Diep duiken in SCP Model-D gebaseerde communicatie

Inhoud

Inleiding

Dit document beschrijft de diepe duik van SCP Model-D Communication-aanpak tussen Cisco AMF/SMF en NF van derden.

Voorwaarden

Vereisten

Cisco raadt kennis van de volgende onderwerpen aan:

- Functionaliteit van de Access and Mobility Management Function (AMF)
- Functionaliteit van de Sessiebeheerfunctie (SMF)
- Functionaliteit van Service Communication Proxy (SCP)

Gebruikte componenten

Dit document is niet beperkt tot specifieke software- en hardware-versies.

De informatie in dit document is gebaseerd op de apparaten in een specifieke laboratoriumomgeving. Alle apparaten die in dit document worden beschreven, hadden een opgeschoonde (standaard)configuratie. Als uw netwerk live is, moet u zorgen dat u de potentiële impact van elke opdracht begrijpt.

Achtergrondinformatie

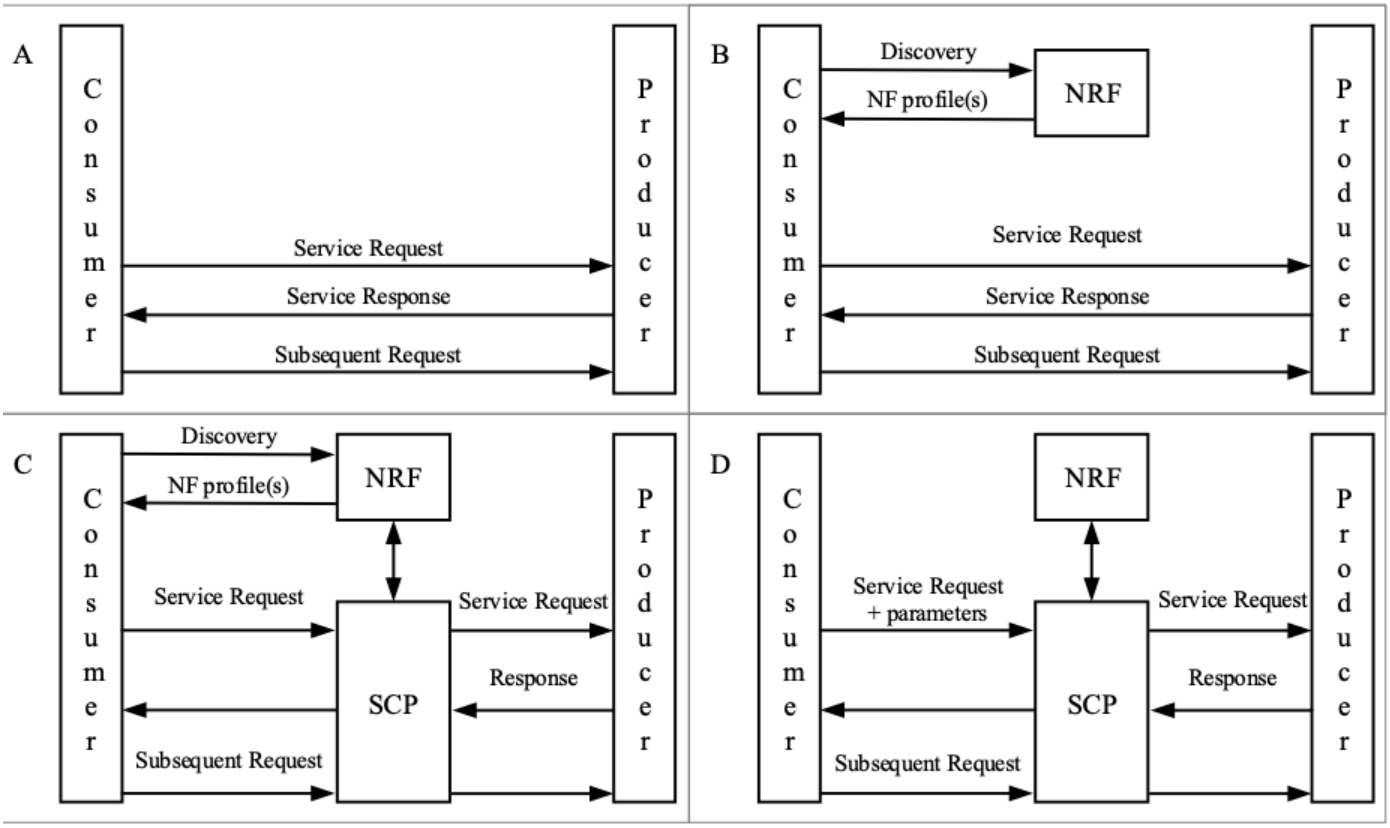
Operators over de hele wereld kunnen kiezen tussen meerdere communicatiemodellen met

behulp van SCP voor de netwerkfunctie (NF) detectie en daaropvolgende NF naar NF communicatie. Dit onderwerp behandelt concepten rond verschillende communicatiemodellen en de veranderingen in de gespreksstroom / configuratie die vereist zijn bij Subscriber Microservices Infrastructure (SMI), AMF / SMF om SCP Model-D-gebaseerde communicatie te hebben.

Overzicht van architectuur en oplossing

In de Service-based architecture (SBA) fungeert het SCP als tussenpersoon en faciliteert het indirecte communicatie tussen NF's door routering, load balancing en service discovery af te handelen, waardoor uiteindelijk de op service gebaseerde architectuur wordt gestroomlijnd.

3GPP 23.501 Annex-E beschrijft de vier communicatiemodellen tussen NF in een 5GC-implementatie.



Figuur A: (Verschillende communicatiemodellen met betrekking tot SCP)

Model-A - Directe communicatie zonder Network Repository Function (NRF) interactie: Consumenten worden geconfigureerd met de 'NF-profielen' van de producenten en communiceren rechtstreeks met een producent van hun keuze. Dit is een soort statische selectie en er wordt geen NRF of SCP gebruikt.

Model-B - Directe communicatie met NRF-interactie: Consumenten doen ontdekking door de NRF te bevragen. Op basis van het ontdekkingsresultaat doet de consument de selectie. De consument stuurt het verzoek naar de geselecteerde producent.

Model-C - Indirecte communicatie zonder gedelegeerde ontdekking: Consumenten ontdekken door de NRF te bevragen. Op basis van ontdekkingsresultaten doet de consument de selectie van

een NF-set of een specifieke NF-instantie van NF-set. De consument stuurt het verzoek naar het SCP met het adres van de gekozen serviceproducent die verwijst naar een NF-serviceinstantie of een reeks NF-serviceinstanties. In het laatste geval kiest het SCP een NF Service-instantie. Indien mogelijk werkt het SCP samen met NRF om selectieparameters zoals locatie, capaciteit, enzovoort te krijgen. Het SCP stuurt het verzoek naar de gekozen NF-serviceproducent.

Model-D - Indirecte communicatie met gedelegeerde detectie: consumenten zijn niet betrokken bij ontdekking of selectie. De consument voegt alle nodige detectie- en selectieparameters die nodig zijn om een geschikte producent te vinden toe aan het serviceverzoek. Het SCP gebruikt het aanvraagadres en de detectie- en selectieparameters in het aanvraagbericht om het verzoek naar een geschikte instantie van de producent te routeren. Het SCP kan een detectie uitvoeren met een NRF en een ontdekkingsresultaat verkrijgen.

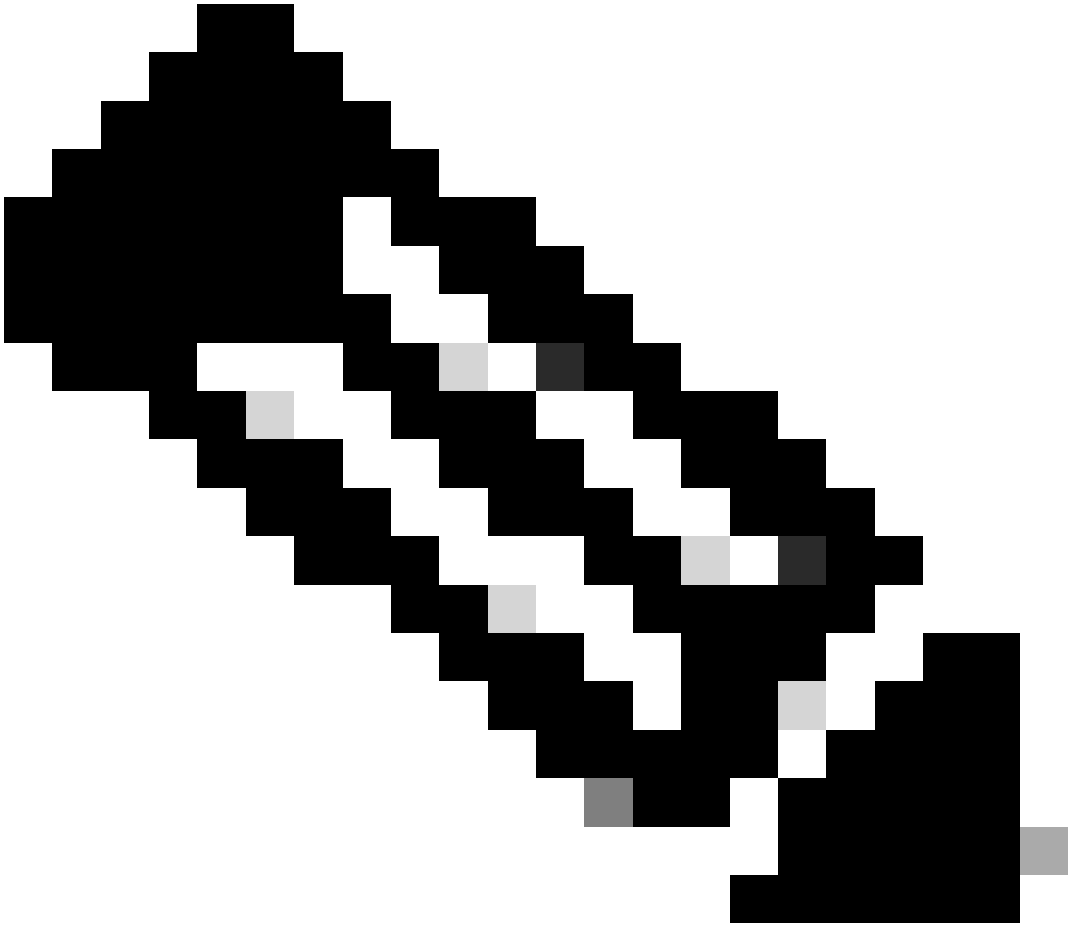
Diepe duik op Model-D gebaseerde communicatie: Wanneer Call Model-D wordt gebruikt, stuurt de NF-consument niet rechtstreeks een verzoek naar de NRF, maar delegeert deze ontdekking aan het SCP. De NF-client stuurt een bericht naar het SCP en voegt voor elk van deze detectiefactoren de tekenreeks '3gpp-sbi-discovery' samen met de naam van de detectiefactor die wordt gebruikt als NF-detectie via de NRF wordt uitgevoerd.

Voor een scenario waarin SMF op zoek is naar Unified Data Management (UDM) met servicenamen nudm-sdm, worden de detectiefactoren doorgegeven aan het SCP:

- Autoriteit Header: de autoriteit draagt ofwel de volledig gekwalificeerde domeinnaam (FQDN) of het IP-adres, waarbij prioriteit wordt gegeven aan de IP-adresconfiguratie.
- 3gpp-sbi-discovery-requester-nf-type: SMF
- 3gpp-sbi-discovery-target-nf-type: UDM
- 3gpp-Sbi-discovery-service-naam: nudm-sdm

```
> Header: :authority: [redacted]
> Header: :method: PUT
> Header: :path: /nudm-uecm/v1/imsi-[redacted]/registrations/smf-registrations/2
> Header: :scheme: http
> Header: 3gpp-sbi-discovery-requester-nf-type: SMF
> Header: 3gpp-sbi-discovery-target-plmn-list: [{"mcc":[redacted],"mnc":[redacted]]
> Header: 3gpp-sbi-discovery-supi: imsi-[redacted]
> Header: content-type: application/json
> Header: user-agent: SMF-[redacted]
> Header: 3gpp-sbi-discovery-target-nf-type: UDM
> Header: content-length: 239
> Header: accept-encoding: gzip
  [Full request URI: [redacted]/nudm-uecm/v1/imsi-[redacted]/registrations/smf-reg
  [Response in frame: 40]
```

Figuur B: (SMF-UDM-communicatie via SCP-model D)



Opmerking: De 3gpp-sbi-discovery-service naam formaat is in plain string formaat en niet in array formaat volgens 3gpp 29.510 en open API definities (4.7.12.4 Style). In de 29.510 3gpp-sbi-discovery-service-naam wordt genoemd als een array formaat.

```
- name: service-names
  in: query
  description: Names of the services offered by the NF
  schema:
    type: array
    items:
      $ref: 'TS29510_Nnrf_NFManagement.yaml#/components/schemas/ServiceName'
    minItems: 1
    uniqueItems: true
  style: form
  explode: false
```

Afbeelding C: (Snapshot van 29.510 Spec)

De stijl: form en explode: false converteert de array echter naar een plain string, wat wordt

uitgelegd door een voorbeeld te nemen van OpenAPI.

Assume a parameter named `color` has one of the following values:

```
string -> "blue"
array -> ["blue","black","brown"]
object -> { "R": 100, "G": 200, "B": 150 }
```

The following table shows examples of rendering differences for each value.

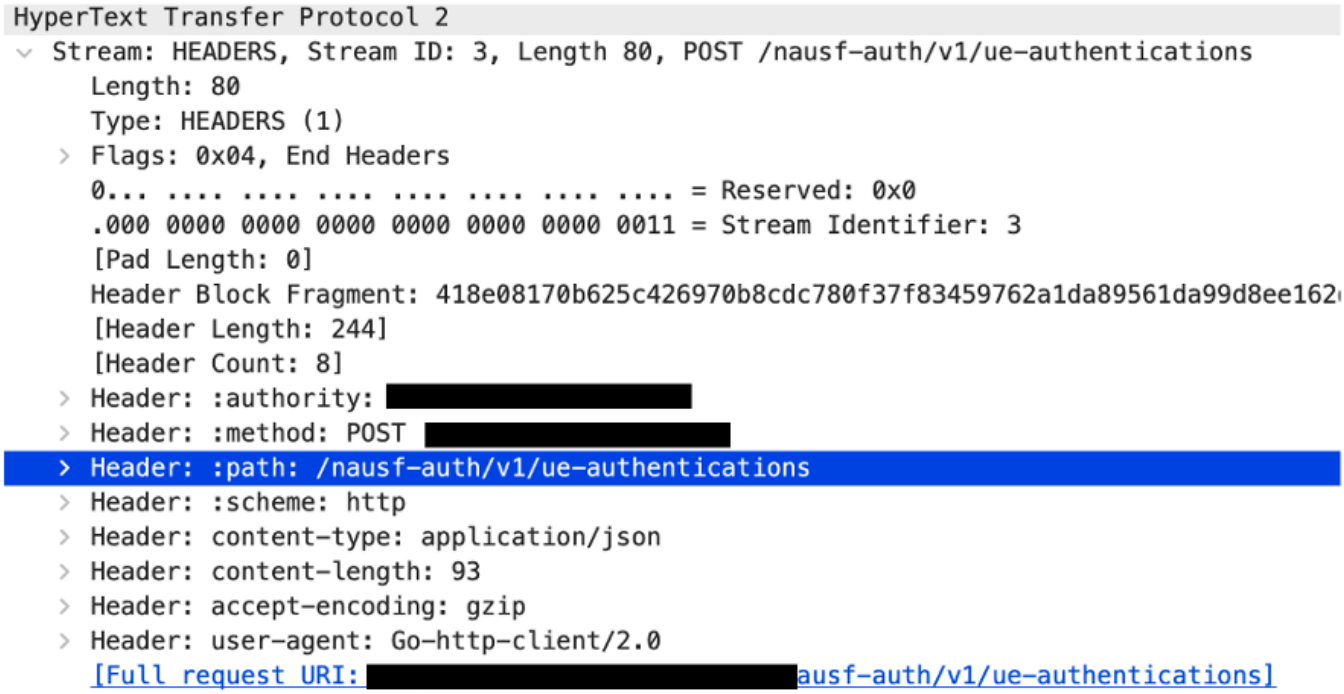
style	explode	empty	string	array	object
matrix	false	;color	;color=blue	;color=blue,black,brown	;color=R,1(
matrix	true	;color	;color=blue	;color=blue;color=black;color=brown	;R=100;G=
label	false	.	.blue	.blue.black.brown	.R.100.G.2
label	true	.	.blue	.blue.black.brown	.R=100.G=
form	false	color=	color=blue	color=blue,black,brown	color=R,10
form	true	color=	color=blue	color=blue&color=black&color=brown	R=100&G=
simple	false	n/a	blue	blue,black,brown	R,100,G,2(
simple	true	n/a	blue	blue,black,brown	R=100,G=2
spaceDelimited	false	n/a	n/a	blue%20black%20brown	R%20100%
pipeDelimited	false	n/a	n/a	blue|black|brown	R|100|G|2(
deepObject	true	n/a	n/a	n/a	color[R]=1(

Afbeelding D: (Snapshot van Open API: (4.7.12.4 Stijlvoorbeelden))

U hebt CLI-controle in zowel AMF als SMF om de parameter 3gpp-sbi-discovery-service te verzenden, omdat dit optioneel is (kan worden gedaan afhankelijk van de implementatieomgeving).

In het geval van Model-B, als u het voorbeeld neemt van AMF- en AUSF-communicatie (Authentication Server Function), verzendt de AMF de POST naar AUSF met AUSF IP / FQDN en poort.

POST http://<ausf-fqdn>:<port>/nausf-auth/v1/ue-authentications.



Afbeelding E: (communicatie tussen AMF en AUSF via model B)

In Model-D verzendt het AMF, aangezien de detectie wordt uitgevoerd door het SCP, in plaats van POST-http(s)://<ausf-fqdn>:<ausf-port>/nausf-auth/v1/ue-authenticaties, het gewijzigde POST-verzoek dat:

POST http(s)://<scp-fqdn>:<scp-port>/nausf-auth/v1/ue-authentications

OF

POST http(s)://<scp-fqdn>:<scp-port>/nscp-route/nausf-auth/v1/ue-authenticaties (indien apiroot=nscp-route)

met

3gpp-Sbi-Discovery-target-nf-type: AUSF

3gpp-Sbi-Discovery-Preferred-locality: LOC1

3gpp-Sbi-Discovery-servicenaam

Waar u kunt zien dat AMF de api-root (<ausf-fqdn>:<ausf-port>) van de AUSF heeft vervangen door de api-root van de SCP.

```
> Header: :authority: [redacted]
> Header: :method: POST
> Header: :path: /nscp-route/nausf-auth/v1/ue-authentications
> Header: :scheme: http
> Header: 3gpp-sbi-discovery-service-names: ["nausf-auth"]
> Header: 3gpp-sbi-discovery-target-nf-type: AUSF
> Header: 3gpp-sbi-discovery-requester-nf-type: AMF
> Header: user-agent: AMF-SLICE-EMBB
> Header: 3gpp-sbi-discovery-target-plmn-list: [{"mcc":[redacted],"mnc":[redacted]]
> Header: content-type: application/json
> Header: content-length: 183
> Header: accept-encoding: gzip
  [Full request URI: http://[redacted]/nscp-route/nausf-auth/v1/ue-authentications]
```

Figuur F: (AMF-AUSF-communicatie via SCP-model D)

Met de 3gpp-sbi-discovery-parameters kan de SCP de beste NF ophalen en vervolgens het POST-verzoek doorsturen waar het de api-root van de SCP vervangt door de api-root die van de NRF is ontvangen nadat het antwoord op het ontdekkingsverzoek is ontvangen.

Configuraties vereist bij AMF/SMF

Om voor elke NF (bijvoorbeeld UDM) aan te geven welk oproepmodel moet worden gebruikt, wordt de nf-selectie-modelconfiguratie gebruikt binnen het bijbehorende 'profielnetwerkelement'.

<#root>

profile network-element udm prf-udm-scp

[...]

nf-selection-model priority <>[local | nrf-query | nrf-query-peer-input | nrf-query-and-scp | scp]

exit

Zodra Model-D is gekozen, worden de queryparameters die zijn geconfigureerd voor het bijbehorende netwerkelement nog steeds gebruikt en worden ze doorgegeven aan het SCP in de indeling '3gpp-Sbi-Discovery-<query-param>'.

<#root>

```
[smf] smf(config)# profile network-element udm prf-udm-scp
```

```
[smf] smf(config-udm-udm1)# query-params
```

```
Possible completions:
```

```
 [  chf-supported-plmn  dnn  requester-snssais  tai  target-nf-instance-id  target-plmn ]
```

Uiteindelijk wordt het profielnetwerkelement toegewezen aan de naam van het gegevensnetwerk (dnn).

<#root>

```
profile dnn ims
```

```
 network-element-profiles udm prf-udm-scp
```

```
 network-element-profiles scp prf-scp
```

```
exit
```

SCP's worden gedefinieerd als netwerkelement.

NF-client-profiel en een profiel voor het afhandelen van storingen is gekoppeld aan netwerkelement.

<#root>

```
profile network-element scp <>
```

```
 nf-client-profile        <>

 failure-handling-profile <>

exit
```

Het nf-client-profiel van het type scp-profiel beschrijft de kenmerken van het SCP-eindpunt.

Hier kan nscp-route worden toegevoegd in api-root.

<#root>

```
profile nf-client nf-type scp

 scp-profile <>

  locality LOC1

   priority 30

   service name type <>

    responsetimeout 4000

    endpoint-profile EP1

     capacity   30

     api-root   nscp-route
```

```
     priority   10

     uri-scheme http

     endpoint-name scp-customer.com

      priority 10

      capacity 50

      primary ip-address ipv4

      primary ip-address port

      fqdn name <>

      fqdn port <>


exit
```

SMF FQDN is geconfigureerd in de SBI (endpoint southbound interface).

<#root>

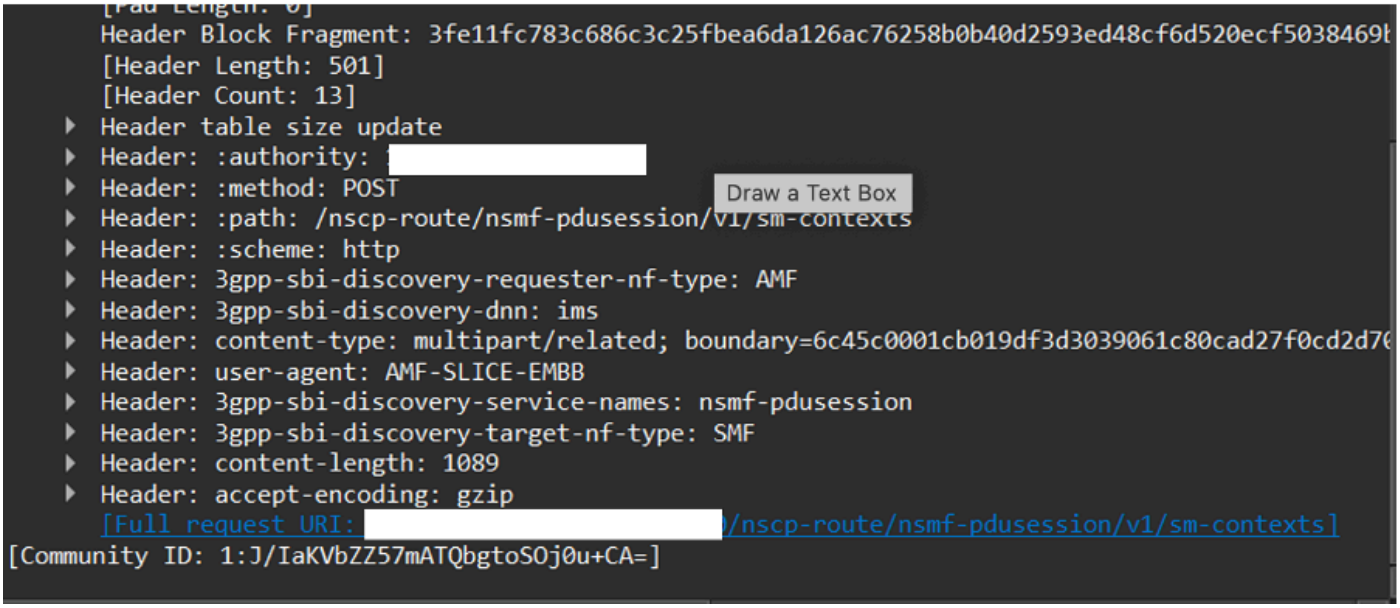
```
endpoint sbi
```

relicas 2

nodes 2

fqdn <>

Voorbeeld van Snap-pakketten



Figuur G: (AMF- SMF-nsmf-pressiecommunicatie via SCP Model D)

U moet vanuit het profiel dnn naar het zojuist geconfigureerde SCP-netwerkelement verwijzen.

<#root>

profile dnn <>

network-element-profiles udm <>

network-element-profiles scp <>

exit

Als SCP-foutverwerking is geconfigureerd met actie als herhaal, probeert SMF alternatieve SCP te maken op basis van SCP-configuratie en herhaal het aantal pogingen.

Als SCP-storingsafhandeling is geconfigureerd met actie als retry-and-fallback voor een bepaalde servicenaam en berichttype, wordt teruggevallen naar Model-A.

Dit FOUTVERWERKINGSPROFIEL voor SCP (FHSCP) wordt gebruikt als de fout wordt geactiveerd vanuit de SCP (serverheader die SCP aangeeft) en de NF-clientconfiguratie voor de peer aanwezig is.

<#root>

```
profile nf-client-failure nf-type scp


   profile failure-handling <>


   service name type npcf-smpolicycontrol


    responsetimeout 1800


    message type PcfSmpolicycontrolCreate


    status-code httpv2 0,307,429,500,503-504


     retry  1


     action retry-and-fallback


     exit
```

Voorbeeld van het nf-clientprofiel voor de beleidscontrolefunctie (Policy Control Function, PCF) voor het scenario waarin de actie retry and falback is geconfigureerd voor het berichttype

PcfSmpolicycontrolCreate:

<#root>

```
profile nf-client nf-type pcf


 pcf-profile <>


  locality LOC1


   priority 1


   service name type npcf-smpolicycontrol


    endpoint-profile epprof


     capacity   10


     priority   1


     uri-scheme http


     endpoint-name ep1


      priority 1


      capacity 10


      primary ip-address ipv4 <>
```

```
 primary ip-address port <>

exit

endpoint-name ep2

 priority 1

 capacity 10

 primary ip-address ipv4 <>

 primary ip-address port <>

exit
```

Core DNS-POD's en configuratie vereist op SMI-laag

CoreDNS-pods, die deel uitmaken van de naamruimte van het kube-systeem, worden geïmplementeerd als een replicaset met twee pods. Deze pods kunnen worden gepland op elk van de twee master/control nodes en zijn niet afhankelijk van waar de nameserver IP is geconfigureerd in clusterbeheer.

Het wordt echter aanbevolen om de nameserver-IP in alle besturings-/masternodes te configureren, omdat u geen labelcontrole hebt om de CoreDNS-pods te draaien volgens uw wens. Als de route naar nameservers niet aanwezig is op een van de master's waar CoreDNS wordt geïmplementeerd, wordt de SMF/AMF-clustersynchronisatie mislukt.

Momenteel stuurt CoreDNS-verzoeken door naar de naamserver die is opgegeven in het bestand resolv.conf.

'Kubectl Edit Configmap Corends -n Kube-System' heb je:

<#root>

```
 {


forward ./etc/resolv.conf{


     max_concurrent 1000


  }
```

Wanneer u /etc/resolv.conf controleert op de hoofdnode waar de service wordt gestart, moet deze het volgende bevatten:

<#root>

```
 name server <>


     name server <>
```

Voorbeeld van nameserverconfiguratie in master/control node:

<#root>

```
nodes <>


initial-boot netplan vlans <>


dhcp4 false


dhcp6 false
```

addresses [<>]

nameserver addresses [<>]

id <>

link <>

exit

Over deze vertaling

Cisco heeft dit document vertaald via een combinatie van machine- en menselijke technologie om onze gebruikers wereldwijd ondersteuningscontent te bieden in hun eigen taal. Houd er rekening mee dat zelfs de beste machinevertaling niet net zo nauwkeurig is als die van een professionele vertaler. Cisco Systems, Inc. is niet aansprakelijk voor de nauwkeurigheid van deze vertalingen en raadt aan altijd het oorspronkelijke Engelstalige document (link) te raadplegen.