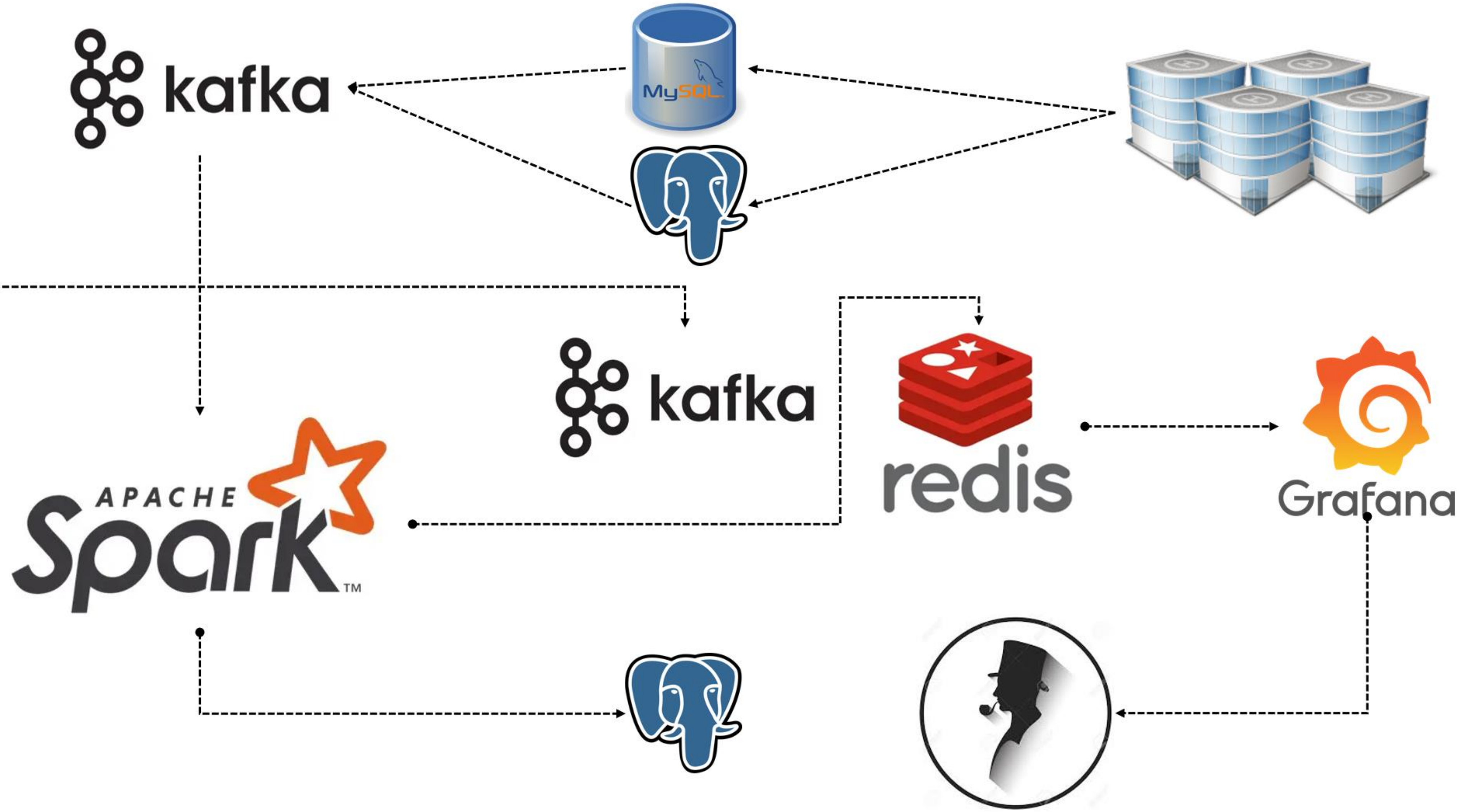
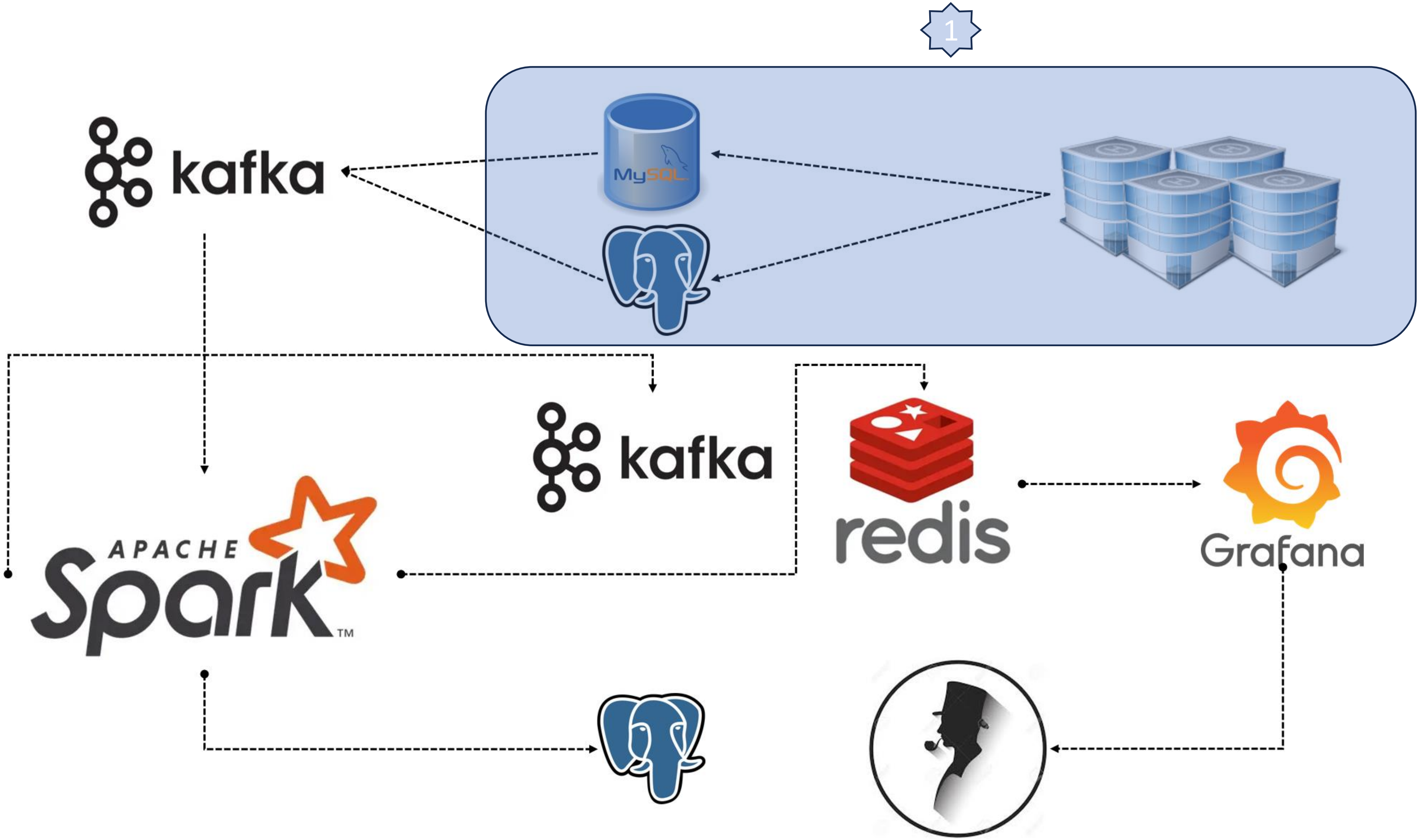
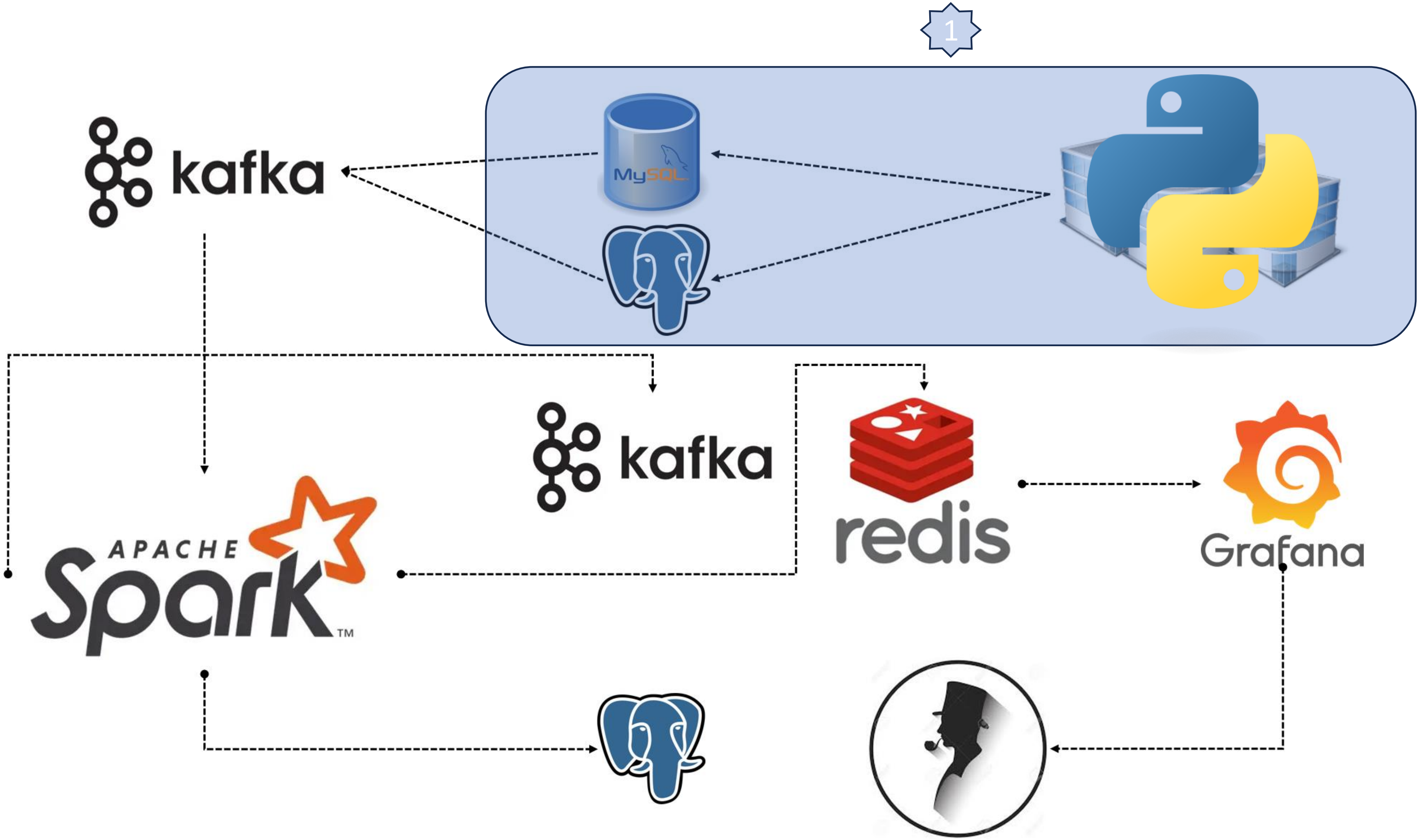
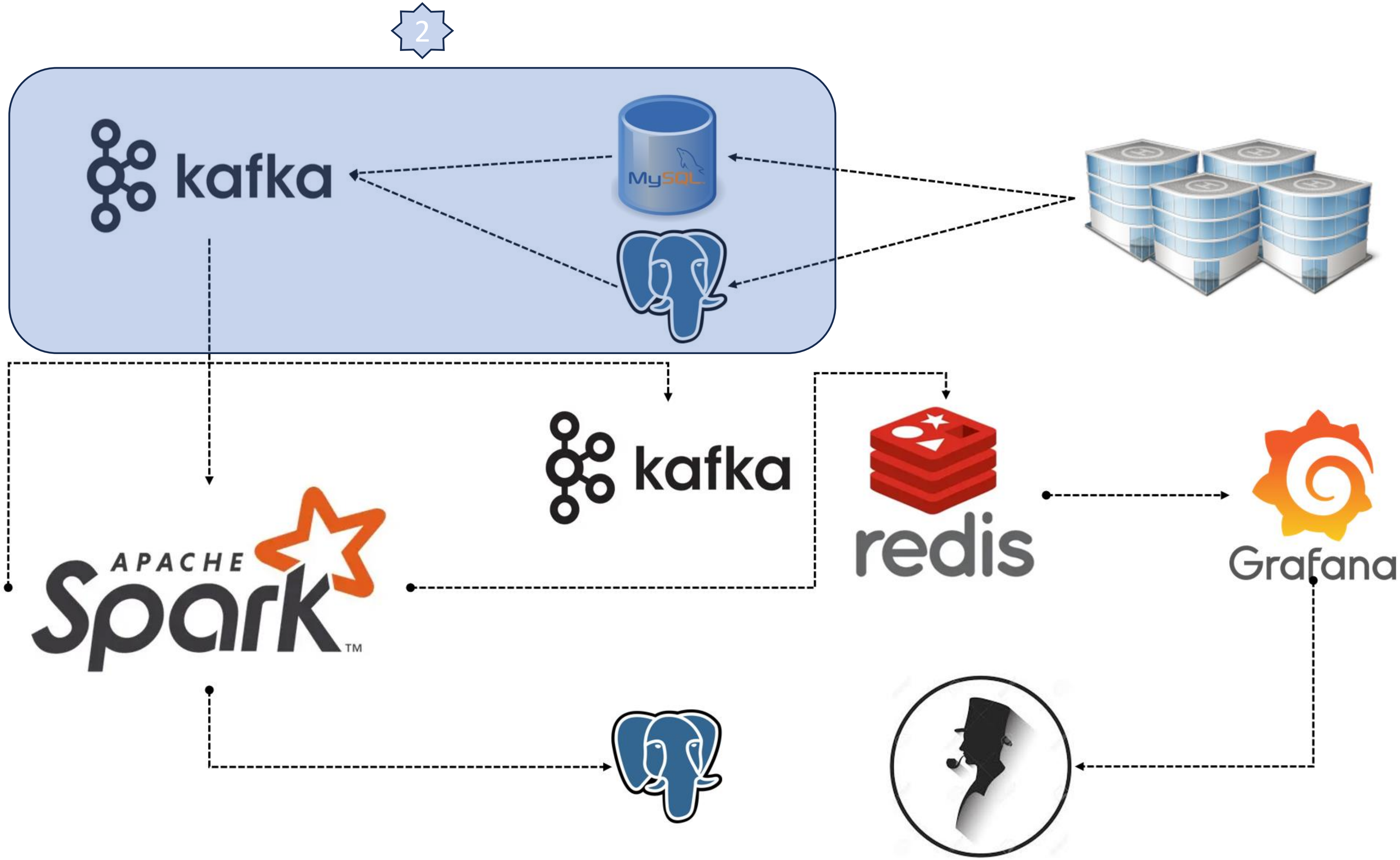
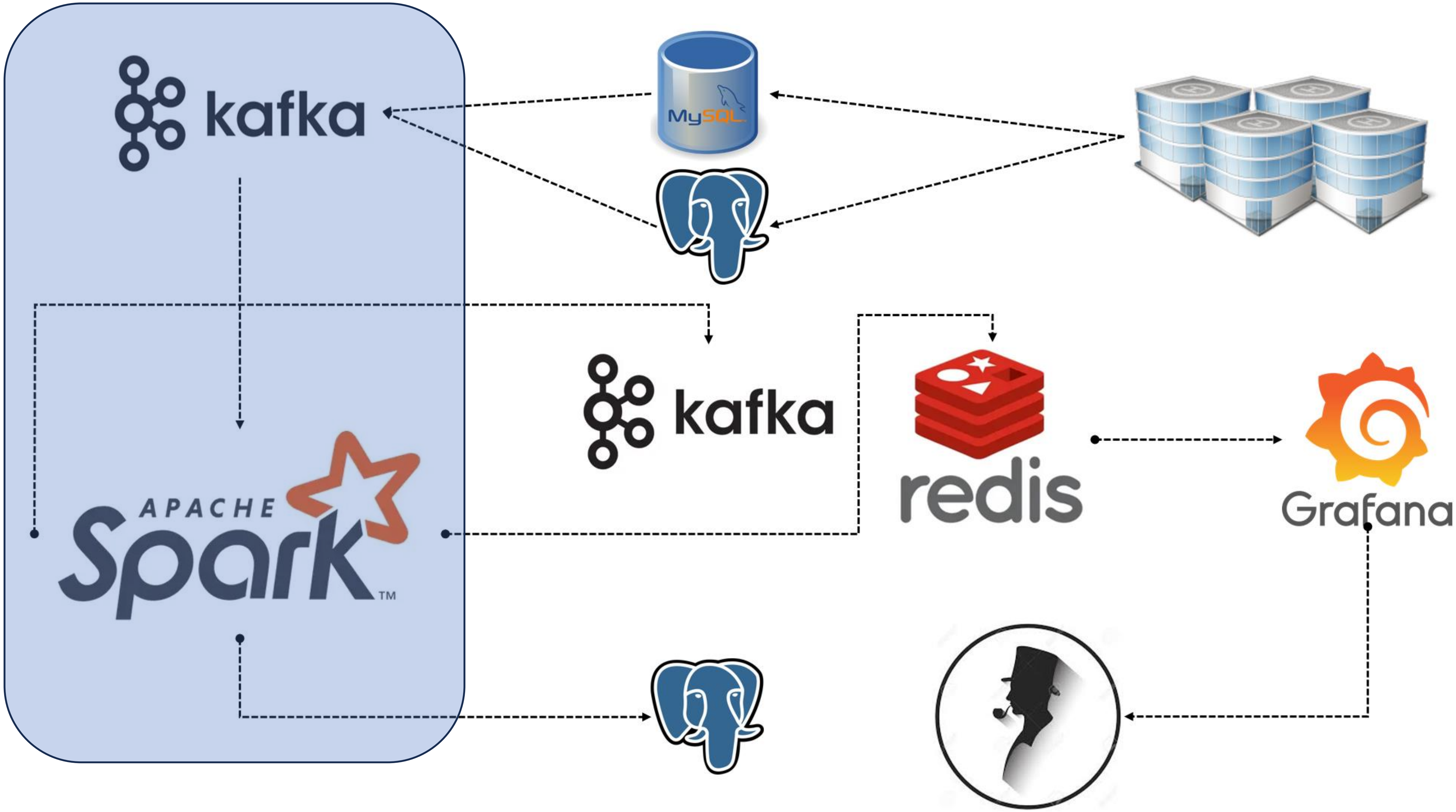

kafka
MySQL
kafka
APACHE Spark™
redis
Grafana

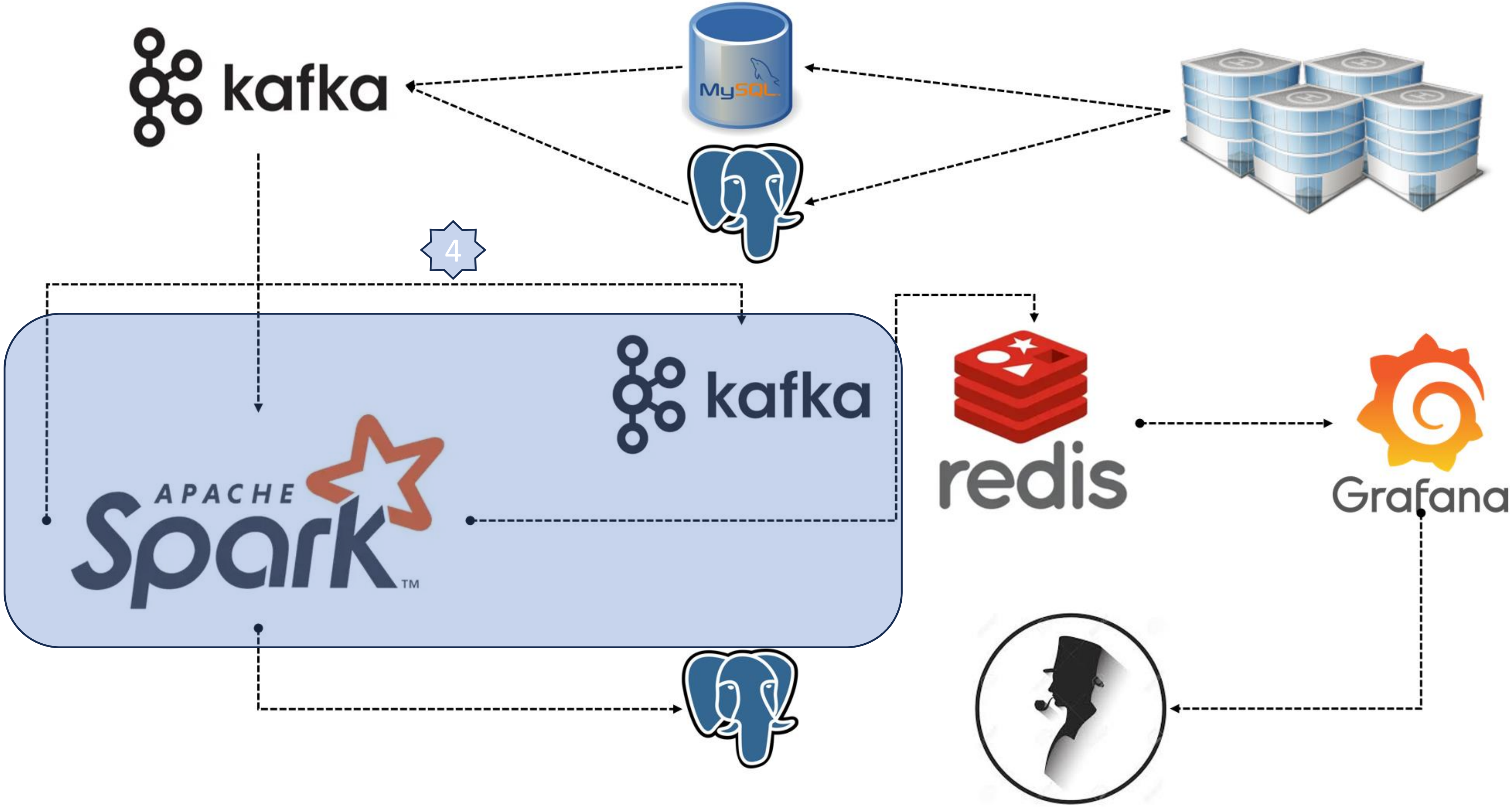

1
kafka
MySQL
kafka
APACHE
Spark
redis
Grafana

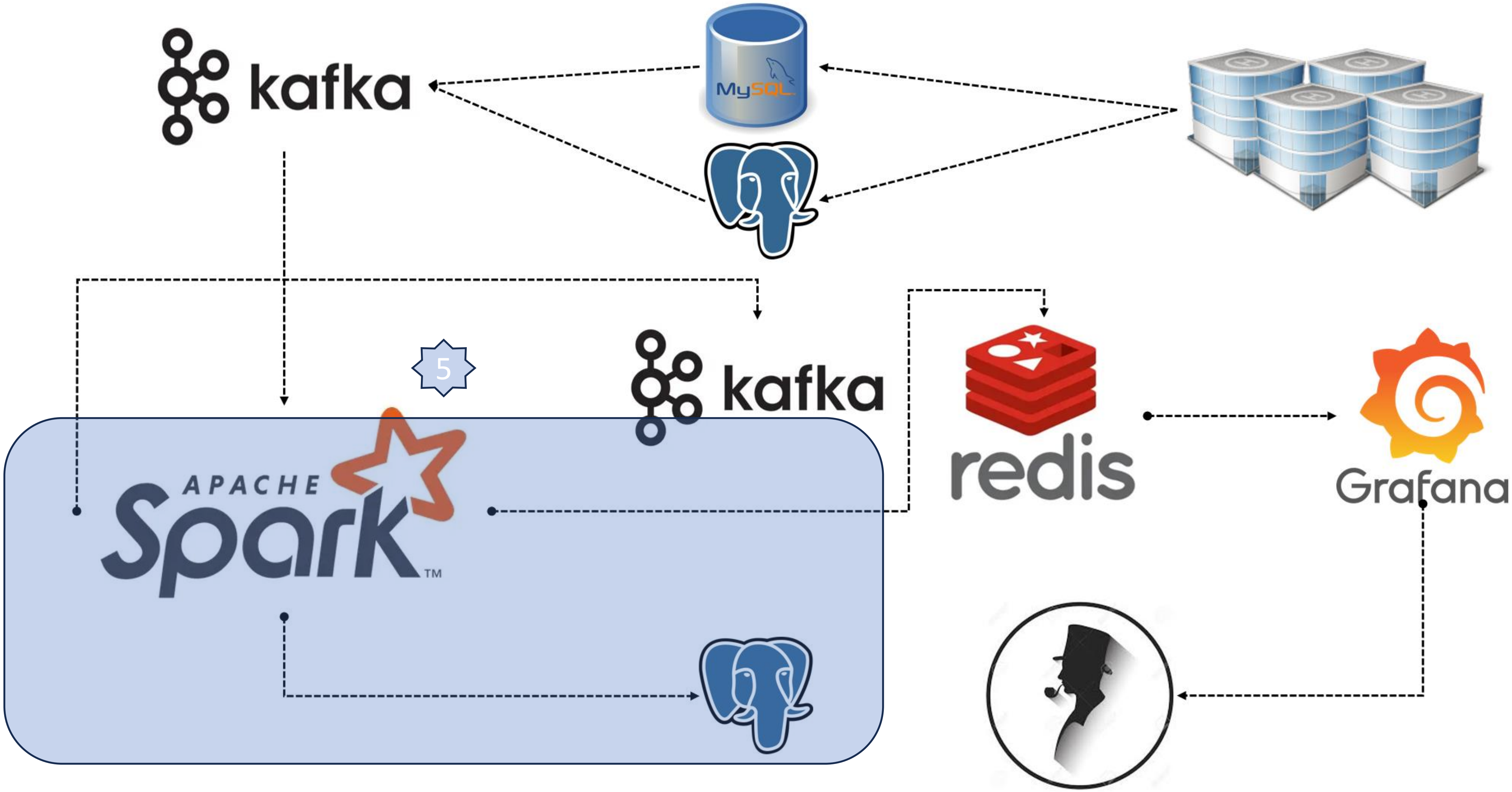

1
kafka
MySQL
kafka
APACHE Spark
redis
Grafana

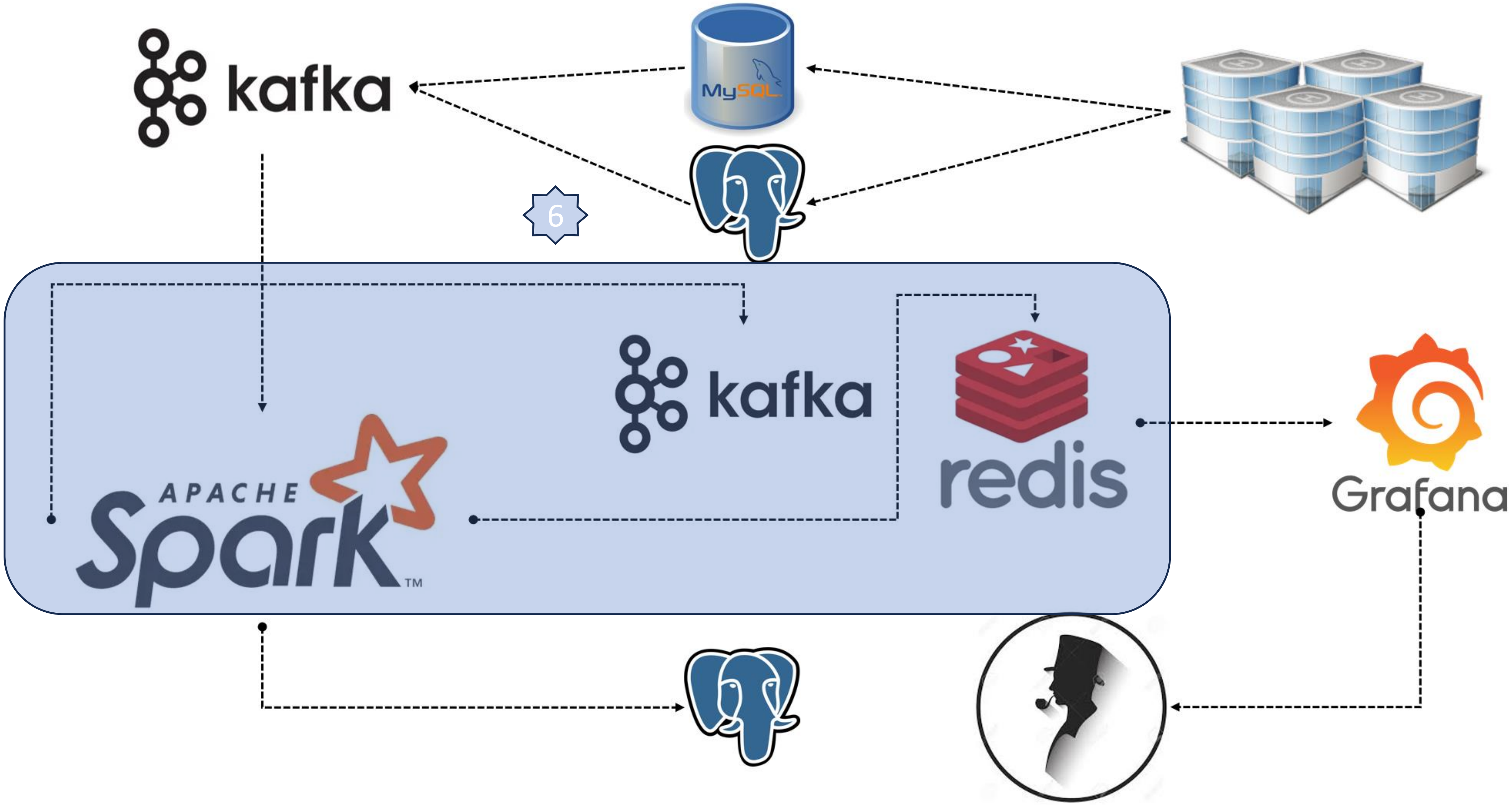

2
kafka
MySQL
kafka
APACHE Spark
redis
Grafana

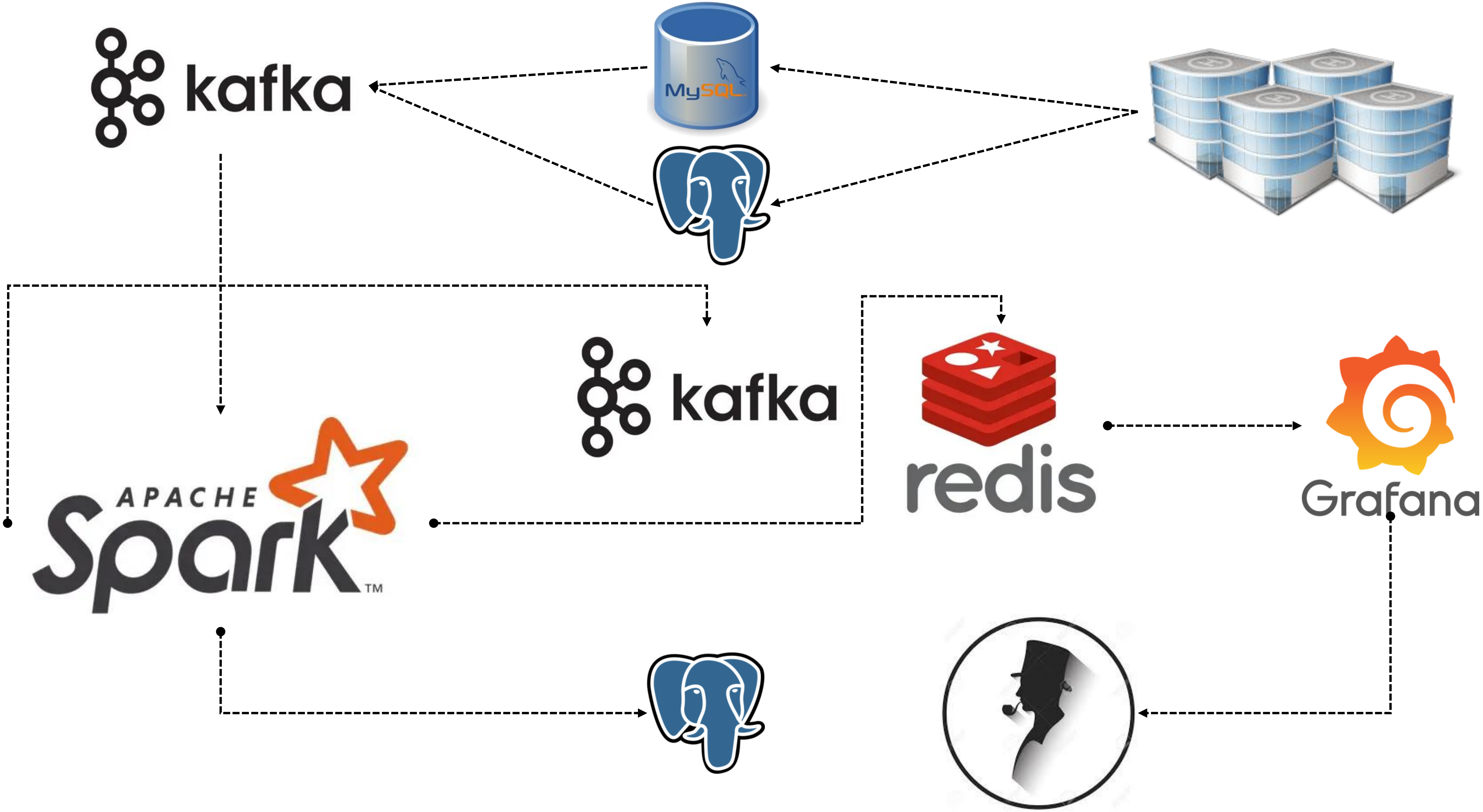
3

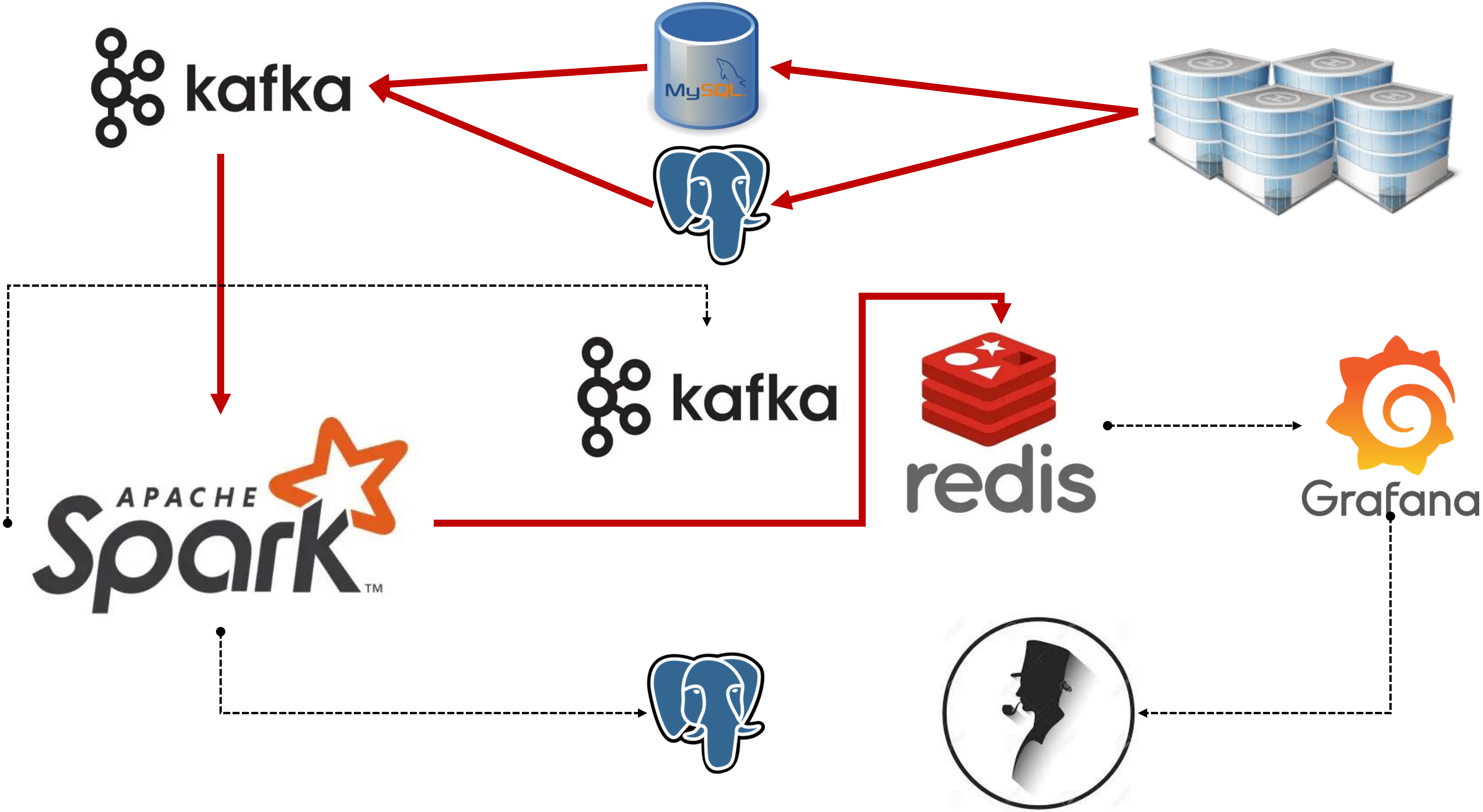


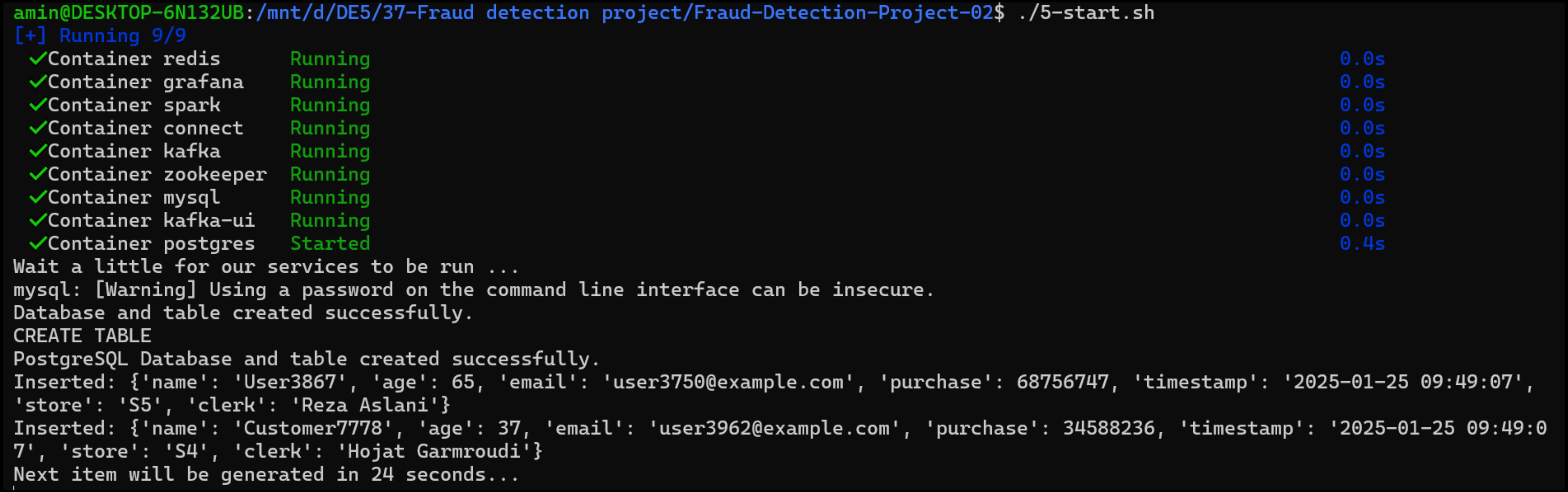

kafka
MySQL
4
kafka
APACHE
Spark
redis
Grafana


kafka
MySQL
5
kafka
APACHE
Spark
redis
Grafana

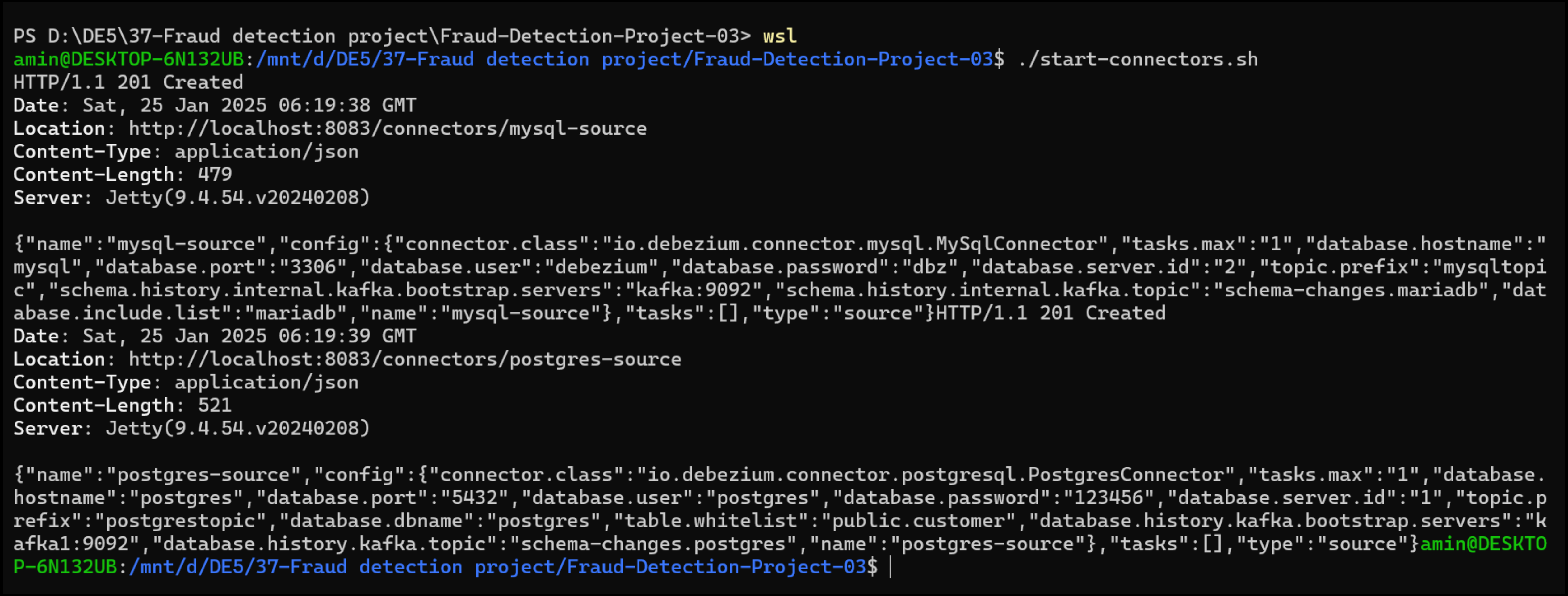

kafka
MySQL
6
kafka
APACHE
Spark
redis
Grafana


kafka
MySQL
kafka
redis
Grafana
APACHE
Spark
TM

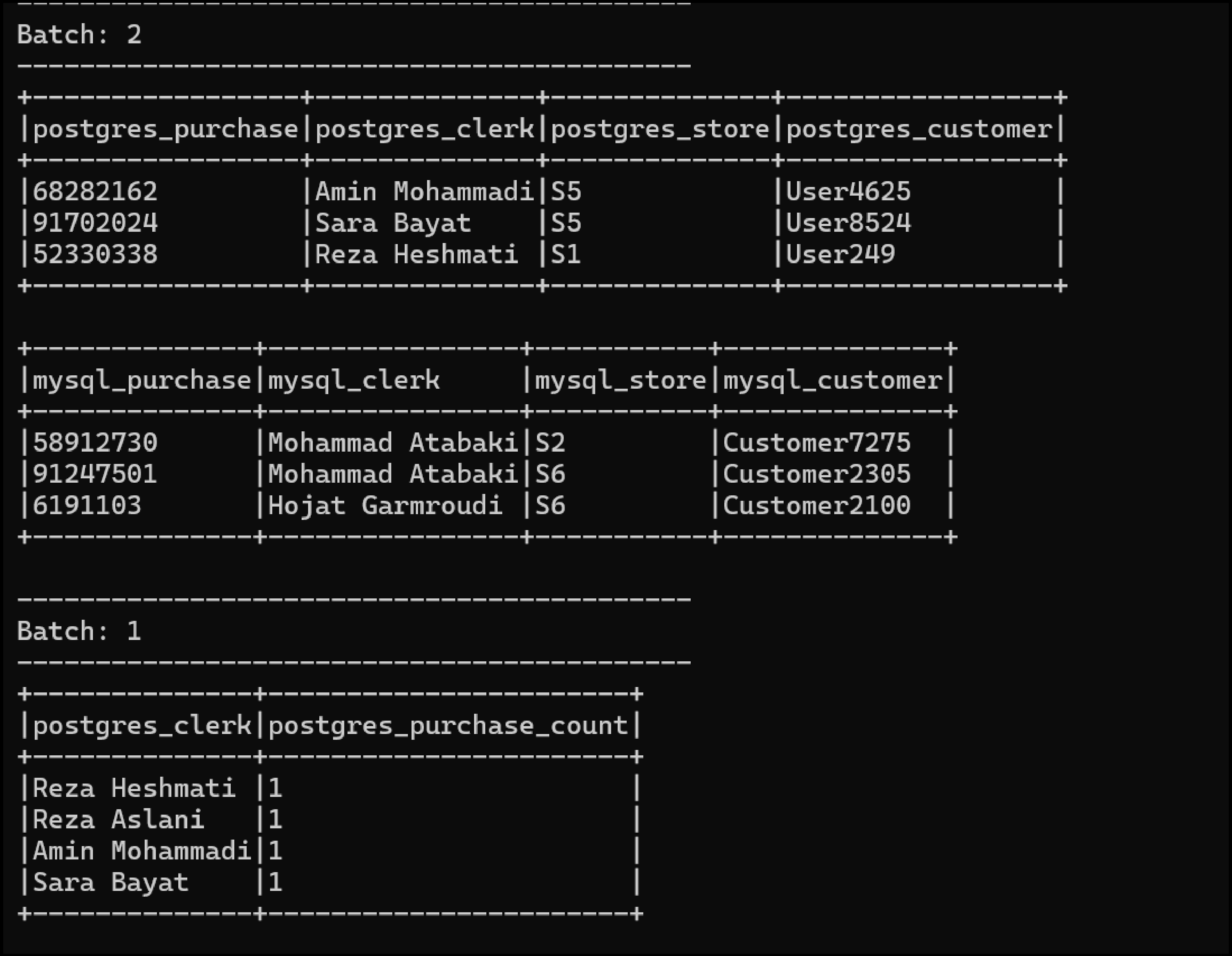

kafka
MySQL
kafka
redis
Grafana
APACHE
Spark
TM



```
amin@DESKTOP-6N132UB:/mnt/d/DE5/37-Fraud detection project/Fraud-Detection-Project-02$ ./5-start.sh
[+] Running 9/9
  ✓Container redis      Running                                                                                    0.0s
  ✓Container grafana    Running                                                                                    0.0s
  ✓Container spark      Running                                                                                    0.0s
  ✓Container connect    Running                                                                                    0.0s
  ✓Container kafka      Running                                                                                    0.0s
  ✓Container zookeeper  Running                                                                                    0.0s
  ✓Container mysql      Running                                                                                    0.0s
  ✓Container kafka-ui   Running                                                                                    0.0s
  ✓Container postgres   Started                                                                                    0.4s
Wait a little for our services to be run ...
mysql: [Warning] Using a password on the command line interface can be insecure.
Database and table created successfully.
CREATE TABLE
PostgreSQL Database and table created successfully.
Inserted: {'name': 'User3867', 'age': 65, 'email': 'user3750@example.com', 'purchase': 68756747, 'timestamp': '2025-01-25 09:49:07',
'store': 'S5', 'clerk': 'Reza Aslani'}
Inserted: {'name': 'Customer7778', 'age': 37, 'email': 'user3962@example.com', 'purchase': 34588236, 'timestamp': '2025-01-25 09:49:0
7', 'store': 'S4', 'clerk': 'Hojat Garmroudi'}
Next item will be generated in 24 seconds...
```



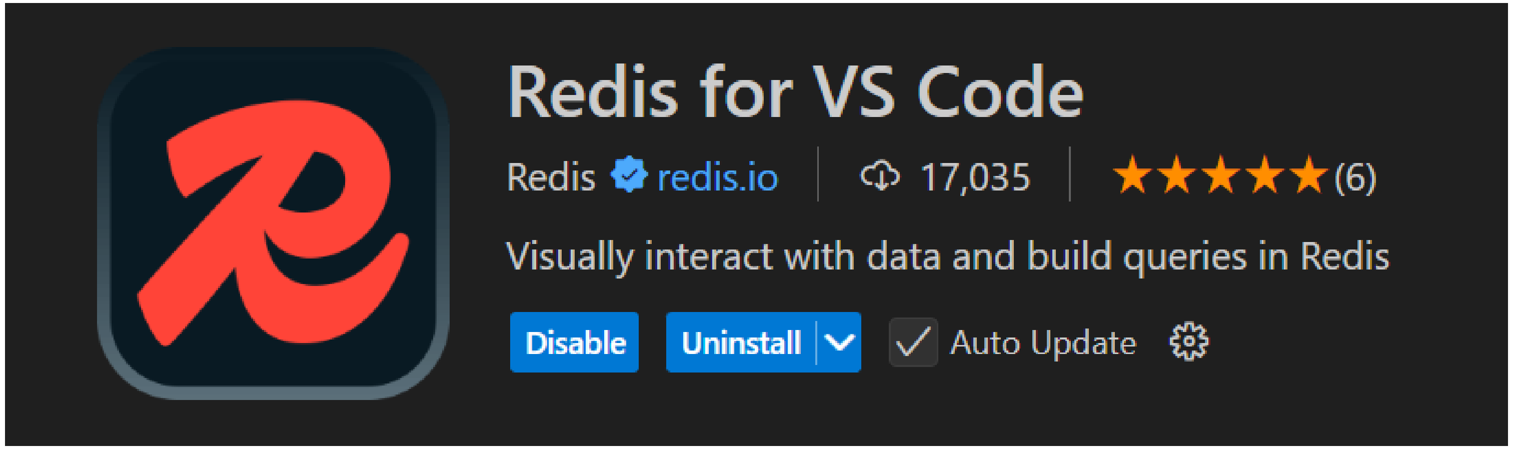
```
PS D:\DE5\37-Fraud detection project\Fraud-Detection-Project-03> wsl
amin@DESKTOP-6N132UB:/mnt/d/DE5/37-Fraud detection project/Fraud-Detection-Project-03$ ./start-connectors.sh
HTTP/1.1 201 Created
Date: Sat, 25 Jan 2025 06:19:38 GMT
Location: http://localhost:8083/connectors/mysql-source
Content-Type: application/json
Content-Length: 479
Server: Jetty(9.4.54.v20240208)

{"name":"mysql-source","config":{"connector.class":"io.debezium.connector.mysql.MySqlConnector","tasks.max":"1","database.hostname":"
mysql","database.port":"3306","database.user":"debezium","database.password":"dbz","database.server.id":"2","topic.prefix":"mysqltopi
c","schema.history.internal.kafka.bootstrap.servers":"kafka:9092","schema.history.internal.kafka.topic":"schema-changes.mariadb","dat
abase.include.list":"mariadb","name":"mysql-source"},"tasks":[],"type":"source"}HTTP/1.1 201 Created
Date: Sat, 25 Jan 2025 06:19:39 GMT
Location: http://localhost:8083/connectors/postgres-source
Content-Type: application/json
Content-Length: 521
Server: Jetty(9.4.54.v20240208)

{"name":"postgres-source","config":{"connector.class":"io.debezium.connector.postgresql.PostgresConnector","tasks.max":"1","database.
hostname":"postgres","database.port":"5432","database.user":"postgres","database.password":"123456","database.server.id":"1","topic.p
refix":"postgrestopic","database.dbname":"postgres","table.whitelist":"public.customer","database.history.kafka.bootstrap.servers":"k
afka1:9092","database.history.kafka.topic":"schema-changes.postgres","name":"postgres-source"},"tasks":[],"type":"source"}amin@DESKTO
P-6N132UB:/mnt/d/DE5/37-Fraud detection project/Fraud-Detection-Project-03$
```



```
-------------------------------------------
Batch: 2
-------------------------------------------
+-----------------+--------------+--------------+-----------------+
|postgres_purchase|postgres_clerk|postgres_store|postgres_customer|
+-----------------+--------------+--------------+-----------------+
|68282162         |Amin Mohammadi|S5            |User4625         |
|91702024         |Sara Bayat    |S5            |User8524         |
|52330338         |Reza Heshmati |S1            |User249          |
+-----------------+--------------+--------------+-----------------+

+--------------+----------------+-----------+--------------+
|mysql_purchase|mysql_clerk     |mysql_store|mysql_customer|
+--------------+----------------+-----------+--------------+
|58912730      |Mohammad Atabaki|S2         |Customer7275  |
|91247501      |Mohammad Atabaki|S6         |Customer2305  |
|6191103       |Hojat Garmroudi |S6         |Customer2100  |
+--------------+----------------+-----------+--------------+

-------------------------------------------
Batch: 1
-------------------------------------------
+--------------+-----------------------+
|postgres_clerk|postgres_purchase_count|
+--------------+-----------------------+
|Reza Heshmati |1                      |
|Reza Aslani   |1                      |
|Amin Mohammadi|1                      |
|Sara Bayat    |1                      |
+--------------+-----------------------+
```

Redis is like a super-fast storage box for your data. Here’s why it’s useful:

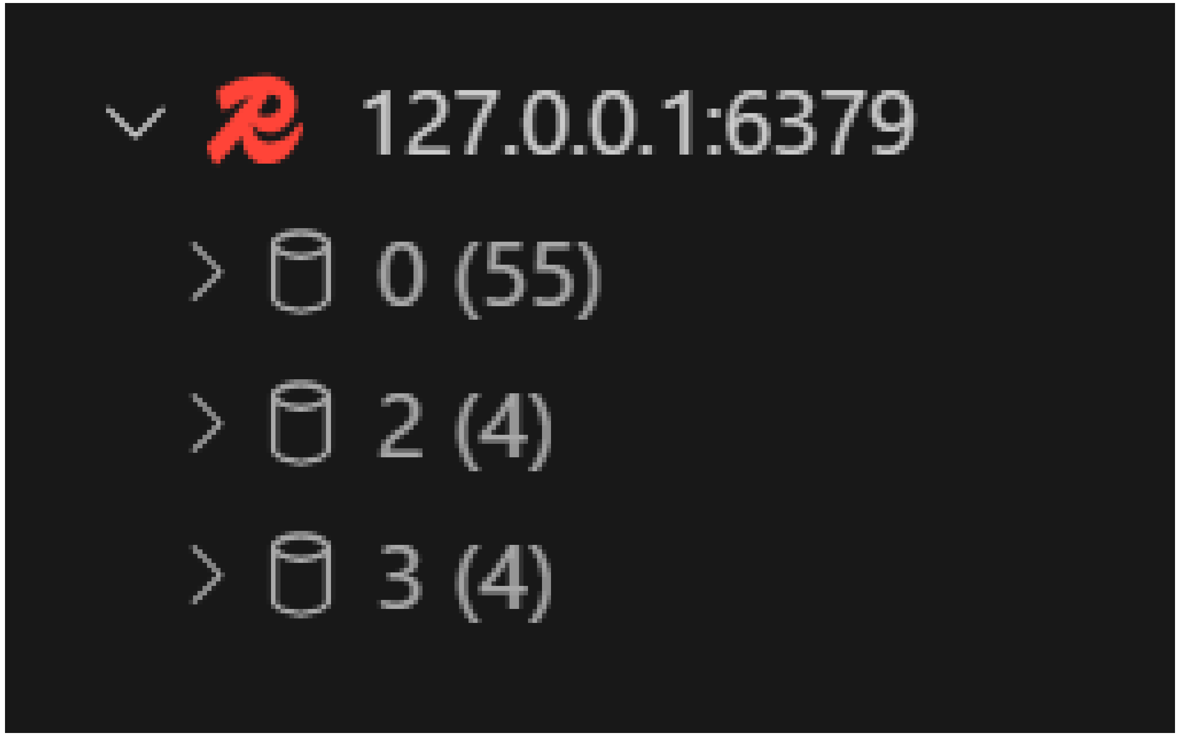
- **Speed**: It stores data in memory (RAM), so it’s much faster than reading from a hard drive.
- **Simple**: It’s easy to use for storing and retrieving things like user sessions, cache, or temporary data.
- **Versatile**: It can handle different types of data (strings, lists, sets, etc.) and is great for real-time apps.
- **Scalable**: It helps your app handle more users by offloading work from your main database.

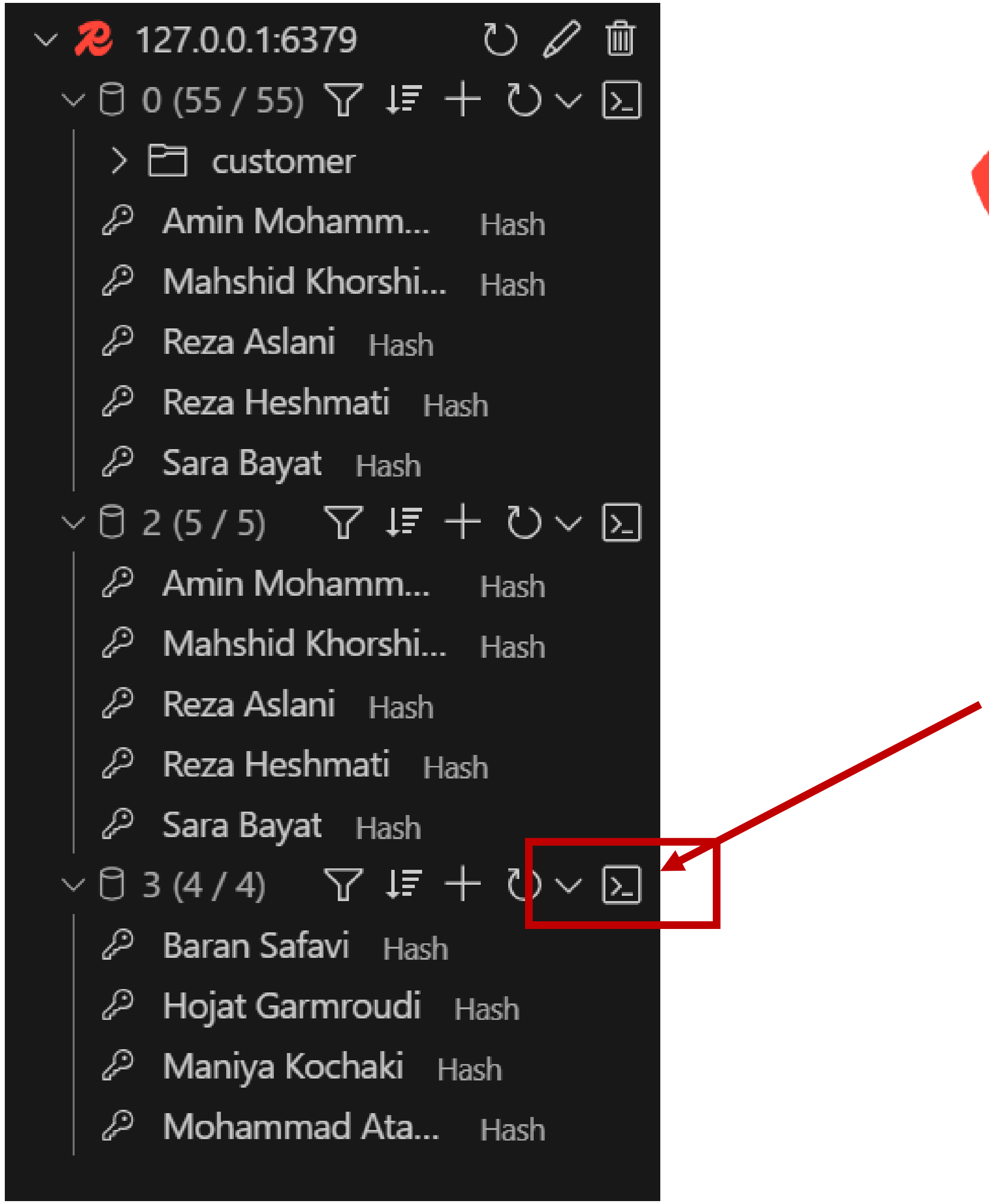


Redis for VS Code

Redis redis.io | 17,035 | ★★★★★ (6)

Visually interact with data and build queries in Redis





127.0.0.1:6379

0 (55 / 55)

customer

Amin Mohamm... Hash

Mahshid Khorshi... Hash

Reza Aslani Hash

Reza Heshmati Hash

Sara Bayat Hash

2 (5 / 5)

Amin Mohamm... Hash

Mahshid Khorshi... Hash

Reza Aslani Hash

Reza Heshmati Hash

Sara Bayat Hash

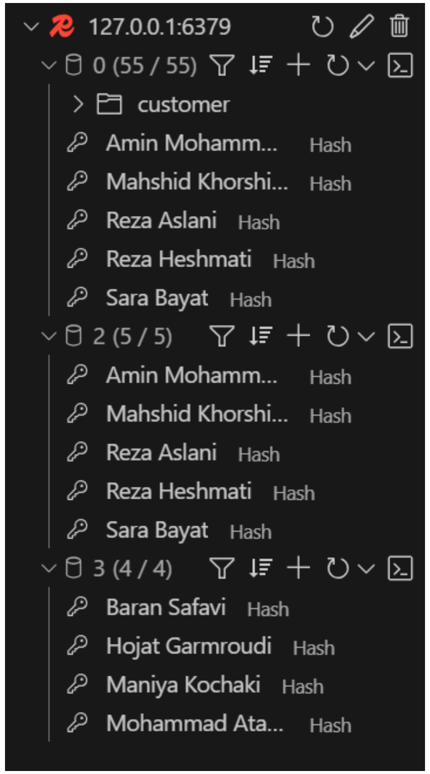
3 (4 / 4)

Baran Safavi Hash

Hojat Garmroudi Hash

Maniya Kochaki Hash

Mohammad Ata... Hash



127.0.0.1:6379

0 (55 / 55)

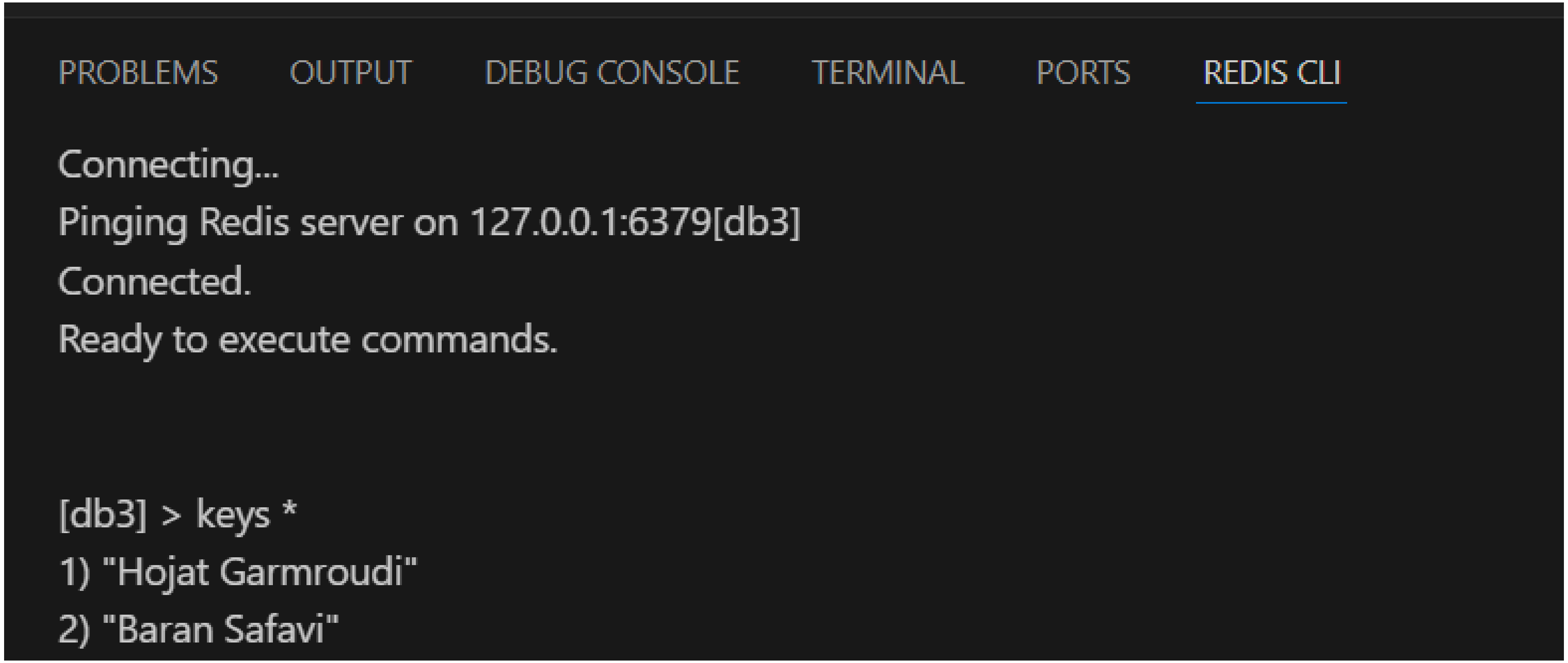
- customer
- Amin Mohamm... Hash
- Mahshid Khorshi... Hash
- Reza Aslani Hash
- Reza Heshmati Hash
- Sara Bayat Hash

2 (5 / 5)

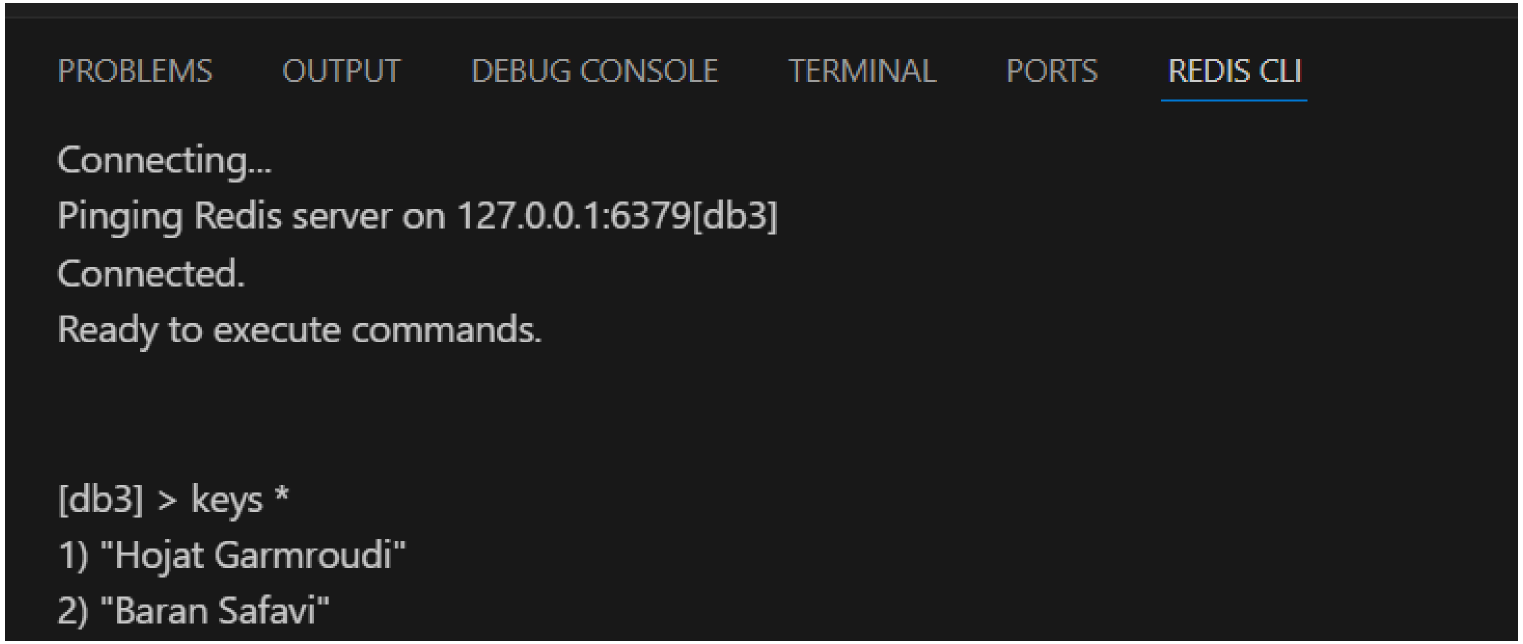
- Amin Mohamm... Hash
- Mahshid Khorshi... Hash
- Reza Aslani Hash
- Reza Heshmati Hash
- Sara Bayat Hash

3 (4 / 4)

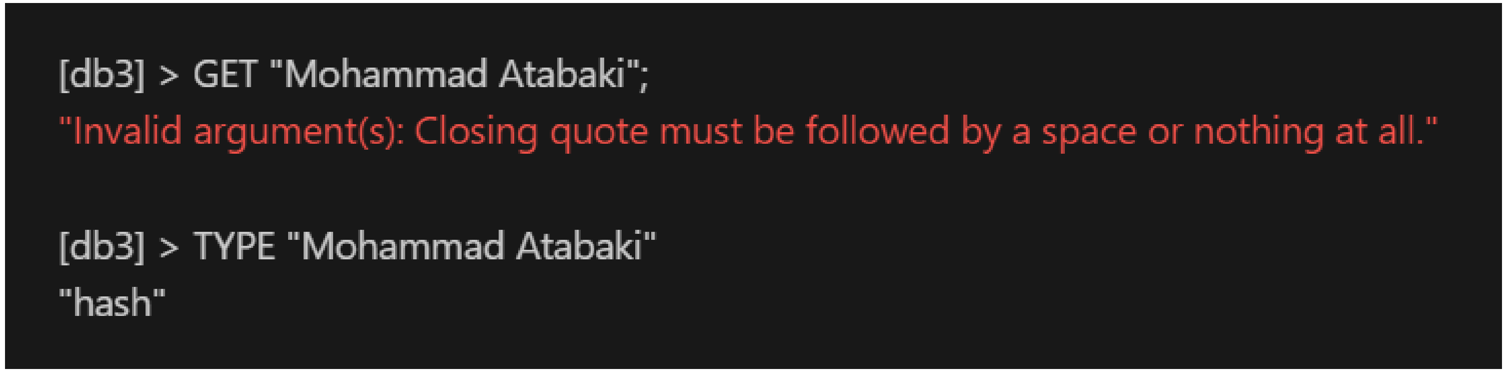
- Baran Safavi Hash
- Hojat Garmroudi Hash
- Maniya Kochaki Hash
- Mohammad Ata... Hash



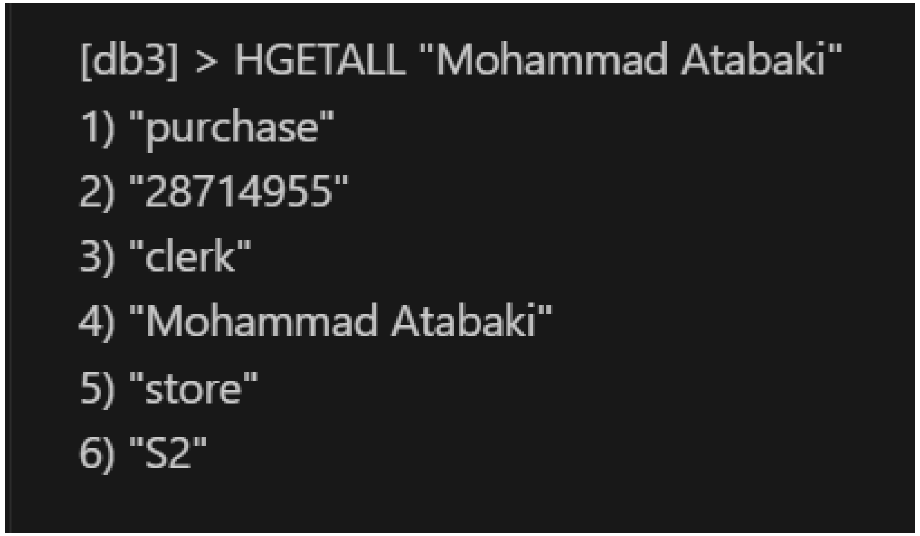
```
Connecting...
Pinging Redis server on 127.0.0.1:6379[db3]
Connected.
Ready to execute commands.


[db3] > keys *
1) "Hojat Garmroudi"
2) "Baran Safavi"
```



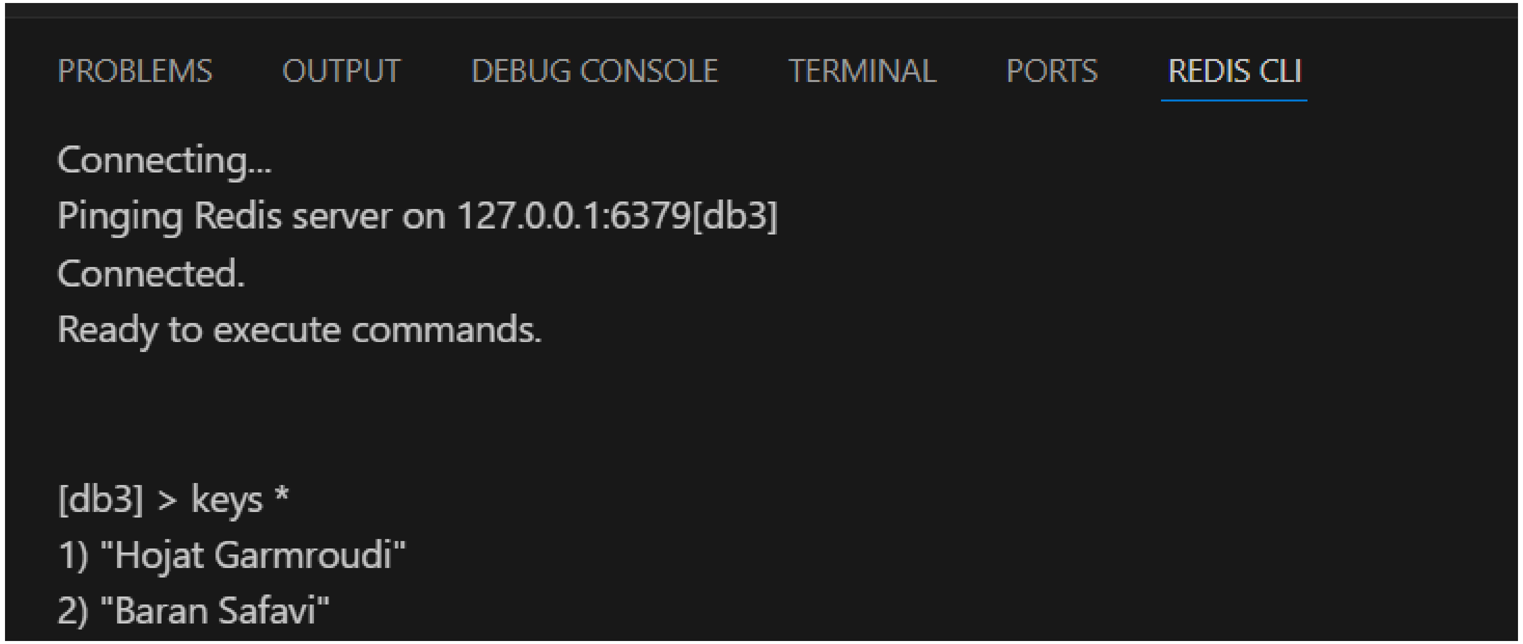
```
Connecting...
Pinging Redis server on 127.0.0.1:6379[db3]
Connected.
Ready to execute commands.


[db3] > keys *
1) "Hojat Garmroudi"
2) "Baran Safavi"
```



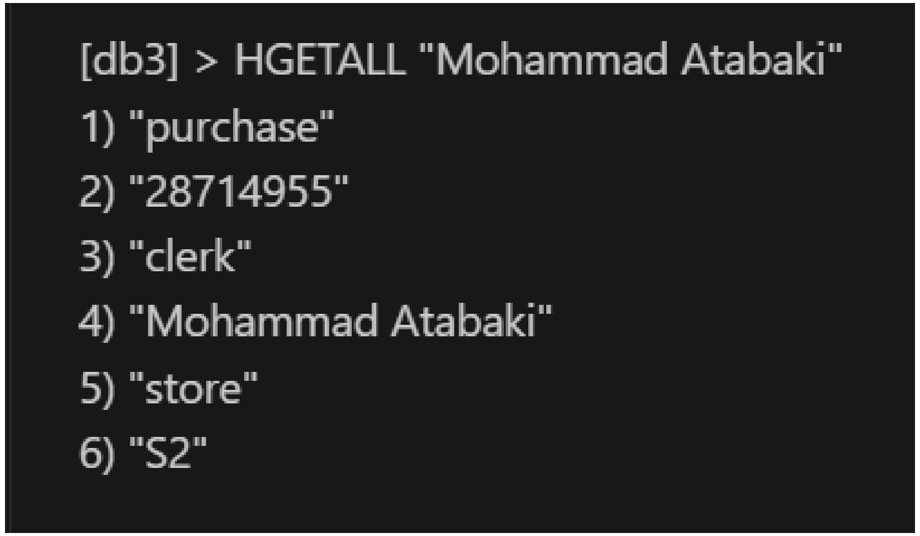
```
[db3] > GET "Mohammad Atabaki";
"Invalid argument(s): Closing quote must be followed by a space or nothing at all."

[db3] > TYPE "Mohammad Atabaki"
"hash"
```



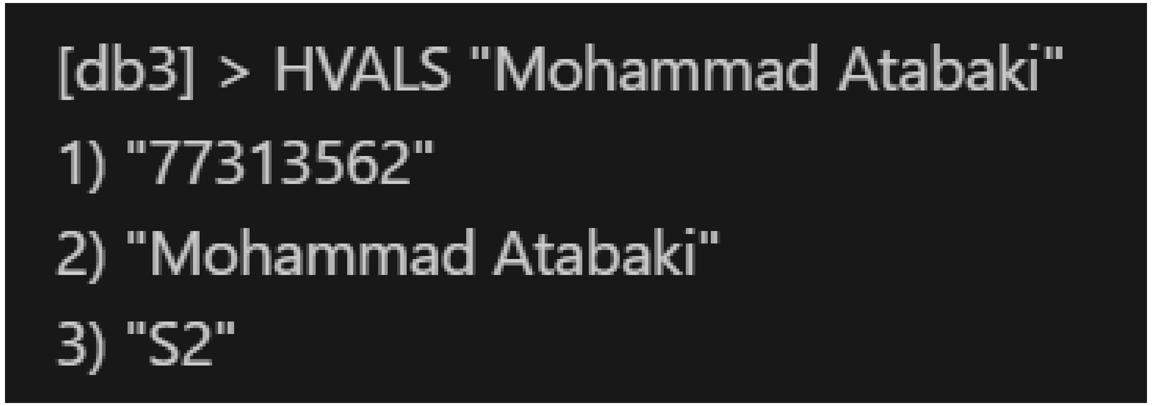
```
[db3] > HGETALL "Mohammad Atabaki"
1) "purchase"
2) "28714955"
3) "clerk"
4) "Mohammad Atabaki"
5) "store"
6) "S2"
```



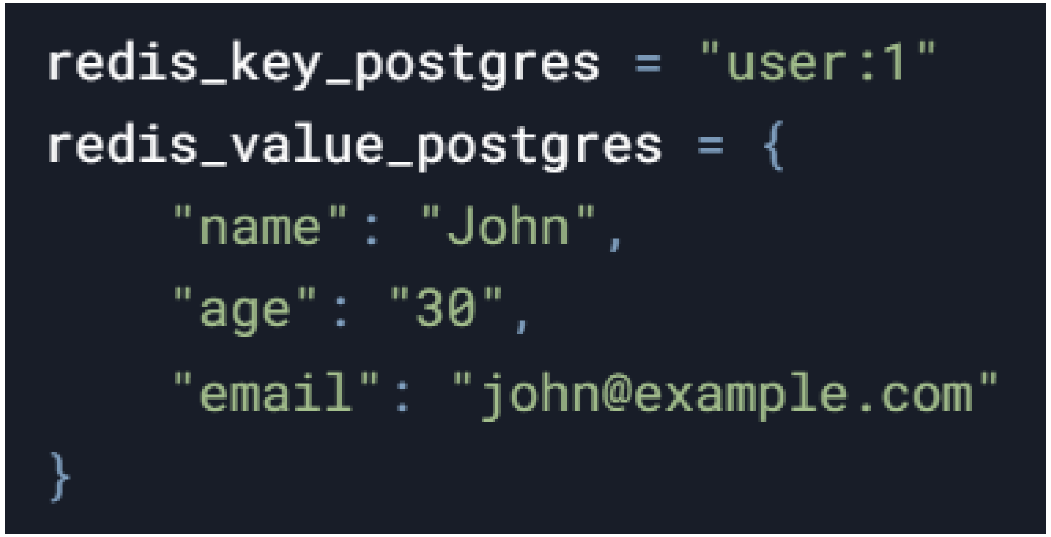
```
Connecting...
Pinging Redis server on 127.0.0.1:6379[db3]
Connected.
Ready to execute commands.


[db3] > keys *
1) "Hojat Garmroudi"
2) "Baran Safavi"
```



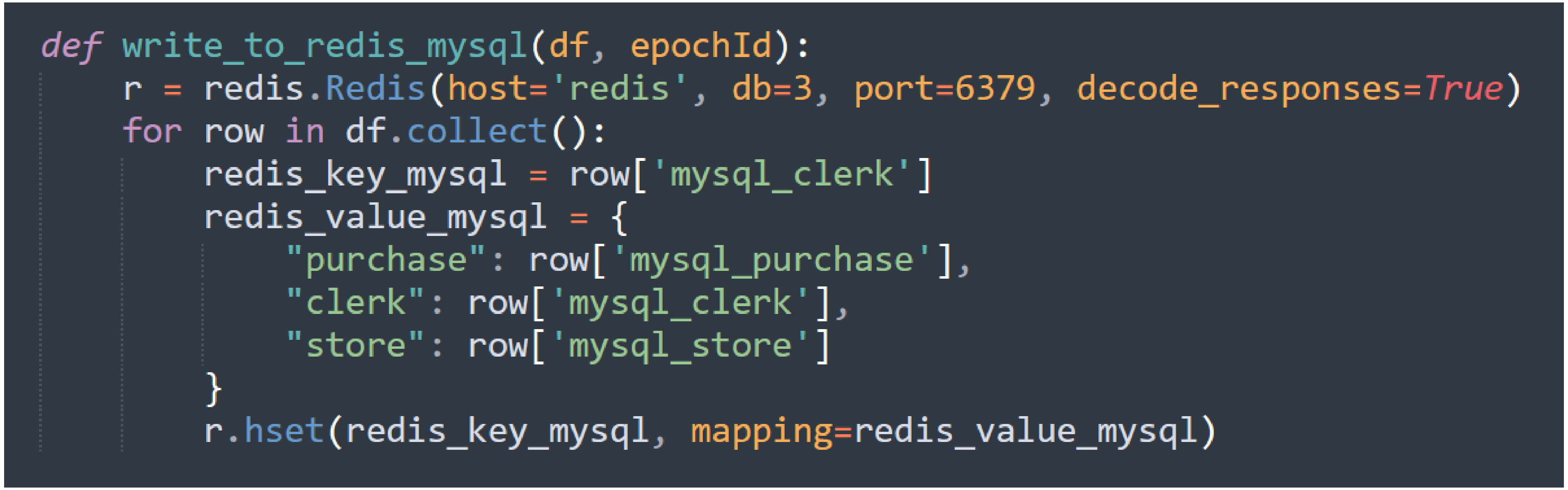
```
[db3] > HGETALL "Mohammad Atabaki"
1) "purchase"
2) "28714955"
3) "clerk"
4) "Mohammad Atabaki"
5) "store"
6) "S2"
```



```
[db3] > HVALS "Mohammad Atabaki"
1) "77313562"
2) "Mohammad Atabaki"
3) "S2"
```



```
redis_key_postgres = "user:1"
redis_value_postgres = {
    "name": "John",
    "age": "30",
    "email": "john@example.com"
}
```



```
def write_to_redis_mysql(df, epochId):
    r = redis.Redis(host='redis', db=3, port=6379, decode_responses=True)
    for row in df.collect():
        redis_key_mysql = row['mysql_clerk']
        redis_value_mysql = {
            "purchase": row['mysql_purchase'],
            "clerk": row['mysql_clerk'],
            "store": row['mysql_store']
        }
        r.hset(redis_key_mysql, mapping=redis_value_mysql)
```

Now Let’s go for coding 🚀