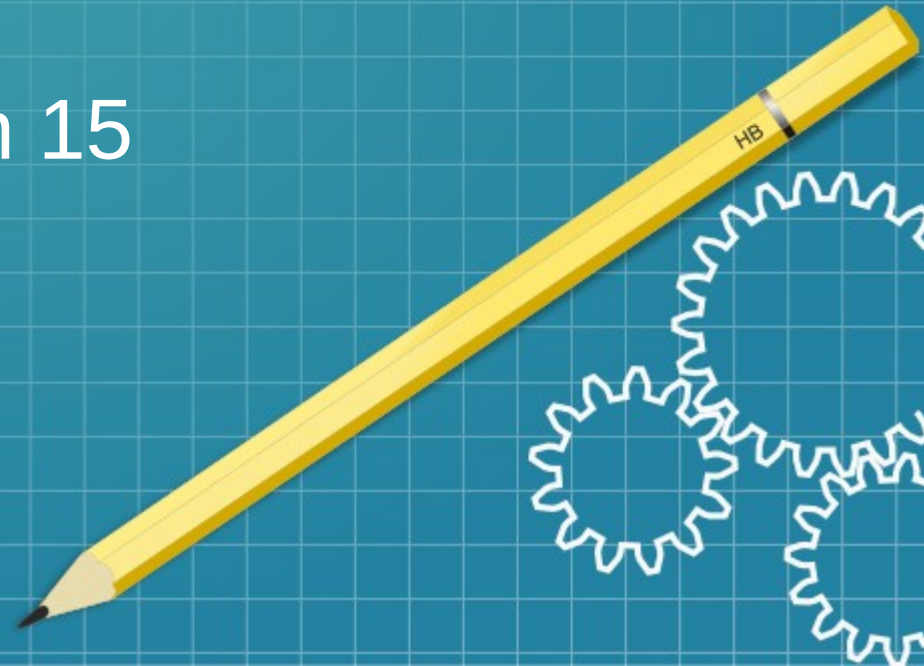




Administrasi & Rilis Spring

Pertemuan 15

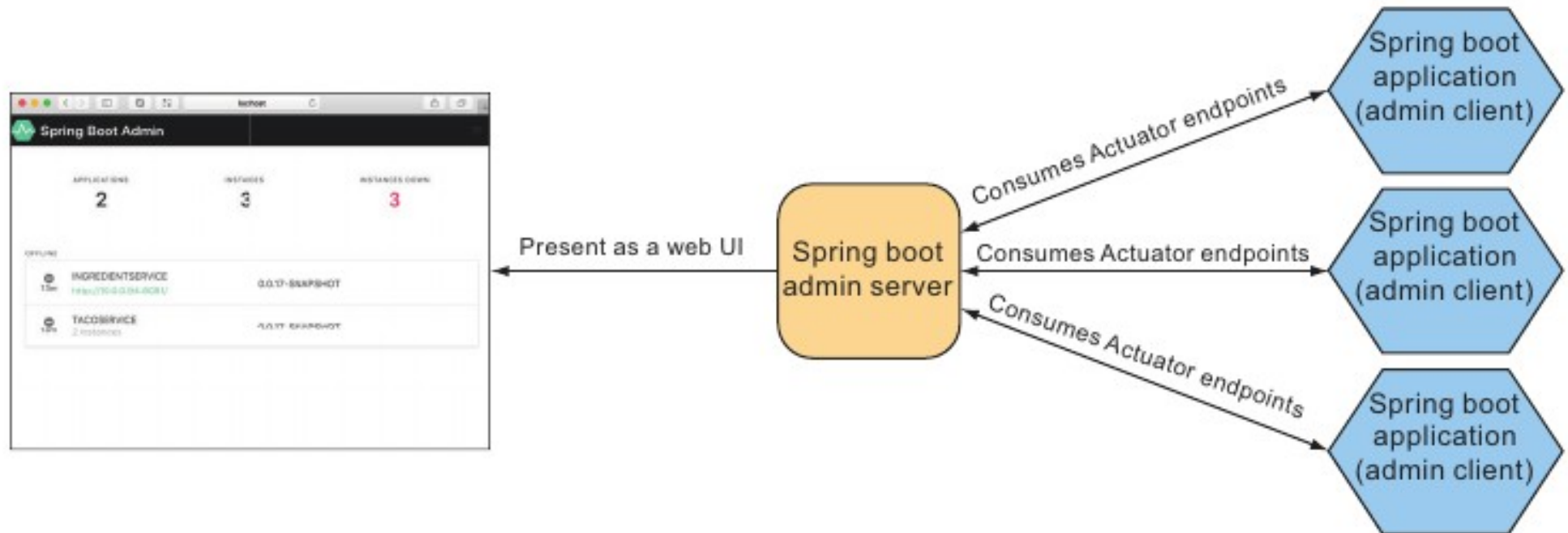


Spring Boot Admin



- Spring Boot Admin adalah aplikasi web administrasi yang membuat endpoint Actuator menjadi lebih bisa digunakan oleh manusia
- Terbagi menjadi dua komponen utama
 - Spring Boot Admin server
 - Dan klien-kliennya

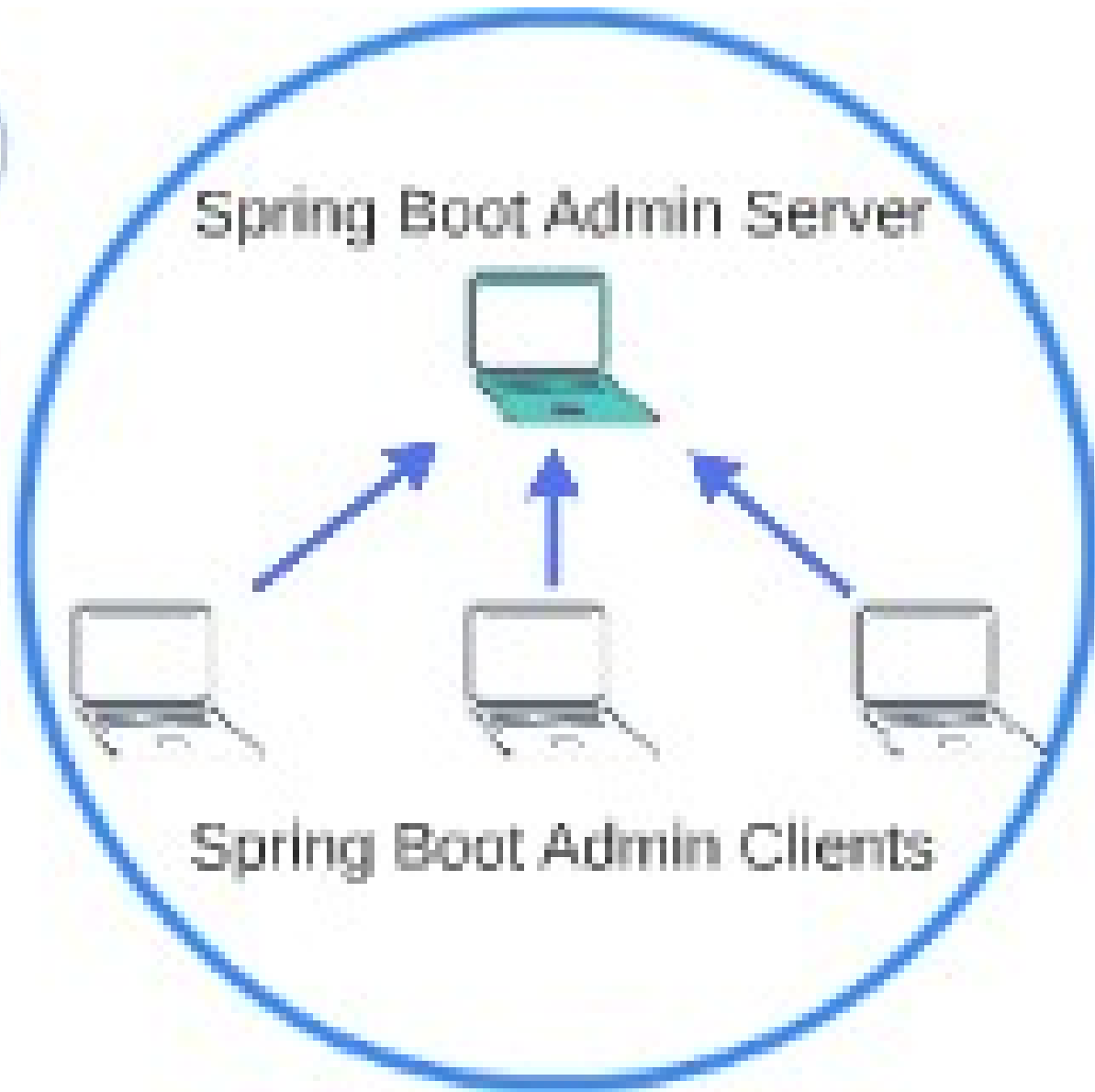
Lanjutan



Aplikasi-aplikasi yang ada (microservice) harus di daftarkan sebagai klien

Spring Boot Admin server harus dikonfigurasi agar bisa menerima informasi kliennya

Lanjutan



Membuat Server Admin



- Agar bisa mempunyai Admin server, anda diharuskan membuat satu aplikasi Spring Boot baru dan menambahkan dependency nya ke projek
- Admin server secara umum digunakan sebagai aplikasi mandiri yang terpisah dari aplikasi lain

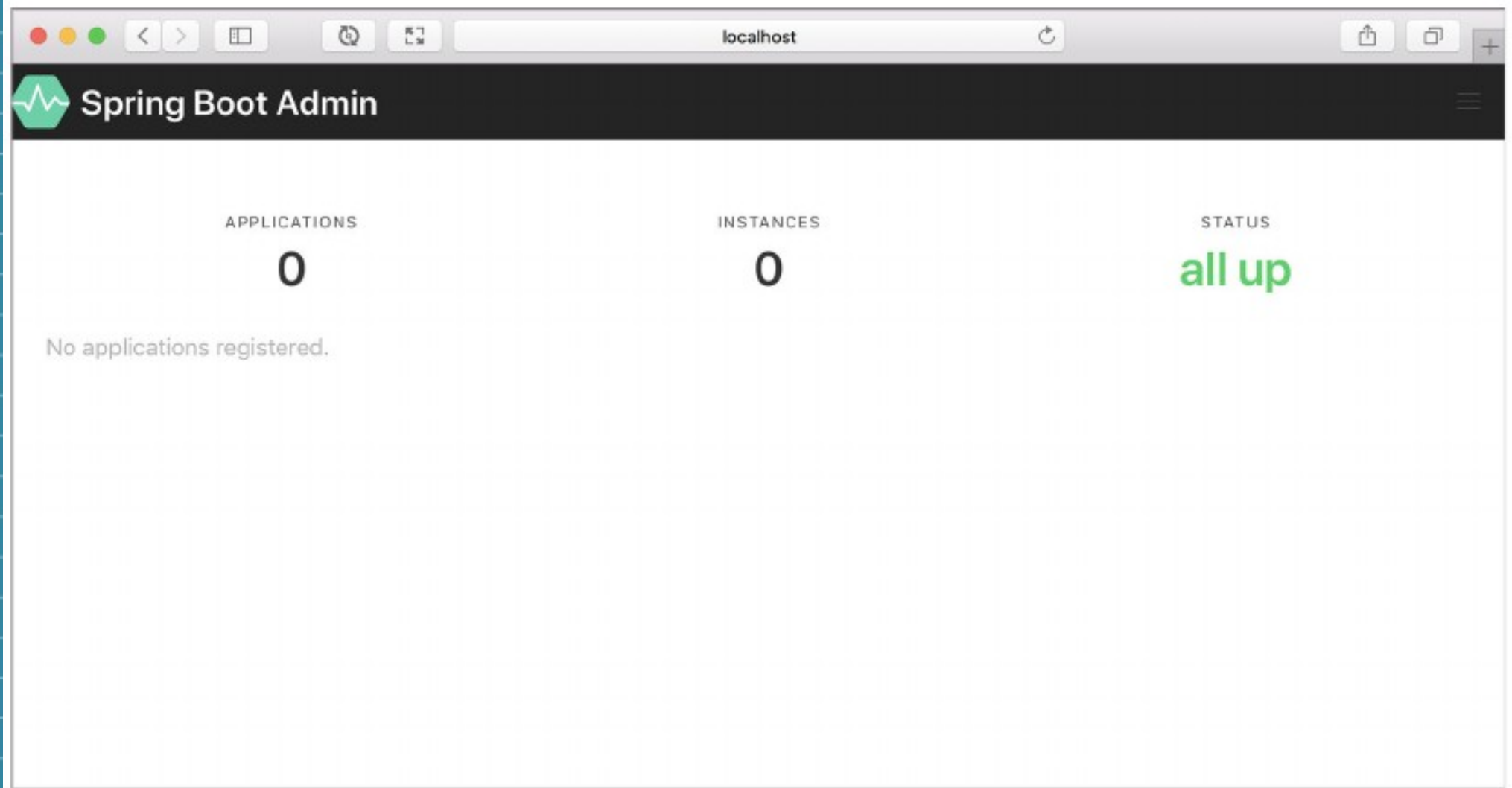
Mendaftarkan Klien



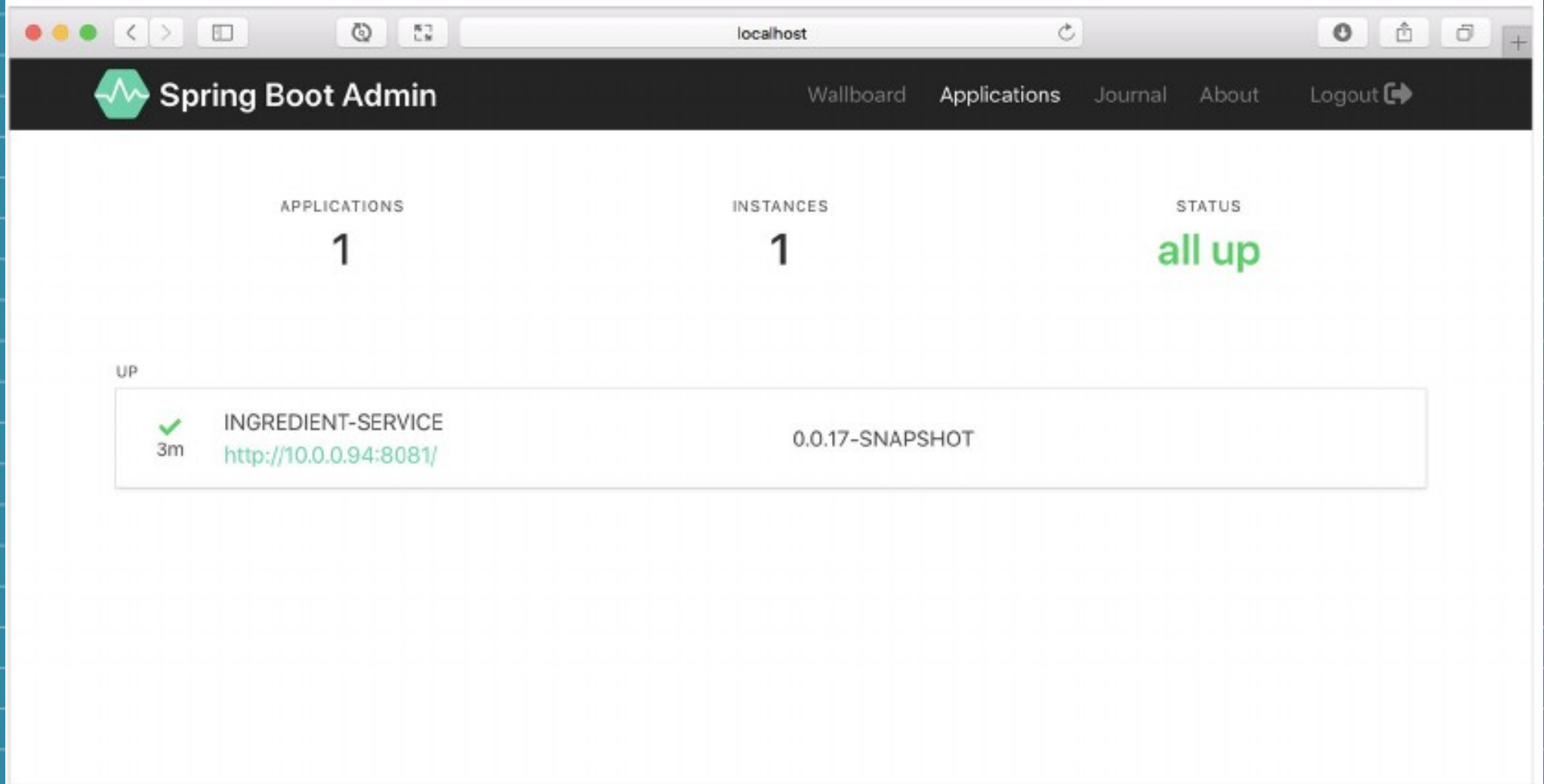
Dikarenakan Admin Server adalah aplikasi yang terpisah dari aplikasi lainnya yang menyajikan data Actuator.

- Maka Admin server harus mengetahui aplikasi mana yang harus dilihat
- Dua cara mendaftarkan Spring Boot Admin clients:
 - Aplikasi secara eksplisit mendaftarkan secara mandiri.
 - Admin server mencari services melalui Eureka service registry.

Lanjutan



Lanjutan – Register Manual



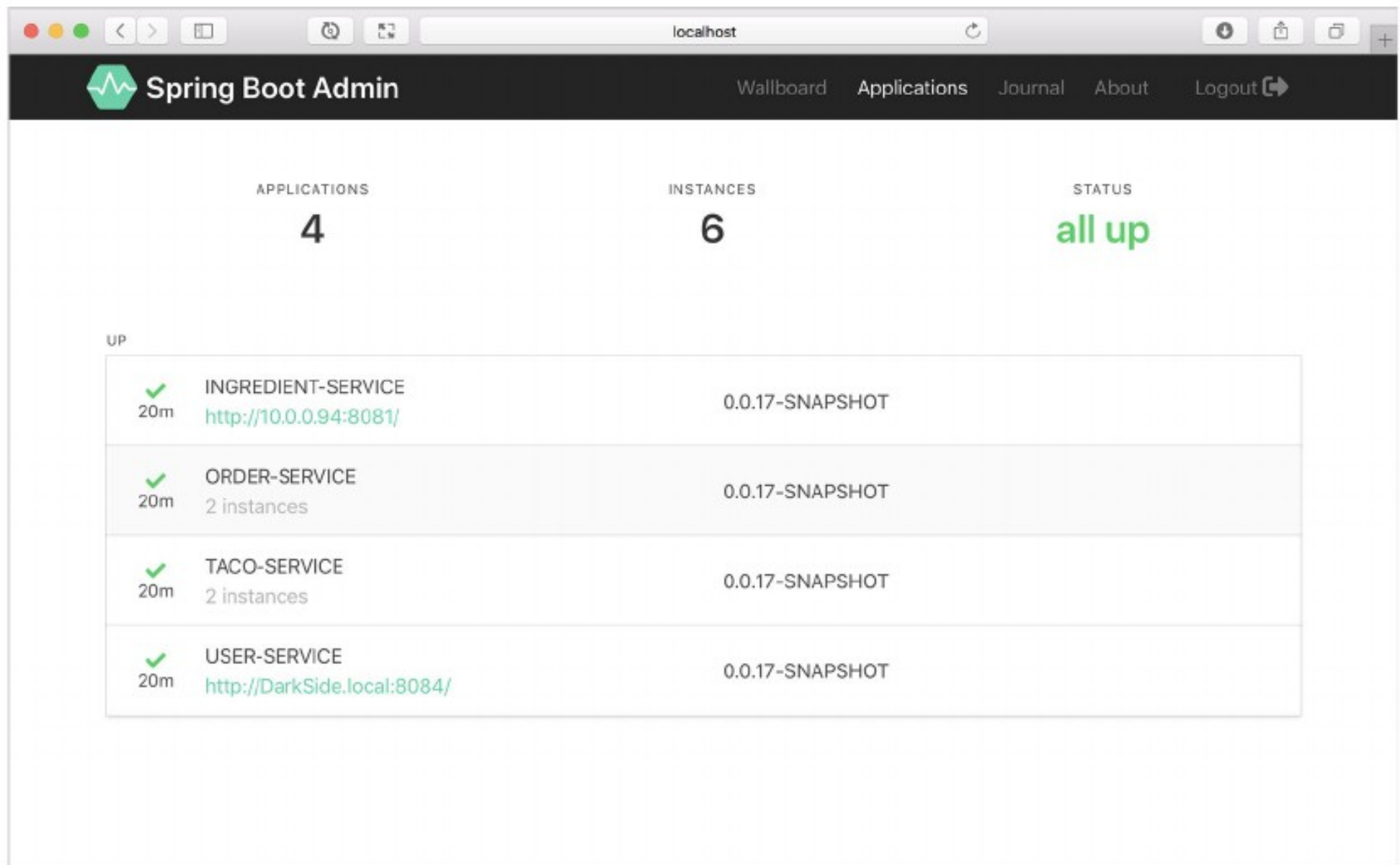
A screenshot of the Spring Boot Admin web application running on localhost. The interface features a dark header with the Spring Boot Admin logo and navigation links: Wallboard, Applications, Journal, About, and Logout. The main content area displays three summary cards: 'APPLICATIONS' with a count of 1, 'INSTANCES' with a count of 1, and 'STATUS' showing 'all up' in green. Below these cards, a table lists the registered applications. The table has columns for application name, version, and status. One application, 'INGREDIENT-SERVICE', is listed with version '0.0.17-SNAPSHOT' and a status of 'UP' (indicated by a green checkmark and '3m' uptime). The application's URL is shown as 'http://10.0.0.94:8081/'.

APPLICATIONS	INSTANCES	STATUS
1	1	all up

UP	Application Name	Version
✓ 3m	INGREDIENT-SERVICE	0.0.17-SNAPSHOT

<http://10.0.0.94:8081/>

Lanjutan – Register Discovery



A screenshot of the Spring Boot Admin web application running on localhost. The interface shows a summary of 4 applications and 6 instances, all with a status of 'all up'. Below this, a table lists the applications: INGREDIENT-SERVICE, ORDER-SERVICE, TACO-SERVICE, and USER-SERVICE, each with a green checkmark icon and a '20m' refresh indicator. The table also shows the version '0.0.17-SNAPSHOT' for each application.

APPLICATIONS	INSTANCES	STATUS
4	6	all up

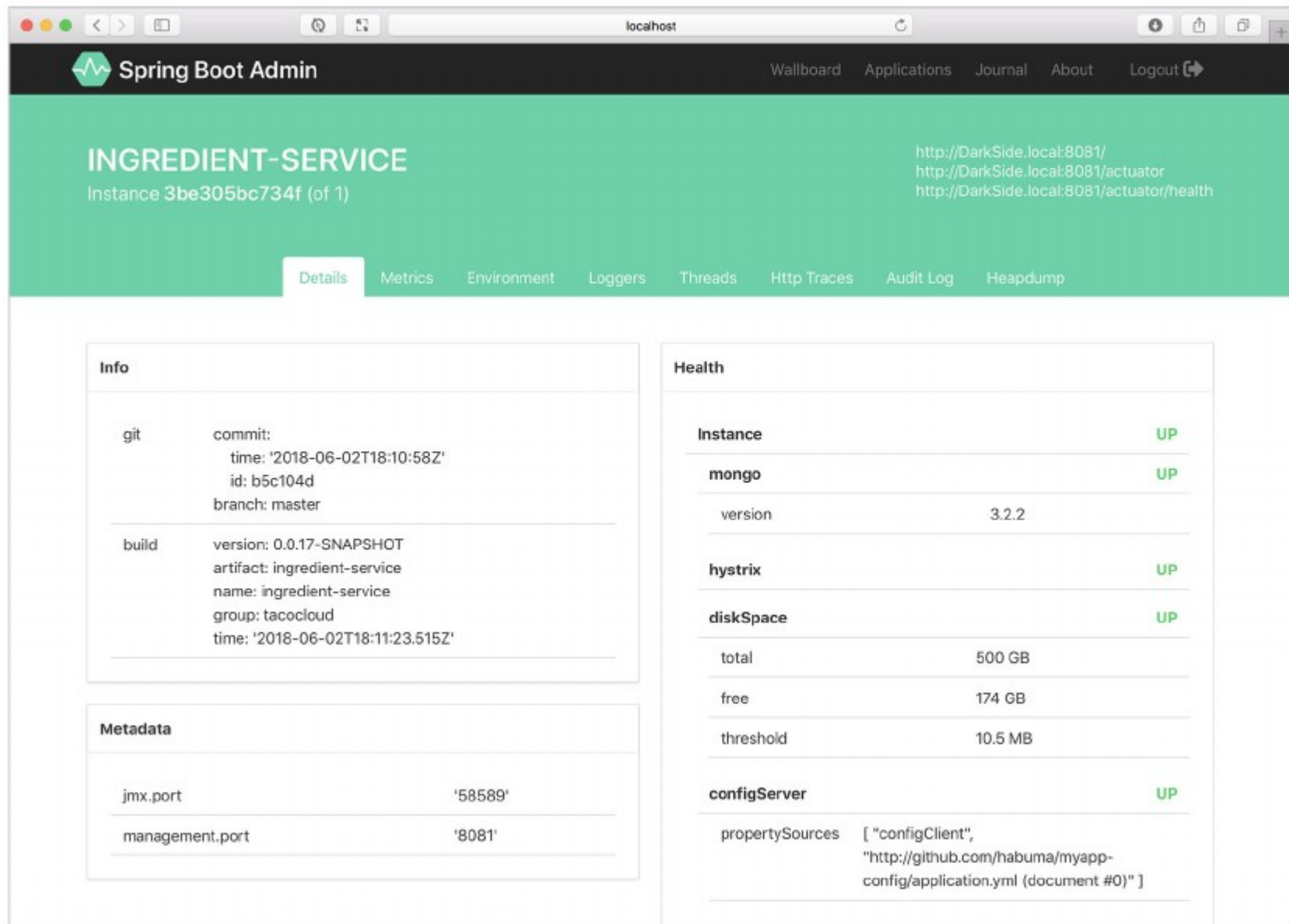
UP	Application	Version
✓ 20m	INGREDIENT-SERVICE http://10.0.0.94:8081/	0.0.17-SNAPSHOT
✓ 20m	ORDER-SERVICE 2 instances	0.0.17-SNAPSHOT
✓ 20m	TACO-SERVICE 2 instances	0.0.17-SNAPSHOT
✓ 20m	USER-SERVICE http://DarkSide.local:8084/	0.0.17-SNAPSHOT

Eksplorasi Admin Server



- Setelah klien sudah didaftarkan, ada banyak informasi yang Admin server bisa lihat di dalam aplikasi seperti:
 - Informasi dan Kesehatan Umum
 - Segala informasi metrik yang dipublikasikan melalui Micrometer ataupun /metrics endpoint
 - Environment properties
 - Tingkatan Logging
 - Detail Thread
 - HTTP traces for requests
 - Audit logs

Lanjutan



The image shows a screenshot of the Spring Boot Admin web application. The browser address bar shows 'localhost'. The page title is 'Spring Boot Admin'. The main header is green and contains the service name 'INGREDIENT-SERVICE' and its instance ID '3be305bc734f (of 1)'. To the right of the header, there are three links: 'http://DarkSide.local:8081/', 'http://DarkSide.local:8081/actuator', and 'http://DarkSide.local:8081/actuator/health'. Below the header, there is a navigation bar with tabs: 'Details', 'Metrics', 'Environment', 'Loggers', 'Threads', 'Http Traces', 'Audit Log', and 'Heapdump'. The 'Details' tab is selected. The main content area is divided into two columns. The left column contains 'Info' and 'Metadata' sections. The 'Info' section shows git commit details and build information. The 'Metadata' section shows jmx.port and management.port. The right column contains a 'Health' section with a table of service status. The table has columns for the service name and its status (UP or DOWN). The services listed are Instance, mongo, hystrix, diskSpace, and configServer, all of which are UP. The Instance row shows version 3.2.2. The mongo row shows version 3.2.2. The hystrix row shows version 1.5.7. The diskSpace row shows total 500 GB, free 174 GB, and threshold 10.5 MB. The configServer row shows propertySources as ["configClient", "http://github.com/habuma/myapp-config/application.yml (document #0)"].

Spring Boot Admin

Wallboard Applications Journal About Logout

INGREDIENT-SERVICE
Instance 3be305bc734f (of 1)

http://DarkSide.local:8081/
http://DarkSide.local:8081/actuator
http://DarkSide.local:8081/actuator/health

Details Metrics Environment Loggers Threads Http Traces Audit Log Heapdump

Info

git	commit: time: '2018-06-02T18:10:58Z' id: b5c104d branch: master
build	version: 0.0.17-SNAPSHOT artifact: ingredient-service name: ingredient-service group: tacocloud time: '2018-06-02T18:11:23.515Z'

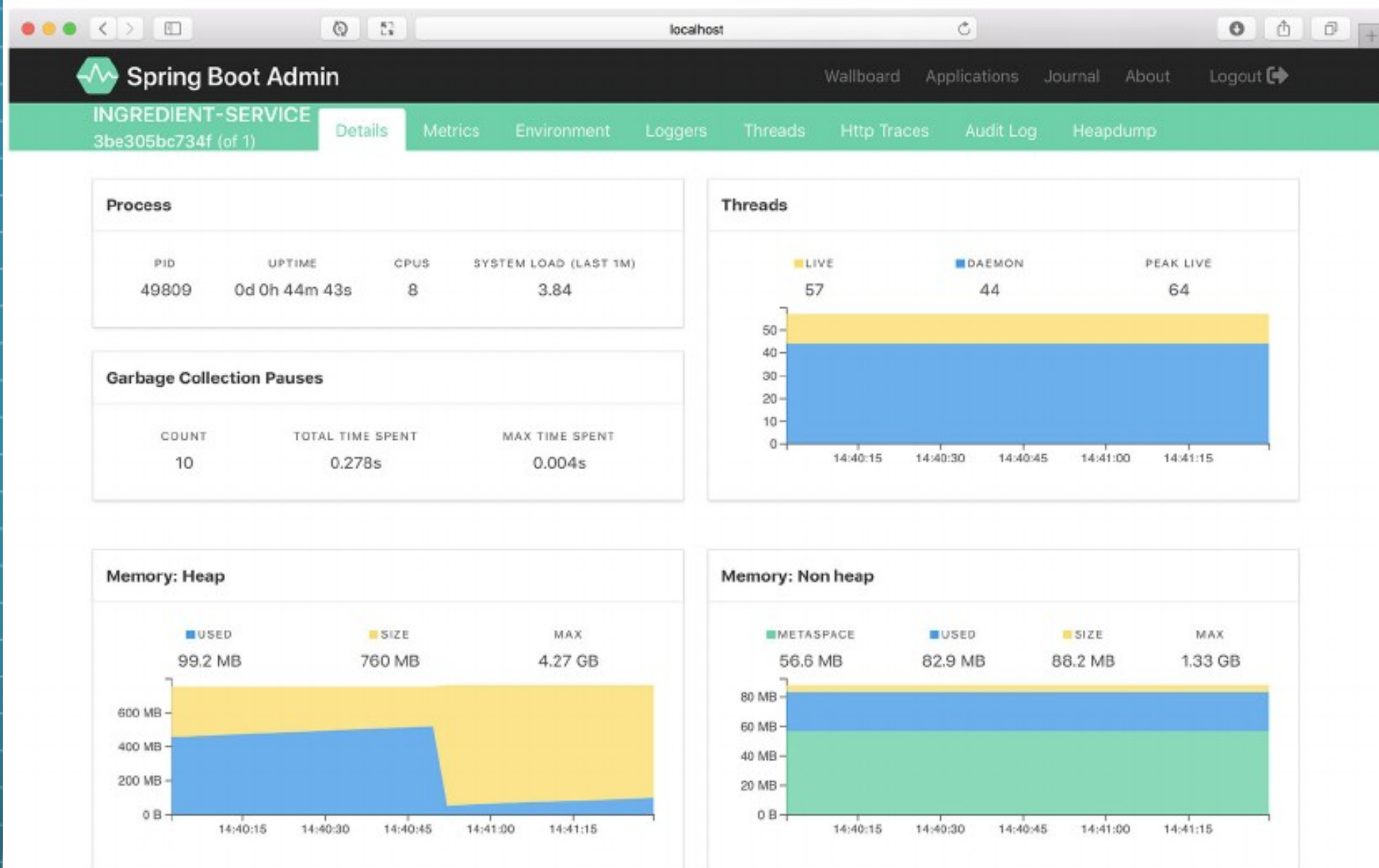
Metadata

jmx.port	'58589'
management.port	'8081'

Health

Instance	UP
mongo	UP
version	3.2.2
hystrix	UP
diskSpace	UP
total	500 GB
free	174 GB
threshold	10.5 MB
configServer	UP
propertySources	["configClient", "http://github.com/habuma/myapp- config/application.yml (document #0)"]

Lanjutan



Mengamankan Server



- Segala informasi yang disiarkan oleh Actuator endpoints tidak diperuntukkan konsumsi umum.
- Mereka mempunyai informasi detail mengenai aplikasi yang hanya boleh Administrator lihat.
- Sama seperti Actuator, Admin server juga sama pentingnya soal keamanan.

Rilis Aplikasi



- Rilis aplikasi berarti aplikasi sudah tidak dalam kotak pengembangan
- Aplikasi yang dirilis hanya bisa:
 - Di eksekusi
 - Beberapa tingkat log dimatikan
 - Tidak bisa masuk kembali ke kotak pengembangan
- Dirilis sebagai WAR/JAR

Opsi Rilis Aplikasi



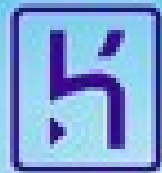
- Dijalankan langsung melalui IDE (kode masih bisa diubah)
- Dijalankan melalui perintah Maven `spring-boot:run`
- Menggunakan Maven atau Gradle untuk membuat executable JAR untuk dibuka melalui commandline
- Menggunakan maven atau Grade untuk membuat file WAR yang bisa dibuka sebagai Aplikasi Server Java Tradisional

Lokasi Rilis



- Rilis ke Server Aplikasi Java – Jika anda harus merilis aplikasi ke Tomcat, WebSphere, WebLogic, atau server aplikasi Java lainnya, maka dianjurkan untuk membuat file WAR.
- Rilis ke cloud – Jika anda berencana melakukan rilis ke cloud seperti Cloud Foundry, Amazon Web Services (AWS), Azure, Google Cloud Platform, atau sistem cloud lainnya maka membuat executable JAR adalah pilihan terbaik

Lanjutan



HEROKU



Spring Boot

Membuat WAR



- Melakukan konfigurasi pom.xml

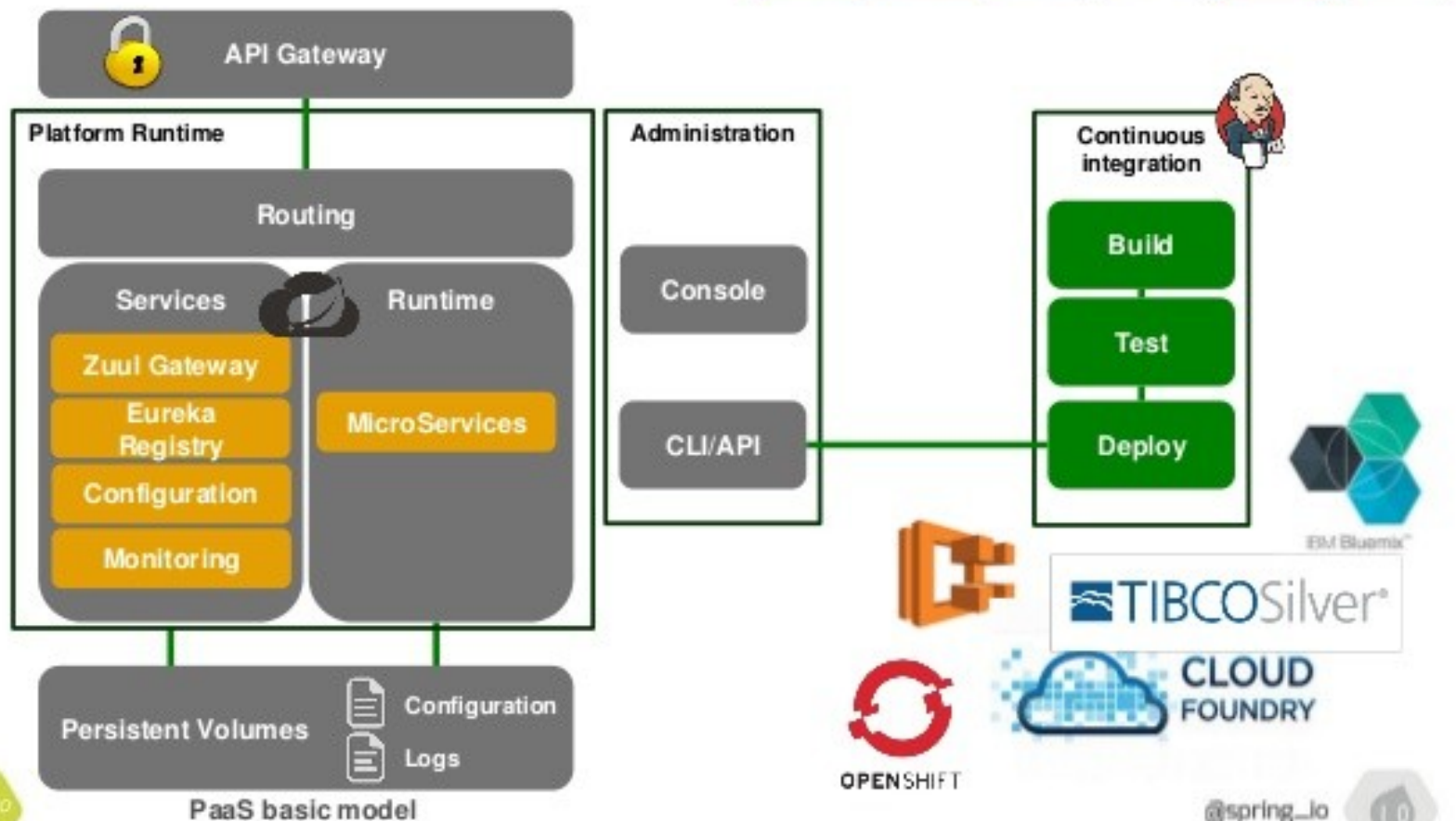
```
<packaging>war</packaging>
```
- Jika menggunakan Gradle maka, ubah gradle.build ke
 - apply plugin: 'war'
- Ketika aplikasi siap untuk dirilis, cukup ketik perintah
 - \$ mvnw package
- Atau
 - \$ gradlew build

Membuat JAR



- Cukup dengan mengubah pom.xml dan gradle.build ke jar
- Lalu gunakan Maven atau Gradle untuk build
- File Jar bisa dieksekusi menggunakan perintah
 - Java -jar aplikasi.jar

Lanjutan



Integrasi Docker

Spring Boot



Lanjutan



- Dengan maven menggunakan perintah
`mvn clean package docker:build`
- Untuk menjalankan aplikasi
`docker run -p 8080:8080 -d
springframeworkguru/masteringthymeleaf`
- Gunakan Docker Start dan Stop untuk
mengontrol Docker