Use multiple (seven) terminals to perform the following steps:

**1) Start the discovery-server application (Eureka app):**

cd vaadin-microservices-demo/discovery-server

java -jar target/discovery-server-0.0.1-SNAPSHOT.jar

**2) Start the config-server application (Spring Cloud Config app):**

cd vaadin-microservices-demo/config-server

java -jar target/config-server-0.0.1-SNAPSHOT.jar

**3) Start an instance of the biz-application microservice (REST app):**

cd vaadin-microservices-demo/biz-application

java -jar target/biz-application-0.0.1-SNAPSHOT.jar

**4) Start an instance of the admin-application microservice (Vaadin app):**

cd vaadin-microservices-demo/admin-application

java -jar target/admin-application-0.0.1-SNAPSHOT.jar

**5) Start an instance of the news-application microservice (Vaadin app):**

cd vaadin-microservices-demo/news-application

java -jar target/news-application-0.0.1-SNAPSHOT.jar

**6) Start an instance of the website-application microservice (Vaadin app):**

cd vaadin-microservices-demo/website-application

java -jar target/website-application-0.0.1-SNAPSHOT.jar

**7) Start the proxy-server application (Zuul app):**

cd vaadin-microservices-demo/proxy-server

java -jar target/proxy-server-0.0.1-SNAPSHOT.jar

You'll see the website-application embedding the admin-application and the news-application microservices.

This is the "edge service" implemented with Netflix Zuul. It acts as a reverse proxy, redirecting requests to the website-application, news-application, and admin-application instances using a load balancer provided by Netflix Ribbon with a *round robin* strategy.

If you get a "Server not available" message, please wait until all the services are registered with the discovery-server (implemented with Netflix Eureka).

**2) Add, update, or delete data.**

Latest tweets from the companies you enter on the left (the admin-application) will be rendered on the right (the news-application).

The admin-application, and news-application instances (implemented with Vaadin) delegate CRUD operations to the biz-application (implemented with Spring Data Rest) using a load balancer (provided by Netflix Ribbon) with a *round robin* strategy.

**3) Add microservice instances.**

You can horizontally scale the system by starting more instances of the biz-application, admin-application, news-application, and website-application microservices. Remember to specify an available port (using -Dserver.port=NNNN) when you start a new instance.

**4) Test high-availability.**

Make sure you are running two instances of the admin-application. Click the *+* (Add) button and enter Vaadin as the *name*, and vaadin as the *Twitter Username*. Don't click the *Add* button yet.

Stop one of the instances of the admin-application and click the *Add* button. The web application should remain functional and save the data you entered without losing the state of the UI thanks to the externalized HTTP Session (implemented with Spring Session and Hazelcast).

**5) Test system resilience.**

Stop all the instances of the biz-application microservice and refresh the browser to see the fallback mechanisms (implemented with Netflix Hystrix) in the admin-application and news-application microservices.