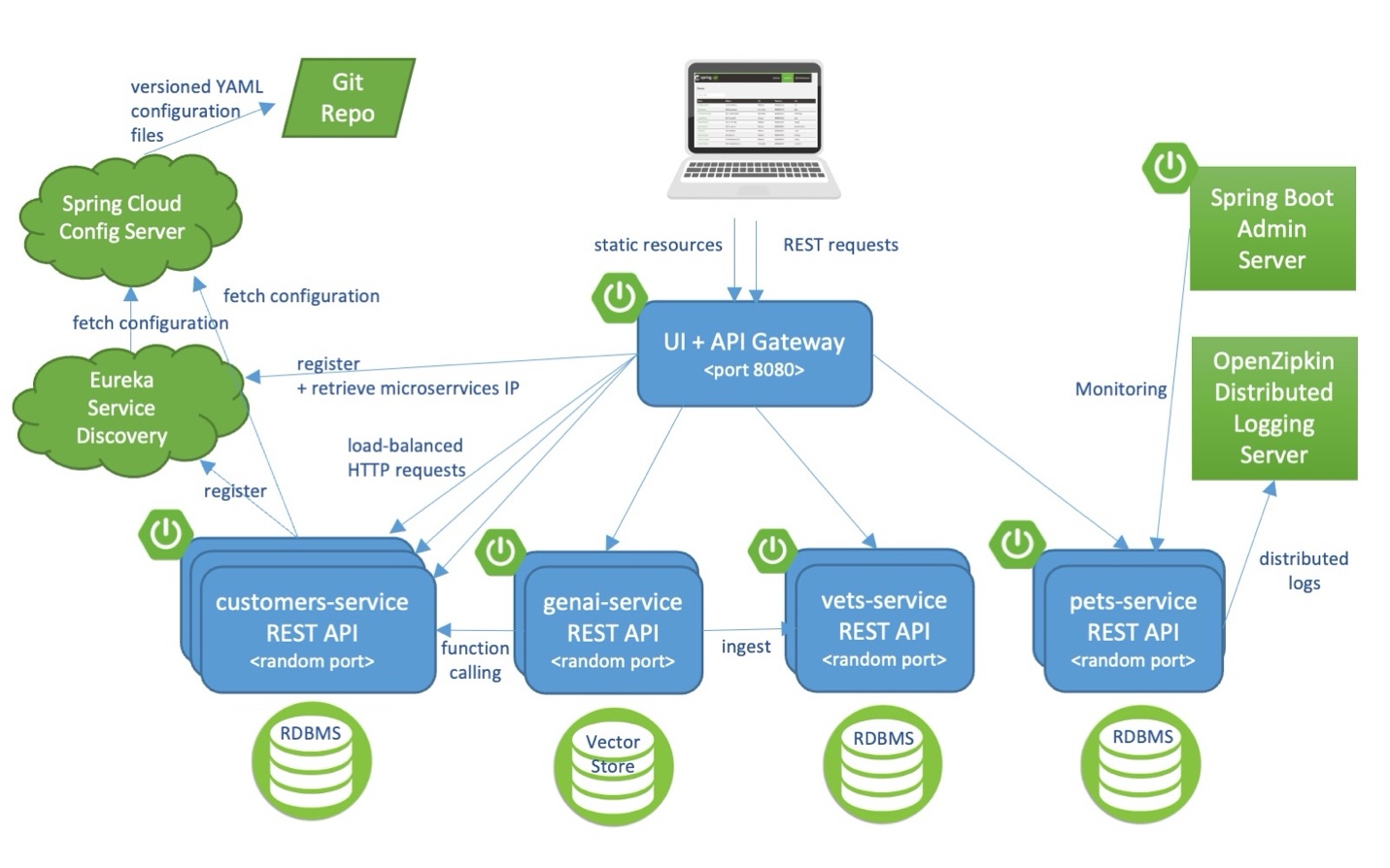
**Đồ án 1:** Triển khai Jenkins để xây dựng hệ thống CI

**1. 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.

**2. 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 CI (Continous Intergration) với những yêu cầu cụ thể sau

1. Setup Jenkins và agent <https://www.jenkins.io/>. Các bạn có thể sử dụng máy ảo hoặc máy thật để setup, khuyến khích sử dụng multipass để tạo máy ảo <https://canonical.com/multipass>.
2. Fork một repo mới từ github <https://github.com/spring-petclinic/spring-petclinic-microservices> cho riêng nhóm của mình
3. Cấu hình github để không cho phép push trực tiếp vào main branch. Mỗi PR cần ít nhất 2 reviewer approve và CI pass mới cho phép merge vào main branch.
4. Configure để jenkins có thể quét và chạy pipeline cho từng branch.
5. Pipeline cần có ít nhất 2 phase test và build. Phase test cần upload test result và độ phủ của testcase.
6. Do hệ thống này đang sử dụng mô hình monorepo các bạn cần phải cấu hình để Jenkins chỉ kích hoạt pipeline cho service cụ thể khi có thay đổi trong thư mục của service đó. Ví dụ khi developer thay đổi trong service vets-service thì chỉ cần build và test lại vets-service, chứ không build và test lại toàn bộ hệ thống.
7. **Yêu cầu nâng cao:** 
   1. thêm unit test vào trong code để có thể tăng độ phủ của testcase. Trong yêu cầu này các bạn cần tạo branch mới để thêm testcase, mỗi branch sẽ ứng với mỗi service customer, vets và visit.
   2. Điều chỉnh lại pipeline để chỉ cho phép pass khi testcase có độ phủ > 70%

**3. Qui định**

1. Đồ án làm nhóm 3 sinh viên
2. Thời gian làm bài 3 tuần (18/3/2025)
3. Nộp bài: Các bạn tạo file báo cáo gồm các thông tin sau
   1. Link tới github repository của nhóm, trong repository này tối thiếu phải có 1 PR đang trong trạng thái open
   2. Chụp hình các bước các bạn cấu hình github, cài đặt và cấu hình Jenkins
   3. 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