Resolving inter-cluster reachability in Kubernetes: A Quick reference guide for PPDR Verticals from a MCX perspective

SUMMARY

The present Success Story provides the lessons learned within the 5G-EPICENTRE project on agile deployment solutions for complex PPDR verticals. The success story in hand focuses on the acceleration of MCX solutions' transition towards 5G and containerised technologies, towards virtualization-based models, for communications supported within a Kubernetes (K8) multi-cluster environment.

In undertaking the transition to virtualised, container-based solutions and adopting approaches like those outlined in the 5G-EPICENTRE project, daunting technical complications may arise. To this end, 5G-EPICENTRE aims to offer the collective experience during its lifetime and share the acquired knowledge to foster the access of future developers to these ecosystems.

ECOSYSTEM IMPACT

5G-EPICENTRE managed to successfully addresses the problems that a containerised vertical may experience when communicating with its own or third-party services, tackling the containerised inter-cluster issue.

The ecosystem of virtualised services that 5G technology seems to be fostering poses new challenges to be overcome to exploit its full potential.

Sets out accessible K8-based solutions, explaining the rationale, and links to the main sources of information. It therefore offers a quick reference guide to streamline the process of adopting microservices and containerisation and anticipates potential problems that future developers will need to face, proposing concrete step-by-step solutions.

INTER-MCX & K8 INTER-CLUSTER: DEPLOYMENT ISSUE RESOLUTION

Problem Description

The present success story provides insights and lessons learned to address effectively one of the core aspects defined by the MCX standard, interoperability between different deployments, by creating campaigns for the coordination of agencies with their own independent MCX systems. This coordination has its own specific vertical-related issues, which are further explored in 3GPP TS 33.180 V17.9.0 (2023-03).

Communication between different clusters to perform interMCX calls with campaigns presents two major technical challenges: (a) resolution of pods of external clusters, and (b) dealing with the implicit Network Address Translation (NAT) performed by K8s. As far as the first technical challenge is concerned, in K8s, pods inside the same cluster are reachable without the need to configure any name resolution. In K8s networking, to resolve pods located in other clusters, such resolution must be added in some way.

The second technical challenge is a bit more complex, but it can be simplified by saying that K8s performs NAT when a pod inside a cluster tries to communicate with anything outside the cluster, and this NAT breaks the Session Initiation Protocol (SIP) communication between agencies to correctly establish and manage calls in a campaign.

The consequence of this second issue is that floor control (controls who is speaking in MCX half-duplex communications) and media packets (the transmitted media, be it voice, video or data) cannot reach their correct destination when traversing to a remote deployment in a different cluster. The NAT imposed by K8s causes the IP and port established to be a private one not reachable from outside the cluster that owns it, and hence the communication is impossible for both floor control and media in the campaign call.

Another situation that may be troublesome due to the NAT is that a destination IP may indicate the IP of the node who hosts a pod (not of the pod itself), while the destination port has not been modified by the NAT and hence this message may reach the node host at a port that is not reachable, causing a drop in the communication.

Solution Description - Name resolution for external clusters

The K8s built-in name resolution is performed by an instance of coreDNS[1], a Domain Name System (DNS) server widely used. The pod running the coreDNS instance is configured through a ConfigMap[2], which is a K8s API object used to store non-confidential data in key-value pairs commonly allowing to decouple environment-specific configuration. In order to allow pod reachability between clusters, the coreDNS instance needs to have the corresponding entries indicating how to resolve the pods of the namespaces present on different clusters.

To achieve this, the file plugin of coreDNS has been used so as to define a DNS zone where the needed entries are defined, in our case for entries of type A and SRV (two different types of resolutions a DNS can make, DNS record type A are typical, returning an IP, SRV returns IP and port). Making use of the kubectl patch command-line tool[3], coreDNS ConfigMap can be updated to include new zone files, and such zone files can also be included in the coreDNS pod volume.

This project has received funding from the European Union's Horizon 2020 Innovation Action programme under Grant Agreement No 101016521.

CONTACT

For more information, do not hesitate to visit the website https://www.5gepicentre.eu/ and/or contact the 5G-EPICENTRE team.

Contact the 5G-EPICENTRE team by filling in the form provided. Apply here!

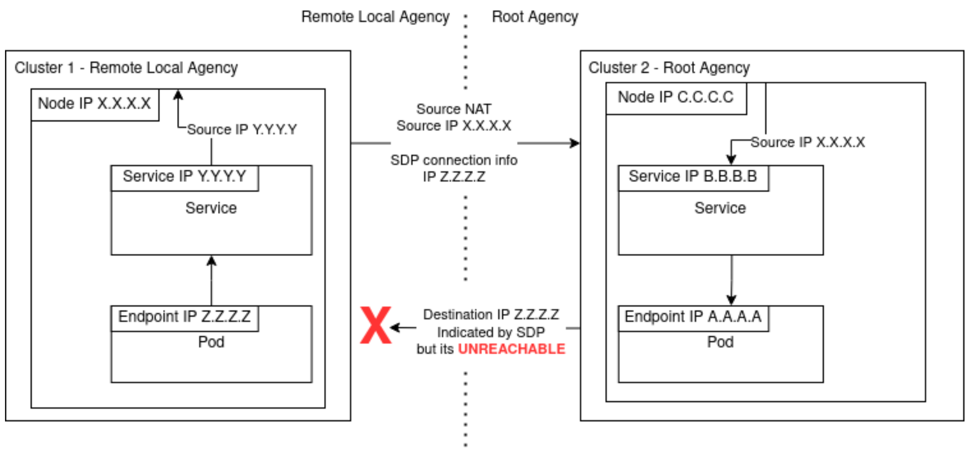
Follow Us on our social media for more NetApps trends:

MICROSERVICES OF MCX APPLICATION

Another approach for this issue has also been developed and validated. Interacting with the built-in coreDNS ConfigMap needs of administrator privileges in the cluster, something that may not be provided depending on the situation. In order to not need privileges, the deployment of a separate instance of coreDNS which forwards queries to the built-in coreDNS instance is possible, and the only needed information to be provided by the cluster administrator would be the IP of the built-in coreDNS instance. Then the pods would be configured to use the new coreDNS instance, which will have the corresponding entries for the resolution of pods in remote clusters.

One of the microservices of the MCX solution will certainly be able to manage NAT correctly since the clients are usually behind NAT and their communication will have this as an endpoint towards the MCX server. Hence, one important part of the solution consists of establishing this endpoint as the one that will receive the communication when traversing between agencies (each of them in one separate cluster).

INVITE for a campaign call started at local agency deployed at remote cluster 1
K8s implicit NAT issue for intercluster interMCS scenario



This rises some new issues, since the messages that come "NATed" will be treated as if they came from a client, and that imposes certain processing that makes the communication unsuccessful again. To overcome this obstacle, it is important to be able to differentiate traffic coming from clients and traffic from a remote agency, which can be done at SIP level. This way the communication for the K8 inter-cluster interMCX scenario follows a path that allows to manage the NAT imposed by K8s, while applying only the needed processing for this case and achieving an almost completely successful communication.

With that in place, a new issue may arise regarding ACKnowledgement (ACK) messages not reaching their destination. ACKs may be discarded at some points due to the modification of the path that was introduced to solve the NAT problem. In order to solve this, the path that the SIP messages follow in the root agency can be modified to assure the ACKs have a matching transaction and are not discarded.

Notes and References

[1] CoreDNS is available here.

[2] ConfigMap is available here.

[3] kubectl patch command-line tool is available here.

This project has received funding from the European Union's Horizon 2020 Innovation Action programme under Grant Agreement No 101016521.