

NSO 서비스 소유권 이해

목차

[소개](#)

[사전 요구 사항](#)

[요구 사항](#)

[사용되는 구성 요소](#)

[기본](#)

[디바이스 소유](#)

[영업 소유권](#)

[Showcase: 서비스 소유만](#)

[Showcase: 디바이스 소유만](#)

[쇼케이스: 디바이스 소유 + 서비스 소유](#)

[조정](#)

[목적](#)

[서비스로 소유권 이전](#)

[원래 값 플래그 제거](#)

[서비스 소유권 수정](#)

[재배포로 조정](#)

[내부 작업 조정](#)

[서비스 소유권 관련 문제](#)

[조정 실패](#)

[서비스에서 서비스 외 소유 구성을 삭제하고 있습니다.](#)

[Showcase: 서비스에서 비서비스 소유 컨피그레이션을 삭제하고 있습니다.](#)

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

[생성 안 함](#)

[병합](#)

소개

이 문서에서는 Cisco® NSO(Network Service Orchestrator)의 일반적인 개념, 일반적인 위험 요소 및 서비스 소유권 솔루션에 대해 설명합니다.

사전 요구 사항

요구 사항

이 문서는 NSO 6을 포함하여 현재 사용 가능한 모든 NSO 버전에 적용됩니다. 서비스 및 비 서비스 컨피그레이션의 조합을 사용하는 NSO 설정에만 적용되는 동작이 설명되어 있습니다. 이 문서 전체의 예제에서 사용되는 특정 명령은 사용된 NED(Network Element Driver)에만 적용되지만, 기본 로직은 NSO에서 관리하는 모든 장치에 적용됩니다.

사용되는 구성 요소

- NSO 6.1.6
- NED: 이 yang 모델을 사용하는 사용자 지정 Built Netconf NED인 테스트용 NED 1.0입니다.

```
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(서비스): example-service 1.0: 이 yang 모델 및 템플릿을 사용하는 맞춤형 구축 서비스

```
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}

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다. 이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 현재 네트워크가 작동 중인 경우 모든 명령의 잠재적인 영향을 미리 숙지하시기 바랍니다.

기본

서비스 소유권의 목적은 NSO가 어떤 컨피그레이션과 어떤 서비스가 관련되어 있는지 추적할 수 있게 하는 것입니다. 서비스를 삭제할 때 NSO는 관련된 컨피그레이션을 삭제해야 하며 삭제할 컨피그레이션을 결정하는 데 서비스 소유권을 사용합니다. 컨피그레이션이 둘 이상의 서비스에서 소유된 경우 서비스 중 하나를 삭제하면 해당 소유권 참조가 제거됩니다. 컨피그레이션 자체는 NSO 데이터베이스(CDB)와 네트워크의 디바이스에 남아 있습니다.

소유권은 리카운트 및 백포인터를 통해 표시됩니다. refcount는 컨피그레이션의 해당 부분을 소유한 엔터티의 수를 표시합니다. refcount는 서비스 인스턴스의 양에 1을 더한 값과 같습니다(컨피그레이션이 디바이스 소유인 경우). 백포인터는 이러한 서비스 인스턴스의 경로를 표시합니다. "장치 소유"를 표시할 백포인터가 없습니다. 백포인터는 목록 및 컨테이너에 대한 CDB에만 표시됩니다. 개별 leaf는 백포인트를 표시하지 않지만 부모로부터 상속합니다.

디바이스 소유

서비스가 컨피그레이션을 소유하는 것 외에도 디바이스에서 이를 소유할 수 있습니다. 이를 "디바이스 소유" 또는 "비서비스 소유"라고도 합니다. 이 문서에서는 "디바이스 소유"를 사용하지만, 이는 이해하기 쉬우나 서비스 외 소유권에는 디바이스가 포함되지 않아도 됩니다. LSA 설정 또는 스택킹된 서비스는 디바이스 없이 비서비스 소유 컨피그레이션을 가질 수 있습니다.

컨피그레이션은 서비스 구축을 사용하지 않고 CDB에 컨피그레이션을 추가할 때 디바이스 소유이지만, 대신 sync-from, load merge 또는 ncs_cli와 같은 방법을 사용하여 컨피그레이션을 설정했습니다. 서비스 인스턴스가 이미 디바이스 소유였던 컨피그레이션의 소유권을 가져오면 refcount는 공유 소유권을 반영하도록 2로 설정됩니다. 서비스 인스턴스가 삭제되면 삭제되기 전에 서비스 인스턴스 중 하나만 컨피그레이션을 소유함에도 불구하고 컨피그레이션은 삭제되지 않습니다. 또한 디바이스 소유 컨피그레이션에는 "original value" 태그가 추가됩니다. 서비스 인스턴스가 디바이스 소유 컨피그레이션을 덮어쓰고 서비스가 나중에 삭제되면 컨피그레이션이 원래 값으로 복원됩니다.

서비스 이외의 수단을 통해 추가할 때 CDB에 컨피그레이션이 존재하지 않는 경우에만 디바이스 소유권이 할당됩니다. 서비스 소유 컨피그레이션은 동기화 시작 후에 디바이스 소유가 되지 않습니다. 그러나 위에서 서비스를 구축할 경우 디바이스 소유 컨피그레이션은 디바이스 소유 컨피그레이션과 서비스 소유 컨피그레이션이 됩니다.

영업 소유권

Showcase: 서비스 소유만

대상 컨피그레이션이 비어 있는 상태에서 서비스를 구축하는 경우, 서비스는 컨피그레이션을 생성하고 그 소유권을 갖습니다. 사용자는 show running-config 명령을 사용하여 소유권을 확인하고 추가할 수 있습니다 | service-meta-data를 표시합니다. 필수는 아니지만 추가하는 것이 좋습니다 | xml을 기본 CLI 스타일 출력으로 표시해도 데이터가 CDB에서 모델링되는 방식이 항상 올바르게 반영되지는 않습니다.

```
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

또한 동일한 컨피그레이션을 대상으로 하는 두 번째 서비스 인스턴스를 추가할 경우 소유권은 두 서비스 인스턴스에서 공유됩니다. Refcount는 2이고 백포인터는 2개입니다.

```
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: 디바이스 소유만

서비스를 사용하지 않고 로드 병합, ncs_cli 또는 sync-from을 사용하여 CDB에 데이터를 추가하면

이 데이터는 디바이스 소유가 됩니다. refcount와 백포인터가 숨겨져 있습니다.

```
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

쇼케이스: 디바이스 소유 + 서비스 소유

이 예에서는 서비스 및 sync-from을 사용하여 결합된 디바이스 및 서비스 소유권을 쉽게 만드는 방법을 보여 줍니다.

서비스를 구축한 다음 Commit no-networking을 사용하여 CDB에서만 서비스를 삭제합니다. 이렇게 하면 컨피그레이션이 엔드 디바이스에 계속 존재합니다. sync-from을 수행하면 컨피그레이션이 CDB에 다시 추가되지만 서비스 소유가 아닌 디바이스 소유가 됩니다. NSO에서 실행할 때 show running-config는 현재 디바이스에 있는 데이터가 아니라 CDB 데이터를 보여 줍니다.

```
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

동기화 시작 후에는 컨피그레이션이 디바이스에서만 소유됩니다. Refcounter가 숨겨져 있습니다. 서비스를 다시 구축할 때 refcount는 2가 되지만 백포인터는 단일 서비스 인스턴스만 표시합니다. 두 번째 refcounter는 디바이스 소유권을 나타냅니다. 공유 서비스 소유자와 동일한 규칙이 적용되며, 서비스가 삭제될 경우 디바이스에서도 컨피그레이션을 부분적으로 소유하므로 컨피그레이션이 제거되지 않습니다. 또한 서비스 데이터가 서비스 메타데이터에 저장된 "original-value"와 일치하지 않으면 서비스가 제거되면 NSO는 값을 "original-value"로 되돌립니다.

```
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

조정

구문: <path-to-service instance> redeploy reconcile

선택적 플래그: { keep-non-service-config } dry-run { outformat native }

목적

조정 기능의 핵심 목적은 사용자가 원치 않는 장치 소유권을 제거하고 서비스에 대한 소유권을 완전히 이전할 수 있도록 하는 것입니다. 사용자가 이미 작동 중인 네트워크를 가지고 있고 소유권을 NSO로 이전하려고 할 경우, 일반적으로 동기화에서 운영을 통해 컨피그레이션을 먼저 도입합니다. CDB에 모든 네트워크 컨피그레이션이 있으면 사용자는 기존 컨피그레이션 위에 서비스 인스턴스를 구축합니다. 이 시점에서 컨피그레이션은 여전히 디바이스 소유이며, 이는 서비스 컨피그레이션 삭제 기능을 제한합니다. 사용자가 서비스에 구성에 대한 전체 소유권을 부여하려면 조정 기능을 사용하여 3가지 작업을 수행할 수 있습니다.

- 1) 1) 소유권을 서비스로 이전
- 2) 2) "original value" 플래그 제거
- 3) 3) 서비스 소유권 수정

서비스로 소유권 이전

조정은 서비스가 소유한 모든 구성을 평가하며, 이 서비스와 디바이스가 모두 소유하거나 서비스가 아닌 다른 소유의 구성을 찾으면 이 장치 소유권을 제거하여 서비스를 단독 소유자로 만듭니다. refcount를 1만큼 줄입니다.

참고: 2개의 서비스가 하나의 구성을 소유하고 해당 구성도 비서비스 소유인 경우 refcount는 3입니다. 두 서비스 중 하나를 조정하면 해당 비 서비스 소유권이 제거되므로 두 서비스를 모두 반영하기 위해 참조 횟수가 2로 줄어듭니다.

원래 값 플래그 제거

서비스가 배포되고 비 서비스 소유 데이터를 덮어쓰면 refcount는 2가 되고 NSO는 "original value" 태그를 추가합니다. 서비스 인스턴스가 삭제되면 NSO는 서비스 이전에 있던 원래 값으로 되돌리려고 시도합니다.

조정 중에 refcount가 감소할 뿐만 아니라 원래 값이 제거됩니다. 이제 서비스를 삭제하면 해당 값이 비어 있거나 yang 모델에 정의된 대로 기본값으로 변경됩니다.

서비스 소유권 수정

경우에 따라 소유권이 잘못 할당될 수 있습니다. 디바이스 소유 컨피그레이션이 서비스에서 잘못 소유되었거나, 컨피그레이션이 서비스와 디바이스 모두에서 잘못 소유되었지만 서비스 소유로만 예상됩니다. 조정으로 이러한 불일치를 수정할 수 있습니다. 이는 서비스가 비서비스 소유 컨피그레이션을 삭제하는 문제를 방지하기 위해 중요합니다.

재배포로 조정

조정은 서비스 재구축의 하위 기능입니다. 서비스가 CDB와 동기화되지 않은 경우 조정 작업은 조정 기능을 수행하는 것 외에 재구축도 실행합니다.

내부 작업 조정

조정 작동 방법에 대한 정확한 세부 정보는 개발자에게만 알려져 있지만 이 문서에서는 다음을 간단하게 이해할 수 있습니다.

- 1) NSO는 서비스 소유권을 수정합니다.
- 2) NSO는 이 서비스 인스턴스가 다른 서비스 소유이거나 디바이스가 소유한 경우에도 CDB에서 이 서비스 인스턴스가 소유한 모든 컨피그레이션을 삭제합니다
- 3) NSO는 서비스 인스턴스를 재구축합니다.
- 4) NSO는 다른 서비스의 서비스 리카운트 및 백포인터를 복원합니다.

서비스 소유권 관련 문제

조정 실패

"재구축"은 작동하지만 "재구축 조정"은 실패한 경우: 이는 서비스 설계와 조정 기능 작동 방식 간에 충돌이 발생했음을 나타낼 수 있습니다.

이 문제는 서비스가 나중에 배포하는 CDB에서 컨피그레이션을 읽으려고 시도하는 서비스 코드에서 발생합니다. 구축하기 전에 컨피그레이션이 CDB에 이미 일부 있기 때문에 이 서비스만 구축할 수 있습니다. 그러나 조정하는 동안 NSO는 다음 단계에서 서비스를 재구축하는 동안 서비스를 읽으려고 시도하는 컨피그레이션을 포함하여 이 서비스가 소유한 모든 컨피그레이션을 일시적으로 삭제합니다. 이 경우 일반적으로 데이터 읽기 실패를 보고하는 java 또는 python 오류가 발생합니다

서비스에서 서비스 외 소유 구성을 삭제하고 있습니다.

이 시나리오에서는 서비스 인스턴스 삭제 또는 재구축 과정에서 NSO가 서비스 미소유 컨피그레이션을 삭제하는 경우가 발생합니다. 이는 서비스 인스턴스가 원래 컨피그레이션을 생성하고 소유하며, 나중에 `ncs_cli`, `sync-from` 또는 기타 방법을 통해 수동으로 서비스 소유 컨테이너 또는 목록 내에 컨피그레이션의 일부를 추가했기 때문입니다.

이 새로운 구성 요소는 서비스에서 소유할 수 없지만, 서비스가 컨테이너나 목록에 대한 전체 소유권을 가지고 있기 때문에 서비스에서 간접적으로 소유하게 됩니다.

이를 해결하려면 `redeploy reconcile { keep-non-service-config }`를 사용하여 서비스 소유권을 수정합니다. 이 조정을 수행할 때 컨테이너 또는 목록의 참조 횟수가 증가하여 이 컨테이너 또는 목록에 서비스 소유 및 비 서비스 소유 하위 노드가 모두 있음을 반영합니다. 상위 노드 내에서는 서비스 소유 노드에만 `refcount` 및 백포인터가 있습니다.

Showcase: 서비스에서 비서비스 소유 컨피그레이션을 삭제하고 있습니다.

전체 소유권을 사용하여 구축된 단일 서비스 인스턴스부터 `ncs_cli`를 사용하여 인터페이스에 설명을 수동으로 추가합니다.

```
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

디바이스 소유 컨피그레이션이 추가되었지만 <interface>의 refcount는 1로 남아 있는지 확인합니다. 서비스 인스턴스를 삭제하려고 하면 서비스 인스턴스의 일부가 아니더라도 설명도 삭제됩니다. 이를 방지하기 위해 reconcile 명령을 실행할 수 있습니다.

```
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

조정 후 목록 인터페이스에 대한 refcount가 2로 증가했습니다. 한편 leaf ip-address에 대한 refcount는 1로 유지되었습니다. 목록 항목 "인터페이스 FE1"에는 서비스 소유 데이터와 비 서비스 소유 데이터가 모두 포함됩니다. NSO는 조정을 사용하여 소유권을 재평가하고 그에 따라 재실사를 할당합니다. 이제 삭제는 서비스 인스턴스가 완전히 소유한 영역만 대상으로 합니다. 설명이나 목록 항목이 삭제되지 않습니다.

```
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

사용자는 때때로 discard-non-service-config의 사용을 오해합니다.

조정 예에서는 "keep-non-service-config"가 사용되었습니다. Discard를 사용하는 경우 다음과 같습니다.

```
admin@ncs(config)# example-service s1 re-deploy reconcile { discard-non-service-config } dry-run
cli {
  local-node {
    data devices {
      device mydevice0 {
        config {
```



```
{/device}
```

FE1

```
{/ipaddress}
```

패키지를 다시 컴파일하고 다시 로드한 다음 테스트합니다.

```
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;
      +}
    }
  }
}
```

이전과 동일한 매개변수가 정의되었더라도 인터페이스 또는 기본 컨피그레이션은 디바이스 컨피그레이션에서 생성되지 않습니다.

병합

인터페이스 내부의 컨피그레이션에 병합 태그를 추가합니다. 목록 인터페이스의 키이므로 "interface-name"에 태그를 추가하지 마십시오. 키는 항상 목록의 동작을 상속하도록 허용되어야 합니다. 패키지를 다시 컴파일하고 다시 로드합니다.

```
{/device}
```

```
FE1
```

```
{/ipaddress}
```

서비스를 구축하기 전에 인터페이스 FE1을 수동으로 구성합니다.

```
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

인터페이스에 숨겨진 refcount 1이 있습니다. 인터페이스는 ncs_cli를 사용하여 구축되었지만 서비스 패키지에 nocreate 태그가 있습니다. 서비스에서 소유권을 가져오지 않았습니다. 디바이스 소유입니다.

기본에 refcount 1이 있습니다. 서비스에서만 소유합니다.

서비스 인스턴스가 삭제되면 ipaddress만 삭제됩니다. 해당 부분만 서비스에서 소유하고 있기 때문
입니다.

이 번역에 관하여

Cisco는 전 세계 사용자에게 다양한 언어로 지원 콘텐츠를 제공하기 위해 기계 번역 기술과 수작업 번역을 병행하여 이 문서를 번역했습니다. 아무리 품질이 높은 기계 번역이라도 전문 번역가의 번역 결과물만큼 정확하지는 않습니다. Cisco Systems, Inc.는 이 같은 번역에 대해 어떠한 책임도 지지 않으며 항상 원본 영문 문서(링크 제공됨)를 참조할 것을 권장합니다.