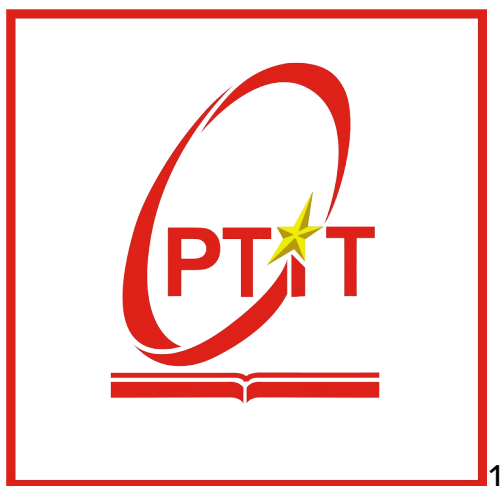


HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



Báo cáo hàng tuần

Môn học: Thực tập cơ sở

Giảng viên: Kim Ngọc Bách

Họ và tên: Nguyễn Hữu Phúc

Mã SV: B22DCAT224

Lớp: E22CQCN04-B

Báo cáo tuần 5

I. Cài Đặt MongoDB Trên Docker và Truy Vấn Cơ Sở Dữ Liệu Glamira

Lý Do Em Sử Dụng Docker

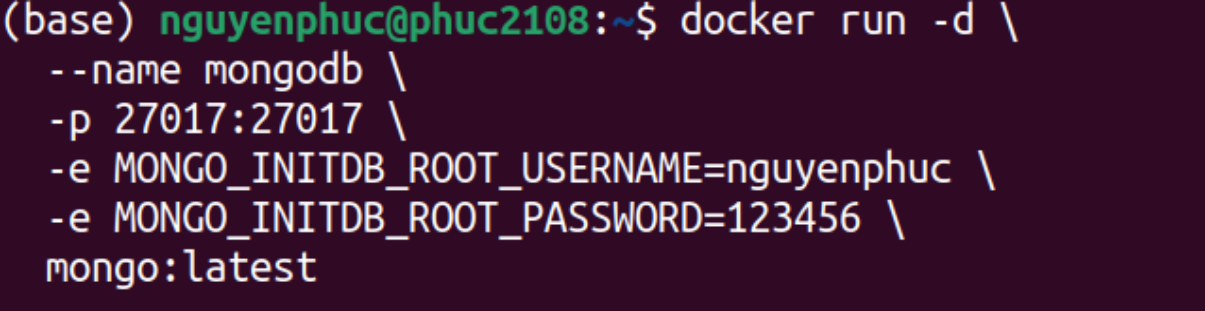
Sau một thời gian cài MongoDB trực tiếp, thấy nhiều bất tiện và khó khăn trong việc thực hiện nên em đã chọn lại cách cài MongoDB trên Docker. Lí do là vì Docker tạo môi trường riêng cho MongoDB, tránh xung đột với phần mềm khác. Nó dễ triển khai trên máy khác mà không cần cài lại. Nếu có lỗi, em chỉ cần xóa container và làm mới, giữ máy tính gọn gàng. Hơn nữa, Docker giúp môi trường làm việc giống nhau ở mọi nơi, tiết kiệm thời gian xử lý vấn đề.

Các Bước Em Thực Hiện

1. Cài Đặt Lại Container MongoDB

Em chạy lại container MongoDB mới:

```
docker run -d \ --name mongodb \ -p 27017:27017 \ -e MONGO_INITDB_ROOT_USERNAME=nguyenphuc \ -e MONGO_INITDB_ROOT_PASSWORD=123456 \ mongo:latest
```



```
(base) nguyenphuc@phuc2108:~$ docker run -d \  
  --name mongodb \  
  -p 27017:27017 \  
  -e MONGO_INITDB_ROOT_USERNAME=nguyenphuc \  
  -e MONGO_INITDB_ROOT_PASSWORD=123456 \  
  mongo:latest
```

Bước này tạo container với tên người dùng nguyenphuc và mật khẩu 123456.

2. Kiểm Tra Docker

Em kiểm tra để đảm bảo container và image đúng:

```
docker ps -a
```

Lệnh này cho thấy trạng thái container mongodb. Em cũng kiểm tra image:

```
docker images
```

```
(base) nguyenphuc@phuc2108:~$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
76337194c5c	mongo:latest	"docker-entrypoint.s..."	16 seconds ago	Up 15 seconds
75496c64bb7	wurstmeister/zookeeper	"/bin/sh -c '/usr/sb..."	26 minutes ago	Up 26 minutes
881b8e4af966	kafka-docker_kafka	"start-kafka.sh"	26 minutes ago	Up 26 minutes
951376cc1963	mysql/mysql-server:latest	"/entrypoint.sh mysql..."	7 days ago	Up 16 hours (healthy)

```
(base) nguyenphuc@phuc2108:~$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
kafka-docker_kafka	latest	c4b93ca5d931	26 minutes ago	508MB
mongo	latest	7ef8fa6da12d	2 weeks ago	888MB
hello-world	latest	74cc54e27dc4	2 months ago	10.1kB
mysql/mysql-server	latest	1d9c2219ff69	2 years ago	496MB
wurstmeister/zookeeper	latest	3f43f72cb283	6 years ago	510MB

3. Khởi Động Container

docker start mongod

Bước này đảm bảo MongoDB sẵn sàng.

4. Sao Chép Dữ Liệu

Copy file summary.bson vào container:

docker cp

/home/nguyenphuc/Documents/GlamiraUserFlowInsightsProject/glamira_ubl_oct2019_nov2019/dump/countly/summary.bson mongodb:/tmp/summary.bson

```
(base) nguyenphuc@phuc2108:~$ docker cp /home/nguyenphuc/Documents/GlamiraUserFlowInsightsProject/glamira_ubl_oct2019_nov2019/dump/countly/summary.bson mongodb:/tmp/summary.bson
Successfully copied 33.5GB to mongodb:/tmp/summary.bson
```

File được đặt trong container để nhập dữ liệu.

5. Truy Cập Shell MongoDB

Em vào shell MongoDB:

docker exec -it mongodb mongosh -u nguyenphuc -p 123456 --authenticationDatabase admin

```
(base) nguyenphuc@phuc2108:~$ docker exec -it mongodb mongosh -u nguyenphuc -p 123456 --authenticationDatabase admin
Current Mongosh Log ID: 67fa73c58c447a03f5d861df
Connecting to: mongodb://<credentials>@127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&authSource=admin&appName=mongosh+2.5.0
Using MongoDB: 8.0.6
Using Mongosh: 2.5.0

For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/

-----
The server generated these startup warnings when booting
2025-04-12T14:07:19.079+00:00: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodn
2025-04-12T14:07:19.375+00:00: For customers running the current memory allocator, we suggest changing the contents of the following sysfsFile
2025-04-12T14:07:19.375+00:00: For customers running the current memory allocator, we suggest changing the contents of the following sysfsFile
2025-04-12T14:07:19.375+00:00: We suggest setting the contents of sysfsFile to 0.
2025-04-12T14:07:19.375+00:00: vm.max_map_count is too low
2025-04-12T14:07:19.376+00:00: We suggest setting swappiness to 0 or 1, as swapping can cause performance problems.
-----
```

Bước này cho phép thao tác với cơ sở dữ liệu.

6. Nhập Dữ Liệu Vào Glamira

Nhập file dữ liệu vào Glamira:

docker exec mongodb mongorestore --db Glamira --collection summary --username nguyenphuc --password 123456 --authenticationDatabase admin /tmp/summary.bson

```
(base) nguyenphuc@phuc2108:~$ docker exec mongodb mongorestore --db Glamira --collection summary --username nguyenphuc --password 123456 --authenticationDatabase admin --
/tmp/summary.bson
2025-04-11T18:27:40.715+0000 checking for collection data in /tmp/summary.bson
2025-04-11T18:27:40.743+0000 restoring Glamira.summary from /tmp/summary.bson
2025-04-11T18:27:43.714+0000 [.....] Glamira.summary 323MB/31.2GB (1.0%)
2025-04-11T18:27:46.714+0000 [.....] Glamira.summary 617MB/31.2GB (1.9%)
2025-04-11T18:27:49.715+0000 [.....] Glamira.summary 883MB/31.2GB (2.8%)
2025-04-11T18:27:52.715+0000 [.....] Glamira.summary 1.14GB/31.2GB (3.6%)
2025-04-11T18:27:55.715+0000 [#.....] Glamira.summary 1.43GB/31.2GB (4.6%)
2025-04-11T18:27:58.715+0000 [#.....] Glamira.summary 1.71GB/31.2GB (5.5%)
2025-04-11T18:28:01.715+0000 [#.....] Glamira.summary 1.99GB/31.2GB (6.4%)
2025-04-11T18:28:04.714+0000 [#.....] Glamira.summary 2.27GB/31.2GB (7.3%)
2025-04-11T18:28:07.714+0000 [#.....] Glamira.summary 2.56GB/31.2GB (8.2%)
2025-04-11T18:28:10.714+0000 [##.....] Glamira.summary 2.84GB/31.2GB (9.1%)
2025-04-11T18:28:13.716+0000 [##.....] Glamira.summary 3.12GB/31.2GB (10.0%)
2025-04-11T18:28:16.715+0000 [##.....] Glamira.summary 3.41GB/31.2GB (10.9%)
2025-04-11T18:28:19.714+0000 [##.....] Glamira.summary 3.69GB/31.2GB (11.8%)
2025-04-11T18:28:22.715+0000 [###.....] Glamira.summary 3.93GB/31.2GB (12.6%)
2025-04-11T18:28:25.715+0000 [###.....] Glamira.summary 4.17GB/31.2GB (13.4%)
2025-04-11T18:28:28.714+0000 [###.....] Glamira.summary 4.44GB/31.2GB (14.2%)
2025-04-11T18:28:31.714+0000 [###.....] Glamira.summary 4.72GB/31.2GB (15.1%)
```

7. Kiểm Tra và Truy Vấn Dữ Liệu

```
test> show dbs
Glamira 11.29 GiB
admin 100.00 KiB
config 72.00 KiB
local 72.00 KiB
test> use Glamira
switched to db Glamira
Glamira> db.summary.find().limit(5).pretty()
[
  {
    _id: ObjectId('5ed8cb2bc671fc36b74653ad'),
    time_stamp: 1591266092,
    ip: '37.170.17.183',
    user_agent: 'Mozilla/5.0 (iPhone; CPU iPhone OS 13_4_1 like Mac OS X) AppleWebKit/60
    resolution: '375x667',
    user_id_db: '502567',
    device_id: 'beb2cacb-20af-4f05-9c03-c98e54a1b71a',
    api_version: '1.0',
    store_id: '12',
    local_time: '2020-06-04 12:21:27',
    show_recommendation: 'false',
    current_url: 'https://www.glamira.fr/glamira-pendant-viktor.html?alloy=yellow-375',
    referrer_url: 'https://www.glamira.fr/men-s-necklaces/',
    email_address: 'pereira.vivien@yahoo.fr',
    recommendation: false,
    utm_source: false,
    utm_medium: false,
    collection: 'view_product_detail',
    product_id: '110474',
    option: [
      {
        option_label: 'alloy',
        option_id: '332084',
        value_label: '',
        value_id: '3279318'
      },
      {
        option_label: 'diamond',
        option_id: '',
        value_label: '',
        value_id: ''
      }
    ]
  },
]
```

Kết Luận

Em đã cài lại MongoDB trên Docker và xử lý thành công dữ liệu Glamira. Docker giúp em làm nhanh, gọn, và ổn định. Em đã truy vấn được năm bản ghi như yêu cầu, sẵn sàng tiếp tục công việc.

II. Viết file python để lấy dữ liệu từ mongodb

Viết file `test_mongodb_connection`:

```
import pymongo
from pymongo.errors import ConnectionFailure, OperationFailure
import time

def test_mongodb_connection():
    """
    Test connection to MongoDB using a hardcoded connection string.

    Returns:
        bool: True if connection successful, False otherwise
    """
    # Define your MongoDB connection string here
    connection_string = "mongodb://nguyenphuc:123456@localhost:27017/"

    # You can specify a specific database if needed
    database_name = "Glamira"

    print(f"Attempting to connect to MongoDB with connection string: {connection_string}")

    try:
        # Create a MongoDB client with a timeout
        start_time = time.time()
        client = pymongo.MongoClient(connection_string,
serverSelectionTimeoutMS=5000)

        # The ismaster command is cheap and does not require auth
        client.admin.command('ismaster')

        # Calculate connection time
        connection_time = time.time() - start_time

        # If we get here, the connection was successful
        print(f"MongoDB connection successful! Connected in {connection_time:.2f} seconds.")

        # Test access to the specific database if provided
        if database_name:
            db = client[database_name]
            collections = db.list_collection_names()
```

```

        print(f"\nCollections in database '{database_name}':")
        if collections:
            for collection in collections:
                print(f"- {collection}")
                # Get count of documents in this collection
                count = db[collection].count_documents({})
                print(f"    • Contains {count} documents")
            else:
                print("No collections found in this database.")

    # List all available databases
    print("\nAll available databases:")
    database_names = client.list_database_names()
    for db_name in database_names:
        print(f"- {db_name}")

    # Close the connection
    client.close()
    return True

except ConnectionFailure as e:
    print(f"MongoDB connection failed: Could not connect to server.
Error: {e}")
    print("\nPossible issues:")
    print("1. The server address or port is incorrect")
    print("2. The server is not running")
    print("3. Network connectivity issues or firewall blocking the
connection")
    return False

except OperationFailure as e:
    if "Authentication failed" in str(e):
        print(f"MongoDB connection failed: Authentication failed.
Error: {e}")
        print("\nPossible issues:")
        print("1. Username or password is incorrect")
        print("2. User does not have access to the specified
database")
        print("3. Authentication database is incorrect (should be
specified in the connection string)")
    else:
        print(f"MongoDB operation failed. Error: {e}")
    return False

except Exception as e:

```

```

        print(f"Unexpected error: {e}")
        print(f"Error type: {type(e).__name__}")
        return False

if __name__ == "__main__":
    # Test the connection
    test_mongodb_connection()

    # Keep console window open if run by double-clicking
    input("\nPress Enter to exit...")

```

Viết file `extract_mongodb_data.py`

```

import pymongo
import json
from bson.objectid import ObjectId
import pandas as pd
from datetime import datetime
import os

# Function to connect to MongoDB
def connect_to_mongodb(connection_string, db_name):
    client = pymongo.MongoClient(connection_string)
    db = client[db_name]
    return db

# Function to extract specific fields from a collection
def extract_data_from_collection(db, collection_name, output_file):
    collection = db[collection_name]

    # Define fields to extract
    fields_to_extract = {
        "_id": 1,
        "device_id": 1,
        "time_stamp": 1,
        "current_url": 1,
        "referrer_url": 1,
        "email_address": 1,
        "collection": 1,
        "product_id": 1,
        "option": 1
    }

    # Query the collection

```

```

cursor = collection.find({}, fields_to_extract)

# Process documents
extracted_data = []
for doc in cursor:
    # Convert ObjectId to string
    if "_id" in doc and isinstance(doc["_id"], ObjectId):
        doc["_id"] = str(doc["_id"])

    extracted_data.append(doc)

# Export as JSON
if extracted_data:
    with open(output_file, 'w', encoding='utf-8') as f:
        json.dump(extracted_data, f, ensure_ascii=False, indent=4)

    print(f"Extracted {len(extracted_data)} records to {output_file}")
    return extracted_data
else:
    print("No data found matching the criteria")
    return None

# Function to extract data from a BSON file directly (without MongoDB)
def extract_from_bson_file(bson_file_path, output_file):
    import bson

    # Check if file exists
    if not os.path.exists(bson_file_path):
        print(f"File {bson_file_path} not found")
        return None

    # Read BSON file
    with open(bson_file_path, 'rb') as f:
        data = bson.decode_all(f.read())

    # Extract required fields
    extracted_data = []
    for doc in data:
        extracted_doc = {}

        # Extract each specified field if it exists
        for field in ["_id", "device_id", "time_stamp", "current_url",

```



```

        "referrer_url", "email_address", "collection",
"product_id", "option"]:
    if field in doc:
        # Handle ObjectId conversion
        if field == "_id" and isinstance(doc[field], ObjectId):
            extracted_doc[field] = str(doc[field])
        else:
            extracted_doc[field] = doc[field]

    extracted_data.append(extracted_doc)

# Export as JSON
if extracted_data:
    with open(output_file, 'w', encoding='utf-8') as f:
        json.dump(extracted_data, f, ensure_ascii=False, indent=4)

    print(f"Extracted {len(extracted_data)} records to
{output_file}")
    return extracted_data
else:
    print("No data found in the BSON file")
    return None

# Function to handle datetime serialization for JSON
class DateTimeEncoder(json.JSONEncoder):
    def default(self, obj):
        if isinstance(obj, datetime):
            return obj.isoformat()
        return super(DateTimeEncoder, self).default(obj)

# Main execution
if __name__ == "__main__":

    connection_string = "mongodb://nguyenphuc:123456@localhost:27017/"
    db_name = "Glamira"
    collection_name = "summary"
    output_file = "extracted_data.json"
    db = connect_to_mongodb(connection_string, db_name)
    data = extract_data_from_collection(db, collection_name,
output_file)

    if data is not None:
        print("Data extraction completed successfully")

```

```
print(f"Total records extracted: {len(data)}")
if len(data) > 0:
    print("Sample of first record structure:")
    print(json.dumps(data[0], indent=2, ensure_ascii=False))
```