

16 ДЗ - Оркестраторы и DI Tools.

Собираем Dockerfile для clickhouse.

Dockerfile:

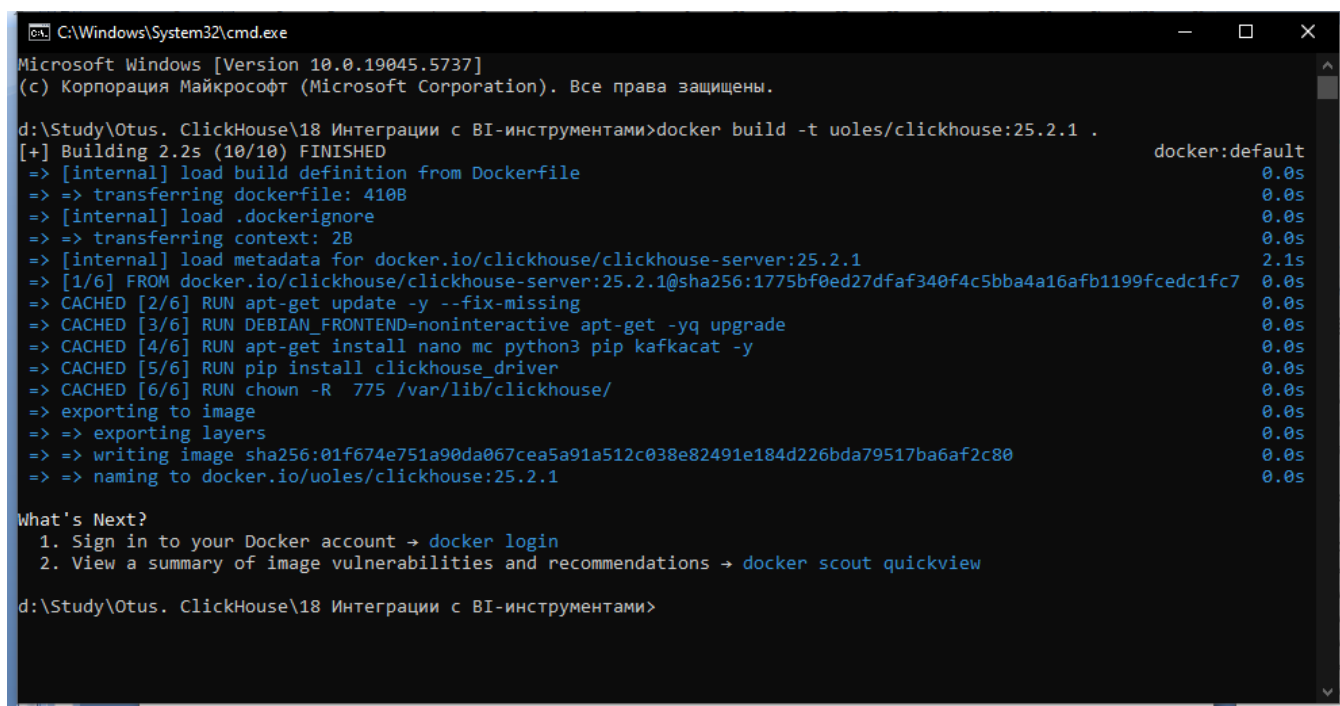
```
FROM clickhouse/clickhouse-server:25.2.1
MAINTAINER Maksim Kulikov max.uoles@rambler.ru
```

```
RUN apt-get update -y --fix-missing
RUN DEBIAN_FRONTEND=noninteractive apt-get -yq upgrade
RUN apt-get install nano mc python3 pip kafkacat -y
RUN pip install clickhouse_driver
```

```
EXPOSE 8123 9000
ENTRYPOINT ["/entrypoint.sh"]
```

Собираем образ командой:

```
docker build -t uoles/clickhouse:25.2.1 .
```



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.5737]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

d:\Study\Otus. ClickHouse\18 Интеграции с BI-инструментами>docker build -t uoles/clickhouse:25.2.1 .
[+] Building 2.2s (10/10) FINISHED
=> [internal] load build definition from Dockerfile                                docker:default
=> => transferring dockerfile: 410B                                              0.0s
=> [internal] load .dockerignore                                                0.0s
=> => transferring context: 2B                                                  0.0s
=> [internal] load metadata for docker.io/clickhouse/clickhouse-server:25.2.1  2.1s
=> [1/6] FROM docker.io/clickhouse/clickhouse-server:25.2.1@sha256:1775bf0ed27dfaf340f4c5bba4a16afb1199fcdcd1fc7 0.0s
=> CACHED [2/6] RUN apt-get update -y --fix-missing                            0.0s
=> CACHED [3/6] RUN DEBIAN_FRONTEND=noninteractive apt-get -yq upgrade          0.0s
=> CACHED [4/6] RUN apt-get install nano mc python3 pip kafkacat -y            0.0s
=> CACHED [5/6] RUN pip install clickhouse_driver                             0.0s
=> CACHED [6/6] RUN chown -R 775 /var/lib/clickhouse/                          0.0s
=> exporting to image                                                         0.0s
=> => exporting layers                                                         0.0s
=> => writing image sha256:01f674e751a90da067cea5a91a512c038e82491e184d226bda79517ba6af2c80 0.0s
=> => naming to docker.io/uoles/clickhouse:25.2.1                             0.0s

What's Next?
 1. Sign in to your Docker account → docker login
 2. View a summary of image vulnerabilities and recommendations → docker scout quickview

d:\Study\Otus. ClickHouse\18 Интеграции с BI-инструментами>
```

Собираем docker-compose для запуска ClickHouse и Dagster.

За основу взял статью <https://dagster.io/blog/interactive-debugging-with-dagster-and-docker>

Docker-compose.yml:

```
name: dagster_clickhouse
services:
  clickhouse-server:
    container_name: uoles-clickhouse-dagster-25
    image: uoles/clickhouse:25.2.1
    environment:
      CLICKHOUSE_DB: my_database
      CLICKHOUSE_USER: username
      CLICKHOUSE_DEFAULT_ACCESS_MANAGEMENT: 1
      CLICKHOUSE_PASSWORD: password
```

ports:
- "18123:8123"
- "19000:9000"

ulimits:
 nofile:
 soft: 262144
 hard: 262144
depends_on:
- code_location
links:
- code_location

db:
 image: postgres:16.4
 environment:
 POSTGRES_USER: user
 POSTGRES_PASSWORD: password
 POSTGRES_DB: postgres
 ports:
 - "5432:5432"

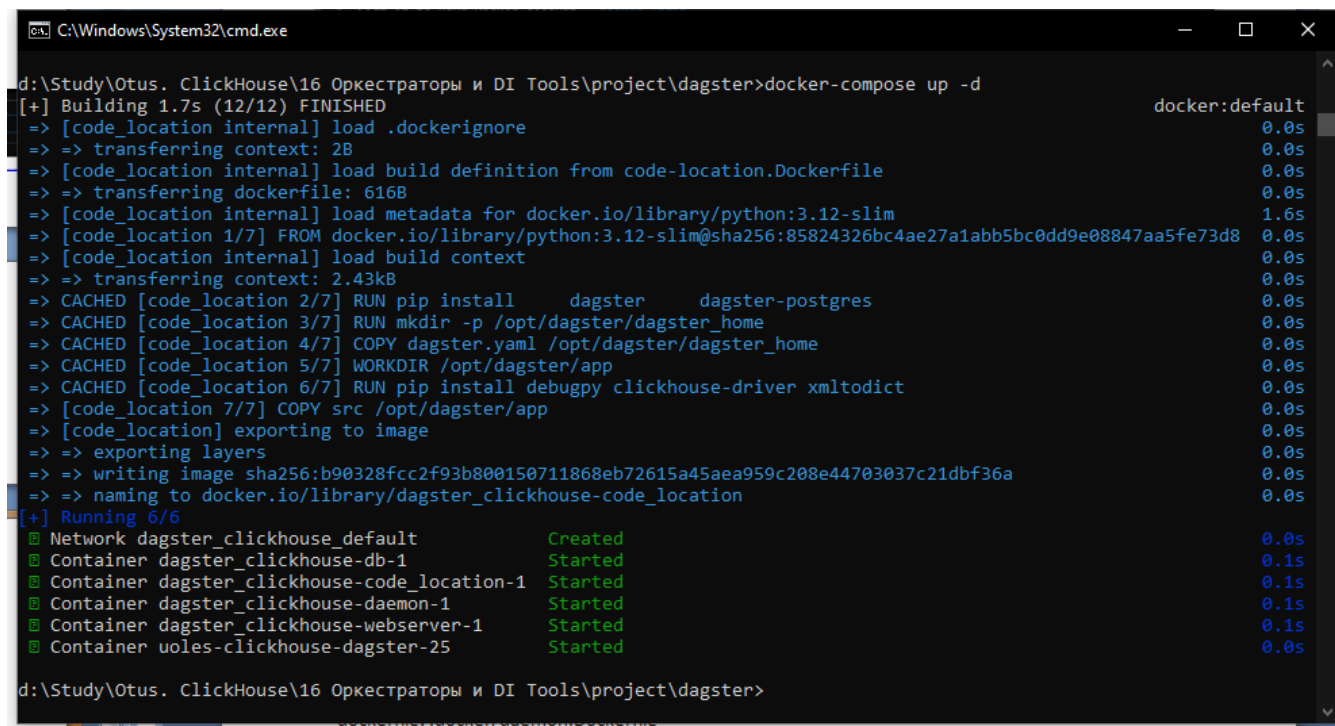
webserver:
 build:
 context: .
 dockerfile: .docker/webserver.Dockerfile
 ports:
 - "3000:3000"
 env_file:
 - .env

code_location:
 build:
 context: .
 dockerfile: .docker/code-location.Dockerfile
 env_file:
 - .env
 ports:
 - "5678:5678"
 entrypoint: python
 command:
 - -m
 - debugpy
 - --listen
 - 0.0.0.0:5678
 - -m
 - dagster
 - code-server
 - start
 - -h
 - 0.0.0.0
 - -p
 - "4000"
 - -m
 - defs

daemon:
 build:

```
context: .
dockerfile: .docker/daemon.Dockerfile
env_file:
- .env
```

Поднимаем приложения командой `docker-compose up -d`:



```
C:\Windows\System32\cmd.exe

d:\Study\Otus. ClickHouse\16 Оркестраторы и DI Tools\project\dagster>docker-compose up -d
[+] Building 1.7s (12/12) FINISHED
=> [code_location internal] load .dockerignore                                0.0s
=> => transferring context: 2B                                                0.0s
=> [code_location internal] load build definition from code-location.Dockerfile 0.0s
=> => transferring dockerfile: 616B                                           0.0s
=> [code_location internal] load metadata for docker.io/library/python:3.12-slim 1.6s
=> [code_location 1/7] FROM docker.io/library/python:3.12-slim@sha256:85824326bc4ae27a1abb5bc0dd9e08847aa5fe73d8 0.0s
=> [code_location internal] load build context                                0.0s
=> => transferring context: 2.43kB                                            0.0s
=> CACHED [code_location 2/7] RUN pip install dagster dagster-postgres        0.0s
=> CACHED [code_location 3/7] RUN mkdir -p /opt/dagster/dagster_home          0.0s
=> CACHED [code_location 4/7] COPY dagster.yaml /opt/dagster/dagster_home     0.0s
=> CACHED [code_location 5/7] WORKDIR /opt/dagster/app                        0.0s
=> CACHED [code_location 6/7] RUN pip install debugpy clickhouse-driver xmldict 0.0s
=> [code_location 7/7] COPY src /opt/dagster/app                             0.0s
=> [code_location] exporting to image                                         0.0s
=> => exporting layers                                                         0.0s
=> => writing image sha256:b90328fcc2f93b800150711868eb72615a45aea959c208e44703037c21dbf36a 0.0s
=> => naming to docker.io/library/dagster_clickhouse-code_location           0.0s
[+] Running 6/6
  Network dagster_clickhouse_default      Created                                0.0s
  Container dagster_clickhouse-db-1       Started                                0.1s
  Container dagster_clickhouse-code_location-1 Started                        0.1s
  Container dagster_clickhouse-daemon-1   Started                                0.1s
  Container dagster_clickhouse-webserver-1 Started                            0.1s
  Container uoles-clickhouse-dagster-25   Started                                0.0s

d:\Study\Otus. ClickHouse\16 Оркестраторы и DI Tools\project\dagster>
```

Скрипты asset'ов.

Скрипты лежат в файлах:

`docker\dagster\src\assets.py`

`docker\dagster\src\defs.py`

Получение данных по курсам валют:

`@asset()`

`def asset_load_xml(context: AssetExecutionContext) -> dict:`

`url = "https://cbr.ru/scripts/XML_daily.asp?date_req=01/01/2020.xml"`

`data = '{}'`

`with requests.get(url) as response:`

`data = xmldict.parse(response.content)`

`context.log.info(f"Output data is: {data}")`

`context.add_output_metadata({"valute": data})`

`return data`

Отправка данных в clickhouse:

`@asset(deps=[asset_load_xml])`

`def asset_add_to_clickhouse(context: AssetExecutionContext, asset_load_xml: dict):`

`data = asset_load_xml`

`context.log.info(f"Prepared data is: {data["ValCurs"]["@name"]}")`

`client = Client(`

`host='clickhouse-server',`

`port=9000,`

`user='username',`

```

password='password',
database='my_database'
)

create_table_query = """
CREATE TABLE IF NOT EXISTS valute (
    c_date String NOT NULL,
    c_name String NOT NULL,
    c_str_id String NOT NULL,
    c_num_code String NOT NULL,
    c_char_code String NOT NULL,
    c_nominal String NOT NULL,
    c_value String NOT NULL
) ENGINE = MergeTree()
ORDER BY (c_date)
"""

client.execute(create_table_query)
context.log.info("Table exists")

c_date = data["ValCurs"]["@Date"]

select_query = "SELECT 1 FROM valute WHERE c_date = %(c_date)s"
result = client.execute(select_query, {'c_date': c_date})

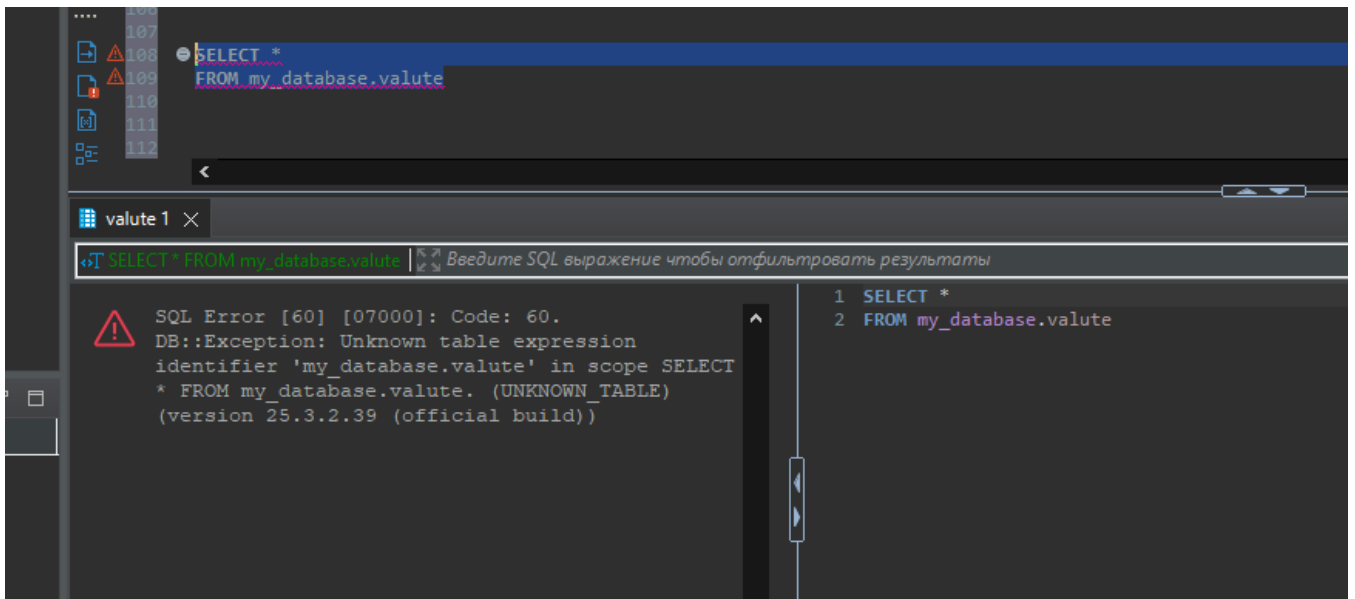
if len(result) == 0:
    insert_query = "INSERT INTO valute (c_date, c_name, c_str_id, c_num_code, c_char_code, c_nominal,
c_value) VALUES"
    insert_data = [
        (data["ValCurs"]["@Date"], item["Name"], item["@ID"], item["NumCode"], item["CharCode"],
item["Nominal"], item["Value"])
        for item in data["ValCurs"]["Valute"]
    ]
    client.execute(insert_query, insert_data)
    context.log.info("Data inserted")
else:
    context.log.info("Data already loaded")

```

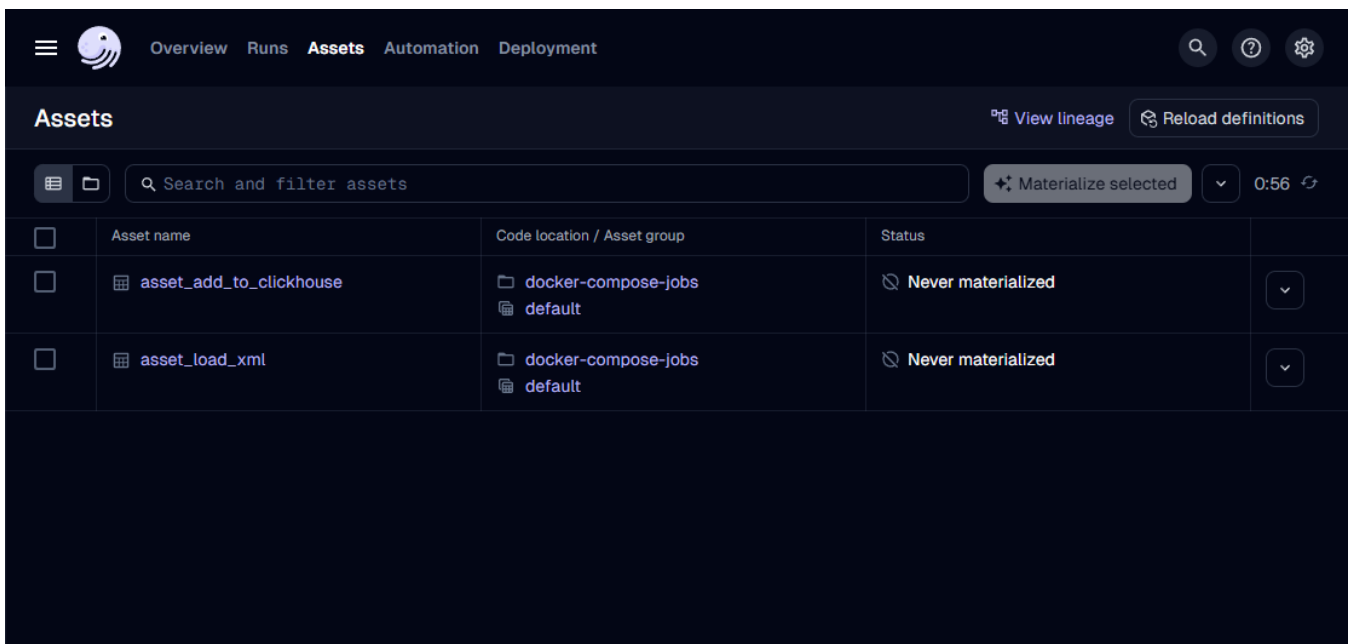
Скрипт отправки данных сначала создает таблицу, если ее нет. Потом проверяет наличие данных по дате из XML. Если даты нет, то добавляет данные.

Запуск скриптов.

Подключаемся к базе и проверяем, что таблицы нет:



Заходим на страницу dagster, на вкладку со скриптами - <http://localhost:3000/assets> .



Заходим сначала в скрипт «asset_load_xml»:



и жмем кнопку «Materialize» для запуска скрипта.

18.04.2025

25.04.2025

Description

Metadata

localhost:3000/runs/de2d3b90-2ef6-4cef-a891-e4c027e61af6

asset

Dagster offers several ways to run data pipelines without manual intervention, including traditional scheduling and event-based

Launched run de2d3b90

View

Появляется сообщение о запуске процесса. Жмем кнопку «View» и переходим к логам выполнения.

Overview

Runs

Assets

Automation

Deployment

Runs / de2d3b90

Success

asset_load_xml

25 apr., 03:18:25

0:00:03

View asset lineage

View tags and config

0.5s

1.0s

1.5s

1.9s

2.0s

2.5s

3.0s

3.5s

asset_load_xml

Hide not started steps

Re-execute all (*)

asset_load_xml

Hide unselected steps

Preparing (0)

No steps are waiting to execute

Executing (0)

No steps are executing

Errored (0)

No steps have errored

Events

stdout

stderr

step:asset_load_xml

Hide non-matches

Levels (6/6)

TIMESTAMP	OP	EVENT TYPE	INFO
03:18:26,001	asset_load_xml	STEP_WORKER_STA_	Launching subprocess for "asset_load_xml".
03:18:27,489	asset_load_xml	STEP_WORKER_STA_	Executing step "asset_load_xml" in subprocess.
03:18:27,795	asset_load_xml	RESOURCE_INIT_S_	Starting initialization of resources [io_manager].
03:18:27,815	asset_load_xml	RESOURCE_INIT_S_	Finished initialization of resources [io_manager].
03:18:27,840	asset_load_xml	STEP_START	Started execution of step "asset_load_xml".
03:18:27,931	asset_load_xml	INFO	Output data is: {'ValCurs': {'@Date': '01.01.2020', '@name': 'Foreign Currency Market', 'Valute': [{'@ID': 'R01010', 'NumCode': '036', 'CharCode': 'AUD', 'Nominal': '1', 'Name': 'Австралийский доллар', 'Value': '43,3835', 'VunitRate': '43,3835'}, {'@ID': 'R01020A', 'NumCode': '944', 'CharCode': 'AZN', 'Nominal': '1', 'Name': 'Азербайджанский манат', 'Value': '36,4902', 'VunitRate': '36,4902'}, {'@ID': 'R01035', 'NumCode': '826', 'CharCode': 'GBP', 'Nominal': '1', 'Name': 'Фунт стерлингов', 'Value': '81,3069', 'VunitRate': '81,3069'}, {'@ID': 'R01060', 'NumCode': '051', 'CharCode': 'AMD', 'Nominal': '100', 'Name': 'Армянских драмов', 'Value': '12,9239', 'VunitRate': '0,129239'}, {'@ID': 'R01090B', 'NumCode': '933', 'CharCode': 'BYN', 'Nominal': '1', 'Name': 'Белорусский рубль', 'Value': '29,3545', 'VunitRate': '29,3545'}, {'@ID': 'R01100', 'NumCode': '975', 'CharCode': 'BGN', 'Nominal': '100', 'Name': 'Болгарский лев', 'Value': '35,4720', 'VunitRate': '35,472'}, {'@ID': 'R010986', 'CharCode': 'BRL', 'Nominal': '1', 'Name': 'Бразильский реал', 'Value': '15,4006', 'VunitRate': '0,154006'}, {'@ID': 'R01135', 'NumCode': '348', 'Name': 'Бразильский реал', 'Value': '15,4006', 'VunitRate': '0,154006'}]}
03:18:27,941	asset_load_xml	STEP_OUTPUT	Yielded output "result" of type "dict". (Type check passed). valute [Show JSON]
03:18:27,952	asset_load_xml	DEBUG	Writing file at: /opt/dagster/dagster_home/storage/asset_load_xml using PickledObjectFilesystemIOManager...

Видно, что скрипт отработал успешно и данные подтянулись. Далее возвращаемся к списку скриптов и запускаем так же второй скрипт «asset_add_to_clickhouse» для отправки данных в clickhouse.

Overview

Runs

Assets

Automation

Deployment

Assets

View lineage

Reload definitions

Search and filter assets

Materialize selected

0:37

	Asset name	Code location / Asset group	Status	
<input type="checkbox"/>	asset_add_to_clickhouse	docker-compose-jobs default	Never materialized	<div></div>
<input type="checkbox"/>	asset_load_xml	docker-compose-jobs default	Materialized 25 apr., 03:18	<div></div>

Можно запустить скрипты так же и с вкладки «Assets».

OverviewRunsAssetsAutomationDeployment

Runs / c656b9a7

Successasset_add_to_clickhouse25 apr, 03:24:120:00:03View asset lineageView tags and config

Hide not started steps

Re-execute all (*)

0.5s1.0s1.5s2.0s2.5s3.0s3.5s

asset

Search and filter stepsHide unselected steps

Preparing (0)
No steps are waiting to execute
Executing (0)
No steps are executing
Errored (0)
No steps have errored

EventsstdoutstderrFilter...Levels (6/6)

TIMESTAMP	OP	EVENT TYPE	INFO
03:24:08,261	-	RUN_ENQUEUED	
03:24:10,879	-	RUN_STARTING	
03:24:12,534	-	ENGINE_EVENT	Started process for run (pid: 130). pid 130
03:24:12,872	-	RUN_START	Started execution of run for "__ASSET_JOB".
03:24:12,928	-	ENGINE_EVENT	Executing steps using multiprocessing executor: parent process (pid: 130) pid 130 step_keys ['asset_add_to_clickhouse']
03:24:12,940	asset_add_to_clickhouse	STEP_WORKER_STA_	Launching subprocess for "asset_add_to_clickhouse".
03:24:14,400	asset_add_to_clickhouse	STEP_WORKER_STA_	Executing step "asset_add_to_clickhouse" in subprocess.
03:24:14,733	asset_add_to_clickhouse	RESOURCE_INIT_S_	Starting initialization of resources [io_manager].
03:24:14,756	asset_add_to_clickhouse	RESOURCE_INIT_S_	Finished initialization of resources [io_manager].
03:24:14,771	-	LOGS_CAPTURED	Started capturing logs in process (pid: 141).

Скрипт отработал успешно.

03:24:14,781	asset_add_to_clickhouse	STEP_START	Started execution of step "asset_add_to_clickhouse".
03:24:14,868	asset_add_to_clickhouse	DEBUG	Loading file from: /opt/dagster/dagster_home/storage/asset_load.xml using PickledObjectFilesystemIOManager...
03:24:14,878	asset_add_to_clickhouse	LOADED_INPUT	Loaded input "asset_load.xml" using input manager "io_manager"
03:24:14,887	asset_add_to_clickhouse	STEP_INPUT	Got input "asset_load.xml" of type "dict". (Type check passed).
03:24:14,897	asset_add_to_clickhouse	INFO	Prepared data is: Foreign Currency Market
03:24:14,918	asset_add_to_clickhouse	INFO	Table exists
03:24:14,937	asset_add_to_clickhouse	INFO	Data inserted
03:24:14,948	asset_add_to_clickhouse	STEP_OUTPUT	Yielded output "result" of type "Any". (Type check passed).
03:24:14,958	asset_add_to_clickhouse	DEBUG	Writing file at: /opt/dagster/dagster_home/storage/asset_add_to_clickhouse using PickledObjectFilesystemIOManager...
03:24:14,966	asset_add_to_clickhouse	ASSET_MATERIALI_	Materialized value asset_add_to_clickhouse. asset_key asset_add_to_clickhouse [View Asset]
03:24:15,001	asset_add_to_clickhouse	HANDLED_OUTPUT	Handled output "result" using IO manager "io_manager"
03:24:15,010	asset_add_to_clickhouse	STEP_SUCCESS	Finished execution of step "asset_add_to_clickhouse" in 219ms.

Проверяем наличие данных в clickhouse:

107

108

109

110

111

112

SELECT *

FROM my_database.valute

valute 1

SELECT * FROM my_database.valute

Введите SQL выражение чтобы отфильтровать результаты

Таблица

Текст

Запись

	A-Z c_date	A-Z c_name	A-Z c_str_id	A-Z c_num_code	A-Z c_char_code	A-Z c_nominal	A-Z c_value
1	01.01.2020	Австралийский доллар	R01010	036	AUD	1	43,3835
2	01.01.2020	Азербайджанский манат	R01020A	944	AZN	1	36,4902
3	01.01.2020	Фунт стерлингов	R01035	826	GBP	1	81,3069
4	01.01.2020	Армянских драмов	R01060	051	AMD	100	12,9239
5	01.01.2020	Белорусский рубль	R01090B	933	BYN	1	29,3545
6	01.01.2020	Болгарский лев	R01100	975	BGN	1	35,4720
7	01.01.2020	Бразильский реал	R01115	986	BRL	1	15,4006
8	01.01.2020	Форинтов	R01135	348	HUF	100	20,9637
9	01.01.2020	Гонконгских долларов	R01200	344	HKD	10	79,5029
10	01.01.2020	Датских крон	R01215	208	DKK	10	92,8776
11	01.01.2020	Доллар США	R01235	840	USD	1	61,9057
12	01.01.2020	Евро	R01239	978	EUR	1	69,3777
13	01.01.2020	Индийских рупий	R01270	356	INR	100	86,8255