Välkommen till Callista Enterprise blogg - här finns tekniska artiklar, presentationer och nyheter om arkitektur och systemutveckling. Håll dig uppdaterad genom att följa oss på Twitter.

Blog Series – Building Microservices

20 MAY 2015 // **ERIK LUPANDER • MAGNUS LARSSON**

This blog series cover various aspects of building microservices using Java and Go with supporting services from Spring Cloud, Netflix OSS and the ELK-stack (Elasticsearch, Logstash and Kibana). The series also cover how to deploy microservices in both the cloud and on premises using application platforms and infrastructures for Docker containers, e.g. container orchestration tools.

INTRODUCTION BLOG POST

An operations model for Microservices - An introduction blog post that motivates why supporting services from *Spring Cloud* and *Netflix OSS* are required in a microservices based system landscape.

LANGUAGE SPECIFIC BLOG POSTS

JAVA	GO
Part 1: Using Netflix Eureka, Ribbon and Zuul	Part 1: Introduction and rationale for Go microservices
Part 2: Trying out the circuit breaker, Netflix Hystrix	Part 2: Building our first Go microservice
Part 3: Secure API's with OAuth 2.0	Part 3: Embedding a data store and serving JSON
Part 4: Dockerize your Microservices	Part 4: Unit testing HTTP services with GoConvey
Part 5: Upgrade to Spring Cloud 1.1 & Docker for Mac	Part 5: Deploying on Docker Swarm

JAVA	GO
Part 6: Adding a Configuration Server	Part 6: Adding health checks
Part 7: Distributed tracing with Zipkin and Spring Cloud Sleuth	Part 7: Service Discovery & Load-balancing
Part 8: Centralized logging with the ELK stack	Part 8: Centralized configuration using Spring Cloud config and Viper
	Part 9: Messaging with AMQP
	Part 10: Logging to a LaaS with Logrus and Docker's log drivers
	Part 11: Circuit Breakers and resilience with Hystrix
	Part 12: Distributed tracing with Zipkin
	Part 13: Distributed persistence with CockroachDB and GORM
	Part 14: GraphQL with Go
	Part 15: Monitoring with Prometheus and Grafana

Stay tuned for more posts!

If you want to learn about new features in Docker, take a look at the blog series - Trying out new features in Docker.

Join the discussion…

suresh kumar • 4 months ago

Hi MAGNUS LARSSON
I gone through your blogs.. I got enough knowledge on Microservices.
I have small doubt how communication will happen in between services (Service A to Service B) I know Feign client, am i correct whether we can do with that or not.
Please suggest me how to do that..

2 ^ v • Reply • Share ›

piyush Raghuvanshi • 8 months ago

very very thank you....

^ v • Reply • Share ›

navono • a year ago

awesome!

^ v • Reply • Share ›

Erkan Aslanel • a year ago

Excellent work.I had question marks in my head for weeks about composite service location on architecture.The blog series solve everything.Thank you so much.

^ v • Reply • Share ›

bam bus • a year ago

Awesome stuff. Thank you a lot. I have one question about Eureka:

I have configured eureka-server and its working correctly. For the client side I have one application with two controllers. In the first controller I have 7 services and in the second i have one. The problem for me is when I set Eureka Client in my client app in Eureka dashboard I can only see the instance for the whole App.

The question is how to tell eureka to discover every service that I have in the application? I am using Spring Boot.

^ v • Reply • Share ›

Vishva ➚ bam bus • a year ago

Its because Eureka identifies your whole app as a individual microservice. having several services on the same project leads to building a monolith application. Most common microservice principle is each of every entity has its own service and data source. you simply violate the principle of microservice. For those 7 services, you should have separate microservices(projects) and then register each ow service(project) on the Eureka using @DiscoveryClient or @EurekaClient annotation.

Reply · Share ›

kundansriv · a year ago

Nice post. I real;y liked it.
I have a beginner port on MicroService here -

http://www.sharepointcafe.n...

Reply · Share ›

Beraldi · 2 years ago

Hello, these tutorials are very good, congratulations!

I would like know if the SMACK stack architecture (Spark, Mesos, Akka, Cassandra and Kafka) would be another option to netxflix oss, spring cloud and docker? Could I create an app with this case study and extend it using SMACK? How use, if possible, these technologies netflix oss, spring cloud and SMACK together? Just an overview or a link that say about.

Once again, congratulations on the series!

Thank you!

Reply · Share ›

Shawn Wong · 2 years ago

I am eagerly waiting for updates of Go series Part 8 and Part 11, our python project want use spring cloud. But there's not yet a good solution to it.

Reply · Share ›

ErikL ➚ Shawn Wong · 2 years ago

Hi! I'm currently wrapping up Part 8 which will deal with Spring Cloud Configuration. I expect to publish on monday or tuesday next week.

Glad to hear you're finding the blog useful!

Reply · Share ›

Shawn Wong ➚ ErikL · 2 years ago

thanks for reply !

Reply · Share ›

Yaya You · 2 years ago

I love you ! I love you ! I love you !

Reply · Share ›

Justin Polidori · 2 years ago

Great posts! I am following the series and I am learning a lot. Thank you.

Reply · Share ›

Marcel · 3 years ago

Great Posts! Thank you very much for this well written tutorial and the time and effort you put into this. I am looking forward to the ELK part. greetings.

Reply • Share ›

Billows • 3 years ago

Great posts!
Waiting for your next post on "ELK-stack(Elasticsearch, Logstash and Kibana)".

Reply • Share ›

Magnus Larsson ➚ Billows • 2 years ago

Hello and sorry of the loong wait :-)

Here is the blog post regarding the ELK stack: http://callistaenterprise.s..., finally :-)

Regards,
Magnus.

Reply • Share ›

Sam • 3 years ago

Good one... I used your post to design microservices.
Waiting for your next post on "data management using microservices " :)

Reply • Share ›

Sam • 3 years ago

Good one... I used your post to design microservices. thanks...
Waiting for your next post on "data management when using microservices " :)

Reply • Share ›

Girish Panchal • 4 years ago

Great Post very nicely explained (NeED HELP)

i have below basic setup

1) Eureka running on 8761

2) Zuul Running on 8080

3) One test micro service running on two system registered with Eureka (i can see two instances running on eureka dashboard)

i am getting Load balancer does not have available server for client exception

com.netflix.zuul.exception.ZuulException: Forwarding error

Can you suggest me what can be done? i was expecting Zuul will route the request in round ribon fashion but it seems something is wrong

Reply • Share ›

Magnus Larsson ➚ Girish Panchal • 4 years ago

Hello Girish and Thanks!

From your description is seems like you done everything correct and I would expect the same behaviour as describe. The only thing I can think of given your

problem description is that the Zuul server is not in contact with the Eureka server or that you misspelled the service name in your zuul config. Can you verify it by checking that also the Zuul server is registered to Eureka an that the name of the service that Zuul fails to find is correct?

Regards,
Magnus.

Reply • Share ›

Girish Panchal ➔ Magnus Larsson • 4 years ago

thank you for reply, it was my mistake i kept ribbon.eureka.enabled: false in my application.yml of Zuul,now its working as expected

continoueing my learning on this ,i have few more doubts

1) Does Zuul only have round robin fashion load balancer ?

2) i have two test services, i need to call them via zuul , do i need to create one more layer which will do client side load balancing using feign or rest template?

3) is their any way where we can have load balancer in zuul or feign other then round robin?

Reply • Share ›

Andy steward • 4 years ago

I am having two issues and I apologise for my lack of experience.|

1) I want to load the source and compile via Spring Tool Suite. I have clone the code from GIT but can't seem to import the project correctly.

2) When I run ./gradlew bootRun It goes to 80% and I do not get back the Terminal Session. Is this expected this is below my output. I am using a MAC Mini.

kaApplication : Started EurekaApplication in 5.49 seconds (JVM running for 6.043)

> Building 80% > :bootRun

Reply • Share ›

Magnus Larsson ➔ Andy steward • 4 years ago

Hello Andy!

1) I'm sorry but I don't use STS. What type or error do you get when you import the projects?

2) Yes, the 80% figure can be confusing in the beginning!

The 80% figure comes from Gradle, not from the Spring Boot application that Gradle starts.

So Gradle, the build tool, will only report 100% when the Spring Boot application stops, i.e. when the build is complete from a Gradle perspective. So that will basically not happen until you stop the Spring Boot application :-)

I hope that at least clarifies your second question!

Regards,
Magnus.

Reply · Share ›

Sam Chu · 4 years ago

great posts,thanks

Reply · Share ›

Taras Khoma · 4 years ago

how to manage entity connections in databases? each microservice has its own db. how make updates in order and customer entities at once?

Reply · Share ›

Magnus Larsson ➔ Taras Khoma · 4 years ago

Hello Taras!

That is for sure a critical question!

Given that you design your microservices in a way that different microservices handles customers and orders you have, in our mind, to rely on "eventual consistency" - patterns. For example, if you send update messages to the two different microservices via a persistent queue you are guaranteed that both updates will take place, however potentially not at the same time. You, of course, have to design your solution to handle this potential time difference...

Note: In general, we don't think patterns based on distributed transactions et al (as an alternative to eventual consistency) is a feasible way forward.

Hope it helps to some extent,
Magnus.

Reply · Share ›

Taras Khoma ➔ Magnus Larsson · 4 years ago

Thanks for quick response :) will you implement some solution of persistence for us? With rollback example and all that stuff

Reply · Share ›

Idan Fridman · 4 years ago

Hi, great posts:) Where the ELK-stack (Elasticsearch, Logstash and Kibana) post?:P

Reply · Share ›

Magnus Larsson ➔ Idan Fridman · 4 years ago

Hello Idan!