TFS for Resilient Systems

Cloud Native Software Defined Network Project **Workshop**

April 29th, 2024

David Araújo (IT)

Objectives

This workshop serves to understand how to deploy TFS, as resiliently as possible, in a MicroK8s cluster.

We do this by:

- Understanding the TFS microservice structure.
- Understanding **Kubernetes resiliency mechanisms** that can be applied to services.
- Configuring **MicroK8s cluster**.
- Configuring **TFS' own resiliency mechanisms**.

TFS Structure

TeraFlowSDN controller runs a number of microservices on top of a Kubernetes-based environment. TFS's services can be segregated into 4 section:

1. CockroachDB: database used by the *Context* service for **long term storage**.
2. NATS: used by the *Context* service has a **message broker to handle communications between services**.
3. QuestDB: used by the *Monitoring* service, for storing **data resulting from monitoring the networks**.
4. TeraFlowSDN: the composition of **individual services** that combined constitute the controller.

Kubernetes Resiliency Mechanisms

To achieve a robust and resilient deployment of TFS, we focused primarily on:

1. Node clustering
2. Pod distribution
3. Horizontal Pod autoscaling

Node clustering

Using MicroK8s we gonna cluster **three nodes** for this workshop. If each node is a separate VM, each should match the minimum requirements:

- Ubuntu 20.04 or 22.04 LTS operating system (server or desktop)
- 4 vCPUs
- 12 GB of RAM
- 60 GB of storage

These requirement are quite considerable, but keep in mind they are **envisioned to be used in production** where there will be **multiple replicas of each pod**.

MicroK8s Clustering

TODO

Pod Distribution

Having three nodes however does **not guarantee a balance distribution of pods** between them, but we can force it.

```
topologySpreadConstraints:
  - maxSkew: 1
    topologyKey: kubernetes.io/hostname
    whenUnsatisfiable: ScheduleAnyway
    labelSelector:
      matchLabels:
        app.kubernetes.io/instance: cockroachdb
```

To achieve this, in **services with more than one replica by default** like CockroachDB and NATS, we specify `topologySpreadConstraints` and **explicitly define a skew of '1' in between hostnames** (which will be unique between nodes).