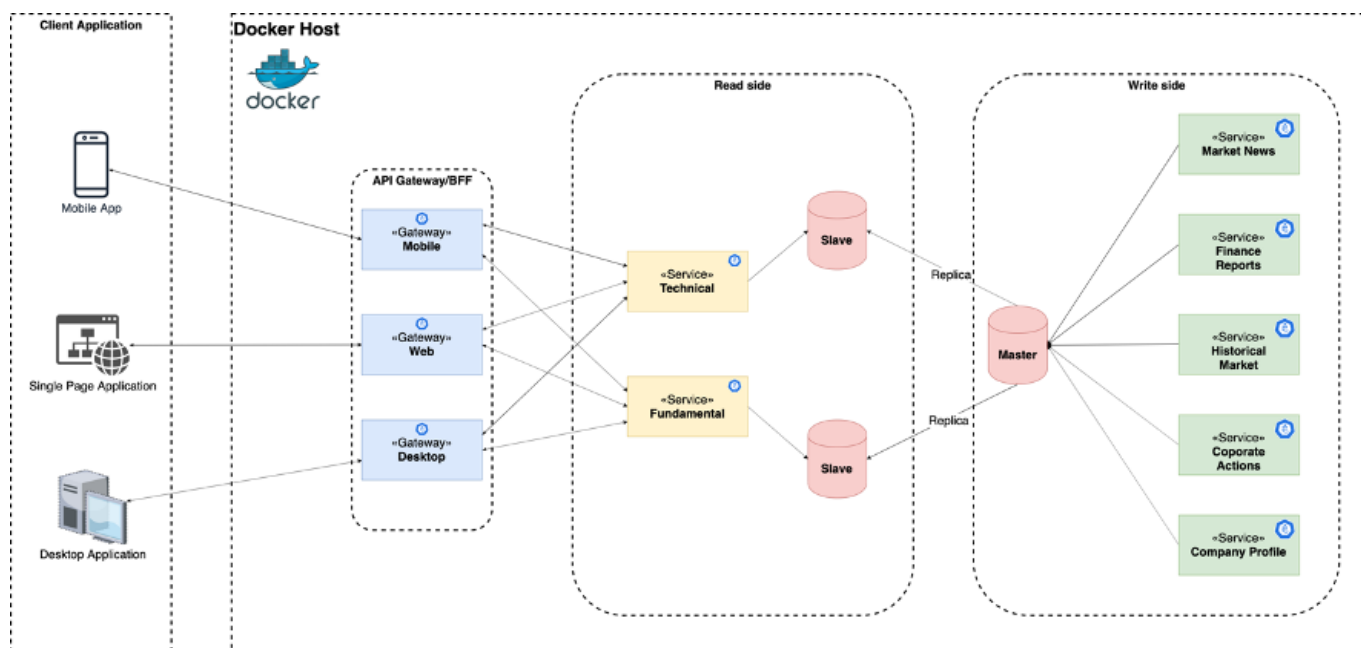


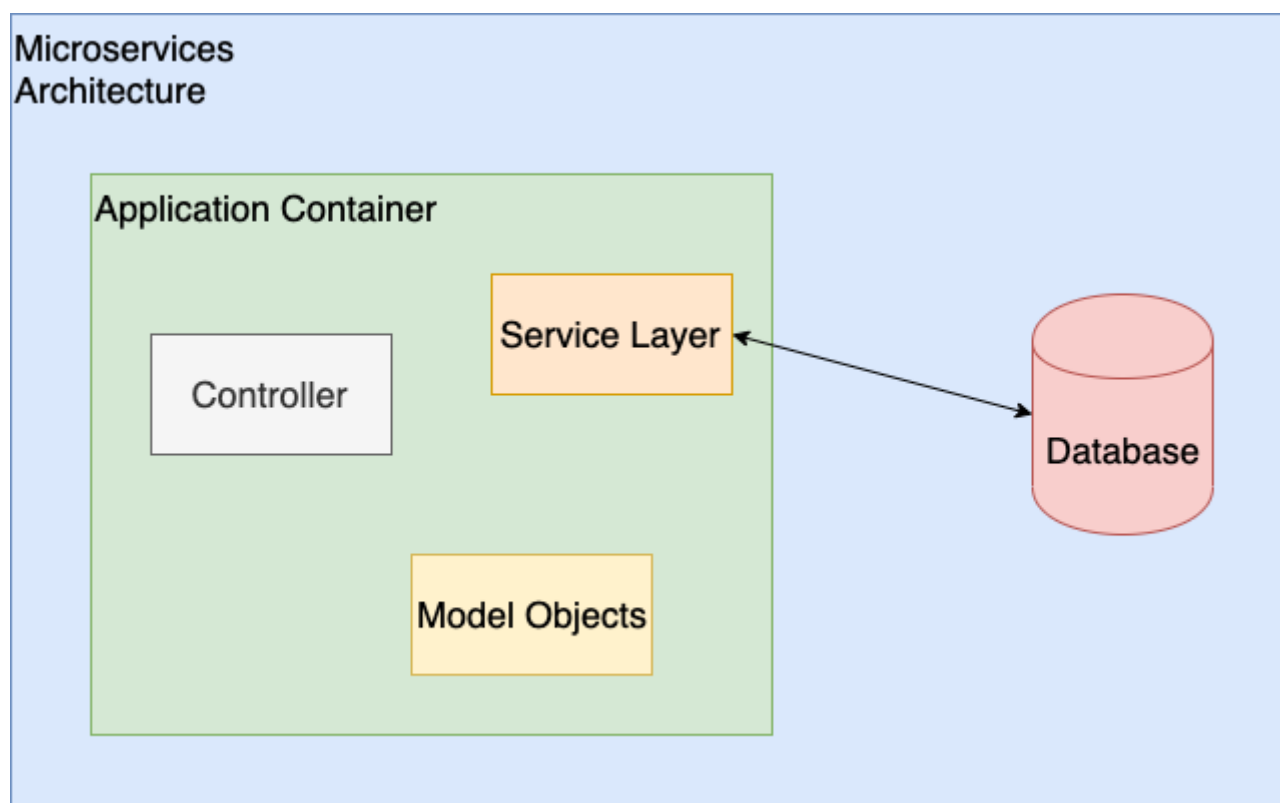
Documentation

- Mô hình kiến trúc hệ thống
- Môi trường, thiết bị, công cụ sử dụng
- Cài đặt môi trường
- Cài đặt các dịch vụ

Mô hình hệ thống



Kiến trúc chung của các microservices





Hệ thống chia làm 2 phần chính với các service tương ứng

- Thu thập dữ liệu(Ghi dữ liệu):
 - Market News Service
 - Finance Report Service
 - Historical Market Service
 - Corporate Action Service
 - Company Profile Service
 - Share Holder Service
- Cung cấp dữ liệu(Đọc dữ liệu)
 - Technical Service
 - Fundamental Service



Môi trường, thiết bị, công cụ sử dụng

- **Môi trường phát triển** - MacOS.
- **Môi trường triển khai** - CentOS 7.6.
- **Ngôn ngữ lập trình** - Java, C++, JavaScript.
- **Framework** - Spring Boot, NodeJS.
- **Thư viện** - ReactJS.
- **Mã nguồn mở** - GitLab, Ant Design.
- **Database** - Oracle Database 12c.
- **Container Platform** - Docker.
- **CI/CD** - GitLabCI.

Cài đặt môi trường

1. Cài đặt Docker trên hệ điều hành CentOS 7.6

- Truy cập <https://download.docker.com/linux/centos/> và chọn phiên bản CentOS của bạn. Sau đó duyệt đến `x86_64/stable/Packages/` và tải xuống .rpm tệp cho phiên bản Docker mà bạn muốn cài đặt.
- Cài đặt Docker Engine, thay đổi đường dẫn bên dưới thành đường dẫn mà bạn đã tải xuống gói Docker

```
sudo yum install /path/to/package.rpm
```

- Khởi động Docker

```
sudo systemctl start docker
```

- Xác minh rằng Docker Engine được cài đặt chính xác bằng cách chạy hello-world hình ảnh

```
sudo docker run hello-world
```

2. Cài đặt Oracle Database 12.2.0.1 trên Docker

- Lấy docker image từ Docker Hub

```
docker pull store/oracle/database-enterprise:12.2.0.1
```

- Khởi chạy docker container Oracle Database 12.2.0.1

```
docker run -d -p 8080:8080 -p 1521:1521 -p 5500:5500 -v /opt/stock-data-storage:/ORCL store/oracle/database-enterprise:12.2.0.1
```

- Kiểm tra khi trạng thái của container là healthy thì bắt đầu khởi tạo người dùng cho cơ sở dữ liệu

```
docker ps
```

- Thực thi các câu lệnh bên trong container bằng cách truy cập /bin/bash

```
docker exec -it <container-id> /bin/sh
```

- Thực hiện việc khởi tạo người dùng

```
source /home/oracle/.bashrc;
sqlplus sys/Oradoc_db1@ORCLCDB as sysdba
alter session set "_ORACLE_SCRIPT"=true;
create người dùng root identified by 123456;
grant all privileges to root;
grant connect to root;
grant create user to root
```

- Khởi tạo các bảng với script tương ứng bên dưới

```
create table TB_ADJUSTED_RATE
(
    SYMBOL          VARCHAR2(50),
    ADJRATE         NUMBER,
    TRADINGDATE     DATE,
```

```

        UPDATEDDATE DATE,
        APPLIED      NUMBER(1),
        ID           NUMBER generated as identity,
        EXCHANGE     VARCHAR2(20),
        CASTATUS     VARCHAR2(20),
        FLAG         VARCHAR2(200)
    )

create table TB_MARKET_DATA
(
    STOCKTYPE      VARCHAR2(20),
    TRADINGDATE    DATE,
    TRADINGTIME    VARCHAR2(20),
    EXCHANGE       VARCHAR2(20),
    CONFIRMNO      VARCHAR2(20),
    STOCKNO        VARCHAR2(20),
    STOCKSYMBOL    VARCHAR2(20),
    TOTALVOL       NUMBER(18, 2),
    TOTALPRICE     NUMBER(18, 2),
    SIDE           VARCHAR2(20),
    ACCUMULATEDVOL NUMBER(18, 2),
    ACCUMULATEDVAL NUMBER(18, 2),
    HIGHEST        NUMBER(18, 2),
    LOWEST         NUMBER(18, 2),
    AVGPRICE       NUMBER,
    PRIORPRICE     NUMBER,
    TOTALMATCHEDVOL NUMBER,
    ID             NUMBER not null,
    TOTALVOLUMETRADED NUMBER,
    TOTALVALUETRADED NUMBER
)

create index TB_MD__INDEX
    on TB_MARKET_DATA (STOCKSYMBOL, TRADINGDATE)

create unique index TB_MARKET_DATA_ID_UINDEX
    on TB_MARKET_DATA (ID)

alter table TB_MARKET_DATA
    add constraint TB_MARKET_DATA_PK
        primary key (ID)

create table TB_DAILY_STOCK_INFO
(
    EXCHANGEID      NUMBER,
    STOCKSYMBOL     NVARCHAR2(50) default NULL,
    TRADINGDATE     DATE,
    PRIORPRICE      NUMBER,
    OPENPRICE       NUMBER,
    CLOSEPRICE      NUMBER,
    HIGHPRICE       NUMBER,
    LOWPRICE        NUMBER,
    AVGPRICE        NUMBER,
    TOTALQTTY       NUMBER,

```

```

        TOTALVALUE                NUMBER,
        BUYTOTALTRADE              NUMBER,
        BUYTOTALQTTY               NUMBER,
        SELLTOTALTRADE             NUMBER,
        SELLTOTALQTTY              NUMBER,
        OPENPRICE_ADJUSTED         NUMBER,
        CLOSEPRICE_ADJUSTED        NUMBER,
        HIGHPRICE_ADJUSTED         NUMBER,
        LOWPRICE_ADJUSTED          NUMBER,
        ID                         NUMBER not null,
        TOTALVOLUMETRADED          NUMBER,
        TOTALVALUETRADED           NUMBER,
        PRICECHANGE                NUMBER,
        PRICECHANGEPERCENTAGE      NUMBER,
        REFPRICE                   NUMBER,
        CEILINGPRICE               NUMBER,
        FLOORPRICE                 NUMBER,
        CLOSEPRICE_ADJUSTED_IDS    NUMBER,
        HIGHPRICE_ADJUSTED_IDS     NUMBER,
        LOWPRICE_ADJUSTED_IDS      NUMBER,
        OPENPRICE_ADJUSTED_IDS     NUMBER,
        BIDCOUNT                  NUMBER,
        TOTALBIDQTTY               NUMBER,
        OFFERCOUNT                  NUMBER,
        TOTALOFFERQTTY             NUMBER,
        EXCHANGE                   VARCHAR2(20)
    )

create index TB_DSI_INDEX
    on TB_DAILY_STOCK_INFO (STOCKSYMBOL, TRADINGDATE)

create unique index TB_DAILY_STOCK_INFO_ID_UINDEX
    on TB_DAILY_STOCK_INFO (ID)

alter table TB_DAILY_STOCK_INFO
    add constraint TB_DAILY_STOCK_INFO_PK
        primary key (ID)

create table CR_TRADING_EXCHANGE
(
    ID                NUMBER default "ROOT"."ISEQ$$_94469".nextval generated
as identity,
    CREATED_DATE      DATE      default CURRENT_DATE,
    MODIFIED_DATE     DATE      default CURRENT_DATE,
    CREATOR_ID        NUMBER,
    EXCHANGE_CODE      VARCHAR2(1000),
    EXCHANGE_NAME      VARCHAR2(1000),
    MODIFIED_ID        NUMBER,
    IS_ACTIVE          NUMBER(1),
    IS_DELETED         NUMBER(1)
)

create table AD_NEWS
(

```

```

        ID                NUMBER not null
            primary key,
        CREATE_DATE        DATE,
        UPDATE_DATE        DATE,
        SYMBOL              VARCHAR2(1000),
        TITLE               VARCHAR2(1000),
        IMAGE_URL           VARCHAR2(1000),
        SHORT_CONTENT       VARCHAR2(1000),
        FULL_CONTENT        VARCHAR2(1000),
        NEWS_SOURCE         VARCHAR2(1000),
        SOURCE_CODE         VARCHAR2(1000),
        NEWS_SOURCE_LINK    VARCHAR2(1000)
    )

create table AD_CORPORATE
(
    ID                NUMBER not null
        primary key,
    SYMBOL            VARCHAR2(1000),
    EVENT_NAME        VARCHAR2(1000),
    EX_RIGHT_DATE     DATE,
    RECORD_DATE       DATE,
    ISSUE_DATE        DATE,
    EVENT_TITLE       VARCHAR2(1000),
    PUBLIC_DATE       VARCHAR2(1000),
    EXCHANGE          VARCHAR2(1000),
    EVENT_LIST_CODE   VARCHAR2(1000),
    VALUE             FLOAT,
    RATIO             FLOAT,
    EVENT_DESCRIPTION VARCHAR2(1000),
    EVENT_CODE        VARCHAR2(1000)
)

create table AD_COMPANY_PROFILE
(
    ID                NUMBER not null
        primary key,
    SYMBOL            VARCHAR2(1000),
    SUB_SECTOR_CODE   VARCHAR2(3000),
    INDUSTRY_NAME     VARCHAR2(1000),
    SUPER_SECTOR      VARCHAR2(1000),
    SECTOR            VARCHAR2(1000),
    SUB_SECTOR        VARCHAR2(1000),
    FOUNDING_DATE     DATE,
    CHARTER_CAPITAL   FLOAT,
    NUMBER_OF_EMPLOYEE NUMBER,
    BANK_NUMBER_OF_BRANCH NUMBER,
    COMPANY_PROFILE    VARCHAR2(1000),
    LISTING_DATE      DATE,
    EXCHANGE          VARCHAR2(1000),
    FIRST_PRICE       FLOAT,
    ISSUE_SHARE       FLOAT,
    LISTED_VALUE      FLOAT,
    COMPANY_NAME      VARCHAR2(1000),

```

```
TYPE_NAME          VARCHAR2(1000),
CREATED_TIME       TIMESTAMP(6) default CURRENT_TIMESTAMP,
CREATE_DATE        DATE
)

create table AD_SHARE_HOLDER
(
  ID                NUMBER not null
    primary key,
  OWNER_SYMBOL      VARCHAR2(1000),
  SYMBOL            VARCHAR2(1000),
  NAME              VARCHAR2(1000),
  QUANTITY          FLOAT,
  PERCENTAGE        FLOAT,
  PUBLIC_DATE       DATE,
  OWNER_SHIP_TYPE_CODE VARCHAR2(100),
  TYPE              VARCHAR2(100)
)

create table AD_FINANCE_INDICATOR
(
  ID                NUMBER not null
    primary key,
  SYMBOL            VARCHAR2(1000),
  REVENUE           FLOAT,
  PROFIT            FLOAT,
  YEAR_REPORT       NUMBER,
  LENGTH_REPORT     NUMBER,
  EPS               FLOAT,
  DILUTE_DEPS       FLOAT,
  PE                FLOAT,
  ROE               FLOAT,
  ROA               FLOAT,
  ROIC              FLOAT,
  GROSS_PROFIT_MARGIN FLOAT,
  NET_PROFIT_MARGIN  FLOAT,
  DEBT_EQUITY        FLOAT,
  DEBT_ASSET         FLOAT,
  QUICK_RATIO        FLOAT,
  CURRENT_RATIO      FLOAT,
  PB                FLOAT
)

create table AD_ASSET
(
  ID                NUMBER not null
    primary key,
  SYMBOL            VARCHAR2(100),
  YEAR              NUMBER,
  ASSET             FLOAT
)

create table AD_CAPITAL
(
```

```
        ID          NUMBER not null
        primary key,
        SYMBOL       VARCHAR2(100),
        YEAR         NUMBER,
        OWNER_CAPITAL FLOAT
    )

create table AD_CASH_DIVIDEND
(
    ID          NUMBER not null
    primary key,
    SYMBOL       VARCHAR2(100),
    YEAR         NUMBER,
    VALUE_PER_SHARE FLOAT
)
```

Cài đặt các dịch vụ

1. Fundamental Service

```
git clone https://github.com/minhdatplus/fundamental-service

docker build -t fundamental-service:1.0 .

docker run -it -d -p 8080:8080 fundamental-service:1.0 .
```

Truy cập API Document tại đường dẫn

```
http://localhost:8080/fundamental/swagger-ui.html#/
```

2. Market News Service

```
git clone https://github.com/minhdatplus/crawl-news
```

Thay đổi đường dẫn trong file application.properties với đường dẫn của database và fundamental-service đã triển khai phía trên

```
url.fundamental.host=<fundamental_service_url>

spring.datasource.url=jdbc:oracle:thin:@<database_url>:1521:xe
```

Sau đó tiến hành build docker image theo lệnh bên dưới


```
docker build -t market-news-service:1.0 .  
docker run -it -d -p 8080:8080 market-news-service:1.0 .
```

Truy cập API Document tại đường dẫn

```
http://localhost:8080/news/swagger-ui.html#/
```

3. Finance Report Service

```
git clone https://github.com/minhdatplus/crawl-finance
```

Thay đổi đường dẫn trong file application.properties với đường dẫn của database và fundametal-service đã triển khai phía trên

```
url.fundamental.host=<fundamental_service_url>  
spring.datasource.url=jdbc:oracle:thin:@<database_url>:1521:xe
```

Sau đó tiến hành build docker image theo lệnh bên dưới

```
docker build -t finance-report-service:1.0 .  
docker run -it -d -p 8080:8080 finance-report-service:1.0 .
```

Truy cập API Document tại đường dẫn

```
http://localhost:8080/finance/swagger-ui.html#/
```

4. Historical Market Service

```
git clone https://github.com/minhdatplus/crawl-dailystock
```

Thay đổi đường dẫn trong file application.properties với đường dẫn của database và fundametal-service đã triển khai phía trên

```
url.fundamental.host=<fundamental_service_url>  
  
spring.datasource.url=jdbc:oracle:thin:@<database_url>:1521:xe
```

Sau đó tiến hành build docker image theo lệnh bên dưới

```
docker build -t historical-market-service:1.0 .  
  
docker run -it -d -p 8080:8080 historical-market-service:1.0 .
```

Truy cập API Document tại đường dẫn

```
http://localhost:8080/dailystock/swagger-ui.html#/
```

5. Corporate Action Service

```
git clone https://github.com/minhdatplus/crawl-corporate
```

Thay đổi đường dẫn trong file application.properties với đường dẫn của database và fundametal-service đã triển khai phía trên

```
url.fundamental.host=<fundamental_service_url>  
  
spring.datasource.url=jdbc:oracle:thin:@<database_url>:1521:xe
```

Sau đó tiến hành build docker image theo lệnh bên dưới

```
docker build -t corporate-action-service:1.0 .  
  
docker run -it -d -p 8080:8080 corporate-action-service:1.0 .
```

Truy cập API Document tại đường dẫn

```
http://localhost:8080/corporate/swagger-ui.html#/
```

6. Company Profile Service

```
git clone https://github.com/minhdatplus/crawl-companyprofile
```

Thay đổi đường dẫn trong file application.properties với đường dẫn của database và fundamental-service đã triển khai phía trên

```
url.fundamental.host=<fundamental_service_url>  
spring.datasource.url=jdbc:oracle:thin:@<database_url>:1521:xe
```

Sau đó tiến hành build docker image theo lệnh bên dưới

```
docker build -t company-profile-service:1.0 .  
docker run -it -d -p 8080:8080 company-profile-service:1.0 .
```

Truy cập API Document tại đường dẫn

```
http://localhost:8080/companyprofile/swagger-ui.html#/
```

7. Share Holder Service

```
git clone https://github.com/minhdatplus/crawl-shareholdeer
```

Thay đổi đường dẫn trong file application.properties với đường dẫn của database và fundamental-service đã triển khai phía trên

```
url.fundamental.host=<fundamental_service_url>  
spring.datasource.url=jdbc:oracle:thin:@<database_url>:1521:xe
```

Sau đó tiến hành build docker image theo lệnh bên dưới

```
docker build -t share-holder-service:1.0 .  
docker run -it -d -p 8080:8080 share-holder-service:1.0 .
```

Truy cập API Document tại đường dẫn

```
http://localhost:8080/shareholder/swagger-ui.html#/
```

8. Technical Service

```
git clone https://github.com/minhdatplus/technical-service
docker-compose up --build
```

9. Single Page Application

```
git clone https://github.com/minhdatplus/stocker-app
```

Cài đặt node modules

```
npm install
```

Cấu hình đường dẫn dịch vụ fundamental-service đã triển khai phía trên tại đường dẫn

```
src/services/const/server.const.js
export const host = '<fundamental_service_url>;
```

Sau đó tiến hành build docker image theo lệnh bên dưới

```
docker build -t stocker-app:1.0 .
docker run -it -d -p 8000:8000 stocker-app:1.0 .
```

9. Amibroker pluginn

Building using Visual Studio 2019 and **vcpkg** Visual Studio 2019 install with feature **Desktop development with C++** and **Individual feature: Windows 10 SDK, MSVC v140 – VS 2015 C++ buiding tools (v14.00)**

Install **vcpkg**

Install vcpkg via **git** source:

```
git clone https://github.com/Microsoft/vcpkg.git
cd vcpkg
./bootstrap-vcpkg.sh
```

Custom triplet

Edit default windows triplet: **x86-windows**, **x64-windows** :

```
#%vcpkg_root%/triplets/x64-windows.cmake
set(VCPKG_TARGET_ARCHITECTURE x64)
set(VCPKG_CRT_LINKAGE dynamic)
set(VCPKG_LIBRARY_LINKAGE dynamic)
set(VCPKG_PLATFORM_TOOLSET "v140")
set(VCPKG_DEP_INFO_OVERRIDE_VARS "v140")

#%vcpkg_root%/triplets/x86-windows.cmake
set(VCPKG_TARGET_ARCHITECTURE x86)
set(VCPKG_CRT_LINKAGE dynamic)
set(VCPKG_LIBRARY_LINKAGE dynamic)
set(VCPKG_PLATFORM_TOOLSET "v140")
set(VCPKG_DEP_INFO_OVERRIDE_VARS "v140")
```

Install **cpprestsdk** via **vcpkg** and integrate with Visual Studio

```
cd %vcpkg_root%
vcpkg install cpprestsdk
vcpkg install cpprestsdk:x64-windows
vcpkg integrate project
```

Copy output last command and run it in **Visual Studio**