User manual

To use OSM_Autoscaler, the following steps must be followed.

Prerequisites:

1. **Include in your VNFD a scaling-group descriptor with scaling-type field set to "manual". Example of a scaling-group descriptor:**

```
min-instance-count: 0
name: vnf_autoscale
scaling-policy:
-   cooldown-time: 15
    name: cpu_util_above_threshold
    scaling-criteria:
    -   name: cpu_util_above_threshold
        scale-in-relational-operation: LT
        scale-in-threshold: 10
        scale-out-relational-operation: GT
        scale-out-threshold: 60
        vnf-monitoring-param-ref: metric_vim_vnf_cpu
    scaling-type: manual
    threshold-time: 60
vdu:
```

Figure 3 Scaling-group descriptor

The description of the scaling settings follow the OSM information model and they are explained below:

- ***max/min-instance-count***: The maximum/minimum number of scaling operations allowed. For example, if our VNF has initially 1 VDU, we can scale-out at maximum 11 additional VDUs. We selected this limit according to our OpenStack resources.
- ***cooldown-time***: The time (after a scaling operation) when no additional operations are allowed.
- ***scale-in/scale-out-threshold/relational-operation***: These settings are self-explanatory. If CPU utilization drops below 10%, a scale-in operation will be triggered. Conversely, if CPU utilization exceeds 60% a scale-out operation will be triggered.

- ***vnf-monitoring-param-ref***: The name of the monitoring parameter, corresponding to *cpu_util* metric of OpenStack's telemetry system.

- ***scaling-type***: Can be automatic or manual. We used manual to replace the OSM stock auto-scaler logic with our own prediction-based auto-scaling stack. When running experiments, we alternated this between the two values, i.e. we used 'automatic' for experiments with the stock autoscaler and we used 'manual' for experiments with our own autoscaler.

- ***threshold-time***: The minimum time that *cpu_util* metric must cross a threshold in order that a scaling operation is ordered. For example, *cpu_util* must be over 60% for at least 60 seconds for a scale-out to be triggered. Note that we used this setting to avoid unnecessary scale-outs during a VDU's start up time.

- ***(vdu) count***: The number of VDUs to commission/decommission with each scale out/in operation.

2. **Install your host machine (Linux OS recommended) with the Docker engine.**

3. **Ensure that OSM is reachable from your host system.**

2.1 Installation guide

1. Clone OSM_Autoscaler repo:

```
$ git clone https://github.com/5GinFIRE/OSM_Autoscaler.git
```

2. Build the Docker containers:

```
$ cd prediction/
prediction$ docker build -t 5ginfire_predictors -f Dockerfile .
prediction$ cd ../metrics_manager/
metrics_manager$ docker build -t 5ginfire_metric_collector -f Dockerfile .
metrics_manager$ cd ..
```

3. Initialize Docker swarm:

```
$ docker swarm init
```

4. Create an overlay network for Docker swarm:

```
$ docker network create -d overlay –scope swarm gateway
```

5. Deploy the docker stack:

```
$ docker stack deploy -c docker-compose.yaml mystack
```

6. The OSM_Autoscaler stack should now be up and running. You may now proceed to configuration as described below in section 2.2.

2.2 Configuration

The OSM_Autoscaler can be customized via environment variables. Most common use cases are provided below.

➔ The OSM_Autoscaler stack communicates with main OSM system to query performance metrics, e.g. the CPU utilization and accordingly issue scaling requests.
To point OSM_Autoscaler stack towards your OSM installation, you need to update the value of OSM_SERVER_IP variable as it is shown below (assuming that the IP address of OSM is 35.228.24.23):

```
$ docker service update --env-add OSM_SERVER_IP=35.228.24.23
mystack_metric_collector
$ docker service update --env-add OSM_SERVER_IP=35.228.24.23  mystack_predictors
```

➔ Three different prediction models are available: ARIMA, HOLT WINTERS and LSTM (RNN). The default model is ARIMA. To switch to a different prediction model during runtiime, update the value of PREDICTOR_MODEL variable as follows:

```
$ docker service update --env-add PREDICTOR_MODEL=LSTM mystack_predictors
```

➔ By default, OSM_Autoscaler evaluates the VNF system load every 120 seconds. To change the evaluation interval, update the value of DEFALUT_GRANULARITY variable as follows:

```
$ docker service update --env-add DEFAULT_GRANULARITY=60 mystack_metric_collector
```