

瞭解NSO服務所有權

目錄

[簡介](#)

[必要條件](#)

[需求](#)

[採用元件](#)

[基礎知識](#)

[裝置所有](#)

[行動中的所有權](#)

[展示：僅服務所有](#)

[展示：僅裝置擁有](#)

[展示：裝置所有+服務所有](#)

[協調](#)

[目的](#)

[將所有權轉移至服務](#)

[刪除原始值標誌](#)

[更正服務所有權](#)

[協調為重新部署](#)

[協調內部工作](#)

[涉及服務所有權的問題](#)

[協調失敗](#)

[服務正在刪除非服務擁有的配置](#)

[展示：服務正在刪除非服務擁有的配置](#)

[Keep-non-service-config vs Discard-non-service-config](#)

[不建立](#)

[合併](#)

簡介

本檔案介紹Cisco® Network Service Orchestrator(NSO)中服務所有權的一般概念、常見缺陷和解決方案。

必要條件

需求

本文檔適用於當前所有可用的NSO版本，包括NSO 6。所描述的行為僅適用於使用服務和非服務配置組合的NSO設定。雖然本文檔示例中使用的特定命令僅適用於使用的網元驅動程式(NED)，但是底層邏輯適用於NSO管理的任何裝置。

採用元件

- NSO 6.1.6
- NED：測試版1.0，使用此yang模型定製的netconf NED。

```
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;
    }
  }
}
```

- 服務：example-service 1.0：使用此yang模型和模板的定製服務

```
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}

本文中的資訊是根據特定實驗室環境內的裝置所建立。文中使用到的所有裝置皆從已清除（預設）的組態來啟動。如果您的網路運作中，請確保您瞭解任何指令可能造成的影響。

基礎知識

服務所有權的目的是允許NSO跟蹤與哪個服務相關的配置。刪除服務時，NSO需要刪除與其相關的配置，它使用服務所有權來確定要刪除的配置。當配置屬於多個服務時，刪除其中一個服務只會刪除該所有權引用；配置本身仍保留在NSO資料庫(CDB)中以及網路裝置上。

所有權通過重新計數和反向指標顯示。引用顯示有多少圖元擁有該配置部分。如果配置也歸裝置所有，則重新計數等於服務例項數加1。反向指標顯示這些服務例項的路徑。沒有用於顯示「裝置所有」的反向指標。只有清單和容器的CDB中才顯示反向指標。單個枝葉不會顯示其反向指標，但會繼承其父項。

裝置所有

除了配置由服務擁有外，它還可以由裝置擁有。這有時稱為「裝置擁有」或「非服務擁有」。雖然本文檔使用「裝置所有」，但請注意，雖然這更易於理解，但非服務所有權不一定包括裝置。LSA設定或堆疊服務可以具有無裝置的非服務擁有的配置。

將配置新增到CDB而不使用服務部署時，配置由裝置所有，而是使用同步自、載入合併或`ncs_cli`等方法來設定配置。當服務例項擁有已歸裝置所有的配置時，`refcount`設定為2以反映共用所有權。刪除服務例項時，不會刪除該配置，儘管在刪除之前只有一個服務例項擁有該配置。此外，裝置擁有的配置會新增「原始值」標籤。如果服務例項覆蓋裝置擁有的配置並在以後刪除該服務，則配置將恢復為原始值。

僅當通過非服務方式新增配置時，CDB中不存在該配置時，才會分配裝置所有權。從同步後，服務擁有的配置不會成為裝置擁有的配置。但是，如果在頂部部署服務，則裝置擁有的配置將變為裝置擁有和服務擁有。

行動中的所有權

展示：僅服務所有

當目標配置為空時部署服務時，服務將建立配置並獲取其所有權。使用者可以使用`show running-config`命令並追加來檢查所有權 | 顯示服務後設資料。雖然不是強制性的，但建議也附加 | 將xml顯示為預設CLI樣式輸出並不總是正確反映資料在CDB中的建模方式。

```
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

此外，如果新增第二個服務例項以相同配置為目標，則兩個服務例項將共用所有權。RefCount為2，且有2個反指標。

```
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

展示：僅裝置擁有

在不使用服務的情況下，使用負載合併、ncs_cli或sync-from向CDB新增資料時，此資料將成為裝置擁有的資料。refcount和backpointer被隱藏。

```
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

展示：裝置所有+服務所有

此示例演示如何使用服務和同步來源輕鬆建立合併的裝置和服務所有權。

您部署該服務，然後僅使用Commit no-networking在CDB中刪除該服務。如此一來，組態仍會存在於終端裝置上。執行同步時，配置會新增回CDB中，但它是裝置擁有的，而不是服務擁有的。請記住，在NSO中運行時的show running-config會顯示CDB資料，而不是裝置上的當前資料。

```
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

從同步後，配置僅歸裝置所有。Refcounter已隱藏。再次部署服務時，refcount將變為2，但反向指標僅顯示單個服務例項。第二個反射計數器表示裝置所有權。與共用服務所有權相同的規則同樣適用，如果服務被刪除，配置不會被刪除，因為裝置還部分擁有配置。此外，如果服務資料與服務後設資料中儲存的「原始值」不匹配，則在刪除服務時，NSO會將值恢復為「原始值」。

```
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

協調

語法 : <path-to-service instance> re-deploy reconcile

可選標誌:{ keep-non-service-config } dry-run { outformat native }

目的

協調功能的核心目的是允許使用者擺脫不需要的裝置所有權，並將所有權完全轉移給服務。當使用者已經擁有正常運行的網路並嘗試將所有權轉移到NSO時，他們通常會首先通過同步起始操作引入配置。一旦CDB具有所有網路配置，使用者就會在現有配置之上部署服務例項。此時，配置仍歸裝置所有，這限制了服務刪除配置的能力。當使用者希望讓其服務完全擁有配置時，他們可以使用協調功能，該功能可執行3項操作。

- 1)將所有權轉移至服務
- 2)刪除「原始值」標誌
- 3)糾正服務所有權

將所有權轉移至服務

Reconcile將評估服務擁有的所有配置，如果它發現任何配置歸該服務和裝置所有，或者為非服務所有，則會刪除此裝置所有權，從而使服務成為獨佔所有者。將重新計數減少1。

注意：如果2個服務擁有部分配置，並且該配置也是非服務擁有的，則refcount為3。對其中任何一個服務進行協調會移除該非服務所有權，將重新計數減少到2以反映這兩種服務。

刪除原始值標誌

部署服務並覆蓋非服務擁有的資料時，refcount將變為2，而NSO將新增「原始值」標籤。如果服務例項被刪除，NSO會嘗試恢復為服務之前存在的原始值。

在協調過程中，不僅會減少引用計數，而且會刪除原始值。現在刪除服務會使該值為空，或者將其更改為由yang模型定義的預設值。

更正服務所有權

在某些情況下，所有權可能分配不正確。裝置擁有配置不正確歸服務所有，或者配置不正確歸服務和裝置所有，而預期只有服務擁有。協調可以糾正這些不對齊。這對於避免服務刪除非服務擁有配置的問題非常重要。

協調為重新部署

協調是重新部署服務的子功能。如果服務與CDB不同步，協調操作除了執行協調功能之外還執行重新部署。

協調內部工作

雖然只有開發人員才能知道協調操作的確切細節，但本文檔提供了簡化的理解：

- 1)NSO糾正服務所有權

2)NSO從CDB中刪除此服務例項所擁有的所有配置，即使它也是其他服務所擁有或裝置所擁有

3)NSO重新部署服務例項

4)NSO從其他服務恢復服務反射和反向指標

涉及服務所有權的問題

協調失敗

如果您發現「重新部署」工作正常但「重新部署協調」失敗：這可能表示您的服務設計和協調功能運行方式之間發生了衝突。

問題源於服務代碼，該代碼嘗試從隨後服務部署的CDB讀取配置。您只能部署此服務，因為在部署之前CDB中已部分存在此配置。但在協調期間，NSO會臨時刪除此服務擁有的所有配置，包括在下一步重新部署服務期間服務嘗試讀取的配置。這通常導致報告讀取資料失敗的java或python錯誤。

服務正在刪除非服務擁有的配置

在此場景中，在刪除或重新部署服務例項期間，您會遇到NSO刪除非服務擁有的配置。這是因為服務例項已建立並擁有原始配置，並且您以後需要手動操作（通過ncs_cli、sync-from或其他方法）將配置的一部分新增到服務擁有的容器或清單中。

此新配置不應由服務擁有，但由於服務擁有容器或清單的完全所有權，因此服務最終間接擁有該容器。

解決此問題的方法是使用re-deploy reconcile { keep-non-service-config }來更正服務所有權。進行此協調時，容器或清單的重新計數將增加，以反映此容器或清單同時具有服務所有子節點和非服務擁有的子節點。在父節點內，只有服務擁有的節點具有重新計數和反向指標。

展示：服務正在刪除非服務擁有的配置

從部署了完全所有權的單個服務例項開始，使用ncs_cli手動向介面新增描述。

```
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

請注意，即使新增了裝置擁有的配置，<interface>上的引用仍保持1。嘗試刪除服務例項時，即使描述不應該是服務例項的一部分，描述也會被刪除。要避免這種情況，可以運行reconcile命令。

```
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

協調後，list介面的refcount增加到2。同時，對枝葉ip-address的引用仍為1。清單條目「interface FE1」包含服務擁有資料和非服務擁有資料。通過使用協調，NSO將重新評估所有權並相應地分配重新計數。現在，刪除僅針對完全歸服務例項所有的區域。說明或清單條目都不會被刪除。

```
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

使用者有時會誤解使用discard-non-service-config。

在協調示例中，使用了「keep-non-service-config」。如果使用discard，則如下所示：

```
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";
          }
        }
      }
    }
  }
}
```

預設值為「keep-non-service-config」。如果兩個選項均未定義，則NSO預設值將保留。Discard很少被使用，因為大多數使用者希望保留其網路上的內容，即使它不是由NSO管理的。協調{ discard-non-service-config } dry-run可用於發現CDB中存在不是服務配置的一部分但在該服務被刪除或重新部署時將被刪除的資料點。

不建立

使用re-deploy reconcile來更正與非服務擁有資料混合時的服務擁有權的替代方法是使用ncreate標籤來防止衝突。

這是一個可以在XML服務模板中使用的標籤。文檔說明「ncreate：合併僅影響模板中已存在的配置項。它從不使用此標籤建立配置。它只修改現有的配置結構。」

使用此標籤會產生有趣的副作用：因為服務不建立節點，所以它不擁有節點的所有權。

這通常用於防止NSO嘗試刪除裝置不允許刪除的配置的情況。

請注意，此標籤由子節點繼承，這意味著如果將ncreate標籤新增到介面，則它也會應用於該介面內的任何節點，除非使用其他標籤（例如merge）來標籤它們

向服務模板新增ncreate標籤。如果介面FE1不存在，則不配置任何內容。

```
{/device}
```

```
FE1
```

```
{/ipaddress}
```

重新編譯並重新載入軟體包，然後進行測試。

```
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;
```

```
    +}  
  }  
}
```

即使定義了與以前相同的引數，也不會在裝置配置中建立介面或任何基礎配置。

合併

在介面內部的配置上新增合併標籤。不要向「interface-name」新增標籤，因為這是清單介面的鍵。必須始終允許金鑰繼承清單的行為。重新編譯並重新載入軟體包。

```
{/device}
```

```
FE1
```

```
{/ipaddress}
```

在部署服務之前手動配置介面FE1。

```
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

介面具有隱藏的refcount 1：介面是使用ncs_cli部署的，但它在服務包中有一個nocreate標籤；服務沒有取得所有權。它是裝置擁有的。

主要具有refcount 1：它僅由服務所有

如果刪除服務例項，則只會刪除ipaddress，因為它是唯一完全由服務擁有的部分。

關於此翻譯

思科已使用電腦和人工技術翻譯本文件，讓全世界的使用者能夠以自己的語言理解支援內容。請注意，即使是最佳機器翻譯，也不如專業譯者翻譯的內容準確。Cisco Systems, Inc. 對這些翻譯的準確度概不負責，並建議一律查看原始英文文件（提供連結）。