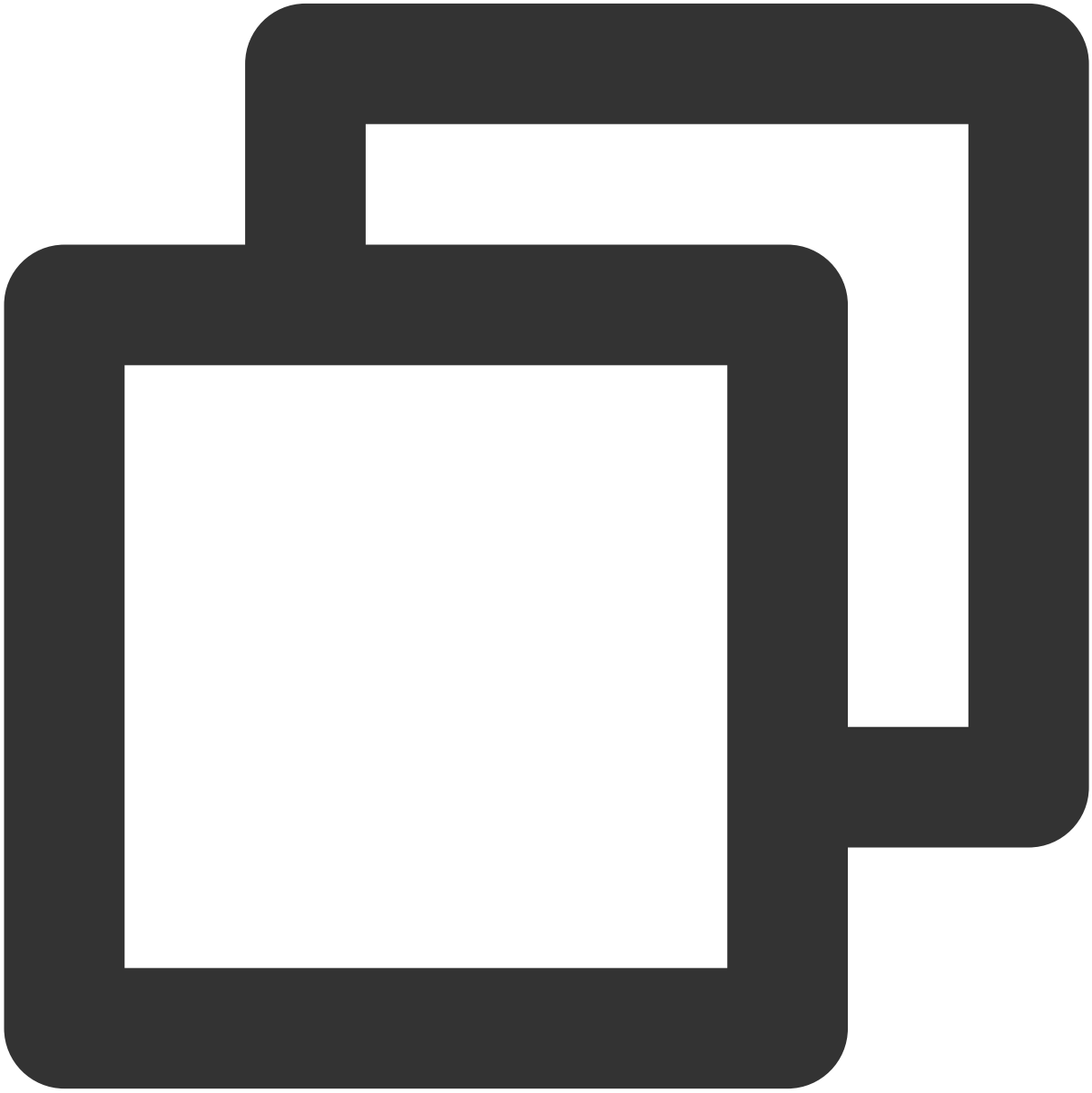
It must be High-Availability or Finance edition, as the Basic edition does not support the `explicit_defaults_for_timestamp` parameter and cannot be used for Airflow storage.

2. Modify parameters.

Set the `explicit_defaults_for_timestamp` parameter to `ON` in the console.

3. Create a database and user.

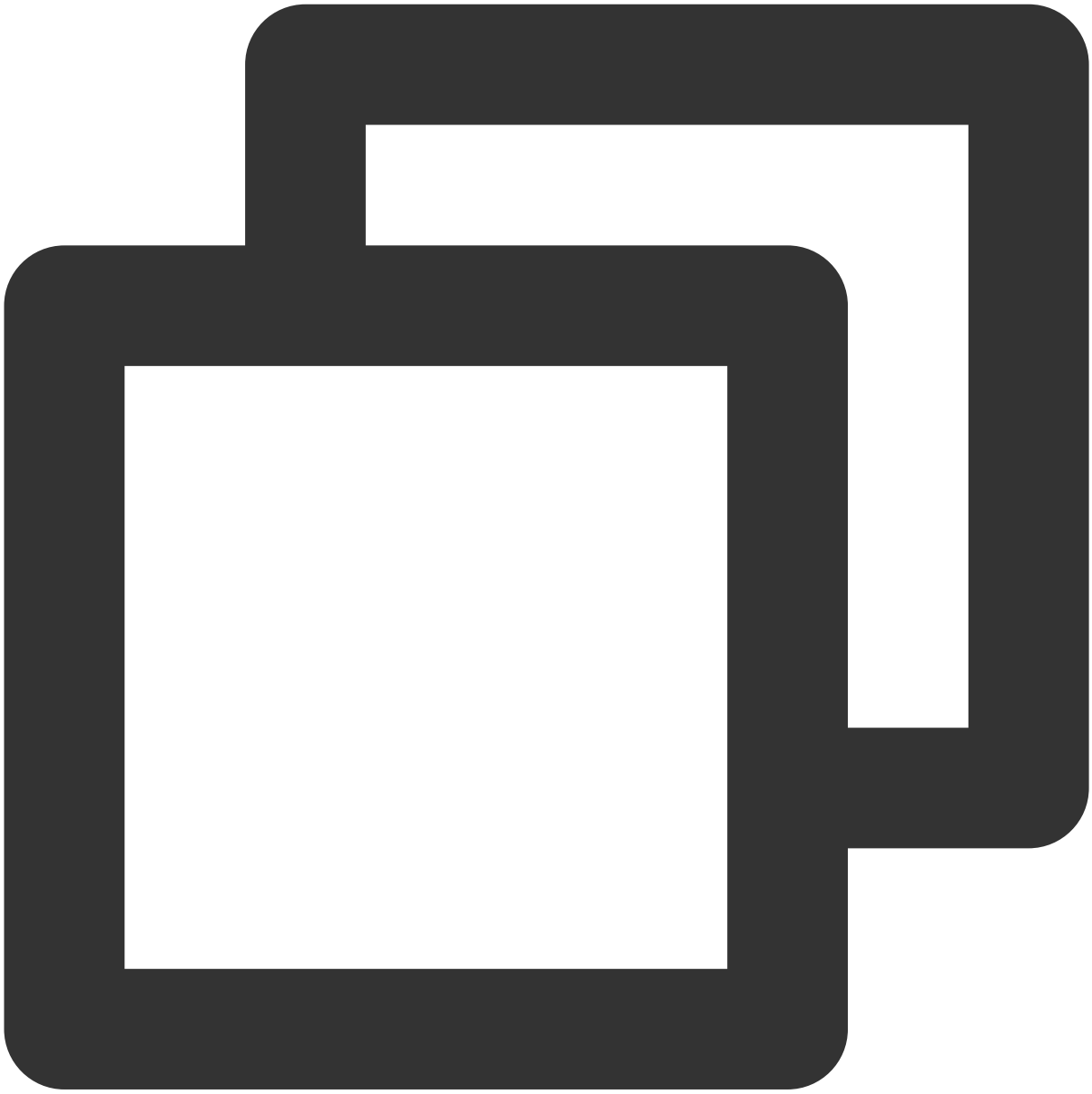
Log in to MySQL and run the following statement, where you can change the username and password as needed.



```
create database airflow;
```

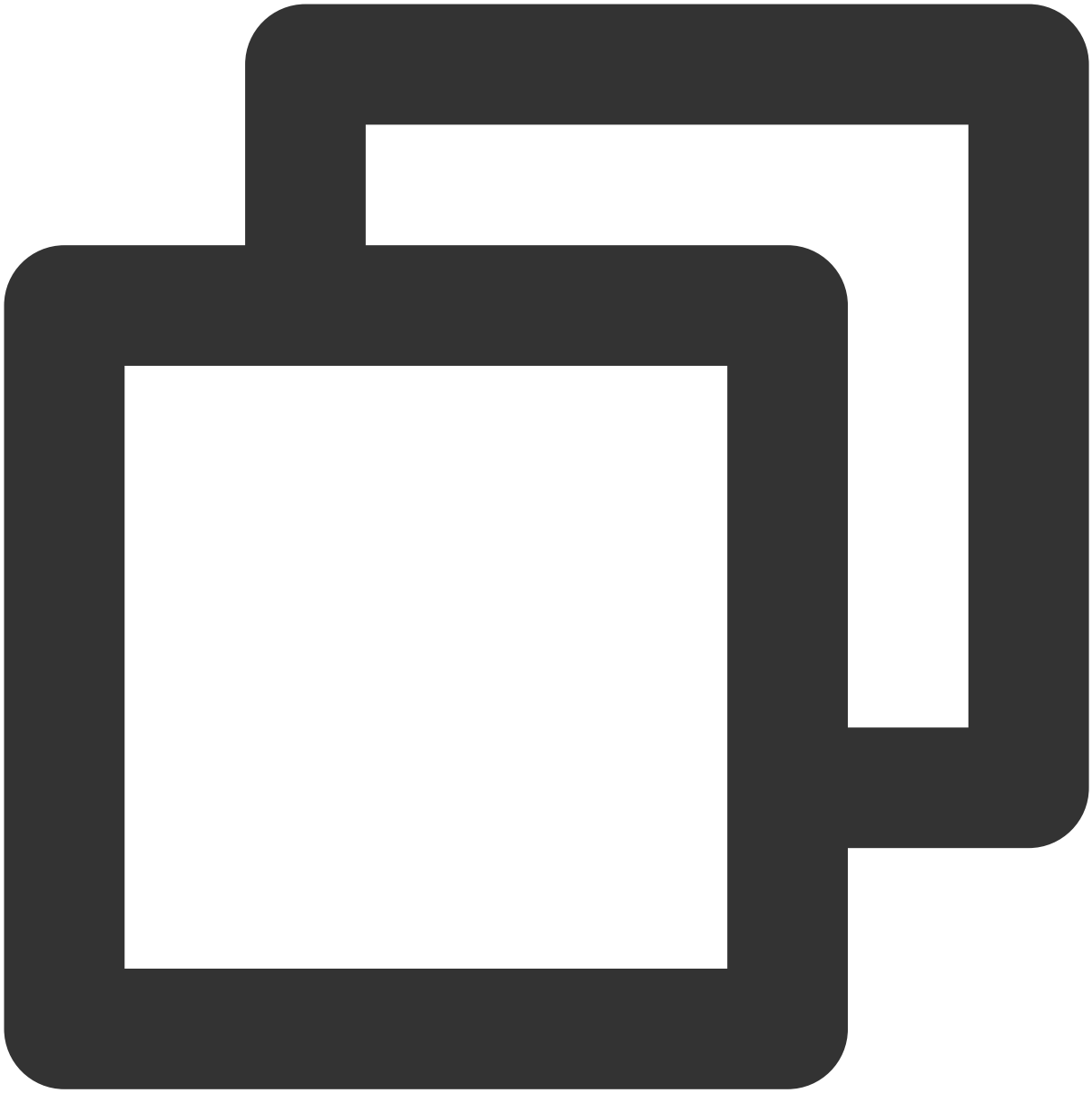
```
create user 'airflowuser'@'%' identified by 'pwd123';
grant all on airflow.* to 'airflowuser'@'%';
flush privileges;
```

4. Modify the configuration in `{AIRFLOW_HOME}/airflow.cfg` .



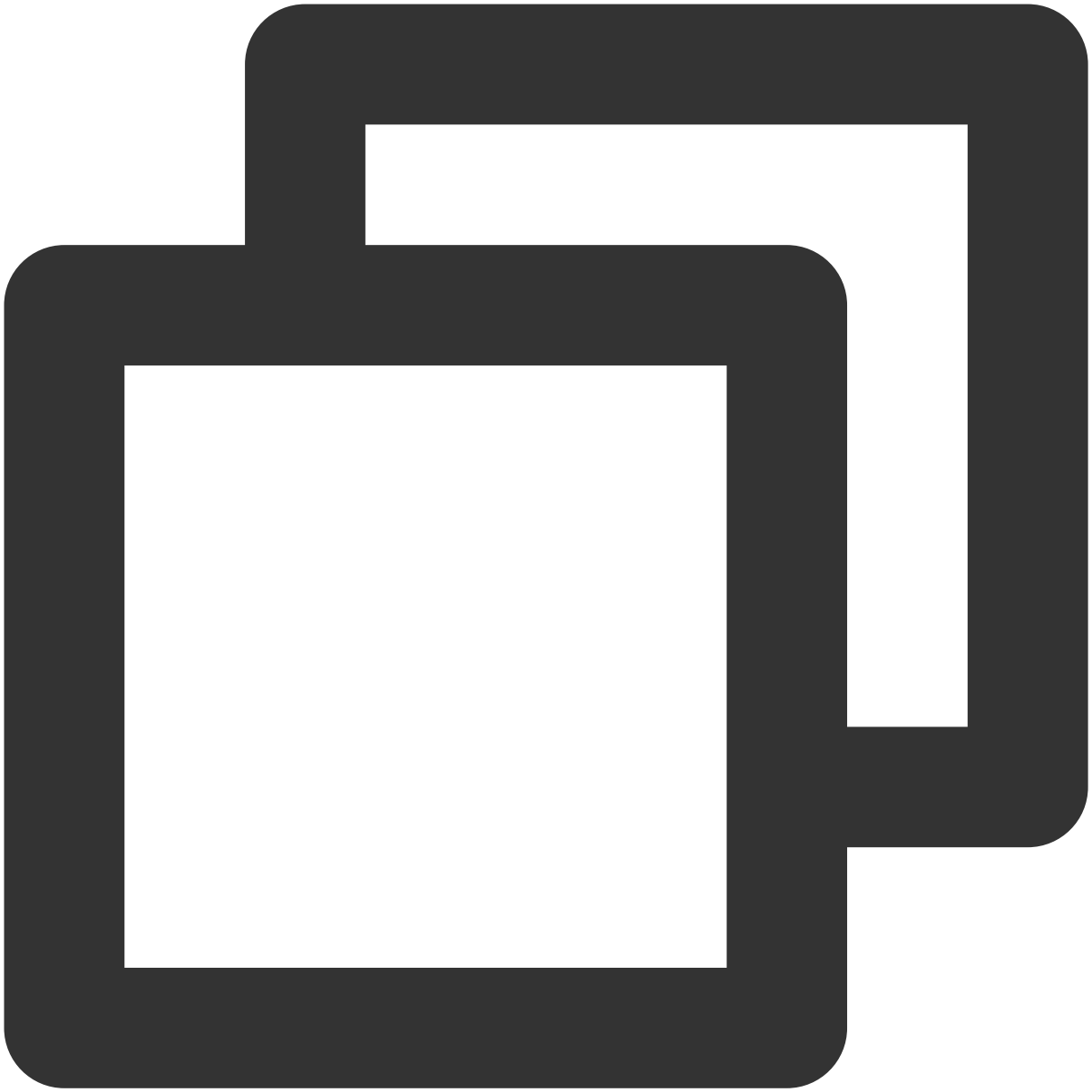
```
sql_alchemy_conn = sqlite:////usr/local/services/airflow/airflow.db
Change to
sql_alchemy_conn = mysql://airflowuser:pwd123@{ip}/airflow
```

5. Reinitialize the database.



```
airflow initdb
```

If you want to retain the data from previous runs, run the following command:



```
airflow resetdb
```