

NSO-serviceeigendom begrijpen

Inhoud

[Inleiding](#)

[Voorwaarden](#)

[Vereisten](#)

[Gebruikte componenten](#)

[Grondbeginselen](#)

[Apparaateigendom](#)

[Eigendom in actie](#)

[Showcase: alleen in eigendom van de service](#)

[Showcase: alleen apparaat in eigendom](#)

[Showcase: apparaat bezeten + service bezeten](#)

[verzoenen](#)

[Doel](#)

[Eigendom van services overdragen](#)

[Oorspronkelijke waardmarkeringen verwijderen](#)

[De eigendom van de dienst corrigeren](#)

[Reconcile tijdens herimplementatie](#)

[Innerlijke werkstukken met elkaar verzoenen](#)

[Problemen met serviceeigendom](#)

[Reconcile mislukt](#)

[Service verwijdert configuratie die geen eigendom is van de service](#)

[Showcase: Service verwijdert configuratie die geen eigendom is van service](#)

[Keep-non-service-config vs verwerping-niet-service-configuratie](#)

[Geen aanmaken](#)

[Samenvoegen](#)

Inleiding

Dit document beschrijft het algemene concept, gemeenschappelijke valkuilen en oplossingen voor serviceeigendom in Cisco® Network Service Orchestrator (NSO).

Voorwaarden

Vereisten

Dit document is van toepassing op alle momenteel beschikbare NSO-versies, met inbegrip van NSO 6. Het beschreven gedrag is alleen van toepassing op NSO-instellingen die gebruikmaken van een combinatie van diensten en niet-dienstconfiguratie. Terwijl de specifieke opdrachten die in voorbeelden in dit document worden gebruikt, alleen van toepassing zijn op de gebruikte Network

Element Driver (NED), is de onderliggende logica van toepassing op elk apparaat dat door NSO wordt beheerd.

Gebruikte componenten

- NSO 6.1.6
- NED: test-ned 1.0 , een op maat gemaakte netconf NED met behulp van dit yang model.

NED yang model:

```
module test-ned{
  namespace "http://example.org/ned/service-ownership";
  prefix ownership;
  import ietf-inet-types{ prefix inet;}

  list interface {
    key interface-name;
    leaf interface-name{
      type string;
    }

    leaf ip-address {
      type inet:ipv4-address;
    }

    leaf description {
      type string;
    }
  }
}
```

- Service: voorbeeld-service 1.0: een op maat gemaakte service met behulp van dit yang model en sjabloon

```
module example-service {
  namespace "http://com/example/exampleservice";
  prefix example-service;

  import ietf-inet-types {
    prefix inet;
  }
  import tailf-ncs {
    prefix ncs;
  }

  list example-service {
    key name;

    uses ncs:service-data;
    ncs:servicepoint "example-service";

    leaf name {
      type string;
    }
  }
}
```

```
leaf-list device {  
  type leafref {  
    path "/ncs:devices/ncs:device/ncs:name";  
  }  
}  
leaf ipaddress {  
  type inet:ipv4-address;  
}  
}  
}
```

{/device}

FE1

{/ipaddress}

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.

Grondbeginselen

Het doel van de dienst eigendom is om NSO toe te staan om te volgen welke configuratie met betrekking tot welke dienst is. Wanneer u een service verwijdert, moet NSO de configuratie die eraan gerelateerd is verwijderen en gebruikt hij serviceeigendom om te bepalen welke configuratie verwijderd moet worden. Wanneer de configuratie door meer dan één dienst wordt bezeten, verwijdert het schrappen van één van de diensten eenvoudig die eigendomsverwijzing; De configuratie zelf blijft in het gegevensbestand van NSO (CDB) en op de apparaten van het netwerk.

Eigendom wordt weergegeven door middel van refcounts en backpointers. Een referentie laat zien hoeveel entiteiten dat deel van de configuratie bezitten. Een hertelling is gelijk aan de hoeveelheid service-instanties plus 1 als de configuratie ook eigendom is van het apparaat. Een backpointer toont het pad van deze service instanties. Er is geen backpointer om "device owned" weer te geven. Backpointers worden alleen in CDB weergegeven voor lijsten en containers. Individuele bladeren tonen hun backpointers niet, maar ze erven van hun ouder.

Apparaateigendom

Naast de configuratie die eigendom is van een service, kan het ook eigendom zijn van het apparaat. Dit wordt soms "device owned" of "non-service owned" genoemd. Hoewel dit document "device-owned" gebruikt, merk op dat dit weliswaar makkelijker te begrijpen is, maar dat de niet-service eigendom geen apparaten hoeft te omvatten. LSA-instellingen of gestapelde services kunnen een configuratie zonder apparaten hebben die geen eigendom is van de service.

De configuratie is apparaat-bezeten wanneer de configuratie aan CDB zonder het gebruik van een de dienstplaatsing wordt toegevoegd maar in plaats daarvan een methode zoals sync-van, ladingssamenvoeging, of ncs_cli gebruikte om de configuratie te plaatsen. Wanneer een service-instantie eigenaar wordt van een configuratie die al in bezit was van een apparaat, wordt de refcount ingesteld op 2 om het gedeelde eigendom weer te geven. Wanneer de service-instantie wordt verwijderd, wordt de configuratie niet verwijderd, ondanks het feit dat slechts één service-instantie de configuratie vóór het wissen bezit. Bovendien voegt de apparaatconfiguratie een "originele waarde"-tag toe. Als een service-instantie de configuratie van het apparaat overschrijft en de service later wordt verwijderd, wordt de oorspronkelijke waarde van de configuratie hersteld.

Apparaateigendom wordt alleen toegewezen als de configuratie nog niet bestond in CDB wanneer u het toevoegt via niet-service middelen. De configuratie van de serviceconfiguratie wordt geen apparaat dat eigendom is na sync-vanaf. Maar de apparaat-bezeten configuratie wordt zowel apparaat bezeten als de dienst bezeten als u de dienst bovenop opstelt.

Eigendom in actie

Showcase: alleen in eigendom van de service

Wanneer u een service implementeert terwijl de doelconfiguratie leeg is, creëert de service de configuratie en neemt eigendom van deze. De gebruiker kan eigendom controleren door een show in werking stelt -in werking stellen-configuratiebevel te gebruiken en toe te voegen | weergave van service-meta-gegevens. Hoewel dit niet verplicht is, wordt aanbevolen om ook | xml weergeven als de standaard CLI stijl uitvoer niet altijd correct weerspiegelt hoe de gegevens worden gemodelleerd in CDB.

```
admin@ncs(config)# do show running-config devices device mydevice0 config
% No entries found.
admin@ncs(config)# example-service s1 device mydevice0 ipaddress 192.0.2.1
admin@ncs(config-example-service-s1)# commit dry-run
cli {
  local-node {
    data devices {
      device mydevice0 {
        config {
          +       interface FE1 {
          +         ip-address 192.0.2.1;
          +       }
        }
      }
    }
  }
  +example-service s1 {
  +  device [ mydevice0 ];
  +  ipaddress 192.0.2.1;
  +}
}
admin@ncs(config-example-service-s1)# commit
Commit complete.
admin@ncs(config-example-service-s1)# do show running-config devices device mydevice0 config | display
```

mydevice0

FE1

192.0.2.1

Daarnaast, als u een tweede service instantie met dezelfde configuratie toevoegt, wordt eigendom gedeeld door de 2 service-instanties. De Refcount is 2 en er zijn 2 backpointers.

```
admin@ncs(config-example-service-s1)# example-service s2 device mydevice0 ipaddress 192.0.2.2
admin@ncs(config-example-service-s2)# commit dry-run
cli {
  local-node {
    data devices {
      device mydevice0 {
        config {
          interface FE1 {
            - ip-address 192.0.2.1;
            + ip-address 192.0.2.2;
          }
        }
      }
    }
  }
}
```

```
        }
    }
}
+example-service s2 {
+  device [ mydevice0 ];
+  ipaddress 192.0.2.2;
+}
}
}
admin@ncs(config-example-service-s2)# commit
Commit complete.
admin@ncs(config-example-service-s2)# do show running-config devices device mydevice0 config | display
```

mydevice0

FE1

192.0.2.2

Showcase: alleen apparaat in eigendom

Wanneer u gegevens aan CDB toevoegt met behulp van `load merge`, `ncs_cli` of `sync-vanaf`, zonder het gebruik van een service, worden deze gegevens apparaat-eigendom. De `refcount` en de `backpointer` zijn verborgen.

```
admin@ncs(config)# no example-service
admin@ncs(config)# commit
Commit complete.
admin@ncs(config)# do show running-config devices device mydevice0 config
% No entries found.
admin@ncs(config)# load merge merge-config.xml
Loading.
386 bytes parsed in 0.00 sec (137.22 KiB/sec)
admin@ncs(config)# commit dry-run
cli {
  local-node {
    data devices {
      device mydevice0 {
        config {
          +       interface FE1 {
          +         ip-address 192.0.2.1;
          +       }
        }
      }
    }
  }
}
admin@ncs(config)# commit
Commit complete.
admin@ncs(config)# do show running-config devices device mydevice0 config | display service-meta-data |
```

mydevice0

FE1

192.0.2.1

Showcase: apparaat bezeten + service bezeten

Dit voorbeeld laat zien hoe u eenvoudig gecombineerde apparaat- en serviceeigendom kunt creëren met behulp van een service en sync-vanaf.

U implementeert de service en verwijdert de service alleen in CDB met behulp van Commit no-networking. Op deze manier bestaat de configuratie nog steeds op het eindapparaat. Wanneer u sync-vanaf uitvoert, wordt de configuratie opnieuw toegevoegd in CDB, maar het is apparaat bezeten in plaats van de dienst bezeten. Vergeet niet dat in werking stellen-configuratie toont wanneer in werking gesteld in NSO u CDB gegevens, niet gegevens momenteel op de apparaten toont.

```
admin@ncs(config)# no devices device mydevice0 config
admin@ncs(config)# commit
admin@ncs(config)# do show running-config devices device mydevice0 config
% No entries found.
```

```
admin@ncs(config)# example-service s1 device mydevice0 ipaddress 192.0.2.1
admin@ncs(config-example-service-s1)# commit
admin@ncs(config-example-service-s1)# top
admin@ncs(config)# no example-service s1
admin@ncs(config)# commit no-networking
Commit complete.
admin@ncs(config)# devices device mydevice0 sync-from
result true
admin@ncs(config)# do show running-config devices device mydevice0 config | display service-meta-data |
```

mydevice0

FE1

192.0.2.1

Na sync-vanaf is de configuratie alleen eigendom van het apparaat. Refcounter is verborgen. Wanneer u de service opnieuw implementeert, wordt de refcount 2, maar de backpointer wordt slechts één service-instantie weergegeven. De tweede refcounter staat voor het eigendom van de apparatuur. Dezelfde regels zijn van toepassing als met gedeelde service-eigendom, als de service wordt verwijderd, wordt de configuratie niet verwijderd omdat het apparaat ook gedeeltelijk eigenaar is van de configuratie. Bovendien, als de de dienstgegevens niet de "origineel-waarde"zoals opgeslagen in de dienst-meta-gegevens aanpassen, keert NSO de waarde terug naar de "origineel-waarde"terug als de dienst wordt verwijderd.

```
admin@ncs(config)# example-service s1 device mydevice0 ipaddress 192.0.2.2
admin@ncs(config-example-service-s1)# commit dry-run
cli {
  local-node {
    data devices {
      device mydevice0 {
        config {
          interface FE1 {
            - ip-address 192.0.2.1;
            + ip-address 192.0.2.2;
          }
        }
      }
    }
  }
  +example-service s1 {
  + device [ mydevice0 ];
  + ipaddress 192.0.2.2;
  +}
}
}
admin@ncs(config-example-service-s1)# commit
Commit complete.
admin@ncs(config-example-service-s1)# do show running-config devices device mydevice0 config | display
```

mydevice0

FE1

192.0.2.2

verzoenen

Syntaxis: <path-to-service instantie> opnieuw implementeren, compatibel

Optionele vlaggen: { keep-non-service-config } dry-run { outformat native}

Doel

Het kerndoel van de compatibele functionaliteit is om gebruikers in staat te stellen zich te ontdoen van ongewenste eigendom van apparaten en eigendom volledig over te dragen aan de diensten. Wanneer gebruikers al een functionerend netwerk hebben en ze proberen eigendom over te dragen aan NSO, introduceren ze meestal eerst de configuratie via sync-vanaf-bewerkingen. Zodra CDB alle netwerkconfiguratie heeft, implementeert de gebruiker serviceinstanties bovenop de bestaande configuratie. Op dit punt, is de configuratie nog apparaat bezeten die de dienst het vermogen beperkt om configuratie te wissen. Wanneer gebruikers hun diensten volledig eigendom van de configuratie willen geven, kunnen ze de compatibele functionaliteit gebruiken die 3 dingen

doet.

- 1) 1) Overdracht van eigendom naar diensten
- 2) 2) Verwijder "original value"-vlaggen
- 3) 3) Correctie van de eigendom van de dienst

Eigendom van services overdragen

Reconcile evalueert alle configuratie die eigendom is van een dienst, en als het een configuratie vindt die eigendom is van zowel deze dienst als het apparaat of anderszins niet-service eigendom is, verwijdert het dit apparaat eigendom, waardoor de dienst de exclusieve eigenaar. Vermindering van het aantal terugbetalingen met 1.

Opmerking: Als 2 diensten een stuk van configuratie bezitten, en dat ook niet-dienst bezeten is, is de refcount 3. Als een van de twee diensten met elkaar in overeenstemming wordt gebracht, wordt dat niet-dienstbezit verwijderd, waardoor de vergoeding wordt teruggebracht tot 2 om beide diensten weer te geven.

Oorspronkelijke waardmarkeringen verwijderen

Wanneer een service wordt geïmplementeerd en gegevens die niet van de service zijn, worden hertellingen 2 en voegt NSO een "oorspronkelijke waarde"-tag toe. Als de service-instantie ooit wordt verwijderd, probeert NSO terug te keren naar de oorspronkelijke waarde die voor de service bestond.

Tijdens reconciliatie wordt niet alleen de refcount verlaagd, maar wordt ook de oorspronkelijke waarde verwijderd. Als u de service verwijdert, wordt de waarde nu leeg of wordt deze gewijzigd in een standaardwaarde zoals gedefinieerd in het yang-model.

De eigendom van de dienst corrigeren

In sommige situaties kan eigendom verkeerd worden toegewezen. Ofwel is de configuratie die eigendom is van het apparaat onjuist eigendom van de service, ofwel is de configuratie onjuist eigendom van zowel de service als het apparaat, terwijl verwacht wordt dat deze alleen eigendom is van de service. Reconcile kan deze verkeerde uitlijning corrigeren. Dit is belangrijk om problemen te voorkomen waarbij de service de configuratie die niet door de service wordt beheerd, verwijdert.

Reconcile tijdens herimplementatie

Reconcile is een subfunctie van een herimplementatie van de service. Als een service niet synchroon is met CDB, voert de bediening van de afstemming ook een herimplementatie uit naast het uitvoeren van de afstemmingsfunctie.

Innerlijke werkstukken met elkaar verzoenen

Hoewel de exacte details van hoe reconile werkt alleen bekend zijn bij de ontwikkelaars, biedt dit document een vereenvoudigd begrip:

- 1) NSO corrigeert eigendom van de dienst
- 2) NSO verwijdert alle configuratie die eigendom is van dit servicecentrum van CDB, zelfs als het ook eigendom is van andere diensten of van het apparaat
- 3) NSO stelt de service-instantie opnieuw in
- 4) NSO herstelt de dienstrekningen en backpointers van andere diensten

Problemen met serviceeigendom

Reconcile mislukt

Mocht u "hergroeperen" vinden werkt, maar "hergroeperen" mislukt: Dit kan erop wijzen dat u een conflict hebt ondervonden tussen hun service-ontwerp en de manier waarop de functie werkt.

Het probleem ontstaat door de servicecode die probeert configuratie te lezen van CDB die de dienst dan later implementeert. U kunt deze service alleen implementeren omdat de configuratie al gedeeltelijk bestaat in CDB voor de implementatie. Maar tijdens reconile verwijdert NSO tijdelijk alle configuratie die eigendom is van deze service, inclusief de configuratie die de service tijdens de volgende stap probeert te lezen. Dit resulteert gewoonlijk in een java of python fout die het nalaten om de gegevens te lezen melden.

Service verwijdert configuratie die geen eigendom is van de service

In dit scenario, ontmoet u NSO die niet-dienst bezeten configuratie tijdens de de dienstinstantie schrappen of opnieuw opstelt. Dit komt doordat de service-instantie de oorspronkelijke configuratie heeft gemaakt en eigendom is, en u later handmatig (via `ncs_cli`, `sync-van` of andere methode) een deel van de configuratie heeft toegevoegd in een container of lijst die eigendom is van de service.

Deze nieuwe configuratie wordt niet verondersteld het eigendom van de dienst te zijn, maar omdat de dienst volledig eigendom van de container of lijst heeft, eindigt de dienst indirect eigendom van het.

De manier om dit op te lossen is door hergroepering van `reconcile { keep-non-service-config }` om de diensteigendom te verbeteren. Wanneer het doen van dit verzoeken, stijgt de hertelling van de container of de lijst om erop te wijzen dat deze container of lijst zowel dienst-bezeten als niet-dienst bezeten kindknopen heeft. Binnen de parent-knooppunt hebben alleen de serviceknooppunten een `refcount` en `backpointer`.

Showcase: Service verwijdert configuratie die geen eigendom is van service

Beginnend met één enkele die de dienstinstantie met volledig eigendom wordt opgesteld, voeg

manueel een beschrijving aan de interface toe met ncs_cli.

```
admin@ncs(config-example-service-s1)# do show running-config devices device mydevice0 config | display
```

```
mydevice0
```

```
FE1
```

```
192.0.2.1
```

```
admin@ncs(config-example-service-s1)# top
```

```
admin@ncs(config)# devices device mydevice0 config interface FE1 description "This is an example descri
admin@ncs(config-interface-FE1)# commit
Commit complete.
admin@ncs(config-interface-FE1)# do show running-config devices device mydevice0 config | display servi
```

mydevice0

FE1

192.0.2.1

This is an example description

Merk op hoe de refcount op <interface> 1 blijft ondanks dat de device-owned configuratie is toegevoegd. Wanneer u de service-instantie probeert te verwijderen, wordt de beschrijving ook verwijderd, ook al is het niet de bedoeling dat deze deel uitmaakt van de service-instantie. Om dit te voorkomen, kunt u de opdracht Conconcile uitvoeren.

```
admin@ncs(config-interface-FE1)# top
admin@ncs(config)# no example-service s1
admin@ncs(config)# commit dry-run
cli {
  local-node {
    data devices {
      device mydevice0 {
        config {
          -       interface FE1 {
          -         ip-address 192.0.2.1;
          -         description "This is an example description";
          -       }
        }
      }
    }
  }
  -example-service s1 {
  -  device [ mydevice0 ];
  -  ipaddress 192.0.2.1;
  -}
}
}
admin@ncs(config)# abort
admin@ncs# config
Entering configuration mode terminal
admin@ncs(config)# example-service s1 re-deploy reconcile { keep-non-service-config }
admin@ncs(config)# do show running-config devices device mydevice0 config | display service-meta-data |
```

mydevice0

FE1

192.0.2.1

This is an example description

Na de afstemming is de refcount voor list interface verhoogd naar 2. Ondertussen is het aantal reacties op het ip-adres van het blad 1 gebleven. De lijstvermelding "interface FE1" bevat zowel gegevens die in eigendom zijn van de dienst als gegevens die niet in eigendom zijn van de dienst. Door reconile te gebruiken, herevalueert NSO de eigendom en kent dienovereenkomstig de opbrengsten toe. Schrapping richt zich nu alleen op de gebieden die volledig eigendom zijn van de service instantie. Noch de beschrijving noch de lijstvermelding wordt verwijderd.

```
admin@ncs(config)# no example-service s1
```

```

admin@ncs(config)# commit dry-run
cli {
  local-node {
    data devices {
      device mydevice0 {
        config {
          interface FE1 {
            - ip-address 192.0.2.1;
          }
        }
      }
    }
    -example-service s1 {
      - device [ mydevice0 ];
      - ipaddress 192.0.2.1;
      -}
  }
}

```

Keep-non-service-config vs verwerping-niet-service-configuratie

Gebruikers begrijpen soms verkeerd het gebruik van verwerping-niet-dienst-config.

In het verzoenende voorbeeld werd "keep-non-service-config" gebruikt. Indien afgedankte apparatuur wordt gebruikt, ziet het er als volgt uit:

```

admin@ncs(config)# example-service s1 re-deploy reconcile { discard-non-service-config } dry-run
cli {
  local-node {
    data devices {
      device mydevice0 {
        config {
          interface FE1 {
            - description "This is an example description";
          }
        }
      }
    }
  }
}

```

Het gebrek is "houd-niet-dienst-Config". Als geen van beide opties is gedefinieerd, blijft NSO standaard behouden. Verwerping wordt zelden gebruikt omdat de meeste gebruikers liever houden wat op hun netwerk is, zelfs als het niet wordt beheerd door NSO. Reconcile { discard-non-service-config } dry-run kan worden gebruikt om te weten te komen welke punten van gegevens er in CDB zijn die geen deel uitmaken van de service config, maar zou worden verwijderd als de service werd verwijderd of opnieuw ingezet.

Geen aanmaken

Een alternatief voor het gebruik van herimplementatie die geschikt is om service-eigendom te corrigeren in combinatie met gegevens die niet van de service afkomstig zijn, is het voorkomen van het conflict met behulp van de nocreation-tag.

Dit is een tag die kan worden gebruikt in uw XML-servicesjabloon. De documentatie verklaart "niet creëren: de fusie beïnvloedt slechts configuratiepunten die reeds in het malplaatje bestaan. Het maakt nooit de configuratie met deze tag. Het wijzigt alleen bestaande configuratiestructuren."

Het gebruik van deze tag heeft een interessant neveneffect: omdat de dienst de knoop niet maakt, neemt het geen eigendom van het.

Dit wordt vaak gebruikt om situaties te voorkomen waarin NSO probeert om configuratie te verwijderen die het apparaat niet kan worden verwijderd.

Merk op dat deze tag wordt geërfd door kindknooppunten, dit betekent dat als je de nocreatie tag aan de interface toevoegt, het ook van toepassing is op alle knooppunten in die interface, tenzij je ze markeert met een andere tag zoals samenvoegen"

Voeg een niet-gemaakte tag toe aan de servicessjabloon. Als de interface FE1 niet bestaat, wordt niets geconfigureerd.

```
{/device}
```

```
{/ipaddress}
```

Opnieuw compileren en opnieuw laden van de verpakking, dan testen.

```
admin@ncs(config)# do show running-config devices device mydevice0 config | display service-meta-data |
No entries found.
admin@ncs(config)# example-service s1 device mydevice0 ipaddress 192.0.2.1
admin@ncs(config-example-service-s1)# commit dry-run
cli {
  local-node {
    data +example-service s1 {
      +   device [ mydevice0 ];
      +   ipaddress 192.0.2.1;
      +}
    }
  }
}
```

Alhoewel de zelfde parameters zoals voordien werden bepaald, worden de interface of om het even welke onderliggende configuratie niet tot stand gebracht in de apparatenconfiguratie.

Samenvoegen

Voeg een merge tag toe op de configuratie binnen de interface. Voeg geen tag toe aan "interface-name", aangezien dit de sleutel is van de lijstinterface. De sleutel moet altijd het gedrag van de lijst kunnen erven. Opnieuw compileren en laden van de verpakking.

```
{/device}
```

```
FE1
```

```
{/ipaddress}
```

Configureer de interface FE1 handmatig voordat u de service implementeert.

```
admin@ncs(config)# no example-service  
admin@ncs(config)# commit  
admin@ncs(config)# do show running-config devices device mydevice0 config | display service-meta-data |
```

No entries found.

```
admin@ncs(config)# devices device mydevice0 config interface FE1
```

```
admin@ncs(config-interface-FE1)# commit dry-run
```

```
cli {
  local-node {
    data devices {
      device mydevice0 {
        config {
          +       interface FE1 {
          +       }
        }
      }
    }
  }
}
```

```
admin@ncs(config-interface-FE1)# commit
```

Commit complete.

```
admin@ncs(config-interface-FE1)# top
```

```
admin@ncs(config)# example-service s1 device mydevice0 ipaddress 192.0.2.1
```

```
admin@ncs(config-example-service-s1)# commit dry-run
```

```
cli {
  local-node {
    data devices {
      device mydevice0 {
        config {
          +       interface FE1 {
          +           ip-address 192.0.2.1;
          +       }
        }
      }
    }
    +example-service s1 {
    +  device [ mydevice0 ];
    +  ipaddress 192.0.2.1;
    +}
  }
}
```

```
admin@ncs(config-example-service-s1)# commit
```

Commit complete.

```
admin@ncs(config-example-service-s1)# do show running-config devices device mydevice0 config | display
```

mydevice0

FE1

192.0.2.1

Interface heeft een verborgen refcount 1: interface is geïmplementeerd met ncs_cli, maar heeft een nocreative tag in het servicepakket; de service heeft geen eigendom genomen. Het is apparaateigendom.

Primary heeft refcount 1: Het is alleen eigendom van de service

Als de service-instantie wordt verwijderd, zou het alleen ipaddress verwijderen omdat dat het enige deel is dat volledig eigendom is van de service.

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.