Comprendre la propriété des services NSO

Table des matières

- Introduction
- Conditions préalables
  - Exigences
  - Composants utilisés
- Notions de base
- Périphérique appartenant
- Propriété en action
  - Vitrine : Service uniquement possédé
  - Vitrine : Seul périphérique possédé
  - Vitrine : Périphérique possédé + Service possédé
- Réconcilier
  - Objectif
    - Transférer la propriété aux services
    - Supprimer les indicateurs de valeur d'origine
    - Correction de la propriété du service
  - Rapprocher comme redéploiement
  - Rapprocher le fonctionnement interne
- Problèmes liés à la propriété des services
  - La réconciliation échoue
  - Le service supprime la configuration non détenue par le service
  - Vitrine : Le service supprime la configuration non détenue par le service
  - Keep-non-service-config vs Discard-non-service-config
- Pas de création
  - Fusionner

Introduction

Ce document décrit le concept général, les pièges courants et les solutions de propriété de service dans Cisco® Network Service Orchestrator (NSO).

Conditions préalables

Exigences

Ce document s'applique à toutes les versions de NSO actuellement disponibles, y compris NSO 6. Le comportement décrit s'applique uniquement aux configurations NSO utilisant une combinaison de services et de configuration de non-service. Bien que les commandes spécifiques utilisées dans les exemples de ce document ne s'appliquent qu'au pilote d'élément réseau (NED) utilisé, la

logique sous-jacente s'applique à tout périphérique géré par NSO.

Composants utilisés

- NSO 6.1.6
- BESOIN : test-end 1.0, un NED netconf personnalisé utilisant ce modèle yang.

```
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 : exemple-service 1.0 : un service personnalisé utilisant ce modèle et ce modèle yang

```
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}
```

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. Si votre réseau est en ligne, assurez-vous de bien comprendre l’incidence possible des commandes.

Notions de base

L'objectif de la propriété du service est de permettre à NSO de suivre quelle configuration est associée à quel service. Lorsque vous supprimez un service, NSO doit supprimer la configuration qui lui est associée et utilise la propriété du service pour déterminer la configuration à supprimer. Lorsque la configuration appartient à plusieurs services, la suppression d'un des services supprime simplement cette référence de propriété ; La configuration elle-même reste dans la base de données NSO (CDB) et sur les périphériques du réseau.

La propriété est affichée par le biais de refcounts et de backpointers. Un refcount indique combien d'entités possèdent cette partie de la configuration. Un recomptage est égal à la quantité d'instances de service plus 1 si la configuration est également détenue par le périphérique. Un pointeur arrière indique le chemin de ces instances de service. Il n'y a pas de pointeur arrière pour afficher "périphérique possédé". Les backpointers ne sont affichés dans CDB que pour les listes et les conteneurs. Les leafs individuels n'affichent pas leurs pointeurs arrière, mais ils héritent de leurs parents.

Périphérique appartenant

En plus de la configuration détenue par un service, elle peut également être détenue par le périphérique. On parle parfois de « périphérique détenu » ou de « non-service détenu ». Bien que ce document utilise le terme « appartenant à un périphérique », notez que bien que cela soit plus facile à comprendre, la non-propriété de service n'inclut pas nécessairement les périphériques. Les configurations LSA ou les services empilés peuvent avoir une configuration non détenue par le service sans périphériques.

La configuration appartient à un périphérique lorsque la configuration est ajoutée à CDB sans utiliser de déploiement de service, mais qu'elle utilise une méthode telle que sync-from, load merge ou ncs_cli pour définir la configuration. Lorsqu'une instance de service prend possession d'une configuration qui appartenait déjà à un périphérique, le refcount est défini sur 2 pour refléter la propriété partagée. Lorsque l'instance de service est supprimée, la configuration n'est pas supprimée malgré le fait qu'une seule instance de service possède la configuration avant la suppression. En outre, la configuration appartenant au périphérique ajoute une balise « valeur d'origine ». Si une instance de service écrase la configuration appartenant au périphérique et que le service est supprimé ultérieurement, la configuration reprend sa valeur d'origine.

La propriété du périphérique n'est attribuée que si la configuration n'existait pas déjà dans CDB lorsque vous l'ajoutez par des moyens autres que des services. La configuration détenue par le service ne devient pas détenue par le périphérique après la synchronisation à partir de. Cependant, la configuration appartenant à un périphérique devient à la fois une configuration appartenant à un périphérique et une configuration appartenant à un service si vous déployez un service au-dessus.

Propriété en action

Vitrine : Service uniquement possédé

Lorsque vous déployez un service alors que la configuration cible est vide, le service crée la configuration et en prend possession. L'utilisateur peut vérifier la propriété à l'aide d'une commande show running-config et ajouter | afficher les métadonnées de service. Bien que cela ne soit pas obligatoire, il est recommandé d'ajouter également | affiche xml comme résultat de style CLI par défaut qui ne reflète pas toujours correctement la façon dont les données sont modélisées dans 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
```

En outre, si vous ajoutez une seconde instance de service ciblant la même configuration, la propriété est partagée par les 2 instances de service. Le nombre de retraits est égal à 2 et il y a 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
```

Vitrine : Seul périphérique possédé

Lorsque vous ajoutez des données à CDB à l'aide de load merge, ncs_cli ou sync-from, sans utiliser de service, ces données deviennent la propriété d'un périphérique. Les options refcount et backpointer sont masquées.

```
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
```

Vitrine : Périphérique possédé + Service possédé

Cet exemple montre comment créer facilement une propriété combinée de périphérique et de service à l'aide d'un service et d'une synchronisation.

Vous déployez le service, puis vous le supprimez uniquement dans CDB en utilisant Commit no-networking. De cette façon, la configuration existe toujours sur le périphérique final. Lorsque vous effectuez une synchronisation à partir de, la configuration est rajoutée dans CDB, mais elle appartient au périphérique et non au service. N'oubliez pas que show running-config lorsqu'il est exécuté dans NSO affiche les données CDB, et non les données actuellement sur les périphériques.

```
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
```

Après la synchronisation à partir de, la configuration appartient uniquement au périphérique. Refcounter est masqué. Lors d'un nouveau déploiement du service, le refcount devient 2, mais backpointer n'affiche qu'une seule instance de service. Le deuxième compteur représente la propriété du périphérique. Les mêmes règles s'appliquent qu'avec la propriété de service partagée. Si le service est supprimé, la configuration n'est pas supprimée car le périphérique possède également partiellement la configuration. En outre, si les données de service ne correspondent pas à la « valeur d'origine » telle que stockée dans les métadonnées de service, NSO rétablit la valeur « valeur d'origine » si le service est supprimé.

```
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
```

Réconcilier

Syntaxe: <path-to-service instance> redéploiement réconciliation

Indicateurs facultatifs : { keep-non-service-config } dry-run { outformat native }

Objectif

L'objectif principal de la fonctionnalité de réconciliation est de permettre aux utilisateurs de se débarrasser de la propriété indésirable des périphériques et de transférer la propriété entièrement aux services. Lorsque les utilisateurs disposent déjà d'un réseau opérationnel et qu'ils tentent de transférer la propriété à NSO, ils introduisent généralement d'abord la configuration par le biais d'opérations de synchronisation. Une fois que CDB dispose de toute la configuration réseau, l'utilisateur déploie des instances de service au-dessus de la configuration existante. À ce stade, la configuration appartient toujours au périphérique, ce qui limite la capacité du service à supprimer la configuration. Lorsque les utilisateurs souhaitent donner à leurs services la pleine propriété de la configuration, ils peuvent utiliser la fonctionnalité de réconciliation qui effectue 3 choses.

1) 1) Transfert de propriété aux services

2) 2) Supprimer les indicateurs de « valeur d'origine »

3) 3) Correction de la propriété des services

Transférer la propriété aux services

Reconcile évalue toute la configuration appartenant à un service et, s'il trouve une configuration appartenant à ce service et au périphérique ou n'appartenant pas au service, il supprime la propriété de ce périphérique, faisant du service le propriétaire exclusif. Réduction du recomptage de 1.

Remarque : Si deux services possèdent une configuration et que cette configuration est également détenue par des non-services, le recomptage est égal à 3. Le rapprochement de l'un des services supprime cette propriété de non-service, réduisant le recomptage à 2 pour refléter les deux services.

Supprimer les indicateurs de valeur d'origine

Lorsqu'un service est déployé et écrase des données non détenues par le service, refcount devient 2 et NSO ajoute une balise « valeur d'origine ». Si l'instance de service est supprimée, NSO tente de revenir à la valeur d'origine qui existait avant le service.

Lors du rapprochement, non seulement le recomptage est réduit, mais la valeur d'origine est supprimée. La suppression du service rend cette valeur vide ou la remplace par une valeur par défaut telle que définie par le modèle yang.

Correction de la propriété du service

Dans certains cas, la propriété peut être incorrectement attribuée. Soit la configuration appartenant à un périphérique est incorrectement détenue par le service, soit la configuration est incorrectement détenue à la fois par le service et le périphérique, alors qu'elle devrait être détenue uniquement par le service. Reconcile peut corriger ces erreurs d'alignement. Ceci est important pour éviter les problèmes où le service supprime la configuration non détenue par le service.

Rapprocher comme redéploiement

La réconciliation est une sous-fonction du redéploiement de service. Si un service n'est pas synchronisé avec CDB, l'opération de réconciliation exécute également un redéploiement en plus d'exécuter la fonction de réconciliation.

Rapprocher le fonctionnement interne

Bien que seuls les développeurs connaissent les détails exacts du fonctionnement de la réconciliation, ce document fournit une compréhension simplifiée :

1) NSO Corrige la propriété du service

2) NSO supprime de CDB toutes les configurations appartenant à cette instance de service, même si elles appartiennent également à d'autres services ou à des périphériques

3) NSO redéploie l'instance de service

4) NSO restaure les recomptes de services et les backpointers d'autres services

Problèmes liés à la propriété des services

La réconciliation échoue

Si vous trouvez que le redéploiement fonctionne, mais que le redéploiement et la réconciliation échouent : Cela peut indiquer que vous avez rencontré un conflit entre leur conception de service et le fonctionnement de la fonction de réconciliation.

Le problème provient du code de service qui tente de lire la configuration à partir de CDB que le service déploie ensuite. Vous ne pouvez déployer ce service que parce que la configuration existe déjà partiellement dans CDB avant le déploiement. Cependant, lors de la réconciliation, NSO supprime temporairement toute la configuration appartenant à ce service, y compris la configuration que le service tente de lire lors du redéploiement du service à l'étape suivante. Cela se traduit généralement par une erreur java ou python signalant l'échec de lecture des données.

Le service supprime la configuration non détenue par le service

Dans ce scénario, vous rencontrez NSO supprimant une configuration non détenue par le service pendant la suppression ou le redéploiement de l'instance de service. En effet, l'instance de service

a créé et possède la configuration d'origine, et vous avez ensuite ajouté manuellement (par le biais de ncs_cli, sync-from ou d'une autre méthode) une partie de la configuration dans un conteneur ou une liste appartenant à un service.

Ce nouvel élément de configuration n'est pas censé appartenir au service, mais étant donné que le service possède la propriété totale du conteneur ou de la liste, le service finit par en être indirectement propriétaire.

Pour résoudre ce problème, utilisez la commande re-deploy concilier { keep-non-service-config } pour corriger la propriété du service. Lors de ce rapprochement, le recomptage du conteneur ou de la liste augmente pour refléter le fait que ce conteneur ou cette liste possède des noeuds enfants appartenant ou non à un service. À l'intérieur du noeud parent, seuls les noeuds appartenant au service ont un refcount et un backpointer.

Vitrine : Le service supprime la configuration non détenue par le service

À partir d'une seule instance de service déployée avec la propriété complète, ajoutez manuellement une description à l'interface à l'aide de ncs_cli.

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

```
          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
```

Notez que le recomptage sur <interface> reste 1 même si la configuration appartenant au périphérique a été ajoutée. Lorsque vous tentez de supprimer l'instance de service, la description est également supprimée même si elle n'est pas censée faire partie de l'instance de service. Pour éviter cela, vous pouvez exécuter la commande concile.

```
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
```

Après le rapprochement, le recomptage pour l'interface de liste est passé à 2. Pendant ce temps, le recomptage sur l'adresse IP de feuille est resté 1. L'entrée de liste « interface FE1 » contient à la fois des données appartenant au service et des données non appartenant au service. En utilisant la fonction de rapprochement, l'ONS réévalue la propriété et attribue les comptes en conséquence. La suppression ne cible désormais que les zones qui appartiennent entièrement à l'instance de service. Ni la description ni l'entrée de liste ne sont supprimées.

```
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 Discard-non-service-config

Les utilisateurs comprennent parfois mal l'utilisation de discard-non-service-config.

Dans l'exemple de réconciliation, « keep-non-service-config » a été utilisé. Si la commande discard est utilisée, elle ressemble à ceci :

```
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";
                            }
                        }
                    }
                }

    }
}
```

La valeur par défaut est « keep-non-service-config ». Si aucune option n'est définie, NSO conserve par défaut. La suppression est rarement utilisée car la plupart des utilisateurs préfèrent conserver ce qui se trouve sur leur réseau, même s'il n'est pas géré par NSO. Reconcile { discard-non-service-config } dry-run peut être utilisé pour découvrir quels points de données existent dans CDB qui ne font pas partie de la configuration du service mais qui seraient supprimés si le service était supprimé ou redéployé.

Pas de création

Une alternative à l'utilisation de la fonction de réconciliation de redéploiement pour corriger la propriété du service lorsqu'elle est associée à des données non détenues par le service consiste à

empêcher le conflit à l'aide de la balise nocreate.

Il s'agit d'une balise qui peut être utilisée dans votre modèle de service XML. La documentation indique « nocreate : la fusion affecte uniquement les éléments de configuration qui existent déjà dans le modèle. Il ne crée jamais la configuration avec cette balise. Elle modifie uniquement les structures de configuration existantes. »

L'utilisation de cette balise a un effet secondaire intéressant : Comme le service ne crée pas le noeud, il n'en devient pas propriétaire.

Il est généralement utilisé pour empêcher les situations où NSO tente de supprimer une configuration que le périphérique ne permet pas de supprimer.

Notez que cette balise est héritée par les noeuds enfants, ce qui signifie que si vous ajoutez la balise nocreate à l'interface, elle s'applique également à tous les noeuds à l'intérieur de cette interface, sauf si vous les marquez avec une autre balise telle que merge"

Ajoutez une balise nocreate au modèle de service. Si l'interface FE1 n'existe pas, rien n'est configuré.

```
{/device}

FE1

{/ipaddress}
```

Recompilez et rechargez le package, puis testez.

```
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;
             +}
    }
}
```

Même si les mêmes paramètres ont été définis comme précédemment, l'interface ou toute configuration sous-jacente ne sont pas créées dans la configuration du périphérique.

Fusionner

Ajoutez une balise merge sur la configuration à l'intérieur de l'interface. N'ajoutez pas de balise à « interface-name », car il s'agit de la clé de l'interface de liste. La clé doit toujours être autorisée à hériter du comportement de la liste. Recompilez et rechargez le package.

```
          {/device}
```

```
            FE1
```

```
            {/ipaddress}
```

Configurez manuellement l'interface FE1 avant de déployer le service.

```
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 s
```

```
          mydevice0
```

```
            FE1
```

```
            192.0.2.1
```

L'interface a un refcount 1 masqué : l'interface a été déployée à l'aide de ncs_cli, mais elle a une balise nocreate dans le package de services ; le service ne s'est pas approprié. Il appartient à un périphérique.

Le principal a refcount 1 : Il est uniquement détenu par le service

Si l'instance de service est supprimée, elle ne supprimera que ipaddress car c'est la seule partie qui est entièrement détenue par le service.

À propos de cette traduction
Cisco a traduit ce document en traduction automatisée vérifiée par une personne dans le cadre d'un service mondial permettant à nos utilisateurs d'obtenir le contenu d'assistance dans leur propre langue.

Il convient cependant de noter que même la meilleure traduction automatisée ne sera pas aussi précise que celle fournie par un traducteur professionnel.