

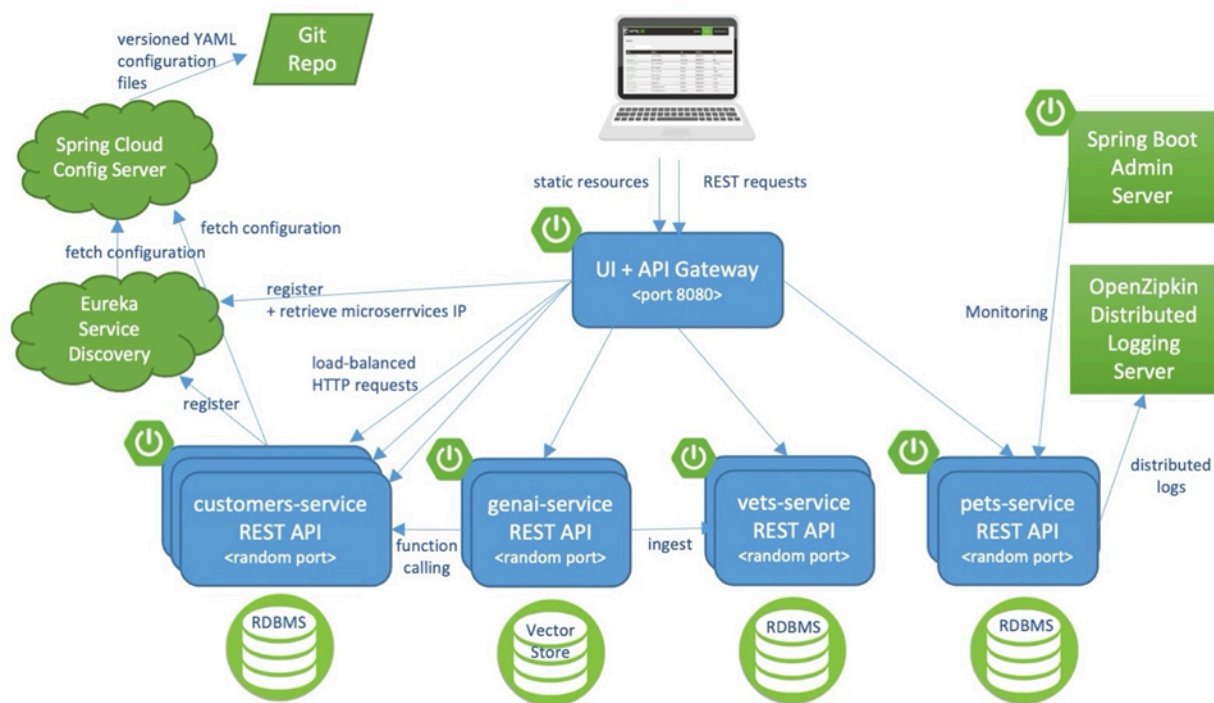
Đồ án 2: Triển khai Jenkins để xây dựng hệ thống CD

I. Mô tả:

Trong môn học này các bạn được yêu cầu xây dựng một quy trình, hệ thống ci/cd và monitor để có thể deploy, vận hành và giám sát được hệ thống petclinic từ link sau:

<https://github.com/spring-petclinic/spring-petclinic-microservices>

Petclinic là hệ thống dùng để quản lý việc khám và trị bệnh của thú cưng được xây dựng dựa trên kiến trúc microservices như sau



Eureka-Service(Discovery-server): đây là dịch vụ cho phép đăng ký và tìm kiếm các service trong hệ thống

Admin-server: dùng để quản lý và monitor các service bên trong hệ thống petclinic

Zipkin: sử dụng cho tracing là distributed logging.

API-Gateway: cung cấp UI và gateway cho các internal service

Customers-service: đây là dịch vụ quản lý khách hàng

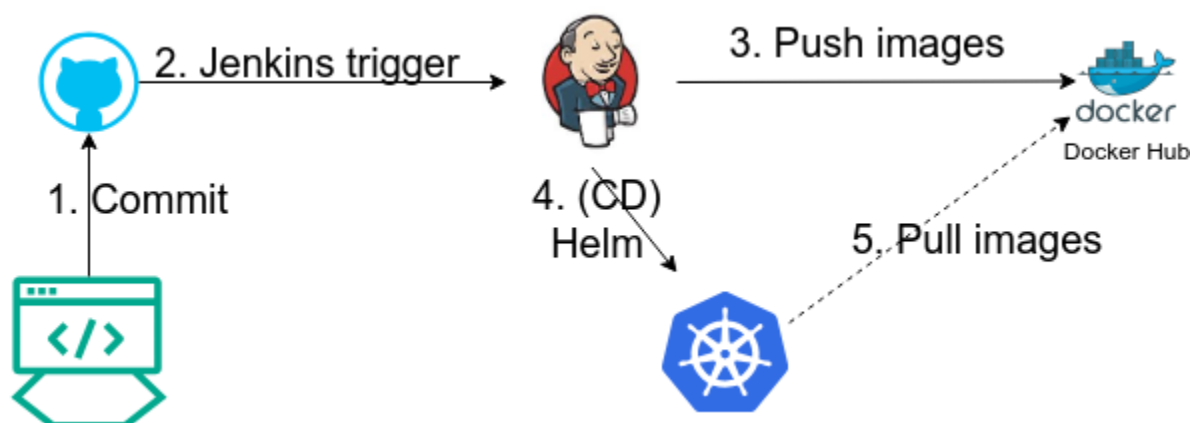
Genai-service: cung cấp hệ thống chatbot

Vets-service: Quản lý thông tin về bác sĩ thú y

Visit-service(Pets): Quản lý thông tin về lần khám chữa bệnh cho pet của khách hàng.

II. Yêu cầu

Đây là đồ án 1 trong chuỗi đồ án môn học DevOps, trong đồ án này các bạn cần phải sử dụng Jenkins để xây dựng pipeline cho quá trình CD với những yêu cầu cụ thể sau (8đ):



1. Mặc định các bạn sẽ có 1 image cho tất cả các service với tag là main hoặc latest, bạn không cần triển khai grafana và Prometheus trong đồ án này.
2. Xây dựng K8s cluster với 1 Master node và 1 worker Node (Hoặc **Minikube**)
3. Phần CI, với mỗi branch của user tạo, sau khi user commit code thay đổi, bạn phải build ra một image với tag là commit id cuối cùng của branch đó, và push image đó lên Docker Hub.
4. Tạo Job CD cho developer làm việc với tên developer_build. Với job này developer có thể input parameter là branch muốn deploy... (hoặc sv tự quyết định).

Ví dụ: developer đang làm việc ở branch: **dev_vets_service** và update code trong service này. Developer cần biết được sau khi sửa code, thì muốn test thử. Lúc này developer sẽ vào "developer_build" job để điền phần "vets-service" parameter là: **dev_vets_service**, còn các branch còn lại là main. Khi đó bạn sẽ deploy code của tất cả các service còn lại theo default là tag main hoặc latest, còn "dev_vets_service" sẽ là image với tag ở mục 3.

customers-service

main

visits-service

main

vets-service

main

genai-service

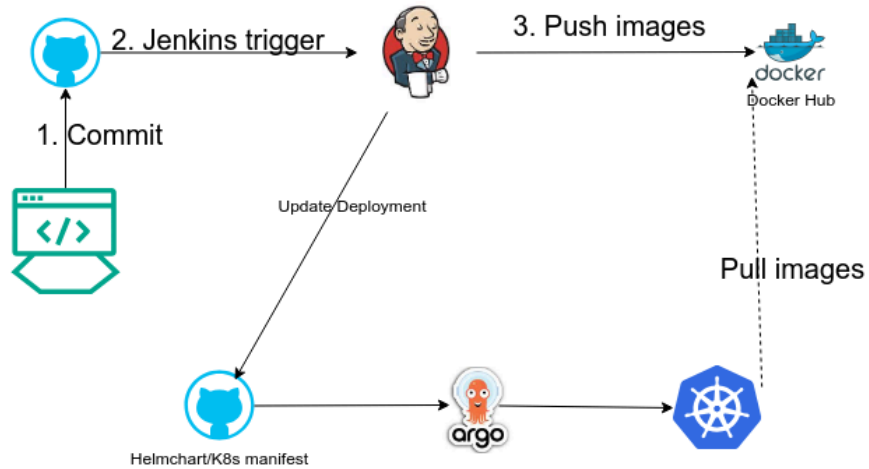
main

Build Cancel

Sau khi deploy, bạn cung cấp domain name:port (dạng service là NodePort), để developer có thể truy cập và test code của mình trực tiếp. Phần domain name, do mình không có dns, vì vậy developer sẽ tự thêm vào file hosts của mình trên máy để chỉ đến Worker node của K8s cluster

5. Tạo Jenkins job để xóa phần triển khai ở mục 4
(<https://community.jenkins.io/t/how-to-add-hyperlink-using-jenkins-job-builder/7091>)
6. Bỏ qua phần này nếu làm phần Nâng Cao: Tương tự, trên Jenkins tạo ra 2 job CI/CD để deploy "dev" và "staging".
 - a. main thay đổi, auto sẽ deploy đề liên tục vào trong namespace dev
 - b. Staging: trên "main" branch sẽ có đánh tag để có dạng release, ví dụ: tag v1.2.3. thì job CI/CD sẽ phát hiện và build image với tag cuối cùng, ví dụ tag v1.2.3 (hoặc tách branch rc_v1.2.3, hoặc vừa tag và tách branch), sau đó push images này lên Docker Hub và deploy vào trong namespace "staging"

Nâng cao (2đ): Sử dụng AgroCD để handle được "dev" và "staging"



III. Qui định

1. Đồ án làm nhóm 4 sinh viên
2. Thời gian làm bài 2 tuần (16/04/2025)
3. Nộp bài: Các bạn tạo file báo cáo gồm các thông tin sau
 - b. Chụp hình các bước các bạn cấu hình
 - c. Đặt tên file theo format <MSSV1>_<MSSV2>_<MSSV3>.docx. Thứ tự MSSV cần được sắp xếp tăng dần. Ví dụ nhóm có 3 SV là 23120000, 23120001, 23120002 thì đặt tên file là 23120000_23120001_23120002.docx, nếu có 2 sinh viên thì đặt tên 23120000_23120001.docx, nếu chỉ có 1 sinh viên thì đặt tên 23120000.docx