SD-Core 1.3 Release

Ankur Upadhyaya, Badhrinath Padmanabhan, Vijaya Tiruveedula
May 31, 2023
Agenda

• Release 1.3 Overview
• SD-Core 1.3 Features
• Q & A
SD-Core 1.3 Release Overview

- SD-Core 1.3 Released on May 05, 2023
- **Highlight** – Stabilizing Cloud Native support to 5G network functions and adding Metrics support
- SD-Core 1.3 [Release blog](#)
- SD-Core 1.3 [Release Notes](#)
- Helm Charts can be found on [Charts Repository](#). Helm Charts code is tagged & branch created
- SD-Core Document [Website](#) has SDCore-1.3 branch
- SD-Core NF Code repository code is not tagged but image versions has git commit id included in them, so if required we can create branch/tag from commit hash
- Thank You all Community Members!
  - ONF, Intel, Intel Labs, Infosys, CPQD, Purdue University, GS Lab, Stanford University
  - There are more individuals contributing with individual Contributor License Agreement (CLA)
Release v1.3 Features

- Real gNodeB & UE Tests
- 5G Metric Dashboard
- NF Cache and Subscribe/Notification
- Test IMSI Support
- Static UE IP-Address allocation support
- UE Subscription Revoking Support
- Multiple gNodeBs Support in gNBSim
- Inter-op stabilisation across multiple instances of SMF and AMF
- Autoscaling of AMF/SMF instances in AIAB using KEDA
- SD-Core/Aether Documentation update
A new NF, Metric Function has been added to provision 5G Cloud Native metrics. The Metric Function acts as both metrics aggregator and processor towards metric consumers like Analytics Engine, API Server or Prometheus Server.

The 5G Metric dashboard shows whether the gNodeB and UPF are connected to the core, how many active UEs are connected, including UE’s IMSI and Slice information, and the uplink (Tx Bitrate) and downlink (Rx Bitrate) throughput at the UPF. Metrics work can be extended to show various metrics from 5G core network.
Currently, only SMF and AMF has been integrated to the Metrics-function with limited statistics. Community members can enhance to include further NFs. Following functionalities have been introduced with the Metric Function:

- API Server exposure
- Prometheus Client exposure
- Analytics Function exposure (not supported in this release)
Sample output from API Service

- GetSubscriberAll (/metric-func/v1/subscriber/all)

```
http://<metricfunc-pod-ip>:9301/metric-func/v1/subscriber/all

["imsi-208930100007487","imsi-208930100007488","imsi-208930100007489","imsi-208930100007490","imsi-208930100007491"]
```

- GetSubscriberSummary (/metric-func/v1/subscriber/<imsi>)

```

{"imsi":"imsi-208930100007487","smfId":"urn:uuid:e573621f-619b-4f67-988b-f7373e67601c","smfIp":"192.168.84.172","smfSubState":"Connected","ipAddress":"172.250.237.121","dnn":"internet","slice":"sd:010203 sst:1","upfId":"upf","upfAddr":"192.168.85.180","amfId":"b17d22c6-4809-43e6-4b6f-afa0f772807b","guti":"20893cafe00002647e6","imsi":"2508774","amfngapId":"2508774","ranngapId":"3405774848","amfSubState":"Registered","gnBid":"208:93:000102","tacId":"000001","amfIp":"192.168.84.159"}
```

- GetNfStatus (/metric-func/v1/nfstatus/<GNB/UPF>)

```

[{"nfType":"UPF","nfStatus":"Connected","nfName":"upf-1"},
{"nfType":"UPF","nfStatus":"Connected","nfName":"upf-2"}]


[{"nfType":"NGB","nfStatus":"Disconnected","nfName":"208:93:000112"},
{"nfType":"NGB","nfStatus":"Disconnected","nfName":"208:93:000102"}]
```
Static UE IP-Address Support

The SMF shall support allocation of static IP-address to UE. This feature shall be available via configuration of SUPI to IP-address mapping under SMF specific section via override values file. It is important that the operator configures the ‘static IP-Address’ from the already configured single dynamic IP-Address pool of any given DNN. The SMF shall reserve the configured ‘static IP-Addresses’ from the IP pool of that DNN and allocate it when UE connects to the 5G-Core.

Following configuration snapshot shows how the user can provision static IP-Address for certain UE

```yaml
smf:
  cfgFiles:
    smfcfg.conf:
      configuration:
        staticipinfo:
          - dnn: internet
            imsiipinfo:
              supi-123456789012341: "172.250.237.18"
              supi-123456789012342: "172.250.237.11"
```
NF Cache - NF Discovery/Response

- NRF stores information about all NF Profiles.
- As part of NF Discovery Service, Each NF discovers Target NF for its profiles from NRF and stores Target NF profiles in a local Profile cache.
- One cache is created per Target NF Type to make the lookup faster.
- Each NF profile entry stores a TTL value that is set based on the validity period received in the NF Discovery Response.
- The expired entries are periodically evicted from the cache by an eviction routine.
- Eviction interval time can be configured per NF ‘nrfCacheEvictionInterval’
- enableNrfCaching flag is set to false for AMF and SMF
The NF creates subscription for the NF profiles received from the NRF.
When a notification is received from the NRF on profile Deregistration, the entry is evicted from the local cache.
When a notification is received from the NRF on profile Changed, the entry is updated in local cache.
This will trigger the host NF to perform requested NF Discovery during the next UE procedure.
Dynamic Subscription revoking provision

- This feature enables 5G network to revoke subscription of specific UE.
- It is required that user identifies the IP-Address of specific UE to be disabled.
- The custom user application can make available the IP-Address of specific UE to 5G core network via Metric-Function API.
- The 5G network shall revoke the specific UE’s subscription based on UE IP-Addresses received.
- Adjacent diagram show one such use case of identifying malicious UE and then revoking its subscription from 5G core.

```plaintext
userAppApiServer:
addr: "userapp.omec.svc"
port: 9381
rocEndPoint:
addr: "aether-roc-umbrella-aether-roc-gui-v2-1-external.aether-roc.svc"
port: 80
```
Revoking UE Subscription from ROC

The Aether ROC control API is available via REST

- GET http://roc/targets => To Get a list of enterprises.
- POST http://roc/aether roc-api/aether/v2.1.x/aiab-enterprise/site/aiab-site/sim-card/sim-ue1
  
  --header 'Content-Type: application/json'
  --data-raw '{
  "enable": false, <= disable the sim-ue1
  "imsi": "001010102000123",
  "sim-id": "sim-ue1"
}'
Real gNBs & UE Tests

- Multiple community members mentioned success with SD-Core being used against real UE & gNodeB. To name the few
  - CPQD employed Moto 5G handset and Baicells/USRP B210 radios
  - ONF Development team employed Google Pixel/Moto 5G handsets and gNodeB from Sercomm.
  - Pronto project for DARPA employed 5G Dongle from APAL while gNodeB from Sercomm.
- We also enhanced 5G core to support the test IMSI. This support allows users to employ Test UEs having leading zeroes (mcc/mnc with leading zeroes) with SD-Core 5G. This eliminates the conflict with commercial PLMNs during UE attach.
Autoscaling of AMF/SMF instances in AIAB using KEDA

Autoscaling cloud native network functions is a critical capability for modern cloud infrastructure. It enables dynamic scaling of network functions to handle increased traffic or workload demands, ensuring optimal performance and cost-effectiveness. Kubernetes Event-driven Autoscaling (KEDA) is an open-source tool that makes it easier to implement autoscaling for cloud-native network functions(https://github.com/kedacore/keda).

When the network functions receive more traffic or workload, KEDA automatically scales up the pods to handle the increased demand. When the demand decreases, KEDA scales down the pods to save resources and minimize costs.

https://docs.sd-core.opennetworking.org/master/developer/auto-scaling-5g-nfs.html
Autoscaling of AMF/SMF instances in AIAB using KEDA

Run the following steps in aether-in-a-box folder:

- Create aiab.diff file as described below
- `patch < aiab.diff`
- Create resources/keda.yaml as described below
- Create resources/5g-monitoring/smf-monitor.yaml as described below
- Create autoscale.yaml as described below
- `make 5g-core`
- `make monitoring-5g`
- `make autoscale-aiab`
- `kubectl get hpa -n omec`: To view the horizontal pod scaler.
- `kubectl get pods -n omec | grep smf`: To view the scaled pods.
Autoscaling of AMF/SMF instances in AIAB using KEDA

Create file aiabdiff with following content

diff --git a/Makefile b/Makefile
index bd4a7a..d9b8e0 100644
--- a/Makefile
+++ b/Makefile
@@ -26,9 +26,10 @@ GET_Helm
 get_helm.sh
KUBEDEPLOY_VERSION := release-2.17
DOCKER_VERSION := v0.10.0
-HELP_VERSION := v0.13.8
-HELPCONFIG_VERSION := v1.23.10
-HELPCONFIG_VERSION := v1.24.11
+HELPDEPLOY_VERSION := v1.23.15+rk82r1
+HELPCONFIG_VERSION := v1.24.11+rk82r1
+KUBEDEPLOY_VERSION := v1.23.15+rk82r1
+KUBEDEPLOY_VERSION := v1.21.6

OAISIM_API_IMAGE := angyavd/aiat lcm:snap@1.1.0-$@shell uname -rf
+ROUTER_ROUTER_NETCONF := /etc/systemd/system/3r-slab-access.netdev
+SYSTEMD_NETCONF := /etc/systemd/system/aliab-ui-net.service

# monitoring
+AUTOSCALE_CHART := keda-keda
+AUTOSCALE_VALUES := $(KUBEDEPLOY) autoscale.yaml
+RANCHER_MONITORING_CRD := rancher/rancher-monitoring-crd
+RANCHER_MONITORING_CHART := rancher/rancher-monitoring
+MONITORING_VALUES := $(KUBEDEPLOY) monitoring.yaml

@@ -575,6 +576,26 @@ test: 4g-core $MY/autoscaler
    |grep -q "Simulation Result: PASS/"Profile Status: PASS" /tmp/ghbhs.out

+autoscaler: $MY/autoscaler
+KUBEDEPLOY: $KUBEDEPLOY
+help ready
+help repo add keda https://kedacore.github.io/charts
+help upgrade --install --wait $HELP_GLOBAL_MANG
+--namespace=autoscale
+--create-namespace
+--value=KUBEDEPLOY\AUTOSCALE_VALUES\keda-slab\$(AUTOSCALE_CHART)\touch $MY/autoscaler
+autoscale: $MY/autoscaler
+kubectl apply -f resources/keda.yaml
+
+autoscale: $MY/autoscaler
+autoscale: $MY/autoscaler
+kubectl delete -f resources/keda.yaml
+kubectl delete -f resources/keda.yaml
+kubectl delete -f resources/keda.yaml
+git $(MY); rm -f $MY autoscaler

Create file resources/keda.yaml with the following content

apiVersion: keda.sh/v1alpha1
kind: ScaleTo
metadata:
  name: keda-scale
spec:
  targetRef:
    kind: Deployment
  resourceRef:
    name: ser
  minReplicas: 1
  maxReplicas: 3
  cooldownPeriod: 30
  triggers:
    - type: prometheus
      metricSelector:
        measure: ni_messages_total
      threshold: 0.7

Create file resources/3g-monitoring/3g-monitoring.yaml with the following content

apiVersion: monitoring.coreos.com/v1
kind: ServiceMonitor
metadata:
  name: ser
spec:
  namespace: ser
  endpoints:
    - path: metrics
      port: prometheus-exporter
      intervals:
        - 80
      metricsPath: /metrics
      interval: 1h
  selector:
    matchLabels:
      app: ser
Add an empty autoscale.yaml in aiab folder. This file can be used to add override values for keda helm chart.

touch autoscaler.yaml
Other Enhancements

Multiple gNBs Support in gNBSim
This enhancement allows deploying multiple gNBs with a single gNBSim instance. This would extend gNBSim capabilities towards XN and N2 handover procedures.

Inter-op stabilisation across multiple instances of SMF and AMF
Enhancements have been done around stabilisation of multiple instances of SMF and AMF inter-operation. This adds robustness towards cloud native stateless and high-availability feature. SMF and AMF scaling can be either be done automatically as mentioned before or done manually by modifying SD-Core helm chart deployment configuration.
Multiple gNB Support in gNBSim

- Support of Multiple gNBs: Two gNBs are configured by default. So User can create profiles by using these gNBs. Configuration of two gNBs can be found here.
Documentation Improvements

- 5G Deployment Guide
- Static IP-Address Support
- Test IMSI Support
- Metric Function Design overview
- Dynamic Subscription Revoking
- Auto-Scaling 5G NFs
- Release Notes for 1.3
How to contribute

• Queries can be sent on slack channel
  – Channels #sdcore-dev #aether-dev

• Identifying value add features and send the proposal for review

• Aether Users
  – Let us know documentation issues
  – Raise pull request on code, document, helm charts

• Developer
  – Raise pull request for code changes

• Quality Assurance
  – Test our released code and let us know issues found

• Join our Aether Community calls.
  – TST meetings | Every Tuesday, 9:00 AM Pacific Time (US and Canada)

• Previous meeting recordings available here
Thank You!