Elastic Monitoring Architecture for Cloud Network Slices

André Beltrami*, Celso Cesila§, Paulo Ditarso Maciel Jr.*†,
Christian Rothenberg§ and Fábio L. Verdi*
*Federal University of São Carlos (UFSCar), Sorocaba, Brazil
Email: {beltrami,verdi}@ufscar.br
†Federal Institute of Paraíba (IFPB), João Pessoa, Brazil
Email: paulo.maciel@ifpb.edu.br
§University of Campinas (UNICAMP), Campinas, Brazil
{ccesila,chesteve}@dca.fee.unicamp.br

***Abstract*—The concept of cloud network slice features not only an end-to-end infrastructure with different types of resources (cloud, network and storage), but also the ability to provide services across multi-domain infrastructures. Therefore, an important challenge encountered is the ability of a slice to increase or decrease its resources according to SLAs/SLOs defined by a Tenant. This demo aims to present the implementation of a monitoring system architecture for cloud network slices, in which such main characteristics can be highlighted: (i) *elasticity* - as it is able to adapt as new elements are instantiated or removed in the slice; and (ii) *heterogeneity* - as it runs across multiple administrative and technological domains; which are intrinsic features of the cloud network slice concept.**

***Index Terms*—cloud network slices, monitoring, vertical elasticity, horizontal elasticity, multi-domain.**

I. Introduction

Recent technological expansion from areas such as 5G mobile networks, IoT, and smart city deployments, caused several entities and projects to study the cloud network slice concept as an enabler of these new technologies [1], [2]. There is also some effort to standardize architectures and solutions, but important operations such as resource monitoring, management and orchestration are still open issues, with several challenges to be overcome. A few examples of such challenges are: multiple technological and administrative domains, multi-tenancy, elasticity operations to grow or shrink resources, the monitoring of heterogeneous resources, among others. In this demo paper, we present an architecture for the monitoring of cloud network slices, which is capable of addressing the previously mentioned challenges.

Elasticity operations in the context of cloud network slices are slightly different when compared to (well-known) vertical and horizontal elasticities in the cloud computing environment [3], [4]. In the "cloud world", vertical elasticity means increasing/decreasing the capacity of a resource, whereas horizontal elasticity means increasing/decreasing the number of resources. On the other hand, in the context of cloud network slices, we assume that vertical elasticity means increasing/decreasing the number of resources in the same slice part (inside an infrastructure provider), and horizontal elasticity means increasing/decreasing the number of slice parts (number of providers). The Fig. 2 shows the idea of this important concept, where vertical elasticity can be seen on the left side of the figure, in which slice part 1 resources are added (scaled up) and removed (scaled down) simultaneously (in orange and purple, respectively); and horizontal elasticity can be seen on the right side, in which slice part 2 has been removed as a result of a horizontal elasticity scale in operation (in purple) and slice part 3 has been added as a result of a horizontal elasticity scale out operation (in yellow).

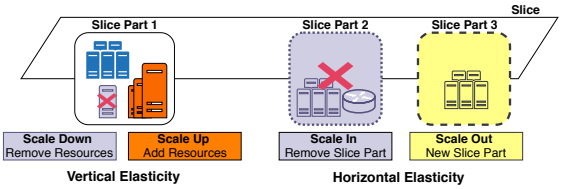

Fig. 1. Elasticity models in cloud network slicing.

Although the concept of elasticity is closely related to the orchestration of resources comprised in a slice, after performing such operations, the monitoring system must also be able to adapt to these changes, in a transparent way to the tenant. Therefore, the demonstration presented in this paper describe the implementation of a monitoring architecture for cloud network slices, capable of adapting to the elasticity operations described above. In addition, two other methods are presented in order to start and stop monitoring components. The architecture is based on adapters that aim to abstract communication with monitoring entities present in each infrastructure provider, exercising the concept of multiple administrative and technological domains.

II. Architecture

The monitoring architecture is mainly composed of five components that are briefly explained below:

- *Engine Controller*: It receives requests from an orchestrator to start, stop and update monitoring components of a cloud network slice;

978-1-7281-4973-8/20/$31.00

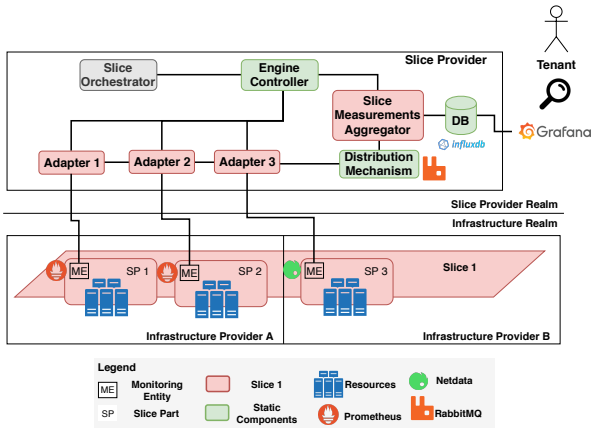

Fig. 2. Elastic Monitoring Cloud Network Slices Architecture.

- *Adapters*: It abstracts communication with different monitoring entities in order to collect metrics of the infrastructures that comprise the slice. Adapters can be developed for different monitoring entities, although in this paper we focus only on both Prometheus and Netdata[1];
- *Distribution Mechanism*: It receives and distributes monitoring metrics from the adapters to the slice measurements aggregators. For the sake of this demo, we use the RabbitMQ queue broker;
- *Slice Measurements Aggregator*: It aggregates the different collected metrics, formatting then accordingly, and adding them to the database considering an information model. Such a minimalist information model includes the following fields: KPI name, timestamp, value, and slice-related tags such as slice ID, slice part ID, resource ID, and resource type (e.g., VMs, containers, routers);
- *Database*: It stores the collected metrics for each slice in a time series database. These metrics can then be consumed directly by Tenant using a analytics platform such as Grafana or can be consumed by an Orchestrator in order to identify and trigger elasticity operations.

The components Slice Measurements Aggregator and Adapters are instantiated under Docker containers. Therefore, they take advantage of a lightweight execution when performing the start, stop and update operations, at the same time that ensure the isolation of components, which is an important concept when taking into account multiple administrative domains and tenants. Moreover, for each slice, a new set of Adapters and Slice Measurements Aggregators are instantiated.

When a horizontal elasticity operation occurs a new adapter is added or removed from the monitoring subsystem through an Orchestrator's request to the Engine Controller. This request carries information about the slice part that has been added or removed. On the other hand, when a vertical elasticity operation occurs, our monitoring solution is able to automatically identify the addition or removal of this new resource.

[1]Prometheus and Netdata are monitoring tools certified by the Cloud Native Computing Foundation.

III. Demonstration

The demo will use the Grafana dashboard in order to show three monitored metrics (CPU, memory and network traffic), from different hosts belonging to a slice. The demonstration to be presented will show the following operations incrementally performed across a time frame:

- The instantiation of monitoring components for a slice that is comprised of 4 hosts, divided into 2 slice parts and being monitored by Prometheus and Netdata;
- The *scale up* of monitoring components originated from a vertical elasticity operation, by adding 1 host in an existing slice part;
- The *scale down* of monitoring components originated from a vertical elasticity operation, by removing 1 host in an existing slice part;
- The *scale out* of monitoring components originated from a horizontal elasticity operation, by adding one slice part with 2 hosts being monitored by Prometheus;
- The *scale in* of monitoring components originated from a horizontal elasticity operation, by removing a slice part;
- The interruption of monitoring slice components.

IV. Conclusion

The concept of cloud network slicing brings a lot of challenges for both academic and industrial communities. Therefore, one of the major challenges is the ability to orchestrate different resources efficiently. For an efficient orchestration to be performed, the orchestrator must be able to query the current state of the resources that comprise the slice through a monitoring system. As stated earlier, one of the characteristics of slices is the capacity of adding and removing resources autonomously, when the Orchestrator detects any SLA violation. Therefore, the monitoring system must also be able to adapt itself when new resources appear in the slice, so as to always deliver to the Orchestrator or Tenant the metrics that are being monitored.

Acknowledgment

Work funded through the H2020 4th EU-BR Collaborative Call, under the grant agreement no. 777067 (NECOS - Novel Enablers for Cloud Slicing).

References

[1] S. Clayman, "D3.1: NECOS System Architecture and Platform Specification. V1," Tech. Rep., 10 2018. [Online]. Available: http://www.h2020-necos.eu/documents/deliverables/

[2] R. V. Rosa and C. E. Rothenberg, "The pandora of network slicing: A multicriteria analysis," *Transactions on Emerging Telecommunications Technologies*, vol. 0, no. 0, p. e3651, e3651 ett.3651. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/ett.3651

[3] A. Medeiros, A. Neto, S. Sampaio, R. Pasquini, and J. Baliosian, "End-to-end elasticity control of cloud-network slices," *Internet Technology Letters*, vol. 2, no. 4, p. e106, 2019. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/itl2.106

[4] ONF, "TR-526: Applying SDN Architecture to 5G Slicing," Tech. Rep., 2016. [Online]. Available: https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/Applying_SDN_Architecture_to_5G_Slicing_TR-526.pdf