Open Networking Foundation

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

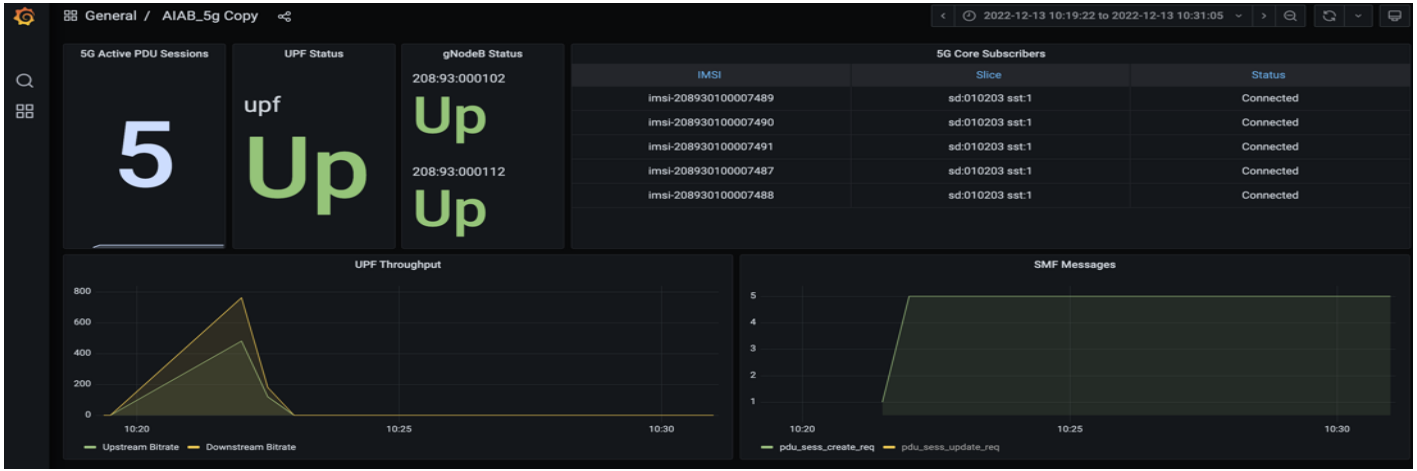
SD-Core 1.3 Release Overview

- 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

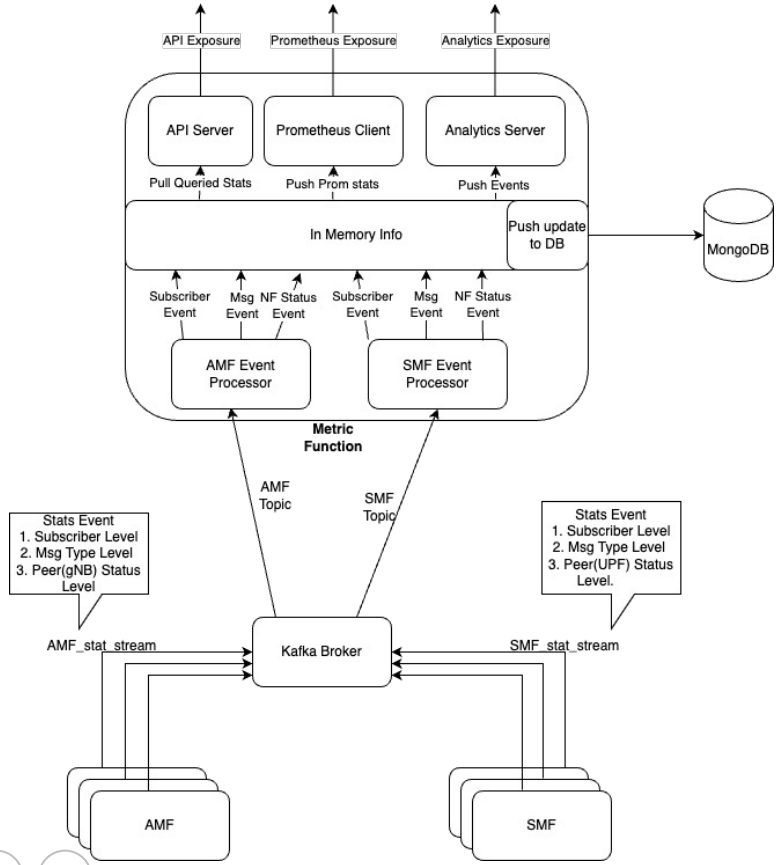
- Real gNodeB & UE Tests
- 5G Metric Dashboard
- NF Cache and Subscribe/Notification
- Test IMSI Support
- Static UE IP-Address allocation support
- UE Subscription Revoking Support
- Multiple gNodeBs Support in gNBSim
- Inter-op stabilisation across multiple instances of SMF and AMF
- Autoscaling of AMF/SMF instances in AIAB using KEDA
- SD-Core/Aether Documentation update

5G Metric Dashboard



- 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.

5G Metrics Function



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)

5G Metrics APIs

Sample output from API Service

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

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

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

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

```
http://<metricfunc-pod-ip>:9301/nmetric-func/v1/subscriber/imsi-208930100007487
{
    "imsi":"imsi-208930100007487",
    "smfId":"urn:uuid:c573621f-e198-4f67-988b-f7373e67601c","smfIp":"192.168.84.172",
    "smfSubState":"Connected","ipaddress":"172.250.237.121","dnn":"internet","slice":"sd:010203 sst:1",
    "upfid":"upf","upfAddr":"192.168.85.188",
    "amfId":"b17f4726-4809-43e6-b5b6-afa0fc72807b","guti":"20893cafe00002647e6","tmsi":2508774,"amfngapId":2508775,
    "ranngapId":3405774848,"amfSubState":"Registered","gnbid":"208:93:000102","tacid":"000001","amfIp":"192.168.84.159"
}
```

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

```
http://<metricfunc-pod-ip>:9301/nmetric-func/v1/nfstatus/UPF
[
    {"nfType":"UPF","nfStatus":"Connected","nfName":"upf-1"},
    {"nfType":"UPF","nfStatus":"Connected","nfName":"upf-2"}
]

http://<metricfunc-pod-ip>:9301/nmetric-func/v1/nfstatus/GNB
[
    {"nfType":"GNB","nfStatus":"Disconnected","nfName":"208:93:000112"},
    {"nfType":"GNB","nfStatus":"Disconnected","nfName":"208:93:000102"}
]
```

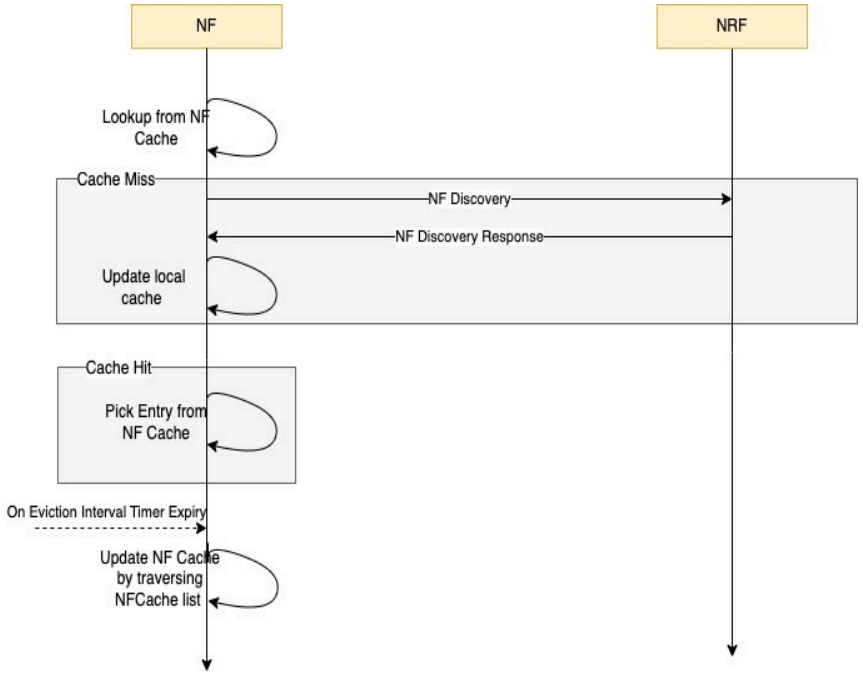
Static UE IP-Address Support

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

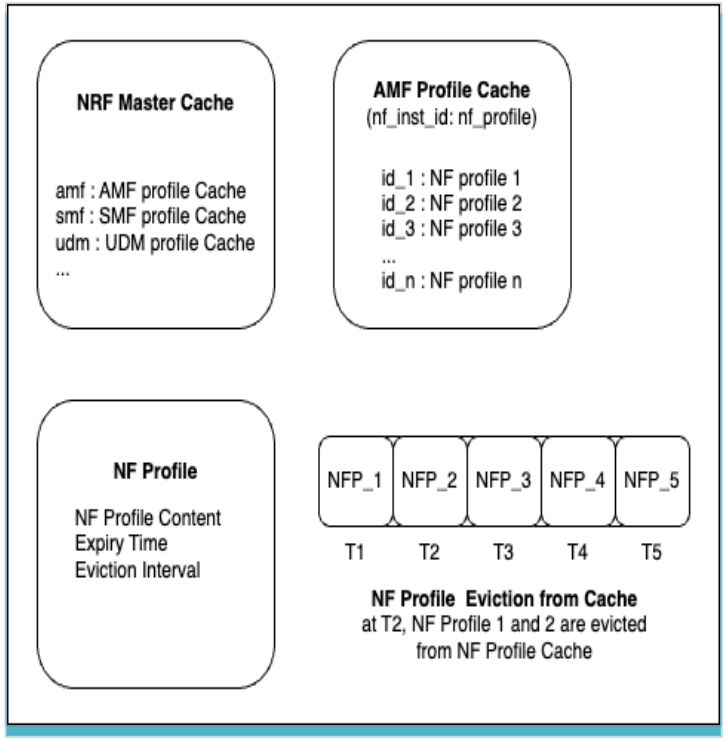
```
smf:
  cfgFiles:
    smfcfg.conf:
      configuration:
        staticIpInfo:
        - dnn: internet
          imsiIpInfo:
            supi-123456789012341: "172.250.237.10"
            supi-123456789012342: "172.250.237.11"
```

NF Cache - NF Discovery/Response



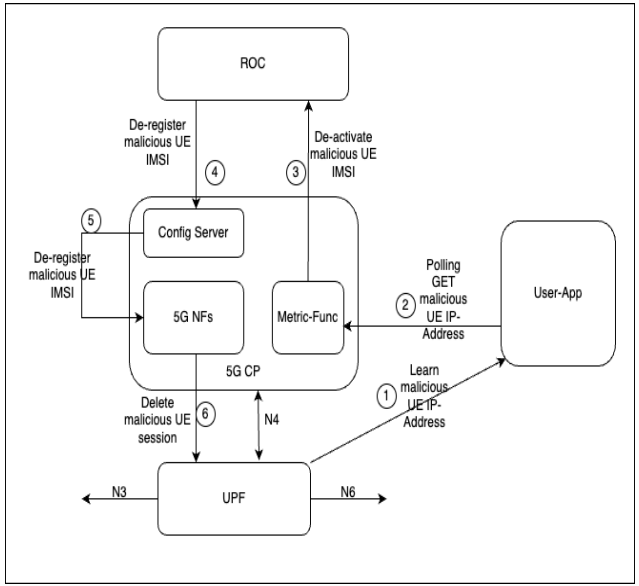
- NRF stores information about all NF Profiles.
- As part of NF Discovery Service, Each NF discovers Target NF for its profiles from NRF and stores Target NF profiles in a local Profile cache.
- One cache is created per Target NF Type to make the lookup faster.
- Each NF profile entry stores a TTL value 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

NF Cache and Subscribe/Notification



- 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



The Metric-Function configuration to poll the user application

```
userAppApiServer:
  addr: "userapp.omec.svc"
  port: 9301
rocEndPoint:
  addr: "aether-roc-umbrella-aether-roc-gui-v2-1-external.aether-roc.svc"
  port: 80
```

- 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.

Revoking UE Subscription from ROC

The Aether ROC control API is available via REST

- *GET http://roc/targets* => To Get a list of enterprises.
- *GET http://roc/aether-roc-api/aether/v2.1.x/**aiab-enterprise**/site*. Get the list of sites in the 'aiab-enterprise' enterprise.
- *GET http://roc/aether-roc-api/aether/v2.1.x/ **aiab-enterprise** /site/**aiab-site**/sim-card*. Get the list of sim-card in the 'aiab-site' site and 'aiab-enterprise' enterprise.
- *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 aiab.diff with following content

```
diff --git a/Makefile b/Makefile
index bd54a7a..df85e0a 100644
--- a/Makefile
+++ b/Makefile
@@ -26,9 +26,10 @@ GET_HELM             = get_helm.sh
 KUBESPRAY_VERSION ?= release-2.17
 DOCKER_VERSION    ?= '20.10'
 HELM_VERSION      ?= v3.10.3
-KUBECTL_VERSION   ?= v1.23.15
+KUBECTL_VERSION   ?= v1.24.11

-RKE2_K8S_VERSION  ?= v1.23.15+rke2r1
+RKE2_K8S_VERSION  ?= v1.24.11+rke2r1
+#RKE2_K8S_VERSION  ?= v1.23.15+rke2r1
 K8S_VERSION       ?= v1.21.6

 OAISIM_UE_IMAGE ?= andybavier/lte-uesoftmodem:1.1.0-$(shell uname -r)
@@ -65,6 +66,8 @@ ROUTER_HOST_NETCONF   := /etc/systemd/network/10-aiab-access.netdev /etc/systemd
 UE_NAT_CONF           := /etc/systemd/system/aiab-ue-nat.service

 # monitoring
+AUTOSCALE_CHART             := kedacore/keda
+AUTOSCALE_VALUES            ?= $(MAKEDIR)/autoscale.yaml
 RANCHER_MONITORING_CRD_CHART := rancher/rancher-monitoring-crd
 RANCHER_MONITORING_CHART     := rancher/rancher-monitoring
 MONITORING_VALUES            ?= $(MAKEDIR)/monitoring.yaml
@@ -675,6 +678,26 @@ test: | 4g-core $(M)/oaisim
     fi
     @grep -q "Simulation Result: PASS\|Profile Status: PASS" /tmp/gnbsim.out

+autoscale: $(M)/autoscale
+$(M)/autoscale: $(M)/helm-ready
+   helm repo add kedacore https://kedacore.github.io/charts
+   helm upgrade --install --wait $(HELM_GLOBAL_ARGS) \
+    --namespace=autoscale \
+    --create-namespace \
+    --values=$(AUTOSCALE_VALUES) \
+    keda-aiab \
+    $(AUTOSCALE_CHART)
+   touch $(M)/autoscale
+
+autoscale-aiab: $(M)/autoscale
+   kubectl apply -f resources/keda.yaml
+
+autoscale-clean:
+   kubectl delete -f resources/keda.yaml
+   helm -n autoscale delete keda-aiab || true
+   kubectl delete namespace autoscale || true
+   rm $(M)/autoscale
+
 reset-test: | oaisim-clean omec-clean router-clean
    @cd $(M); rm -f omec oaisim 5g-core
```

```
diff --git a/resources/5g-monitoring/kustomization.yaml b/resources/5g-monitoring/kustomization.yaml
index 96bc72b..0b757e9 100644
--- a/resources/5g-monitoring/kustomization.yaml
+++ b/resources/5g-monitoring/kustomization.yaml
@@ -5,6 +5,7 @@
 resources:
   - ./metricfunc-monitor.yaml
   - ./upf-monitor.yaml
+  - ./smf-monitor.yaml

 configMapGenerator:
   - name: grafana-ops-dashboards
```

Create a file resources/keda.yaml with the following content

```
---
apiVersion: keda.sh/v1alpha1
kind: ScaledObject
metadata:
  name: smf-scale
  namespace: omec
spec:
  scaleTargetRef:
     kind: Deployment
     name: smf
  minReplicaCount: 1
  maxReplicaCount: 5
  cooldownPeriod: 30
  pollingInterval: 1
  triggers:
  - type: prometheus
    metadata:
      serverAddress: http://rancher-monitoring-prometheus.cattle-monitoring-system.svc:9090
      metricName: n4_messages_total
      query: |
        sum(n4_messages_total{job="smf"})
      threshold: "50"
```

Create file resources/5g-monitoring/smf-monitor.yaml with following content

```
apiVersion: monitoring.coreos.com/v1
kind: ServiceMonitor
metadata:
  name: smf
  namespace: omec
spec:
  endpoints:
    - path: /metrics
      port: prometheus-exporter
  namespaceSelector:
    matchNames:
      - omec
  selector:
    matchLabels:
      app: smf
```

Add an empty autoscale.yaml in aiab folder. This file can be used to add override values for keda helm chart.

```
touch autoscale.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

```
gnb:
  ips:
  - '"192.168.251.5/24"' #gnb1 IP
  - '"192.168.251.6/32"' #gnb2 IP
configuration:
  runConfigProfilesAtStart: true
  singleInterface: #this will be added thorugh configmap script
  execInParallel: false #run all profiles in parallel
  gnbs: # pool of gNodeBs
    gnb1:
      n2IpAddr: # gNB N2 interface IP address used to connect to AMF
      n2Port: 9487 # gNB N2 Port used to connect to AMF
      n3IpAddr: 192.168.251.5 # gNB N3 interface IP address used to connect to UPF
      n3Port: 2152 # gNB N3 Port used to connect to UPF
      name: gnb1 # gNB name that uniquely identify a gNB within application
      globalRanId:
        plmnId:
          mcc: 208 # Mobile Country Code (3 digits string, digit: 0~9)
          mnc: 93 # Mobile Network Code (2 or 3 digits string, digit: 0~9)
        gNbId:
          bitLength: 24
          gNBValue: "000102" # gNB identifier (3 bytes hex string, range: 000000~FFFFFF)
      supportedTaList:
      - tac: "000001" # Tracking Area Code (3 bytes hex string, range: 000000~FFFFFF)
        broadcastPlmnList:
          - plmnId:
              mcc: 208
              mnc: 93
            taiSliceSupportList:
                - sst: 1 # Slice/Service Type (uinteger, range: 0~255)
                  sd: "010203" # Slice Differentiator (3 bytes hex string, range: 000000~FFFFFF)
      defaultAmf:
        hostName: amf # Host name of AMF
        ipAddr: # AMF IP address
        port: 38412 # AMF port
    gnb2:
      n2IpAddr: # gNB N2 interface IP address used to connect to AMF
      n2Port: 9488 # gNB N2 Port used to connect to AMF
      n3IpAddr: 192.168.251.6 # gNB N3 interface IP address used to connect to UPF
      n3Port: 2152 # gNB N3 Port used to connect to UPF
      name: gnb2 # gNB name that uniquely identify a gNB within application
      globalRanId:
        plmnId:
          mcc: 208 # Mobile Country Code (3 digits string, digit: 0~9)
          mnc: 93 # Mobile Network Code (2 or 3 digits string, digit: 0~9)
        gNbId:
          bitLength: 24
          gNBValue: "000112" # gNB identifier (3 bytes hex string, range: 000000~FFFFFF)
      supportedTaList:
      - tac: "000001" # Tracking Area Code (3 bytes hex string, range: 000000~FFFFFF)
        broadcastPlmnList:
          - plmnId:
              mcc: 208
              mnc: 93
            taiSliceSupportList:
                - sst: 1 # Slice/Service Type (uinteger, range: 0~255)
                  sd: "010203" # Slice Differentiator (3 bytes hex string, range: 000000~FFFFFF)
      defaultAmf:
        hostName: amf # Host name of AMF
        ipAddr: # AMF IP address
        port: 38412 # AMF port
```

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
  - Workspace - https://onf-community.slack.com/
  - 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!