International Journal of Scientific Research and Engineering Development-– Volume 4 Issue 3, May -June 2021

Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

Possible Enhancements in Spring cloud config server

S B Rakshith*, Girish Rao Salanke N S**

*(Computer Science and Engineering, R V College of Engineering, Bengaluru, India
Email: sbrakshith.cs17@rvce.edu.in)

** (Computer Science and Engineering, R V College of Engineering, Bengaluru, India
Email:girishraosalanke@rvce.edu.in)

----------**********----------

Abstract:

This paper brings out the importance and advantages of using spring cloud config server as a service of kubernetes. The paper also brings out importance of externalising the application properties and the security it brings to the application. It also discusses various enhancments that can be done to spring cloud config server to make it more robust.

Keywords **—config server,kubernetes,springboot,autoreload.**

----------**********----------

I. INTRODUCTION

Microservices approach has greatly advanced to become an industry standard for any new development, Many organizations are supporting the same to follow and use for development. Spring cloud plays a vital role in providing excellent tools to build these microservice on top of the Spring boot framework.

Kubernetes is an orchestrator based open container system that automates numerous operations associated with the deployment, administration, and scalability of containerized applications. K8s has become a critical tool over the last several years, since the demand and because containers are increasingly being used in all stages of a continuous delivery pipeline due to their unique benefits.

Now the question that would arise is why to use kubernetes? Management of individual containers is a challenging task. Because the number of containers that a team can handle has a limit but the actual containers that are present are way more than that. This would be difficult even for a set of experienced dvelopers.K8s has facilities that would make the work simpler by giving facilities like update rollout.

A. Spring Cloud Config Server

It's always a better practice to keep the properties as safe as possible and one of the ways would be by making the properties externalised. This can be applicable on a system which is distributed. By externalising we mean that there can be single point of contact to retrieve the desired properties for any client. Therefore Config Server provides this server-client style which can be applied on the microservices to levarage safety. This can be usually used for projects written in any of the langauges available which is an advantage for the developers.It also has the capabality of handling the properties that are in different environment which is dev,test,prod etc ad it is capable of returning the propoerties based on environment which it is currrently in. Server for storing the properties usually use Bitbucket which is a version control system also known as VCS. Using VCS brings advantage like using different versions of the properties whenerver required.

B. Microservice Architecture

Microservices represent a new architectural style where small and loosely coupled modules can be

developed and deployed independently to compose an application[1].It is usually where applications has the capability to run independently hence making it loosely coupled. This can have advantages like maintaing it independently so that there is more micromanagement . This will also be having the responsibility of requesting as well as responding to the requests with other services, In this way it brings a great advantage to enterprise applications.

Once developed, these services can also be deployed independently of each other and hence it's easy to identify hot services and scale them independent of whole application. Microservices also increases fault isolation, where in the case of an error in one service the whole application doesn't necessarily stop functioning. When the error is debugged and fixed, the strain is reduced by deploying only to that particular service instead of entire application. Microservices architecture also brings benifit to the table by making it easier to choose the technology stack (programming languages, databases, etc.) which is best suited for the required functionality (service) instead of taking an approach that is standard and one-size-fits-all approach.

C. Kubernetes

Kubernetes is responsible for managing the apps that are containerized.These containerized applications are usually spread across a cluster.Cluster usually consists of server.K8s was primarily developed so that it can have the ability to manage the whole life cycle of containerized applications.K8s also provide advantages like scalability which means the ability to increase it's capability with respect to traffic,predictability, and high availability which means the downtime of the application decreases considerably.

K8s usually have a shared network,This shared network consists of physical,virtual machines that are brought together. It will also be responsible for communication that happens through shared network. This cluster serves as the primary physical platform for configuring all K8 components, capabilities, and workloads.

Coming to the architecture of K8s every part of the cluster has a role assigned.There would be one server which is usually the central processing system for the cluster.It is known to be as the master server.It is responsible of checking the status and health of the other components present in the cluster. It also decides how the work must be allocated to different components.

After master in the cluster all the other components which are present are known as nodes,These are the components that performs the work alloted to it by the master. The main reason for using K8s as a container is because of the advantages that it provides like isolation from other components making it independent,Flexibility where it gets to adapt according to the situation.

II. DESIGN

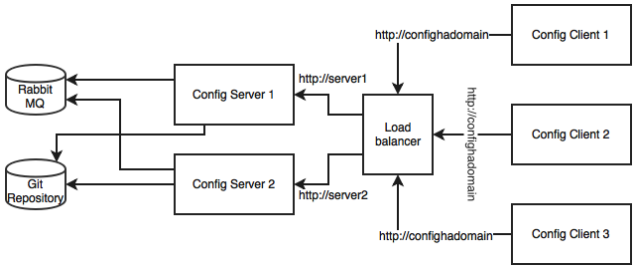

Fig. 1 Proposed design of config server architecture

Through decoupling of front-end and back-end, load capacity of the system has been raised by big percentages and good performance of the system embody the design advantage of micro-services and separation of front-end and back-end[2].The above figure is the architecture of the proposed system

Spring Cloud &NetFlix OSS is suitable for large-scale systems and experienced development teams,

and it holds a higher number of the architecture characteristics[3].The configuration properties of the application are stored on a version control system also known as VCS such as Bitbucket or some other similar kind of data source.

The usual method that everyone follows to store data is using Bitbucket,It can be made changes and given the configurations to some other form like RDBMS.

Server retrieves the application properties from the bitbucket and forwards them to the respective clients which needs the properties.

Config clients or Microservices are configured to retrieve the application properties specific to that application from the spring server but the loading happens during startup.

When the Spring Boot client application is started initially, it requests the properties from the config server in its bootstrap phase and loads them into the PropertySources in the Spring application context as key-values pairs.

These key-values pairs can be accessible via Spring's Environment interface to the other Spring components just like Spring Property Management , It can be used by using annotations like Value and ConfigurationProperties

D. Kubernetes in Action

Now we see why kubernetes plays an important role in sping cloud config server. As mentioned above kuberneteshas unique advantages like Portability and flexibility. Kubernetes works with any type of container runtime but virtually, capability in multi-cloud,Increase in developer productivity. Kubernetes provides key features like deployment, easy ways to scale, and monitoring[4].

Its important to make use of these advantages and leverage greater advantages of spring cloud config server. The main being Replication abilities of kubernetes.Replication will be used to make the application to be available.Availability will be one of the prime factor to look out for when an application is running which will be discussed in the upcoming sections

E. Replication

Every application has its own traffic and its important to cater the needs of traffic so as all the service gets the application properties on time and there is no delay since delay could impact the performance on a whole for the ecosystem.Replica coordination is provided as a service, with lightweight coupling to applications[5]. Kubernetes works out good in this strategy since it can be used to replicate as our wish.Also the auto-restart ability of pods makes it suitable to use making the server highly available

F. Load Balancer

Load Balancer by kubernetes plays another important role since it takes the responsibility of smartly dividing the requests among the replicas it has.When one server of the cluster found itself heavily

loaded, it can forward the request to other server to process

by the IP tunnelingtechnology[6]. Suppose Server1 has 80 percent workload and server2 has 40 percent it should be able to forward the request to Server2 since it has higher workload support available.

III. IMPROVING CONFIG SERVER

There are few ways in which we can automate spring cloud config and make it more efficient which will be discussing in the upcoming subsections. These are the few methods which favours on both server and client side.

G. Generating credentials automatically at client side

Expiry handling and the ability to generate credentials automatically will play an important role in spring cloud config since it would reduce manual effort of developers managing the credentials and updating it after every expiry. It is a good practice to regenerate since it would make the application more secure.Therefore config server should have the ability to autogenerate and update at the client side whenever there is an expiry upcoming.

Different microservices authorize with events of rabbitmq . Rabbit mq is usually backed by amq.This service listens for delete and failure events.The approach for listening for database authentication problems and initiating refresh is identical. The service understands and gets to know that it has to reload itself.It should also bootstrap again from Server but this will happen whenever there is a match that happens in the name .The mapping is against the RabbitMQ user but usually the latest known one. This reduces effort as it is built into the centralised system so that different Spring microservices use .Which will be time saving as well.

IV. CONCLUSIONS

As seen there is always a scope for improvement in any framework and we saw few of the possible advancements that can be done in spring cloud config server and advantages of deploying the server in kubernetes due to it's unique capability to balance the load and it's power to replicate so that it can cater the needs of any number of microservices.

ACKNOWLEDGMENT

Our sincere thanks to Dr. Ramakanth Kumar P., Professor and Head, Department of Computer Science and Engineering, RVCE for his support and encouragement.We express sincere gratitude to our beloved Principal, Dr. K. N. Subramanya for his appreciation towards this work

REFERENCES

[1] L. Abdollahi Vayghan, M. A. Saied, M. Toeroe and F. Khendek, "Deploying Microservice Based Applications with Kubernetes: Experiments and Lessons Learned," *2018 IEEE 11th International Conference on Cloud Computing (CLOUD),* 2018, pp. 970-973, doi: 10.1109/CLOUD.2018.00148.J

[2] Yifei Gong,FengGu,Kengbin Chen and Fei Wang, `The Architecture of Micro-services and the Separation of Frond-end and Back-end Applied in a Campus Information System" *2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications(AEECA).*

[3] R. ManciolaMeloca, R. Ré and A. Luis Schwerz, "An Analysis of Frameworks for Microservices*," 2018 XLIV Latin American Computer Conference (CLEI*), 2018, pp. 542-551, doi: 10.1109/CLEI.2018.00071.

[4] J. Shah and D. Dubaria, "Building Modern Clouds: Using Docker, Kubernetes & Google Cloud Platform," *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC),* 2019, pp. 0184-0189, doi: 10.1109/CCWC.2019.8666479.

[5] H. V. Netto, A. F. Luiz, M. Correia, L. de Oliveira Rech and C. P. Oliveira, "Koordinator: A Service Approach for Replicating Docker Containers in Kubernetes*," 2018 IEEE Symposium on Computers and Communications (ISCC)*, 2018, pp. 00058-00063, doi: 10.1109/ISCC.2018.8538452.

[6] Y. Jiao and W. Wang, "Design and Implementation of Load Balancing of Distributed-system-based Web Server*," 2010 Third International Symposium on Electronic Commerce and Security*, 2010, pp. 337-342, doi: 10.1109/ISECS.2010.81.

