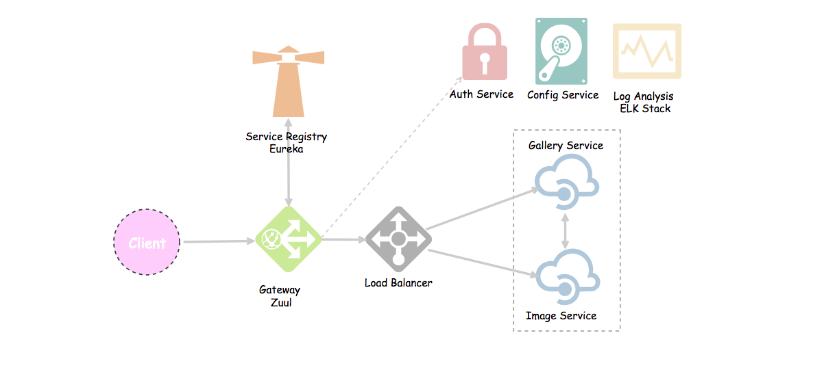


Có tổng cộng 9 service chính chạy gần như độc lập với nhau (\*-service). Đây là công nghệ Microservice. Microservices là Một giải pháp cho phép chia một hệ thống lớn thành một vài các component độc lập về phát triển, test và deploy.



@Eureka Server

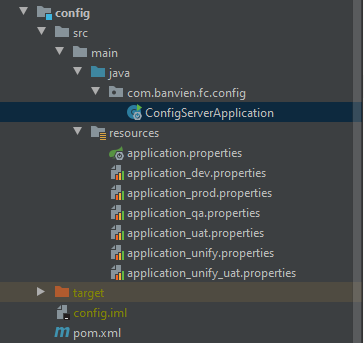
Đây là một máy chủ dùng để quản lý, đặt tên cho các service, hay còn gọi là service registry ( trong trường hợp này là registry-service)

Thứ tự chạy Microservice :

Config -> Registry -> (Các service) -> gateway

Các service sẽ được tiêm vào Register, sau đó với Gateway sẽ được hỏi Registry các Service cụ thể khi có request đến.

Nếu chúng ta có nhiều instance của một service, mỗi instance lại sử dụng một port khác nhau. Vậy làm thế nào chúng ta có thể gọi tất cả các service từ browser và phân tán những request đến các instance đó thông qua các cổng khác nhau? Câu trả lời là sử dụng một @Gateway. Một gateway là một entry point đơn trong hệ thống, được dùng để handle các request bằng cách định tuyến chúng đến các service tương ứng.





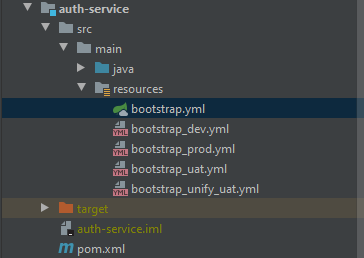


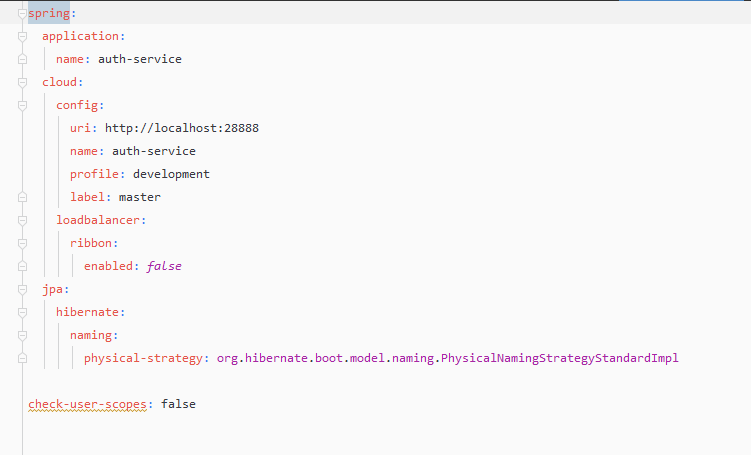
**Config**  :

* Setup port khởi chạy chính config service này.
* Datasource : Tại DB này là chuyên để lưu trữ toàn bộ Config của toàn hệ thống các Service bao gồm : ports của các service, URL đến các Database và Schema của các Service, url nơi lưu trữ file upload, Redis -id…

Có nhiều **application.properties** mục đích dễ dàng setup cùng một SourceCode nhưng có thể thành nhiều Version khác nhau : hoặc tất cả cùng chạy trên 1 server device (khác ports); hoặc có thể chạy với các DataSource khác nhau dễ dàng.

Với mỗi **application.properties** muốn sử dụng khi truyền Parameter để start môi trường đó lên (dev/prod/qa/..) cần lưu ý điều chỉnh trong file pom.xml các file tương ứng với Parameter (Id) và resources.



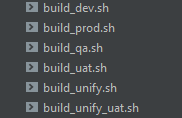


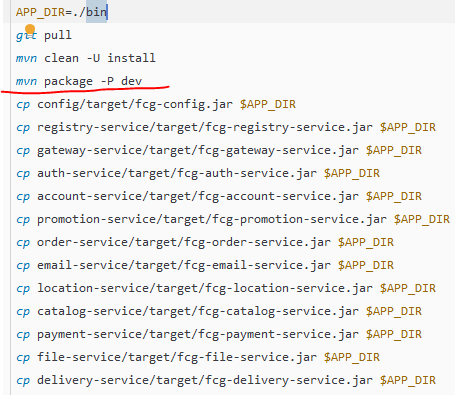
*< bootstrap.yml>*



Ở mỗi service cần có 1 file *bootstrap.yml/ bootstrap.yaml* để config Service với Register Port (28888)

Tương tự nếu có nhiều các Môi trường khác nhau nhưng dùng chung source code cần có nhiều file *bootstrap.yml* và setup chúng trong pom.xml



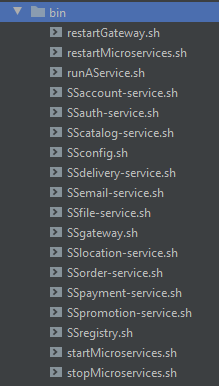


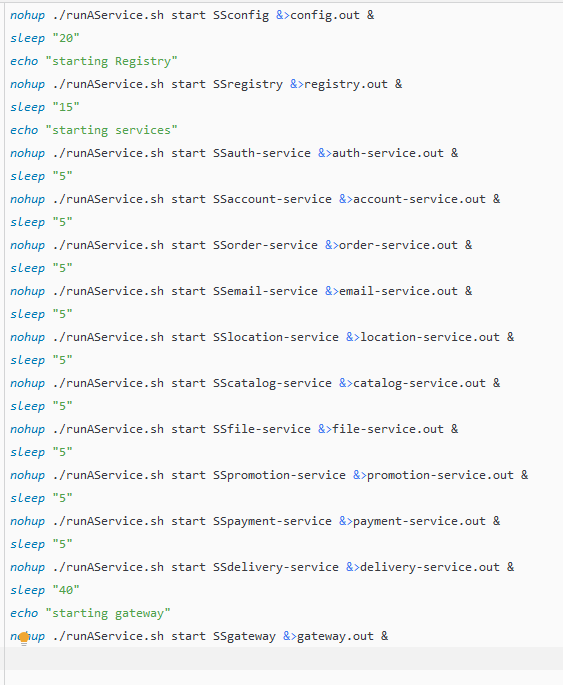
Nên tạo thành những đoạn script chứa sẵn để dễ dang thực hiện việc Build lại Source dựa trên Git cũng như dùng Maven.

Trong mỗi file build\*.sh

*mvn* package -P dev

sẽ quyết định việc Source này sẽ phục vụ cho môi trường nào (lưu ý các -P phải giống nhau trên toàn Project : pom, config, boostrap file).





*<startService.sh>*



*<stopService.sh>*