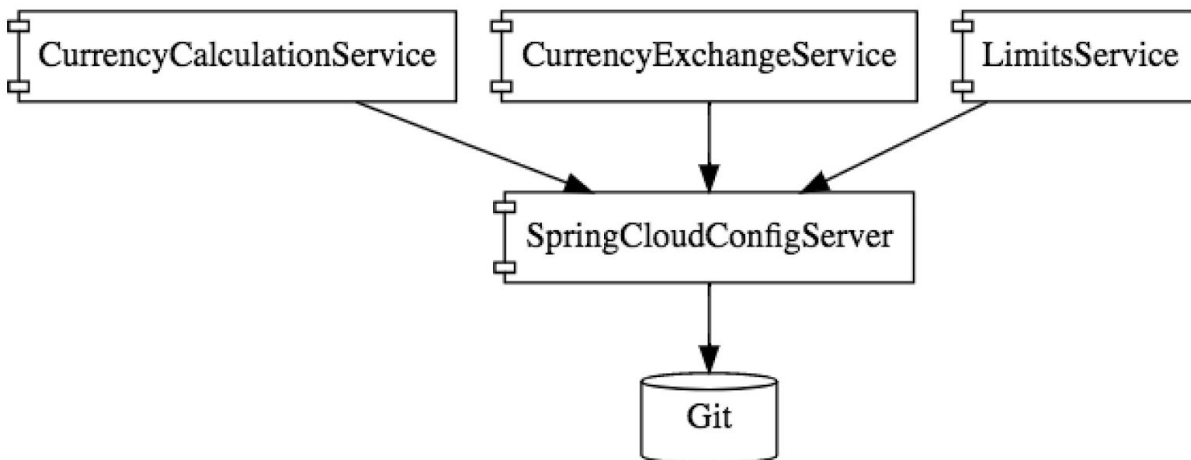


| | |
|---|----------|
| Centralized Configuration | 2 |
| Service Discovery Load Balancer API Gateway | 2 |
| Application Performance Monitoring Using micrometer and prometheus | 3 |
| Distributed Tracing | 3 |
| Frameworks Used | 3 |
| Spring Boot Admin | 4 |
| Frameworks Used | 4 |
| Eureka | 4 |
| Zuul | 4 |
| Feign | 5 |
| Ribbon | 5 |
| Hystrix | 5 |
| Zipkin | 6 |
| Prometheus | 6 |
| Running the Code | 6 |
| Installing Rabbit MQ | 6 |
| Windows | 6 |
| Mac | 6 |
| Setting up Distributed Tracing with Zipkin | 6 |
| Command to run | 6 |
| Setup Prometheus | 6 |
| Start Services | 7 |
| Screen shots | 7 |
| Spring Boot Admin | 7 |
| Distributed Trace | 9 |
| Service Dependency | 9 |
| Find distributed trace | 10 |
| Drill down a HTTP trace | 11 |
| APM Using Prometheus | 11 |

Centralized Configuration

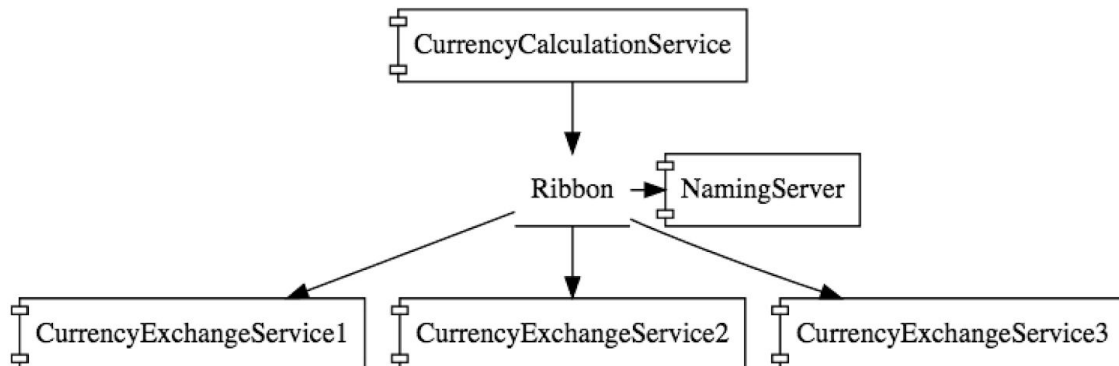


Code -

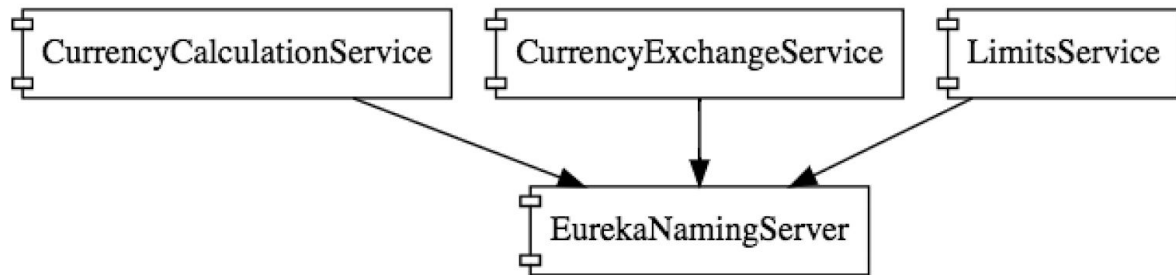
<https://github.com/anilgmaipady/spring-cloud-example/tree/master/spring-cloud-config-server>

- Configuration separated from code and centralized.
- Changes are pushed to all services without service restart.

Service Discovery Load Balancer API Gateway



All services register themselves with naming server.



Ribbon - Client side load balancer

<https://github.com/anilgmaipady/spring-cloud-example/tree/master/currency-conversion-service>

Eureka - Service registry (Naming server)

<https://github.com/anilgmaipady/spring-cloud-example/tree/master/netflix-eureka-naming-server>

Zuul - API Gateway

<https://github.com/anilgmaipady/spring-cloud-example/tree/master/netflix-zuul-api-gateway-server>

Application Performance Monitoring Using micrometer and prometheus

- All services exposes /actuator/prometheus REST end points.
- Prometheus stores and exposes time series metrics for visualization frameworks.
- <https://micrometer.io/>
- <https://prometheus.io/>

Distributed Tracing

Code is part of the services:

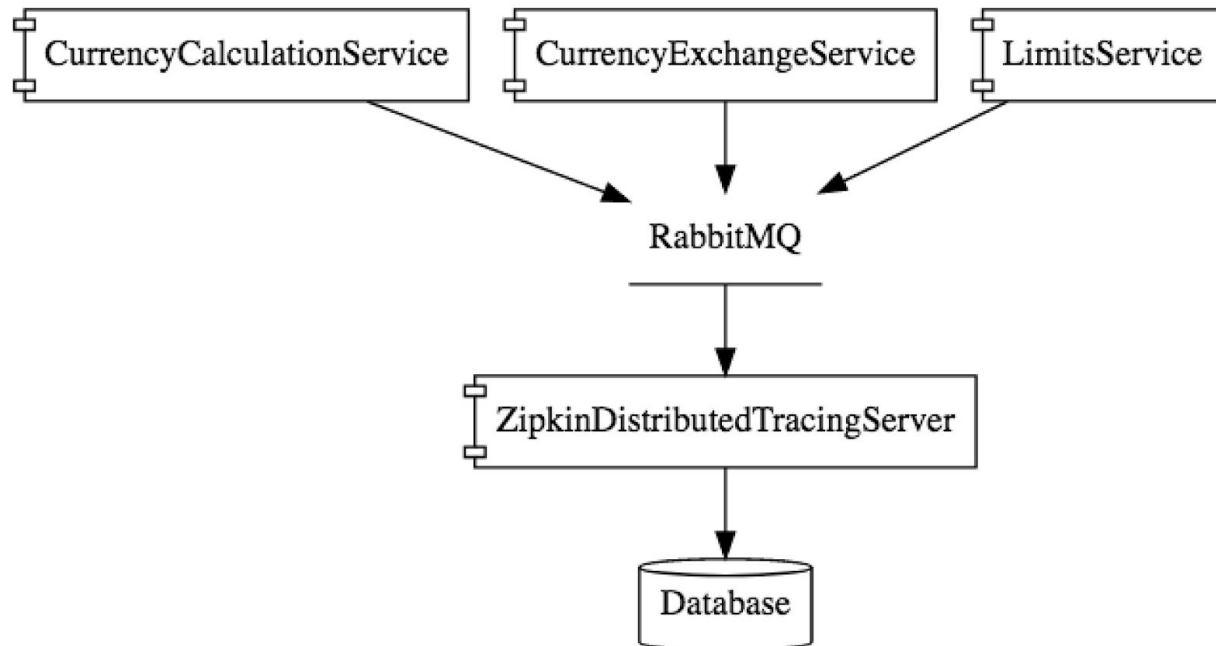
<https://github.com/anilgmaipady/spring-cloud-example/tree/master/currency-conversion-service>

<https://github.com/anilgmaipady/spring-cloud-example/tree/master/currency-exchange-service>

- <https://zipkin.io/>
- <https://www.rabbitmq.com/>

Frameworks Used

- Services send traces to rabbit MQ
- Zipkin provides trace storage and retrieval.
- Rabbit MQ/Zipkin can be replaced with other similar frameworks.



Spring Boot Admin

- Administration app for spring based micro services.
- Details - <https://github.com/codecentric/spring-boot-admin>
- Code - <https://github.com/anilgmaipady/spring-cloud-example/tree/master/netflix-eureka-naming-server>

Frameworks Used

Eureka

Eureka used for locating services for the purpose of load balancing and failover of middle-tier servers

Zuul

Zuul uses a range of different types of filters that enables us to quickly and nimbly apply functionality to our edge service. These filters help us perform the following functions:

- Authentication and Security - identifying authentication requirements for each resource and rejecting requests that do not satisfy them.

- Insights and Monitoring - tracking meaningful data and statistics at the edge in order to give us an accurate view of production.
- Dynamic Routing - dynamically routing requests to different backend clusters as needed.
- Stress Testing - gradually increasing the traffic to a cluster in order to gauge performance.
- Load Shedding - allocating capacity for each type of request and dropping requests that go over the limit.
- Static Response handling - building some responses directly at the edge instead of forwarding them to an internal cluster
- Multiregion Resiliency - routing requests across AWS regions in order to diversify our ELB usage and move our edge closer to our members

Feign

Feign is a java to http client binder inspired by [Retrofit](#), [JAXRS-2.0](#), and [WebSocket](#). Feign's first goal was reducing the complexity of binding [Denominator](#) uniformly to http apis regardless of [restfulness](#).

Ribbon

Ribbon is a client side IPC library that is battle-tested in cloud. It provides the following features

- Load balancing
- Fault tolerance
- Multiple protocol (HTTP, TCP, UDP) support in an asynchronous and reactive model
- Caching and batching

Hystrix

- Give protection from and control over latency and failure from dependencies accessed (typically over the network) via third-party client libraries.
- Stop cascading failures in a complex distributed system.
- Fail fast and rapidly recover.
- Fallback and gracefully degrade when possible.

- Enable near real-time monitoring, alerting, and operational control

Zipkin

- <https://zipkin.io/>

Prometheus

- <https://prometheus.io/>

Runing the Code

Installing Rabbit MQ

Windows

<https://www.rabbitmq.com/install-windows.html>

<https://www.rabbitmq.com/which-erlang.html>

<http://www.erlang.org/downloads>

Video - <https://www.youtube.com/watch?v=gKzKUmtOwR4>

Mac

<https://www.rabbitmq.com/install-homebrew.html>

Setting up Distributed Tracing with Zipkin

Quick Start Page:

<https://zipkin.io/pages/quickstart>

Downloading Zipkin Jar:

https://search.maven.org/remote_content?g=io.zipkin.java&a=zipkinserver&v=LATEST&c=exec

Command to run

RABBIT_URI=amqp://localhost java -jar zipkin-server-2.5.2-Exec.jar

Setup Prometheus

https://prometheus.io/docs/prometheus/latest/getting_started/

Start Services

Clone code - <https://github.com/anilgmaipady/spring-cloud-example>

Got to following sub directories and start services in the following order using the command:

mvn spring-boot:run &

Service order:

spring-cloud-config-server

Netflix-eureka-naming-server

Netflix-zuul-api-gateway-server

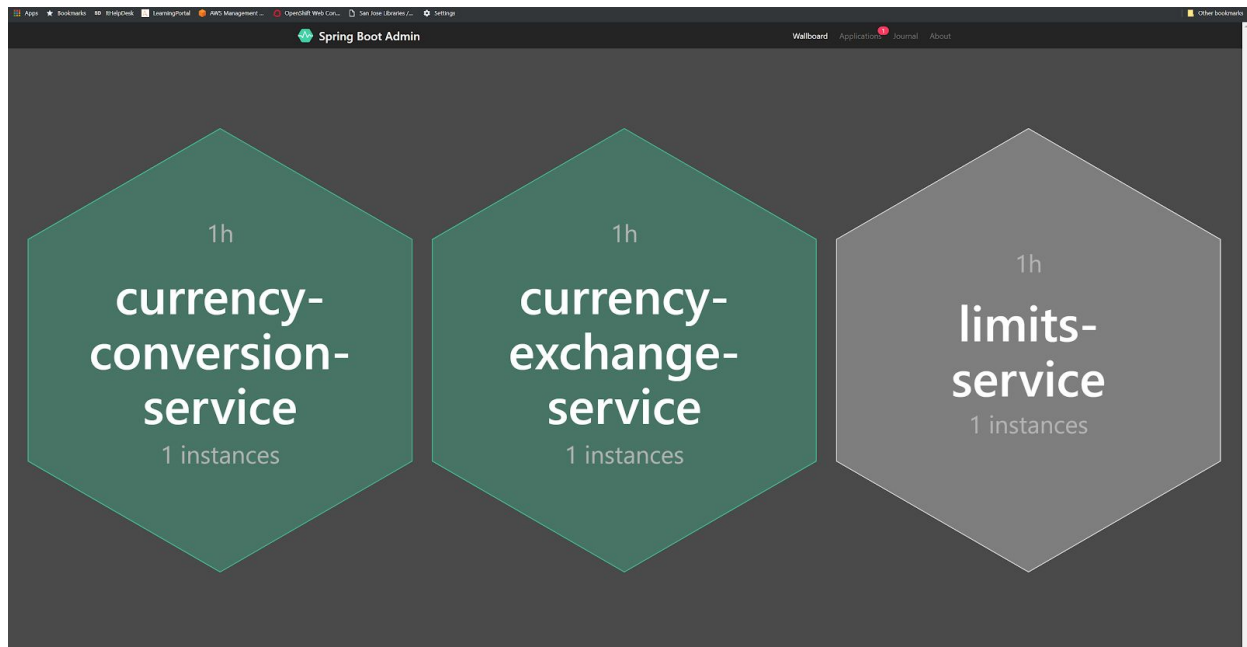
Limits-service

Currency-exchange-service

currency-conversion-service

Screen shots

Spring Boot Admin



Spring Boot Admin

WallboardApplications1JournalAbout

APPLICATIONS3

INSTANCES3

INSTANCES DOWN1

OFFLINE

limits-service

1h

http://anil-Precision-M4800:8080/

UP

currency-conversion-service

1h

http://anil-Precision-M4800:8100/

currency-exchange-service

1h

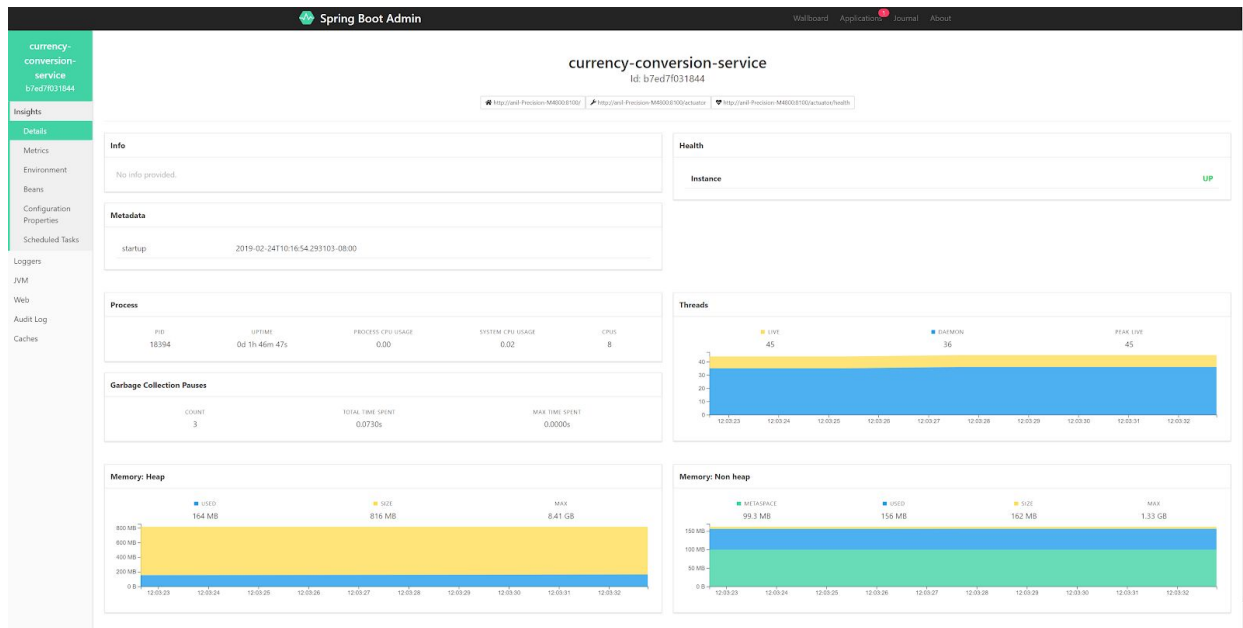
http://anil-Precision-M4800:8000/

Spring Boot Admin

WallboardApplications1JournalAbout

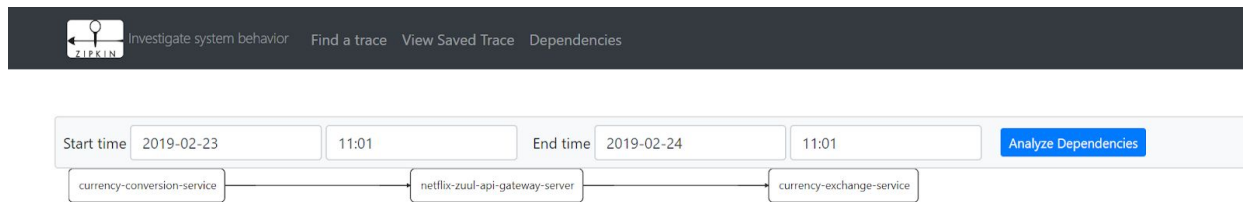
Event Journal

| Application | Instance | Time | Event |
|-----------------------------|--------------|-------------------------|-------------------------|
| limits-service | 4f393f614343 | 02/24/2019 10:32:36.001 | STATUS_CHANGED(OFFLINE) |
| currency-exchange-service | cb9d8708155a | 02/24/2019 10:17:46.611 | ENDPOINTS_DETECTED |
| currency-exchange-service | cb9d8708155a | 02/24/2019 10:17:46.600 | STATUS_CHANGED(UP) |
| currency-exchange-service | cb9d8708155a | 02/24/2019 10:17:46.179 | REGISTERED |
| currency-conversion-service | b7ed7f031844 | 02/24/2019 10:16:56.438 | STATUS_CHANGED(UP) |
| currency-conversion-service | b7ed7f031844 | 02/24/2019 10:16:55.891 | REGISTRATION_UPDATED |
| currency-conversion-service | b7ed7f031844 | 02/24/2019 10:16:46.000 | STATUS_CHANGED(OFFLINE) |
| currency-conversion-service | b7ed7f031844 | 02/24/2019 10:16:30.951 | ENDPOINTS_DETECTED |
| currency-conversion-service | b7ed7f031844 | 02/24/2019 10:16:30.932 | STATUS_CHANGED(UP) |
| currency-conversion-service | b7ed7f031844 | 02/24/2019 10:16:30.438 | REGISTERED |
| limits-service | 4f393f614343 | 02/24/2019 09:55:40.971 | ENDPOINTS_DETECTED |
| limits-service | 4f393f614343 | 02/24/2019 09:55:40.945 | STATUS_CHANGED(UP) |
| limits-service | 4f393f614343 | 02/24/2019 09:55:40.600 | REGISTERED |

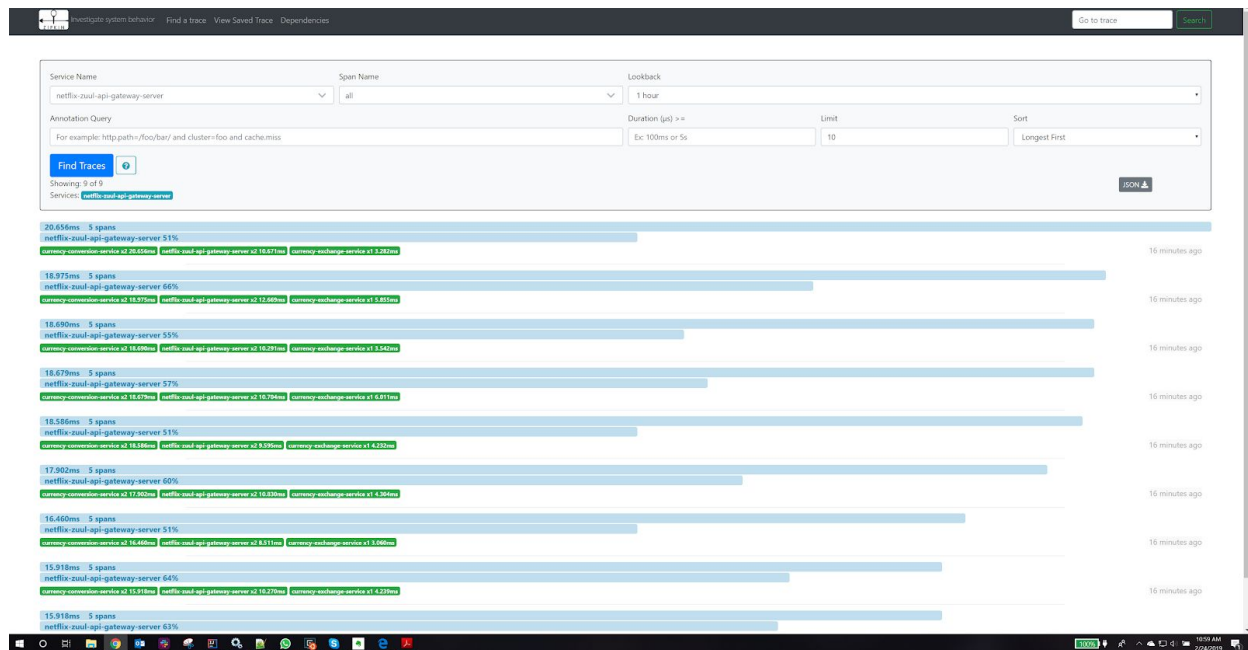


Distributed Trace

Service Dependency



Find distributed trace



Investigate system behavior Find a trace View Scaled Trace Dependencies

Go to trace Search

Duration: 20.05ms Services: 2 Depth: 4 Total Spans: 5

Expand All Collapse All

currency-conversion-service 2 currency-exchange-service 1 credit-and-api-gateway-server 2

| Service | Operation | Duration | Service | Operation | Duration | Service | Operation | Duration | Service | Operation | Duration | | | |
|-------------------------------|-----------------------|----------|-------------------------------|-----------------------|----------|-------------------------------|-----------------------|----------|-------------------------------|-----------------------|----------|-------------------------------|-----------------------|---------|
| currency-conversion-service | getCurrencyConversion | 4.17ms | currency-exchange-service | getCurrencyExchange | 0.26ms | credit-and-api-gateway-server | getCurrencyConversion | 12.35ms | currency-exchange-client | getCurrencyExchange | 16.32ms | currency-conversion-service | getCurrencyConversion | 20.05ms |
| credit-and-api-gateway-server | getCurrencyConversion | 3.53ms | credit-and-api-gateway-server | getCurrencyConversion | 3.53ms | credit-and-api-gateway-server | getCurrencyConversion | 3.53ms | credit-and-api-gateway-server | getCurrencyConversion | 3.53ms | credit-and-api-gateway-server | getCurrencyConversion | 3.53ms |

[illegible]

