

SCPモデルDベースのコミュニケーションの詳細

内容

[はじめに](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[背景説明](#)

[アーキテクチャとソリューションの概要](#)

[AMF/SMFで必要な設定](#)

[パケットのスナップ例](#)

[SMIレイヤで必要なコアDNS PODおよび設定](#)

はじめに

このドキュメントでは、Cisco AMF/SMFとサードパーティのNF間のSCPモデルD通信アプローチの詳細について説明します。

前提条件

要件

次の項目に関する知識があることが推奨されます。

- Access and Mobility Management Function(AMF)の機能
- セッション管理機能(SMF)の機能
- サービスコミュニケーションプロキシ(SCP)の機能

使用するコンポーネント

このドキュメントの内容は、特定のソフトウェアやハードウェアのバージョンに限定されるものではありません。

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、クリアな（デフォルト）設定で作業を開始しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

背景説明

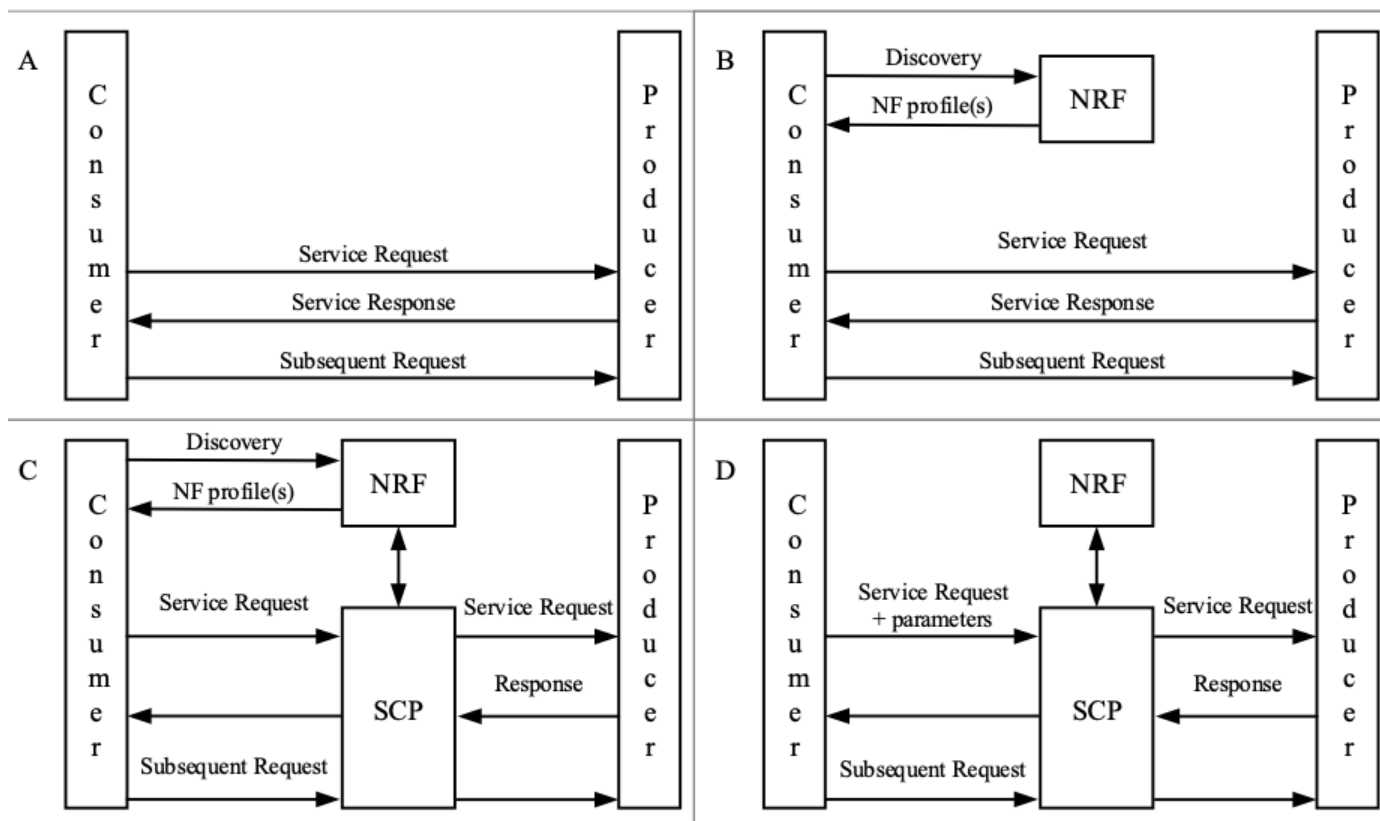
世界中のオペレータは、ネットワーク機能(NF)の検出にSCPを使用し、NF間の通信にSCPを使用する複数の通信モデルから選択できます。この項では、さまざまな通信モデルに関する概念と、

SCPモデルDベースの通信を実現するために加入者マイクロサービスインフラストラクチャ (SMI)、AMF/SMFで必要なコールフロー/設定の変更について説明します。

アーキテクチャとソリューションの概要

Service Based Architecture(SBA)では、SCPが仲介として機能し、ルーティング、ロードバランシング、およびサービスディスカバリの処理によってNF間の間接的な通信を促進し、最終的にサービスベースアーキテクチャを合理化します。

3GPP 23.501 Annex-Eでは、5GC導入におけるNF間の4つの通信モデルについて詳しく説明します。



図A: (SCPに関連するさまざまなコミュニケーションモデル)

モデルA - Network Repository Function(NRF)インタラクションを伴わない直接通信：コンシューマは、プロデューサの「NFプロファイル」を使用して設定され、選択したプロデューサと直接通信します。これはスタティック選択のタイプであり、NRFもSCPも使用されません。

モデルB:NRFインタラクションとの直接通信：コンシューマはNRFに対してクエリーを実行することで検出を行います。検出結果に基づいて、コンシューマが選択を行います。コンシューマは、選択されたプロデューサに要求を送信します。

Model-C：委託されたディスカバリを使用しない間接的な通信：コンシューマはNRFのクエリーを実行して検出します。発見結果に基づいて、消費者は、NFセットまたはNFセットの特定のNFインスタンスの選択を行う。コンシューマは、NFサービスインスタンスまたはNFサービスインスタンスのセットを指し示す、選択されたサービスプロデューサのアドレスを含む要求をSCPに送信します。後者の場合、SCPはNFサービスインスタンスを選択します。可能であれば、

SCPはNRFと通信して、場所や容量などの選択パラメータを取得します。SCPは、選択されたNFサービス開発インスタンスに要求をルーティングします。

モデルD – 委任されたディスカバリを使用した間接的な通信：コンシューマはディスカバリや選択に関与しません。コンシューマは、適切なプロデューサを見つけるために必要な検出および選択パラメータをサービスリクエストに追加します。SCPは、要求アドレスと、要求メッセージ内の検出および選択パラメータを使用して、要求を適切なプロデューサインスタンスにルーティングします。SCPは、NRFを使用して検出を実行し、検出結果を取得できます。

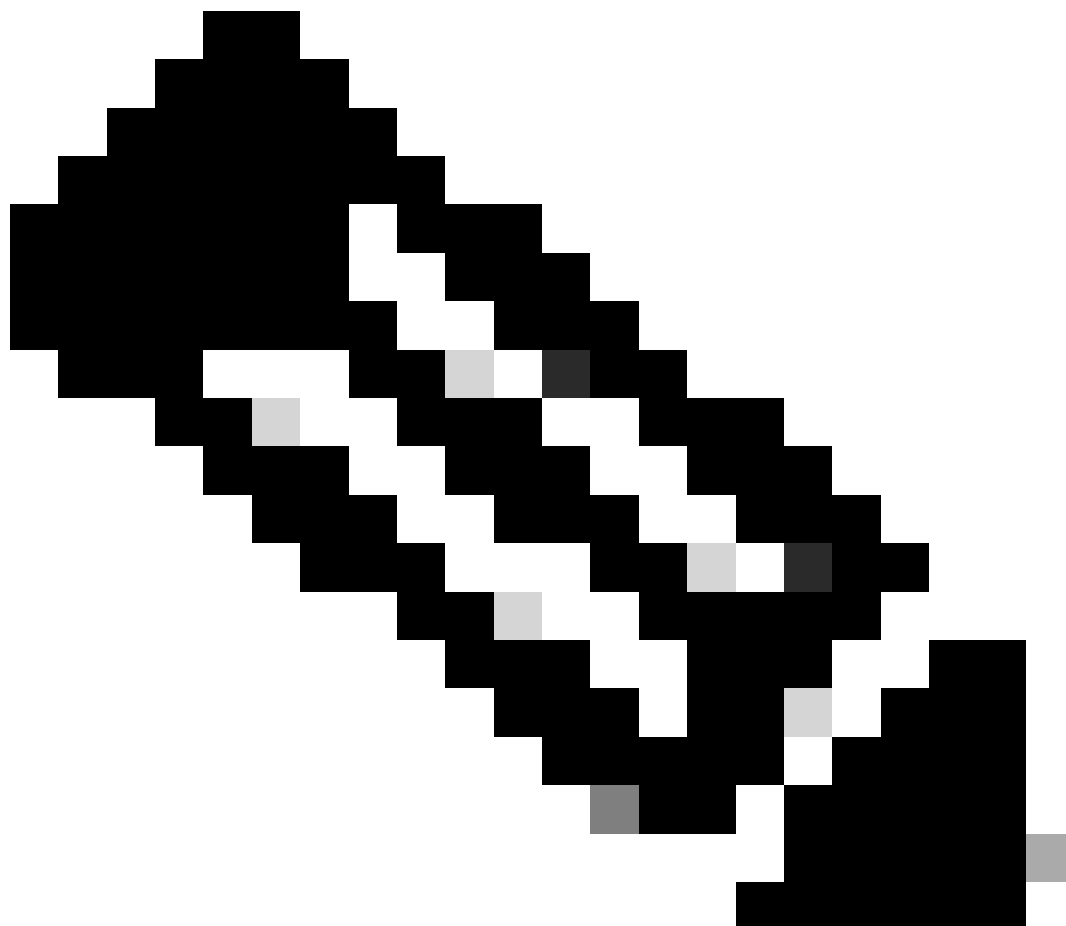
モデルDベースの通信の詳細：コールモデルDが使用される場合、NFコンシューマはNRFに直接要求を送信するのではなく、このディスカバリをSCPに委任します。NFクライアントはSCPにメッセージを送信し、これらのディスカバリ要素ごとに、文字列「3gpp-sbi-discovery」をディスカバリ要素の名前と連結します。この文字列は、NFディスカバリがNRF経由で実行される場合に使用されます。

SMFがサービス名nudm-sdmを持つUnified Data Management(UDM)を検索するシナリオの場合、検出ファクタはSCPに渡されます。

- 機関ヘッダー：機関は完全修飾ドメイン名(FQDN)またはIPアドレスのいずれかを伝送し、IPアドレス設定に優先順位が与えられます。
- 3gpp-sbi-discovery-requester-nf-type:SMF
- 3gpp-sbi-discovery-target-nf-type:UDM
- 3gpp-Sbi-discovery-service-name: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]
```

図B: (SCPモデルD経由のSMF-UDM通信)



注:3gpp-sbi-discovery-service名前形式は、3gpp 29.510およびオープンAPI定義
(4.7.12.4スタイル) に従い、プレーンストリング形式であり、アレイ形式ではありません。
29.510では、3gpp-sbi-discovery-service-nameがアレイ形式として示されています

。

```
- 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
```

図C: (29.510仕様のスナップショット)

ただし、style:formおよびexplode:falseは、配列を単純な文字列に変換します。これは、OpenAPIから例を取ることで説明します。

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,100,G=200,B=150
matrix	true	;color	;color=blue	;color=blue;color=black;color=brown	;R=100;G=200;B=150
label	false	.	.blue	.blue.black.brown	.R.100.G.200.B.150
label	true	.	.blue	.blue.black.brown	.R=100.G=200.B=150
form	false	color=	color=blue	color=blue,black,brown	color=R,100,G=200,B=150
form	true	color=	color=blue	color=blue&color=black&color=brown	R=100&G=200&B=150
simple	false	n/a	blue	blue,black,brown	R,100,G,200,B,150
simple	true	n/a	blue	blue,black,brown	R=100,G=200,B=150
spaceDelimited	false	n/a	n/a	blue%20black%20brown	R%20100%20G%20200%20B%20150
pipeDelimited	false	n/a	n/a	blue black brown	R 100 G 200 B 150
deepObject	true	n/a	n/a	n/a	color[R]=100&G=200&B=150

図D: (Open APIからのスナップショット : (4.7.12.4スタイルの例))

パラメータ3gpp-sbi-discovery-serviceの送信はオプションであるため (導入環境に応じて実行可能) 、AMFとSMFの両方でCLI制御を使用できます。

モデルBの場合、AMFとAuthentication Server Function(AUSF)通信の例を取ると、AUSFが検出されると、AMFはAUSF IP/FQDNとポートを使用してPOSTをAUSFに送信します。

<http://<ausf-fqdn>:<port>/nausf-auth/v1/ue-authentications>後にログインします。

HyperText Transfer Protocol 2

```
√ Stream: HEADERS, Stream ID: 3, Length 80, POST /nausf-auth/v1/ue-authentications
  Length: 80
  Type: HEADERS (1)
  > Flags: 0x04, End Headers
    0... .. = Reserved: 0x0
    .000 0000 0000 0000 0000 0000 0000 0011 = Stream Identifier: 3
    [Pad Length: 0]
    Header Block Fragment: 418e08170b625c426970b8cdc780f37f83459762a1da89561da99d8ee162a
    [Header Length: 244]
    [Header Count: 8]
  > Header: :authority: [REDACTED]
  > Header: :method: POST [REDACTED]
  > Header: :path: /nausf-auth/v1/ue-authentications
  > Header: :scheme: http
  > Header: content-type: application/json
  > Header: content-length: 93
  > Header: accept-encoding: gzip
  > Header: user-agent: Go-http-client/2.0
  [Full request URI: [REDACTED]ausf-auth/v1/ue-authentications]
```

図E: (モデルB経由のAMF-AUSF通信)

モデルDでは、検出はPOST http(s)://<ausf-fqdn>:<ausf-port>/nausf-auth/v1/ue-authenticationsの代わりにSCPによって実行されるため、AMFは変更されたPOST要求を送信します。この要求の内容は次のとおりです。

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

または

POST http(s)://<scp-fqdn>:<scp-port>/nscf-route/nausf-auth/v1/ue-authentications(apiroot=nscf-routeの場合)

さらにトラブルシューティングを行うために、

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

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

3gpp-Sbi-Discovery-service-name

AMFがAUSFのapi-root(<ausf-fqdn>:<ausf-port>)をSCPのapi-rootに置き換えたことがわかります。

```

> Header: :authority: [REDACTED]
> Header: :method: POST
> Header: :path: /nscf-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]/nscf-route/nausf-auth/v1/ue-authentications]

```

図F: (SCPモデルD経由のAMF-AUSF通信)

3gpp-sbi-discoveryパラメータを使用すると、SCPは最適なNFを取得してPOST要求を転送し、ディスカバリ要求への応答を受信した後、SCPのapi-rootをNRFから受信したapi-rootに置き換えることができます。

AMF/SMFで必要な設定

使用する必要があるコールモデルを各NF (UDMなど) に示すために、関連する「profile network-element」内でnf-selection-model設定が使用されます。

```
<#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
```

Model-Dを選択すると、関連付けられたネットワーク要素に対して設定されたquery-paramsが引き続き使用され、「3gpp-Sbi-Discovery-<query-param>」の形式でSCPに渡されます。

<#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 ]
```

最終的に、プロファイルのネットワーク要素がプロファイルのデータネットワーク名(dnn)にマッピングされます。

<#root>

```
profile dnn ims
```

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

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

```
exit
```

SCPはネットワーク要素として定義されます。

nf-client-profileと障害処理プロファイルはnetwork-elementでマッピングされます。

<#root>

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

```
nf-client-profile <>
```

failure-handling-profile <>

exit

scp-profileタイプのnf-client-profileには、SCPエンドポイントの特性の詳細が記述されます。
ここでは、nscp-routeをapi-rootに追加できます。

<#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は、エンドポイントサウスバウンドインターフェイス(SBI)で設定します。

<#root>

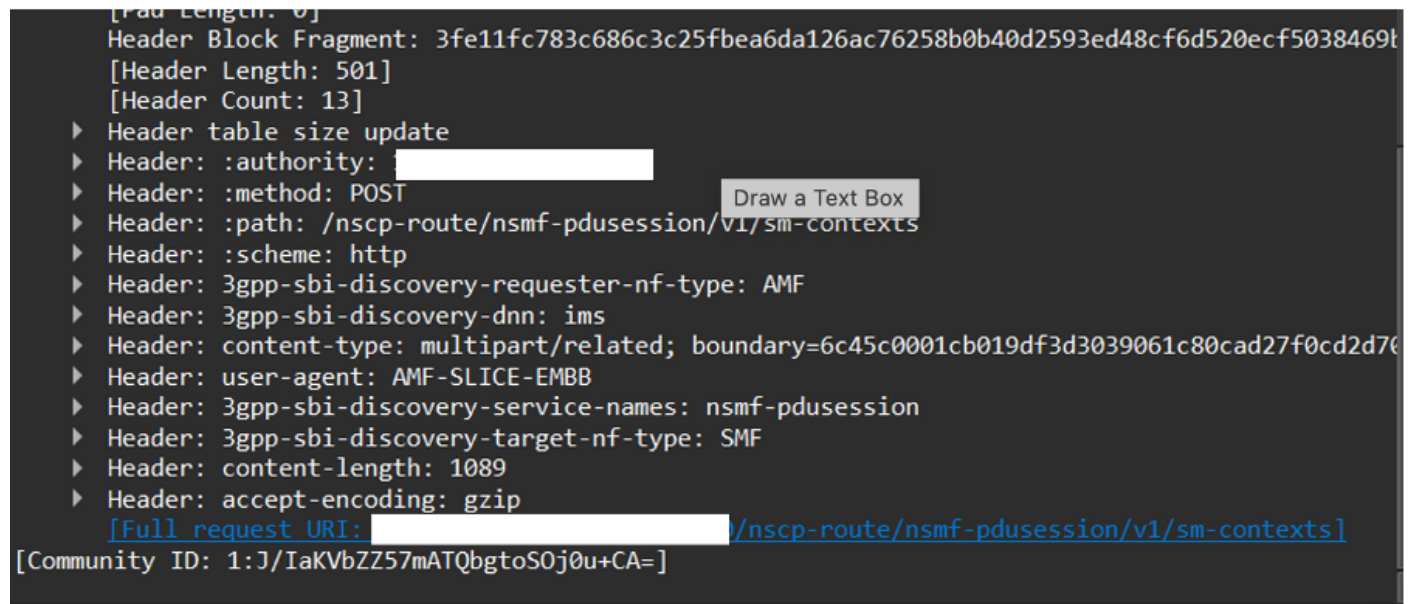
endpoint sbi

relicas 2

nodes 2

fqdn <>

パケットのスナップ例



```
[Pad Length: 0]
Header Block Fragment: 3fe11fc783c686c3c25fbea6da126ac76258b0b40d2593ed48cf6d520ecf5038469t
[Header Length: 501]
[Header Count: 13]
▶ Header table size update
▶ Header: :authority: [redacted]
▶ Header: :method: POST
▶ Header: :path: /nscp-route/nsmf-pdusession/v1/sm-contexts
▶ Header: :scheme: http
▶ Header: 3gpp-sbi-discovery-requester-nf-type: AMF
▶ Header: 3gpp-sbi-discovery-dnn: ims
▶ Header: content-type: multipart/related; boundary=6c45c0001cb019df3d3039061c80cad27f0cd2d76
▶ Header: user-agent: AMF-SLICE-EMBB
▶ Header: 3gpp-sbi-discovery-service-names: nsmf-pdusession
▶ Header: 3gpp-sbi-discovery-target-nf-type: SMF
▶ Header: content-length: 1089
▶ Header: accept-encoding: gzip
[Full request URI: [redacted]/nscp-route/nsmf-pdusession/v1/sm-contexts]
[Community ID: 1:J/IaKVbZZ57mATQbgtoSOj0u+CA=]
```

図G: (AMF- SMF nsmf-pdusession通信 (SCPモデルD)

プロファイルdnnから、設定したSCPネットワーク要素を参照する必要があります。

<#root>

profile dnn <>

network-element-profiles udm <>

network-element-profiles scp <>

exit

SCP障害処理にretryのアクションが設定されている場合、SMFはSCP設定と再試行回数に基づいて代替SCPを試行します。

特定のサービス名とメッセージタイプに対して、アクションとしてretry-and-fallbackを使用してSCP障害処理が設定されている場合、モデルAへのフォールバックが発生します。

SCP(FHSCP)用のこの障害処理プロファイルは、エラーがSCP (SCPを示すサーバヘッダー) からトリガーされ、ピアのNFクライアント設定が存在する場合に使用されます。

```
<#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
```

アクションリトライおよびフォールバックがメッセージタイプPcfSmpolicycontrolCreateに対して設定されるシナリオの、ポリシー制御機能(PCF)のnf-clientプロファイルの例。

<#root>

profile nf-client nf-type pcf

pcf-profile <>

locality LOC1

priority 1

service name type npcfsmpolicycontrol

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
```

SMIレイヤで必要なコアDNS PODおよび設定

kube-system名前空間の一部であるCoreDNSポッドは、2ポッドレプリカセットとして導入されます。これらのポッドは、2つのマスター/制御ノードのいずれかでスケジュールでき、ネームサーバIPがクラスタマネージャのどこに設定されているかには依存しません。

ただし、必要に応じてCoreDNSポッドをスピンするためのラベリング制御がないため、すべてのコントロール/マスターノードでネームサーバIPを設定することをお勧めします。CoreDNSが配備されているどのマスターにもネームサーバへのルートが存在しない場合、SMF/AMFクラスタの同期は失敗します。

現在、CoreDNSはノードのresolv.confファイルで指定されたネームサーバにDNS要求を転送します。

'kubectl edit configmap coredns -n kube-system'を使用します。

```
<#root>
```

```
{
```

```
forward ./etc/resolv.conf{

    max_concurrent 1000

}
```

サービスを開始するマスターノード上で/etc/resolv.confをチェックする場合、次の情報が含まれている必要があります。

```
<#root>
```

```
name server <>
```

```
name server <>
```

マスター/コントロールノードでのネームサーバの設定例：

```
<#root>
```

```
nodes <>
```

```
initial-boot netplan vlans <>
```

```
dhcp4 false
```

```
dhcp6 false
```

```
addresses [<>]
```

nameserver addresses [<>]

id <>

link <>

exit

翻訳について

シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。