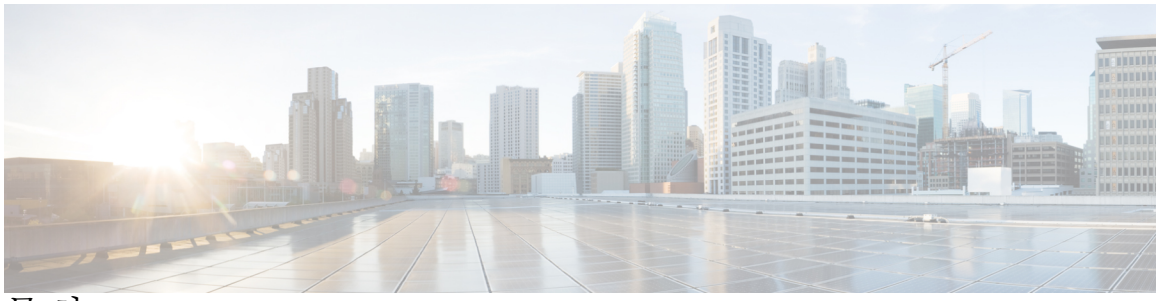
Cisco Firepower 4100/9300 FXOS Firepower 섀시 관리자 구성 가이드, 2.9(1)

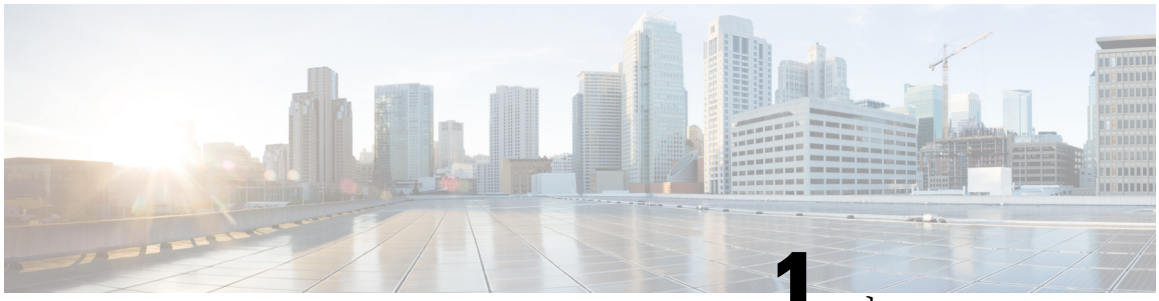
초판: 2020년 11월 2일

최종 변경: 2020년 11월 2일

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883



목 차



1 장

Security Appliance(보안 어플라이언스) 소개

Firepower Security Appliance 정보

Cisco Firepower 4100/9300 섀시는 네트워크 및 콘텐츠 보안 솔루션을 위한 차세대 플랫폼입니다. Firepower 4100/9300 섀시는 Cisco ACI(Application Centric Infrastructure) 보안 솔루션에 포함되며 확장성, 제어 일관성 및 관리 간소화를 위해 구축된 민첩한 개방형 보안 플랫폼을 제공합니다.

Firepower 4100/9300 섀시에서 제공하는 기능은 다음과 같습니다.

- 모듈형 섀시 기반 보안 시스템 — 고성능의 유연한 입/출력 구성 및 확장성을 제공합니다.
- Firepower Chassis Manager- 그래픽 사용자 인터페이스는 현재 섀시 상태를 간단하게 시각적으로 표시하며 간소화된 섀시 기능 구성을 제공합니다.
- Firepower eXtensible Operating SystemFXOS CLI - 기능 구성, 섀시 상태 모니터링 및 고급 트러블슈팅 기능에 액세스하기 위한 명령 기반 인터페이스를 제공합니다.
- FXOS REST API- 사용자가 섀시를 프로그래밍 방식으로 구성 및 관리할 수 있습니다.

Firepower 4100/9300에서 논리적 디바이스가 작동하는 방식

Firepower 4100/9300은 FXOS(Firepower eXtensible Operating System)라는 수퍼바이저에서 자체 운영 체제를 실행합니다. 온더박스 Firepower Chassis Manager는 간단한 GUI 기반 관리 기능을 제공합니다. Firepower Chassis Manager를 사용하여 수퍼바이저에서 하드웨어 인터페이스 설정, 스마트 라이선스(ASA용) 및 기타 기본 작동 매개변수를 구성합니다.

논리적 디바이스를 사용하면 하나의 애플리케이션 인스턴스와 하나의 선택적 데코레이터 애플리케이션을 실행하여 서비스 체인을 형성할 수 있습니다. 논리 디바이스를 배포할 때 수퍼바이저는 선택한 애플리케이션 이미지를 다운로드하고 기본 구성을 설정합니다. 그런 다음 애플리케이션 운영 체제 내에서 보안 정책을 구성할 수 있습니다.

논리적 디바이스는 서로 서비스 체인을 형성할 수 없으며, 백플레인을 통해 서로 통신할 수 없습니다. 모든 트래픽은 하나의 인터페이스에서 섀시를 종료하고 다른 인터페이스로 돌아가서 다른 논리적 디바이스에 연결해야 합니다. 컨테이너 인스턴스의 경우 데이터 인터페이스를 공유할 수 있습니다. 이 경우에만 여러 논리적 디바이스가 백플레인을 통해 통신할 수 있습니다.

지원되는 애플리케이션

다음 애플리케이션 유형을 사용하여 섀시에 논리적 디바이스를 구축할 수 있습니다.

FTD

FTD은 스테이트풀 방화벽, 라우팅, NGIPS(Next-Generation Intrusion Prevention System), AVC(Application Visibility and Control), URL 필터링, 악성코드 디펜스와 같은 차세대 방화벽 서비스를 제공합니다.

다음 manager(관리자) 중 하나를 사용하여 FTD을 관리할 수 있습니다.

- FMC - 별도 서버에 전체 기능을 갖춘 다중 디바이스 관리자
- FDM - 디바이스에 포함된 간소화된 단일 디바이스 관리자
- CDO - 클라우드 기반의 다중 디바이스 관리자

ASA

ASA는 고급 스테이트풀 방화벽 및 VPN 집선 장치 기능을 하나의 디바이스에서 제공합니다. 다음 관리자 중 하나를 사용해 ASA를 관리할 수 있습니다.

- ASDM—디바이스에 포함된 단일 디바이스 관리자입니다.
- CLI
- CDO - 클라우드 기반의 다중 디바이스 관리자
- CSM - 별도 서버에 전체 기능을 갖춘 다중 디바이스 관리자

Radware DefensePro (Decorator)

Radware DefensePro(vDP)를 설치하여 ASA 앞에서 실행하거나 decorator 애플리케이션으로서 FTD를 실행할 수 있습니다. vDP는 DDoS(분산 서비스 거부) 탐지 및 Firepower 4100/9300에서 mitigation 능력을 제공하는 KVM 기반 가상 플랫폼입니다. 네트워크의 트래픽은 ASA 또는 FTD에 도달하기 전에 vDP를 먼저 통과해야 합니다.

Firepower Chassis Manager 개요

FXOS에서는 플랫폼 설정 및 인터페이스 구성, 디바이스 프로비저닝, 시스템 상태 모니터링을 쉽게 수행할 수 있도록 지원하는 웹 인터페이스를 제공합니다. 사용자 인터페이스 상단에 있는 네비게이션 바를 통해 다음에 액세스할 수 있습니다.

- 개요 — 개요 페이지에서 섀시의 상태를 간편하게 모니터링할 수 있습니다. 자세한 내용은 섀시 상태 모니터링, 3 페이지를 참고하십시오.
- 인터페이스 — 인터페이스 페이지에서 섀시에 설치된 인터페이스의 상태를 확인하고 인터페이스 속성을 편집하며 인터페이스를 활성화 또는 비활성화하고 포트 채널을 생성할 수 있습니다. 자세한 내용은 인터페이스 관리, 163 페이지를 참고하십시오.
- 논리적 디바이스 — 논리적 디바이스 페이지에서 논리적 디바이스를 생성, 수정 및 삭제할 수 있습니다. 또한 기존의 논리적 디바이스의 현재 상태를 볼 수 있습니다. 자세한 내용은 논리적 디바이스, 201 페이지를 참고하십시오.
- Security Modules/Security Engine(보안 모듈/보안 엔진) - Security Modules/Security Engine(보안 모듈/보안 엔진) 페이지에서 보안 모듈/엔진의 상태를 확인하고 전원 주기, 다시 초기화, 승인 및 해제와 같은 다양한 기능을 수행할 수 있습니다. 자세한 내용은 보안 모듈/엔진 관리, 297 페이지를 참고하십시오.
- 플랫폼 설정 — 플랫폼 설정 페이지에서 날짜 및 시간, SSH, SNMP, HTTPS, AAA, Syslog 및 DNS 등 섀시 설정을 구성할 수 있습니다. 자세한 내용은 플랫폼 설정, 111 페이지를 참고하십시오.
- 시스템 설정 — 시스템 메뉴에서 다음 설정을 관리할 수 있습니다.
  - 라이선싱 — 라이선싱 페이지에서 Smart Call Home 설정을 구성하고 License Authority를 통해 섀시를 등록할 수 있습니다. 자세한 내용은 ASA의 라이선스 관리, 19 페이지를 참고하십시오.
  - 업데이트 — 업데이트 페이지에서 섀시에 플랫폼 번들 및 애플리케이션 이미지를 업로드할 수 있습니다. 자세한 내용은 이미지 관리, 61 페이지를 참고하십시오.
  - User Management(사용자 관리) — User Management(사용자 관리) 페이지에서 Firepower 4100/9300 섀시에 대한 사용자 설정을 구성하고 사용자 어카운트를 정의할 수 있습니다. 자세한 내용은 사용자 관리, 43 페이지를 참고하십시오.

섀시 상태 모니터링

개요 페이지에서 Firepower 4100/9300 섀시 섀시의 상태를 간편하게 모니터링할 수 있습니다. 개요 페이지에서는 다음 요소를 제공합니다.

- 디바이스 정보 — 개요 페이지 상단에는 Firepower 4100/9300 섀시에 대한 다음 정보가 포함되어 있습니다.
  - 섀시 이름 — 초기 구성 중 섀시에 할당된 이름 표시.
  - IP 주소 — 초기 구성 중 섀시에 할당된 관리 IP 주소 표시.
  - 모델 — Firepower 4100/9300 섀시 모델 표시.
  - 버전 — 섀시에서 실행 중인 FXOS 버전 표시.
  - 작동 상태 — 섀시의 작동 가능 상태 표시.

- 섀시 업타임 — 시스템이 마지막으로 재시작된 이후 경과한 시간 표시.
- 셧다운 버튼 — Firepower 4100/9300 섀시를 정상적으로 종료(Firepower 4100/9300 섀시 전원 끄기, 107 페이지 참조).

참고 보안 모듈/보안 엔진 페이지에서 보안 모듈/엔진 전원을 끄거나 켤 수 있습니다(보안 모듈/엔진 전원 껐다 켜기, 300 페이지 참조).

- 리부팅 버튼 — Firepower 4100/9300 섀시를 정상적으로 종료(Firepower 4100/9300 섀시 리부팅, 107 페이지 참조).
- 업타임 정보 아이콘 — 아이콘에 마우스 커서를 대면 섀시 및 설치된 보안 모듈/엔진의 업타임을 확인할 수 있습니다.

- 시각적 상태 표시 — 디바이스 정보 섹션에서는 섀시를 시각적으로 표현하여 섀시에 설치된 구성 요소를 보여주고 해당 구성 요소에 대한 일반적인 상태 정보를 제공합니다. 시각적 상태 표시에 나타난 포트에 마우스 커서를 대면 인터페이스 이름, 속도, 유형, 관리자 상태 및 작동 상태와 같은 추가 정보를 얻을 수 있습니다. 여러 보안 모듈이 있는 모델의 경우, 시각적 상태 표시에 나타난 모듈에 마우스 커서를 대면 디바이스 이름, 템플릿 유형, 관리자 상태 및 작동 상태와 같은 추가 정보를 얻을 수 있습니다. 논리적 디바이스가 해당 보안 모듈에 설치되어 있으면 관리 IP 주소, 소프트웨어 버전 및 논리적 디바이스 모드를 확인할 수도 있습니다.
- 상세한 상태 정보 — 시각적 상태 표시에서는 섀시의 상세한 상태 정보가 포함된 표를 제공합니다. 상태 정보는 결함, 인터페이스, 디바이스, 라이선스 및 인벤토리의 5가지 섹션으로 나뉩니다. 확인하려는 정보의 요약 영역을 클릭하여 표에 있는 각 해당 섹션에 대한 요약을 확인할 수 있으며 각 섹션에 대한 추가적인 세부사항을 확인할 수 있습니다.

  시스템은 섀시에 대해 다음의 상세한 상태 정보를 제공합니다.

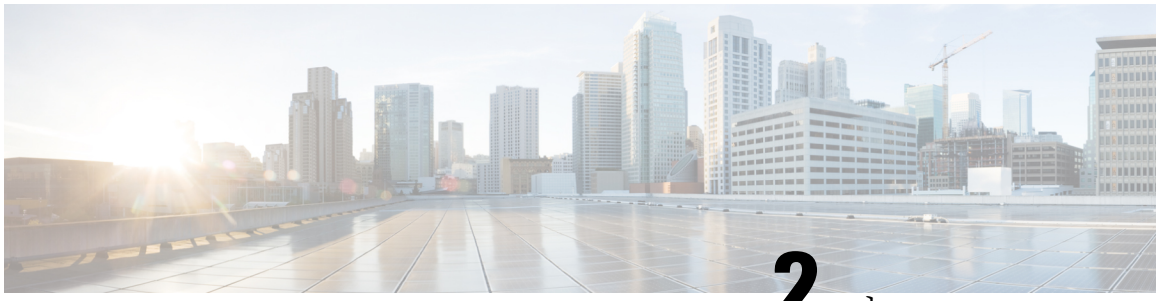
  - 결함 — 시스템에서 생성된 결함을 나열합니다. 결함은 중대, 주요, 사소, 경고 및 정보의 심각도별로 정렬됩니다. 나열된 각 결함에 대해 심각도, 결함 설명, 원인, 발생 횟수 및 최근 발생 시간을 확인할 수 있습니다. 또한 결함 승인 여부를 확인할 수 있습니다.

    결함 중 하나를 클릭하여 해당 결함에 대한 추가적인 세부사항을 확인하거나 결함을 승인할 수 있습니다. 여러 결함을 승인하려면 승인할 각 결함 옆의 체크 박스를 클릭하고 **Acknowledge(**승인**)**를 클릭합니다. **Select All Faults(**모든 결함 선택**)** 및 **Cancel Selected Faults(**선택한 결함 취소**)** 버튼을 사용하여 여러 결함을 신속하게 선택하거나 선택 취소할 수 있습니다.

참고 결함의 근본 원인이 해결되면 해당 결함은 다음 폴링 간격 동안 목록에서 자동으로 지워집니다. 사용자가 특정 결함에 대한 해결책과 관련된 작업을 진행 중인 경우, 결함을 승인하여 해당 결함이 현재 해결 중이라는 사실을 다른 사용자에게 알릴 수 있습니다.

- Interfaces(인터페이스) — 시스템에 설치된 인터페이스를 나열합니다. **All Interfaces(**모든 인터페이스**)** 탭에는 인터페이스 이름, 운영 상태, 관리 상태, 수신한 바이트의 수, 전송한 바이트의 수가 표시됩니다. 하드웨어 바이패스 탭에는 FTD 애플리케이션에서 하드웨어 바이패스 기능이 지원되는 인터페이스 쌍만 표시됩니다. 각 쌍에 대해 작동 상태가 표시됩니다. 작동 상태는 disabled(비활성화됨, 쌍에 대해 하드웨어 바이패스가 구성되지 않음), standby(대기, 하드웨어 바이패스가 구성되었지만 현재 활성 상태는 아님) 및 bypass(우회, 하드웨어 바이패스에서 활성 상태임) 중 하나입니다.
- 인스턴스 - 시스템에 구성된 논리적 디바이스가 나열되며 각 논리적 디바이스에 대해 세부 정보가 제공됩니다(막대 위에 커서를 대면 표시됨). 제공되는 세부 정보는 디바이스 이름, 상태, 이미지 버전, 관리 IP 주소 및 코어 수입니다.페이지 하단에서 Ingress VLAN Group Entry Utilization(인그레스 VLAN 그룹 항목 사용률) 및 Switch Forwarding Path Entry Utilization(스위치 전달 경로 항목 사용률)을 볼 수도 있습니다.
- 라이선스 - (ASA 논리적 디바이스의 경우) 스마트 라이선싱 활성화 여부를 표시하며, 라이선스의 현재 등록 상태를 제공하고, 섀시의 라이선스 권한 부여 정보를 표시합니다.
- 인벤토리 — 섀시에 설치된 구성 요소를 나열하고 해당 구성 요소와 관련된 세부사항(예: 구성 요소 이름, 코어 수, 설치 위치, 작동 상태, 동작 가능성, 용량, 전원, 열, 일련 번호, 모델 번호, 부품 번호 및 벤더)을 제공합니다.

참고 전원 이중화가 구현된 경우, FXOS에서 전원 이중화와 관련된 설정을 변경하지 마십시오.



2 장

시작하기

작업 흐름

다음 절차에서는 Firepower 4100/9300 새시 구성 시 완료해야 하는 기본 작업을 보여줍니다.

프로시저

단계 **1** Firepower 4100/9300 새시 하드웨어를 구성합니다(*Cisco Firepower Security Appliance* 하드웨어 설치 가이드 참조).

단계 **2** 초기 구성을 완료합니다(초기 구성, 8 페이지 참고).

단계 **3** Firepower Chassis Manager에 로그인합니다(로그인 또는 로그아웃 Firepower Chassis Manager, 15 페이지 참조).

단계 **4** 날짜 및 시간을 설정합니다(날짜 및 시간 설정, 111 페이지 참고).

단계 **5** DNS 서버를 구성합니다(DNS 서버 구성, 156 페이지 참고).

단계 **6** 제품 라이선스를 등록합니다(ASA의 라이선스 관리, 19 페이지 참고).

단계 **7** 사용자를 구성합니다(사용자 관리, 43 페이지 참고).

단계 **8** 필요 시 소프트웨어 업데이트를 수행합니다(이미지 관리, 61 페이지 참고).

단계 **9** 추가 플랫폼 설정을 구성합니다(플랫폼 설정, 111 페이지 참고).

단계 **10** 인터페이스를 구성합니다(인터페이스 관리, 163 페이지 참고).

단계 **11** 논리적 디바이스를 생성합니다(논리적 디바이스, 201 페이지 참고).

초기 구성

Firepower Chassis Manager 또는 FXOS CLI를 사용하여 시스템을 구성하고 관리할 수 있으려면 먼저 몇 가지 초기 구성 작업을 수행해야 합니다. 콘솔 포트를 통해 액세스되는 FXOS CLI를 사용하거나 관리 포트를 통해 액세스하는 SSH, HTTPS 또는 REST API를 사용하여 초기 구성을 수행할 수 있습니다(이 절차를 로우 터치(low-touch) 프로비저닝이라고도 함).

콘솔 포트를 사용한 초기 구성

FXOS CLI를 사용하여 처음으로 Firepower 4100/9300 섀시에 액세스할 때 시스템을 구성하는 데 사용할 수 있는 설정 마법사가 나타납니다.

참고 초기 설정을 반복하려면 다음 명령을 사용하여 기존 구성을 지워야 합니다.

```
Firepower-chassis# connect local-mgmt
firepower-chassis(local-mgmt)# erase configuration
```

Firepower 4100/9300 섀시의 단일 관리 포트에는 IPv4 주소, 게이트웨이 및 서브넷 마스크 하나만, 또는 IPv6 주소, 게이트웨이 및 네트워크 접두사 하나만 지정해야 합니다. 관리 포트 IP 주소로 IPv4 또는 IPv6 주소 중 하나를 구성할 수 있습니다.

시작하기 전에

1. Firepower 4100/9300 섀시에서 다음의 물리적 연결을 확인합니다.
   - 콘솔 포트는 컴퓨터 터미널 또는 콘솔 서버에 물리적으로 연결됩니다.
   - 1Gbps 이더넷 관리 포트는 외부 허브, 스위치 또는 라우터에 연결됩니다.

   자세한 내용은 하드웨어 설치 가이드를 참조하십시오.
2. 콘솔 포트에 연결된 컴퓨터 터미널(또는 콘솔 서버)의 콘솔 포트 파라미터가 다음과 같은지 확인합니다.
   - 9600보드
   - 8 데이터 비트
   - 패리티 없음
   - 1 스톱 비트
3. 설정 스크립트에 사용할 다음 정보를 수집합니다.
   - 새 관리자 비밀번호
   - 관리 IP 주소 및 서브넷 마스크

- 게이트웨이 IP 주소
- HTTPS 및 SSH 액세스를 허용할 서브넷
- 호스트 이름 및 도메인 이름
- DNS 서버 IP 주소

프로시저

단계 **1** 섀시 전원을 켭니다.

단계 **2** 터미널 에뮬레이터를 사용하여 시리얼 콘솔 포트에 연결합니다.

Firepower 4100/9300에는 RS-232-to-RJ-45 시리얼 콘솔 케이블이 포함되어 있습니다. 연결을 설정하려면 서드파티 시리얼-USB 케이블을 사용해야 할 수도 있습니다. 다음 시리얼 매개변수를 사용합니다.

- 9600보드
- 8 데이터 비트
- 패리티 없음
- 1 스톱 비트

단계 **3** 표시되는 화면 컨피그레이션을 완료합니다.

참고 초기 구성 중에 언제든지 선택적으로 debug(디버그) 메뉴에 진입하여 설정 문제를 디버깅하거나 구성을 중단하고 시스템을 리부팅할 수 있습니다. 디버그 메뉴를 시작하려면 Ctrl-C를 누릅니다. 디버그 메뉴를 종료하려면 Ctrl-D를 두 번 누릅니다. Ctrl-D를 처음 누른 후 두 번째 키를 누르는 동안 중간에 입력하는 모든 내용은 Ctrl-D를 두 번째 누른 후에 실행됩니다.

예제:

```
          ---- Basic System Configuration Dialog ----

This setup utility will guide you through the basic configuration of
the system. Only minimal configuration including IP connectivity to
the FXOS Supervisor is performed through these steps.

Type Ctrl-C at any time for more options or to abort configuration
and reboot system.
To back track or make modifications to already entered values,
complete input till end of section and answer no when prompted
to apply configuration.


You have chosen to setup a new Security Appliance.
      Continue? (yes/no): y

Enforce strong password? (yes/no) [y]: n

Enter the password for "admin": Farscape&32
```

```
  Confirm the password for "admin": Farscape&32
  Enter the system name:  firepower-9300

  Supervisor Mgmt IP address : 10.80.6.12

  Supervisor Mgmt IPv4 netmask : 255.255.255.0

  IPv4 address of the default gateway : 10.80.6.1

  The system cannot be accessed via SSH if SSH Mgmt Access is not configured.

  Do you want to configure SSH Mgmt Access? (yes/no) [y]: y

  SSH Mgmt Access host/network address (IPv4/IPv6): 10.0.0.0

  SSH Mgmt Access IPv4 netmask: 255.0.0.0

  Firepower Chassis Manager cannot be accessed if HTTPS Mgmt Access is not configured.

  Do you want to configure HTTPS Mgmt Access? (yes/no) [y]: y

  HTTPS Mgmt Access host/network address (IPv4/IPv6): 10.0.0.0

  HTTPS Mgmt Access IPv4 netmask: 255.0.0.0

  Configure the DNS Server IP address? (yes/no) [n]: y

    DNS IP address : 10.164.47.13

  Configure the default domain name? (yes/no) [n]: y

    Default domain name : cisco.com

  Following configurations will be applied:

    Switch Fabric=A
    System Name=firepower-9300
    Enforced Strong Password=no
    Supervisor Mgmt IP Address=10.89.5.14
    Supervisor Mgmt IP Netmask=255.255.255.192
    Default Gateway=10.89.5.1
    SSH Access Configured=yes
        SSH IP Address=10.0.0.0
        SSH IP Netmask=255.0.0.0
    HTTPS Access Configured=yes
        HTTPS IP Address=10.0.0.0
        HTTPS IP Netmask=255.0.0.0
    DNS Server=72.163.47.11
    Domain Name=cisco.com

  Apply and save the configuration (select 'no' if you want to re-enter)? (yes/no): y
  Applying configuration. Please wait...  Configuration file - Ok
.....

Cisco FPR Series Security Appliance
firepower-9300 login:  admin
Password: Farscape&32
Successful login attempts for user 'admin' : 1
Cisco Firepower Extensible Operating System (FX-OS) Software
TAC support: http://www.cisco.com/tac
Copyright (c) 2009-2019, Cisco Systems, Inc. All rights reserved.

[...]
```

```
firepower-chassis#
```

관리 포트를 사용한 로우 터치(Low-Touch) 프로비저닝

Firepower 4100/9300 섀시이 부팅 시 시작 구성을 찾을 수 없는 경우, 디바이스는 Low-Touch 프로비저닝 모드를 시작하여, DHCP(Dynamic Host Control Protocol) 서버를 찾은 다음 관리 인터페이스 IP 주소를 사용하여 자체적으로 부트스트랩합니다. 그런 다음 관리 인터페이스를 통해 연결하여 SSH, HTTPS 또는 FXOS REST API를 사용하여 시스템을 구성할 수 있습니다.

참고 초기 설정을 반복하려면 다음 명령을 사용하여 기존 구성을 지워야 합니다.

```
Firepower-chassis# connect local-mgmt
firepower-chassis(local-mgmt)# erase configuration
```

Firepower 4100/9300 섀시의 단일 관리 포트에는 IPv4 주소, 게이트웨이 및 서브넷 마스크 하나만, 또는 IPv6 주소, 게이트웨이 및 네트워크 접두사 하나만 지정해야 합니다. 관리 포트 IP 주소로 IPv4 또는 IPv6 주소 중 하나를 구성할 수 있습니다.

시작하기 전에

설정 스크립트에 사용할 다음 정보를 수집합니다.

- 새 관리자 비밀번호
- 관리 IP 주소 및 서브넷 마스크
- 게이트웨이 IP 주소
- HTTPS 및 SSH 액세스를 허용할 서브넷
- 호스트 이름 및 도메인 이름
- DNS 서버 IP 주소

프로시저

단계 **1** Firepower 4100/9300 섀시의 관리 포트에 IP 주소를 할당하도록 DHCP 서버를 구성합니다.

Firepower 4100/9300 섀시의 DHCP 클라이언트 요청에는 다음이 포함됩니다.

- 관리 인터페이스의 MAC 주소입니다.
- DHCP 옵션 60(vendor-class-identifier) - "FPR9300" 또는 "FPR4100"으로 설정합니다.

- DHCP 옵션 61(dhcp-client-identifier) - Firepower 4100/9300 섀시 일련 번호로 설정합니다. 이 일련 번호는 섀시의 풀아웃 탭에서 확인할 수 있습니다.

단계 **2** Firepower 4100/9300 섀시의 전원을 켭니다.
섀시가 부팅될 때 시작 구성을 찾을 수 없는 경우, 디바이스는 Low-Touch 프로비저닝 모드를 시작합니다.

단계 **3** HTTPS를 사용하여 시스템을 구성하려면 다음을 수행합니다.

a) 지원되는 브라우저를 사용하여 주소 표시줄에 다음 URL을 입력합니다.

https://<*ip_address*>/api

여기서<*ip_address*>는 DHCP 서버에 의해 할당된 Firepower 4100/9300 섀시에 있는 관리 포트의 IP 주소입니다.

참고 지원되는 브라우저에 대한 정보는 사용 중인 버전에 대한 릴리스 노트를 참고하십시오 (http://www.cisco.com/c/en/us/support/security/firepower-9000-series/products-release-notes-list.html 참고).

b) 메시지가 표시되면 사용자 이름 **install** 및 비밀번호 <*chassis_serial_number*> 를 사용하여 로그인합니다.

<*chassis_serial_number*>는 섀시의 태그를 검사하여 얻을 수 있습니다.

c) 표시되는 화면 컨피그레이션을 완료합니다.

- 강력한 비밀번호 시행 정책(강력한 비밀번호 지침에 대해서는 사용자 계정, 43 페이지 참고).
- 관리자 계정의 비밀번호.
- 시스템 이름
- 관리자 관리 IPv4 주소 및 서브넷 마스크, 또는 IPv6 주소 및 프리픽스.
- 기본 게이트웨이 IPv4 또는 IPv6 주소
- SSH 액세스가 허용되는 호스트/네트워크 주소 및 넷마스크/프리픽스.
- HTTPS 액세스가 허용되는 호스트/네트워크 주소 및 넷마스크/프리픽스.
- DNS 서버 IPv4 또는 IPv6 주소.
- 기본 도메인 이름

d) **Submit(**제출**)**을 클릭합니다.

단계 **4** SSH를 사용하여 시스템을 구성하려면 다음을 수행합니다.

a) 다음 명령을 사용하여 관리 포트에 연결합니다.

ssh install@<*ip_address*>

여기서 <*ip_address*>는 DHCP 서버에 의해 할당된 Firepower 4100/9300 섀시에 있는 관리 포트의 IP 주소입니다.

b) 메시지가 표시되면 비밀번호 **Admin123**을 사용하여 로그인합니다.

c) 표시되는 화면 컨피그레이션을 완료합니다.

참고 초기 구성 중에 언제든지 선택적으로 debug(디버그) 메뉴에 진입하여 설정 문제를 디버깅하거나 구성을 중단하고 시스템을 리부팅할 수 있습니다. 디버그 메뉴를 시작하려면 Ctrl-C를 누릅니다. 디버그 메뉴를 종료하려면 Ctrl-D를 두 번 누릅니다. Ctrl-D를 처음 누른 후 두 번째 키를 누르는 동안 중간에 입력하는 모든 내용은 Ctrl-D를 두 번째 누른 후에 실행됩니다.

예제:

```
          ---- Basic System Configuration Dialog ----

This setup utility will guide you through the basic configuration of
the system. Only minimal configuration including IP connectivity to
the FXOS Supervisor is performed through these steps.

Type Ctrl-C at any time for more options or to abort configuration
and reboot system.
To back track or make modifications to already entered values,
complete input till end of section and answer no when prompted
to apply configuration.


You have chosen to setup a new Security Appliance.
      Continue? (yes/no): y

Enforce strong password? (yes/no) [y]: n

Enter the password for "admin": Farscape&32
Confirm the password for "admin": Farscape&32
Enter the system name:  firepower-9300

Supervisor Mgmt IP address : 10.80.6.12

Supervisor Mgmt IPv4 netmask : 255.255.255.0

IPv4 address of the default gateway : 10.80.6.1

The system cannot be accessed via SSH if SSH Mgmt Access is not configured.

Do you want to configure SSH Mgmt Access? (yes/no) [y]: y

SSH Mgmt Access host/network address (IPv4/IPv6): 10.0.0.0

SSH Mgmt Access IPv4 netmask: 255.0.0.0

Firepower Chassis Manager cannot be accessed if HTTPS Mgmt Access is not configured.

Do you want to configure HTTPS Mgmt Access? (yes/no) [y]: y

HTTPS Mgmt Access host/network address (IPv4/IPv6): 10.0.0.0

HTTPS Mgmt Access IPv4 netmask: 255.0.0.0

Configure the DNS Server IP address? (yes/no) [n]: y

  DNS IP address : 10.164.47.13

Configure the default domain name? (yes/no) [n]: y
```

```
    Default domain name : cisco.com

  Following configurations will be applied:

    Switch Fabric=A
    System Name=firepower-9300
    Enforced Strong Password=no
    Supervisor Mgmt IP Address=10.89.5.14
    Supervisor Mgmt IP Netmask=255.255.255.192
    Default Gateway=10.89.5.1
    SSH Access Configured=yes
        SSH IP Address=10.0.0.0
        SSH IP Netmask=255.0.0.0
    HTTPS Access Configured=yes
        HTTPS IP Address=10.0.0.0
        HTTPS IP Netmask=255.0.0.0
    DNS Server=72.163.47.11
    Domain Name=cisco.com

  Apply and save the configuration (select 'no' if you want to re-enter)? (yes/no): y
  Applying configuration. Please wait...  Configuration file - Ok
.....

Initial Setup complete, Terminating sessions
.Connection to <ip_address> closed.
```

단계 **5** FXOS REST API를 사용하여 시스템을 구성하려면 다음을 수행합니다.

REST API를 사용하여 시스템을 구성하려면 다음 예를 사용합니다. 자세한 내용은 https://developer.cisco.com/site/ssp/firepower/를 참고하십시오.

참고 dns, domain_name, https_net, https_mask, ssh_net 및 ssh_mask 속성은 선택 사항입니다. 다른 모든 속성은 REST API 구성에 필수입니다.

```
IPv4 REST API example:

{
    "fxosBootstrap": {
        "dns": "1.1.1.1",
        "domain_name": "cisco.com",
        "mgmt_gw": "192.168.0.1",
        "mgmt_ip": "192.168.93.3",
        "mgmt_mask": "255.255.0.0",
        "password1": "admin123",
        "password2": "admin123",
        "strong_password": "yes",
        "system_name": "firepower-9300",
        "https_mask": "2",
        "https_net": "::",
        "ssh_mask": "0",
        "ssh_net": "::"
    }
}


IPV6 REST API example

{
    "fxosBootstrap": {
        "dns": "2001::3434:4343",
        "domain_name": "cisco.com",
        "https_mask": "2",
```

```
        "https_net": "::",
        "mgmt_gw": "2001::1",
        "mgmt_ip": "2001::2001",
        "mgmt_mask": "64",
        "password1": "admin123",
        "password2": "admin123",
        "ssh_mask": "0",
        "ssh_net": "::",
        "strong_password": "yes",
        "system_name": "firepower-9300"
    }
}
```

로그인 또는 로그아웃 Firepower Chassis Manager

Firepower Chassis Manager를 사용하여 Firepower 4100/9300 섀시를 구성하려면 유효한 사용자 어카운트를 사용하여 로그인해야 합니다. 사용자 어카운트에 대한 자세한 내용은 사용자 관리, 43 페이지 섹션을 참조하십시오.

일정 기간 동안 아무 작업도 하지 않으면 시스템에서 자동으로 로그아웃됩니다. 기본적으로는 10분 동안 작업을 하지 않으면 시스템에서 로그아웃됩니다. 이 시간 초과 설정을 구성하려면 세션 시간 초과 구성, 53 페이지 섹션을 참조하십시오. 세션이 활성 상태이더라도 일정 기간이 지나면 사용자가 시스템에서 로그아웃되는 절대 시간 초과 설정을 구성할 수도 있습니다. 절대 시간 초과 설정을 구성하려면 절대 세션 시간 초과 구성, 54 페이지 섹션을 참조하십시오.

Firepower Chassis Manager에서 자동으로 로그아웃하게 하는 모든 시스템 변경 사항 목록은 Firepower Chassis Manager 세션을 종료시키는 시스템 변경 사항, 89 페이지 섹션을 참조하십시오.

참고 선택적으로, 로그인 시도 실패를 특정 횟수만큼만 허용하고 그 이후에는 지정된 시간 동안 사용자가 잠기도록 Firepower Chassis Manager를 구성할 수 있습니다. 자세한 내용은 최대 로그인 시도 횟수 설정, 55 페이지를 참고하십시오.

프로시저

단계 **1** Firepower Chassis Manager에 로그인하려면 다음 작업을 수행하십시오.

a) 지원되는 브라우저를 사용하여 주소 표시줄에 다음 URL을 입력합니다.

`https://<chassis_mgmt_ip_address>`

여기서 <*chassis_mgmt_ip_address*>는 초기 구성을 설정하는 동안 입력한 Firepower 4100/9300 섀시의 IP 주소 또는 호스트 이름입니다.

참고 지원되는 브라우저에 대한 정보는 사용 중인 버전에 대한 릴리스 노트를 참고하십시오 (http://www.cisco.com/c/en/us/support/security/firepower-9000-series/products-release-notes-list.html 참고).

b) 사용자 이름 및 비밀번호를 입력합니다.

c) **Login(**로그인**)**을 클릭합니다.

로그인하면 Firepower Chassis Manager가 열리고 요약 페이지가 표시됩니다.

단계 **2** Firepower Chassis Manager에서 로그아웃하려면 네비게이션 바에서 사용자 이름을 가리킨 다음 **Logout(**로그아웃**)**을 선택합니다.
Firepower Chassis Manager에서 로그아웃되고 로그인 화면으로 돌아갑니다.

액세스 - FXOS CLI

콘솔 포트에 전원이 연결된 터미널을 사용하여 FXOS CLI에 연결할 수 있습니다. 콘솔 포트에 연결된 컴퓨터 터미널(또는 콘솔 서버)의 콘솔 포트 파라미터가 다음과 같은지 확인합니다.

- 9600보드
- 8 데이터 비트
- 패리티 없음
- 1 스톱 비트

또한 SSH 및 텔넷을 사용하여 FXOS CLI에 연결할 수 있습니다. FXOS는 최대 8개의 동시 SSH 연결을 지원합니다. SSH를 사용하여 연결하려면 Firepower 4100/9300 섀시의 IP 주소 또는 호스트 이름을 알아야 합니다.

다음 구문 예시 중에서 하나를 사용하여 SSH, 텔넷 또는 Putty를 통해 로그인할 수 있습니다.

참고 SSH 로그인은 대/소문자를 구분합니다.

SSH를 사용하는 Linux 터미널에서 다음 구문을 사용합니다.

- **ssh ucs-***auth-domain**username*@{*UCSM-ip-address* | *UCMS-ipv6-address*}

```
ssh ucs-example\\jsmith@192.0.20.11
ssh ucs-example\\jsmith@2001::1
```

- **ssh -l ucs-***auth-domain**username* {*UCSM-ip-address*| *UCSM-ipv6-address*| *UCSM-host-name*}

```
ssh -l ucs-example\\jsmith 192.0.20.11
ssh -l ucs-example\\jsmith 2001::1
```

- **ssh** {*UCSM-ip-address* | *UCSM-ipv6-address* | *UCSM-host-name*} **-l ucs-***auth-domain**username*

```
ssh 192.0.20.11 -l ucs-example\\jsmith
ssh 2001::1 -l ucs-example\\jsmith
```

- **ssh ucs-***auth-domain**username*@{*UCSM-ip-address* | *UCSM-ipv6-address*}

```
ssh ucs-ldap23\\jsmith@192.0.20.11
```

```
ssh ucs-ldap23\\jsmith@2001::1
```

텔넷을 사용하는 Linux 터미널에서 다음 구문을 사용합니다.

참고 텔넷은 기본적으로 비활성화되어 있습니다. 텔넷 활성화에 대한 지침은 텔넷 구성, 119 페이지를 참고하십시오.

- **telnet ucs-***UCSM-host-name* **ucs-***auth-domain**username*

```
telnet ucs-qa-10
login: ucs-ldap23\blradmin
```

- **telnet ucs-**{*UCSM-ip-address* | *UCSM-ipv6-address*}*ucs-auth-domain**username*

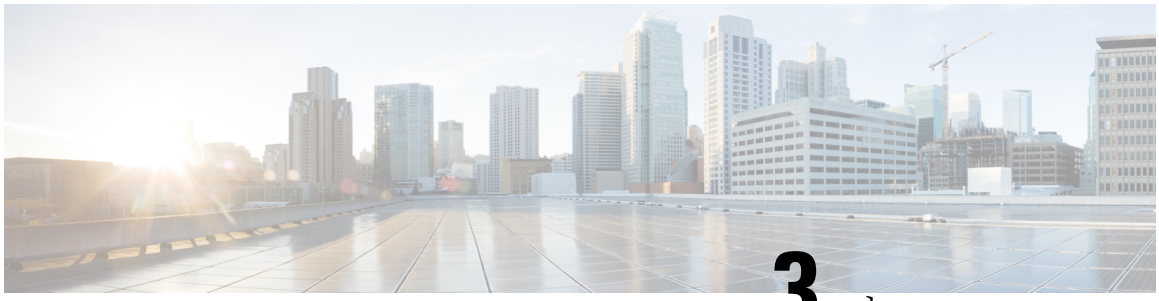
```
telnet 10.106.19.12 2052
ucs-qa-10-A login: ucs-ldap23\blradmin
```

Putty 클라이언트에서 다음 구문을 사용합니다.

- 다음으로 로그인: **ucs-***auth-domain**username*

```
Login as: ucs-example\jsmith
```

참고 기본 인증이 로컬로 설정되어 있고 콘솔 인증이 LDAP으로 설정된 경우, **ucs-local\admin**을 사용하여 Putty 클라이언트에서 패브릭 인터커넥트에 로그인할 수 있으며 이때 admin은 로컬 어카운트의 이름입니다.



3 장

ASA의 라이선스 관리

시스코 스마트 라이선싱은 시스코 포트폴리오 및 조직 전체에서 소프트웨어를 보다 쉽고 빠르고 일관적인 방식으로 구매하고 관리할 수 있는 유연한 라이선싱 모델입니다. 또한 사용자가 액세스할 수 있는 항목을 제어할 수 있어 안전합니다. 스마트 라이선싱을 사용하면 다음과 같은 이점을 누릴 수 있습니다.

- 손쉬운 활성화: 스마트 라이선싱은 전체 조직에서 사용할 수 있는 소프트웨어 라이선스 풀을 설정하므로 더 이상 PAK(제품 활성화 키)가 필요하지 않습니다.
- 통합 관리: MCE(My Cisco Entitlements)는 사용하기 쉬운 포털에서 모든 시스코 제품 및 서비스에 대한 완벽한 보기를 제공하므로 무엇을 보유하고 있으며 무엇을 사용 중인지 항상 파악할 수 있습니다.
- 라이선스 유연성: 소프트웨어가 하드웨어에 노드로 고정되어 있지 않으므로 필요에 따라 라이선스를 쉽게 사용하고 전송할 수 있습니다.

스마트 라이선싱을 사용하려면 먼저 Cisco Software Central(software.cisco.com)에서 스마트 어카운트를 설정해야 합니다.

시스코 라이선싱에 대한 자세한 내용은 cisco.com/go/licensingguide를 참조하세요.

참고 이 섹션은 Firepower 4100/9300 섀시의 ASA 논리적 디바이스에만 적용됩니다. FTD 논리적 디바이스의 라이선싱에 대한 자세한 내용은 FMC 컨피그레이션 가이드를 참조하십시오.

Smart Software Licensing 정보

이 섹션에서는 Smart Software Licensing이 적용되는 방법에 관해 설명합니다.

참고 이 섹션은 Firepower 4100/9300 섀시의 ASA 논리적 디바이스에만 적용됩니다. FTD 논리적 디바이스의 라이선싱에 대한 자세한 내용은 FMC 컨피그레이션 가이드를 참조하십시오.

ASA의 Smart Software Licensing

Firepower 4100/9300 섀시의 ASA 애플리케이션의 경우, Smart Software Licensing 구성은 Firepower 4100/9300 섀시 수퍼바이저와 애플리케이션으로 나뉩니다.

- Firepower 4100/9300 섀시— 수퍼바이저에 모든 Smart Software Licensing 인프라를 구성하며 여기에는 License Authority와 통신하는 데 필요한 파라미터가 포함됩니다. Firepower 4100/9300 섀시 자체는 작동하기 위한 라이선스가 필요하지 않습니다.

참고 섀시 간 클러스터링에서는 클러스터의 각 섀시에서 동일한 Smart Licensing 방법을 활성화해야 합니다.

- ASA 애플리케이션 — 애플리케이션의 모든 라이선스 엔타이틀먼트를 구성합니다.

참고 Cisco 전송 게이트웨이는 Firepower 4100/9300 보안 어플라이언스에서 지원되지 않습니다.

Smart Software Manager 및 어카운트

디바이스 라이선스를 1개 이상 구매한 경우, Cisco Smart Software Manager에서 라이선스를 관리할 수 있습니다.

https://software.cisco.com/#module/SmartLicensing

Smart Software Manager에서 조직의 마스터 계정을 만들 수 있습니다.

참고 아직 어카운트가 없는 경우 새 어카운트 설정 링크를 클릭합니다. Smart Software Manager에서 조직의 마스터 계정을 만들 수 있습니다.

기본적으로 마스터 계정의 기본 가상 계정에 라이선스가 지정됩니다. 계정 관리자는 선택적으로 추가 가상 계정을 만들 수 있습니다. 이를테면 지역, 부서, 자회사를 위한 계정을 만들 수 있습니다. 여러 가상 계정이 있으면 많은 수의 라이선스 및 디바이스를 더 편리하게 관리할 수 있습니다.

오프라인 관리

디바이스에서 인터넷에 액세스할 수 없으며 License Authority에 등록할 수 없는 경우, 오프라인 라이선싱을 구성할 수 있습니다.

영구 라이선스 예약

보안상의 이유로 디바이스에서 인터넷에 액세스할 수 없는 경우 선택적으로 각 ASA에 대한 영구 라이선스를 요청할 수 있습니다. 영구 라이선스 사용 시에는 License Authority에 주기적으로 액세스할 필요가 없습니다. PAK 라이선스와 마찬가지로 라이선스를 구매한 후 ASA용 라이선스 키를 설치하면 됩니다. 그러나 PAK 라이선스와는 달리 Smart Software Manager를 사용하여 라이선스를 받고 관리합니다. 일반 Smart Licensing 모드와 영구 라이선스 예약 모드 간을 쉽게 전환할 수 있습니다.

Carrier 라이선스 및 최대 보안 컨텍스트를 갖춘 표준 Tier 등 모든 기능을 활성화하는 라이선스를 얻을 수 있습니다. 이 라이선스는 Firepower 4100/9300 섀시에서 관리되지만 ASA에서 엔타이틀먼트 사용을 허용하도록 ASA 구성의 엔타이틀먼트도 요청해야 합니다.

Satellite 서버

보안상의 이유로 디바이스가 인터넷에 액세스할 수 없는 경우 선택적으로 로컬 Smart Software Manager Satellite 서버를 VM(가상 머신)으로 설치할 수 있습니다. Smart Software Manager 기능의 하위 집합을 제공하는 이 Satellite을 통해 모든 로컬 디바이스에 필수 라이선싱 서비스를 제공할 수 있습니다. Satellite는 라이선스 사용량 동기화를 위해 메인 License Authority에 주기적으로 연결하기만 하면 됩니다. 일정에 따라 동기화하거나 수동으로 동기화할 수 있습니다.

Satellite 애플리케이션을 다운로드하고 구축하면 인터넷을 사용하여 Cisco SSM에 데이터를 전송하지 않고 다음 기능을 수행할 수 있습니다.

- 라이선스 활성화 또는 등록
- 회사의 라이선스 보기
- 회사 엔터티 간 라이선스 양도

자세한 내용은 Smart Account Manager Satellite의 Smart Software Manager Satellite 설치 및 환경 설정 가이드를 참고하십시오.

가상 어카운트별로 관리되는 라이선스 및 디바이스

라이선스 및 디바이스는 가상 어카운트별로 관리됩니다. 가상 계정의 디바이스에서만 해당 계정에 지정된 라이선스를 사용할 수 있습니다. 추가 라이선스가 필요할 경우 다른 가상 계정의 미사용 라이선스를 이전할 수 있습니다. 또한 가상 어카운트 간에 디바이스를 이전할 수도 있습니다.

Firepower 4100/9300 섀시만 디바이스로 등록되며 섀시의 ASA 애플리케이션은 고유한 라이선스를 요청합니다. 예를 들어, 보안 모듈이 3개 있는 Firepower 9300 섀시의 경우 섀시는 디바이스 1개로 간주되지만 모듈은 개별 라이선스 3개를 사용합니다.

평가판 라이선스

Firepower 4100/9300 섀시는 두 가지 유형의 평가판 라이선스를 지원합니다.

- 섀시 레벨 평가 모드 — Firepower 4100/9300 섀시가 Licensing Authority에 등록되기 전에 평가 모드로 90일(총 사용량) 동안 작동됩니다. 이 모드에서 ASA는 특정 엔타이틀먼트를 요청할 수 없으며 기본 엔타이틀먼트만 활성화됩니다. 이 기간이 종료되면 Firepower 4100/9300 섀시는 컴플라이언스 미준수 상태가 됩니다.
- 엔타이틀먼트 기반 평가 모드 - Firepower 4100/9300 섀시가 Licensing Authority에 등록되고 나면 ASA에 할당할 수 있는 시간 기반 평가판 라이선스를 받을 수 있습니다. ASA에서는 평소대로 엔타이틀먼트를 요청합니다. 시간 기반 라이선스가 만료되면 시간 기반 라이선스를 갱신하거나 영구 라이선스를 받아야 합니다.

참고 Strong Encryption(3DES/AES)용 평가판 라이선스를 받을 수는 없으며 영구 라이선스만 이 엔타이틀먼트를 지원합니다.

Smart Software Manager 통신

이 섹션에서는 디바이스가 Smart Software Manager와 통신하는 방법을 설명합니다.

디바이스 등록 및 토큰

각 가상 어카운트에서 등록 토큰을 만들 수 있습니다. 이 토큰은 기본적으로 30일간 유효합니다. 각 섀시를 구축할 때 또는 기존 섀시를 등록할 때 이 토큰 ID와 엔타이틀먼트 레벨을 입력합니다. 기존 토큰이 만료되면 새 토큰을 생성할 수 있습니다.

구축 후 시작 시 또는 기존 섀시에서 이 파라미터를 직접 구성한 이후에 섀시는 Cisco License Authority에 등록됩니다. 섀시를 토큰과 함께 등록하면 License Authority는 섀시와 License Authority 간의 통신을 위한 ID 인증서를 발급합니다. 이 인증서는 6개월마다 갱신되지만 1년간 유효합니다.

License Authority와의 정기적인 통신

디바이스는 30일마다 License Authority와 통신합니다. Smart Software Manager에서 변경할 경우 변경 사항이 즉시 적용되도록 디바이스에서 권한 부여를 새로 고칠 수 있습니다. 또는 디바이스에서 예정대로 통신할 때까지 기다릴 수 있습니다.

선택 사항으로 HTTP 프록시를 구성할 수 있습니다.

최소 90일마다 Firepower 4100/9300 섀시가 직접 또는 HTTP 프록시를 통해 인터넷에 연결되어야 합니다. 일반 라이선스 통신은 30일마다 이루어지지만, 유예 기간이 있으므로 디바이스는 최대 90일간

콜 홈 없이 작동할 수 있습니다. 유예 기간이 지난 후 Licensing Authority에 연락해야 합니다. 아니면 특별 라이선스가 필요한 기능의 구성을 변경할 수 없습니다. 이를 제외하면 작동에 영향을 미치지 않습니다.

참고 디바이스가 1년 동안 License Authority와 통신할 수 없는 경우, 디바이스는 등록되지 않은 상태로 전환되지만 이전에 활성화된 강력한 암호화 기능은 손실되지 않습니다.

규정 위반 상태

디바이스는 다음과 같은 상황에서 규정 위반이 될 수 있습니다.

- 과다 사용 — 디바이스에서 사용 불가한 라이선스를 사용할 경우.
- 라이선스 만료—한시적인 라이선스가 만료된 경우.
- 통신 부재 — 디바이스에서 권한 재부여를 위해 Licensing Authority에 연결하지 못한 경우.

어카운트가 컴플라이언스 미준수 상태인지 또는 컴플라이언스 미준수 상태에 근접한지를 확인하려면 Firepower 4100/9300 섀시에서 현재 사용 중인 엔타이틀먼트와 Smart Account의 엔타이틀먼트를 비교해야 합니다.

컴플라이언스 미준수 상태에서는 특수 라이선스가 필요한 기능의 구성을 변경할 수는 없지만 작업은 달리 영향을 받지 않습니다. 예를 들어 표준 라이선스 한도를 초과하는 기존 컨텍스트를 계속 실행할 수 있으며 해당 구성을 수정할 수는 있지만 새 컨텍스트를 추가할 수는 없습니다.

Smart Call Home 인프라

기본적으로, Smart Call Home 프로파일은 Licensing Authority의 URL을 지정하는 FXOS 구성에 있습니다. 이 프로필을 제거할 수 없습니다. License 프로필의 유일한 구성 옵션은 License Authority의 대상 주소 URL입니다. Cisco TAC에서 지시하지 않는 한 License Authority URL을 변경해서는 안 됩니다.

참고 Cisco 전송 게이트웨이는 Firepower 4100/9300 보안 어플라이언스에서 지원되지 않습니다.

Cisco Success Network

Cisco Success Network는 사용자가 활성화하는 클라우드 서비스입니다. Cisco Success Network를 활성화하는 경우, Firepower 4100/9300 섀시과 Cisco Cloud 사이에 보안 연결이 설정되어 사용 정보와 통계가 스트리밍됩니다. 스트리밍 텔레메트리는 ASA에서 관심 있는 데이터를 선택하고 이를 구조화된 형식으로 원격 관리 스테이션에 전송하여 다음을 수행하는 메커니즘을 제공합니다.

- 네트워크에서 제품의 효율성을 향상시킬 수있는 활용 가능한 미사용 기능을 알려줍니다.
- 제품에 사용할 수 있는 추가 기술 지원 서비스 및 모니터링에 대해 알려줍니다.

- Cisco가 제품을 개선하도록 돕습니다.

Firepower 4100/9300을 Cisco Smart Software Manager에 등록할 때 Cisco Success Network를 활성화하십시오. 라이센스 기관에 Firepower 4100/9300 섀시을 등록합니다., 36 페이지의 내용을 참조하십시오.

다음 조건이 모두 충족되는 경우에만 Cisco Success Network에 등록할 수 있습니다.

- 스마트 소프트웨어 라이선스가 등록되었습니다.
- 스마트 라이선스 위성 모드가 비활성화되었습니다.
- 영구 라이선스가 비활성화되었습니다.

Cisco Success Network에 등록하면 섀시가 항상 보안 연결을 설정하고 유지합니다. Cisco Success Network를 비활성화하여 언제든지 이 연결을 끌 수 있으며, 이 경우 디바이스와Cisco Success Network 클라우드의 연결이 끊어집니다.

Cisco Success Network 현재 등록 상태를 시스템 > 라이센싱 > **Cisco Success Network**페이지에서 볼 수 있으며, 등록 상태를 변경할 수 있습니다. Cisco Success Network 등록 변경, 37 페이지의 내용을 참조하십시오.

Cisco Success Network 텔레메트리 데이터

Cisco Success Network를 사용하면 등록된 섀시가 구성 및 운영 상태 정보를 24시간마다 Cisco Success Network 클라우드로 스트리밍할 수 있습니다. 수집 및 모니터되는 데이터는 다음과 같습니다.

- 등록 디바이스 정보— Firepower 4100/9300 섀시 모델명, 제품 ID, 일련번호, UUID, 시스템 가동 시간 및 스마트 라이선싱 정보를 포함합니다. 등록된 디바이스 데이터, 25 페이지의 내용을 참조하십시오.
- 소프트웨어 정보 - Firepower 4100/9300 섀시에서 실행 중인 소프트웨어의 유형 및 버전 번호입니다. 소프트웨어 버전 데이터, 25 페이지의 내용을 참조하십시오.
- **ASA** 디바이스 정보 — Firepower 4100/9300의 보안 모듈/엔진에서 실행 중인 ASA 디바이스에 대한 정보. Firepower 4100 Series의 경우 단일 ASA 디바이스에 대한 정보만 포함된다는 점에 유의하십시오. ASA 디바이스 정보에는 각 디바이스, 디바이스 모델, 일련 번호 및 소프트웨어 버전에 사용 중인 스마트 라이선스가 포함됩니다. ASA 디바이스 데이터, 26 페이지의 내용을 참조하십시오.
  - 성능 정보 — ASA 디바이스의 시스템 가동 시간, CPU 사용량, 메모리 사용량, 디스크 공간 사용량 및 대역폭 사용량 정보입니다. 성능 데이터, 26 페이지의 내용을 참조하십시오.
  - 사용 정보 — 기능 상태, 클러스터, 장애 조치 및 로그인 정보:
    - 기능 상태 — 사용자가 구성했거나 기본적으로 활성화되어 있는 활성화된 ASA 기능의 목록입니다.
    - 클러스터 정보 — ASA 디바이스가 클러스터 모드인 경우 클러스터 정보를 포함합니다. ASA 디바이스가 클러스터 모드가 아닌 경우 이 정보가 표시되지 않습니다. 클러스터 정보에는 ASA 디바이스의 클러스터 그룹 이름, 클러스터 인터페이스 모드, 유닛 이름

및 상태가 포함됩니다. 동일한 클러스터에 있는 다른 피어 ASA 디바이스의 경우, 이름, 상태 및 일련 번호가 정보에 포함됩니다.

- 장애 조치 정보 — ASA가 장애 조치 모드인 경우 장애 조치 정보를 포함합니다. ASA가 장애 조치 모드가 아닌 경우 이 정보가 표시되지 않습니다. 장애 조치 정보에는 ASA의 역할 및 상태, 피어 ASA 디바이스의 역할, 상태 및 일련 번호가 포함됩니다.
- 로그인 기록 — 사용자 로그인 빈도, 로그인 시간 및 ASA 디바이스에서 가장 최근에 성공한 로그인 날짜 스탬프입니다. 그러나 로그인 기록에는 사용자 로그인 이름, 자격 증명 또는 기타 개인 정보가 포함되지 않습니다.

자세한 내용은 사용량 데이터, 27 페이지를 참조하십시오.

등록된 디바이스 데이터

Firepower 4100/9300 섀시를 Cisco Success Network에 등록한 경우, 섀시에 대한 일부 텔레메트리 데이터가 Cisco 클라우드로 스트리밍됩니다. 다음 표는 수집 및 모니터링된 소프트웨어 정보에 관해 설명합니다.

표 *1:* 등록된 디바이스 텔레메트리 데이터

데이터 포인트	예제 값
디바이스 모델	Cisco Firepower FP9300 Security 어플라이언스
Serial number(일련 번호)	GMX1135L01K
스마트 라이선스 PIID	752107e9-e473-4916-8566-e26d0c4a5bd9
스마트 라이선스 가상 어카운트 이름	FXOS-일반
시스템 가동 시간	32115
UDI 제품 ID	FPR-C9300-AC

소프트웨어 버전 데이터

Cisco Success Network는 유형 및 소프트웨어 버전을 포함하여 섀시와 관련된 소프트웨어 정보를 수집합니다. 다음 표는 수집 및 모니터링된 소프트웨어 정보에 관해 설명합니다.

표 *2:* 소프트웨어 버전 텔레메트리 데이터

데이터 포인트	예제 값
Type(유형)	package_version
버전	2.7(1.52)

ASA 디바이스 데이터

Cisco Success Network는 Firepower 4100/9300의 보안 모듈/엔진에서 실행 중인 ASA 디바이스에 대한 정보를 수집합니다. 다음 표는 ASA 디바이스에 대해 수집 및 모니터링된 소프트웨어 정보를 설명합니다.

표 *3: ASA* 디바이스 텔레메트리 데이터

데이터 포인트	예제 값
ASA 디바이스 PID	FPR9K-SM-36
ASA 디바이스 모델	Cisco Adaptive Security Appliance
ASA 디바이스 일련 번호	XDQ311841WA
구축 유형(기본 또는 컨테이너)	네이티브
보안 상황 모드(단일 또는 다중)	단일

ASA 소프트웨어 버전	{
	type: "asa_version",
	ersion: "9.13.1.5"
	}

디바이스 매니저 버전	{
	type: "device_mgr_version",
	version: "7.10.1"
	}

활성화된 스마트 라이선스 사용 중	{
	"type": "Strong encryption",
	"tag":
	"regid.2016-05.com.cisco.ASA-GEN-STRONG-ENCRYPTION,
	5.7_982308k4-74w2-5f38-64na-707q99g10cce",
	"count": 1
	}

성능 데이터

Cisco Success Network는 ASA 디바이스에 대한 성능 관련 정보를 수집합니다. 정보에는 시스템 가동 시간, CPU 사용량, 메모리 사용량, 디스크 공간 사용량 및 대역폭 사용량 정보가 포함됩니다.

- **CPU usage** - 지난 5분 동안의 CPU 사용량 정보
- 메모리 사용량 - 시스템의 여유, 사용 및 총 메모리
- 디스크 사용량 - 여유 공간, 사용된 공간, 총 디스크 공간 정보
- 시스템 **uptime(**실행시간**)** - 시스템 uptime(실행시간) 정보
- **Bandwidth(**대역폭**)** 사용량 - 시스템 Bandwidth 사용량. 모든 nameif-ed 인터페이스에서 집계 시스템 가동 이후 초당 수신 및 전송된 패킷(또는 바이트)에 대한 통계를 보여줍니다.

다음 표는 수집 및 모니터링된 소프트웨어 정보에 관해 설명합니다.

표 4: 성능 텔레메트리 데이터

데이터 포인트	예제 값

지난 5분 동안의 시스템 CPU 사용량	{
	“fiveSecondsPercentage”:0.2000000,
	“oneMinutePercentage”: 0,
	“fiveMinutesPercentage”: 0
	
	}

시스템 메모리 사용량	{
	“freeMemoryInBytes”:225854966384,
	“usedMemoryInBytes”: 17798281616,
	“totalMemoryInBytes”:243653248000
	
	}

시스템 디스크 사용량	{
	"freeGB": 21.237285,
	"usedGB": 0.238805,
	"totalGB": 21.476090
	
	}

시스템 가동 시간	99700000

시스템 bandwidth(대역폭) 사용량	{
	"receivedPktsPerSec": 3,
	"receivedBytesPerSec": 212,
	"transmittedPktsPerSec": 3,
	"transmittedBytesPerSec": 399
	}

사용량 데이터

Cisco Success Network는 섀시의 보안 모듈/엔진에서 실행 중인 ASA 디바이스에 대한 기능 상태, 클러스터, 장애 조치 및 로그인 정보를 수집합니다. 다음 표는 ASA 디바이스 사용에 대해 수집 및 모니터링된 데이터를 설명합니다.

표 5: 텔레메트리 데이터 사용

데이터 포인트	예제 값

Feature status(기능 상태)	[{
	“name”: “cluster”,
	“status”: “enabled”
	},
	{
	“name”: “webvpn”,
	“status”: “enabled”
	},
	{
	“name”: “logging-buffered”,
	“status”: “debugging”
	}]

데이터 포인트	예제 값
Cluster information(클러스터 정보)	{ "clusterGroupName": "asa-cluster", "interfaceMode": "spanned", "unitName": "unit-3-3", "unitState": "SLAVE", "otherMembers": { "items": [{ "memberName": "unit-2-1", "memberState": "MASTER", "memberSerialNum": "DAK391674E" }] } }
Failover information(장애 조치 정보)	{ myRole: "Primary", peerRole: "Secondary", myState: "active", peerState: "standby", peerSerialNum: "DAK39162B" }
로그인 기록	{ "loginTimes": "1 times in last 1 days", "lastSuccessfulLogin": "12:25:36 PDT Mar 11 2019" }

텔레메트리 예시 파일

Firepower 4100/9300 섀시는 데이터를 Cisco 클라우드로 전송하기 전에 텔레메트리가 활성화되고 섀시 관련 정보 및 추가 필드와 함께 온라인 상태인 모든 ASA 디바이스에서 수신한 데이터를 집계합니다. 텔레메트리 데이터가 포함된 애플리케이션이 없는 경우에도 섀시 정보와 함께 텔레메트리가 Cisco 클라우드로 전송됩니다.

다음은 Firepower 9300의 ASA 디바이스 2개에 대해 Cisco 클라우드로 전송된 정보를 포함하는 Cisco Success Network 텔레메트리 파일의 예입니다.

```
{
  "version": "1.0",
  "metadata": {
    "topic": "ASA.telemetry",
    "contentType": "application/json",
    "msgID": "2227"
  },
  "payload": {
    "recordType": "CST_ASA",
    "recordVersion": "1.0",
    "recordedAt": 1560868270055,
    "FXOS": {
      "FXOSdeviceInfo": {
        "deviceModel": "Cisco Firepower FP9300 Security Appliance",
        "serialNumber": "HNY4475P01K",
```

```
    "smartLicenseProductInstanceIdentifier": "413509m0-f952-5822-7492-r62c0a5h4gf4",
    "smartLicenseVirtualAccountName": "FXOS-general",
    "systemUptime": 32115,
    "udiProductIdentifier": "FPR-C9300-AC"
  },
  "versions": {
    "items": [
      {
        "type": "package_version",
        "version": "2.7(1.52)"
      }
    ]
  }
},
"asaDevices": {
  "items": [
    {
      "CPUUsage": {
        "fiveMinutesPercentage": 0,
        "fiveSecondsPercentage": 0,
        "oneMinutePercentage": 0
      },
      "bandwidthUsage": {
        "receivedBytesPerSec": 1,
        "receivedPktsPerSec": 0,
        "transmittedBytesPerSec": 1,
        "transmittedPktsPerSec": 0
      },
      "deviceInfo": {
        "deploymentType": "Native",
        "deviceModel": "Cisco Adaptive Security Appliance",
        "securityContextMode": "Single",
        "serialNumber": "ADG2158508T",
        "systemUptime": 31084,
        "udiProductIdentifier": "FPR9K-SM-24"
      },
      "diskUsage": {
        "freeGB": 19.781810760498047,
        "totalGB": 20.0009765625,
        "usedGB": 0.21916580200195312
      },
      "featureStatus": {
        "items": [
          {
            "name": "aaa-proxy-limit",
            "status": "enabled"
          },
          {
            "name": "firewall_user_authentication",
            "status": "enabled"
          },
          {
            "name": "IKEv2 fragmentation",
            "status": "enabled"
          },
          {
            "name": "inspection-dns",
            "status": "enabled"
          },
          {
            "name": "inspection-esmtp",
            "status": "enabled"
          },
          {
```

```
  "name": "inspection-ftp",
  "status": "enabled"
},
{
  "name": "inspection-hs232",
  "status": "enabled"
},
{
  "name": "inspection-netbios",
  "status": "enabled"
},
{
  "name": "inspection-rsh",
  "status": "enabled"
},
{
  "name": "inspection-rtsp",
  "status": "enabled"
},
{
  "name": "inspection-sip",
  "status": "enabled"
},
{
  "name": "inspection-skinny",
  "status": "enabled"
},
{
  "name": "inspection-snmp",
  "status": "enabled"
},
{
  "name": "inspection-sqlnet",
  "status": "enabled"
},
{
  "name": "inspection-sunrpc",
  "status": "enabled"
},
{
  "name": "inspection-tftp",
  "status": "enabled"
},
{
  "name": "inspection-xdmcp",
  "status": "enabled"
},
{
  "name": "management-mode",
  "status": "normal"
},
{
  "name": "mobike",
  "status": "enabled"
},
{
  "name": "ntp",
  "status": "enabled"
},
{
  "name": "sctp-engine",
  "status": "enabled"
},
{
```

```
          "name": "smart-licensing",
          "status": "enabled"
        },
        {
          "name": "static-route",
          "status": "enabled"
        },
        {
          "name": "threat_detection_basic_threat",
          "status": "enabled"
        },
        {
          "name": "threat_detection_stat_access_list",
          "status": "enabled"
        }
      ]
    },
    "licenseActivated": {
      "items": []
    },
    "loginHistory": {
      "lastSuccessfulLogin": "05:53:18 UTC Jun 18 2019",
      "loginTimes": "1 times in last 1 days"
    },
    "memoryUsage": {
      "freeMemoryInBytes": 226031548496,
      "totalMemoryInBytes": 241583656960,
      "usedMemoryInBytes": 15552108464
    },
    "versions": {
      "items": [
        {
          "type": "asa_version",
          "version": "9.13(1)248"
        },
        {
          "type": "device_mgr_version",
          "version": "7.13(1)31"
        }
      ]
    }
  },
  {
    "CPUUsage": {
      "fiveMinutesPercentage": 0,
      "fiveSecondsPercentage": 0,
      "oneMinutePercentage": 0
    },
    "bandwidthUsage": {
      "receivedBytesPerSec": 1,
      "receivedPktsPerSec": 0,
      "transmittedBytesPerSec": 1,
      "transmittedPktsPerSec": 0
    },
    "deviceInfo": {
      "deploymentType": "Native",
      "deviceModel": "Cisco Adaptive Security Appliance",
      "securityContextMode": "Single",
      "serialNumber": "RFL21764S1D",
      "systemUptime": 31083,
      "udiProductIdentifier": "FPR9K-SM-24"
    },
    "diskUsage": {
      "freeGB": 19.781543731689453,
```

```
  "totalGB": 20.0009765625,
  "usedGB": 0.21943283081054688
},
"featureStatus": {
  "items": [
    {
      "name": "aaa-proxy-limit",
      "status": "enabled"
    },
    {
      "name": "call-home",
      "status": "enabled"
    },
    {
      "name": "crypto-ca-trustpoint-id-usage-ssl-ipsec",
      "status": "enabled"
    },
    {
      "name": "firewall_user_authentication",
      "status": "enabled"
    },
    {
      "name": "IKEv2 fragmentation",
      "status": "enabled"
    },
    {
      "name": "inspection-dns",
      "status": "enabled"
    },
    {
      "name": "inspection-esmtp",
      "status": "enabled"
    },
    {
      "name": "inspection-ftp",
      "status": "enabled"
    },
    {
      "name": "inspection-hs232",
      "status": "enabled"
    },
    {
      "name": "inspection-netbios",
      "status": "enabled"
    },
    {
      "name": "inspection-rsh",
      "status": "enabled"
    },
    {
      "name": "inspection-rtsp",
      "status": "enabled"
    },
    {
      "name": "inspection-sip",
      "status": "enabled"
    },
    {
      "name": "inspection-skinny",
      "status": "enabled"
    },
    {
      "name": "inspection-snmp",
      "status": "enabled"
```

```
      },
      {
        "name": "inspection-sqlnet",
        "status": "enabled"
      },
      {
        "name": "inspection-sunrpc",
        "status": "enabled"
      },
      {
        "name": "inspection-tftp",
        "status": "enabled"
      },
      {
        "name": "inspection-xdmcp",
        "status": "enabled"
      },
      {
        "name": "management-mode",
        "status": "normal"
      },
      {
        "name": "mobike",
        "status": "enabled"
      },
      {
        "name": "ntp",
        "status": "enabled"
      },
      {
        "name": "sctp-engine",
        "status": "enabled"
      },
      {
        "name": "smart-licensing",
        "status": "enabled"
      },
      {
        "name": "static-route",
        "status": "enabled"
      },
      {
        "name": "threat_detection_basic_threat",
        "status": "enabled"
      },
      {
        "name": "threat_detection_stat_access_list",
        "status": "enabled"
      }
    ]
  },
  "licenseActivated": {
    "items": []
  },
  "loginHistory": {
    "lastSuccessfulLogin": "05:53:16 UTC Jun 18 2019",
    "loginTimes": "1 times in last 1 days"
  },
  "memoryUsage": {
    "freeMemoryInBytes": 226028740080,
    "totalMemoryInBytes": 241581195264,
    "usedMemoryInBytes": 15552455184
  },
  "versions": {
```

```
            "items": [
              {
                "type": "asa_version",
                "version": "9.13(1)248"
              },
              {
                "type": "device_mgr_version",
                "version": "7.13(1)31"
              }
            ]
          }
        }
      ]
    }
  }
}
```


Smart Software Licensing 사전 요구 사항

- 이 장은 Firepower 4100/9300 섀시의 ASA 논리적 디바이스에만 적용됩니다. FTD 논리적 디바이스의 라이선싱에 대한 자세한 내용은 FMC 구성 가이드를 참조하십시오.
- Cisco Smart Software Manager에서 마스터 계정을 만듭니다.

  https://software.cisco.com/#module/SmartLicensing

  아직 어카운트가 없는 경우 새 어카운트 설정 링크를 클릭합니다. Smart Software Manager에서 조직의 마스터 계정을 만들 수 있습니다.
- Cisco Commerce Workspace에서 라이선스를 1개 이상 구매합니다. 홈 페이지의 **Find Products and Solutions(**제품 및 솔루션 찾기**)** 검색 필드에서 사용 중인 플랫폼을 검색합니다. 일부 라이선스는 무료이지만 Smart Software Licensing 어카운트에 추가해야 합니다.
- 섀시가 Licensing Authority에 연결할 수 있도록 섀시에서 인터넷 액세스 또는 HTTP 프록시 액세스가 가능한지 확인합니다.
- 섀시에서 Licensing Authority의 이름을 확인할 수 있도록 DNS 서버를 구성합니다.
- 섀시의 시간을 설정합니다.
- ASA 라이선싱 엔타이틀먼트를 구성하기 전에 Firepower 4100/9300 섀시에 Smart Software Licensing 인프라를 구성합니다.

스마트 소프트웨어 라이센싱을 위한 지침

페일오버 및 클러스터링을 위한 **ASA** 지침

각 Firepower 4100/9300 섀시를 License Authority 또는 Satellite 서버에 등록해야 합니다. 보조 유닛에 대한 추가 비용은 없습니다. 영구 라이선스를 예약하려면 각 섀시용으로 별도의 라이선스를 구매해야 합니다.

Smart Software Licensing의 기본값

Firepower 4100/9300 섀시 기본 구성은 Smart Call Home 프로필인 "SLProf"를 포함하며, 이는 Licensing Authority의 URL을 지정합니다.

일반 Smart Software Licensing 구성

Cisco License Authority와 통신하기 위해 HTTP 프록시를 선택적으로 구성할 수 있습니다. License Authority에 등록하려면 Smart Software 라이선스 어카운트에서 얻은 Firepower 4100/9300 섀시에 등록 토큰 ID를 입력해야 합니다.

프로시저

단계 **1** (선택 사항) HTTP 프록시 구성, 35 페이지.

단계 **2** (선택 사항) Call Home URL 삭제, 36 페이지

단계 **3** 라이센스 기관에 Firepower 4100/9300 섀시을 등록합니다., 36 페이지.

(선택 사항) HTTP 프록시 구성

네트워크에서 인터넷 액세스에 HTTP 프록시를 사용할 경우 스마트 소프트웨어 라이선싱에 대해 프록시 주소를 구성해야 합니다. 일반적으로 이 프록시는 Smart Call Home에도 사용됩니다.

참고 인증이 있는 HTTP 프록시는 지원되지 않습니다.

프로시저

단계 **1** **System(**시스템**)** > **Licensing(**라이선싱**)** > **Call Home**을 선택합니다.

Call Home 페이지는 License Authority의 대상 주소 URL 구성 및 HTTP 프록시 구성을 위한 필드를 제공합니다.

참고 Cisco TAC에서 지시하지 않는 한 License Authority URL을 변경해서는 안 됩니다.

단계 **2** Server Enable(서버 활성화) 드롭다운 목록에서 **on(**설정**)**을 선택합니다.

단계 **3** **Server URL(**서버 **URL)** 및 **Server Port(**서버 포트**)** 필드에 프록시 IP 주소와 포트를 입력합니다. 이를테면 HTTPS 서버에 대해 포트 443을 입력합니다.

단계 4 **Save**(저장)를 클릭합니다.

(선택 사항) Call Home URL 삭제

앞에서 구성한 Call Home URL을 삭제하려면 다음 절차를 사용하십시오.

프로시저

단계 1 **System**(시스템) > **Licensing**(라이선싱) > **Call Home**을 선택합니다.

단계 2 **Call home Configuration**(**Call home** 구성) 영역에서 Delete(삭제)를 선택합니다.

라이센스 기관에 Firepower 4100/9300 섀시을 등록합니다.

Firepower 4100/9300 섀시를 등록할 때 License Authority에서는 Firepower 4100/9300 섀시와 License Authority의 통신을 위해 ID 인증서를 발급합니다. 또한 Firepower 4100/9300 섀시를 적절한 가상 계정에 지정합니다. 일반적으로 이 절차는 1회 수행됩니다. 그러나 이를테면 통신 문제 때문에 ID 인증서가 만료되면 나중에 Firepower 4100/9300 섀시를 다시 등록해야 할 수 있습니다.

프로시저

단계 1 Smart Software Manager 또는 Smart Software Manager Satellite에서 이 Firepower 4100/9300 섀시를 추가하려는 가상 어카운트에 대한 등록 토큰을 요청 및 복사합니다.

Smart Software Manager Satellite를 사용하여 등록 토큰을 요청하는 방법에 대한 자세한 내용은 Cisco Smart Software Manager Satellite 사용 설명서(https://www.cisco.com/c/en/us/buy/smart-accounts/software-manager-satellite.html)를 참조하십시오.

단계 2 Firepower Chassis Manager에서 **System**(시스템) > **Licensing**(라이선싱) > **Smart License**(스마트 라이선스)를 선택합니다.

단계 3 **Enter Product Instance Registration Token**(제품 인스턴스 등록 토큰 입력) 필드에 등록 토큰을 입력합니다.

단계 4 (선택 사항) **Enable Cisco Success Network** 체크 박스의 선택을 취소하여 Cisco Success Network 기능을 비활성화할 수 있습니다.

자세한 내용은 Cisco Success Network, 23 페이지를 참조하십시오.

단계 5 **Register**(등록)를 클릭합니다.

Firepower 4100/9300 섀시에서 License Authority 등록을 시도합니다.

디바이스의 등록을 취소하려면 **Unregister**(등록 취소)를 클릭합니다.

Firepower 4100/9300 섀시를 등록 취소하면 계정에서 디바이스가 제거됩니다. 디바이스의 모든 라이선스 엔타이틀먼트 및 인증서가 제거됩니다. 새 Firepower 4100/9300 섀시의 라이선스를 확보하기 위해 등록을 취소하는 경우가 있습니다. 또는 Smart Software Manager에서 해당 디바이스를 제거할 수 있습니다.

Cisco Success Network 등록 변경

Firepower 4100/9300을 Cisco Smart Software Manager에 등록할 때 Cisco Success Network를 활성화하십시오. 그 후 다음 절차를 사용하여 등록 상태를 확인 또는 변경합니다.

참고 Cisco Success Network는 평가판 모드에서 작동하지 않습니다.

프로시저

단계 1 시스템 > 라이선싱 > **Cisco Success Network**를 선택합니다**.**

단계 2 **Cisco Success Network** 기본 설정 아래에서, Cisco에서 제공하는 정보를 읽고 Cisco로 전송될 샘플 데이터를 확인하려면 여기를 클릭을 클릭합니다.

단계 3 **Cisco Success Network** 활성화 여부를 선택하고 **Save(**저장**)**를 클릭합니다.

Smart License Satellite Server 구성 Firepower 4100/9300 섀시

다음 절차는 Smart Licence Satellite 서버를 사용하도록 Firepower 4100/9300 섀시를 구성하는 방법을 보여줍니다.

시작하기 전에

- Smart Software Licensing 사전 요구 사항, 34 페이지에 나열된 모든 전제 조건을 완료합니다.
- Smart Software Satellite Server를 구축하고 설정합니다.

  Cisco.com에서 Smart License Satellite OVA 파일을 다운로드하고 VMwareESXi 서버에 이 파일을 설치 및 구성합니다. 자세한 내용은 Smart Software Manager Satellite 설치 가이드를 참고하십시오.
- Smart Software Satellite Server의 FQDN을 내부 DNSserver에서 확인할 수 있는지 확인합니다.
- 위성 트러스트 포인트가 이미 있는지를 확인합니다.

  scope security

  show trustpoint

트러스트 포인트는 FXOS 버전 2.4(1) 이상에서 기본적으로 추가됩니다. 트러스트 포인트가 없는 경우 다음 단계를 사용하여 트러스트 포인트 하나를 직접 추가해야 합니다.

1. http://www.cisco.com/security/pki/certs/clrca.cer로 이동한 다음, 구성하는 동안 액세스할 수 있는 위치에 SSL 인증서("-----BEGIN CERTIFICATE-----"부터 "-----END CERTIFICATE-----"까지)의 전체 본문을 복사합니다.
2. 보안 모드를 입력합니다.

   scope security
3. Trust Point를 생성하고 이름을 지정합니다.

   create trustpoint *trustpoint_name*
4. Trust Point의 인증서 정보를 지정합니다. 참고: 인증서는 Base64 암호화 X.509(CER) 형식이어야 합니다.

   set certchain *certchain*

   *certchain*변수의 경우 1단계에서 복사한 인증서 텍스트를 붙여넣습니다.

   명령에 인증서 정보를 지정하지 않은 경우, 루트 CA(Certificate Authority)에 인증 경로를 정의하는 신뢰 지점 목록 또는 인증서를 입력하라는 프롬프트가 표시됩니다. 해당 정보를 입력한 후 다음 행에 **ENDOFBUF** 를 입력하여 완료합니다.
5. 구성을 커밋합니다.

   commit-buffer

프로시저

단계 **1** **System(**시스템**)** > **Licensing(**라이선싱**)** > **Call Home**을 선택합니다.

단계 **2** **Call home Configuration(Call Home** 구성**)** 영역에서 **Address(**주소**)** 필드의 기본 URL을 이 절차의 사전 요구 사항에서 수집한 정보를 사용하는 Smart Software Satellite Server의 URL로 교체합니다 (**https://***[Satellite* 서버의 *FQDN]***/Transportgateway/services/DeviceRequestHandler** 형식 사용).

단계 **3** 라이센스 기관에 Firepower 4100/9300 섀시을 등록합니다., 36 페이지. Smart License Manager Satellite에서 등록 토큰을 요청하고 복사해야 합니다.

영구 라이선스 예약 구성

Firepower 4100/9300 섀시에 영구 라이선스를 할당할 수 있습니다. 이 범용 예약을 사용하면 디바이스에서 어떤 엔타이틀먼트라도 무제한 사용할 수 있습니다.

참고 시작하기 전에 Smart Software Manager에서 사용할 수 있도록 영구 라이선스를 구매해야 합니다. 모든 계정에 대해 영구 라이선스 예약이 승인되는 것은 아닙니다. 구성을 시도하기 전에 Cisco에서 이 기능에 대한 승인을 받았는지 확인하십시오.

영구 라이선스 설치

다음 절차는 Firepower 4100/9300 섀시에 영구 라이선스를 할당하는 방법을 보여줍니다.

프로시저

단계 **1** **System** > **Licensing** > **Permanent License**를 선택합니다.

단계 **2** **Generate**를 클릭하여 예약 요청 코드를 생성합니다. 예약 요청 코드를 클립보드에 복사합니다.

단계 **3** Cisco Smart Software Manager 포털의 Smart Software Manager Inventory(인벤토리) 화면으로 이동하여 **Licenses** 탭을 클릭합니다.

https://software.cisco.com/#SmartLicensing-Inventory

Licenses 탭에는 계정과 연결된 모든 기존 라이선스(일반 및 영구)가 표시됩니다.

단계 **4** **License Reservation**을 클릭하고, 생성된 예약 요청 코드를 상자에 붙여넣습니다.

단계 **5** **Reserve License** 버튼을 클릭합니다.

Smart Software Manager에서 인증 코드를 생성합니다. 코드를 다운로드하거나 클립보드로 복사할 수 있습니다. 이 시점에서는 Smart Software Manager에 따라 라이선스가 사용됩니다.

License Reservation 버튼이 표시되지 않으면 계정이 영구 라이선스 예약에 대해 인증되지 않은 것입니다. 이 경우 영구 라이선스 예약을 비활성화하고 일반 smart license 명령을 다시 입력해야 합니다.

단계 **6** Firepower Chassis Manager에서 **Authorization Code** 입력란에 생성된 인증 코드를 입력합니다.

단계 **7** **Install** 버튼을 클릭합니다.

Firepower 4100/9300 섀시에 PLR로 완전히 라이선스가 부여되면, Permanent License(영구 라이선스) 페이지에 라이선스 상태가 표시되고 영구 라이선스를 반환할 수 있는 옵션이 제공됩니다.

단계 **8** ASA 논리적 디바이스에서 기능 엔타이틀먼트를 활성화합니다. 엔타이틀먼트를 활성화하려면 ASA 라이선싱 장을 참조하십시오.

(선택 사항) 영구 라이선스 반환

영구 라이선스가 더 이상 필요하지 않으면 다음 절차를 사용하여 공식적으로 Smart Software Manager에 반환해야 합니다. 모든 단계를 수행하지 않으면 라이선스가 사용 중 상태로 유지되므로 다른 곳에서 사용할 수 없습니다.

프로시저

단계 **1** **System** > **Licensing** > **Permanent License**를 선택합니다.

단계 **2** **Return**을 클릭하여 반환 코드를 생성합니다. 반환 코드를 클립보드에 복사합니다.

Firepower 4100/9300 섀시의 라이선스가 즉시 취소되고 Evaluation(평가) 상태로 전환됩니다.

단계 **3** Smart Software Manager Inventory(인벤토리) 화면으로 이동하여 **Product Instances** 탭을 클릭합니다.

https://software.cisco.com/#SmartLicensing-Inventory

단계 **4** UDI(universal device identifier)를 사용하여 Firepower 4100/9300 섀시를 검색합니다.

단계 **5** **Actions** > **Remove**를 선택하고, 생성된 반환 코드를 상자에 붙여넣습니다.

단계 **6** **Remove Product Instance** 버튼을 클릭합니다.

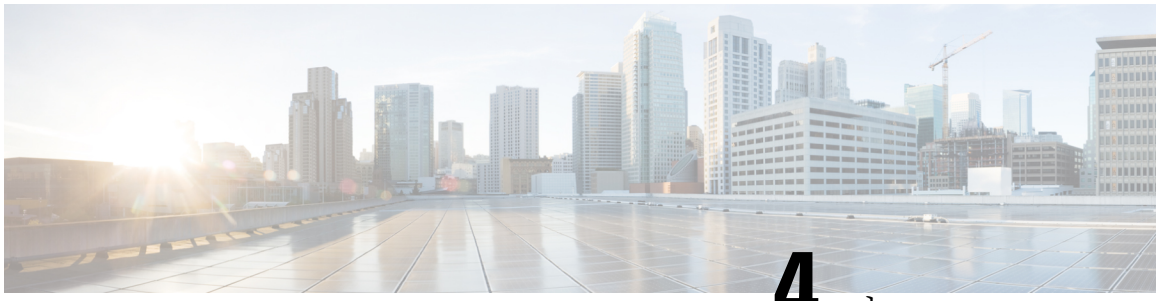
영구 라이선스가 사용 가능한 풀로 반환됩니다.

단계 **7** 시스템을 재부팅합니다. Firepower 4100/9300 섀시 리부팅 방법에 대한 자세한 내용은 Firepower 4100/9300 섀시 리부팅, 107 페이지 섹션을 참조하십시오.

Smart Software Licensing 기록

기능 이름	플랫폼 릴리스	설명
Cisco Success Network	2.7.1	Cisco Success Network는 사용자가 활성화하는 클라우드 서비스입니다. Cisco Success Network를 활성화하는 경우, Firepower 4100/9300 섀시과 Cisco Cloud 사이에 보안 연결이 설정되어 사용 정보와 통계가 스트리밍됩니다. 스트리밍 텔레메트리는 ASA에서 관심 있는 데이터를 선택하고 이를 구조화된 형식으로 원격 관리 스테이션에 전송하여 다음을 수행하는 메커니즘을 제공합니다. • 네트워크에서 제품의 효율성을 향상시킬 수있는 활용 가능한 미사용 기능을 알려줍니다. • 제품에 사용할 수 있는 추가 기술 지원 서비스 및 모니터링에 대해 알려줍니다. • Cisco가 제품을 개선하도록 돕습니다. Cisco Success Network에 등록하면 섀시가 항상 보안 연결을 설정하고 유지합니다. Cisco Success Network를 비활성화하여 언제든지 이 연결을 끌 수 있으며, 이 경우 디바이스와Cisco Success Network 클라우드의 연결이 끊어집니다. 다음 명령을 도입했습니다. **scope telemetry {enable | disable}** 추가된 화면: 시스템 > 라이센싱 > **Cisco Success Network**

기능 이름	플랫폼 릴리스	설명
Firepower 4100/9300 섀시의 Cisco 스마트 소프트웨어 라이선싱	1.1(1)	스마트 소프트웨어 라이선싱에서는 라이선스 풀을 구매하여 관리할 수 있습니다. 스마트 라이선스는 특정 일련 번호에 연결되지 않습니다. 각 유닛의 라이선스 키를 관리할 필요 없이 손쉽게 디바이스를 구축하거나 사용 중단할 수 있습니다. 또한 스마트 소프트웨어 라이선싱에서는 라이선스 사용량 및 필요량을 한눈에 볼 수 있습니다. Smart Software 라이선싱 구성은 Firepower 4100/9300 섀시 Supervisor(관리자)와 보안 모듈로 나뉩니다. 추가된 화면: **System(**시스템**)** > **Licensing(**라이선싱**)** > **Call Home** **System(**시스템**)** > **Licensing(**라이선싱**)** > **Smart License(**스마트 라이선스**)**



4 장

사용자 관리

사용자 계정

사용자 계정을 사용하여 시스템에 액세스합니다. 최대 48개의 로컬 사용자 계정을 구성할 수 있습니다. 각 사용자 계정에는 고유한 사용자 이름 및 비밀번호가 있어야 합니다.

관리자 어카운트

관리자 계정은 기본 사용자 계정이며 수정하거나 삭제할 수 없습니다. 이 어카운트는 시스템 관리자 또는 Superuser 어카운트이며 전체 권한을 가집니다. 관리자 어카운트에 할당된 기본 비밀번호가 없습니다. 초기 시스템 설정을 하는 동안 비밀번호를 선택해야 합니다.

관리자 어카운트는 항상 활성 상태이며 만료되지 않습니다. 관리자 어카운트는 비활성 상태로 구성할 수 없습니다.

로컬 인증 사용자 계정

로컬로 인증된 사용자 계정은 섀시를 통해 직접 인증되며 관리자 또는 AAA 권한을 보유한 사용자에 의해 활성화 또는 비활성화될 수 있습니다. 로컬 사용자 계정이 비활성화되면 사용자가 로그인할 수 없습니다. 비활성화된 로컬 사용자 계정에 대한 구성 세부사항은 데이터베이스에 의해 삭제되지 않습니다. 비활성화된 로컬 사용자 계정을 다시 활성화하면 계정이 기존 구성으로 다시 활성화됩니다. 그러나 계정 비밀번호는 재설정해야 합니다.

원격 인증 사용자 계정

원격으로 인증된 사용자 계정은 LDAP, RADIUS 또는 TACACS+를 통해 인증되는 사용자 계정입니다. 모든 원격 사용자는 초기에 기본적으로 **Read-Only(**읽기 전용**)** 역할이 할당됩니다.

사용자가 로컬 사용자 계정과 원격 사용자 계정을 동시에 유지할 경우 로컬 사용자 계정에 정의된 역할이 원격 사용자 계정의 역할을 재정의합니다.

폴백 인증 방법은 로컬 데이터베이스를 사용하는 것입니다. 이 폴백 방법은 구성할 수 없습니다.

원격 인증 지침, 그리고 원격 인증 공급자의 구성 및 삭제 방법에 대한 자세한 내용은 다음 항목을 참조하십시오.

- 원격 인증에 대한 지침, 46 페이지
- LDAP 제공자 구성, 144 페이지
- RADIUS 제공자 구성, 148 페이지
- TACACS+ 제공자 구성, 150 페이지

사용자 계정 만료

미리 정의된 시간에 만료하도록 사용자 계정을 구성할 수 있습니다. 만료 시간이 되면 사용자 계정은 비활성화됩니다.

기본적으로, 사용자 계정은 만료되지 않습니다.

만료일이 지정된 사용자 계정을 구성한 후에는 이 어카운트가 만료되지 않도록 재구성할 수 없습니다. 단, 최신 만료일이 있는 어카운트를 구성할 수 있습니다.

사용자 이름 지침

사용자 이름은 Firepower Chassis Manager 및 FXOS CLI의 로그인 ID로도 사용됩니다. 사용자 계정에 로그인 ID를 할당할 때 다음 지침 및 제한 사항을 고려합니다.

- 로그인 ID는 1~32자로 구성하며 다음을 포함할 수 있습니다.
  - 알파벳 문자
  - 숫자
  - _(밑줄)

- -(대시)
- . (점)

- 로그인 ID는 고유해야 합니다.
- 로그인 ID는 알파벳 문자로 시작해야 합니다. 숫자 또는 밑줄과 같은 특수 문자로 시작할 수 없습니다.
- 로그인 ID는 대/소문자를 구분합니다.
- 모두 숫자인 로그인 ID를 생성할 수 없습니다.
- 사용자 계정을 생성한 후, 로그인 ID를 변경할 수 없습니다. 사용자 계정을 삭제하고 새로 만들어야 합니다.

비밀번호 지침

로컬에서 인증되는 각 사용자 계정에는 비밀번호가 필요합니다. 관리자 또는 AAA 권한이 있는 사용자는 사용자 비밀번호에 대한 비밀번호 보안 수준을 확인하도록 시스템을 구성할 수 있습니다. 비밀번호 길이 검사를 활성화하면 각 사용자는 강력한 비밀번호를 사용해야 합니다.

각 사용자가 강력한 비밀번호를 사용하는 것이 좋습니다. 로컬로 인증된 사용자를 위해 비밀번호 보안 수준 확인을 활성화한 경우, FXOS에서는 다음 요건을 충족하지 않는 비밀번호를 거부합니다.

- 8자 이상, 127자 이하여야 합니다.

참고 Common Criteria 요구 사항을 준수하기 위해 시스템에서 최소 15자 비밀번호 길이를 선택적으로 구성할 수 있습니다. 자세한 내용은 최소 비밀번호 길이 확인 구성, 56 페이지를 참고하십시오.

- 하나 이상의 알파벳 대문자를 포함해야 합니다.
- 하나 이상의 알파벳 소문자를 포함해야 합니다.
- 하나 이상의 영숫자 외 문자(특수 문자)를 포함해야 합니다.
- 공백을 포함할 수 없습니다.
- aaabbb와 같이 한 문자가 3번 이상 연속적으로 나와서는 안 됩니다.
- 어떤 순서로든 3개의 연속 숫자 또는 문자를 포함해서는 안 됩니다(예: passwordABC 또는 password321).
- 사용자 이름 또는 사용자 이름을 반대로 한 이름과 동일하지 않아야 합니다.
- 비밀번호 딕셔너리 검사를 통과해야 합니다. 예를 들어, 비밀번호는 표준 사전 단어에 기반을 둘 수 없습니다.

- 다음 기호는 포함할 수 없습니다. 예: $(달러 기호), ? (물음표) 및 =(등호)

참고 이 제한 사항은 비밀번호 보안 수준 확인의 활성화 여부를 적용합니다.

- 로컬 사용자 및 관리자 계정 비밀번호는 비어 있지 않아야 합니다.

원격 인증에 대한 지침

지원되는 원격 인증 서비스 중 하나가 시스템에 구성될 경우, Firepower 4100/9300 섀시에서 시스템과 통신할 수 있도록 그 서비스에 대한 제공자를 생성해야 합니다. 다음 지침은 사용자 인증에 영향을 미칩니다.

원격 인증 서비스의 사용자 계정

사용자 계정은 Firepower 4100/9300 섀시의 로컬에 두거나 원격 인증 서버에 둘 수 있습니다.

Firepower Chassis Manager 또는 FXOS CLI에서 원격 인증 서비스로 로그인한 사용자의 임시 세션을 볼 수 있습니다.

원격 인증 서비스의 사용자 역할

원격 인증 서버에 사용자 계정을 생성할 경우 그 계정은 Firepower 4100/9300 섀시에서 작업하는 데 필요한 역할을 포함하고 그 역할의 이름이 FXOS에서 사용되는 이름과 일치해야 합니다. 역할 정책에 따라 사용자가 로그인하지 못하거나 읽기 전용 권한만 가질 수도 있습니다.

원격 인증 제공자의 사용자 특성

RADIUS 및 TACACS+ 구성에서는 사용자가 Firepower Chassis Manager 또는 FXOS CLI에 로그인하는 원격 인증 제공자 각각에서 Firepower 4100/9300 섀시에 대한 사용자 속성을 구성해야 합니다. 이 사용자 특성은 각 사용자에 지정된 역할 및 로캘을 저장합니다.

사용자가 로그인하면 FXOS에서 다음을 수행합니다.

1. 원격 인증 서비스를 쿼리합니다.
2. 사용자를 검증합니다.
3. 사용자가 검증되면 해당 사용자에게 할당된 역할 및 로케일을 확인합니다.

다음 표에서는 FXOS에서 지원하는 원격 인증 제공자의 사용자 특성 요구 사항을 비교합니다.

인증 제공자	맞춤형 속성	스키마 확장	속성 **ID** 요구 사항
LDAP	선택 사항	다음 중 하나를 선택하여 수행할 수 있습니다. • LDAP 스키마를 확장하지 않고 요구 사항에 맞는 기존의 미사용 속성을 구성합니다. • LDAP 스키마를 확장하고 CiscoAVPair와 같은 고유한 이름으로 맞춤형 속성을 생성합니다.	Cisco LDAP 구현에서는 유니코드 형식의 속성이 필요합니다. CiscoAVPair 맞춤형 속성을 생성하려는 경우 속성 ID로 1.3.6.1.4.1.9.287247.1을 사용합니다. 샘플 OID가 다음 섹션에 나와 있습니다.
RADIUS	선택 사항	다음 중 하나를 선택하여 수행할 수 있습니다. • RADIUS 스키마를 확장하지 않고 요구 사항에 맞는 기존의 미사용 속성을 사용합니다. • RADIUS 스키마를 확장하고 cisco-avpair와 같은 고유한 이름으로 맞춤형 속성을 생성합니다.	Cisco RADIUS 구현의 벤더 ID는 009, 속성의 벤더 ID는 001입니다. 다음 구문의 예에서는 cisco-avpair 속성을 생성하려는 경우 여러 사용자 역할 및 로케일을 지정하는 방법을 보여줍니다. `shell:roles="admin,aaa" shell:locales="L1,abc"`. 여러 값을 구분하는 기호로 쉼표 ","를 사용합니다.
TACACS+	필수	스키마를 확장하고 cisco-av-pair라는 이름으로 맞춤형 속성을 생성해야 합니다.	cisco-av-pair 이름은 TACACS+ 제공자에 대한 속성 ID를 제공하는 문자열입니다. 다음 구문의 예에서는 cisco-av-pair 속성을 생성할 경우 여러 사용자 역할 및 로케일을 지정하는 방법을 보여줍니다. `cisco-av-pair=shell:roles="admin aaa" shell:locales*"L1 abc"`. cisco-av-pair 속성 구문에 별표(*)를 사용하면 로케일에 선택 사항 플래그를 지정합니다. 그러면 동일한 권한 부여 프로필을 사용하는 다른 Cisco 디바이스의 인증이 실패하지 않습니다. 여러 값을 구분하는 구분 기호로 공백을 사용합니다.

LDAP 사용자 속성에 대한 샘플 **OID**

다음은 맞춤형 CiscoAVPair 속성에 대한 샘플 OID입니다.

```
CN=CiscoAVPair,CN=Schema,
CN=Configuration,CN=X
objectClass: top
objectClass: attributeSchema
cn: CiscoAVPair
distinguishedName: CN=CiscoAVPair,CN=Schema,CN=Configuration,CN=X
instanceType: 0x4
uSNCreated: 26318654
attributeID: 1.3.6.1.4.1.9.287247.1
attributeSyntax: 2.5.5.12
isSingleValued: TRUE
showInAdvancedViewOnly: TRUE
adminDisplayName: CiscoAVPair
adminDescription: UCS User Authorization Field
oMSyntax: 64
lDAPDisplayName: CiscoAVPair
name: CiscoAVPair
objectCategory: CN=Attribute-Schema,CN=Schema,CN=Configuration,CN=X
```

사용자 역할

시스템에는 다음과 같은 사용자 역할이 포함됩니다.

관리자

전체 시스템에 대한 완전한 읽기 및 쓰기 액세스가 가능합니다. 기본 관리자 계정이 기본적으로 이 역할에 할당되며 변경할 수 없습니다.

읽기 전용

시스템 구성에 대한 읽기 전용 액세스로, 시스템 상태를 수정할 권한이 없습니다.

운영

NTP 구성, Smart Licensing에 대한 Smart Call Home 구성, 시스템 로그(syslog 서버 및 장애 포함)에 대한 읽기 및 쓰기 액세스. 나머지 시스템에 대한 읽기 액세스 권한입니다.

AAA 관리자

사용자, 역할, AAA 구성에 대한 읽기-쓰기 액세스 권한입니다. 나머지 시스템에 대한 읽기 액세스 권한입니다.

로컬 인증 사용자에 대한 비밀번호 프로파일

비밀번호 프로파일에는 모든 로컬로 인증된 사용자에 대한 비밀번호 기록 및 비밀번호 변경 간격 속성이 포함되어 있습니다. 로컬에서 인증된 각 사용자에게는 다른 비밀번호 프로필을 지정할 수 없습니다.

비밀번호 기록 수

비밀번호 기록 수를 사용하면 로컬로 인증된 사용자가 동일한 비밀번호를 계속해서 재사용하는 것을 방지할 수 있습니다. 이 속성을 구성할 때, Firepower 섀시는 로컬로 인증된 사용자가 이전에 사용한 비밀번호를 최대 15개까지 저장합니다. 비밀번호는 최근 항목부터 시간순으로 저장되며, 기록 수 임계값에 도달할 경우 가장 오래된 비밀번호만 재사용될 수 있습니다.

사용자는 비밀번호를 재사용할 수 있기 전에 비밀번호 기록 수에 구성되어 있는 비밀번호 수를 생성하고 사용해야 합니다. 예를 들어, 비밀번호 기록 수를 8로 설정한 경우 로컬로 인증된 사용자는 9번째 비밀번호가 만료될 때까지 첫 번째 비밀번호를 재사용할 수 없습니다.

기본적으로 비밀번호 기록은 0으로 설정되어 있습니다. 이 값이 설정되면 기록 수를 비활성화하고 사용자가 언제든지 이전의 비밀번호를 재사용할 수 있습니다.

필요한 경우, 로컬로 인증된 사용자의 비밀번호 기록 수를 지우고 이전 비밀번호 재사용을 활성화할 수 있습니다.

비밀번호 변경 간격

비밀번호 변경 간격을 사용하면 로컬로 인증된 사용자가 지정된 시간 이내에 변경할 수 있는 비밀번호 변경 횟수를 제한할 수 있습니다. 다음 표는 비밀번호 변경 간격의 구성 옵션 2개를 설명합니다.

간격 구성	설명	예
비밀번호 변경 허용 안 됨	이 옵션을 사용하면 비밀번호 변경 이후 지정된 시간 동안 로컬로 인증된 사용자 비밀번호의 변경이 허용되지 않습니다. 변경 안 함 간격을 1~745시간으로 지정할 수 있습니다. 기본적으로, 변경 안 함 간격은 24시간입니다.	예를 들어, 로컬로 인증된 사용자가 비밀번호를 변경한 후 48시간 이내에 비밀번호가 변경되는 것을 방지하려면 다음을 설정합니다. • 해당 간격 동안 변경을 비활성화로 설정 • 변경 안 함 간격을 48시간으로 설정
변경 간격 내에 비밀번호 변경 허용됨	이 옵션은 로컬로 인증된 사용자가 미리 정의한 간격 동안 비밀번호를 변경할 수 있는 최대 횟수를 지정합니다. 변경 간격을 1~745시간으로 지정하고 비밀번호 변경 최대 횟수를 0~10으로 지정할 수 있습니다. 기본적으로, 로컬로 인증된 사용자는 48시간 동안 비밀번호 변경이 최대 2회 허용됩니다.	예를 들어, 로컬로 인증된 사용자가 비밀번호를 변경한 후 24시간 이내에 비밀번호를 최대 한 번 변경하도록 허용하려면 다음을 설정합니다. • 해당 간격 동안 변경을 활성화로 설정 • 변경 횟수를 1로 설정 • 변경 간격을 24로 설정

사용자 설정 구성

프로시저

단계 **1** **System(**시스템**)** > **User Management(**사용자 관리**)**를 선택합니다.

단계 **2** **Settings(**설정**)** 탭을 클릭합니다.

단계 **3** 다음 필드에 필수 정보를 입력합니다.

참고 **Default Authentication(**기본 인증**)** 및 **Console Authentication(**콘솔 인증**)**이 모두 동일한 원격 인증 프로토콜(RADIUS, TACACS+ 또는 LDAP)을 사용하도록 설정된 경우, 이러한 사용자 설정을 업데이트해야 해당 서버 구성의 특정 측면(예: 해당 서버 삭제 또는 할당 순서 변경)을 변경할 수 있습니다.

이름	설명
Default Authentication(기본 인증**)** 필드	사용자가 원격 로그인 중에 인증되는 기본 방법입니다. 다음 중 하나일 수 있습니다. • 로컬 — 사용자 계정이 섀시에서 로컬로 정의되어야 합니다. • **Radius** — 사용자 계정이 섀시에 지정된 RADIUS 서버에서 정의되어야 합니다. • **TACACS** — 사용자 계정이 섀시에 지정된 TACACS+ 서버에서 정의되어야 합니다. • **LDAP** — 사용자 계정이 섀시에 지정된 LDAP/MS-AD 서버에서 정의되어야 합니다. • **None(**없음**)** — 사용자 계정이 섀시에서 로컬인 경우, 사용자가 원격으로 로그인할 때 비밀번호가 필요하지 않습니다. 참고 모든 **Radius**, **TACACS** 및 **LDAP** 설정은 Platform Settings(플랫폼 설정)에서 구성해야 합니다. 자세한 내용은 플랫폼 설정 장의 AAA 정보, 142 페이지를 참조하십시오.

이름	설명
Console Authentication(콘솔 인증**)** 필드	콘솔 포트를 통해 FXOS CLI에 연결될 때 사용자를 인증하는 방식입니다. 다음 중 하나일 수 있습니다. • 로컬 — 사용자 계정이 섀시에서 로컬로 정의되어야 합니다. • **Radius** — 사용자 계정이 섀시에 지정된 RADIUS 서버에서 정의되어야 합니다. • **TACACS** — 사용자 계정이 섀시에 지정된 TACACS+ 서버에서 정의되어야 합니다. • **LDAP** — 사용자 계정이 섀시에 지정된 LDAP/MS-AD 서버에서 정의되어야 합니다. • **None(**없음**)** — 사용자 계정이 섀시에 대해 로컬인 경우, 사용자가 콘솔 포트를 사용하여 FXOS CLI에 연결할 때 비밀번호가 필요하지 않습니다.
원격 사용자 설정	
원격 사용자 역할 정책	사용자가 로그인을 시도하고 원격 인증 제공자가 인증 정보와 함께 사용자 역할을 제공하지 않는 경우, 발생하는 결과를 제어합니다. • **Assign Default Role(**기본 역할 지정**)**—사용자가 읽기 전용 사용자 역할로 로그인할 수 있습니다. • **No-Login(**로그인 안 함**)** — 사용자 이름 및 비밀번호가 올바른 경우에도 사용자가 시스템에 로그인할 수 없습니다.
로컬 사용자 설정	
Password Strength Check(비밀번호 길이 검사**)** 체크 박스	이 옵션을 선택하면 모든 로컬 사용자 비밀번호가 강력한 비밀번호의 지침을 따라야 합니다(비밀번호 지침, 45 페이지 참조). 기본적으로 강력한 비밀번호가 활성화되어 있습니다.
History Count(기록 수**)** 필드	사용자가 이전에 사용한 비밀번호를 재사용하기 전에 생성해야 하는 고유한 비밀번호 수입니다. 기록 수는 최근 항목부터 시간순으로 저장되며, 기록 수 임계값에 도달할 경우 가장 오래된 비밀번호만 재사용될 수 있습니다. 0 ~ 15의 어떤 값이든 가능합니다. **History Count(**기록 수**)** 필드를 0으로 설정하여 기록 수를 비활성화하고 사용자가 언제든지 이전에 사용한 비밀번호를 재사용하게 할 수 있습니다.

이름	설명
Change During Interval(해당 간격 동안 변경**)** 필드	로컬로 인증된 사용자가 비밀번호를 변경할 수 있는 시기를 제어합니다. 가능한 값은 다음과 같습니다. • **Enable(**활성화**)** — 로컬로 인증된 사용자는 변경 간격 및 변경 횟수에 대한 설정을 기초로 비밀번호를 변경할 수 있습니다. • **Disable(**비활성화**)** — 로컬로 인증된 사용자는 변경 안 함 간격 동안 지정된 시간 간격에 비밀번호를 변경할 수 없습니다.
Change Interval(변경 간격**)** 필드	**Change Count(**변경 횟수**)** 필드에 지정된 비밀번호 변경 횟수가 적용되는 시간입니다. 1시간 ~ 745시간의 어떤 값이든 가능합니다. 예를 들어, 이 필드가 48로 설정되고 **Change Count(**변경 횟수**)** 필드가 2로 설정된 경우 로컬로 인증된 사용자는 48시간 이내에 비밀번호를 최대 2번 변경할 수 있습니다.
Change Count(변경 수**)** 필드	로컬로 인증된 사용자가 변경 간격 동안 비밀번호를 변경할 수 있는 최대 횟수입니다. 0 ~ 10의 어떤 값이든 가능합니다.
No Change Interval(변경 안 함 간격**)** 필드	로컬로 인증된 사용자가 새로 생성된 비밀번호를 변경하기 전에 기다려야 하는 최소 시간입니다. 이 값은 1~745시간으로 선택할 수 있습니다. 이 간격은 **Change During Interval(**해당 간격 동안 변경**)** 속성이 **Disable(**비활성화**)**로 설정되지 않은 경우 무시됩니다.
암호문구 만료일 필드	1~9999일 사이의 만료일을 설정합니다. 기본적으로 만료는 비활성화되어 있습니다.
암호문구 만료 경고 기간 필드	로그인할 때마다 사용자에게 비밀번호 만료에 대해 경고할 만료 전 일수를 0~9999 범위에서 설정합니다. 기본값은 14일입니다.
만료 유예 기간 필드	비밀번호 만료 후 사용자가 비밀번호를 변경해야 하는 일수를 0~9999 사이로 설정합니다. 기본값은 3일입니다.
비밀번호 재사용 간격 필드	비밀번호를 재사용할 수 있는 기간(일)을 1~365 범위에서 설정합니다. 기본값은 15일입니다. **History Count(**기록 수**)**와 **Password Reuse Interval(**비밀번호 재사용 간격**)**을 모두 활성화하는 경우 두 요구사항을 모두 충족해야 합니다. 예를 들어 기록 수를 3으로 설정하고 재사용 간격을 10일로 설정한 경우, 10일이 경과하고 비밀번호를 3번 변경해야 비밀번호를 변경할 수 있습니다.

단계 **4** **Save(**저장**)**를 클릭합니다.

세션 시간 초과 구성

FXOS CLI를 사용하여 Firepower 4100/9300 섀시에서 사용자 세션을 종료할 때까지 사용자가 아무런 작업을 수행하지 않는 상태로 경과할 수 있는 시간을 지정할 수 있습니다. 콘솔 세션과 HTTPS, SSH, 텔넷 세션에 대해 각기 다른 설정을 구성할 수 있습니다.

최대 3600초(60분)의 시간 초과 값을 설정할 수 있습니다. 기본값은 600초입니다. 이 설정을 비활성화하려면 세션 시간 초과 값을 0으로 설정합니다.

프로시저

단계 **1** 보안 모드를 입력합니다.

Firepower-chassis #**scope security**

단계 **2** 기본 권한 부여 보안 모드를 입력합니다.

Firepower-chassis /security # **scope default-auth**

단계 **3** HTTPS, SSH 및 텔넷 세션에 대한 유휴 시간 초과를 설정합니다.

Firepower-chassis /security/default-auth # **set session-timeout** 초

단계 **4** (선택 사항) 콘솔 세션에 대한 유휴 시간 초과를 설정합니다.

Firepower-chassis /security/default-auth # **set con-session-timeout** 초

단계 **5** 시스템 구성에 트랜잭션을 커밋합니다.

Firepower-chassis /security/default-auth # **commit-buffer**

단계 **6** (선택 사항) 세션 및 절대 세션 시간 초과 설정을 봅니다.

Firepower-chassis /security/default-auth # **show detail**

예제**:**

```
Default authentication:
 Admin Realm: Local
 Operational Realm: Local
 Web session refresh period(in secs): 600
 Idle Session timeout (in secs) for web, ssh, telnet sessions: 600
 Absolute Session timeout (in secs) for web, ssh, telnet sessions: 3600
 Serial Console Session timeout(in secs): 600
 Serial Console Absolute Session timeout(in secs): 3600
 Admin Authentication server group:
 Operational Authentication server group:
 Use of 2nd factor: No
```

절대 세션 시간 초과 구성

Firepower 4100/9300 섀시에는 세션 사용과 상관없이 절대 세션 시간 초과 기간이 지나면 사용자 세션을 닫는 절대 세션 시간 초과 설정이 있습니다. 이 절대 시간 초과 기능은 시리얼 콘솔, SSH, HTTPS를 비롯한 모든 액세스 형식에서 전역적으로 적용됩니다.

시리얼 콘솔 세션의 절대 세션 시간 초과를 별도로 구성할 수 있습니다. 이렇게 하면 다른 형식의 액세스에 대한 시간 초과를 유지하면서 디버깅 요구에 대한 시리얼 콘솔 절대 세션 시간 초과를 비활성화할 수 있습니다.

절대 시간 초과 기본값은 3600초(60분)이며 FXOS CLI를 사용해 변경할 수 있습니다. 이 설정을 비활성화하려면 절대 세션 시간 초과 값을 0으로 설정합니다.

프로시저

단계 **1** 보안 모드를 입력합니다.

Firepower-chassis #**scope security**

단계 **2** 기본 권한 부여 보안 모드를 입력합니다.

Firepower-chassis /security # **scope default-auth**

단계 **3** 절대 세션 시간 초과를 설정합니다.

Firepower-chassis /security/default-auth # **set absolute-session-timeout** 초

단계 **4** (선택 사항) 별도의 콘솔 절대 세션 시간 초과를 설정합니다.

Firepower-chassis /security/default-auth # **set con-absolute-session-timeout** 초

단계 **5** 시스템 구성에 트랜잭션을 커밋합니다.

Firepower-chassis /security/default-auth # **commit-buffer**

단계 **6** (선택 사항) 세션 및 절대 세션 시간 초과 설정을 봅니다.

Firepower-chassis /security/default-auth # **show detail**

예제:

```
Default authentication:
 Admin Realm: Local
 Operational Realm: Local
 Web session refresh period(in secs): 600
 Idle Session timeout (in secs) for web, ssh, telnet sessions: 600
 Absolute Session timeout (in secs) for web, ssh, telnet sessions: 3600
 Serial Console Session timeout(in secs): 600
 Serial Console Absolute Session timeout(in secs): 3600
 Admin Authentication server group:
 Operational Authentication server group:
 Use of 2nd factor: No
```

최대 로그인 시도 횟수 설정

허용된 최대 횟수만큼 로그인 시도에 실패하면 지정된 시간 동안 사용자가 잠기도록 Firepower 4100/9300 섀시를 구성할 수 있습니다. 설정된 로그인 최대 시도 횟수를 초과하면 사용자가 시스템에서 잠깁니다. 사용자가 잠겼음을 나타내는 알림이 표시되지 않습니다. 이 경우 사용자는 다시 로그인을 시도하려면 지정된 시간 동안 기다려야 합니다.

최대 로그인 시도 횟수를 구성하려면 다음 단계를 수행하십시오.

참고

- 최대 로그인 시도 횟수를 초과하면 모든 유형의 사용자 계정(관리자 포함)이 시스템에서 잠깁니다.
- 기본 최대 로그인 시도 실패 횟수는 0입니다. 최대 로그인 시도 횟수를 초과한 후 사용자가 시스템에서 잠기는 기본 시간은 30분(1800초)입니다.
- 사용자의 잠금 상태를 보고 이를 지우기 위한 단계는 사용자 잠금 상태 보기 및 지우기, 56 페이지 섹션을 참조하십시오.

이 옵션은 시스템에서 Common Criteria 인증 컴플라이언스를 얻기 위해 제공되는 숫자 중 하나입니다. 자세한 내용은 보안 인증서 컴플라이언스, 71 페이지를 참고하십시오.

프로시저

단계 **1** FXOS CLI에서 보안 모드로 들어갑니다.

scope security

단계 **2** 최대 로그인 시도 실패 횟수를 설정합니다.

set max-login-attempts *num_attempts*

num_attempts 값은 0~10의 정수입니다.

단계 **3** 최대 로그인 시도 횟수에 도달한 후 사용자가 시스템에서 잠긴 상태로 유지되는 시간(초)을 지정합니다.

set user-account-unlock-time

unlock_time

단계 **4** 구성을 커밋합니다.

commit-buffer

사용자 잠금 상태 보기 및 지우기

관리자는 Maximum Number of Login Attempts(최대 로그인 시도 횟수) CLI 설정에 지정된 최대 로그인 시도 실패 횟수를 초과한 후 Firepower 4100/9300 섀시에서 잠긴 사용자의 잠금 상태를 확인하고 해제할 수 있습니다. 자세한 내용은 최대 로그인 시도 횟수 설정, 55 페이지을 참고하십시오.

프로시저

단계 **1** FXOS CLI에서 보안 모드로 들어갑니다.

scope security

단계 **2** 해당 사용자의 사용자 정보(잠금 상태 포함)를 표시합니다.

Firepower-chassis /security # **show local-user** *user* **detail**

예제**:**

```
Local User(□□ □□□) □□□:
 □□:
 □:
 □□□:
 □□:
 □□: □□□□ □□
Password:
 □□□ □□ □□: □□
 □□□□ □□: □□
 □□□ □□:
 □□: □□ □□
 □□□ SSH □□ □:
```

단계 **3** (선택 사항) 사용자의 잠금 상태를 지웁니다.

Firepower-chassis /security # **scope local-user** *user*

Firepower-chassis /security/local-user # **clear lock-status**

최소 비밀번호 길이 확인 구성

최소 비밀번호 길이 확인을 활성화하는 경우 지정된 최소 문자 수의 비밀번호를 만들어야 합니다. 예를 들어 *min_length* 옵션이 15로 설정된 경우 15자 이상을 사용해 비밀번호를 만들어야 합니다. 이 옵션은 시스템에서 Common Criteria 인증 컴플라이언스를 허용하는 숫자 중 하나입니다. 더 자세한 내용은 보안 인증서 컴플라이언스의 내용을 참조하십시오.

최소 비밀번호 길이 확인을 구성하려면 다음 단계를 수행하십시오.

프로시저

단계 **1** FXOS CLI에서 보안 모드로 들어갑니다.

scope security

단계 **2** 최소 비밀번호 길이를 지정합니다.

set min-password-length *min_length*

단계 **3** 구성을 커밋합니다.

commit-buffer

로컬 사용자 계정 생성

프로시저

단계 **1** **System(**시스템**)** > **User Management(**사용자 관리**)**를 선택합니다.

단계 **2** **Local Users(**로컬 사용자**)** 탭을 클릭합니다.

단계 **3** **Add User(**사용자 추가**)**를 클릭하여 **Add User(**사용자 추가**)** 대화 상자를 엽니다.

단계 **4** 사용자에 대한 필수 정보로 다음 필드를 완성합니다.

이름	설명
User Name(사용자 이름**)** 필드	계정 로그인에 사용하는 계정 이름. 이름은 고유해야 하며 사용자 계정 이름에 대한 지침 및 제한 사항을 따라야 합니다(사용자 이름 지침, 44 페이지 참조). 사용자를 저장하면 로그인 ID는 변경할 수 없습니다. 사용자 계정을 삭제하고 새로 만들어야 합니다.
First Name(이름**)** 필드	사용자의 이름입니다. 최대 32자입니다.
Last Name(성**)** 필드	사용자의 성입니다. 최대 32자입니다.
Email(이메일**)** 필드	사용자의 이메일 주소입니다.
Phone Number(전화번호**)** 필드	사용자의 전화 번호.

이름	설명
Password(비밀번호**)** 필드	이 계정의 비밀번호. 비밀번호 보안 수준 확인을 활성화하면 사용자의 비밀번호가 더욱 강력해지며, 보안 수준 확인 요건을 충족하지 않는 비밀번호를 FXOS에서 거부합니다(비밀번호 지침, 45 페이지 참조). 참고 비밀번호는 다음 기호는 포함할 수 없습니다. 예: $(달러 기호), ? (물음표) 및 =(등호) 이 제한 사항은 비밀번호 보안 수준 확인의 활성화 여부를 적용합니다.
Confirm Password(비밀번호 확인**)** 필드	확인을 위해 두 번째로 입력하는 비밀번호.
Account Status(계정 상태**)** 필드	상태가 **Active(**활성**)**로 설정된 경우, 사용자는 이 로그인 ID와 비밀번호를 사용하여 Firepower Chassis Manager 및 FXOS CLI에 로그인할 수 있습니다.
User Role(사용자 역할**)** 목록	사용자 계정에 할당할 수 있는 권한에 해당하는 역할입니다(사용자 역할, 48 페이지 참조). 모든 사용자에게 기본적으로 읽기 전용 역할이 할당되며 이 역할은 선택 취소할 수 없습니다. 여러 역할을 할당하려면 **Ctrl** 키를 누른 상태에서 원하는 역할을 클릭합니다. 참고 사용자 역할을 삭제하면 사용자의 현재 세션 ID가 취소됩니다. 즉, 사용자의 모든 활성 세션(CLI 및 웹)이 즉시 종료됩니다.
Account Expires(어카운트 만료**)** 체크 박스	이 체크 박스를 선택한 경우, 해당 계정은 만료되며 **Expiration Date(**만료일**)** 필드에 지정된 날짜 이후에 사용할 수 없습니다. 참고 만료일이 지정된 사용자 계정을 구성한 후에는 이 어카운트가 만료되지 않도록 재구성할 수 없습니다. 단, 최신 만료일이 있는 어카운트를 구성할 수 있습니다.
Expiry Date(만료일**)** 필드	계정이 만료되는 날. 날짜는 yyyy-mm-dd 형식이어야 합니다. 만료일을 선택하기 위해 달력을 보려면 이 필드의 마지막에 있는 달력 아이콘을 클릭합니다.

단계 **5** **Add(**추가**)**를 클릭합니다.

로컬 사용자 계정 삭제

프로시저

단계 **1** **System(**시스템**)** > **User Management(**사용자 관리**)**를 선택합니다.

단계 **2** **Local Users(**로컬 사용자**)** 탭을 클릭합니다.

단계 **3** 삭제하려는 사용자 계정 행에서 **Delete(**삭제**)**를 클릭합니다.

단계 **4** **Confirm(**확인**)** 대화 상자에서 **Yes(**예**)**를 클릭합니다.

로컬 사용자 계정 활성화 또는 비활성화

로컬 사용자 계정을 활성화하거나 비활성화하려면 사용자에게 관리자 또는 AAA 권한이 있어야 합니다.

프로시저

단계 **1** **System(**시스템**)** > **User Management(**사용자 관리**)**를 선택합니다.

단계 **2** **Local Users(**로컬 사용자**)** 탭을 클릭합니다.

단계 **3** 활성화 또는 비활성화하려는 사용자 계정 행에서 **Edit(**편집**)(**연필 모양 아이콘**))**을 클릭합니다.

단계 **4** **Edit User(**사용자 편집**)** 대화 상자에서 다음 중 하나를 수행합니다.

- 사용자 계정을 활성화하려면 **Account Status(**어카운트 상태**)** 필드에서 **Active(**활성**)** 라디오 버튼을 클릭합니다. 사용자 어카운트를 재활성화할 때 어카운트 비밀번호를 재설정해야 합니다.
- 사용자 계정을 비활성화하려면 **Account Status(**어카운트 상태**)** 필드에서 **Inactive(**비활성**)** 라디오 버튼을 클릭합니다.

관리자 사용자 계정은 항상 활성 상태로 설정됩니다. 수정할 수 없습니다.

단계 **5** **Save(**저장**)**를 클릭합니다.

단계 **6** 시스템 구성에 트랜잭션을 커밋합니다.

Firepower-chassis /security/local-user # **commit-buffer**

로컬로 인증된 사용자의 비밀번호 기록 지우기

프로시저

단계 **1** 보안 모드를 입력합니다.

Firepower-chassis #**scope security**

단계 **2** 지정된 사용자 계정에 대한 로컬 사용자 보안 모드를 입력합니다.

Firepower-chassis /security # **scope local-user** *user-name*

단계 **3** 지정된 사용자 계정에 대한 비밀번호 기록을 지웁니다.

Firepower-chassis /security/local-user # **clear password-history**

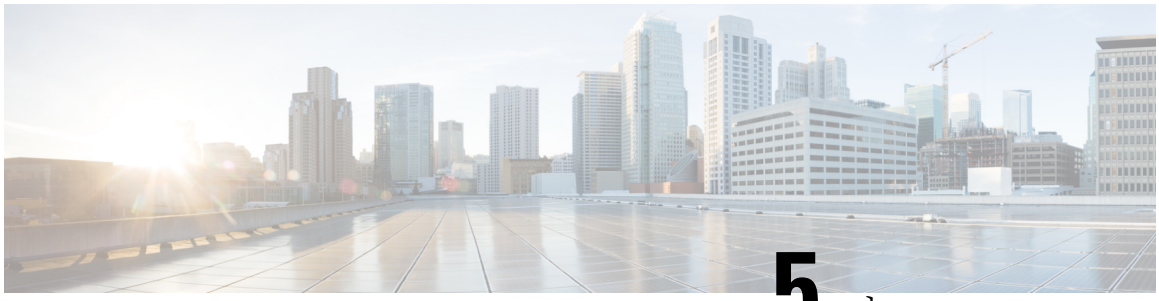
단계 **4** 시스템 구성에 트랜잭션을 커밋합니다.

Firepower-chassis /security/local-user # **commit-buffer**

예

다음 예에서는 비밀번호 기록을 지우고 트랜잭션을 커밋합니다.

```
Firepower-chassis # scope security
Firepower-chassis /security # scope local-user admin
Firepower-chassis /security/local-user # clear password-history
Firepower-chassis /security/local-user* # commit-buffer
Firepower-chassis /security/local-user #
```



5 장

이미지 관리

이미지 관리 정보

Firepower 4100/9300 섀시는 다음의 2가지 기본 이미지 유형을 사용합니다.

참고 모든 이미지는 보안 부팅을 통해 디지털로 서명되고 검증됩니다. 이미지를 수정하지 마십시오. 이미지를 수정하면 검증 오류를 수신하게 됩니다.

- 플랫폼 번들 — Firepower 플랫폼 번들은 수퍼바이저 및 보안 모듈/엔진에서 작동하는 여러 개별 이미지가 모여 있는 컬렉션입니다. 플랫폼 번들은 FXOS 소프트웨어 패키지입니다.
- 애플리케이션 - 애플리케이션 이미지는 Firepower 4100/9300 섀시의 보안 모듈/엔진에 구축할 소프트웨어 이미지입니다. 애플리케이션 이미지는 CSP(Cisco Secure Package) 파일로 전송되고 논리적 디바이스를 생성하거나 이후 논리적 디바이스 생성에 대비하기 위해 보안 모듈/엔진에 구축될 때까지 Supervisor(관리자)에 저장됩니다. 수퍼바이저에 저장된 동일한 애플리케이션 이미지 유형의 서로 다른 여러 버전을 둘 수 있습니다.

참고 플랫폼 번들 이미지와 하나 이상의 애플리케이션 이미지를 모두 업그레이드하려면 먼저 플랫폼 번들을 업그레이드해야 합니다.

참고 디바이스에 ASA 애플리케이션을 설치하는 경우, 기존 애플리케이션 FTD의 이미지를 삭제할 수 있으며 그 반대의 경우도 마찬가지입니다. 모든 FTD 이미지를 삭제하려고 하면, 기본 FTD/ASA APP가 남지 않으므로 `Invalid operation(잘못된 작업) 오류 메시지와 함께 하나 이상의 이미지 삭제가 거부됩니다. 새로운 기본 FTD 앱`을 선택하십시오. 모든 FTD 이미지를 삭제하려면, 기본 이미지를 그대로 두고 나머지 이미지를 삭제한 다음 마지막으로 기본 이미지를 삭제해야 합니다.

Cisco.com에서 이미지 다운로드

FXOS 및 애플리케이션 이미지를 Cisco.com에서 다운로드하여 섀시에 업로드할 수 있습니다.

시작하기 전에

Cisco.com 어카운트가 있어야 합니다.

프로시저

단계 **1** 웹 브라우저를 사용하여 http://www.cisco.com/go/firepower9300-software 또는 http://www.cisco.com/go/firepower4100-software로 이동합니다.
Firepower 4100/9300 섀시에 대한 소프트웨어 다운로드 페이지가 브라우저에서 열립니다.

단계 **2** 적절한 소프트웨어 이미지를 찾은 다음 로컬 컴퓨터에 다운로드합니다.

Security Appliance에 이미지 업로드

FXOS 및 애플리케이션 이미지를 섀시에 업로드할 수 있습니다.

시작하기 전에

업로드할 이미지를 로컬 컴퓨터에서 사용할 수 있는지 확인합니다.

프로시저

단계 **1** **System(**시스템**)** > **Updates(**업데이트**)**를 선택합니다.
Available Updates(사용 가능한 업데이트) 페이지는 섀시에서 사용 가능한 FXOS 플랫폼 번들 이미지와 애플리케이션 이미지 목록을 보여줍니다.

단계 **2** **Upload Image(**이미지 업로드**)**를 클릭하여 Upload Image(이미지 업로드) 대화 상자를 엽니다.

단계 **3** **Choose File(**파일 선택**)**을 클릭하여 업로드할 이미지로 이동한 다음 해당 이미지를 선택합니다.

단계 **4** **Upload(**업로드**)**를 클릭합니다.

선택한 이미지가 Firepower 4100/9300 섀시에 업로드됩니다. 이미지가 업로드되는 동안 시스템에는 완료된 업로드 백분율을 나타내는 진행률 표시줄이 나타납니다.

단계 **5** 이미지를 업로드한 후에 특정 소프트웨어 이미지에 대한 최종 사용자 라이선스 계약이 표시됩니다. 시스템 프롬프트에 따라 최종 사용자 라이선스 계약에 동의합니다.

이미지의 무결성 확인

Firepower 4100/9300 섀시에 새 이미지가 추가되면 이미지의 무결성이 자동으로 확인됩니다. 필요한 경우 다음 절차를 사용하여 이미지의 무결성을 수동으로 확인할 수 있습니다.

프로시저

단계 **1** **System(**시스템**)** > **Updates(**업데이트**)**를 선택합니다.
Available Updates(사용 가능한 업데이트) 페이지는 섀시에서 사용 가능한 FXOS 플랫폼 번들 이미지와 애플리케이션 이미지 목록을 보여줍니다.

단계 **2** 확인하려는 이미지에 대해 **Verify(**확인**)**(확인 표시 아이콘)를 클릭합니다.

시스템에서 이미지의 무결성을 확인하고 Image Integrity(이미지 무결성) 필드에 결과를 표시합니다.

FXOS 플랫폼 번들 업그레이드

시작하기 전에

Cisco.com에서 플랫폼 번들 소프트웨어 이미지를 다운로드(Cisco.com에서 이미지 다운로드, 62 페이지 참조)한 다음 해당 이미지를 Firepower 4100/9300 섀시에 업로드합니다(Security Appliance에 이미지 업로드, 62 페이지 참조).

참고 업그레이드 프로세스에는 일반적으로 20~30분이 소요됩니다.

독립형 논리적 디바이스를 실행 중인 Firepower 9300 또는 Firepower 4100 시리즈 보안 어플라이언스를 업그레이드하려는 경우 또는 섀시 내 클러스터를 실행 중인 Firepower 9300 보안 어플라이언스를 업그레이드하려는 경우, 트래픽은 디바이스 업그레이드 중에 해당 디바이스를 통과하지 않습니다.

섀시 간 클러스터에 속한 Firepower 9300 또는 Firepower 4100 시리즈 보안 어플라이언스를 업그레이드하려는 경우, 트래픽은 디바이스 업그레이드 중에 업그레이드되고 있는 디바이스를 통과하지 않습니다. 그러나 클러스터의 다른 디바이스는 트래픽을 계속 전달합니다.

프로시저

단계 1 **System(**시스템**)** > **Updates(**업데이트**)**를 선택합니다.
Available Updates(사용 가능한 업데이트) 페이지는 섀시에서 사용 가능한 FXOS 플랫폼 번들 이미지와 애플리케이션 이미지 목록을 보여줍니다.

단계 2 업그레이드하려는 FXOS 플랫폼 번들에 대해 **Upgrade(**업그레이드**)**를 클릭합니다.

시스템은 설치할 소프트웨어 패키지를 먼저 확인합니다. 시스템은 현재 설치된 애플리케이션과 지정된 FXOS 플랫폼 소프트웨어 패키지 간에 비호환성이 있는지 알려줍니다. 또한 기존 세션이 종료되고 시스템이 업그레이드의 일부로 재부팅되어야 한다고 경고합니다.

단계 3 **Yes(**예**)**를 클릭하여 설치를 계속할지 확인하거나 **No(**아니요**)**를 클릭하여 설치를 취소합니다.

FXOS에서는 번들의 압축을 풀고 구성 요소를 업그레이드하거나 다시 로드합니다.

논리적 디바이스 소프트웨어 이미지 다운로드 - Firepower 4100/9300 섀시

FTP, HTTP/HTTPS, SCP, SFTP 또는 TFTP를 사용하여 논리적 디바이스 소프트웨어 이미지를 Firepower 4100/9300 섀시에 복사할 수 있습니다.

시작하기 전에

구성 파일을 가져오기 위해 필요한 다음 정보를 수집합니다.

- 이미지를 복사하고 있는 원본 서버의 IP 주소 및 인증 크리덴셜
- 소프트웨어 이미지 파일의 정규화된 이름

참고 FXOS 2.8.1 이상 버전은 펌웨어 및 애플리케이션 이미지 다운로드를 위한 HTTP/HTTPS 프로토콜을 지원합니다.

프로시저

단계 1 보안 서비스 모드를 입력합니다.

Firepower-chassis #**scope ssa**

단계 2 애플리케이션 소프트웨어 모드를 입력합니다.

Firepower-chassis /ssa # **scope app-software**

단계 **3** 논리적 디바이스 소프트웨어 이미지를 다운로드합니다.

Firepower-chassis /ssa/app-software # **download image** *URL*

다음 구문 중 하나를 사용하여 가져올 파일의 URL을 지정합니다.

- **ftp**://*username@hostname*/*path*
- **http**://*username@hostname*/*path*
- **https**://*username@hostname*/*path*
- **scp**://*username@hostname*/*path*
- **sftp**://*username@hostname*/*path*
- **tftp**://*hostname*:*port-num*/*path*

단계 **4** 다음 명령을 사용하여 다운로드 프로세스를 모니터링합니다.

Firepower-chassis /ssa/app-software # **show download-task**

단계 **5** 다음 명령을 사용하여 다운로드한 애플리케이션을 확인합니다.

Firepower-chassis /ssa/app-software # **up**

Firepower-chassis /ssa # **show app**

단계 **6** 다음의 명령을 사용하여 특정 애플리케이션에 대한 세부사항을 확인합니다.

Firepower-chassis /ssa # **scope app** *application_type image_version*

Firepower-chassis /ssa/app # **show expand**

예

다음 예에서는 SCP 프로토콜을 사용하여 이미지를 복사합니다.

```
Firepower-chassis # scope ssa
Firepower-chassis /ssa # scope app-software
Firepower-chassis /ssa/app-software # download image
scp://user@192.168.1.1/images/cisco-asa.9.4.1.65.csp
Firepower-chassis /ssa/app-software # show download-task

Downloads for Application Software:
    File Name                     Protocol   Server               Userid          State
    ----------------------------- ---------- -------------------- --------------- -----
    cisco-asa.9.4.1.65.csp        Scp        192.168.1.1          user            Downloaded

Firepower-chassis /ssa/app-software # up

Firepower-chassis /ssa # show app

Application:
    Name       Version    Description Author     Deploy Type CSP Type    Is Default App
    ---------- ---------- ----------- ---------- ----------- ----------- --------------
    asa        9.4.1.41   N/A                    Native      Application No
```

```
    asa        9.4.1.65   N/A                    Native      Application Yes

Firepower-chassis /ssa # scope app asa 9.4.1.65
Firepower-chassis /ssa/app # show expand

Application:
    Name: asa
    Version: 9.4.1.65
    Description: N/A
    Author:
    Deploy Type: Native
    CSP Type: Application
    Is Default App: Yes

    App Attribute Key for the Application:
        App Attribute Key Description
        ----------------- -----------
        cluster-role      This is the role of the blade in the cluster
        mgmt-ip           This is the IP for the management interface
        mgmt-url          This is the management URL for this application

    Net Mgmt Bootstrap Key for the Application:
        Bootstrap Key Key Data Type Is the Key Secret Description
        ------------- ------------- ----------------- -----------
        PASSWORD      String        Yes               The admin user password.

    Port Requirement for the Application:
        Port Type: Data
        Max Ports: 120
        Min Ports: 1

        Port Type: Mgmt
        Max Ports: 1
        Min Ports: 1

        Mgmt Port Sub Type for the Application:
            Management Sub Type
            -------------------
            Default

        Port Type: Cluster
        Max Ports: 1
        Min Ports: 0
Firepower-chassis /ssa/app #
```

논리적 디바이스를 위한 이미지 버전 업데이트

이 절차를 사용하여 ASA 애플리케이션 이미지를 새 버전으로 업그레이드하거나 재해 복구 시나리오에서 사용할 새 시작 버전으로 FTD 애플리케이션 이미지를 설정합니다.

Firepower Chassis Manager 또는 FXOS CLI를 사용하여 FTD 논리적 디바이스에서 시작 버전을 변경하면 애플리케이션이 새 버전으로 즉시 업그레이드되지 않습니다. 논리적 디바이스 시작 버전은 재해 복구 시나리오에서 FTD가 다시 설치하는 버전입니다. FTD 논리적 디바이스를 처음 생성한 후에는 Firepower Chassis Manager 또는 FXOS CLI를 사용하여 FTD 논리적 디바이스를 업그레이드하지 않습니다. FTD 논리적 디바이스를 업그레이드하려면 FMC를 사용해야 합니다. 자세한 내용은 시스

템 릴리스 노트 http://www.cisco.com/c/en/us/support/security/defense-center/products-release-notes-list.html 를 참조하십시오.

또한 FTD 논리적 디바이스에 대한 업데이트는 Firepower Chassis Manager의 **Logical Devices(**논리적 디바이스**)** > **Edit(**수정**)** 및 **System(**시스템**)** > **Updates(**업데이트**)** 페이지에 반영되지 않습니다. 이러한 페이지에 표시되는 버전은 FTD 논리적 디바이스를 만드는 데 사용된 소프트웨어 버전(CSP 이미지)을 나타냅니다.

참고 FTD에 대한 시작 버전을 설정하면 애플리케이션의 시작 버전이 업데이트됩니다. 따라서 선택한 버전을 적용하려면 애플리케이션을 수동으로 다시 설치하거나 블레이드를 다시 초기화해야 합니다. 이 절차는 FTD 소프트웨어를 업그레이드하거나 다운그레이드하는 것이 아니라 전체 재설치(리이미징)합니다. 따라서 애플리케이션이 삭제되고 기존 구성이 손실됩니다.

ASA 논리적 디바이스에서 시작 버전을 변경하면 ASA가 해당 버전으로 업그레이드되며 모든 구성이 복원됩니다. 사용 중인 구성에 따라 ASA 시작 버전을 변경하려면 다음 워크플로를 사용합니다.

참고 ASA의 시작 버전을 설정하면, 애플리케이션이 자동으로 재시작됩니다. 이 절차는 ASA 소프트웨어를 업그레이드하거나 다운그레이드하는 것과 같습니다(기존 구성은 유지됨).

ASA 고가용성 -

1. 스탠바이 유닛에서 논리적 디바이스 이미지 버전을 변경합니다.
2. 스탠바이 유닛을 액티브로 설정합니다.
3. 다른 유닛에서 애플리케이션 버전을 변경합니다.

ASA 섀시 간 클러스터 -

1. 슬레이브 유닛에서 시작 버전을 변경합니다.
2. 데이터 유닛을 제어 유닛으로 설정합니다.
3. 원래 제어 디바이스(현재 데이터)에서 시작 버전을 변경합니다.

시작하기 전에

Cisco.com에서 논리적 디바이스에 사용할 애플리케이션 이미지를 다운로드(Cisco.com에서 이미지 다운로드, 62 페이지 참조)한 다음 해당 이미지를 Firepower 4100/9300 섀시에 업로드합니다(Security Appliance에 이미지 업로드, 62 페이지 참조).

플랫폼 번들 이미지와 하나 이상의 애플리케이션 이미지를 모두 업그레이드하려면 먼저 플랫폼 번들을 업그레이드해야 합니다.

프로시저

단계 **1** **Logical Devices(**논리적 디바이스**)**를 선택하여 Logical Devices(논리적 디바이스) 페이지를 엽니다. Logical Devices(논리적 디바이스) 페이지는 섀시에 구성되어 있는 논리적 디바이스 목록을 보여줍니다. 논리적 디바이스가 구성되어 있지 않은 경우, 대신 이를 알리는 메시지가 표시됩니다.

단계 **2** 업데이트하려는 논리적 디바이스의 **Update Version(**버전 업데이트**)**을 클릭하여 **Update Image Version(**이미지 버전 업데이트**)** 대화 상자를 엽니다.

단계 **3** **New Version(**새 버전**)**으로는 소프트웨어 버전을 선택합니다.

단계 **4** **OK(**확인**)**를 클릭합니다.

펌웨어 업그레이드

Firepower 4100/9300 섀시의 펌웨어 업그레이드에 대한 자세한 내용은 *Cisco Firepower 4100/9300 FXOS 펌웨어 업그레이드 가이드*를 참조하십시오.

버전 **2.0.1** 이하로 수동 다운그레이드

보안 모듈에서 CIMC 이미지를 수동으로 다운그레이드하려면 다음 CLI 단계를 수행합니다.

참고 이 절차는 버전 2.1.1 이상에서 버전 2.0.1 이하로 다운그레이드하는 데 특별히 사용됩니다.

시작하기 전에

다운그레이드할 애플리케이션 이미지가 Firepower 4100/9300 섀시에 다운로드되었는지 확인합니다 (Cisco.com에서 이미지 다운로드, 62 페이지 및 논리적 디바이스 소프트웨어 이미지 다운로드 - Firepower 4100/9300 섀시, 64 페이지 참조).

프로시저

단계 **1** CIMC 이미지를 다운그레이드하기 전에 이미지 버전 비교를 비활성화합니다.

기본 플랫폼 이미지 버전을 지우려면 이 예의 단계를 수행합니다.

예제:

```
firepower# scope org
firepower /org # scope fw-platform-pack default
firepower /org/fw-platform-pack # set platform-bundle-version ""
Warning: Set platform version to empty will result software/firmware incompatibility issue.
firepower /org/fw-platform-pack* # commit-buffer
firepower /org/fw-platform-pack #
```

단계 **2** 모듈 이미지를 다운그레이드합니다.

CIMC 이미지를 변경하려면 이 예의 단계를 수행합니다.

예제:

```
firepower# scope server 1/1
firepower /chassis/server # scope cimc
firepower /chassis/server/cimc # update firmware <version_num>
firepower /chassis/server/cimc* # activate firmware <version_num>
firepower /chassis/server/cimc* # commit-buffer
firepower /chassis/server/cimc #
```

다른 모듈을 업데이트하려면 필요에 따라 이 단계를 반복합니다.

단계 **3** 새 펌웨어 번들을 설치합니다.

이 예의 단계에 따라 다운그레이드 이미지를 설치합니다.

예제:

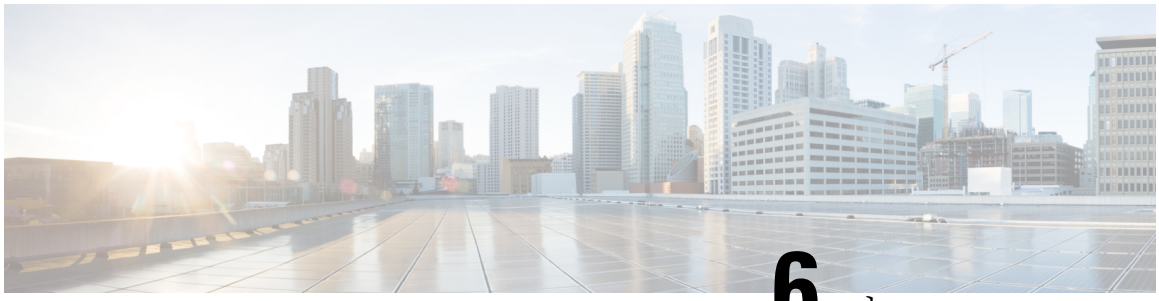
```
firepower# scope firmware
firepower /firmware # scope auto-install
firepower /firmware/auto-install # install platform platform-vers <version_num>
The currently installed FXOS platform software package is <version_num>

WARNING: If you proceed with the upgrade, the system will reboot.

This operation upgrades firmware and software on Security Platform Components
Here is the checklist of things that are recommended before starting Auto-Install
(1) Review current critical/major faults
(2) Initiate a configuration backup
Do you want to proceed? (yes/no):
```

다음에 수행할 작업

펌웨어/자동 설치 모드에서 **show fsm status expand** 명령을 사용하여 설치 프로세스를 모니터링할 수 있습니다.



6 장

보안 인증서 컴플라이언스

보안 인증서 컴플라이언스

미국 연방 정부 기관은 미 국방성 및 글로벌 인증 기관에서 마련한 보안 표준을 준수하는 장비 및 소프트웨어만 사용해야 할 경우가 있습니다. Firepower 4100/9300 섀시는 이러한 보안 인증 표준의 컴플라이언스를 지원합니다.

이러한 표준의 컴플라이언스를 지원하는 기능을 활성하하는 단계는 다음 항목을 참조하십시오.

- FIPS 모드 활성화
- Common Criteria 모드 활성화
- IPSec 보안 채널 구성, 73 페이지
- 트러스트 포인트에 대한 정적 CRL 구성, 79 페이지
- 인증서 해지 목록 확인 정보, 80 페이지
- CRL 주기적 다운로드 구성, 85 페이지
- NTP를 사용하여 날짜 및 시간 설정, 112 페이지
- LDAP 키 링 인증서 설정, 86 페이지
- IP 액세스 목록 구성, 158 페이지
- 클라이언트 인증서 인증 활성화, 87 페이지

- 최소 비밀번호 길이 확인 구성
- 최대 로그인 시도 횟수 설정, 55 페이지

참고 이러한 항목은 Firepower 4100/9300 섀시에서 인증 컴플라이언스를 활성화하는 방법에 대해서만 설명합니다. Firepower 4100/9300 섀시에서 인증 컴플라이언스를 활성화한다고 해서 연결된 논리적 디바이스로 컴플라이언스가 자동으로 전파되지는 않습니다.

SSH 호스트 키 생성

FXOS 릴리스 2.0.1 이전에는, 디바이스의 초기 설정 중 생성된 SSH 호스트 키가 1024비트로 하드 코딩되었습니다. FIPS 및 Common Criteria 인증을 준수하려면 이러한 과거의 호스트 키를 삭제하고 새 호스트 키를 생성해야 합니다. 자세한 내용은 FIPS 모드 활성화 또는 Common Criteria 모드 활성화를 참조하십시오.

과거의 SSH 호스트 키를 삭제하고 인증을 준수하는 새 호스트 키를 생성하려면 다음 단계를 수행하십시오.

프로시저

단계 **1** FXOS CLI에서 서비스 모드로 들어갑니다.

scope system

scope services

단계 **2** SSH 호스트 키를 삭제합니다.

delete ssh-server host-key

단계 **3** 구성을 커밋합니다.

commit-buffer

단계 **4** SSH 호스트 키 크기를 2048비트로 설정합니다.

set ssh-server host-key rsa 2048

단계 **5** 구성을 커밋합니다.

commit-buffer

단계 **6** 새 SSH 호스트 키를 생성합니다.

create ssh-server host-key

commit-buffer

단계 **7** 새 호스트 키 크기를 확인합니다.

show ssh-server host-key

호스트 키 크기: 2048

IPSec 보안 채널 구성

IPSec은 IETF(Internet Engineering Task Force)에서 개발한 개방형 표준 프레임워크입니다. IP 네트워크를 통해 안전하고 인증되고 믿을 수 있는 통신을 생성합니다. IPSec 보안 서비스는 다음을 제공합니다.

- Connectionless Integrity(연결없는 무결성) – 수신된 트래픽이 수정되지 않았다는 보장.
- 데이터 원본 인증 – 합법적인 당사자가 트래픽을 전송한다는 보장.
- Confidentiality (encryption)(기밀성(암호화)) – 인증되지 않은 당사자가 사용자의 트래픽을 검사하지 않음을 보장.
- 액세스 제어 – 리소스의 무단 사용 방지.

IPSec 보안 채널은 다음 알고리즘을 지원합니다.

- 1단계

  ```
  aes128gcm16-prfsha384-prfsha512-prfsha256-prfsha1-ecp256-ecp384-ecp521-modp2048-modp3072-modp4096
  aes128-aes192-aes256-sha256-sha384-sha1_160-sha1-sha512-prfsha384-prfsha512-prfsha256-prfsha1-ecp256-ecp384-ecp521
  aes128-aes192-aes256-sha256-sha384-sha1_160-sha1-sha512-prfsha384-prfsha512-prfsha256-prfsha1-modp2048-modp3072-modp4096
  ```

- 2단계
  - AES SHA 기반 암호화 알고리즘만 지원됩니다. (DES 및 MD5는 지원되지 않음)
  - 지원되는 DH 그룹은 14, 15, 16, 19, 20 및 21입니다.

참고 IPSec 연결은 FXOS에서만 시작할 수 있습니다. FXOS는 수신 IPSec 연결 요청을 수락하지 않습니다.

IPsec 터널은 FXOS가 피어 간에 설정하는 SA 집합입니다. SA는 프로토콜과 알고리즘을 지정하여 민감한 데이터에 지정하고 피어가 사용하는 키 요소도 지정합니다. IPsec SA는 사용자 트래픽의 실제 전송을 제어합니다. SA는 단방향이지만 일반적으로 쌍(인바운드 및 아웃바운드)으로 설정됩니다.

섀시 관리자의 IPSec에는 두 가지 모드가 있습니다.

전송 모드
: IP 헤더, IPSec 헤더, TCP 헤더, 데이터

터널 모드
: 새 IP 헤더, IPSec 헤더, 원래 IP 헤더, TCP 헤더, 데이터

IPSec의 작업은 5 가지 주요 단계로 나눌 수 있습니다.

1. 트래픽 선택 – IPSec 정책과 일치하는 트래픽이 IKE 프로세스를 시작합니다. 예를 들어 src/dst 호스트 IP 또는 서브넷을 사용하여 트래픽을 선택할 수 있습니다. 또는 사용자가 admin 명령을 통해 IKE 프로세스를 트리거할 수 있습니다.
2. IKE 1 단계 – IPSec 피어를 인증하고 IKE 교환을 활성화하는 보안 채널을 설정합니다.
3. IKE 2 단계 – IPSec 터널을 설정하기 위해 SA를 협상합니다. SA는 Security Association(보안 연결)을 나타내며, 데이터 트래픽을 보호하는 데 사용되는 보안 서비스를 설명하는 IPSec 엔드포인트 간의 관계입니다.
4. 데이터 전송 – 데이터 패킷은 SA에 저장된 매개변수 및 키를 사용하여 IPSec 헤더에서 암호화되고 캡슐화 됩니다.
5. IPSec 터널 종료 – IPSec SA는 삭제를 통해 또는 시간 초과로 종료됩니다.

공용 네트워크를 통과하는 데이터 패킷에 대해 엔드 투 엔드 암호화 및 인증 서비스를 제공하기 위해 Firepower 4100/9300 섀시에서 IPSec를 구성할 수 있습니다. 이 옵션은 시스템에서 Common Criteria 인증 컴플라이언스를 얻기 위해 제공되는 숫자 중 하나입니다. 자세한 내용은 보안 인증서 컴플라이언스, 71 페이지를 참고하십시오.

참고

- FIPS 모드에서 IPSec 보안 채널을 사용하는 경우 IPSec 피어가 RFC 7427을 지원해야 합니다.
- IKE 및 SA 연결 간에 암호화 키 강도 매칭의 적용을 구성하도록 선택한 경우(아래의 절차에서 sa-strength-enforcement를 yes로 설정):

SA 적용이 활성화된 경우	IKE 협상 키 크기가 ESP 협상 키 크기보다 작은 경우 연결이 실패합니다. IKE 협상 키 크기가 ESP 협상 키 크기보다 크거나 같은 경우 SA 적용 확인이 통과하고 연결이 성공합니다.
SA 적용이 비활성화된 경우	SA 적용 확인이 통과하고 연결이 성공합니다.

IPSec 보안 채널을 구성하려면 다음 단계를 수행하십시오.

프로시저

단계 **1** FXOS CLI에서 보안 모드로 들어갑니다.

scope security

단계 **2** 키 링을 생성합니다.

enter keyring ssp

! create certreq subject-name *subject-name* **ip** *ip*

단계 **3** 연결된 인증서 요청 정보를 입력합니다.

enter certreq

단계 **4** 국가를 설정합니다.

set country *country*

단계 **5** DNS를 설정합니다.

set dns *dns*

단계 **6** 이메일을 설정합니다.

set e-mail *email*

단계 **7** IP 정보를 설정합니다.

set ip *ip-address*

set ipv6 *ipv6*

단계 **8** 지역 정보를 설정합니다.

set locality *locality*

단계 **9** 조직 이름을 설정합니다.

set org-name *org-name*

단계 **10** 조직 단위 이름을 설정합니다.

set org-unit-name *org-unit-name*

단계 **11** 비밀번호를 설정합니다.

! set password

단계 **12** 상태를 설정합니다.

set state *state*

단계 **13** certreq의 주체 이름을 설정합니다.

set subject-name *subject-name*

단계 **14** 종료합니다.

exit

단계 **15** 모듈러스를 설정합니다.

set modulus *modulus*

단계 **16** 인증서 요청의 재생성을 설정합니다.

setregenerate *{yes | no}*

단계 **17** 트러스트 포인트를 설정합니다.

set trustpoint interca

단계 **18** 종료합니다.

exit

단계 **19** 새로 만든 트러스트 포인트를 입력합니다.

enter trustpoint interca

단계 **20** 인증서 서명 요청을 생성합니다.

set certchain

예제**:**

```
-----BEGIN CERTIFICATE-----
MIIF3TCCA8WgAwIBAgIBADANBgkqhkiG9w0BAQsFADBwMQswCQYDVQQGEwJVUzEL
MAkGA1UECAwCQ0ExDDAKBgNVBAcMA1NKQzEOMAwGA1UECgwFQ2lzY28xDTALBgNV
BAsMBFNUQlUxCzAJBgNVBAMMAkNBMRowGAYJKoZIhvcNAQkBFgtzc3BAc3NwLm5l
dDAeFw0xNjEyMDgxOTMzNTJaFw0yNjEyMDYxOTMzNTJaMHAxCzAJBgNVBAYTAlVT
MQswCQYDVQQIDAJDQTEMMAoGA1UEBwwDU0pDMQ4wDAYDVQQKDAVDaXNjbzENMAsG
A1UECwwEU1RCVTELMAkGA1UEAwwCQ0ExGjAYBgkqhkiG9w0BCQEWC3NzcEBzc3Au
bmV0MIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEA2ukWyMLQuLqTvhq7
zFb3Oz/iyDG/ui6mrLIYn8wE3E39XcXA1/x9IHCmxFKNJdD7EbsggfOuy0Bj+Y4s
+uZ1VapBXV/JrAie7bNn3ZYrI29yuyOrIqoi9k9gL/oRBzH18BwBwGHBOz3hGrSK
Yc2yhsq9y/6yI3nSuLZm6ybmUKjTa+B4YuhDTz4hl/I9x/J5nbGiab3vLDKss1nO
xP9+1+Lc690Vl8/mNPWdjCjDI+U/L9keYs/rbZdRSeXy9kMae42+4FlRHDJjPcSN
Yw1g/gcR2F7QUKRygKckJKXDX2QIiGYSctlSHj18O87o5s/pmQAWWRGkKpfDv3oH
cMPgI2T9rC0D8NNcgPXj9PFKfexoGNGwNTO85fK3kjgMOdWbdeMG3EihxEEOUPD0
Fdu0HrTM5lvwb+vr5wE9HsAiMJ8UuujmHqH5mlwyy3Me+cEDHo0hLeNs+AFrqEXQ
e9S+KZC/dq/9zOLpRsVqSfJsAuVl/QdPDbWShjflE/fP2Wj01PqXywQydzymVvgE
wEZaoFg+mlGJm0+q4RDvnpzEviOYNSAGmOkILh5HQ/eYDcxvd0qbORWb31H32ySl
Ila6UTT9+vnND1f838fxvNvr8nyGD2S/LVaxnZlO4jcSIvtdizbbT8u5B4VcLKIC
x0vkqjo6RvNZJ52sUaD9C3UodTUCAwEAAaOBgTB/MC8GA1UdHwQoMCYwJKAioCCG
Hmh0dHA6Ly8xOTIuMTY4LjQuMjkvcm9vdGNhLmNybDAdBgNVHQ4EFgQU7Jg01A74
jpx8U0APk76pVfYQQ5AwHwYDVR0jBBgwFoAU7Jg01A74jpx8U0APk76pVfYQQ5Aw
DAYDVR0TBAUwAwEB/zANBgkqhkiG9w0BAQsFAAOCAgEAvI8ky2jiXc4wPiMuxIfY
W7DRmszPUWQ7edor7yxuCqzHLVFFOwYRudsyXbv7INR3rJ/X1cRQj9+KidWWVxpo
pFahRhzYxVZ10DHKIzGTQS3jiHgrF3Z8ohWbL15L7PEDlrxMBoJvabPeQRgTmY/n
XZJ7qRYbypO3gUMCaCZ12raJc3/DIpBQ29yweCbUkc9qiHKA0IbnvAxoroHWmBld
94LrJCggfMQTuNJQszJiVVsYJfZ+utlDp2QwfdDv7B0JkwTBjdwRSfotEbc5R18n
BNXYHqxuoNMmqbS3KjCLXcH6xIN8t+UkfP89hvJt/fluJ+s/VJSVZWK4tAWvR7wl
QngCKRJW6FYpzeyNBctiJ07wO+Wt4e3KhIjJDYvA9hFixWcVGDf2r6QW5BYbgGOK
DkHb/gdr/bcdLBKN/PtSJ+prSrpBSaA6rJX8D9UmfhqqN/3f+sS1fM4qWORJc6G2
gAcg7AjEQ/0do512vAI8p8idOg/Wv1O17mavzLpcue05cwMCX9fKxKZZ/+7Pk19Y
ZrXS6uMn/CGnViptn0w+uJ1IRj1oulk+/ZyPtBvFHUkFRnhoWj5SMFyds2IaatyI
47N2ViaZBxhU3GICaH+3O+8rs9Kkz9tBZDSnEJVZA6yxaNCVP1bRUO20G3oRTmSx
8iLBjN+BXggxMmG8ssHisgw=
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MIIFqDCCA5CgAwIBAgIBBDANBgkqhkiG9w0BAQsFADBwMQswCQYDVQQGEwJVUzEL
MAkGA1UECAwCQ0ExDDAKBgNVBAcMA1NKQzEOMAwGA1UECgwFQ2lzY28xDTALBgNV
BAsMBFNUQlUxCzAJBgNVBAMMAkNBMRowGAYJKoZIhvcNAQkBFgtzc3BAc3NwLm5l
dDAeFw0xNjEyMTUyMTM0NTRaFw0yNjEyMTMyMTM0NTRaMHwxCzAJBgNVBAYTAlVT
MQswCQYDVQQIDAJDQTEPMA0GA1UECgwGbmV3c3RnMRAwDgYDVQQLDAduZXdzdGJ1
```

```
MRMwEQYDVQQDDAppbnRlcm0xLWNhMSgwJgYJKoZIhvcNAQkBFhlpbnRlcm0xLWNh
QGludGVybTEtY2EubmV0MIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEA
wLpNnyEx5I4P8uDoWKWF3IZsegjhLANsodxuAUmhmwKekd0OpZZxHMw1wSO4IBX5
4itJS0xyXFzPmeptG3OXvNqCcsT+4BXl3DoGgPMULccc4NesHeg2z8+q3SPA6uZh
iseWNvKfnUjixbQEBtcrWBiSKnZuOz1cpuBn34gtgeFFoCEXN+EZVpPESiancDVh
8pCPlipc/08ZJ3o9GW2j0eHJN84sguIEDL812ROejQvpmfqGUq11stkIIuh+wB+V
VRhUBVG7pV57I6DHeeRp6cDMLXaM3iMTelhdShyo5YUaRJMak/t8kCqhtGXfuLlI
E2AkxKXeeveR9n6cpQd5JiNzCT/t9IQL/T/CCqMICRXLFpLCS9o5S5O2B6QFgcTZ
yKR6hsmwe22wpK8QI7/5oWNXlolb96hHJ7RPbG7RXYqmcLiXY/d2j9/RuNoPJawI
hLkfhoIdPA28xlnfIB1azCmMmdPcBO6cbUQfCj5hSmk3StVQKgJCjaujz55TGGd1
GjnxDMX9twwz7Ee51895Xmtr24qqaCXJoW/dPhcIIXRdJPMsTJ4yPG0BieuRwd0p
i8w/rFwbHzv4C9Fthw1JrRxH1yeHJHrLlZgJ5txSaVUIgrgVCJaf6/jrRRWoRJwt
AzvnzYql2dZPCcEAYgP7JcaQpvdpuDgq++NgBtygiqECAwEAAaNBMD8wDAYDVR0T
BAUwAwEB/zAvBgNVHR8EKDAmMCSgIqAghh5odHRwOi8vMTkyLjE2OC40LjI5L2lu
dGVybS5jcmwwDQYJKoZIhvcNAQELBQADggIBAG/XujJh5G5UWo+cwTSitAezWbJA
h1dAiXZ/OYWZSxkFRliErKdupLqL0ThjnX/wRFfEXbrBQwm5kWAUUDr97D1Uz+2A
8LC5I8SWKXmyf0jUtsnEQbDZb33oVL7yXJk/A0SF0jihpPheMA+YRazaIT9xj9KH
PE7nHCJMbb2ptrHUyvBrKSYrSeEqOpQU2+otnFyV3rS9aelgVjuaWyaWOc3lZ1Oi
CC2tJvY3NnM56j5iesxUCeY/SZ2/ECXN7RRBViLHmA3gFKmWf3xeNiKkxmJCxOaa
UWPC1x2V66I8DG9uUzlWyd79O2dy52aAphAHC6hqlzb6v+gw1Tld7UxaqVd8CD5W
ATjNs+ifkJS1h5ERxHjgcurZXOpR+NWpwF+UDzbMXxx+KAAXCI6ltCd8Pb3wOUC3
PKvwEXaIcCcxGx71eRLpWPZFyEoi4N2NGE9OXRjz0K/KERZgNhsIW3bQMjcw3aX6
OXskEuKgsayctnWyxVqNnqvpuz06kqyubh4+ZgGKZ5LNEXYmGNz3oED1rUN636Tw
SjGAPHgeROzyTFDixCei6aROlGdP/Hwvb0/+uThIe89g8WZ0djTKFUM8uBO3f+II
/cbuyBO1+JrDMq8NkAjxKlJlp1c3WbfCue/qcwtcfUBYZ4i53a56UNF5Ef0rpy/8
B/+07Me/p2y9Luqa
-----END CERTIFICATE-----
ENDOFBUF
```

단계 **21** 인증서 서명 요청을 표시합니다.

show certreq

예제**:**

```
Firepower-chassis# /security/keyring # show certreq
Certificate request subject name: SSP
Certificate request ip address: 192.168.0.111
Certificate request FI A ip address: 0.0.0.0
Certificate request FI B ip address: 0.0.0.0
Certificate request e-mail name:
Certificate request ipv6 address: ::
Certificate request FI A ipv6 address: ::
Certificate request FI B ipv6 address: ::
Certificate request country name: US
State, province or county (full name): CA
Locality name (eg, city): SJC
Organisation name (eg, company): Cisco
Organisational Unit Name (eg, section): Sec
DNS name (subject alternative name):
□□:
-----BEGIN CERTIFICATE REQUEST-----
MIICwTCCAakCAQAwVTELMAkGA1UEBhMCVVMxCzAJBgNVBAgMAkNBMQwwCgYDVQQH
DANTSkMxDjAMBgNVBAoMBUNpc2NvMQ0wCwYDVQQLDARTVEJVMQwwCgYDVQQDDANT
U1AwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDq292Rq3t0laoxPbfE
p/lTKr6rxFhPqSSbtm6sXer//VZFiDTWODockDItuf4Kja215mIS0RyvEYVeRgAs
wbN459wm0BASd8xCjIhsuHDV7yHu539BnvRW6Q2o+gHeSRwckqjClK/tsIxsPkV0
```

```
6OduZYXk2bnsLWs6tNk3uzOIT2Q0FcZ1ET66C8fyyKWTrmvcZjDjkMm2nDFsPlX9
39TYPItDkJE3PocqyaCqmT4uobOuvQeLJh/efkBvwhb4BF8vwzRpHWTdjjU5YnR1
qiR4q7j1RmzVFxCDY3IVP/KDBoa5NyCLEUZECP5QCQFDzIRETZwVOKtxUVG0Nljd
K5TxAgMBAAGgJzAlBgkqhkiG9w0BCQ4xGDAWMBQGA1UdEQQNMAuCA1NTUIcEwKgA
rjANBgkqhkiG9w0BAQsFAAOCAQEARtRBoInxXkBYNlVeEoFCqKttu3+Hc7UdyoRM
2L2pjx5OHbQICC+8NRVRMYujTnp67BWuUZZl03dGP4/lbN6bC9P3CvkZdKUsJkN0
m1Ye9dgz7MO/KEcosarmoMl9WB8LlweVdt6ycSdJzs9shOxwT6TAZPwL7gq/1ShF
RJh6sq5W9p6E0SjYefK62E7MatRjDjS8DXoxj6gfn9DqK15iVpkK2QqT5rneSGj+
R+20TcUnT0h/S5K/bySEM/3U1gFxQCOzbzPuHkj28kXAVczmTxXEkJBFLVduWNo6
DT3u0xImiPR1sqW1jpMwbhC+ZGDtvgKjKHToagup9+8R9IMcBQ==
-----END CERTIFICATE REQUEST-----
```

단계 **22** IPSec 모드로 들어갑니다.

scope ipsec

단계 **23** 로그 자세한 정보 레벨을 설정합니다.

set log-level *log_level*

단계 **24** IPSec 연결을 만들고 입력합니다.

enter connection *connection_name*

단계 **25** IPSec 모드를 tunnel 또는 transport로 설정합니다.

set mode *tunnel_or_transport*

단계 **26** 로컬 IP 주소를 설정합니다.

set local-addr *ip_address*

단계 **27** 원격 IP 주소를 설정합니다.

set remote-addr *ip_address*

단계 **28** 터널 모드를 사용하는 경우 원격 서브넷을 설정합니다.

set remote-subnet *ip/mask*

단계 **29** (선택 사항) 원격 ID를 설정합니다.

set remote-ike-ident *remote_identity_name*

단계 **30** 키 링 이름을 설정합니다.

set keyring-name *name*

단계 **31** (선택 사항) 키 링 비밀번호를 설정합니다.

set keyring-passwd *passphrase*

단계 **32** (선택 사항) IKE-SA 수명을 분 단위로 설정합니다.

set ike-rekey-time *minutes*

minutes 값은 60~1440의 정수일 수 있습니다.

단계 **33** (선택 사항) Child SA 수명을 분 단위로 설정합니다(30-480).

set esp-rekey-time *minutes*

minutes 값은 30~480의 정수일 수 있습니다.

단계 **34** (선택 사항) 초기 연결 중에 수행할 재전송 시퀀스의 수를 설정합니다.

set keyringtries *retry_number*

retry_number 값은 1~5의 정수일 수 있습니다.

단계 **35** (선택 사항) 인증서 해지 목록 확인을 활성화 또는 비활성화합니다.

set revoke-policy *{relaxed | strict}*

단계 **36** 연결을 활성화합니다.

set admin-state enable

단계 **37** 모든 연결을 다시 로드합니다.

reload-conns

시스템이 모든 연결을 중지한 다음 다시 로드합니다. 모든 연결이 재설정을 시도합니다.

단계 **38** (선택 사항) 기존 트러스트 포인트 이름을 IPsec에 추가합니다.

create authority *trustpoint_name*

단계 **39** IKE 및 SA 연결 간 암호화 키 강도 매칭의 적용을 구성합니다.

set sa-strength-enforcement *yes_or_no*

트러스트 포인트에 대한 정적 CRL 구성

해지된 인증서는 CRL(Certification Revocation List)에 유지됩니다. 클라이언트 애플리케이션은 CRL을 사용하여 서버의 인증을 확인합니다. 서버 애플리케이션은 CRL을 사용하여, 더 이상 신뢰할 수 없는 클라이언트 애플리케이션의 액세스 요청을 허용 또는 거부합니다.

CRL(Certification Revocation List) 정보를 사용하여 피어 인증서를 검증하도록 Firepower 4100/9300 섀시를 구성할 수 있습니다. 이 옵션은 시스템에서 Common Criteria 인증 컴플라이언스를 얻기 위해 제공되는 숫자 중 하나입니다. 자세한 내용은 보안 인증서 컴플라이언스, 71 페이지를 참고하십시오.

CRL 정보를 사용하여 피어 인증서를 검증하려면 다음 단계를 수행하십시오.

프로시저

단계 **1** FXOS CLI에서 보안 모드로 들어갑니다.

scope security

단계 **2** 트러스트 포인트 모드로 들어갑니다.

scopetrustpoint *trustname*

단계 **3** 해지 모드로 들어갑니다.

scope revoke

단계 **4** CRL 파일을 다운로드합니다.

import crl *protocol*://*user_id@CA_or_CRL_issuer_IP*/tmp/*DoDCA1CRL1.crl*

참고 DER 형식 정적 CRL은 FXOS에서 지원되지 않습니다. 다음 명령을 사용하여 DER 형식 CRL 파일을 PEM 형식으로 변환해야 합니다.

```
openssl crl -in filename.crl -inform DER -outform PEM -out crl.pem
```

단계 **5** (선택 사항) CRL 정보 가져오기 프로세스의 상태를 표시합니다.

show import-task detail

단계 **6** 인증서 해지 메서드를 CRL-only로 설정합니다.

set certrevokemethod {crl}

인증서 해지 목록 확인 정보

IPSec, HTTPS 및 안전한 LDAP 연결에서 CRL(Certificate Revocation List) 확인 모드를 엄격하게 또는 엄격하지 않게 구성할 수 있습니다.

FXOS는 동적(정적이 아님) CRL 정보를 동적 CRL 정보를 나타내는 X.509 인증서의 CDP 정보에서 수집합니다. 시스템 관리가 FXOS 시스템에서 로컬 CRL 정보를 나타내는 정적 CRL 정보를 수동으로 다운로드합니다. FXOS는 인증서 체인에서 현재 처리 중인 인증서에 대해 동적 CRL 정보를 처리합니다. 정적 CRL은 전체 피어 인증서 체인에 적용됩니다.

안전한 IPSec, LDAP 및 HTTPS 연결을 위한 인증서 해지 확인을 활성화 또는 비활성화하는 단계에 대해서는 IPSec 안전한 채널 구성, LDAP 제공자 생성 및 HTTPS 구성 섹션을 참조하십시오.

참고

- Certificate Revocation Check Mode(인증서 해지 확인 모드)를 Strict(엄격)로 설정하는 경우 피어 인증서 체인의 레벨이 1 이상일 때만 정적 CRL이 적용됩니다. (예를 들어, 피어 인증서 체인이 루트 CA 인증서 및 루트 CA에서 서명한 피어 인증서만 포함한 경우)
- IPSec에 대해 정적 CRL을 구성할 때는 가져온 CRL 파일에 Authority Key Identifier(기관 키 식별자)(authkey) 필드가 있어야 합니다. 이 필드가 없으면 IPSec에서는 해당 파일이 유효하지 않은 것으로 간주합니다.
- 정적 CRL은 동일한 발급자의 동적 CRL보다 먼저 사용됩니다. FXOS가 피어 인증서를 검증할 때, 동일 발급자의 유효한(확인된) 정적 CRL이 있는 경우, FXOS는 피어 인증서의 CDP를 무시합니다.
- 다음 시나리오에서는 엄격한 CRL 확인이 기본적으로 활성화됩니다.
  - 새로 생성된 보안 LDAP 제공자 연결, IPSec 연결 또는 클라이언트 인증서 항목
  - 새로 구축한 FXOS 섀시 관리자(FXOS 2.3.1.x 이상의 초기 시작 버전으로 구축됨)

다음 표에서는 인증서 해지 목록 확인 설정 및 인증서 검증에 따라 연결 결과를 설명합니다.

표 ***6:*** 로컬 정적 ***CRL*** 없이 정적으로 설정된 인증서 해제 확인 모드

로컬 정적 **CRL** 없음	**LDAP** 연결	**IPSec** 연결	클라이언트 인증서 인증
피어 인증서 체인 확인	전체 인증서 체인 필요	전체 인증서 체인 필요	전체 인증서 체인 필요
피어 인증서 체인에서 CDP 확인	전체 인증서 체인 필요	전체 인증서 체인 필요	전체 인증서 체인 필요
피어 인증서 체인의 루트 CA 인증서에 대한 CDP 확인	예	해당 없음	예
피어 인증서 체인에서 인증서 유효성 검사 실패	syslog 메시지와 함께 연결 실패	syslog 메시지와 함께 연결 실패	syslog 메시지와 함께 연결 실패
피어 인증서 체인에서 해지된 인증서	syslog 메시지와 함께 연결 실패	syslog 메시지와 함께 연결 실패	syslog 메시지와 함께 연결 실패
피어 인증서 체인에서 하나의 CDP가 누락	syslog 메시지와 함께 연결 실패	피어 인증서: syslog 메시지와 함께 연결 실패 중간 CA: 연결 장애	syslog 메시지와 함께 연결 실패

로컬 정적 **CRL** 없음	**LDAP** 연결	**IPSec** 연결	클라이언트 인증서 인증
유효한 서명이 있는 피어 인증서 체인에서 하나의 CDP CRL이 비어 있음	연결 성공	연결 성공	syslog 메시지와 함께 연결 실패
피어 인증서 체인의 CDP 중 다운로드할 수 없는 것이 있음	syslog 메시지와 함께 연결 실패	피어 인증서: syslog 메시지와 함께 연결 실패 중간 CA: 연결 장애	syslog 메시지와 함께 연결 실패
인증서에 CDP가 있지만 CDP 서버가 다운됨	syslog 메시지와 함께 연결 실패	피어 인증서: syslog 메시지와 함께 연결 실패 중간 CA: 연결 장애	syslog 메시지와 함께 연결 실패
인증서에 CDP가 있고 서버가 가동 중이고 CRL이 CDP에 있지만, CRL에 유효하지 않은 서명이 있음	syslog 메시지와 함께 연결 실패	피어 인증서: syslog 메시지와 함께 연결 실패 중간 CA: 연결 장애	syslog 메시지와 함께 연결 실패

표 ***7:*** 로컬 정적 ***CRL***과 함께 ***Strict***로 설정된 인증서 해제 확인 모드

로컬 정적 **CRL** 있음	**LDAP** 연결	**IPSec** 연결
피어 인증서 체인 확인	전체 인증서 체인 필요	전체 인증서 체인 필요
피어 인증서 체인에서 CDP 확인	전체 인증서 체인 필요	전체 인증서 체인 필요
피어 인증서 체인의 루트 CA 인증서에 대한 CDP 확인	예	해당 없음
피어 인증서 체인에서 인증서 유효성 검사 실패	syslog 메시지와 함께 연결 실패	syslog 메시지와 함께 연결 실패
피어 인증서 체인에서 해지된 인증서	syslog 메시지와 함께 연결 실패	syslog 메시지와 함께 연결 실패
피어 인증서 체인에서 하나의 CDP가 누락(인증서 체인 레벨 1)	연결 성공	연결 성공
피어 인증서 체인에서 하나의 CDP CRL이 비어 있음(인증서 체인 레벨 1)	연결 성공	연결 성공
피어 인증서 체인의 CDP 중 다운로드할 수 없는 것이 있음(인증서 체인 레벨 1)	연결 성공	연결 성공

로컬 정적 **CRL** 있음	**LDAP** 연결	**IPSec** 연결
인증서에 CDP가 있지만 CDP 서버가 다운됨(인증서 체인 레벨 1)	연결 성공	연결 성공
인증서에 CDP가 있고 서버가 가동 중이고 CRL이 CDP에 있지만, CRL에 유효하지 않은 서명이 있음(인증서 체인 레벨 1)	연결 성공	연결 성공
피어 인증서 체인 레벨이 1보다 높음	syslog 메시지와 함께 연결 실패	CDP와 결합하는 경우 연결이 성공함 CDP가 없으면 연결에서 장애가 발생하며 syslog 메시지가 제공됨

표 ***8:*** 로컬 정적 ***CRL*** 없이 ***Relaxed***로 설정된 인증서 해제 확인 모드

로컬 정적 **CRL** 없음	**LDAP** 연결	**IPSec** 연결	클라이언트 인증서 인증
피어 인증서 체인 확인	전체 인증서 체인	전체 인증서 체인	전체 인증서 체인
피어 인증서 체인에서 CDP 확인	전체 인증서 체인	전체 인증서 체인	전체 인증서 체인
피어 인증서 체인의 루트 CA 인증서에 대한 CDP 확인	예	해당 없음	예
피어 인증서 체인에서 인증서 유효성 검사 실패	syslog 메시지와 함께 연결 실패	syslog 메시지와 함께 연결 실패	syslog 메시지와 함께 연결 실패
피어 인증서 체인에서 해지된 인증서	syslog 메시지와 함께 연결 실패	syslog 메시지와 함께 연결 실패	syslog 메시지와 함께 연결 실패
피어 인증서 체인에서 하나의 CDP가 누락	연결 성공	연결 성공	syslog 메시지와 함께 연결 실패
유효한 서명이 있는 피어 인증서 체인에서 하나의 CDP CRL이 비어 있음	연결 성공	연결 성공	연결 성공
피어 인증서 체인의 CDP 중 다운로드할 수 없는 것이 있음	연결 성공	연결 성공	연결 성공

로컬 정적 **CRL** 없음	**LDAP** 연결	**IPSec** 연결	클라이언트 인증서 인증
인증서에 CDP가 있지만 CDP 서버가 다운됨	연결 성공	연결 성공	연결 성공
인증서에 CDP가 있고 서버가 가동 중이고 CRL이 CDP에 있지만, CRL에 유효하지 않은 서명이 있음	연결 성공	연결 성공	연결 성공

표 ***9:*** 로컬 정적 ***CRL***과 함께 ***Relaxed***로 설정된 인증서 해제 확인 모드

로컬 정적 **CRL** 있음	**LDAP** 연결	**IPSec** 연결
피어 인증서 체인 확인	전체 인증서 체인	전체 인증서 체인
피어 인증서 체인에서 CDP 확인	전체 인증서 체인	전체 인증서 체인
피어 인증서 체인의 루트 CA 인증서에 대한 CDP 확인	예	해당 없음
피어 인증서 체인에서 인증서 유효성 검사 실패	syslog 메시지와 함께 연결 실패	syslog 메시지와 함께 연결 실패
피어 인증서 체인에서 해지된 인증서	syslog 메시지와 함께 연결 실패	syslog 메시지와 함께 연결 실패
피어 인증서 체인에서 하나의 CDP가 누락(인증서 체인 레벨 1)	연결 성공	연결 성공
피어 인증서 체인에서 하나의 CDP CRL이 비어 있음(인증서 체인 레벨 1)	연결 성공	연결 성공
피어 인증서 체인의 CDP 중 다운로드할 수 없는 것이 있음(인증서 체인 레벨 1)	연결 성공	연결 성공
인증서에 CDP가 있지만 CDP 서버가 다운됨(인증서 체인 레벨 1)	연결 성공	연결 성공
인증서에 CDP가 있고 서버가 가동 중이고 CRL이 CDP에 있지만, CRL에 유효하지 않은 서명이 있음(인증서 체인 레벨 1)	연결 성공	연결 성공

로컬 정적 **CRL** 있음	**LDAP** 연결	**IPSec** 연결
피어 인증서 체인 레벨이 1보다 높음	syslog 메시지와 함께 연결 실패	CDP와 결합하는 경우 연결이 성공함 CDP가 없으면 연결에서 장애가 발생하며 syslog 메시지가 제공됨

CRL 주기적 다운로드 구성

CRL을 주기적으로 다운로드하도록 시스템을 구성하여 1~24시간마다 새 CRL을 사용하여 인증서를 검증할 수 있습니다.

이 기능과 함께 다음 프로토콜 및 인터페이스를 사용할 수 있습니다.

- FTP
- SCP
- SFTP
- TFTP
- USB

참고
- SCEP 및 OCSP는 지원되지 않습니다.
- 주기적 다운로드는 CRL당 하나만 구성할 수 있습니다.
- 트러스트 포인트당 하나의 CRL이 지원됩니다.

참고 기간은 1시간 간격으로만 구성할 수 있습니다.

CRL 주기적 다운로드를 구성하려면 다음 단계를 수행하십시오.

시작하기 전에

CRL 정보를 사용하여 피어 인증서를 검증하도록 Firepower 4100/9300 섀시를 이미 구성했는지 확인하십시오. 자세한 내용은 트러스트 포인트에 대한 정적 CRL 구성, 79 페이지를 참고하십시오.

프로시저

단계 **1** FXOS CLI에서 보안 모드로 들어갑니다.

scope security

단계 **2** 트러스트 포인트 모드로 들어갑니다.

scope trustpoint

단계 **3** 해지 모드로 들어갑니다.

scope revoke

단계 **4** 해지 구성을 수정합니다.

sh config

단계 **5** 원하는 구성을 설정합니다.

예제**:**

```
set certrevokemethod crl
set crl-poll-filename rootCA.crl
set crl-poll-path /users/myname
set crl-poll-period 1
set crl-poll-port 0
set crl-poll-protocol scp
! set crl-poll-pwd
set crl-poll-server 182.23.33.113
set crl-poll-user myname
```

단계 **6** 구성 파일을 종료합니다.

exit

단계 **7** (선택 사항) 새 CRL을 다운로드하여 새로운 구성을 테스트합니다.

예제**:**

```
Firepower-chassis /security/trustpoint/revoke # sh import-task

□□□□ □□:
□□ □□ □□□□ □□          Port(□□)     Userid       □/□
--------- -------- --------------- ---------- --------------- -----
rootCA.crl Scp     182.23.33.113   0          myname       Downloading
```

LDAP 키 링 인증서 설정

Firepower 4100/9300 섀시에서 TLS 연결을 지원하기 위해 안전한 LDAP 클라이언트 키 링 인증서를 구성할 수 있습니다. 이 옵션은 시스템에서 Common Criteria 인증 컴플라이언스를 얻기 위해 제공되는 숫자 중 하나입니다. 자세한 내용은 보안 인증서 컴플라이언스, 71 페이지를 참고하십시오.

참고 Common Criteria 모드가 활성화되면 SSL을 활성화하고, 서버 DNS 정보를 사용하여 키 링 인증서를 생성해야 합니다.

LDAP 서버 항목에 대해 SSL이 활성화되면 연결을 설정할 때 키 링 정보를 참조하고 확인해야 합니다.

안전한 LDAP 연결(SSL 활성화)을 위해 LDAP 서버 정보는 CC 모드에서 DNS 정보여야 합니다.

안전한 LDAP 클라이언트 키 링 인증서를 구성하려면 다음 단계를 수행하십시오.

프로시저

단계 **1** FXOS CLI에서 보안 모드로 들어갑니다.

scope security

단계 **2** LDAP 모드로 들어갑니다.

scope ldap

단계 **3** LDAP 서버 모드로 들어갑니다.

enter server *{server_ip|server_dns}*

단계 **4** LDAP 키 링을 설정합니다.

set keyring *keyring_name*

단계 **5** 구성을 커밋합니다.

commit-buffer

클라이언트 인증서 인증 활성화

LDAP와 함께 클라이언트 인증서를 사용하여 사용자의 HTTPS 액세스를 인증하도록 시스템을 설정할 수 있습니다. Firepower 4100/9300 섀시의 기본 인증 구성은 자격 증명 기반입니다.

참고 인증서 인증이 활성화된 경우, 이것이 HTTPS에 대해 허용되는 유일한 인증 형식입니다.

클라이언트 인증서 인증 기능의 FXOS 2.1.1 릴리스에서는 인증서 해지 확인이 지원되지 않습니다.

이 기능을 사용하려면 클라이언트 인증서에서 다음 요구 사항을 충족해야 합니다.

- X509 특성 Subject Alternative Name - Email(주체 대체 이름 - 이메일)에 사용자 이름을 포함해야 합니다.
- Supervisor의 트러스트 포인트로 인증서를 가져온 루트 CA가 클라이언트 인증서에 서명해야 합니다.

프로시저

단계 **1** FXOS CLI에서 서비스 모드로 들어갑니다.

scope system

scope services

단계 **2** (선택 사항) HTTPS 인증에 대한 옵션을 확인합니다.

set https auth-type

예제**:**

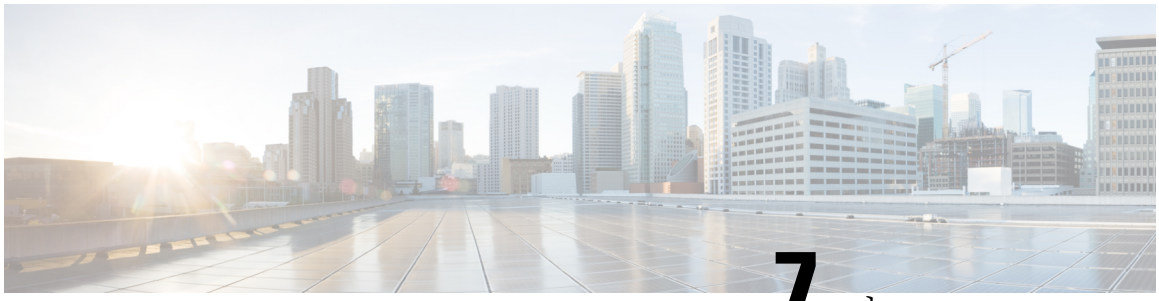
```
Firepower-chassis /system/services # set https auth-type
cert-auth Client certificate based authentication
cred-auth Credential based authentication
```

단계 **3** HTTPS 인증을 클라이언트 기반으로 설정합니다.

set https auth-type cert-auth

단계 **4** 구성을 커밋합니다.

commit-buffer



7 장

시스템 관리

Firepower Chassis Manager 세션을 종료시키는 시스템 변경 사항

다음 시스템 변경 사항으로 인해 Firepower Chassis Manager에서 자동으로 로그아웃될 수 있습니다.

- 시스템 시간을 10분보다 길게 수정하는 경우
- Firepower Chassis Manager 또는 FXOS CLI를 사용하여 시스템을 리부팅하거나 종료하는 경우
- Firepower 4100/9300 섀시에서 FXOS 버전을 업그레이드하는 경우
- FIPS 또는 Common Criteria 모드를 활성화하거나 비활성화하는 경우

참고 위의 변경을 수행하는 경우와 더불어, 아무 작업도 수행하지 않는 상태로 일정 기간이 경과하는 경우에도 시스템에서 자동 로그아웃됩니다. 기본적으로는 10분 동안 작업을 하지 않으면 시스템에서 로그아웃됩니다. 이 시간 초과 설정을 구성하려면 세션 시간 초과 구성, 53 페이지 섹션을 참조하십시오. 세션이 활성 상태이더라도 일정 기간이 지나면 사용자가 시스템에서 로그아웃되는 절대 시간 초과 설정을 구성할 수도 있습니다. 절대 시간 초과 설정을 구성하려면 절대 세션 시간 초과 구성, 54 페이지 섹션을 참조하십시오.

관리 IP 주소 변경

시작하기 전에

Firepower 4100/9300 섀시의 관리 IP 주소를 FXOS CLI에서 변경할 수 있습니다.

참고 관리 IP 주소를 변경한 후, 새 주소를 사용하여 Firepower Chassis Manager 또는 FXOS CLI에 대한 모든 연결을 다시 설정해야 합니다.

프로시저

단계 **1** FXOS CLI에 연결합니다(액세스 - FXOS CLI, 16 페이지 참고).

단계 **2** 다음과 같이 IPv4 관리 IP 주소를 구성합니다.

a) 패브릭 인터커넥트 a의 범위를 설정합니다.

Firepower-chassis# **scope fabric-interconnect a**

b) 다음 명령을 입력하여 현재 관리 IP 주소를 확인합니다.

Firepower-chassis /fabric-interconnect # **show**

c) 다음 명령을 입력하여 새로운 관리 IP 주소 및 게이트웨이를 구성합니다.

Firepower-chassis /fabric-interconnect # **set out-of-band ip** *ip_address* **netmask** *network_mask* **gw** *gateway_ip_address*

d) 시스템 구성에 트랜잭션을 커밋합니다.

Firepower-chassis /fabric-interconnect* # **commit-buffer**

단계 **3** 다음과 같이 IPv6 관리 IP 주소를 구성합니다.

a) 패브릭 인터커넥트 a의 범위를 설정합니다.

Firepower-chassis# **scope fabric-interconnect a**

b) 관리 IPv6 구성의 범위를 설정합니다.

Firepower-chassis /fabric-interconnect # **scope ipv6-config**

c) 다음 명령을 입력하여 현재 관리 IPv6 주소를 확인합니다.

Firepower-chassis /fabric-interconnect/ipv6-config # **show ipv6-if**

d) 다음 명령을 입력하여 새로운 관리 IP 주소 및 게이트웨이를 구성합니다.

Firepower-chassis /fabric-interconnect/ipv6-config # **set out-of-band ipv6** *ipv6_address* **ipv6-prefix** *prefix_length* **ipv6-gw** *gateway_address*

참고 IPv6 글로벌 유니캐스트 주소만 섀시의 IPv6 관리 주소로 지원됩니다.

e) 시스템 구성에 트랜잭션을 커밋합니다.

Firepower-chassis /fabric-interconnect/ipv6-config* # **commit-buffer**

예

다음 예에서는 IPv4 관리 인터페이스 및 게이트웨이를 구성합니다.

```
Firepower-chassis# scope fabric-interconnect a
Firepower-chassis /fabric-interconnect # show

Fabric Interconnect:
    ID   OOB IP Addr     OOB Gateway     OOB Netmask     OOB IPv6 Address OOB IPv6 Gateway
 Prefix Operability
    ---- --------------- --------------- --------------- ---------------- ----------------
 ------ -----------
    A    192.0.2.112     192.0.2.1       255.255.255.0   ::               ::
 64     Operable
Firepower-chassis /fabric-interconnect # set out-of-band ip 192.0.2.111 netmask 255.255.255.0
 gw 192.0.2.1
Warning: When committed, this change may disconnect the current CLI session
Firepower-chassis /fabric-interconnect* #commit-buffer
Firepower-chassis /fabric-interconnect #
```

다음 예에서는 IPv6 관리 인터페이스 및 게이트웨이를 구성합니다.

```
Firepower-chassis# scope fabric-interconnect a
Firepower-chassis /fabric-interconnect # scope ipv6-config
Firepower-chassis /fabric-interconnect/ipv6-config # show ipv6-if

Management IPv6 Interface:
    IPv6 Address                       Prefix     IPv6 Gateway
    ---------------------------------- ---------- ------------
    2001::8998                         64         2001::1
Firepower-chassis /fabric-interconnect/ipv6-config # set out-of-band ipv6 2001::8999
ipv6-prefix 64 ipv6-gw 2001::1
Firepower-chassis /fabric-interconnect/ipv6-config* # commit-buffer
Firepower-chassis /fabric-interconnect/ipv6-config #
```

애플리케이션 관리 IP 변경

Firepower 4100/9300 섀시에 연결된 애플리케이션의 관리 IP 주소를 FXOS CLI에서 변경할 수 있습니다. 그렇게 하려면 먼저 FXOS 플랫폼 레벨에서 IP 정보를 변경한 다음, 애플리케이션 레벨에서 IP 정보를 변경해야 합니다.

참고 애플리케이션 관리 IP를 변경하면 서비스가 중단됩니다.

프로시저

단계 **1** FXOS CLI에 연결합니다. (액세스 - FXOS CLI, 16 페이지를 참조하십시오.)

단계 **2** 논리적 디바이스로 범위를 지정합니다.

scope ssa

scopelogical-device *logical_device_name*

단계 **3** 관리 부트스트랩으로 범위를 지정하고 새로운 관리 부트스트랩 파라미터를 구성합니다. 구축 간에는 다음과 같은 차이점이 있습니다.

ASA 논리적 디바이스의 독립형 구성:

a) 논리적 디바이스 관리 부트스트랩을 입력합니다.
scope mgmt-bootstrap *asa*

b) 슬롯에 대한 IP 모드를 입력합니다.
scope *ipv4_or_6 slot_number* default

c) (IPv4만 해당) 새 IP 주소를 설정합니다.
set ip *ipv4_address* **mask** *network_mask*

d) (IPv6만 해당) 새 IP 주소를 설정합니다.
set ip *ipv6_address* **prefix-length** *prefix_length_number*

e) 게이트웨이 주소를 설정합니다.
set gateway *gateway_ip_address*

f) 구성을 커밋합니다.
commit-buffer

ASA 논리적 디바이스의 클러스터 구성:

a) 클러스터 관리 부트스트랩을 입력합니다.
scope cluster-bootstrap *asa*

b) (IPv4만 해당) 새 가상 IP를 설정합니다.

set virtual ipv4 *ip_address* **mask** *network_mask*

c) (IPv6만 해당) 새 가상 IP를 설정합니다.

set virtual ipv6 *ipv6_address* **prefix-length** *prefix_length_number*

d) 새 IP 풀을 설정합니다.

set ip pool *start_ip end_ip*

e) 게이트웨이 주소를 설정합니다.

set gateway *gateway_ip_address*

f) 구성을 커밋합니다.

commit-buffer

FTD의 독립 실행형 및 클러스터 구성:

a) 논리적 디바이스 관리 부트스트랩을 입력합니다.
scope mgmt-bootstrap *ftd*

b) 슬롯에 대한 IP 모드를 입력합니다.

scope*ipv4_or_6 slot_number* firepower

c) (IPv4만 해당) 새 IP 주소를 설정합니다.

set ip *ipv4_address* **mask** *network_mask*

d) (IPv6만 해당) 새 IP 주소를 설정합니다.

set ip *ipv6_address* **prefix-length** *prefix_length_number*

e) 게이트웨이 주소를 설정합니다.

set gateway *gateway_ip_address*

f) 구성을 커밋합니다.

commit-buffer

참고 클러스터 구성의 경우 Firepower 4100/9300 섀시에 연결된 각 애플리케이션에 대해 새 IP 주소를 설정해야 합니다. 섀시 간 클러스터 또는 HA 구성의 경우, 두 섀시의 각 애플리케이션에 대해 이러한 단계를 반복해야 합니다.

단계 **4** 각 애플리케이션에 대한 관리 부트스트랩 정보를 지웁니다.

a) ssa 모드로 범위를 지정합니다.

scope ssa

b) slot로 범위를 지정합니다.

scope slot *slot_number*

c) 애플리케이션 인스턴스로 범위를 지정합니다.

scopeapp-instance *asa_or_ftd*

d) 관리 부트스트랩 정보를 지웁니다.

clear-mgmt-bootstrap

e) 구성을 커밋합니다.

commit-buffer

단계 **5** 애플리케이션을 비활성화합니다.

disable

commit-buffer

참고 클러스터 구성의 경우 Firepower 4100/9300 섀시에 연결된 각 애플리케이션에 대한 관리 부트스트랩 정보를 지우고 비활성화해야 합니다. 섀시 간 클러스터 또는 HA 구성의 경우, 두 섀시의 각 애플리케이션에 대해 이러한 단계를 반복해야 합니다.

단계 **6** 애플리케이션이 오프라인 상태이고 슬롯이 다시 온라인 상태가 되면 애플리케이션을 다시 활성화합니다.

a) ssa 모드로 다시 범위를 지정합니다.

scope ssa

b) slot로 범위를 지정합니다.

scope slot *slot_number*

c) 애플리케이션 인스턴스로 범위를 지정합니다.

scopeapp-instance *asa_or_ftd*

d) 애플리케이션을 활성화합니다.

enable

e) 구성을 커밋합니다.

commit-buffer

참고 클러스터형 구성의 경우 Firepower 4100/9300 섀시에 연결된 각 애플리케이션을 다시 활성화하려면 다음 단계를 반복해야 합니다. 섀시 간 클러스터 또는 HA 구성의 경우, 두 섀시의 각 애플리케이션에 대해 이러한 단계를 반복해야 합니다.

Firepower 4100/9300 섀시 이름 변경

Firepower 4100/9300 섀시에 사용된 이름을 FXOS CLI에서 변경할 수 있습니다.

프로시저

단계 1 FXOS CLI에 연결합니다(액세스 - FXOS CLI, 16 페이지 참고).

단계 2 시스템 모드로 들어갑니다.

Firepower-chassis-A# **scope system**

단계 3 현재 이름을 확인합니다.

Firepower-chassis-A /system # **show**

단계 4 새 이름을 구성합니다.

Firepower-chassis-A /system # **set name** *device_name*

단계 5 시스템 구성에 트랜잭션을 커밋합니다.

Firepower-chassis-A /fabric-interconnect* # **commit-buffer**

예

다음 예는 디바이스 이름을 변경합니다.

```
Firepower-chassis-A# scope system
Firepower-chassis-A /system # set name New-name
Warning: System name modification changes FC zone name and redeploys them non-disruptively
Firepower-chassis-A /system* # commit-buffer
Firepower-chassis-A /system # show

Systems:
    Name       Mode        System IP Address System IPv6 Address
    ---------- ----------- ----------------- --------------------
    New-name   Stand Alone 192.168.100.10    ::
New-name-A /system #
```

신뢰할 수 있는 ID 인증서 설치

초기 구성 이후 Firepower 4100/9300 섀시 웹 애플리케이션에서 사용하기 위한 자체 서명 SSL 인증서가 생성됩니다. 인증서가 자체 서명된 것이므로 클라이언트 브라우저에서 이를 자동으로 신뢰하지 않습니다. 새 클라이언트 브라우저는 Firepower 4100/9300 섀시 웹 인터페이스에 처음 액세스할 때, Firepower 4100/9300 섀시에 액세스하려면 먼저 인증서를 수락하도록 사용자에게 요구하는 SSL 경고를 표시합니다. FXOS CLI를 사용하여 CSR(Certificate Signing Request)을 생성하고 Firepower 4100/9300 섀시에서 사용할 결과 ID 인증서를 설치하려면 다음 절차를 사용할 수 있습니다. 이 ID 인증서를 사용하면 클라이언트 브라우저가 연결을 신뢰하며 경고 없이 웹 인터페이스를 표시합니다.

프로시저

단계 1 FXOS CLI에 연결합니다. (액세스 - FXOS CLI, 16 페이지를 참조하십시오.)

단계 2 보안 모듈을 입력합니다.

scope security

단계 3 키 링을 생성합니다.

create keyring *keyring_name*

단계 4 개인 키의 모듈러스 크기를 설정합니다.

set modulus *size*

단계 5 구성을 커밋합니다.

commit-buffer

단계 6 CSR 필드를 구성합니다. 기본 옵션(예: subject-name)으로 인증서를 생성할 수도 있고, 인증서에 로케일 및 조직과 같은 정보를 포함하도록 허용하는 좀 더 고급 옵션을 선택적으로 사용할 수도 있습니다. CSR 필드를 구성할 때 인증서 비밀번호를 입력하라는 프롬프트가 표시됩니다.

create certreq subject-name *subject_name*

password

set country *country*

set state *state*

set locality *locality*

set org-name *organization_name*

setorg-unit-name *organization_unit_name*

set subject-name *subject_name*

단계 7 구성을 커밋합니다.

commit-buffer

단계 8 인증 증명에 제공할 CSR을 내보냅니다. 인증 기관은 CSR을 사용하여 ID 인증서를 생성합니다.

a) 전체 CSR을 표시합니다.

show certreq

b) "-----BEGIN CERTIFICATE REQUEST-----"로 시작하고(포함) "-----END CERTIFICATE REQUEST-----"로 끝나는(포함) 출력을 복사합니다.

예제:

```
-----BEGIN CERTIFICATE REQUEST-----
MIIC6zCCAdMCAQAwdzELMAkGA1UEBhMCVVMxEzARBgNVBAgMCkNhbGlmb3JuaWEx
ETAPBgNVBAcMCFNhbiBKb3NlMRYwFAYDVQQKDA1DaXNjbyBTeXN0ZW1zMQwwCgYD
VQQLDANUQUMxGjAYBgNVBAMMEWZwNDEyMC50ZXN0LmxvY2FsMIIBIjANBgkqhkiG
9w0BAQEFAAOCAQ8AMIIBCgKCAQEAs0ON5gagkfZ2fi4JVEANG+7YGgcHbnUt7LpV
yMChnKOPJjBwkUMNQA1mQsRQDcbJ232/sK0fMSnyqOL8JzC7itxeVEZRyz7/ax7W
```

```
GNveg/XP+zd03nt4GXM63FsrPcPmA7EwgqDSLoShtBEV10hhf4+Nw4pKCZ+eSSkS
JkTB1ZHaKV9bttYg3kf/UEUUgk/EyrVq3B+u2DsooPVq76mTm8BwYMqHbJEv4Pmu
RjWE88yEvVwH7JTEij9OvxbatjDjVSJHZBURtCanvyBvGuLP/Q/Nmv3Lo3G9ITbL
L5gIYZVatTxp6HTUezH2MIIzOavU6d1tB9rnyxgGth5dPV0dhQIDAQABoC8wLQYJ
KoZIhvcNAQkOMSAwHjAcBgNVHREEFTATghFmcDQxMjAudGVzdC5sb2NhbDANBgkq
hkiG9w0BAQsFAAOCAQEAZUfCbwx9vt5aVDcL+tATu5xFE3LA310ck6GjlNv6W/6r
jBNLxusYi1rZZcW+CgnvNs4ArqYGyNVBySOavJO/VvQ1KfyxxJ1OIkyx3RzEjgK0
zzyoyrG+EZXC5ShiraS8HuWvE2wFM2wwWNtHWtvcQy55+/hDPD2Bv8pQOC2Zng3I
kLfG1dxWf1xAxLzf5J+AuIQ0CM5HzM9Zm8zREoWT+xHtLSqAqg/aCuomN9/vEwyU
OYfoJMvAqC6AZyUnMfUfCoyuLpLwgkxB0gyaRdnea5RhiGjYQ2lDXYDjEXp7rCx9
+6bvDl1n70JCegHdCWtP75SaNyaBEPkO0365rTckbw==
-----END CERTIFICATE REQUEST-----
```

단계 **9** certreq 모드를 종료합니다.

exit

단계 **10** 키 링 모드를 종료합니다.

exit

단계 **11** 인증 기관의 등록 프로세스에 따라 인증 기관에 CSR 출력을 제공합니다. 요청에 성공하면 인증 기관은 CA의 개인 키를 사용하여 디지털 서명된 ID 인증서를 다시 전송합니다.

단계 **12** 참고 FXOS로 가져오려면 모든 ID 인증서는 Base64 형식이어야 합니다. 인증 기관에서 받은 ID 인증서 체인이 다른 형식인 경우 먼저 OpenSSL과 같은 SSL 툴로 변환해야 합니다.

ID 인증서 체인을 유지할 새 트러스트 포인트를 생성합니다.

create trustpoint *trustpoint_name*

단계 **13** 화면의 지침에 따라 11단계에서, 인증 기관에서 받은 ID 인증서 체인을 입력합니다.

참고 중간 인증서를 사용하는 인증 증명의 경우 루트 인증서와 중간 인증서를 결합해야 합니다. 텍스트 파일에서 맨 위에 루트 인증서를 붙여넣고, 그 뒤에 체인의 각 중간 인증서를 붙여넣습니다(모든 BEGIN CERTIFICATE 및 END CERTIFICATE 플래그 포함). 전체 텍스트 블록을 트러스트 포인트에 복사하여 붙여넣습니다.

set certchain

예제**:**

```
firepower /security/trustpoint* # set certchain
Enter lines one at a time. Enter ENDOFBUF to finish. Press ^C to abort.
Trustpoint Certificate Chain:
>-----BEGIN CERTIFICATE-----
>MIICDTCCAbOgAwIBAgIQYIutxPDPw6BOp3uKNgJHZDAKBggqhkjOPQQDAjBTMRUw
>EwYKCZImiZPyLGQBGRYFbG9jYWwxGDAWBgoJkiaJk/IsZAEZFghuYWF1c3RpbjEg
>MB4GA1UEAxMXbmFhdXN0aW4tTkFBVVNUSU4tUEMtQ0EwHhcNMTUwNzI4MTc1NjU2
>WhcNMjAwNzI4MTgwNjU2WjBTMRUwEwYKCZImiZPyLGQBGRYFbG9jYWwxGDAWBgoJ
>kiaJk/IsZAEZFghuYWF1c3RpbjEgMB4GA1UEAxMXbmFhdXN0aW4tTkFBVVNUSU4t
>UEMtQ0EwWTATBgcqhkjOPQIBBggqhkjOPQMBBwNCAASvEA27V1Enq1gMtLkvJ6rx
>GXRpXWIEyuiBM4eQRoqZKnkeJUkm1xmqlubaDHPJ5TMGfJQYszLBRJPq+mdrKcDl
>o2kwZzATBgkrBgEEAYI3FAIEBh4EAEMAQTAOBgNVHQ8BAf8EBAMCAYYwDwYDVR0T
>AQH/BAUwAwEB/zAdBgNVHQ4EFgQUyInbDHPrFwEEBcbxGSgQW7pOVIkwEAYJKwYB
>BAGCNxUBBAMCAQAwCgYIKoZIzj0EAwIDSAAwRQIhAP++QJTUmniB/AxPDDN63Lqy
>18odMDoFTkG4p3Tb/2yMAiAtMYhlsv1gCxsQVOw0xZVRugSdoOak6n7wCjTFX9jr
>RA==
>-----END CERTIFICATE-----
```

```
>ENDOFBUF
```

단계 **14** 구성을 커밋합니다.

commit-buffer

단계 **15** 트러스트 포인트 모드를 종료합니다.

exit

단계 **16** 키 링 모드로 들어갑니다.

scope keyring *keyring_name*

단계 **17** 13단계에서 생성한 트러스트 포인트를 CSR에 대해 생성한 키 링과 연결합니다.

set trustpoint *trustpoint_name*

단계 **18** 서명한 서버용 ID 인증서를 가져옵니다.

set cert

단계 **19** 인증 증명에서 제공한 ID 인증서의 내용을 붙여넣습니다.

예제:

```
Enter lines one at a time. Enter ENDOFBUF to finish. Press ^C to abort.
Keyring certificate:
>-----BEGIN CERTIFICATE-----
>MIIE8DCCBJagAwIBAgITRQAAAArehlUWgiTzvgAAAAACjAKBggqhkjOPQQDAjBT
>MRUwEwYKCZImiZPyLGQBGRYFbG9jYWwxGDAWBgoJkiaJk/IsZAEZFghuYWF1c3Rp
>bjEgMB4GA1UEAxMXbmFhdXN0aW4tTkFBVVNUSU4tUEMtQ0EwHhcNMTYwNDI4MTMw
>OTU0WhcNMTgwNDI4MTMwOTU0WjB3MQswCQYDVQQGEwJVUzETMBEGA1UECBMKQ2Fs
>aWZvcm5pYTERMA8GA1UEBxMIU2FuIEpvc2UxFjAUBgNVBAoTDUNpc2NvIFN5c3Rl
>bXMxDDAKBgNVBAsTA1RBQzEaMBgGA1UEAxMRZnA0MTIwLnRlc3QubG9jYWwwggEi
>MA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCzQ43mBqCR9nZ+LglUQA0b7tga
>BwdudS3sulXIwKGco48mMHCRQw1ADWZCxFANxsnbfb+wrR8xKfKo4vwnMLuK3F5U
>RlHLPv9rHtYY296D9c/7N3Tee3gZczrcWys9w+YDsTCCoNIuhKG0ERXXSGF/j43D
>ikoJn55JKRImRMHVkdopX1u21iDeR/9QRRSCT8TKtWrcH67YOyig9WrvqZObwHBg
>yodskS/g+a5GNYTzzIS9XAfslMSKP06/Ftq2MONVIkdkFRG0Jqe/IG8a4s/9D82a
>/cujcb0hNssvmAhhlVq1PGnodNR7MfYwgjM5q9Tp3W0H2ufLGAa2Hl09XR2FAgMB
>AAGjggJYMIICVDAcBgNVHREEFTATghFmcDQxMjAudGVzdC5sb2NhbDAdBgNVHQ4E
>FgQU/1WpstiEYExs8DlZWcuHZwPtu5QwHwYDVR0jBBgwFoAUyInbDHPrFwEEBcbx
>GSgQW7pOVIkwgdwGA1UdHwSB1DCB0TCBzqCBy6CByIaBxWxkYXA6Ly8vQ049bmFh
>dXN0aW4tTkFBVVNUSU4tUEMtQ0EsQ049bmFhdXN0aW4tcGMsQ049Q0RQLENOPVB1
>YmxpYyUyMEtleSUyMFNlcnZpY2VzLENOPVNlcnZpY2VzLENOPUNvbmZpZ3VyYXRp
>b24sREM9bmFhdXN0aW4sREM9bG9jYWw/Y2VydGlmaWNhdGVSZXZvY2F0aW9uTGlz
>dD9iYXNlP29iamVjdENsYXNzPWNSTERpc3RyaWJ1dGlvblBvaW50MIHMBggrBgEF
>BQcBAQSBvzCBvDCBuQYIKwYBBQUHMAKGgaxsZGFwOi8vL0NOPW5hYXVzdGluLU5B
>QVVTVElOLVBDLUNBLENOPUFJQSxDTj1QdWJsaWMlMjBLZXklMjBTZXJ2aWNlcyxD
>Tj1TZXJ2aWNlcyxDTj1Db25maWd1cmF0aW9uLERDPW5hYXVzdGluLERDPWxvY2Fs
>P2NBQ2VydGlmaWNhdGU/YmFzZT9vYmplY3RDbGFzcz1jZXJ0aWZpY2F0aW9uQXV0
>aG9yaXR5MCEGCSsGAQQBgjcUAgQUHhIAVwBlAGIAUwBlAHIAdgBlAHIwDgYDVR0P
>AQH/BAQDAgWgMBMGA1UdJQQMMAoGCCsGAQUFBwMBMAoGCCqGSM49BAMCA0gAMEUC
>IFew7NcJirEtFRvyxjkQ4/dVo2oI6CRB308WQbYHNUu/AiEA7UdObiSJBG/PBZjm
>sgoIK60akbjotOTvUdUd9b6K1Uw=
>-----END CERTIFICATE-----
>ENDOFBUF
```

단계 **20** 키 링 모드를 종료합니다.

exit

단계 **21** 보안 모드를 종료합니다.

exit

단계 **22** 시스템 모드로 들어갑니다.

scope system

단계 **23** 서비스 모드로 들어갑니다.

scope services

단계 **24** 새 인증서를 사용하도록 FXOS 웹 서비스를 구성합니다.

sethttps keyring *keyring_name*

단계 **25** 구성을 커밋합니다.

commit-buffer

단계 **26** HTTPS 서버와 연결된 키 링을 표시합니다. 이 절차의 3단계에서 생성한 키 링 이름을 반영해야 합니다. 화면 출력에 기본 키 링 이름이 표시되면 HTTPS 서버가 아직 새 인증서를 사용하도록 업데이트되지 않은 것입니다.

show https

예제**:**

```
fp4120 /system/services # show https
Name: https
    Admin State: Enabled
    Port: 443
    Operational port: 443
    Key Ring: firepower_cert
    Cipher suite mode: Medium Strength
   Cipher suite: ALL:!ADH:!EXPORT40:!EXPORT56:!LOW:!RC4:!MD5:!IDEA:+HIGH:+MEDIUM:+EXP:+eNULL
```

단계 **27** 가져온 인증서의 내용을 표시하고 **Certificate Status** 값이 **Valid**로 표시되는지 확인합니다.

scope security

showkeyring *keyring_name***detail**

예제**:**

```
fp4120 /security #  scope security
fp4120 /security #  show keyring firepower_cert detail
Keyring firepower_cert:
    RSA key modulus: Mod2048
    Trustpoint CA: firepower_chain
    Certificate status: Valid
    Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            45:00:00:00:0a:de:86:55:16:82:24:f3:be:00:00:00:00:00:0a
    Signature Algorithm: ecdsa-with-SHA256
        Issuer: DC=local, DC=naaustin, CN=naaustin-NAAUSTIN-PC-CA
        Validity
```

```
            Not Before: Apr 28 13:09:54 2016 GMT
            Not After : Apr 28 13:09:54 2018 GMT
        Subject: C=US, ST=California, L=San Jose, O=Cisco Systems, OU=TAC,
CN=fp4120.test.local
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:b3:43:8d:e6:06:a0:91:f6:76:7e:2e:09:54:40:
                    0d:1b:ee:d8:1a:07:07:6e:75:2d:ec:ba:55:c8:c0:
                    a1:9c:a3:8f:26:30:70:91:43:0d:40:0d:66:42:c4:
                    50:0d:c6:c9:db:7d:bf:b0:ad:1f:31:29:f2:a8:e2:
                    fc:27:30:bb:8a:dc:5e:54:46:51:cb:3e:ff:6b:1e:
                    d6:18:db:de:83:f5:cf:fb:37:74:de:7b:78:19:73:
                    3a:dc:5b:2b:3d:c3:e6:03:b1:30:82:a0:d2:2e:84:
                    a1:b4:11:15:d7:48:61:7f:8f:8d:c3:8a:4a:09:9f:
                    9e:49:29:12:26:44:c1:d5:91:da:29:5f:5b:b6:d6:
                    20:de:47:ff:50:45:14:82:4f:c4:ca:b5:6a:dc:1f:
                    ae:d8:3b:28:a0:f5:6a:ef:a9:93:9b:c0:70:60:ca:
                    87:6c:91:2f:e0:f9:ae:46:35:84:f3:cc:84:bd:5c:
                    07:ec:94:c4:8a:3f:4e:bf:16:da:b6:30:e3:55:22:
                    47:64:15:11:b4:26:a7:bf:20:6f:1a:e2:cf:fd:0f:
                    cd:9a:fd:cb:a3:71:bd:21:36:cb:2f:98:08:61:95:
                    5a:b5:3c:69:e8:74:d4:7b:31:f6:30:82:33:39:ab:
                    d4:e9:dd:6d:07:da:e7:cb:18:06:b6:1e:5d:3d:5d:
                    1d:85
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Subject Alternative Name:
                DNS:fp4120.test.local
            X509v3 Subject Key Identifier:
                FF:55:A9:B2:D8:84:60:4C:6C:F0:39:59:59:CB:87:67:03:ED:BB:94
            X509v3 Authority Key Identifier:
                keyid:C8:89:DB:0C:73:EB:17:01:04:05:C6:F1:19:28:10:5B:BA:4E:54:89
            X509v3 CRL Distribution Points:
                Full Name:
                  URI:ldap:///CN=naaustin-NAAUSTIN-PC-CA,CN=naaustin-pc,CN=CDP,
                    CN=Public%20Key%20Services,CN=Services,CN=Configuration,DC=naaustin,
                    DC=local?certificateRevocationList?base?objectClass=cRLDistributionPoint

            Authority Information Access:
                CA Issuers - URI:ldap:///CN=naaustin-NAAUSTIN-PC-CA,CN=AIA,
                  CN=Public%20Key%20Services,CN=Services,CN=Configuration,DC=naaustin,
                  DC=local?cACertificate?base?objectClass=certificationAuthority
            1.3.6.1.4.1.311.20.2:
                ...W.e.b.S.e.r.v.e.r
            X509v3 Key Usage: critical
                Digital Signature, Key Encipherment
            X509v3 Extended Key Usage:
                TLS Web Server Authentication
    Signature Algorithm: ecdsa-with-SHA256
         30:45:02:20:57:b0:ec:d7:09:8a:b1:2d:15:1b:f2:c6:39:10:
         e3:f7:55:a3:6a:08:e8:24:41:df:4f:16:41:b6:07:35:4b:bf:
         02:21:00:ed:47:4e:6e:24:89:04:6f:cf:05:98:e6:b2:0a:08:
         2b:ad:1a:91:b8:e8:b4:e4:ef:51:d5:1d:f5:be:8a:d5:4c
-----BEGIN CERTIFICATE-----
MIIE8DCCBJagAwIBAgITRQAAAArehlUWgiTzvgAAAAACjAKBggqhkjOPQQDAjBT
MRUwEwYKCZImiZPyLGQBGRYFbG9jYWwxGDAWBgoJkiaJk/IsZAEZFghuYWF1c3Rp
bjEgMB4GA1UEAxMXbmFhdXN0aW4tTkFBVVNUSU4tUEMtQ0EwHhcNMTYwNDI4MTMw
OTU0WhcNMTgwNDI4MTMwOTU0WjB3MQswCQYDVQQGEwJVUzETMBEGA1UECBMKQ2Fs
aWZvcm5pYTERMA8GA1UEBxMIU2FuIEpvc2UxFjAUBgNVBAoTDUNpc2NvIFN5c3Rl
bXMxDDAKBgNVBAsTA1RBQzEaMBgGA1UEAxMRZnA0MTIwLnRlc3QubG9jYWwwggEi
MA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCzQ43mBqCR9nZ+LglUQA0b7tga
BwdudS3sulXIwKGco48mMHCRQw1ADWZCxFANxsnbfb+wrR8xKfKo4vwnMLuK3F5U
```

```
RlHLPv9rHtYY296D9c/7N3Tee3gZczrcWys9w+YDsTCCoNIuhKG0ERXXSGF/j43D
ikoJn55JKRImRMHVkdopX1u21iDeR/9QRRSCT8TKtWrcH67YOyig9WrvqZObwHBg
yodskS/g+a5GNYTzzIS9XAfslMSKP06/Ftq2MONVIkdkFRG0Jqe/IG8a4s/9D82a
/cujcb0hNssvmAhhlVq1PGnodNR7MfYwgjM5q9Tp3W0H2ufLGAa2Hl09XR2FAgMB
AAGjggJYMIICVDAcBgNVHREEFTATghFmcDQxMjAudGVzdC5sb2NhbDAdBgNVHQ4E
FgQU/1WpstiEYExs8DlZWcuHZwPtu5QwHwYDVR0jBBgwFoAUyInbDHPrFwEEBcbx
GSgQW7pOVIkwgdwGA1UdHwSB1DCB0TCBzqCBy6CByIaBxWxkYXA6Ly8vQ049bmFh
dXN0aW4tTkFBVVNUSU4tUEMtQ0EsQ049bmFhdXN0aW4tcGMsQ049Q0RQLENOPVB1
YmxpYyUyMEtleSUyMFNlcnZpY2VzLENOPVNlcnZpY2VzLENOPUNvbmZpZ3VyYXRp
b24sREM9bmFhdXN0aW4sREM9bG9jYWw/Y2VydGlmaWNhdGVSZXZvY2F0aW9uTGlz
dD9iYXNlP29iamVjdENsYXNzPWNSTERpc3RyaWJ1dGlvblBvaW50MIHMBggrBgEF
BQcBAQSBvzCBvDCBuQYIKwYBBQUHMAKGgaxsZGFwOi8vL0NOPW5hYXVzdGluLU5B
QVVTVElOLVBDLUNBLENOPUFJQSxDTj1QdWJsaWMlMjBLZXklMjBTZXJ2aWNlcyxD
Tj1TZXJ2aWNlcyxDTj1Db25maWd1cmF0aW9uLERDPW5hYXVzdGluLERDPWxvY2Fs
P2NBQ2VydGlmaWNhdGU/YmFzZT9vYmplY3RDbGFzcz1jZXJ0aWZpY2F0aW9uQXV0
aG9yaXR5MCEGCSsGAQQBgjcUAgQUHhIAVwBlAGIAUwBlAHIAdgBlAHIwDgYDVR0P
AQH/BAQDAgWgMBMGA1UdJQQMMAoGCCsGAQUFBwMBMAoGCCqGSM49BAMCA0gAMEUC
IFew7NcJirEtFRvyxjkQ4/dVo2oI6CRB308WQbYHNUu/AiEA7UdObiSJBG/PBZjm
sgoIK60akbjotOTvUdUd9b6K1Uw=
-----END CERTIFICATE-----

    Zeroized: No
```

다음에 수행할 작업

신뢰할 수 있는 새 인증서가 표시되는지 확인하려면, 웹 브라우저의 주소 표시줄에 *https://<FQDN_or_IP>/*를 입력하여 Firepower Chassis Manager로 이동합니다.

참고 브라우저는 또한 주소 표시줄의 입력을 기준으로 인증서의 subject-name을 확인합니다. 인증서가 FQDN(Fully Qualified Domain Name)으로 발급된 경우 브라우저에서 해당 방식으로 액세스해야 합니다. IP 주소를 통해 액세스하는 경우, 신뢰할 수 있는 인증서가 사용되더라도 다른 SSL 오류가 표시됩니다(Common Name Invalid).

인증서 업데이트 자동 가져오기

Cisco 인증서 서버가 다른 루트 CA를 활용하도록 ID 인증서를 변경하면 ASA 디바이스를 실행하는 4100 또는 9300에서 스마트 라이선싱에 대한 연결이 끊어집니다. 라이선싱 연결은 애플리케이션의 Lina 대신 수퍼바이저에 의해 처리되므로 Smart Licensing 기능이 실패합니다. FXOS 기반 디바이스의 경우, FXOS 소프트웨어를 업그레이드하지 않고도 자동 가져오기 기능을 사용하여 문제를 해결할 수 있습니다.

자동 가져오기 기능은 기본적으로 비활성화됩니다. 다음 절차에 따라 FXOS CLI를 사용하여 자동 가져오기 기능을 활성화할 수 있습니다.

시작하기 전에

Cisco 인증서 서버에 연결하도록 DNS 서버를 구성해야 합니다.

프로시저

단계 **1** FXOS CLI에 연결합니다.

단계 **2** 보안 모듈을 입력합니다.

scope security

단계 **3** 자동 가져오기 기능을 활성화합니다.

enter tp-auto-import

예제**:**

```
FXOS# scope security
FXOS /security # enter tp-auto-import
 FXOS /security #
```

단계 **4** 구성을 커밋합니다.

commit-buffer

단계 **5** 자동 가져오기 상태 확인

show detail

예제**:**

자동 가져오기 성공:

```
FXOS /security/tp-auto-import #
FXOS /security/tp-auto-import # show detail
Trustpoints auto import source URL: http://www.cisco.com/security/pki/trs/ios_core.p7b
TrustPoints auto import scheduled time : 22:00
Last Importing Status : Success, Imported with 23 TrustPoint(s)
TrustPoints auto Import function : Enabled
FXOS /security/tp-auto-import #
```

자동 가져오기 실패:

```
FXOS /security/tp-auto-import #
FXOS /security/tp-auto-import # show detail
Trustpoints auto import source URL: http://www.cisco.com/security/pki/trs/ios_core.p7b
TrustPoints auto import scheduled time : 22:00
Last Importing Status : Failure
TrustPoints auto Import function : Enabled
FXOS /security/tp-auto-import #
```

단계 **6** tp-auto-import 기능을 구성합니다. import-time-hour를 설정합니다.

set import-time-hour 시간 **import-time-min** 분

예제**:**

```
FXOS /security/tp-auto-import # set
  import-time-hour  Trustpoints auto import hour time
FXOS /security/tp-auto-import # set import-time-hour
  0-23  Import Time Hour
FXOS /security/tp-auto-import # set import-time-hour 7 import-time-min
  0-59  Import Time Min
FXOS /security/tp-auto-import # set import-time-hour 7 import-time-min 20
<CR>
FXOS /security/tp-auto-import # set import-time-hour 7 import-time-min 20
```

```
FXOS /security/tp-auto-import* # commit-buffer
FXOS /security/tp-auto-import #
```

참고 자동 가져오기 소스 URL은 고정되어 있으며 가져오기 시간 세부사항을 일별 분으로 변경해야 합니다. 매일 예약된 시간에 가져오기가 수행됩니다. 시간과 분이 설정되지 않은 경우 인증서 가져오기는 활성화하는 동안 한 번만 발생합니다. 인증서는 secure-login 옵션을 통해서만 액세스할 수 있는 /opt/certstore 경로 아래의 상자에 번들로 다운로드됩니다. 번들(ios_core.p7b)과 함께 개별 인증서(AutoTP1~AutoTPn)가 자동으로 추출됩니다.

단계 **7** 자동 가져오기 구성이 완료되면 show detail 명령을 입력합니다.

show detail

예제**:**

```
FXOS  /security/tp-auto-import # show detail
Trustpoints auto import source URL: http://www.cisco.com/security/pki/trs/ios_core.p7b
TrustPoints auto import scheduled time : 07:20
Last Importing Status : Success, Imported with 23 TrustPoint(s)
TrustPoints auto Import function : Enabled
```

참고 가져올 수 있는 최대 인증서는 30개입니다. 각 가져오기는 Cisco Certificate Server에 대한 연결 문제가 있는 경우 6번 반복된 다음 show 명령에서 마지막 가져오기 상태를 업데이트합니다.

단계 **8** (선택 사항) 자동 가져오기 기능을 비활성화하려면, delete auto-import 명령을 입력합니다.

delete tp-auto-import

예제**:**

```
FXOS /security #
FXOS /security # delete tp-auto-import
FXOS /security* # commit-buffer
FXOS /security # show detail
security mode:
    Password Strength Check: No
    Minimum Password Length: 8
    Is configuration export key set: No
    Current Task:
FXOS /security # scope tp-auto-import
Error: Managed object does not exist
FXOS /security #
FXOS /security # enter tp-auto-import
FXOS /security/tp-auto-import* # show detail
FXOS /security/tp-auto-import* #
```

참고 자동 가져오기 기능을 비활성화하면, 가져온 인증서는 빌드가 변경되지 않을 때까지 영구적으로 유지됩니다. 자동 가져오기 기능을 비활성화한 다음 빌드를 다운그레이드/업그레이드하면 인증서가 제거됩니다.

Pre-Login 배너

Pre-login 배너가 있으면 사용자가 Firepower Chassis Manager에 로그인할 때 시스템에 배너 텍스트가 표시됩니다. 사용자가 메시지 화면에서 **OK(**확인**)**를 클릭하면 사용자 이름과 비밀번호 프롬프트 창이 표시됩니다. Pre-login 배너가 구성되어 있지 않으면 사용자 이름과 비밀번호 프롬프트 창이 바로 표시됩니다.

사용자가 FXOS CLI에 로그인하면, 비밀번호 프롬프트가 나타나기 전에 배너 텍스트(구성한 경우)가 표시됩니다.

Pre-Login 배너 생성

프로시저

단계 **1** FXOS CLI에 연결합니다(액세스 - FXOS CLI, 16 페이지 참고).

단계 **2** 보안 모드를 입력합니다.

Firepower-chassis# **scope security**

단계 **3** 배너 보안 모드로 들어갑니다.

Firepower-chassis /security # **scope banner**

단계 **4** 다음 명령을 입력하여 pre-login 배너를 만듭니다.

Firepower-chassis /security/banner # **create pre-login-banner**

단계 **5** 사용자가 Firepower Chassis Manager 또는 FXOS CLI에 로그인하기 전에 FXOS에서 사용자에게 표시해야 할 메시지를 지정합니다.

Firepower-chassis /security/banner/pre-login-banner* # **set message**

pre-login 배너 메시지 텍스트를 입력하기 위한 대화 상자가 열립니다.

단계 **6** 프롬프트에서 pre-login 배너 메시지를 입력합니다. 이 필드에는 어떤 표준 ASCII 문자도 사용할 수 있습니다. 여러 줄의 텍스트를 입력할 수 있으며 각 줄의 최대 문자 수는 192자입니다. 줄 사이에 **Enter**를 누릅니다.

입력 다음 줄에 **`ENDOFBUF`**를 입력하고 **Enter**를 눌러 완료합니다.

메시지 설정 대화 상자를 취소하려면 **`Ctrl`** 및 **`C`**를 누릅니다.

단계 **7** 시스템 구성에 트랜잭션을 커밋합니다.

Firepower-chassis /security/banner/pre-login-banner* # **commit-buffer**

예

다음 예에서는 pre-login 배너를 생성합니다.

```
Firepower-chassis# scope security
Firepower-chassis /security # scope banner
Firepower-chassis /security/banner # create pre-login-banner
Firepower-chassis /security/banner/pre-login-banner* # set message
Enter lines one at a time. Enter ENDOFBUF to finish. Press ^C to abort.
Enter prelogin banner:
>Welcome to the Firepower Security Appliance
>**Unauthorized use is prohibited**
>ENDOFBUF
Firepower-chassis /security/banner/pre-login-banner* # commit-buffer
Firepower-chassis /security/banner/pre-login-banner #
```

Pre-Login 배너 수정

프로시저

단계 **1** FXOS CLI에 연결합니다(액세스 - FXOS CLI, 16 페이지 참고).

단계 **2** 보안 모드를 입력합니다.

Firepower-chassis# **scope security**

단계 **3** 배너 보안 모드로 들어갑니다.

Firepower-chassis /security # **scope banner**

단계 **4** pre-login-banner 배너 보안 모드로 들어갑니다.

Firepower-chassis /security/banner # **scope pre-login-banner**

단계 **5** 사용자가 Firepower Chassis Manager 또는 FXOS CLI에 로그인하기 전에 FXOS에서 사용자에게 표시해야 할 메시지를 지정합니다.

Firepower-chassis /security/banner/pre-login-banner # **set message**

pre-login 배너 메시지 텍스트를 입력하기 위한 대화 상자가 열립니다.

단계 **6** 프롬프트에서 pre-login 배너 메시지를 입력합니다. 이 필드에는 어떤 표준 ASCII 문자도 사용할 수 있습니다. 여러 줄의 텍스트를 입력할 수 있으며 각 줄의 최대 문자 수는 192자입니다. 줄 사이에 **Enter**를 누릅니다.

입력 다음 줄에 **ENDOFBUF**를 입력하고 **Enter**를 눌러 완료합니다.

메시지 설정 대화 상자를 취소하려면 **Ctrl** 및 **C**를 누릅니다.

단계 **7** 시스템 구성에 트랜잭션을 커밋합니다.

Firepower-chassis /security/banner/pre-login-banner* # **commit-buffer**

예

다음 예에서는 pre-login 배너를 수정합니다.

```
Firepower-chassis# scope security
Firepower-chassis /security # scope banner
Firepower-chassis /security/banner # scope pre-login-banner
Firepower-chassis /security/banner/pre-login-banner # set message
Enter lines one at a time. Enter ENDOFBUF to finish. Press ^C to abort.
Enter prelogin banner:
>Welcome to the Firepower Security Appliance
>**Unauthorized use is prohibited**
>ENDOFBUF
Firepower-chassis /security/banner/pre-login-banner* # commit-buffer
Firepower-chassis /security/banner/pre-login-banner #
```

Pre-Login 배너 삭제

프로시저

단계 **1** FXOS CLI에 연결합니다(액세스 - FXOS CLI, 16 페이지 참고).

단계 **2** 보안 모드를 입력합니다.

Firepower-chassis# **scope security**

단계 **3** 배너 보안 모드로 들어갑니다.

Firepower-chassis /security # **scope banner**

단계 **4** 시스템에서 pre-login 배너를 삭제합니다.

Firepower-chassis /security/banner # **delete pre-login-banner**

단계 **5** 시스템 구성에 트랜잭션을 커밋합니다.

Firepower-chassis /security/banner* # **commit-buffer**

예

다음 예에서는 pre-login 배너를 삭제합니다.

```
Firepower-chassis# scope security
Firepower-chassis /security # scope banner
Firepower-chassis /security/banner # delete pre-login-banner
Firepower-chassis /security/banner* # commit-buffer
Firepower-chassis /security/banner #
```

Firepower 4100/9300 섀시 리부팅

프로시저

단계 **1** **Overview(**개요**)**를 선택하여 Overview(개요) 페이지를 엽니다.

단계 **2** Overview(개요) 페이지 오른쪽 위에서 Chassis Uptime(섀시 업타임) 옆에 있는 **Reboot(**리부팅**)**를 클릭합니다.

단계 **3** **Yes(**예**)**를 클릭하여 Firepower 4100/9300 섀시의 전원 끄기를 확인합니다.
시스템에 구성된 모든 논리적 디바이스가 정상적으로 셧다운된 후 각 보안 모듈/엔진의 전원이 꺼지고, 마지막으로 Firepower 4100/9300 섀시의 전원이 꺼진 후 재시작됩니다. 이 프로세스는 보통 15~20분 정도 걸립니다.

Firepower 4100/9300 섀시 전원 끄기

프로시저

단계 **1** **Overview(**개요**)**를 선택하여 Overview(개요) 페이지를 엽니다.

단계 **2** Overview(개요) 페이지 오른쪽 위에서 Chassis Uptime(섀시 업타임) 옆에 있는 **Shutdown(**셧다운**)**을 클릭합니다.

단계 **3** **Yes(**예**)**를 클릭하여 Firepower 4100/9300 섀시의 전원 끄기를 확인합니다.
시스템에 구성된 모든 논리적 디바이스가 정상적으로 셧다운된 후 각 보안 모듈/엔진의 전원이 꺼지고, 마지막으로 Firepower 4100/9300 섀시의 전원이 꺼집니다.

공장 기본 구성 복원

FXOS CLI를 사용하여 Firepower 4100/9300 섀시를 공장 기본 구성으로 복원할 수 있습니다.

참고 이 프로세스는 모든 논리적 디바이스 구성을 포함하여 섀시의 모든 사용자 구성을 지웁니다. 이 절차를 완료한 후 시스템을 재구성해야 합니다(초기 구성, 8 페이지 참조).

프로시저

단계 **1** (선택 사항) **erase configuration** 명령은 섀시에서 스마트 라이선스 구성을 제거하지 않습니다. 스마트 라이선스 구성을 제거하려는 경우에도 다음 단계를 수행합니다.

scope license

deregister

Firepower 4100/9300 섀시를 등록 취소하면 계정에서 디바이스가 제거됩니다. 디바이스의 모든 라이선스 엔타이틀먼트 및 인증서가 제거됩니다.

단계 **2** 로컬 관리 셸에 연결합니다.

connect local-mgmt

단계 **3** Firepower 4100/9300 섀시에서 모든 사용자 구성을 지우고 섀시를 원래 공장 기본 구성으로 복원하려면 다음 명령을 입력합니다.

erase configuration

시스템에서 모든 사용자 구성을 지울지 확인하는 메시지를 표시합니다.

단계 **4** 명령 프롬프트에 **yes**를 입력하여 구성을 지운다는 것을 확인합니다.
모든 사용자 구성이 Firepower 4100/9300 섀시에서 지워진 후 시스템이 재부팅됩니다.

시스템 구성 요소를 안전하게 지우기

FXOS CLI를 사용하여 어플라이언스의 구성 요소를 지우고 안전하게 지울 수 있습니다.

erase configuration 명령은 공장 기본 구성 복원, 107 페이지에 설명된 대로 섀시에서 모든 사용자 구성 정보를 제거하고 원래 공장 기본 구성으로 복원합니다.

secure erase 명령은 지정된 어플라이언스 구성 요소를 안전하게 지웁니다. 즉, 데이터만 삭제되는 것이 아니라 물리적 스토리지가 "삭제"됩니다(완전히 지워짐). 이는 하드웨어 스토리지 구성 요소가 잔여 데이터 또는 스텁을 유지하지 않으므로 어플라이언스를 전송하거나 반품할 때 중요합니다.

참고 보안 지우기 중에 디바이스가 재부팅되고, SSH 연결이 종료됩니다. 따라서 직렬 콘솔 포트 연결을 통해 보안 지우기를 수행하는 것이 좋습니다.

프로시저

단계 **1** 로컬 관리 셸에 연결합니다.

connect local-mgmt

단계 **2** 다음 **erase configuration** 명령 중 하나를 입력하여 지정된 어플라이언스 구성 요소를 안전하게 지웁니다.

a) **erase configuration chassis**

모든 데이터와 이미지가 손실되며 복구할 수 없다는 경고 메시지가 표시되고 계속 진행할 것인지 확인하는 메시지가 표시됩니다. **y**를 입력하면, 전체 섀시가 안전하게 지워집니다. 보안 모듈이 먼저 지워지고 수퍼바이저가 지워집니다.

디바이스의 모든 데이터와 소프트웨어가 지워지므로 ROMMON(ROM 모니터)에서만 디바이스 복구를 수행할 수 있습니다.

b) **erase configuration security_module** *module_id*

모듈의 모든 데이터와 이미지가 손실되어 복구할 수 없다는 경고 메시지가 표시되고, 계속 진행할 것인지 확인하는 메시지가 표시됩니다. **y**를 입력하면 모듈이 지워집니다.

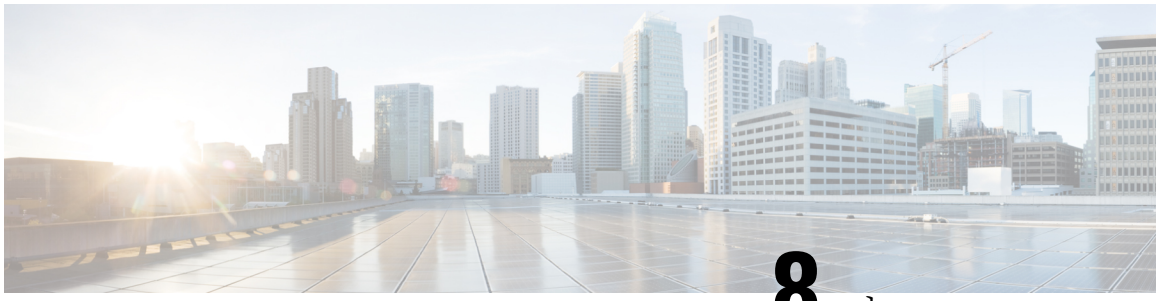
참고 **decommission-secure** 명령은 기본적으로 이 명령과 동일한 결과를 생성합니다.

보안 모듈은 삭제된 후 승인될 때까지 중단된 상태로 유지됩니다(해제되는 모듈과 유사).

c) **erase configuration supervisor**

모든 데이터와 이미지가 손실되며 복구할 수 없다는 경고 메시지가 표시되고, 계속 진행할 것인지 확인하는 메시지가 표시됩니다. **y**를 입력하면 수퍼바이저가 안전하게 지워집니다.

수퍼바이저의 모든 데이터와 소프트웨어가 지워지므로 ROMMON(ROM 모니터)에서만 디바이스 복구를 수행할 수 있습니다.



8 장

플랫폼 설정

날짜 및 시간 설정

시스템에서 NTP(network time protocol)를 구성하거나, 수동으로 날짜 및 시간을 설정하거나, 현재 시스템 시간을 보려면 NTP 페이지을 사용하십시오.

NTP 설정은 Firepower 4100/9300 섀시 및 섀시에 설치된 논리적 디바이스 간에 자동으로 동기화됩니다.

참고 Firepower 4100/9300 섀시에 FTD를 구축할 경우, 스마트 라이센싱의 올바른 작동 및 디바이스 등록 시 올바른 타임스탬프를 보장하려면 Firepower 4100/9300 섀시에서 NTP를 구성해야 합니다. Firepower 4100/9300 섀시 및 FMC에 동일한 NTP 서버를 사용해야 하지만 FMC를 Firepower 4100/9300 섀시의 NTP 서버로 사용할 수는 없습니다.

NTP를 사용하는 경우 **Current Time(**현재 시간**)** 탭에서 전반적인 동기화 상태를 볼 수 있습니다. 또는 **Time Synchronization(**시간 동기화**)** 탭의 **NTP Server(NTP** 서버**)** 테이블에 있는 Server Status(서버 상태) 필드에서 구성된 각 NTP 서버의 동기화 상태를 볼 수 있습니다. 시스템을 특정 NTP 서버와 동기화할 수 없는 경우 Server Status(서버 상태) 옆에 있는 정보 아이콘에 마우스 커서를 대면 자세한 내용을 확인할 수 있습니다.

구성된 날짜 및 시간 보기

프로시저

단계 **1** **Platform Settings(**플랫폼 설정**)** > **NTP**를 선택합니다.

단계 **2** **Current Time(**현재 시간**)** 탭을 클릭합니다.

디바이스에 구성된 날짜, 시간 및 시간대가 표시됩니다.

NTP를 사용 중인 경우 **Current Time(**현재 시간**)** 탭에 전체적인 동기화 상태도 표시됩니다. **Time Synchronization(**시간 동기화**)** 탭에서 **NTP Server(NTP** 서버**)** 테이블의 Server Status(서버 상태) 필드를 확인하여 구성된 각 NTP 서버의 동기화 상태를 볼 수 있습니다. 시스템을 특정 NTP 서버와 동기화할 수 없는 경우 Server Status(서버 상태) 옆에 있는 정보 아이콘에 마우스 커서를 대면 자세한 내용을 확인할 수 있습니다.

표준 시간대 설정

프로시저

단계 **1** **Platform Settings(**플랫폼 설정**)** > **NTP**를 선택합니다.

단계 **2** **Current Time(**현재 시간**)** 탭을 클릭합니다.

단계 **3** **Time Zone(**표준 시간대**)** 드롭다운 목록에서 섀시에 적절한 표준 시간대를 선택합니다.

NTP를 사용하여 날짜 및 시간 설정

NTP는 네트워크 시스템 간에 정확하게 동기화된 시간을 제공하는 계층적 서버 시스템을 구현하는 데 사용합니다. 정밀한 타임 스탬프가 포함된 CRL 검증과 같이 시간에 민감한 작업에는 이러한 정확성이 필요합니다. 최대 4개까지 NTP 서버를 구성할 수 있습니다.

참고

- FXOS는 NTP 버전 3을 사용합니다.
- 외부 NTP 서버의 stratum 값이 13 이상인 경우 애플리케이션 인스턴스는 FXOS 섀시의 NTP 서버와 동기화할 수 없습니다. NTP 클라이언트가 NTP 서버와 동기화될 때마다 stratum 값이 1씩 증가합니다.

  자체 NTP 서버를 설정한 경우, 서버의 /etc/ntp.conf 파일에서 해당 계층 값을 찾을 수 있습니다. NTP 서버의 stratum 값이 13 이상인 경우 ntp.conf 파일에서 stratum 값을 변경하고 서버를 다시 시작하거나 다른 NTP 서버(예: pool.ntp.org)를 사용할 수 있습니다.

시작하기 전에

NTP 서버의 호스트 이름을 사용하는 경우 DNS 서버를 구성해야 합니다. DNS 서버 구성, 156 페이지를 참조하십시오.

프로시저

단계 **1** **Platform Settings(**플랫폼 설정**)** > **NTP**를 선택합니다.

Time Synchronization(시간 동기화**)** 탭이 기본적으로 선택되어 있습니다.

단계 **2** **Set Time Source(**시간 소스 설정**)**에서 **Use NTP Server(NTP** 서버 사용**)**.

단계 **3** (선택 사항) NTP 서버로 인증해야 하는 경우 **NTP Server Authentication: Enable(NTP** 서버 인증**:** 활성화**)** 체크 박스을 선택합니다.

인증 키 ID 및 값을 요구하려면 **Yes(**예**)**를 클릭합니다.

NTP 서버 인증에는 SHA1만 지원됩니다.

단계 **4** IP 주소 또는 호스트 이름별로 최대 4개의 NTP 서버를 식별하려면 **Add(**추가**)**를 클릭합니다.

단계 **5** (선택 사항) NTP 서버의 **Authentication Key(**인증 키**)** ID 및 **Authentication Value(**인증 값**)**를 입력합니다.

NTP 서버에서 키 ID 및 값을 가져옵니다. 예를 들어 OpenSSL이 설치된 NTP 서버 버전 4.2.8p8 이상에서 SHA1 키를 생성하려면 **ntp-keygen -M** 명령을 입력한 다음 ntp.keys 파일에서 키 ID 및 값을 확인합니다. message digest를 계산할 때 어떤 키 값을 사용할지를 클라이언트 및 서버에 알려줄 때 키 ID가 사용됩니다.

단계 **6** **Save(**저장**)**를 클릭합니다.

NTP Server(NTP 서버**)** 테이블의 Server Status(서버 상태) 필드를 확인하여 각 서버의 동기화 상태를 볼 수 있습니다. 시스템을 특정 NTP 서버와 동기화할 수 없는 경우 Server Status(서버 상태) 옆에 있는 정보 아이콘에 마우스 커서를 대면 자세한 내용을 확인할 수 있습니다.

참고 시스템 시간을 10분 이상 수정하는 경우, 시스템에서 로그아웃되며 Firepower Chassis Manager에 다시 로그인해야 합니다.

NTP 서버 삭제

프로시저

단계 **1** **Platform Settings(**플랫폼 설정**)** > **NTP**를 선택합니다.

단계 **2** **Time Synchronization(**시간 동기화**)** 탭을 클릭합니다.

단계 **3** 제거할 각 NTP 서버에 대해 **NTP Server(NTP** 서버**)** 테이블에서 해당 서버의 **Delete(**삭제**)** 아이콘을 클릭합니다.

단계 **4** **Save(**저장**)**를 클릭합니다.

날짜 및 시간 직접 설정

이 섹션에서는 섀시에 날짜 및 시간을 수동으로 설정하는 방법을 설명합니다. 섀시 날짜 및 시간을 수동으로 설정한 후에는 설치된 논리적 디바이스에 변경 사항이 반영되는 데 다소 시간이 걸릴 수 있습니다.

프로시저

단계 **1** **Platform Settings(**플랫폼 설정**)** > **NTP**를 선택합니다.

단계 **2** **Time Synchronization(**시간 동기화**)** 탭을 클릭합니다.

단계 **3** **Set Time Source(**시간 소스 설정**)**에서 **Set Time Manually(**수동으로 시간 설정**)**를 클릭합니다.

단계 **4** **Date(**날짜**)** 드롭다운 목록을 클릭하여 달력을 표시한 다음 달력에서 사용 가능한 컨트롤을 통해 날짜를 설정합니다.

단계 **5** 해당하는 드롭다운 목록을 사용하여 시간을 시, 분 및 AM/PM으로 지정합니다.

팁 **Get System Time(**시스템 시간 가져오기**)**을 클릭하여 Firepower Chassis Manager에 대한 연결에 사용 중인 시스템에 구성되어 있는 날짜 및 시간과 일치하도록 날짜 및 시간을 설정할 수 있습니다.

단계 **6** **Save(**저장**)**를 클릭합니다.

섀시는 날짜 및 시간이 지정된 상태로 구성됩니다.

참고 시스템 시간을 10분 이상 수정하는 경우, 시스템에서 로그아웃되며 Firepower Chassis Manager에 다시 로그인해야 합니다.

SSH 구성

다음 절차에서는 섀시에 대한 SSH 액세스를 활성화하거나 비활성화하는 방법, FXOS 섀시를 SSH 클라이언트로 활성화하는 방법, SSH 서버와 SSH 클라이언트 모두에 대해 암호화, 키 교환 및 메시지 인증을 위해 SSH에서 사용하는 다양한 알고리즘을 구성하는 방법을 설명합니다.

SSH는 기본적으로 활성화되어 있습니다.

프로시저

단계 **1** **Platform Settings(**플랫폼 설정**)** > **SSH** > **SSH Server(SSH** 서버**)**를 선택합니다.

단계 **2** 섀시에 대한 SSH 액세스를 활성화하려면 **Enable SSH(SSH** 활성화**)** 체크 박스를 선택합니다. SSH 액세스를 비활성화하려면 **Enable SSH(SSH** 활성화**)** 확인란의 선택을 취소합니다.

단계 **3** 서버의 **Encryption Algorithm(**암호화 알고리즘**)**에 대해 허용되는 각 암호화 알고리즘의 체크 박스를 선택합니다.

참고
- 다음 암호화 알고리즘은 Common Criteria 모드에서 지원되지 않습니다.
  - 3des-cbc
  - chacha20-poly1305@openssh.com
- chacha20-poly1305@openssh.com은 FIPS에서 지원되지 않습니다. FXOS 섀시에서 FIPS 모드가 활성화되어 있으면, chacha20-poly1305@openssh.com를 암호화 알고리즘으로 사용할 수 없습니다.
- 다음 암호화 알고리즘은 기본적으로 활성화되지 않습니다.

```
aes128-cbc
aes192-cbc
aes265-cbc
```

단계 **4** 서버의 **Key Exchange Algorithm(**키 교환 알고리즘**)**에 대해 허용되는 각 DH(Diffie-Hellman) 키 교환의 체크 박스를 선택합니다. DH 키 교환에서는 어느 한쪽에서 단독으로 확인할 수 없는 공유 암호를 제공합니다. 이 키 교환은 서명 및 호스트 키와 연계하여 호스트 인증을 수행합니다. 이 키 교환 방식은 명시적 서버 인증을 수행합니다. DH 키 교환 방법 사용에 대한 자세한 내용은 RFC 4253을 참조하십시오.

참고
- 다음 키 교환 알고리즘은 Common Criteria 모드에서 지원되지 않습니다.
  - diffie-hellman-group14-sha256
  - curve25519-sha256
  - curve25519-sha256@libssh.org
- 다음 키 교환 알고리즘은 FIPS 모드에서 지원되지 않습니다.
  - curve25519-sha256
  - curve25519-sha256@libssh.org

단계 **5** 서버의 **Mac Algorithm(Mac** 알고리즘**)**에 대해 허용되는 각 무결성 알고리즘의 체크 박스를 선택합니다.

단계 **6** 서버의 **Host Key(**호스트 키**)**에 대해 RSA 키 쌍에 대한 모듈러스 크기를 입력합니다.

모듈러스 값(비트 단위)은 1024~2048 범위의 8의 배수입니다. 지정하는 키 모듈러스 크기가 클수록 RSA 키 쌍을 생성하는 데 오래 걸립니다. 권장되는 값은 2048입니다.

단계 **7** 서버의 **Volume Rekey Limit(**볼륨 재생성 제한**)**에 대해 FXOS가 세션에서 연결을 해제하기 전에 연결을 통해 허용되는 트래픽 양(KB 단위)을 설정합니다.

단계 **8** 서버의 **Time Rekey Limit(**시간 키 재생성 제한**)**에 대해 FXOS가 세션에서 연결을 해제하기 전에 SSH 세션이 유휴 상태가 될 수 있는 시간(분 단위)을 설정합니다.

단계 **9** **Save(**저장**)**를 클릭합니다.

단계 **10** FXOS 섀시 SSH 클라이언트를 맞춤화하려면 **SSH Client(SSH** 클라이언트**)** 탭을 클릭합니다.

단계 **11** **Strict Host Keycheck(**엄격한 호스트 키 확인**)**에 대해 **enable(**활성화**)**, **disable(**비활성화**)** 또는 **prompt(**프롬프트**)**를 선택하여 SSH 호스트 키 확인을 제어합니다.

- **enable(**활성화**)** - 호스트 키가 FXOS의 알려진 호스트 파일에 없는 경우 연결이 거부됩니다. 시스템/서비스 범위에서 **enter ssh-host** 명령을 사용하여 FXOS CLI에서 호스트를 수동으로 추가해야 합니다.
- **prompt(**프롬프트**)** - 호스트 키가 섀시에 저장되어 있지 않은 경우 호스트 키를 수락하거나 거부하라는 프롬프트가 표시됩니다.
- **disable(**비활성화**)** - (기본값) 이전에 저장한 호스트 키가 없는 경우 섀시가 호스트 키를 자동으로 수락합니다.

단계 **12** 클라이언트의 **Encryption Algorithm(**암호화 알고리즘**)**에 대해 허용되는 각 암호화 알고리즘의 체크 박스를 선택합니다.

참고
- 다음 암호화 알고리즘은 Common Criteria 모드에서 지원되지 않습니다.
  - 3des-cbc
  - chacha20-poly1305@openssh.com

  FXOS 섀시에서 Common Criteria 모드가 활성화되어 있으면 3des-cbc를 암호화 알고리즘으로 사용할 수 없습니다.
- chacha20-poly1305@openssh.com은 FIPS에서 지원되지 않습니다. FXOS 섀시에서 FIPS 모드가 활성화되어 있으면, chacha20-poly1305@openssh.com를 암호화 알고리즘으로 사용할 수 없습니다.
- 다음 암호화 알고리즘은 기본적으로 활성화되지 않습니다.

```
aes128-cbc
aes192-cbc
aes265-cbc
```

단계 **13** 클라이언트의 **Key Exchange Algorithm(**키 교환 알고리즘**)**에 대해 허용되는 각 DH(Diffie-Hellman) 키 교환의 체크 박스를 선택합니다. DH 키 교환에서는 어느 한쪽에서 단독으로 확인할 수 없는 공유 암호를 제공합니다. 이 키 교환은 서명 및 호스트 키와 연계하여 호스트 인증을 수행합니다. 이 키 교환 방식은 명시적 서버 인증을 수행합니다. DH 키 교환 방법 사용에 대한 자세한 내용은 RFC 4253을 참조하십시오.

참고
- 다음 키 교환 알고리즘은 Common Criteria 모드에서 지원되지 않습니다.
  - diffie-hellman-group14-sha256
  - curve25519-sha256
  - curve25519-sha256@libssh.org
- 다음 키 교환 알고리즘은 FIPS 모드에서 지원되지 않습니다.
  - curve25519-sha256
  - curve25519-sha256@libssh.org

단계 **14** 클라이언트의 **Mac Algorithm(Mac** 알고리즘**)**에 대해 허용되는 각 무결성 알고리즘의 체크 박스를 선택합니다.

단계 **15** 클라이언트의 **Volume Rekey Limit(**볼륨 재생성 제한**)**에 대해 FXOS가 세션에서 연결을 해제하기 전에 연결을 통해 허용되는 트래픽 양(KB 단위)을 설정합니다.

단계 **16** 클라이언트의 **Time Rekey Limit(**시간 키 재생성 제한**)**에 대해 FXOS가 세션에서 연결을 해제하기 전에 SSH 세션이 유휴 상태가 될 수 있는 시간(분 단위)을 설정합니다.

단계 **17** **Save(**저장**)**를 클릭합니다.

TLS 구성

TLS(Transport Layer Security) 프로토콜은 통신 중인 두 애플리케이션 간에 프라이버시 및 데이터 무결성을 제공합니다. FXOS CLI를 사용하여 FXOS 섀시가 외부 디바이스와 통신할 때 허용되는 최소 TLS 버전을 구성할 수 있습니다. 최신 TLS 버전은 더 안전한 통신을 제공하며, 이전 TLS 버전에서는 오래된 애플리케이션에 대한 이전 버전과의 호환성이 허용됩니다.

예를 들어 FXOS 섀시에 구성된 최소 TLS 버전이 v1.1인데 클라이언트 브라우저가 v1.0만 실행하도록 구성되어 있으면 클라이언트가 HTTPS를 통해 FXOS Chassis Manager와의 연결을 열 수 없습니다. 따라서 피어 애플리케이션 및 LDAP 서버를 적절하게 구성해야 합니다.

이 절차에서는 FXOS 섀시와 외부 디바이스 간의 통신에 허용되는 최소 TLS 버전을 구성하고 확인하는 방법을 설명합니다.

참고

- FXOS 2.3(1) 릴리스를 기준으로, FXOS 섀시용 기본 최소 TLS 버전은 v1.1입니다.

프로시저

단계 **1** 시스템 모드를 입력합니다.

Firepower-chassis# **scope system**

단계 **2** 시스템에서 사용 가능한 TLS 버전 옵션을 확인합니다.

Firepower-chassis /system #**set services tls-ver**

예제**:**

```
Firepower-chassis /system #
Firepower-chassis /system # set services tls-ver
    v1_0  v1.0
    v1_1  v1.1
    v1_2  v1.2
```

단계 **3** 최소 TLS 버전을 설정합니다.

Firepower-chassis /system # **set services tls-ver** *version*

예제**:**

```
Firepower-chassis /system #
Firepower-chassis /system # set services tls-ver v1_2
```

단계 **4** 구성을 커밋합니다.

Firepower-chassis /system #**commit-buffer**

단계 **5** 시스템에 구성된 최소 TLS 버전을 표시합니다.

Firepower-chassis /system #**scope services**

Firepower-chassis /system/services # **show**

예제:

```
Firepower-chassis /system/services # show
Name: ssh
    Admin State: Enabled
    Port: 22
Kex Algo: Diffie Hellman Group1 Sha1,Diffie Hellman Group14 Sha1
Mac Algo: Hmac Sha1,Hmac Sha1 96,Hmac Sha2 512,Hmac Sha2 256
Encrypt Algo: 3des Cbc,Aes256 Cbc,Aes128 Cbc,Aes192 Cbc,Aes256 Ctr,Aes128 Ctr,Ae
s192 Ctr
Auth Algo: Rsa
    Host Key Size: 2048
Volume: None Time: None
Name: telnet
    Admin State: Disabled
    Port: 23
Name: https
    Admin State: Enabled
    Port: 443
    Operational port: 443
    Key Ring: default
    Cipher suite mode: Medium Strength
    Cipher suite: ALL:!ADH:!EXPORT40:!EXPORT56:!LOW:!RC4:!MD5:!IDEA:+HIGH:+MEDIU
M:+EXP:+eNULL
    Https authentication type: Cert Auth
    Crl mode: Relaxed
TLS:
    TLS version: v1.2
```

텔넷 구성

다음 절차에서는 섀시에 대한 Telnet 액세스를 활성화하거나 비활성화하는 방법을 설명합니다. 텔넷은 기본적으로 비활성화되어 있습니다.

참고 텔넷 구성은 현재 CLI를 사용하는 경우에만 사용할 수 있습니다.

프로시저

단계 **1** 시스템 모드를 입력합니다.

Firepower-chassis # **scope system**

단계 **2** 시스템 서비스 모드를 입력합니다.

Firepower-chassis /system # **scope services**

단계 **3** 섀시에 대한 Telnet 액세스를 구성하려면 다음 중 하나를 수행합니다.

- 섀시에 대한 Telnet 액세스를 허용하려면 다음 명령을 입력합니다.

  Firepower-chassis /system/services # **enable telnet-server**

- 섀시에 대한 Telnet 액세스를 거부하려면 다음 명령을 입력합니다.

  Firepower-chassis /system/services # **disable telnet-server**

단계 **4** 시스템 구성에 트랜잭션을 커밋합니다.

Firepower /system/services # **commit-buffer**

예

다음의 예에서는 텔넷을 활성화하고 트랜잭션을 커밋합니다.

```
Firepower-chassis# scope system
Firepower-chassis /system # scope services
Firepower-chassis /services # enable telnet-server
Firepower-chassis /services* # commit-buffer
Firepower-chassis /services #
```

SNMP 구성

SNMP 페이지를 사용하여 섀시에서 단순 네트워크 관리 프로토콜(SNMP)을 구성합니다. 자세한 내용은 다음 항목을 참고하십시오.

SNMP 정보

단순 네트워크 관리 프로토콜(SNMP)은 SNMP 관리자 및 에이전트 간 통신에 메시지 형식을 제공하는 애플리케이션 레이어 프로토콜입니다. SNMP는 네트워크에 있는 디바이스의 모니터링 및 관리에 사용되는 표준화된 프레임워크 및 공통 언어를 제공합니다.

SNMP 프레임워크는 다음 3가지 항목으로 구성됩니다.

- SNMP 관리자 — SNMP를 사용하는 네트워크 디바이스의 활동을 제어하고 모니터링하는 데 쓰이는 시스템.
- SNMP 에이전트 - 섀시에 대한 데이터를 유지 관리하고 필요에 따라 데이터를 SNMP 관리자에게 보고하는 섀시 내의 소프트웨어 구성 요소입니다. 섀시는 MIB 컬렉션 및 에이전트를 포함합니다. SNMP 에이전트를 활성화하고 관리자와 에이전트 간의 관계를 생성하려면 Firepower Chassis Manager 또는 FXOS CLI에서 SNMP를 활성화하고 구성합니다.
- MIB(managed information base) - SNMP 에이전트에 있는 관리되는 개체의 모음.

섀시는 SNMPv1, SNMPv2c 및 SNMPv3를 지원합니다. SNMPv1 및 SNMPv2c는 모두 보안 커뮤니티 기반 양식을 사용합니다. SNMP는 다음에 정의되어 있습니다.

- RFC 3410(http://tools.ietf.org/html/rfc3410)
- RFC 3411(http://tools.ietf.org/html/rfc3411)

- RFC 3412(http://tools.ietf.org/html/rfc3412)
- RFC 3413(http://tools.ietf.org/html/rfc3413)
- RFC 3414(http://tools.ietf.org/html/rfc3414)
- RFC 3415(http://tools.ietf.org/html/rfc3415)
- RFC 3416(http://tools.ietf.org/html/rfc3416)
- RFC 3417(http://tools.ietf.org/html/rfc3417)
- RFC 3418(http://tools.ietf.org/html/rfc3418)
- RFC 3584(http://tools.ietf.org/html/rfc3584)

참고 SNMP 버전 1 및 2c에는 알려진 심각한 보안 문제가 있습니다. 이러한 버전에서 유일한 인증 형식으로 사용되는 커뮤니티 문자열을 포함하여 모든 정보를 암호화 없이 전송합니다.

SNMP 알림

SNMP의 주요 기능은 SNMP 에이전트에서 알림을 생성하는 기능입니다. 이러한 알림에는 SNMP 관리자가 요청을 전송하지 않아도 됩니다. 알림은 잘못된 사용자 인증, 재시작, 연결 종료, 네이버 라우터에 대한 연결 손실 또는 기타 중요한 이벤트를 나타낼 수 있습니다.

섀시는 트랩 또는 알림 중 하나로 SNMP 알림을 생성합니다. 트랩은 SNMP 관리자가 트랩을 수신할 때 승인을 보내지 않고 섀시가 트랩 수신 여부를 확인할 수 없기 때문에 알림보다 신뢰성이 떨어집니다. inform 요청을 수신한 SNMP 관리자는 SNMP 응답 PDU(protocol data unit)로 메시지를 승인합니다. 섀시가 PDU를 수신하지 않으면 알림 요청을 다시 보낼 수 있습니다.

그러나 알림은 안전하지 않은 것으로 간주되어 권장되지 않는 SNMPv2c에서만 사용할 수 있습니다.

참고 SNMP를 사용하는 인터페이스의 ifindex 순서는 FXOS를 재부팅한 후에도 변경되지 않습니다. 그러나 FXOS를 재부팅하면 FXOS 디스크 사용량 OID의 인덱스 번호가 변경됩니다.

SNMP 보안 수준 및 권한

SNMPv1, SNMPv2c 및 SNMPv3는 각각 다른 보안 모델을 나타냅니다. 보안 모델은 선택한 보안 수준과 결합하여 SNMP 메시지를 처리할 때 적용된 보안 메커니즘을 결정합니다.

보안 수준은 SNMP 트랩에 연결된 메시지를 표시하는 데 필요한 권한을 결정합니다. 권한 수준은 메시지가 공개되지 않도록 보호해야 하는지 또는 인증되어야 하는지를 결정합니다. 어떤 보안 모델이 구현되는지에 따라 지원되는 보안 수준이 달라집니다. SNMP 보안 수준은 다음 권한 중 하나 이상을 지원합니다.

- noAuthNoPriv — 인증 또는 암호화 없음
- authNoPriv — 인증은 있지만 암호화 없음
- authPriv — 인증 및 암호화

SNMPv3는 보안 모델 및 보안 수준을 모두 제공합니다. 보안 모델은 사용자 및 사용자 역할을 위해 설정된 인증 전략입니다. 보안 수준은 보안 모델에서 허용된 보안 수준입니다. 보안 모델과 보안 수준을 결합하여 SNMP 패킷을 처리할 때 어떤 보안 메커니즘이 적용되는지 결정합니다.

지원되는 **SNMP** 보안 모델과 수준 결합

다음 표에서는 어떻게 보안 모델과 수준을 결합할 수 있는지에 대해 설명합니다.

표 ***10: SNMP*** 보안 모델과 수준

모델	수준	인증	암호화	결과
v1	noAuthNoPriv	커뮤니티 문자열	없음	인증에 커뮤니티 문자열 일치를 사용합니다.
v2c	noAuthNoPriv	커뮤니티 문자열	없음	인증에 커뮤니티 문자열 일치를 사용합니다.
v3	noAuthNoPriv	사용자 이름	없음	인증에 사용자 이름 일치를 사용합니다. 참고 이를 구성할 수는 있지만, FXOS는 SNMP 버전 3에서 `noAuthNoPriv` 사용을 지원하지 않습니다.
v3	authNoPriv	HMAC-SHA	없음	HMAC SHA(Secure Hash Algorithm) 기반 인증을 제공합니다.
v3	authPriv	HMAC-SHA	DES	HMAC-SHA 알고리즘 기반 인증을 제공합니다. CBC(Cipher Block Chaining) DES(DES-56) 표준 기반의 인증과 함께 DES(Data Encryption Standard) 56비트 암호화도 제공합니다.

SNMPv3 보안 기능

SNMPv3는 네트워크에서 인증 및 암호화 프레임을 결합하여 디바이스에 대한 보안 액세스를 제공합니다. SNMPv3는 구성된 사용자가 수행하는 관리 작업에만 권한을 부여하고 SNMP 메시지를 암호화합니다. SNMPv3 USM(User-Based Security Model)은 SNMP 메시지 수준 보안을 참조하며 다음 서비스를 제공합니다.

- 메시지 통합 — 메시지가 무단으로 변경 또는 손상되지 않았는지, 그리고 데이터 시퀀스가 비악의적인 방식으로 발생할 수 있는 것보다 더 많이 변경되지 않았는지 확인합니다.

- 메시지 출처 인증 — 수신 데이터를 만든 사용자의 클레임된 ID가 확인되도록 보장합니다.
- 메시지 기밀성 및 암호화 — 권한이 없는 개인, 엔티티 또는 프로세스에 정보가 노출 또는 사용되지 않도록 합니다.

SNMP 지원

섀시는 SNMP에 다음을 지원합니다.

MIB 지원

섀시는 MIB에 대해 읽기 전용 액세스를 지원합니다.

사용 가능한 MIB와 이러한 MIB를 획득할 수 있는 위치에 대한 내용은 Cisco FXOS MIB 참조 가이드를 참조하십시오.

SNMPv3 사용자의 인증 프로토콜

섀시는 SNMPv3 사용자에 대해 HMAC-SHA-96(SHA) 인증 프로토콜을 지원합니다.

SNMPv3 사용자를 위한 AES 프라이버시 프로토콜

섀시는 SNMPv3 메시지 암호화를 위한 프라이버시 프로토콜 중 하나로 AES(Advanced Encryption Standard)를 사용하며 RFC 3826을 준수합니다.

프라이버시 비밀번호, 즉 priv 옵션에서는 SNMP 보안 암호화를 위해 DES 또는 128비트 AES 암호화를 선택할 수 있습니다. AES-128 구성을 활성화하고 SNMPv3 사용자에 대한 프라이버시 비밀번호가 있는 경우, Firepower 섀시는 해당 프라이버시 비밀번호를 사용하여 128비트 AES 키를 생성합니다. AES 프라이버시 비밀번호에는 최소 8자 이상을 포함할 수 있습니다. 암호가 일반 텍스트로 지정된 경우, 최대 64자를 지정할 수 있습니다.

SNMP 활성화 및 SNMP 속성 구성

프로시저

단계 **1** **Platform Settings(**플랫폼 설정**)** > **SNMP**를 선택합니다.

단계 **2** **SNMP** 영역에서 다음 필드를 완성합니다.

이름	설명
Admin State(관리 상태**)** 체크 박스	SNMP 활성화 또는 비활성화 여부. 시스템에 SNMP 서버와의 통합이 포함된 경우에만 이 서비스를 활성화합니다.
Port(포트**)** 필드	섀시가 SNMP 호스트와 통신할 때 사용하는 포트. 기본 포트를 변경할 수 없습니다.

이름	설명
Community/Username(커뮤니티/사용자 이름**)** 필드	(선택적) SNMP v1 및 v2에서 폴링에 사용되는 커뮤니티 문자열. SNMP 커뮤니티 이름을 지정하면, SNMP 원격 관리자의 폴링 요청에 대해 SNMP 버전 1 및 2c도 자동으로 활성화됩니다. 이 필드는 SNMP v3에는 적용되지 않습니다. SNMP 버전 1 및 2c에는 알려진 심각한 보안 문제가 있습니다. 이러한 버전에서 유일한 인증 형식으로 사용되는 커뮤니티 문자열을 포함하여 모든 정보를 암호화 없이 전송합니다. 영숫자 문자열은 1자~32자로 입력합니다. @, &, ?를 사용하시 마십시오.(물음표)또는 공백을사용하지 마십시오. 기본값은**public**입니다. **Community/Username(**커뮤니티/사용자 이름**)** 필드가 이미 설정된 경우 빈 필드 오른쪽의 텍스트에 **Set: Yes(**설정**:** 예**)**가 표시됩니다. **Community/Username** 필드에 아직 값이 채워지지 않은 경우 빈 필드 오른쪽의 텍스트에 **Set: No(**설정**:** 아니요**)**가 표시됩니다. 참고 CLI 명령 **set snmp community**을 사용하여 기존 커뮤니티 문자열을 삭제할 수 있으므로, SNMP 원격 관리자의 폴링 요청에 대해 SNMP 버전 1 및 2c를 비활성화할 수 있습니다.
System Administrator Name(시스템 관리자 이름**)** 필드	SNMP 구현을 책임지는 담당자입니다. 이메일 주소, 이름, 전화 번호 등 최대 255자의 문자열로 입력합니다.
Location(위치**)** 필드	SNMP 에이전트(서버)가 실행되는 호스트의 위치입니다. 최대 510자의 영숫자 문자열을 입력합니다.

단계 **3** **Save(**저장**)**를 클릭합니다.

다음에 수행할 작업

SNMP 트랩 및 사용자를 생성합니다.

SNMP 트랩 생성

다음 절차에서는 SNMP 트랩을 생성하는 방법을 설명합니다.

참고 최대 8개의 SNMP 트랩을 정의할 수 있습니다.

프로시저

단계 **1** **Platform Settings(**플랫폼 설정**)** > **SNMP**를 선택합니다.

단계 **2** **SNMP Traps(SNMP** 트랩**)** 영역에서 **Add(**추가**)**를 클릭합니다.

단계 **3** **Add SNMP Trap(SNMP** 트랩 추가**)** 대화 상자에서 다음 필드를 작성합니다.

이름	설명
Host Name(호스트 이름**)** 필드	섀시가 트랩을 전송해야 하는 SNMP 호스트의 호스트 이름 또는 IP 주소.
Community/Username(커뮤니티/사용자 이름**)** 필드	트랩 대상에 대한 액세스를 허용하는 데 필요한 SNMPv1/v2c 커뮤니티 문자열 또는 SNMPv3 사용자 이름을 입력합니다. 이것은 SNMP 서비스를 위해 구성된 커뮤니티 또는 사용자 이름과 동일해야 합니다. 영숫자 문자열은 1자~32자로 입력합니다. @(at 기호), \(백슬래시), "(큰 따옴표), ? (물음표) 또는 공백을 사용하지 마십시오.
Port(포트**)** 필드	섀시가 트랩을 위해 SNMP 호스트와 통신하는 포트입니다. 1 ~ 65535 범위의 정수를 입력합니다.
Version(버전**)** 필드	트랩에 사용되는 SNMP 버전 및 모델입니다. 다음 중 하나일 수 있습니다. • **V1** • **V2** • **V3** 참고 SNMP 버전 1 및 2c에는 알려진 심각한 보안 문제가 있습니다. 이러한 버전에서 유일한 인증 형식으로 사용되는 커뮤니티 문자열을 포함하여 모든 정보를 암호화 없이 전송합니다.
Type(유형**)** 필드	전송할 트랩 유형을 지정합니다. • 트랩 • 알림(버전이 **V2**인 경우에만 유효)
v3 Privilege(v3 권한**)** 필드	버전을 **V3**로 선택한 경우 트랩과 연결된 권한을 지정합니다. • **Auth** — 인증하지만 암호화 없음 • **Noauth** — 인증 또는 암호화 없음 선택할 수는 있지만 FXOS는 SNMPv3에서 이 보안 레벨을 지원하지 않습니다. • **Priv** — 인증 및 암호화

단계 **4** **OK(**확인**)**를 클릭하여 **Add SNMP Trap(SNMP** 트랩 추가**)** 대화 상자를 닫습니다.

단계 **5** **Save(**저장**)**를 클릭합니다.

SNMP 트랩 삭제

프로시저

단계 **1** **Platform Settings(**플랫폼 설정**)** > **SNMP**를 선택합니다.

단계 **2** **SNMP Traps(SNMP** 트랩**)** 영역에서 삭제할 트랩에 해당하는 테이블의 행에 있는 **Delete(**삭제**)** 아이콘을 클릭합니다.

SNMPv3 사용자 생성

프로시저

단계 **1** **Platform Settings(**플랫폼 설정**)** > **SNMP**를 선택합니다.

단계 **2** **SNMP Users(SNMP** 사용자**)** 영역에서 **Add(**추가**)**를 클릭합니다.

단계 **3** **Add SNMP User(SNMP** 사용자 추가**)** 대화 상자에서 다음 필드를 작성합니다.

이름	설명
Name(이름**)** 필드	SNMPv3 사용자에게 할당된 사용자 이름입니다. 최대 32자까지 입력할 수 있습니다. 이름은 문자로 시작해야 합니다. 올바른 문자에는 글자, 숫자, _(밑줄)이 포함됩니다. (마침표), @(at 기호) 및 -(하이픈)을 지정할 수 있습니다.
Auth Type(인증 유형**)** 필드	권한 부여 유형: **SHA**.
Use AES-128(AES-128 사용**)** 체크 박스	이 확인란을 선택한 경우, 해당 사용자는 AES-128 암호화를 사용합니다. 참고 SNMPv3는 DES를 지원하지 않습니다. AES-128 상자를 선택하지 않은 상태로 두면 프라이버시 암호화가 수행되지 않으며, 설정된 프라이버시 비밀번호가 적용되지 않습니다.

이름	설명
Password(비밀번호**)** 필드	이 사용자의 비밀번호입니다. FXOS에서는 다음 요건을 충족하지 않는 모든 비밀번호를 거부합니다. • 8자 이상, 80자 이하여야 합니다. • 문자, 숫자 및 다음 문자만 포함해야 합니다. ~`!@#%^&*()_-+{}[]\:;"'<,>./ • 다음 기호는 포함할 수 없습니다. 예: $(달러 기호), ? (물음표) 또는 = (등호). • 각기 다른 문자를 5자 이상 포함해야 합니다. • 연속적으로 증가하거나 감소하는 문자나 숫자를 너무 많이 포함하면 안 됩니다. 예를 들어 "12345" 문자열에는 이러한 문자가 4개 포함되고 "ZYXW" 문자열에는 3개 포함됩니다. 증가/감소 문자의 총수가 특정 한도를 초과하는 경우(대개 해당 문자가 4~6개 이상 포함되는 경우) 단순성 검사에 실패하게 됩니다. 참고 연속적으로 증가하거나 감소하는 문자 사이에 증가하거나 감소하지 않는 문자가 사용되는 경우에는 증가/감소 문자 수가 재설정되지 않습니다. 예를 들어 abcd&!21의 경우 비밀번호 검사에 실패하지만 abcd&!25의 경우에는 비밀번호 검사에 통과합니다.
Confirm Password(비밀번호 확인**)** 필드	확인을 위해 다시 한 번 입력하는 비밀번호입니다.

이름	설명
Privacy Password(프라이버시 비밀번호**)** 필드	이 사용자의 프라이버시 비밀번호입니다. FXOS에서는 다음 요건을 충족하지 않는 모든 비밀번호를 거부합니다. • 8자 이상, 80자 이하여야 합니다. • 문자, 숫자 및 다음 문자만 포함해야 합니다. ~`!@#%^&*()_-+{}[]\\:;"'<,>./ • 다음 기호는 포함할 수 없습니다. 예: $(달러 기호), ? (물음표) 또는 = (등호). • 각기 다른 문자를 5자 이상 포함해야 합니다. • 연속적으로 증가하거나 감소하는 문자나 숫자를 너무 많이 포함하면 안 됩니다. 예를 들어 "12345" 문자열에는 이러한 문자가 4개 포함되고 "ZYXW" 문자열에는 3개 포함됩니다. 증가/감소 문자의 총수가 특정 한도를 초과하는 경우(대개 해당 문자가 4~6개 이상 포함되는 경우) 단순성 검사에 실패하게 됩니다. 참고 연속적으로 증가하거나 감소하는 문자 사이에 증가하거나 감소하지 않는 문자가 사용되는 경우에는 증가/감소 문자 수가 재설정되지 않습니다. 예를 들어 abcd&!21의 경우 비밀번호 검사에 실패하지만 abcd&!25의 경우에는 비밀번호 검사에 통과합니다.
Confirm Privacy Password(프라이버시 비밀번호 확인**)** 필드	확인을 위해 다시 한 번 입력하는 프라이버시 비밀번호입니다.

단계 **4** **OK(**확인**)**를 클릭하여 **Add SNMP User(SNMP** 사용자 추가**)** 대화 상자를 닫습니다.

단계 **5** **Save(**저장**)**를 클릭합니다.

SNMPv3 사용자 삭제

프로시저

단계 **1** **Platform Settings(**플랫폼 설정**)** > **SNMP**를 선택합니다.

단계 **2** **SNMP Users(SNMP** 사용자**)** 영역에서 삭제할 사용자에 해당하는 테이블의 행에 있는 **Delete(**삭제**)** 아이콘을 클릭합니다.

HTTPS 구성

이 섹션에서는 Firepower 4100/9300 섀시에서 HTTPS를 구성하는 방법을 설명합니다.

참고 Firepower Chassis Manager 또는 FXOS CLI를 사용하여 HTTPS 포트를 변경할 수 있습니다. 다른 모든 HTTPS 구성 작업에는 FXOS CLI만 사용해야 합니다.

인증서, 키 링, 트러스트 포인트

HTTPS에서는 PKI(Public Key Infrastructure)의 구성 요소를 사용하여 두 디바이스, 이를테면 클라이언트 브라우저와 Firepower 4100/9300 섀시 간의 보안 통신을 설정합니다.

암호화 키 및 키 링

각 PKI 디바이스는 비대칭 RSA(Rivest-Shamir-Adleman) 암호화 키의 쌍을 보유합니다. 개인 키와 공개 키로 구성된 이 쌍은 내부 키 링에 저장됩니다. 두 키 중 하나로 암호화한 메시지는 나머지 키로 해독할 수 있습니다. 암호화된 메시지를 보낼 때 발신자는 수신자의 공개 키로 메시지를 암호화하며 수신자는 자신의 개인 키로 그 메시지를 해독합니다. 또한 발신자는 자체 개인 키로 알려진 메시지를 암호화('서명'이라고도 함)하여 공개 키의 소유권을 증명할 수도 있습니다. 수신자가 해당 공개 키를 사용하여 성공적으로 메시지를 해독할 수 있다면 발신자가 개인 키를 소유하고 있음이 입증됩니다. 암호화 키의 길이는 다양하지만, 일반적으로 512바이트 ~ 2048바이트입니다. 일반적으로는 길이가 더 긴 키가 짧은 키보다 안전합니다. FXOS에서는 초기 2048비트 키 쌍으로 기본 키 링을 제공하며 사용자가 추가 키 링을 생성할 수 있습니다.

클러스터 이름이 바뀌거나 인증서가 만료될 경우 기본 키 링 인증서를 수동으로 재생성해야 합니다.

인증서

안전한 통신을 위해 일차적으로 두 디바이스가 디지털 인증서를 교환합니다. 인증서는 디바이스 공개 키 및 디바이스 ID에 대한 서명된 정보를 포함하는 파일입니다. 디바이스에서 단순히 암호화된 통신을 지원하기 위해서는 자신의 키 쌍 및 자체 서명된 인증서를 생성할 수 있습니다. 원격 사용자가 자체 서명 인증서가 있는 디바이스에 연결할 경우 이 사용자가 디바이스의 ID를 용이하게 확인할 방법이 없으므로 사용자의 브라우저는 초기에 인증 경고를 표시합니다. 기본적으로 FXOS에는 기본 키 링의 공개 키를 포함하는 자체 서명 인증서가 내장되어 있습니다.

신뢰 지점

FXOS에 대한 더 강력한 인증을 제공하기 위해 신뢰할 수 있는 소스 또는 트러스트 포인트로부터 디바이스의 ID를 확인하는 서드파티 인증서를 얻어 설치할 수 있습니다. 서드파티 인증서는 해당 신뢰 지점에서 서명하는데, 이는 루트 CA(certificate authority), 중간 CA 또는 루트 CA로 연결되는 신뢰 체인의 일부인 Trust anchor가 될 수 있습니다. 새 인증서를 얻으려면 FXOS를 통해 인증서 요청을 생성하고 트러스트 포인트에 해당 요청을 제출해야 합니다.

중요 인증서는 Base64 인코딩 X.509(CER) 형식이어야 합니다.

키 링 생성

FXOS는 기본 키 링을 포함하여 최대 8개의 키 링을 지원합니다.

프로시저

단계 **1** 보안 모드를 입력합니다.

Firepower-chassis # **scope security**

단계 **2** 키 링의 이름을 생성합니다.

Firepower-chassis # **create keyring** *keyring-name*

단계 **3** SSL 키 길이(비트)를 설정합니다.

Firepower-chassis # **set modulus** {**mod1024** | **mod1536** | **mod2048** | **mod512**}

단계 **4** 트랜잭션을 커밋합니다.

Firepower-chassis # **commit-buffer**

예

다음 예에서는 키 크기 1024비트의 키 링을 생성합니다.

```
Firepower-chassis# scope security
Firepower-chassis /security # create keyring kr220
Firepower-chassis /security/keyring* # set modulus mod1024
Firepower-chassis /security/keyring* # commit-buffer
Firepower-chassis /security/keyring #
```

다음에 수행할 작업

이 키 링에 대한 인증서 요청을 생성합니다.

기본 키 링 재생성

클러스터 이름이 바뀌거나 인증서가 만료될 경우 기본 키 링 인증서를 수동으로 재생성해야 합니다.

참고 기본 키 링은 FXOS의 FCM에서만 사용됩니다.

프로시저

단계 **1** 보안 모드를 입력합니다.

Firepower-chassis # **scope security**

단계 **2** 기본 키 링에 대한 키 링 보안 모드로 들어갑니다.

Firepower-chassis /security # **scope keyring default**

단계 **3** 기본 키 링 재생성:

Firepower-chassis /security/keyring # **set regenerate yes**

단계 **4** 트랜잭션을 커밋합니다.

Firepower-chassis # **commit-buffer**

예

다음 예에서는 기본 키 링을 재생성합니다.

```
Firepower-chassis# scope security
Firepower-chassis /security # scope keyring default
Firepower-chassis /security/keyring* # set regenerate yes
Firepower-chassis /security/keyring* # commit-buffer
Firepower-chassis /security/keyring #
```

키 링에 대한 인증서 요청 생성

기본 옵션으로 키 링에 대한 인증서 요청 생성

프로시저

단계 **1** 보안 모드를 입력합니다.

Firepower-chassis # **scope security**

단계 **2** 키 링에 대한 구성 모드로 들어갑니다.

Firepower-chassis /security # **scope keyring** *keyring-name*

단계 **3** 지정된 IPv4 또는 IPv6 주소 또는 fabric interconnect의 이름을 사용하여 인증서 요청을 만듭니다. 인증서 요청에 대한 비밀번호를 입력하라는 프롬프트가 표시됩니다.

Firepower-chassis /security/keyring # **create certreq** {**ip** [*ipv4-addr* | *ipv6-v6*] | **subject-name** *name*}

단계 **4** 트랜잭션을 커밋합니다.

Firepower-chassis /security/keyring/certreq # **commit-buffer**

단계 **5** Trust anchor 또는 인증 증명으로 복사하여 전송할 수 있는 인증서 요청을 표시합니다.

Firepower-chassis /security/keyring # **show certreq**

예

다음 예는 기본 옵션으로 키 링에 대한 IPv4 주소와 함께 인증서 요청을 만들고 표시합니다.

```
Firepower-chassis# scope security
Firepower-chassis /security # scope keyring kr220
Firepower-chassis /security/keyring # create certreq ip 192.168.200.123 subject-name sjc04
Certificate request password:
Confirm certificate request password:
Firepower-chassis /security/keyring* # commit-buffer
Firepower-chassis /security/keyring # show certreq
Certificate request subject name: sjc04
Certificate request ip address: 192.168.200.123
Certificate request e-mail name:
Certificate request country name:
State, province or county (full name):
Locality (eg, city):
Organization name (eg, company):
Organization Unit name (eg, section):
Request:
-----BEGIN CERTIFICATE REQUEST-----
MIIBfTCB5wIBADARMQ8wDQYDVQQDEwZzYW1jMDQwgZ8wDQYJKoZIhvcNAQEBBQAD
gY0AMIGJAoGBALpKn1t8qMZO4UGqILKFXQQc2c8b/vW2rnRF8OPhKbhghLA1YZ1F
JqcYEG5Yl1+vgohLBTd45s0GC8m4RTLJWHo4SwccAUXQ5Zngf45YtX1WsylwUWV4
0re/zgTk/WCd56RfOBvWR2Dtztu2pGA14sd761zLxt29K7R8mzj6CAUVAgMBAAGg
LTArBgkqhkiG9w0BCQ4xHjAcMBoGA1UdEQEB/wQQMA6CBnNhbWMwNIcECsEiXjAN
BgkqhkiG9w0BAQQFAAOBgQCsxN0qUHYGFoQw56RwQueLTNPnrndqUwuZHUO03Teg
nhsyu4satpyiPqVV9viKZ+spvc6x5PWIcTWgHhH8BimOb/0OKuG8kwfIGGsEDlAv
TTYvUP+BZ9OFiPbRIA718S+V8ndXr1HejiQGxlDNqoN+odCXPc5kjoXD0lZTL09H
BA==
-----END CERTIFICATE REQUEST-----

Firepower-chassis /security/keyring #
```

다음에 수행할 작업

- BEGIN 및 END 줄을 포함한 인증서 요청의 텍스트를 복사하여 파일에 저장합니다. 키 링에 대한 인증서를 얻을 수 있도록, 인증서 요청이 포함된 파일을 Trust anchor 또는 인증 증명으로 전송합니다.
- 신뢰 지점을 생성하고 Trust anchor로부터 받은 신뢰 인증서에 대한 인증서 체인을 설정합니다.

고급 옵션으로 키 링에 대한 인증서 요청 생성

프로시저

단계 **1** 보안 모드를 입력합니다.

Firepower-chassis # **scope security**

단계 **2** 키 링에 대한 구성 모드로 들어갑니다.

Firepower-chassis /security # **scope keyring** *keyring-name*

단계 **3** 인증서 요청을 생성합니다.

Firepower-chassis /security/keyring # **create certreq**

단계 **4** 회사가 소재한 국가의 국가 코드를 지정합니다.

Firepower-chassis /security/keyring/certreq* # **set country** *country name*

단계 **5** 요청과 연결된 DNS(Domain Name Server) 주소를 지정합니다.

Firepower-chassis /security/keyring/certreq* # **set dns** *DNS Name*

단계 **6** 인증서 요청과 연결된 이메일 주소를 지정합니다.

Firepower-chassis /security/keyring/certreq* # **set e-mail** *E-mail name*

단계 **7** Firepower 4100/9300 섀시의 IP 주소를 지정합니다.

Firepower-chassis /security/keyring/certreq*# **set ip** {*certificate request ip-address*|*certificate request ip6-address*}

단계 **8** 인증서를 요청하는 회사의 본사가 위치한 시/읍/면을 지정합니다.

Firepower-chassis /security/keyring/certreq* # **set locality** *locality name (eg, city)*

단계 **9** 인증서를 요청하는 조직을 지정합니다.

Firepower-chassis /security/keyring/certreq* # **set org-name** *organization name*

단계 **10** 조직 단위를 지정합니다.

Firepower-chassis /security/keyring/certreq* # **set org-unit-name** *organizational unit name*

단계 **11** 인증서 요청에 대한 비밀번호를 지정합니다(선택 사항).

Firepower-chassis /security/keyring/certreq* # **set password** *certificate request password*

단계 **12** 인증서를 요청하는 회사의 본사가 위치한 시/도를 지정합니다.

Firepower-chassis /security/keyring/certreq* # **set state** *state, province or county*

단계 **13** Firepower 4100/9300 섀시의 FQDN(Fully Qualified Domain Name)을 지정합니다.

Firepower-chassis /security/keyring/certreq* # **set subject-name** *certificate request name*

단계 **14** 트랜잭션을 커밋합니다.

Firepower-chassis /security/keyring/certreq # **commit-buffer**

단계 **15** Trust anchor 또는 인증 증명으로 복사하여 전송할 수 있는 인증서 요청을 표시합니다.

Firepower-chassis /security/keyring # **show certreq**

예

참고 2.7 이전 릴리스의 경우 FQDN 없이 "set dns" 또는 "set subject-name"을 사용하여 버퍼를 커밋하지 않는 것이 좋습니다. FQDN이 아닌 DNS 또는 주체 이름으로 인증 요구 사항을 생성하려고 하면 오류가 발생합니다.

다음 예는 고급 옵션으로 키 링에 대한 IPv4 주소와 함께 인증서 요청을 만들고 표시합니다.

```
Firepower-chassis# scope security
Firepower-chassis /security # scope keyring kr220
Firepower-chassis /security/keyring # create certreq
Firepower-chassis /security/keyring/certreq* # set "ip 192.168.200.123"
Firepower-chassis /security/keyring/certreq* #  set subject-name "sjc04"
Firepower-chassis /security/keyring/certreq* # set country "US"
Firepower-chassis /security/keyring/certreq* # set dns "bg1-samc-15A"
Firepower-chassis /security/keyring/certreq* # set email "test@cisco.com"
Firepower-chassis /security/keyring/certreq* # set locality "new york city"
Firepower-chassis /security/keyring/certreq* # set org-name "Cisco Systems"
Firepower-chassis /security/keyring/certreq* # set org-unit-name "Testing"
Firepower-chassis /security/keyring/certreq* # set state "new york"
Firepower-chassis /security/keyring/certreq* # commit-buffer
Firepower-chassis /security/keyring/certreq # show certreq
Certificate request subject name: sjc04
Certificate request ip address: 192.168.200.123
Certificate request e-mail name: test@cisco.com
Certificate request country name: US
State, province or county (full name): New York
Locality name (eg, city): new york city
Organization name (eg, company): Cisco
Organization Unit name (eg, section): Testing
Request:
-----BEGIN CERTIFICATE REQUEST-----
MIIBfTCB5wIBADARMQ8wDQYDVQQDEwZzYW1jMDQwgZ8wDQYJKoZIhvcNAQEBBQAD
gY0AMIGJAoGBALpKn1t8qMZO4UGqILKFXQQc2c8b/vW2rnRF8OPhKbhghLA1YZ1F
JqcYEG5Yl1+vgohLBTd45s0GC8m4RTLJWHo4SwccAUXQ5Zngf45YtX1WsylwUWV4
0re/zgTk/WCd56RfOBvWR2Dtztu2pGA14sd761zLxt29K7R8mzj6CAUVAgMBAAGg
LTArBgkqhkiG9w0BCQ4xHjAcMBoGA1UdEQEB/wQQMA6CBnNhbWMwNIcECsEiXjAN
BgkqhkiG9w0BAQQFAAOBgQCsxN0qUHYGFoQw56RwQueLTNPnrndqUwuZHUO03Teg
nhsyu4satpyiPqVV9viKZ+spvc6x5PWIcTWgHhH8BimOb/0OKuG8kwfIGGsEDlAv
TTYvUP+BZ9OFiPbRIA718S+V8ndXr1HejiQGxlDNqoN+odCXPc5kjoXD0lZTL09H
BA==
-----END CERTIFICATE REQUEST-----

Firepower-chassis /security/keyring/certreq #
```

다음에 수행할 작업

- BEGIN 및 END 줄을 포함한 인증서 요청의 텍스트를 복사하여 파일에 저장합니다. 키 링에 대한 인증서를 얻을 수 있도록, 인증서 요청이 포함된 파일을 Trust anchor 또는 인증 증명으로 전송합니다.
- 신뢰 지점을 생성하고 Trust anchor로부터 받은 신뢰 인증서에 대한 인증서 체인을 설정합니다.

트러스트 포인트 생성

프로시저

단계 **1** 보안 모드를 입력합니다.

Firepower-chassis # **scope security**

단계 **2** 신뢰 지점을 생성합니다.

Firepower-chassis /security # **create trustpoint** *name*

단계 **3** 이 신뢰 지점에 대한 인증서 정보를 지정합니다.

Firepower-chassis /security/trustpoint # **set certchain** [*certchain*]

명령에 인증서 정보를 지정하지 않은 경우, 루트 CA(Certificate Authority)에 인증 경로를 정의하는 Trust Point 목록 또는 인증서를 입력하라는 프롬프트가 표시됩니다. 해당 정보를 입력한 후 다음 행에 **ENDOFBUF**를 입력하여 완료합니다.

중요 인증서는 Base64 인코딩 X.509(CER) 형식이어야 합니다.

단계 **4** 트랜잭션을 커밋합니다.

Firepower-chassis /security/trustpoint # **commit-buffer**

예

다음 예에서는 신뢰 지점을 만들고 신뢰 지점에 대한 인증서를 제공합니다.

```
Firepower-chassis# scope security
Firepower-chassis /security # create trustpoint tPoint10
Firepower-chassis /security/trustpoint* # set certchain
Enter lines one at a time. Enter ENDOFBUF to finish. Press ^C to abort.
Trustpoint Certificate Chain:
> -----BEGIN CERTIFICATE-----
> MIIDMDCCApmgAwIBAgIBADANBgkqhkiG9w0BAQQFADB0MQswCQYDVQQGEwJVUzEL
> BxMMU2FuIEpvc2UsIENBMRUwEwYDVQQKEwxFeGFtcGxlIEluYy4xEzARBgNVBAsT
> ClRlc3QgR3JvdXAxGTAXBgNVBAMTEHRlc3QuZXhhbXBsZS5jb20xHzAdBgkqhkiG
> 9w0BCQEWEHVzZXJAZXhhbXBsZS5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJ
> AoGBAMZw4nTepNIDhVzb0j7Z2Je4xAG56zmSHRMQeOGHemdh66u2/XAoLx7YCcYU
> ZgAMivyCsKgb/6CjQtsofvtrmC/eAehuK3/SINv7wd6Vv2pBt6ZpXgD4VBNKOND1
> GMbkPayVlQjbG4MD2dx2+H8EH3LMtdZrgKvPxPTE+bF5wZVNAgMBAAGgJTAjBgkq
> hkiG9w0BCQcxFhMUQSBjaGFsbGVuZ2UgcGFzc3dvcmQwDQYJKoZIhvcNAQEFBQAD
> gYEAG61CaJoJaVMhzCl903O6Mg51zq1zXcz75+VFj2I6rH9asckCld3mkOVx5gJU
> Ptt5CVQpNgNLdvbDPSsXretysOhqHmp9+CLv8FDuy1CDYfuaLtvlWvfhevskV0j6
> jtcEMyZ+f7+3yh421ido3nO4MIGeBgNVHSMEgZYwgZOAFLlNjtcEMyZ+f7+3yh42
> 1ido3nO4oXikdjB0MQswCQYDVQQGEwJVUzELMAkGA1UECBMCQ0ExFDASBgNVBAcT
> C1NhbnRhIENsYXJhMRswGQYDVQQKExJOdW92YSBTeXN0ZW1zIEluYy4xFDASBgNV
> BAsTC0VuZ2luZWVyaW5nMQ8wDQYDVQQDEwZ0ZXN0Q0GCAQAwDAYDVR0TBAUwAwEB
> /zANBgkqhkiG9w0BAQQFAAOBgQAhWaRwXNR6B4g6Lsnr+fptHv+WVhB5fKqGQqXc
> wR4pYiO4z42/j9Ijenh75tCKMhW51az8copP1EBmOcyuhf5C6vasrenn1ddkkYt4
> PR0vxGc40whuiozBolesmsmjBbedUCwQgdFDWhDIZJwK5+N3x/kfa2EHU6id1avt
```

```
> 4YL5Jg==
> -----END CERTIFICATE-----
> ENDOFBUF
Firepower-chassis /security/trustpoint* # commit-buffer
Firepower-chassis /security/trustpoint #
```

다음에 수행할 작업

Trust anchor 또는 인증 증명에서 키 링 인증서를 받아 키 링으로 가져옵니다.

키 링으로 인증서 가져오기

시작하기 전에

- 키 링 인증서에 대한 인증서 체인을 포함하는 신뢰 지점을 구성합니다.
- Trust anchor 또는 인증 증명에서 키 링 인증서를 가져옵니다.

참고 HTTPS에 이미 구성된 키 링에서 인증서를 변경하는 경우 새 인증서를 적용하려면 HTTPS를 다시 시작해야 합니다. 자세한 내용은 HTTPS 재시작, 139 페이지를 참조하십시오.

프로시저

단계 **1** 보안 모드를 입력합니다.

Firepower-chassis # **scope security**

단계 **2** 인증서를 수신할 키 링에 대한 구성 모드로 들어갑니다.

Firepower-chassis /security # **scope keyring** *keyring-name*

단계 **3** 키 링 인증서를 수신한 Trust anchor 또는 인증 증명에 대한 신뢰 지점을 지정합니다.

Firepower-chassis /security/keyring # **set trustpoint** *name*

단계 **4** 키 링 인증서를 입력 및 업로드할 대화 상자를 엽니다.

Firepower-chassis /security/keyring # **set cert**

프롬프트에 Trust anchor 또는 인증 증명으로부터 받은 인증서의 텍스트를 붙여넣습니다. 인증서의 바로 다음 줄에 **ENDOFBUF**를 입력하여 인증서 입력을 완료합니다.

중요 인증서는 Base64 인코딩 X.509(CER) 형식이어야 합니다.

단계 **5** 트랜잭션을 커밋합니다.

Firepower-chassis /security/keyring # **commit-buffer**

예

다음 예에서는 신뢰 지점을 지정하고 인증서를 키 링으로 가져옵니다.

```
Firepower-chassis# scope security
Firepower-chassis /security # scope keyring kr220
Firepower-chassis /security/keyring # set trustpoint tPoint10
Firepower-chassis /security/keyring* # set cert
Enter lines one at a time. Enter ENDOFBUF to finish. Press ^C to abort.
Keyring certificate:
> -----BEGIN CERTIFICATE-----
> MIIB/zCCAWgCAQAwgZkxCzAJBgNVBAYTAlVTMQswCQYDVQQIEwJDQTEVMBMGA1UE
> BxMMU2FuIEpvc2UsIENBMRUwEwYDVQQKEwxFeGFtcGxlIEluYy4xEzARBgNVBAsT
> ClRlc3QgR3JvdXAxGTAXBgNVBAMTEHRlc3QuZXhhbXBsZS5jb20xHzAdBgkqhkiG
> 9w0BCQEWEHVzZXJAZXhhbXBsZS5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJ
> AoGBAMZw4nTepNIDhVzb0j7Z2Je4xAG56zmSHRMQeOGHemdh66u2/XAoLx7YCcYU
> ZgAMivyCsKgb/6CjQtsofvtrmC/eAehuK3/SINv7wd6Vv2pBt6ZpXgD4VBNKONDl
> GMbkPayVlQjbG4MD2dx2+H8EH3LMtdZrgKvPxPTE+bF5wZVNAgMBAAGgJTAjBgkq
> hkiG9w0BCQcxFhMUQSBjaGFsbGVuZ2UgcGFzc3dvcmQwDQYJKoZIhvcNAQEFBQAD
> gYEAG61CaJoJaVMhzCl903O6Mg51zq1zXcz75+VFj2I6rH9asckCld3mkOVx5gJU
> Ptt5CVQpNgNLdvbDPSsXretysOhqHmp9+CLv8FDuy1CDYfuaLtvlWvfhevskV0j6
> mK3Ku+YiORnv6DhxrOoqau8r/hyI/L43l7IPN1HhOi3oha4=
> -----END CERTIFICATE-----
> ENDOFBUF
Firepower-chassis /security/keyring* # commit-buffer
Firepower-chassis /security/keyring #
```

다음에 수행할 작업

HTTPS 서비스를 키 링으로 구성합니다.

HTTPS 구성

⚠

주의 HTTPS에서 사용하는 포트 및 키 링 변경을 포함하여 HTTPS 구성을 완료한 후 트랜잭션을 저장하거나 커밋하자마자 모든 현재 HTTP 및 HTTPS 세션이 종료됩니다.

프로시저

단계 **1** 시스템 모드를 입력합니다.

Firepower-chassis# **scope system**

단계 **2** 시스템 서비스 모드를 입력합니다.

Firepower-chassis /system #**scope services**

단계 **3** HTTPS 서비스를 활성화합니다.

Firepower-chassis /system/services # **enable https**

단계 **4** (선택 사항) HTTPS 연결에 사용할 포트를 지정합니다.

Firepower-chassis /system/services # **set https port** *port-num*

단계 **5** (선택 사항) HTTPS에 대해 생성한 키 링의 이름을 지정합니다.

Firepower-chassis /system/services # **set https keyring** *keyring-name*

단계 **6** (선택 사항) 도메인에서 사용하는 Cipher Suite 보안 레벨을 지정합니다.

Firepower-chassis /system/services # **set https cipher-suite-mode** *cipher-suite-mode*

*cipher-suite-mode*는 다음 키워드 중 하나일 수 있습니다.

- **high-strength**
- **medium-strength**
- **low-strength**
- **custom**- 사용자 정의 Cipher Suite 사양 문자열을 지정할 수 있습니다.

단계 **7** (선택 사항) **cipher-suite-mode**가 **custom**으로 설정된 경우 도메인에 대한 Cipher Suite 보안의 커스텀 레벨을 지정합니다.

Firepower-chassis /system/services # **set https cipher-suite** *cipher-suite-spec-string*

*cipher-suite-spec-string*은 최대 256자이며 OpenSSL Cipher Suite 사양을 준수해야 합니다. 공백 또는 특수 문자를 사용할 수 없습니다. 단, !(느낌표), +(덧셈 기호), -(하이픈), :(콜론)은 사용할 수 있습니다. 자세한 내용은 http://httpd.apache.org/docs/2.0/mod/mod_ssl.html#sslciphersuite를 참조하십시오.

예를 들어 FXOS에서 기본값으로 사용하는 중간 강도 사양 문자열은 다음과 같습니다.
ALL:!ADH:!EXPORT56:!LOW:RC4+RSA:+HIGH:+MEDIUM:+EXP:+eNULL

참고 **cipher-suite-mode**가 **custom** 이외의 값으로 설정되어 있으면 이 옵션은 무시됩니다.

단계 **8** (선택 사항) 인증서 해지 목록 확인을 활성화 또는 비활성화합니다.

set revoke-policy *{ relaxed | strict }*

단계 **9** 시스템 구성에 트랜잭션을 커밋합니다.

Firepower-chassis /system/services # **commit-buffer**

예

다음 예에서는 HTTPS를 활성화하고, 포터 번호를 443으로 설정하고, 키 링 이름을 kring7984로 설정하고, Cipher Suite 보안 레벨을 high로 설정하고, 트랜잭션을 커밋합니다.

```
Firepower-chassis# scope system
Firepower-chassis /system # scope services
Firepower-chassis /system/services # enable https
Firepower-chassis /system/services* # set https port 443
Warning: When committed, this closes all the web sessions.
Firepower-chassis /system/services* # set https keyring kring7984
Firepower-chassis /system/services* # set https cipher-suite-mode high
```

```
Firepower-chassis /system/services* # commit-buffer
Firepower-chassis /system/services #
```

HTTPS 포트 변경

HTTPS 서비스는 기본적으로 포트 443에서 활성화되어 있습니다. HTTPS를 비활성화할 수는 없지만, HTTPS 연결에 사용할 포트를 변경할 수 있습니다.

프로시저

단계 **1** **Platform Settings(**플랫폼 설정**)** > **HTTPS**를 선택합니다.

단계 **2** HTTPS 연결에 사용할 포트를 **Port(**포트**)** 필드에 입력합니다. 1~65535 사이의 정수를 입력합니다. 이 서비스는 기본적으로 포트 443에서 활성화됩니다.

단계 **3** **Save(**저장**)**를 클릭합니다.

섀시는 HTTPS 포트가 지정된 상태로 구성됩니다.

HTTPS 포트를 변경한 후에는 현재의 모든 HTTPS 세션이 종료됩니다. 사용자는 다음과 같이 새 포트를 사용하여 Firepower Chassis Manager에 다시 로그인해야 합니다.

https://<*chassis_mgmt_ip_address*>:<*chassis_mgmt_port*>

이때 <*chassis_mgmt_ip_address*>는 사용자가 초기 구성을 설정하는 동안 입력한 섀시의 IP 주소 또는 호스트 이름이며 <*chassis_mgmt_port*>는 방금 구성한 HTTPS 포트입니다.

HTTPS 재시작

HTTPS에 이미 구성된 키 링에서 인증서를 변경하는 경우, 새 인증서를 적용하려면 HTTPS를 다시 시작해야 합니다. 업데이트된 키 링으로 HTTPS를 재설정하려면 다음 절차를 사용합니다.

프로시저

단계 **1** 시스템 모드를 입력합니다.

Firepower-chassis# **scope system**

단계 **2** 시스템 서비스 모드를 입력합니다.

Firepower-chassis /system #**scope services**

단계 **3** HTTPS 키 링을 기본값으로 다시 설정합니다.

Firepower-chassis /system/services # **set https keyring default**

단계 **4** 시스템 구성에 트랜잭션을 커밋합니다.

Firepower-chassis /system/services # **commit-buffer**

단계 **5** 5초 동안 기다립니다.

단계 **6** 생성한 키 링으로 HTTPS를 설정합니다.

Firepower-chassis /system/services # **set https keyring** *keyring-name*

단계 **7** 시스템 구성에 트랜잭션을 커밋합니다.

Firepower-chassis /system/services # **commit-buffer**

키 링 삭제

프로시저

단계 **1** 보안 모드를 입력합니다.

Firepower-chassis # **scope security**

단계 **2** 명명된 키 링을 삭제합니다.

Firepower-chassis /security # **delete keyring** *name*

단계 **3** 트랜잭션을 커밋합니다.

Firepower-chassis /security # **commit-buffer**

예

다음 예에서는 사용자 계정을 삭제합니다.

```
Firepower-chassis# scope security
Firepower-chassis /security # delete keyring key10
Firepower-chassis /security* # commit-buffer
Firepower-chassis /security #
```

트러스트 포인트 삭제

시작하기 전에

신뢰 지점이 키 링에서 사용하지 않음을 확인합니다.

프로시저

단계 **1** 보안 모드로 들어갑니다.

Firepower-chassis# **scope security**

단계 **2** 명명된 신뢰 지점을 삭제합니다.

Firepower-chassis /security # **delete trustpoint** *name*

단계 **3** 트랜잭션을 커밋합니다.

Firepower-chassis /security # **commit-buffer**

예

다음 예에서는 신뢰 지점을 삭제합니다.

```
Firepower-chassis# scope security
Firepower-chassis /security # delete trustpoint tPoint10
Firepower-chassis /security* # commit-buffer
Firepower-chassis /security #
```

HTTPS 비활성화

프로시저

단계 **1** 시스템 모드를 입력합니다.

Firepower-chassis# **scope system**

단계 **2** 시스템 서비스 모드를 입력합니다.

Firepower-chassis /system # **scope services**

단계 **3** HTTPS 서비스를 비활성화합니다.

Firepower-chassis /system/services # **disable https**

단계 **4** 시스템 구성에 트랜잭션을 커밋합니다.

Firepower-chassis /system/services # **commit-buffer**

예

다음 예에서는 HTTPS를 비활성화하고 트랜잭션을 커밋합니다.

```
Firepower-chassis# scope system
Firepower-chassis /system # scope services
Firepower-chassis /system/services # disable https
Firepower-chassis /system/services* # commit-buffer
Firepower-chassis /system/services #
```

AAA 구성

이 섹션에서는 인증, 권한 부여 및 어카운팅에 관해 설명합니다. 자세한 내용은 다음 항목을 참고하십시오.

AAA 정보

AAA(인증, 권한 부여 및 계정 관리)는 네트워크 리소스에 대한 액세스 제어를 위한 서비스의 집합으로, 정책을 구현하고, 사용량을 평가하고 서비스에 대한 청구에 필요한 정보를 제공합니다. 인증은 사용자를 식별합니다. 권한 부여는 사용자가 액세스할 수 있는 리소스와 서비스를 결정하는 정책을 구현합니다. 어카운팅은 청구 및 분석을 위해 사용되는 시간과 데이터를 추적합니다. 이 과정은 효과적인 네트워크 관리 및 보안을 위해 중요한 부분으로 간주됩니다.

인증

인증은 액세스를 부여하기 전에 사용자가 유효한 사용자 이름과 유효한 암호를 입력하게 하여 각 사용자를 식별하는 방법을 제공합니다. AAA 서버는 사용자의 인증 크리덴셜을 데이터베이스에 저장된 다른 사용자의 크리덴셜과 비교합니다. 크리덴셜이 일치하면 사용자는 네트워크에 액세스할 수 있습니다. 크리덴셜이 일치하지 않으면, 인증에 실패하고 네트워크 액세스가 거부됩니다.

다음 세션을 포함하는 섀시에 대한 관리 연결을 인증하도록 Firepower 4100/9300 섀시를 구성할 수 있습니다.

- HTTPS
- SSH
- 시리얼 콘솔

권한 부여

권한 부여는 정책을 구현하는 프로세스로, 사용자의 액세스가 허용된 활동, 리소스 또는 서비스 유형을 결정합니다. 권한이 부여되면, 사용자는 다양한 액세스 또는 활동 유형에 대한 권한을 가질 수 있습니다.

어카운팅

어카운팅은 사용자가 액세스 중 소비하는 리소스를 측정합니다. 여기에는 시스템 사용 시간, 사용자가 세션 중 보내거나 받는 데이터의 양 등이 포함됩니다. 어카운팅은 세션 통계 및 사용량 정보 기록을 통해 이루어지며 이는 권한 부여 제어, 청구, 경향 분석, 리소스 활용도 및 용량 계획 활동에 사용됩니다.

인증, 권한 부여 및 어카운팅 간 상호 작용

인증을 단독으로 사용하거나 권한 부여 및 어카운팅과 함께 사용할 수 있습니다. 권한 부여에서는 항상 먼저 사용자의 인증 여부를 확인해야 합니다. 어카운팅을 단독으로 사용하거나 인증 및 권한 부여와 함께 사용할 수 있습니다.

지원되는 인증 유형

FXOS는 다음 유형의 사용자 인증을 지원합니다.

- 원격 - 다음 네트워크 AAA 서비스가 지원됩니다.
  - LDAP
  - RADIUS
  - TACACS+
- 로컬 - 섀시는 사용자가 사용자 프로파일을 채울 수 있는 로컬 데이터베이스를 유지합니다. AAA 서버 대신 로컬 데이터베이스를 사용하여 사용자 인증, 권한 부여 및 어카운팅을 제공할 수 있습니다.

사용자 역할

FXOS는 사용자 역할 할당 형식으로 로컬 및 원격 권한 부여를 지원합니다. 할당할 수 있는 역할은 다음과 같습니다.

- **Admin** - 전체 시스템에 대한 완전한 읽기 및 쓰기 액세스. 기본 관리자 계정이 기본적으로 이 역할에 할당되며 변경할 수 없습니다.
- **AAA Administrator** - 사용자, 역할 및 AAA 구성에 대한 읽기-쓰기 액세스. 나머지 시스템에 대한 읽기 액세스
- **Operations** - NTP 구성, 스마트 라이센싱을 위한 Smart Call Home 구성, 시스템 로그(syslog 서버 및 장애 포함)에 대한 읽기 및 쓰기 액세스. 나머지 시스템에 대한 읽기 액세스
- **Read-Only** - 시스템 상태를 수정할 수 있는 권한이 없는 시스템 구성에 대한 읽기 전용 액세스

로컬 사용자 및 역할 할당에 대한 자세한 내용은 사용자 관리, 43 페이지 섹션을 참조하십시오.

AAA 설정

이 단계에서는 Firepower 4100/9300 어플라이언스에서 AAA(Authentication, Authorization and Accounting)를 설정하기 위한 기본 개요를 제공합니다.

1. 원하는 사용자 인증 유형을 구성합니다.
   - 로컬 - 사용자 정의 및 로컬 인증이 사용자 관리, 43 페이지의 일부입니다.
   - 원격 - 원격 AAA 서버 액세스 구성은 플랫폼 설정의 일부입니다. 구체적으로는 다음과 같습니다.
     - LDAP 제공자 구성, 144 페이지
     - RADIUS 제공자 구성, 148 페이지
     - TACACS+ 제공자 구성, 150 페이지

참고 원격 AAA 서버를 사용하려는 경우, 섀시에서 원격 AAA 서버 액세스를 구성하기 전에 원격 서버에서 AAA 서비스를 활성화하고 구성해야 합니다.

2. 기본 인증 방법을 지정합니다. 이 역시 사용자 관리, 43 페이지의 일부입니다.

참고 Default Authentication(기본 인증) 및 Console Authentication(콘솔 인증)이 모두 동일한 원격 인증 프로토콜(RADIUS, TACACS+ 또는 LDAP)을 사용하도록 설정된 경우, 이러한 사용자 설정을 업데이트해야 해당 서버 구성의 특정 측면(예: 해당 서버 삭제 또는 할당 순서 변경)을 변경할 수 있습니다.

LDAP 제공자 구성

LDAP 제공자 속성 구성

이 작업에서 구성하는 속성은 이 유형의 모든 제공자 연결에 대한 기본 설정입니다. 개별 제공자에 이러한 속성 중 하나에 대한 설정이 포함된 경우 FXOS에서는 해당 설정을 사용하고 기본 설정을 무시합니다.

Active Directory를 LDAP 서버로 사용하는 경우에는 Active Directory 서버에서 사용자 어카운트를 생성하여 FXOS와 바인딩합니다. 이 계정은 만료되지 않는 비밀번호를 가져야 합니다.

프로시저

단계 1 **Platform Settings(**플랫폼 설정**)** > **AAA**를 선택합니다.

단계 2 **LDAP** 탭을 클릭합니다.

단계 3 **Properties(**속성**)** 영역에서 다음 필드를 완성합니다.

이름	설명
Timeout(시간 초과**)** 필드	시간이 초과되기 전에 시스템이 TACACS+ 데이터베이스에 연결을 시도하는 데 필요한 시간(초). 1~60초의 정수를 입력합니다. 기본값은 30초입니다. 이 속성은 필수 항목입니다.
Attribute(속성**)** 필드	사용자 역할 및 로케일에 대해 값을 저장하는 LDAP 속성. 이 속성은 항상 이름값 쌍입니다. 시스템은 사용자 레코드를 쿼리하여 이 속성 이름과 일치하는 값을 찾습니다. LDAP 제공자에 대한 속성을 구성할 때는 `shell:roles="admin,aaa"` 속성 값이 필요합니다.

이름	설명
Base DN(기본 **DN)** 필드	원격 사용자가 로그인하고 시스템에서 사용자 이름에 기초한 사용자 DN을 얻으려고 시도할 때 서버에서 검색을 시작해야 하는 LDAP 계층 구조에서 특정한 고유 이름입니다. 기본 DN 길이는 최대 255자에서 *cn=$userid* 길이를 뺀 문자 수로 설정될 수 있습니다. 이때 $userid는 LDAP 인증을 사용하여 섀시에 액세스를 시도하는 원격 사용자를 식별합니다. 이 속성은 LDAP 제공자에 필요합니다. 이 탭에서 기본 DN을 지정하지 않으면 정의하는 각 LDAP 제공자에 하나를 지정해야 합니다.
Filter(필터**)** 필드	LDAP 서버에 사용할 필터 속성을 입력합니다(예: *cn=$userid* 또는 *sAMAccountName=$userid*). LDAP 검색은 정의된 필터와 일치하는 사용자 이름으로 제한됩니다. 필터는 *$userid*를 포함해야 합니다. 이 속성은 필수 항목입니다. 이 탭에서 필터를 지정하지 않으면 정의하는 각 LDAP 제공자에 하나를 지정해야 합니다.

단계 **4** **Save(**저장**)**를 클릭합니다.

다음에 수행할 작업

LDAP 제공자를 생성합니다.

LDAP 제공자 생성

다음 단계에 따라 LDAP 공급자, 즉 이 어플라이언스에 LDAP 기반 AAA 서비스를 제공하는 특정 원격 서버를 정의하고 구성합니다.

참고 FXOS에서는 최대 16개의 LDAP 제공자를 지원합니다.

시작하기 전에

Active Directory를 LDAP 서버로 사용하는 경우에는 Active Directory 서버에서 사용자 어카운트를 생성하여 FXOS와 바인딩합니다. 이 계정은 만료되지 않는 비밀번호를 가져야 합니다.

프로시저

단계 **1** **Platform Settings(**플랫폼 설정**)** > **AAA**를 선택합니다.

단계 **2** **LDAP** 탭을 클릭합니다.

단계 **3** 추가할 각 LDAP 제공자에 대해 다음을 수행합니다.

a) **LDAP Providers(LDAP** 제공자**)** 영역에서 **Add(**추가**)**를 클릭합니다.

b) **Add LDAP Provider(LDAP** 제공자 추가**)** 대화 상자에서 다음 필드를 작성합니다.

이름	설명
호스트 이름**/FQDN(**또는 **IP** 주소**)** 필드	LDAP 서버의 호스트 이름 또는 IP 주소. SSL이 활성화된 경우 이 필드는 LDAP 데이터베이스 보안 인증서의 CN(Common Name)과 정확히 일치해야 합니다.
Order(순서**)** 필드	FXOS에서 사용자를 인증하기 위해 이 제공자를 사용하는 순서입니다. Firepower Chassis Manager 또는 FXOS CLI에 정의되어 있는 다른 제공자에 기반하여 FXOS이 다음으로 사용 가능한 순서를 할당하게 하려면 1~16 사이의 정수를 입력하거나 **`lowest-available`** 또는 **`0`**(숫자 0)을 입력합니다.
Bind DN(바인드 **DN)** 필드	기본 DN에 속하는 모든 객체에 대한 읽기 및 검색 권한이 있는 LDAP 데이터베이스 어카운트의 고유 이름(DN)입니다. 지원되는 최대 문자열 길이는 ASCII 255자입니다.
Base DN(기본 **DN)** 필드	원격 사용자가 로그인하고 시스템에서 사용자 이름에 기초한 사용자 DN을 얻으려고 시도할 때 서버에서 검색을 시작해야 하는 LDAP 계층 구조에서 특정한 고유 이름입니다. 기본 DN 길이는 최대 255자에서 CN=$userid 길이를 뺀 문자 수로 설정될 수 있습니다. 이때 $userid는 LDAP 인증을 사용하여 Firepower Chassis Manager 또는 FXOS CLI에 액세스를 시도하는 원격 사용자를 식별합니다. 이 값은 기본 DN의 기본값이 **LDAP** 탭에 설정되지 않은 경우 필요합니다.
Port(포트**)** 필드	Firepower Chassis Manager 또는 FXOS CLI에서 LDAP 데이터베이스와 통신할 때 사용하는 포트입니다. 표준 포트 번호는 389입니다.
Enable SSL(SSL 활성화**)** 체크박스	이 확인란을 선택한 경우, LDAP 데이터베이스와의 통신에 암호화가 필요합니다. 이 확인란이 선택되지 않은 경우, 인증 정보는 암호화되지 않은 텍스트로 전송됩니다. LDAP는 STARTTLS를 사용합니다. 이는 포트 389를 사용하여 암호화된 통신을 허용합니다. 참고 STARTTLS 작업을 수행하려면 FXOS 인증서 체인에 LDAP 제공자의 CA 인증서를 설치해야 합니다.

이름	설명
Filter(필터**)** 필드	LDAP 서버에 사용할 필터 속성을 입력합니다(예: cn=$userid 또는 sAMAccountName=$userid). LDAP 검색은 정의된 필터와 일치하는 사용자 이름으로 제한됩니다. 필터는 $userid를 포함해야 합니다. 이 값은 기본 필터가 **LDAP** 탭에 설정되지 않은 경우 필요합니다.
Attribute(속성**)** 필드	사용자 역할 및 로케일에 대해 값을 저장하는 LDAP 속성. 이 속성은 항상 이름값 쌍입니다. 시스템은 사용자 레코드를 쿼리하여 이 속성 이름과 일치하는 값을 찾습니다. 이 값은 기본 속성이 **LDAP** 탭에 설정되지 않은 경우 필요합니다.
Key(키**)** 필드	**Bind DN(**바인드 **DN)** 필드에 지정된 LDAP 데이터베이스 어카운트의 비밀번호입니다. 공백, §(섹션 기호), ?(물음표) 또는 =(등호)를 제외한 모든 표준 ASCII 문자를 입력할 수 있습니다.
Confirm Key(키 확인**)** 필드	확인을 위해 다시 입력하는 LDAP 데이터베이스 비밀번호.
Timeout(시간 초과**)** 필드	시간이 초과되기 전에 시스템이 TACACS+ 데이터베이스에 연결을 시도하는 데 필요한 시간(초). 1~60초의 정수를 입력하거나 0(숫자 0)을 입력하여 **LDAP** 탭에 지정된 전역 시간제한 값을 사용합니다. 기본값은 30초입니다.
Vendor(벤더**)** 필드	이 선택사항으로 LDAP 제공자 또는 서버 상세 정보를 제공하는 벤더를 식별합니다. • LDAP 제공자가 Microsoft Active Directory인 경우, **MS AD**를 선택합니다. • LDAP 제공자가 Microsoft Active Directory가 아닌 경우, **Open LDAP(LDAP** 열기**)**를 선택합니다. 기본값은 **Open LDAP(LDAP** 열기**)**입니다.

c) **OK(**확인**)**를 클릭하여 **Add LDAP Provider(LDAP** 제공자 추가**)** 대화 상자를 닫습니다.

단계 **4** **Save(**저장**)**를 클릭합니다.

단계 **5** (선택 사항) 인증서 해지 목록 확인을 활성화합니다.

Firepower-chassis /security/ldap/server # **set revoke-policy** *{strict | relaxed}*

참고 이 구성은 SSL 연결이 활성화된 경우에만 적용됩니다.

LDAP 제공자 삭제

프로시저

단계 1 **Platform Settings(**플랫폼 설정**)** > **AAA**를 선택합니다.

단계 2 **LDAP** 탭을 클릭합니다.

단계 3 **LDAP Providers(LDAP** 제공자**)** 영역에서 삭제할 LDAP 제공자에 해당하는 테이블의 행에 있는 **Delete(**삭제**)** 아이콘을 클릭합니다.

RADIUS 제공자 구성

RADIUS 제공자의 속성 구성

이 작업에서 구성하는 속성은 이 유형의 모든 제공자 연결에 대한 기본 설정입니다. 개별 제공자에 이러한 속성 중 하나에 대한 설정이 포함된 경우, FXOS에서는 해당 설정을 사용하고 기본 설정을 무시합니다.

프로시저

단계 1 **Platform Settings(**플랫폼 설정**)** > **AAA**를 선택합니다.

단계 2 **RADIUS** 탭을 클릭합니다.

단계 3 **Properties(**속성**)** 영역에서 다음 필드를 완성합니다.

이름	설명
Timeout(시간 초과**)** 필드	시간이 초과되기 전에 시스템이 RADIUS 데이터베이스에 연결을 시도하는 데 필요한 시간(초)입니다. 1~60의 정수를 입력합니다. 기본값은 180초입니다. 이 속성은 필수입니다.
Retries(재시도 횟수**)** 필드	요청에 실패한 것으로 간주하기 전에 연결을 재시도할 횟수입니다.

단계 4 **Save(**저장**)**를 클릭합니다.

다음에 수행할 작업

RADIUS 제공자를 생성합니다.

RADIUS 제공자 생성

RADIUS 제공자, 즉 이 어플라이언스에 대해 RADIUS 기반 AAA 서비스를 제공하는 특정 원격 서버를 정의하고 구성하려면 다음 단계를 수행하십시오.

참고 FXOS에서는 최대 16개의 RADIUS 제공자를 지원합니다.

프로시저

단계 **1** **Platform Settings(**플랫폼 설정**)** > **AAA**를 선택합니다.

단계 **2** **RADIUS** 탭을 클릭합니다.

단계 **3** 추가할 각 RADIUS 제공자에 대해 다음을 수행합니다.

a) **RADIUS Providers(RADIUS** 제공자**)** 영역에서 **Add(**추가**)**를 클릭합니다.

b) **Add RADIUS Provider(RADIUS** 제공자 추가**)** 대화 상자에서 다음 필드를 작성합니다.

이름	설명
호스트 이름/**FQDN(**또는 **IP** 주소**)** 필드	RADIUS 서버의 호스트 이름 또는 IP 주소
Order(순서**)** 필드	FXOS에서 사용자를 인증하기 위해 이 제공자를 사용하는 순서입니다. Firepower Chassis Manager 또는 FXOS CLI에 정의되어 있는 다른 제공자에 기반하여 FXOS이 다음으로 사용 가능한 순서를 할당하게 하려면 1~16 사이의 정수를 입력하거나 **`lowest-available`** 또는 **0**(숫자 0)을 입력합니다.
Key(키**)** 필드	데이터베이스에 대한 SSL 암호화 키입니다. 공백, §(섹션 기호), ? (물음표) 또는 =(등호)를 제외한 모든 표준 ASCII 문자를 입력할 수 있습니다.
Confirm Key(키 확인**)** 필드	확인을 위해 다시 한 번 입력하는 SSL 암호화 키
Authorization Port(권한 부여 포트**)** 필드	Firepower Chassis Manager 또는 FXOS CLI에서 RADIUS 데이터베이스와 통신할 때 사용하는 포트입니다. 유효한 범위는 1~65535입니다. 표준 포트 번호는 1700입니다.
Timeout(시간 초과**)** 필드	시간이 초과되기 전에 시스템이 RADIUS 데이터베이스에 연결을 시도하는 데 필요한 시간(초) 길이 1~60초의 정수를 입력하거나 0(숫자 0)을 입력하여 **RADIUS** 탭에 지정된 전역 시간제한 값을 사용합니다. 기본값은 5일입니다.

이름	설명
Retries(재시도 횟수**)** 필드	요청에 실패한 것으로 간주하기 전에 연결을 재시도할 횟수입니다. 필요 시 0~5의 정수를 입력합니다. 값을 지정하지 않은 경우, Firepower Chassis Manager에서는 **RADIUS** 탭에 지정된 값을 사용합니다.

c) **OK(**확인**)**를 클릭하여 **Add RADIUS Provider(RADIUS** 제공자 추가**)** 대화 상자를 닫습니다.

단계 **4** **Save(**저장**)**를 클릭합니다.

RADIUS 제공자 삭제

프로시저

단계 **1** **Platform Settings(**플랫폼 설정**)** > **AAA**를 선택합니다.

단계 **2** **RADIUS** 탭을 클릭합니다.

단계 **3** **RADIUS Providers(RADIUS** 제공자**)** 영역에서 삭제할 RADIUS 제공자에 해당하는 테이블의 행에 있는 **Delete(**삭제**)** 아이콘을 클릭합니다.

TACACS+ 제공자 구성

TACACS+ 제공자 속성 구성

이 작업에서 구성하는 속성은 이 유형의 모든 제공자 연결에 대한 기본 설정입니다. 개별 제공자에 이러한 속성 중 하나에 대한 설정이 포함된 경우, FXOS에서는 해당 설정을 사용하고 기본 설정을 무시합니다.

참고 FXOS 섀시는 TACACS+(Terminal Access Controller Access-Control System Plus) 프로토콜에 대한 명령 어카운팅을 지원하지 않습니다.

프로시저

단계 **1** **Platform Settings(**플랫폼 설정**)** > **AAA**를 선택합니다.

단계 **2** **TACACS** 탭을 클릭합니다.

단계 **3** **Properties(**속성**)** 영역에서 다음 필드를 완성합니다.

이름	설명
Timeout(시간 초과**)** 필드	시간이 초과되기 전에 시스템이 TACACS+ 데이터베이스에 연결을 시도하는 데 필요한 시간(초). 1~60의 정수를 입력합니다. 기본값은 180초입니다. 이 속성은 필수입니다.

단계 **4** **Save(**저장**)**를 클릭합니다.

다음에 수행할 작업

TACACS+ 제공자를 만듭니다.

TACACS+ 제공자 생성

다음 단계를 따라 TACACS+ 제공자, 즉 이 어플라이언스에 대해 TACACS 기반 AAA 서비스를 제공하는 특정 원격 서버를 정의하고 구성합니다.

참고 FXOS에서는 최대 16개의 TACACS+ 제공자를 지원합니다.

프로시저

단계 **1** **Platform Settings(**플랫폼 설정**)** > **AAA**를 선택합니다.

단계 **2** **TACACS** 탭을 클릭합니다.

단계 **3** 추가할 각 TACACS+ 제공자에 대해 다음을 수행합니다.

a) **TACACS Providers(TACACS** 제공자**)** 영역에서 **Add(**추가**)**를 클릭합니다.

b) **Add TACACS Provider(TACACS** 제공자 추가**)** 대화 상자에서 다음 필드를 작성합니다.

이름	설명
호스트 이름**/FQDN(**또는 **IP** 주소**)** 필드	TACACS+ 서버의 호스트 이름 또는 IP 주소
Order(순서**)** 필드	FXOS에서 사용자를 인증하기 위해 이 제공자를 사용하는 순서입니다. Firepower Chassis Manager 또는 FXOS CLI에 정의되어 있는 다른 제공자에 기반하여 FXOS이 다음으로 사용 가능한 순서를 할당하게 하려면 1~16 사이의 정수를 입력하거나 **`lowest-available`** 또는 **0**(숫자 0)을 입력합니다.

이름	설명
Key(키**)** 필드	데이터베이스에 대한 SSL 암호화 키입니다. 공백, §(섹션 기호), ? (물음표) 또는 =(등호)를 제외한 모든 표준 ASCII 문자를 입력할 수 있습니다.
Confirm Key(키 확인**)** 필드	확인을 위해 다시 한 번 입력하는 SSL 암호화 키
Port(포트**)** 필드	Firepower Chassis Manager 또는 FXOS CLI이 TACACS+ 서버와 통신할 때 사용하는 포트. 1~65535의 정수를 입력합니다. 기본 포트는 49입니다.
Timeout(시간 초과**)** 필드	시간이 초과되기 전에 시스템이 TACACS+ 데이터베이스에 연결을 시도하는 데 필요한 시간(초). 1~60초의 정수를 입력하거나 0(숫자 0)을 입력하여 **TACACS+** 탭에 지정된 전역 시간제한 값을 사용합니다. 기본값은 5일입니다.

c) **OK(**확인**)**를 클릭하여 **Add TACACS Provider(TACACS** 제공자 추가**)** 대화 상자를 닫습니다.

단계 **4** **Save(**저장**)**를 클릭합니다.

TACACS+ 제공자 삭제

프로시저

단계 **1** **Platform Settings(**플랫폼 설정**)** > **AAA**를 선택합니다.

단계 **2** **TACACS** 탭을 클릭합니다.

단계 **3** **TACACS Providers(TACACS** 제공자**)** 영역에서 삭제할 TACACS+ 제공자에 해당하는 테이블의 행에 있는 **Delete(**삭제**)** 아이콘을 클릭합니다.

Syslog 구성

시스템 로깅은 디바이스의 메시지를 syslog 데몬을 실행 중인 서버로 수집하는 방식입니다. 중앙 syslog 서버에 로깅하면 로그와 경고를 종합하는 데 도움이 됩니다. syslog 서비스는 메시지를 수신하고 파일로 저장하거나 간단한 구성 파일에 따라 인쇄합니다. 이 로깅 양식을 통해 로그를 안전하게 장기 보관할 수 있습니다. 로그는 일상적인 문제 해결과 인시던트 처리에 모두 유용합니다.

프로시저

단계 **1** **Platform Settings(**플랫폼 설정**)** > **Syslog**를 선택합니다.

단계 **2** 로컬 대상을 구성합니다.

a) **Local Destinations(**로컬 대상**)** 탭을 클릭합니다.

b) **Local Destinations(**로컬 대상**)** 탭에서 다음 필드를 입력합니다.

이름	설명
Console(콘솔**)** 섹션	
Admin State(관리 상태**)** 필드	섀시가 콘솔에 syslog 메시지를 표시하는지 여부. 콘솔에 syslog 메시지를 표시하고 로그에 추가하려는 경우 **Enable(**활성화**)** 확인란을 선택합니다. **Enable(**활성화**)** 확인란이 선택되지 않은 경우, syslog 메시지는 로그에 추가되지만 콘솔에 표시되지 않습니다.
Level(레벨**)** 필드	**Console - Admin State(**콘솔 - 관리 상태**)**의 **Enable(**활성화**)** 확인란을 선택한 경우, 콘솔에 표시할 가장 낮은 메시지 수준을 선택합니다. 섀시가 콘솔에 해당 레벨 이상의 메시지를 표시합니다. 다음 중 하나일 수 있습니다. • **Emergencies** • **Alerts** • **Critical**
Monitor(모니터**)** 섹션	
Admin State(관리 상태**)** 필드	섀시가 모니터에 syslog 메시지를 표시하는지 여부. 모니터에 syslog 메시지를 표시하고 로그에 추가하려는 경우 **Enable(**활성화**)** 확인란을 선택합니다. **Enable(**활성화**)** 확인란이 선택되지 않은 경우, syslog 메시지는 로그에 추가되지만 모니터에 표시되지 않습니다.

이름	설명
Level(수준**)** 드롭다운 목록	**Monitor - Admin State(**모니터 - 관리 상태**)**의 **Enable(**활성화**)** 확인란을 선택한 경우, 모니터에 표시할 가장 낮은 메시지 수준을 선택합니다. Firepower 섀시는 모니터에 해당 수준 이상의 메시지를 표시합니다. 다음 중 하나일 수 있습니다. • **Emergencies** • **Alerts** • **Critical** • **Errors** • **Warnings** • **Notifications** • **Information(**정보**)** • **Debugging**

c) **Save(**저장**)**를 클릭합니다.

단계 **3** 원격 대상을 구성합니다.

a) **Remote Destination(**원격 대상**)** 탭을 클릭합니다.

b) **Remote Destination(**원격 대상**)** 탭에서, 섀시에서 생성된 메시지를 저장할 수 있는 최대 3개의 외부 로그에 대해 다음 필드를 입력합니다.

원격 대상에 syslog 메시지를 전송하여 외부 syslog 서버의 사용 가능한 디스크 공간에 따라 메시지를 보관할 수 있으며, 저장한 후에 로그 데이터를 조작할 수 있습니다. 예를 들어 특정 유형의 syslog 메시지가 기록될 때 실행할 작업을 지정하고, 로그에서 데이터를 추출하고 보고를 위해 기록을 다른 파일에 저장하거나, 사이트별 스크립트를 사용하여 통계를 추적할 수 있습니다.

이름	설명
Admin State(관리 상태**)** 필드	원격 로그 파일에 syslog 메시지를 저장하려는 경우 **Enable(**활성화**)** 확인란을 선택합니다.

이름	설명
Level(수준**)** 드롭다운 목록	시스템에서 저장할 가장 낮은 메시지 수준을 선택합니다. 시스템은 원격 파일에 해당 수준 이상의 메시지를 저장합니다. 다음 중 하나일 수 있습니다. • **Emergencies** • **Alerts** • **Critical** • **Errors** • **Warnings** • **Notifications** • **Information(**정보**)** • **Debugging**
Hostname/IP Address(호스트 이름**/IP** 주소**)** 필드	원격 로그 파일이 있는 호스트 이름 또는 IP 주소입니다. 참고 IP 주소 대신 호스트 이름을 사용하는 경우 DNS 서버를 구성해야 합니다.
Facility(기능**)** 드롭다운 목록	파일 메시지의 기반으로 사용할 syslog 서버에 대한 시스템 로그를 선택합니다. 다음 중 하나일 수 있습니다. • **Local0** • **Local1** • **Local2** • **Local3** • **Local4** • **Local5** • **Local6** • **Local7**

c) **Save(**저장**)**를 클릭합니다.

단계 **4** 로컬 소스를 구성합니다.

a) **Local Sources(**로컬 소스**)** 탭을 클릭합니다.

b) **Local Sources(**로컬 소스**)** 탭에서 다음 필드를 입력합니다.

이름	설명
Faults Admin State(결함 관리 상태**)** 필드	시스템 결함 로깅의 활성화 여부. **Enable(**활성화**)** 체크 박스를 선택한 경우, 섀시 로그가 모든 시스템 오류를 로깅합니다.
Audits Admin State(감사 관리 상태**)** 필드	감사 로깅의 활성화 여부. **Enable(**활성화**)** 체크 박스를 선택한 경우 섀시가 모든 감사 로그 이벤트를 로깅합니다.
Events Admin State(이벤트 관리 상태**)** 필드	시스템 이벤트 로깅의 활성화 여부. **Enable(**활성화**)** 체크 박스를 선택한 경우, 섀시가 모든 시스템 이벤트를 로깅합니다.

c) **Save(**저장**)**를 클릭합니다.

DNS 서버 구성

시스템에서 호스트의 IP 주소를 확인해야 하는 경우 DNS 서버를 지정해야 합니다. 예를 들어 DNS 서버를 구성하지 않으면 섀시에서 설정을 구성할 때 www.cisco.com 과 같은 이름을 사용할 수 없습니다. IPv4 또는 IPv6 주소 중 하나로 서버의 IP 주소를 사용해야 합니다. 최대 4개까지 DNS 서버를 구성할 수 있습니다.

참고 여러 DNS 서버를 구성할 경우 임의의 순서로만 서버를 검색합니다. 로컬 관리 명령에 DNS 서버 조회가 필요한 경우, 임의 순서로 DNS 서버 3개만 검색할 수 있습니다.

프로시저

단계 **1** **Platform Settings(**플랫폼 설정**)** > **DNS**를 선택합니다.

단계 **2** **Enable DNS Server(DNS** 서버 활성화**)** 체크 박스를 선택합니다.

단계 **3** 추가하려는 각 DNS 서버에 대해 최대 4개까지 **DNS Server(DNS** 서버**)** 필드에 DNS 서버의 IP 주소를 입력하고 **Add(**추가**)**를 클릭합니다.

단계 **4** **Save(**저장**)**를 클릭합니다.

FIPS 모드 활성화

Firepower 4100/9300 섀시에서 FIPS 모드를 활성화하려면 다음 단계를 수행하십시오.

프로시저

단계 **1** 관리자 사용자로 Firepower 4100/9300 섀시에 로그인합니다.

단계 **2** **Platform Settings**를 선택하여 Platform Settings(플랫폼 설정) 창을 엽니다.

단계 **3** **FIPS/CC mode**를 선택하여 FIPS and Common Criteria(FIPS 및 Common Criteria) 창을 엽니다.

단계 **4** FIPS에 대한 **Enable** 체크 박스를 선택합니다.

단계 **5** **Save** 를 클릭하여 구성을 저장합니다.

단계 **6** 프롬프트에 따라 시스템을 리부팅합니다.

FIPS 모드가 활성화되면, 허용되는 키 크기 및 알고리즘이 제한됩니다. MIO는 암호화 요구에 CiscoSSL 및 FOM(FIPS Object Module)을 사용합니다. 이를 통해 ASA의 독점 암호화 라이브러리 구현 및 HW 가속에 비해 FIPS 검증이 더 쉬워집니다.

다음에 수행할 작업

FXOS 릴리스 2.0.1 이전에는, 디바이스의 최초 설정 중 생성된 SSH 호스트 키가 1024비트로 하드 코딩되었습니다. FIPS 및 Common Criteria 인증 요구 사항을 충족하려면 이러한 과거의 호스트 키를 삭제하고 SSH 호스트 키 생성에 설명된 절차를 사용하여 새 호스트 키를 생성해야 합니다. 이 추가 단계를 수행하지 않으면, FIPS 모드가 활성화되어 디바이스가 리부팅된 후 SSH를 사용하여 Supervisor에 연결할 수 없습니다. FXOS 2.0.1 이상을 사용하여 초기 설정을 수행한 경우 새 호스트 키를 생성할 필요가 없습니다.

Common Criteria 모드 활성화

Firepower 4100/9300 섀시에서 Common Criteria 모드를 활성화하려면 다음 단계를 수행하십시오.

프로시저

단계 **1** 관리자 사용자로 Firepower 4100/9300 섀시에 로그인합니다.

단계 **2** **Platform Settings**를 선택하여 Platform Settings(플랫폼 설정) 창을 엽니다.

단계 **3** **FIPS/CC mode**를 선택하여 FIPS and Common Criteria(FIPS 및 Common Criteria) 창을 엽니다.

단계 **4** Common Criteria에 대한 **Enable** 확인란을 선택합니다.

단계 **5** **Save** 를 클릭하여 구성을 저장합니다.

단계 **6** 프롬프트에 따라 시스템을 리부팅합니다.

Common Criteria는 컴퓨터 보안에 대한 국제 표준입니다. CC는 인증서, 감사, 로깅, 비밀번호, TLS, SSH 등에 중점을 둡니다. 기본적으로 FIPS 규정 준수를 가정합니다. FIPS와 마찬가지로 Cisco는 NIST 공인 랩 벤더와 계약을 체결하여 테스트를 수행하고 NIAP에 제출합니다.

CC 모드가 활성화되면 지원해야 하는 알고리즘, 암호 그룹 및 기능의 목록이 제한됩니다. MIO는 NDcPP(Network Device Collaborative Protection Profile)를 기준으로 평가됩니다. CiscoSSL은 대부분의 CC 규정 준수 가이드에서 다루는 요구사항의 일부만 시행할 수 있습니다.

다음에 수행할 작업

FXOS 릴리스 2.0.1 이전에는, 디바이스의 최초 설정 중 생성된 SSH 호스트 키가 1024비트로 하드 코딩되었습니다. FIPS 및 Common Criteria 인증 요구사항을 충족하려면 이러한 과거의 호스트 키를 삭제하고 SSH 호스트 키 생성에 설명된 절차를 사용하여 새 호스트 키를 생성해야 합니다. 이 추가 단계를 수행하지 않으면, Common Criteria 모드가 활성화되어 디바이스가 리부팅된 후 SSH를 사용하여 Supervisor에 연결할 수 없습니다. FXOS 2.0.1 이상을 사용하여 초기 설정을 수행한 경우 새 호스트 키를 생성할 필요가 없습니다.

IP 액세스 목록 구성

기본적으로 Firepower 4100/9300 섀시는 로컬 웹 서버에 대한 모든 액세스를 거부합니다. 각 IP 블록에 대해 허용된 서비스 목록으로 IP 액세스 목록을 구성해야 합니다.

IP 액세스 목록은 다음 프로토콜을 지원합니다.

- HTTPS
- SNMP
- SSH

IP 주소(v4 또는 v6) 각 블록에서 각 디바이스에 대해 최대 100개의 서로 다른 서브넷을 구성할 수 있습니다. 서브넷 0과 접두사 0은 서비스에 대한 무제한 액세스를 허용합니다.

프로시저

단계 **1** 관리자 사용자로 Firepower 4100/9300 섀시에 로그인합니다.

단계 **2** **Platform Settings**를 선택하여 Platform Settings(플랫폼 설정) 페이지를 엽니다.

단계 **3** **Access List**를 선택하여 Access List(액세스 목록) 영역을 엽니다.

단계 **4** 이 영역에서 IP 액세스 목록에 나열된 IPv4 및 IPv6 주소를 보고 추가하고 삭제할 수 있습니다.

IPv4 블록을 추가하려면 유효한 IPv4 IP 주소 및 [0-32] 길이의 접두사를 입력하고 프로토콜을 선택해야 합니다.

IPv6 블록을 추가하려면 유효한 IPv6 IP 주소 및 [0-128] 길이의 접두사를 입력하고 프로토콜을 선택해야 합니다.

컨테이너 인스턴스 인터페이스에 대해 **MAC** 풀 접두사 추가 및 **MAC** 주소 확인

FXOS 섀시는 컨테이너 인스턴스 인터페이스용 MAC 주소를 자동으로 생성하며 각 인스턴스의 공유 인터페이스가 고유한 MAC 주소를 사용하도록 보장합니다. FXOS 섀시는 다음 형식을 사용하여 MAC 주소를 생성합니다.

A2*xx.yyzz.zzzz*

여기서 *xx.yy*는 사용자 정의 접두사 또는 시스템 정의 접두사이고 *zz.zzzz*는 섀시에서 생성하는 내부 카운터입니다. 시스템 정의 접두사는 IDPROM에 프로그래밍되는 번인된 MAC 주소 풀의 첫 번째 MAC 주소의 하위 2바이트와 일치합니다. MAC 주소 풀을 확인하려면 **connect fxos**, **show module**을 차례로 사용합니다. 예를 들어 모듈 1에 대해 표시되는 MAC 주소 범위가 b0aa.772f.f0b0~b0aa.772f.f0bf이면 시스템 접두사는 f0b0입니다.

자세한 내용은 컨테이너 인스턴스 인터페이스용 자동 MAC 주소, 209 페이지를 참조하십시오.

이 절차에서는 MAC 주소를 확인하고 필요에 따라 생성에 사용되는 접두사를 정의하는 방법을 설명합니다.

참고 논리적 디바이스를 구축한 후에 MAC 주소 접두사를 변경하는 경우 트래픽 중단이 발생할 수 있습니다.

프로시저

단계 **1** **Platform Settings(**플랫폼 설정**)** > **MAC Pool(MAC** 풀**)**을 선택합니다.

이 페이지에는 생성된 MAC 주소와 해당 MAC 주소를 사용하는 컨테이너 인스턴스 및 인터페이스가 표시됩니다.

단계 **2** (선택 사항) MAC 주소 생성에 사용되는 MAC 주소 접두사를 추가합니다.

a) **Add Prefix(**접두사 추가**)**를 클릭합니다.

Set the Prefix for the MAC Pool(MAC 풀에 대한 접두사 설정**)** 대화 상자가 나타납니다.

a) 1~65535 사이의 10진수 값을 입력합니다. 이 접두사가 4자리 16진수로 변환되어 MAC 주소의 일부로 사용됩니다.

접두사가 어떻게 사용되는지 예를 들어 설명하자면, 접두사를 77로 설정하는 경우 섀시에서는 77을 16진수 값 004D(*yyxx*)로 변환합니다. 접두사가 MAC 주소에서 쓰일 때는 섀시 기본 형식에 부합하도록 역전됩니다(*xxyy*).

A2**4D.00***zz.zzzz*

접두사가 1009 (03F1)일 때 MAC 주소는 다음과 같습니다.

A2**F1.03***zz.zzzz*

b) **OK(**확인**)**를 클릭합니다.

이 접두사를 사용하는 새 MAC 주소가 생성되어 할당됩니다. 현재 접두사와 생성된 16진수 값이 테이블 위에 표시됩니다.

컨테이너 인스턴스에 대한 리소스 프로파일 추가

컨테이너 인스턴스당 리소스 사용량을 지정하려면 리소스 프로필을 하나 이상 생성합니다. 논리적 디바이스/애플리케이션 인스턴스를 구축할 때 사용할 리소스 프로필을 지정합니다. 리소스 프로파일은 CPU 코어 수를 설정합니다. RAM은 코어 수에 따라 동적으로 할당되며 디스크 공간은 인스턴스당 40GB로 설정됩니다.

- 최소 코어 수는 6입니다.

참고 코어 수가 적은 인스턴스는 코어 수가 더 많은 CPU 사용률보다 CPU 사용률이 상대적으로 높아질 수 있습니다. 코어 수가 적은 인스턴스는 트래픽 로드 변경에 더욱 민감합니다. 트래픽 삭제를 경험하는 경우 더 많은 코어를 할당해 보십시오.

- 코어는 최대값까지 짝수(6, 8, 10, 12, 14 등)로 할당할 수 있습니다.
- 사용 가능한 코어의 최대 수는 보안 모듈/섀시 모델에 따라 달라집니다. 컨테이너 인스턴스의 요구 사항 및 사전 요구 사항, 218 페이지 섹션을 참조하십시오.

섀시에는 최소 코어 수가 포함된 "Default-Small"이라는 기본 리소스 프로필이 있습니다. 이 프로필의 정의를 변경할 수 있으며 해당 프로필을 사용하지 않으면 삭제할 수도 있습니다. 이 프로필은 섀시를 다시 로드할 때 생성되며, 시스템에 다른 프로필은 없습니다.

리소스 프로파일이 현재 사용 중이라면 해당 설정을 변경할 수 없습니다. 해당 프로파일을 사용하는 인스턴스를 비활성화하고 리소스 프로파일을 변경한 후에 마지막으로 인스턴스를 다시 활성화해야 합니다. 설정된 고가용성 쌍 또는 클러스터에서 인스턴스 크기를 조정하는 경우에는 최대한 빠른 시간 내에 모든 멤버를 같은 크기로 설정해야 합니다.

FTD 인스턴스를 FMC에 추가한 후 리소스 프로파일 설정을 변경하는 경우 FMC **Devices(**디바이스**)** > **Device Management(**디바이스 관리**)** > **Device(**디바이스**)** > **System(**시스템**)** > **Inventory(**재고 목록**)** 대화 상자에서 재고 목록을 업데이트합니다.

프로시저

단계 **1** **Platform Settings(**플랫폼 설정**)** > **Resource Profiles(**리소스 프로필**)**를 선택한 다음 **Add(**추가**)**를 클릭합니다.

Add Resource Profile(리소스 프로필 추가**)** 대화 상자가 나타납니다.

단계 **2** 다음 파라미터를 설정합니다.

- **Name(**이름**)** - 1~64자 사이의 프로필 이름을 설정합니다. 프로필을 추가한 후에는 이 프로필 이름을 변경할 수 없습니다.
- **Description(**설명**)** - 프로필에 대한 설명(최대 510자)을 설정합니다.
- **Number of Cores(**코어 수**)** - 섀시에 따라 프로필의 코어 수를 6~최대값 사이의 짝수로 설정합니다.

단계 **3** **OK(**확인**)**를 클릭합니다.

네트워크 제어 정책 구성

Cisco 이외 디바이스의 검색을 허용하기 위해 FXOS에서는 IEEE 802.1ab 표준에 정의된 벤더 중립적인 디바이스 검색 프로토콜인 *LLDP(Link Layer Discovery Protocol)*를 지원합니다. LLDP를 통해 네트워크 디바이스에서 네트워크의 다른 디바이스에 자신에 관한 정보를 광고할 수 있습니다. 이 프로토콜은 데이터 링크 레이어를 통해 실행되므로 서로 다른 네트워크 레이어 프로토콜을 실행하는 두 시스템에서 서로에 관한 정보를 얻을 수 있습니다.

LLDP는 디바이스와 해당 인터페이스의 기능 및 현재 상태에 관한 정보를 전송하는 단방향 프로토콜입니다. LLDP 디바이스는 다른 LLDP 디바이스로부터 정보를 얻을 때만 이 프로토콜을 사용합니다.

FXOS 섀시에서 이 기능을 활성화하기 위해 LLDP 전송 및 수신 동작을 지정하는 네트워크 제어 정책을 구성할 수 있습니다. 네트워크 제어 정책을 생성한 후에는 인터페이스에 할당해야 합니다. 고정 포트, EPM 포트, 포트 채널 및 breakout 포트를 비롯한 전면 인터페이스에서 LLDP를 활성화할 수 있습니다.

참고

- LLDP는 전용 관리 포트에서 구성할 수 없습니다.
- 블레이드에 연결되는 내부 백플레인 포트에서는 기본적으로 LLDP가 활성화되어 있으며, 비활성화를 위한 옵션은 없습니다. 다른 모든 포트에서는 기본적으로 LLDP가 비활성화되어 있습니다.

프로시저

단계 **1** **Platform Settings(**플랫폼 설정**)** > **Network Control Policy(**네트워크 제어 정책**)**를 선택합니다.

단계 **2** **Add(**추가**)**를 클릭합니다.

단계 **3** Network Control Policy(네트워크 제어 정책) 대화 상자에서 다음 필드를 수정합니다.

이름	설명
Name(이름**)** 필드	네트워크 제어 정책에 대한 고유한 이름입니다.
LLDP recieve(LLDP 수신**)** 확인란	LLDP 패킷을 수신하도록 FXOS를 활성화합니다.
LLDP transmit 확인란	LLDP 패킷을 전송하도록 FXOS를 활성화합니다.
설명 필드	네트워크 제어 정책에 대한 설명입니다.

단계 **4** **Save(**저장**)**를 클릭합니다. 네트워크 제어 정책을 생성한 후에는 인터페이스에 할당해야 합니다. 네트워크 제어 정책을 사용하여 인터페이스를 수정하고 구성하는 단계는 실제 인터페이스 구성, 184 페이지의 내용을 참조하십시오.

섀시 URL 구성

FMC에서 직접 FTD 인스턴스를 위해 Firepower Chassis Manager를 쉽게 열 수 있도록 관리 URL을 지정할 수 있습니다. 섀시 관리 URL을 지정하지 않으면 섀시 이름이 대신 사용됩니다.

FTD 인스턴스를 FMC에 추가한 후 섀시 URL 설정을 변경하는 경우 **Devices(**디바이스**)** > **Device Management(**디바이스 관리**)** > **Device(**디바이스**)** > **System(**시스템**)** > **Inventory(**인벤토리**)** 대화 상자에서 각 유닛의 인벤토리를 업데이트합니다.

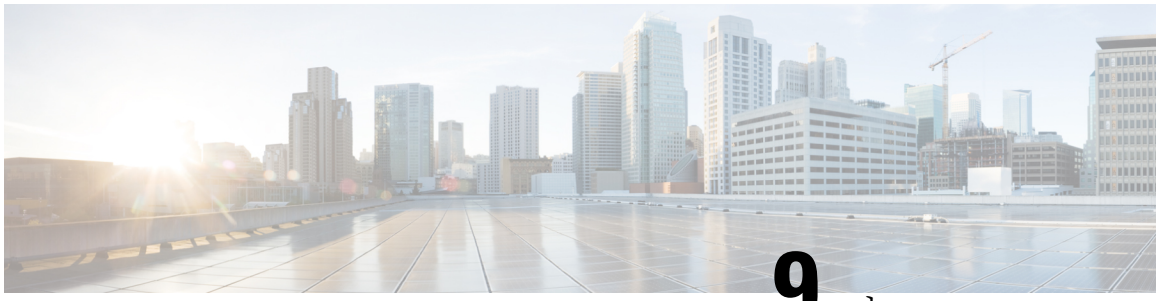
프로시저

단계 **1** **Platform Settings(**플랫폼 설정**)** > **Chassis URL(**섀시 **URL)**을 선택합니다.

단계 **2** 다음 파라미터를 설정합니다.

- **Chassis Name(**섀시 이름**)** - 1~60자 사이의 섀시 이름을 설정합니다.
- **Chassis URL(**섀시 **URL)** - FMC가 Firepower Chassis Manager 내에서 FTD 인스턴스에 연결할 때 사용해야 하는 URL을 설정합니다. URL은 https://로 시작해야 합니다. 섀시 관리 URL을 지정하지 않으면 섀시 이름이 대신 사용됩니다.

단계 **3** **Update(**업데이트**)**를 클릭합니다.



9 장

인터페이스 관리

인터페이스 정보

Firepower 4100/9300 섀시에서는 물리적 인터페이스, 컨테이너 인스턴스용 VLAN 하위 인터페이스 및 EtherChannel(포트-채널) 인터페이스를 지원합니다. EtherChannel 인터페이스는 동일한 유형의 멤버 인터페이스를 최대 16개까지 포함할 수 있습니다.

섀시 관리 인터페이스

섀시 관리 인터페이스는 SSH 또는 Firepower Chassis Manager를 통한 FXOS 섀시 관리에 사용됩니다. 이 인터페이스는 **Interfaces(**인터페이스**)** 탭의 상단에 **MGMT**로 표시되며 **Interfaces(**인터페이스**)** 탭에서 이 인터페이스를 활성화하거나 비활성화할 수만 있습니다. 이 인터페이스는 애플리케이션 관리용 논리적 디바이스에 할당하는 관리 유형 인터페이스와는 별개입니다.

이 인터페이스의 파라미터는 CLI에서 구성해야 합니다. 관리 IP 주소 변경, 90 페이지 섹션도 참조하십시오. FXOS CLI에서 이 인터페이스에 대한 정보를 확인하려면 로컬 관리에 연결한 다음 관리 포트를 표시합니다.

Firepower # **connect local-mgmt**

Firepower(local-mgmt) # **show mgmt-port**

실제 케이블이나 SFP 모듈 연결을 해제하거나 **mgmt-port shut** 명령을 수행하더라도 섀시 관리 인터페이스는 계속 작동합니다.

참고 섀시 관리 인터페이스는 점보 프레임을 지원하지 않습니다.

인터페이스 유형

물리적 인터페이스, 컨테이너 인스턴스용 VLAN 하위 인터페이스, EtherChannel(포트-채널) 인터페이스는 다음 유형 중 하나가 될 수 있습니다.

- Data(데이터) - 일반 데이터에 사용합니다. 데이터 인터페이스는 논리적 디바이스 간에 공유할 수 없으며 논리적 디바이스는 백플레인을 통해 다른 논리적 디바이스와 통신할 수 없습니다. 데이터 인터페이스의 트래픽의 경우, 모든 트래픽은 하나의 인터페이스에서 섀시를 종료하고 다른 인터페이스로 돌아가서 다른 논리적 디바이스에 연결해야 합니다.
- Data-sharing(데이터 공유) - 일반 데이터에 사용합니다. 컨테이너 인스턴스에서만 지원되는 이러한 데이터 인터페이스는 하나 이상의 논리적 디바이스/컨테이너 인스턴스(FTD-사용-FMC 전용)에서 공유할 수 있습니다. 각 컨테이너 인스턴스는 이 인터페이스를 공유하는 다른 모든 인스턴스와 백플레인을 통해 통신할 수 있습니다. 공유 인터페이스는 구축할 수 있는 컨테이너 인스턴스 수에 영향을 줄 수 있습니다. 브리지 그룹 멤버 인터페이스(투명 모드 또는 라우팅 모드), 인라인 집합, 패시브 인터페이스, 클러스터, 또는 페일오버 링크에 대해서는 공유 인터페이스가 지원되지 않습니다.
- Mgmt(관리) - 애플리케이션 인스턴스를 관리하는 데 사용합니다. 이러한 인터페이스는 외부 호스트에 액세스하기 위해 하나 이상의 논리적 디바이스에서 공유할 수 있습니다. 단, 논리적 디바이스에서는 인터페이스를 공유하는 다른 논리적 디바이스와 이 인터페이스를 통해 통신할 수 없습니다. 논리적 디바이스당 관리 인터페이스 1개만 할당할 수 있습니다. 애플리케이션 및 관리자에 따라 나중에 데이터 인터페이스에서 관리를 활성화할 수 있습니다. 데이터 관리를 활성화한 후 이를 사용하지 않으려는 경우에도 관리 인터페이스를 논리적 디바이스에 할당해야 합니다.

참고 관리 인터페이스를 변경하면 논리적 디바이스가 재부팅됩니다. 예를 들어 e1/1에서 e1/2로 변경하면 논리적 디바이스가 재부팅되어 새 관리가 적용됩니다.

- 이벤트 처리—FTD-사용-FMC 디바이스의 보조 관리 인터페이스로 사용합니다.이 인터페이스를 사용하려면 FTDCLI에서 해당 IP 주소 및 기타 매개변수를 구성해야 합니다. 예를 들면 관리 트래픽을 이벤트(예: 웹 이벤트)에서 분리할 수 있습니다. 자세한 내용은 Management Center 컨피그레이션 가이드를 참조하세요. 하나 이상의 논리적 디바이스가 외부 호스트에 액세스하기 위해 이벤트 인터페이스를 공유할 수 있습니다. 논리적 디바이스가 인터페이스를 공유하는 다른 논리적 디바이스와 이 인터페이스를 통해 통신할 수는 없습니다. 나중에 관리를 위해 데이터 인터페이스를 구성하는 경우 별도의 이벤트 인터페이스를 사용할 수 없습니다.

참고 각 애플리케이션 인스턴스가 설치될 때 가상 이더넷 인터페이스가 할당됩니다. 애플리케이션에서 이벤트 인터페이스를 사용하지 않는 경우 가상 인터페이스는 관리자 중단 상태가 됩니다.

```
Firepower # show interface Vethernet775
Firepower # Vethernet775 is down (Administratively down)
Bound Interface is Ethernet1/10
Port description is server 1/1, VNIC ext-mgmt-nic5
```

- Cluster(클러스터) - 클러스터형 논리적 디바이스용 클러스터 제어 링크로 사용합니다. 기본적으로, 클러스터 제어 링크는 Port-channel 48에서 자동으로 생성됩니다. 이 클러스터 유형은 EtherChannel 인터페이스에서만 지원됩니다. 다중 인스턴스 클러스터링의 경우 디바이스 간에 클러스터 유형 인터페이스를 공유할 수 없습니다. VLAN 하위 인터페이스를 클러스터 EtherChannel에 추가하여 클러스터당 별도의 클러스터 제어 링크를 제공할 수 있습니다. 클러스터 인터페이스에 하위 인터페이스를 추가하면 네이티브 클러스터에서 해당 인터페이스를 사용할 수 없습니다. FDM 및 CDO는 클러스터링을 지원하지 않습니다.

참고 이 장에서는 *FXOS* VLAN 하위 인터페이스에 대해서만 설명합니다. FTD 애플리케이션 내에 하위 인터페이스를 추가할 수 있습니다. 자세한 내용은 FXOS 인터페이스와 애플리케이션 인터페이스 비교, 166 페이지를 참조하십시오.

독립형 및 클러스터 구축에서 FTD 및 ASA 애플리케이션에 대한 인터페이스 유형 지원은 다음 표를 참조하십시오.

표 *11:* 인터페이스 유형 지원

애플리케이션		데이터	데이터: 하위 인터페이스	데이터 공유	데이터 공유: 하위 인터페이스	관리	이벤트	클러스터 (EtherChannel에만 해당)	클러스터: 하위 인터페이스
FTD	독립형 네이티브 인스턴스	예	—	—	—	예	예	—	—
	독립형 컨테이너 인스턴스	예	예	예	예	예	예	—	—
	클러스터 기본 인스턴스	예 (섀시 간 클러스터 전용 EtherChannel)	—	—	—	예	예	예	—
	클러스터 컨테이너 인스턴스	예 (섀시 간 클러스터 전용 EtherChannel)	—	—	—	예	예	예	예
ASA	독립형 네이티브 인스턴스	예	—	—	—	예	—	예	—
	클러스터 기본 인스턴스	예 (섀시 간 클러스터 전용 EtherChannel)	—	—	—	예	—	예	—

FXOS 인터페이스와 애플리케이션 인터페이스 비교

Firepower 4100/9300에서는 물리적 인터페이스, 컨테이너 인스턴스용 VLAN 하위 인터페이스 및 EtherChannel(포트-채널) 인터페이스의 기본 이더넷 설정을 관리합니다. 애플리케이션 내에서는 상위 레벨 설정을 구성합니다. 예를 들어 FXOS에서는 Etherchannel만 생성할 수 있습니다. 그러나 애플리케이션 내의 EtherChannel에 IP 주소를 할당할 수 있습니다.

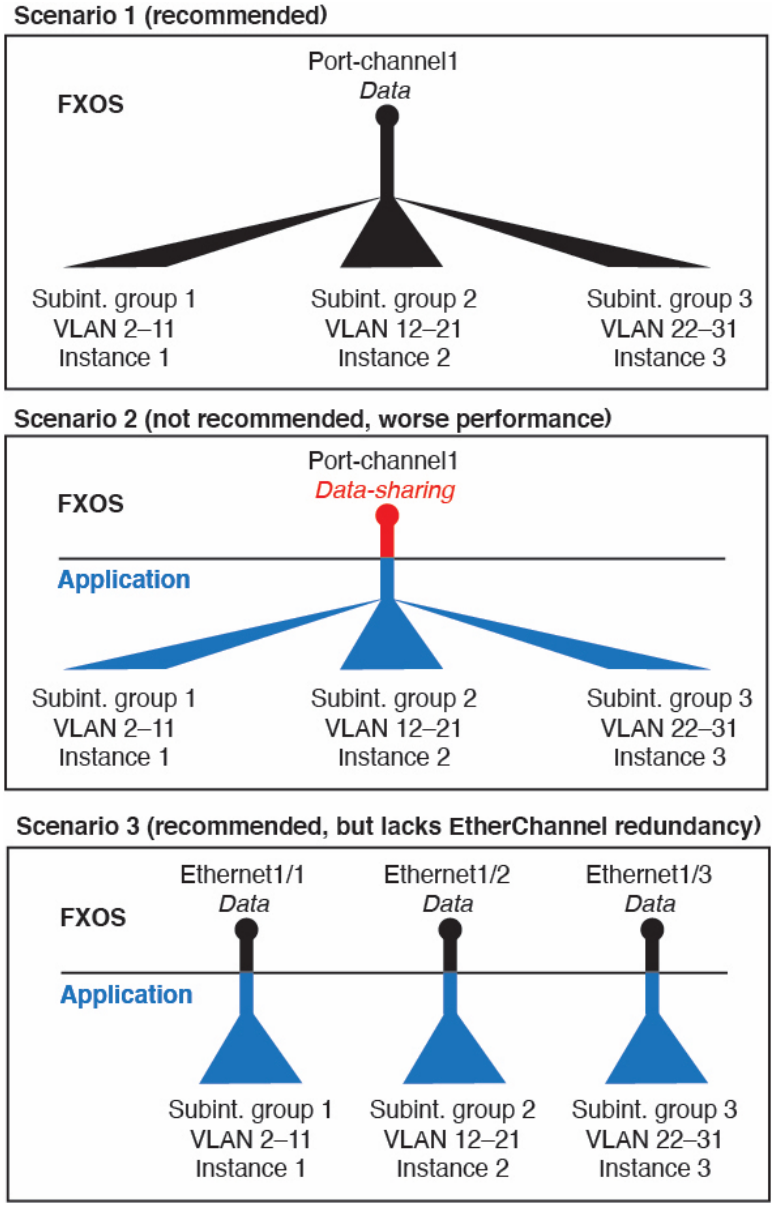
다음 섹션에서는 FXOS와 인터페이스에 대한 애플리케이션 간의 상호 작용에 대해 설명합니다.

VLAN 하위 인터페이스

논리적 디바이스의 경우에는 애플리케이션 내에서 VLAN 하위 인터페이스를 생성할 수 있습니다.

독립형 모드만의 컨테이너 인스턴스의 경우에는 또한 FXOS(FXOS 하위 인터페이스가 없는 인터페이스)에서 VLAN 하위 인터페이스를 생성할 수도 있습니다. 다중 인스턴스 클러스터는 클러스터 유형 인터페이스를 제외하고는 FXOS에서 하위 인터페이스를 지원하지 않습니다. 애플리케이션 정의 하위 인터페이스는 FXOS 제한에 영향을 받지 않습니다. 네트워크 구축 및 개인 기본 설정에 따라 하위 인터페이스를 생성할 운영 체제를 선택합니다. 예를 들어 하위 인터페이스를 공유하려면 FXOS에서 하위 인터페이스를 생성해야 합니다. FXOS 하위 인터페이스를 이용하는 또 다른 시나리오는 단일 인터페이스에서 하위 인터페이스 그룹을 여러 인스턴스로 할당하는 것입니다. 인스턴스 A에는 VLAN 2~11이, 인스턴스 B에는 VLAN 12~21, 인스턴스 C에는 VLAN 22~31이 있는 Port-channel1을 사용하려는 경우를 예로 들어 보겠습니다. 애플리케이션 내에서 이러한 하위 인터페이스를 생성하는 경우에는 FXOS에서 상위 인터페이스를 공유해야 하는데, 이러한 방식은 효율적이지 않을 수 있습니다. 다음 그림에서 이 시나리오를 수행할 수 있는 세 가지 방법을 참조하십시오.

그림 *1: FXOS*의 *VLAN* 및 컨테이너 인스턴스의 애플리케이션의 비교



섀시와 애플리케이션의 독립 인터페이스 상태

관리를 위해 섀시와 애플리케이션에서 인터페이스를 활성화하고 비활성화할 수 있습니다. 인터페이스는 두 운영 체제에서 모두 활성화해야 작동합니다. 인터페이스 상태는 독립적으로 제어되므로 섀시와 애플리케이션에서 상태가 일치하지 않을 수도 있습니다.

애플리케이션 내의 인터페이스 기본 상태는 인터페이스 유형에 따라 달라집니다. 예를 들어 물리적 인터페이스 또는 EtherChannel은 애플리케이션 내에서 기본적으로 비활성화되지만 하위 인터페이스는 기본적으로 활성화됩니다.

하드웨어 바이패스 쌍

FTD의 경우 Firepower 9300 및 4100 Series의 특정 인터페이스 모듈에서 하드웨어 바이패스 기능을 활성화할 수 있습니다. 하드웨어 바이패스는 정전 중에도 트래픽이 인라인 인터페이스 쌍 사이에서 계속 흐르도록 합니다. 이 기능은 소프트웨어 또는 하드웨어 오류의 경우 네트워크 연결성을 유지 관리하는 데 사용될 수 있습니다.

하드웨어 바이패스 기능은 FTD 애플리케이션 내에서 구성됩니다. 이러한 인터페이스를 하드웨어 바이패스 쌍으로 사용할 필요가 없습니다. 이들은 ASA 및 FTD 애플리케이션에서 모두 일반 인터페이스로 사용할 수 있습니다. Breakout 포트에 대해 하드웨어 바이패스 지원 인터페이스를 구성할 수 없습니다. 하드웨어 바이패스 기능을 사용하려면 포트를 EtherChannel로 구성하지 마십시오. 그렇게 하지 않으면 이러한 인터페이스를 일반 인터페이스 모드에서 EtherChannel 멤버로 포함할 수 있습니다.

인라인 쌍에서 하드웨어 바이패스가 활성화된 경우 스위치 우회가 먼저 시도됩니다. 스위치 오류로 인해 우회 구성이 실패하면 물리적 우회가 활성화됩니다.

참고 하드웨어 우회(FTW)는 VDP/Radware와 같은 서드파티 애플리케이션을 사용하는 FTD에 설치된 에서 지원되지 않습니다.

FTD는 다음 모델에서 특정 네트워크 모듈의 인터페이스 쌍에 대해 하드웨어 바이패스를 지원합니다.

- Firepower 9300
- Firepower 4100 Series

이러한 모델에 대해 지원되는 하드웨어 바이패스 네트워크 모듈은 다음과 같습니다.

- Firepower 6 포트 1G SX FTW Network Module single-wide(FPR-NM-6X1SX-F)
- Firepower 6 포트 10G SR FTW Network Module single-wide(FPR-NM-6X10SR-F)
- Firepower 6 포트 10G LR FTW Network Module single-wide(FPR-NM-6X10LR-F)
- Firepower 2 포트 40G SR FTW Network Module single-wide(FPR-NM-2X40G-F)
- Firepower 8 포트 1G Copper FTW Network Module single-wide(FPR-NM-8X1G-F)

하드웨어 바이패스는 다음 포트 쌍만 사용할 수 있습니다.

- 1 및 2
- 3 및 4
- 5 및 6
- 7 및 8

Jumbo Frame Support

Firepower 4100/9300 섀시에서는 기본적으로 점보 프레임 지원이 활성화되어 있습니다. Firepower 4100/9300 섀시에 설치된 특정 논리적 디바이스에서 점보 프레임 지원을 활성화하려면 논리적 디바이스에서 인터페이스에 대한 적절한 MTU 설정을 구성해야 합니다.

Firepower 4100/9300 섀시의 애플리케이션에 대해 지원되는 최대 MTU는 9184입니다.

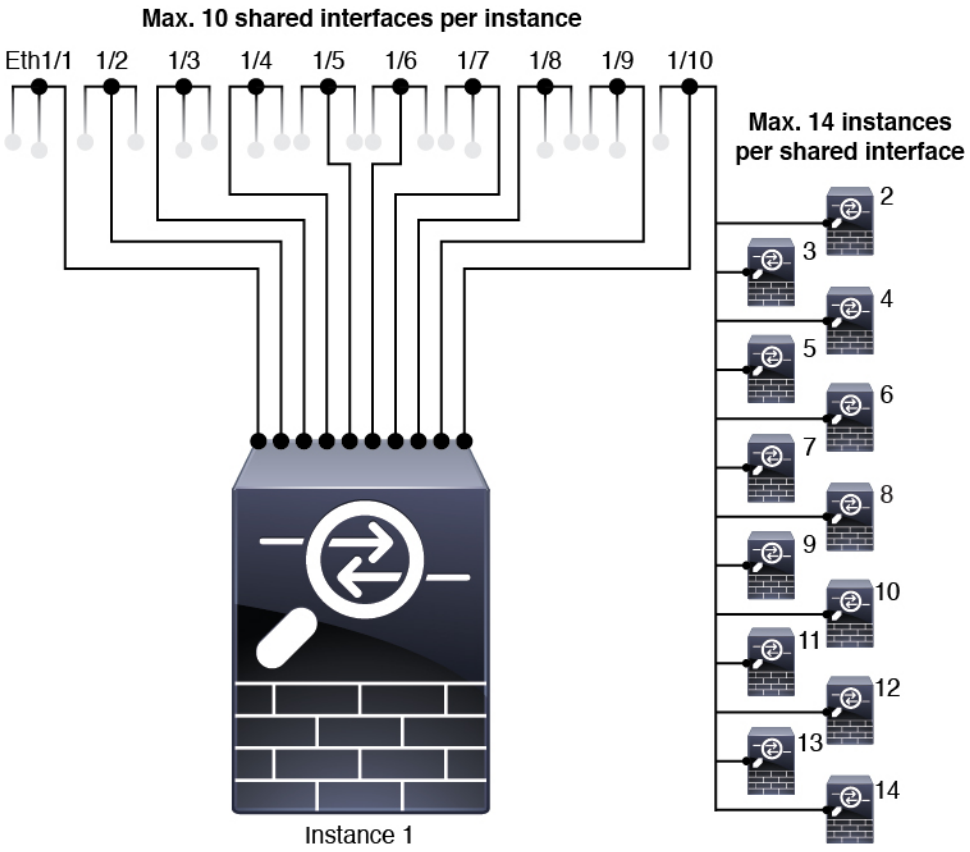
참고 섀시 관리 인터페이스는 점보 프레임을 지원하지 않습니다.

공유 인터페이스 확장성

컨테이너 인스턴스는 데이터 공유 유형 인터페이스를 공유할 수 있습니다. 이 기능을 통해 물리적 인터페이스 사용량을 절약하면서 유연한 네트워킹 구축도 지원할 수 있습니다. 인터페이스를 공유할 때 섀시는 고유한 MAC 주소를 사용하여 올바른 인스턴스로 트래픽을 포워딩합니다. 그러나 공유 인터페이스로 인해 섀시 내에 전체 메시 토폴로지가 필요해져서 포워딩 테이블이 커질 수 있습니다. 모든 인스턴스가 동일한 인터페이스를 공유하는 다른 모든 인스턴스와 통신할 수 있어야 하기 때문입니다. 따라서 공유할 수 있는 인터페이스 수에는 제한이 있습니다.

섀시는 포워딩 테이블 외에 VLAN 하위 인터페이스 포워딩용 VLAN 그룹 테이블도 유지합니다.최대 500개의 VLAN 하위 인터페이스를 생성할 수 있습니다.

공유 인터페이스 할당과 관련한 다음 제한을 참조하십시오.



공유 인터페이스 모범 사례

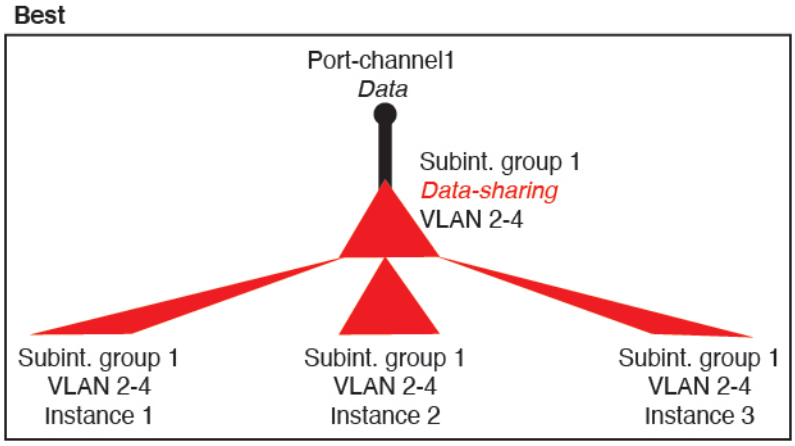
포워딩 테이블의 최적의 확장성을 위해 최대한 적은 수의 인터페이스를 공유합니다. 대신, 하나 이상의 물리적 인터페이스에서 최대 500개의 VLAN 하위 인터페이스를 생성하고 컨테이너 인스턴스 사이에 VLAN을 나눌 수 있습니다.

인터페이스 공유 시에는 다음 사례를 확장성이 높은 방식부터 차례로 따르십시오.

1. 최고 - 단일 상위 인터페이스에 속한 하위 인터페이스를 공유하고 동일한 인스턴스 그룹과 동일한 하위 인터페이스 집합을 사용합니다.

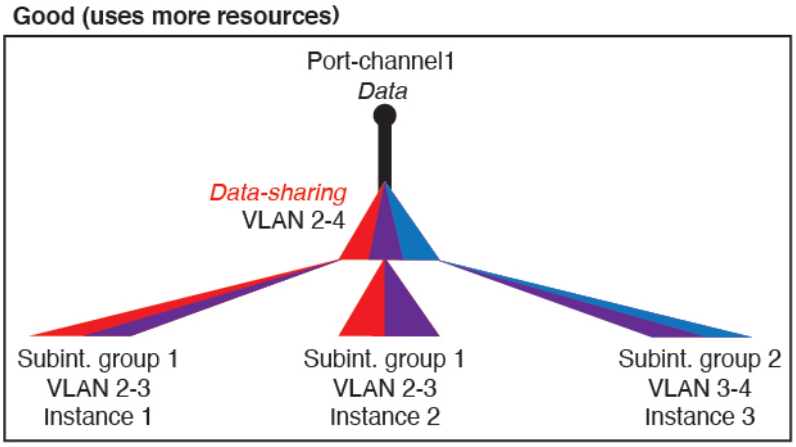
   예를 들어 대규모 EtherChannel 하나를 생성해 유사한 종류의 모든 인터페이스를 함께 번들로 묶은 다음 해당 EtherChannel의 하위 인터페이스를 공유합니다. 즉, Port-Channel2, Port-Channel3 및 Port-Channel4를 공유하는 대신 Port-Channel1.2, 3 및 4를 공유합니다. 단일 상위 인터페이스의 하위 인터페이스를 공유하면 상위 인터페이스 전체에서 하위 인터페이스를 공유하거나 물리적/EtherChannel 인터페이스를 공유할 때 VLAN 그룹 테이블이 전달 테이블보다 더 잘 확장됩니다.

그림 *2:* 최고 *:* 하나의 상위에 있는 공유 하위 인터페이스 그룹



인스턴스의 그룹과 동일한 하위 인터페이스 집합을 공유하지 않는 경우 구성으로 인해 더 많은 리소스 사용량(더 많은 VLAN 그룹)이 발생할 수 있습니다. Port-Channel1.3 및 4를 인스턴스 3(2개의 VLAN 그룹)과 공유하는 동안 Port-Channel1.2 및 3을 인스턴스 1 및 2와 공유하는 대신 Port-Channel1.2, 3 및 4를 인스턴스 1, 2 및 3(1개의 VLAN 그룹)과 공유하는 경우를 예로 들 수 있습니다.

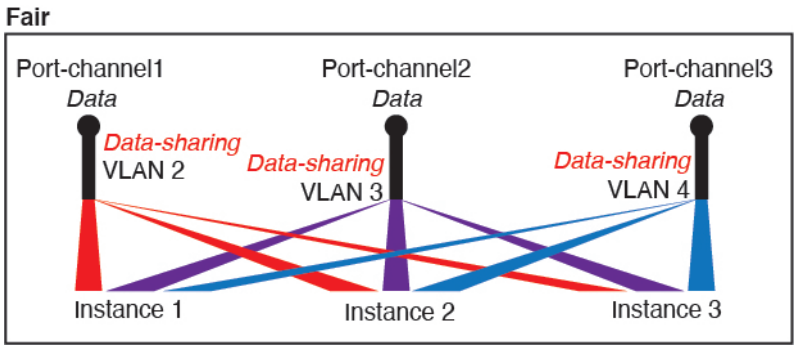
그림 *3:* 좋음 *:* 하나의 상위에서 여러 하위 인터페이스 그룹 공유



2. 양호 - 여러 상위 인터페이스 간에 하위 인터페이스를 공유합니다.

   예를 들어 Port-Channel2, Port-Channel4 및 Port-Channel4 대신 Port-Channel1.2, Port-Channel2.3 및 Port-Channel3.4를 공유합니다. 이러한 사용 방법은 동일한 상위 인터페이스에서 하위 인터페이스만 공유하는 것만큼 효율적이지는 않지만 여전히 VLAN 그룹의 장점을 활용합니다.

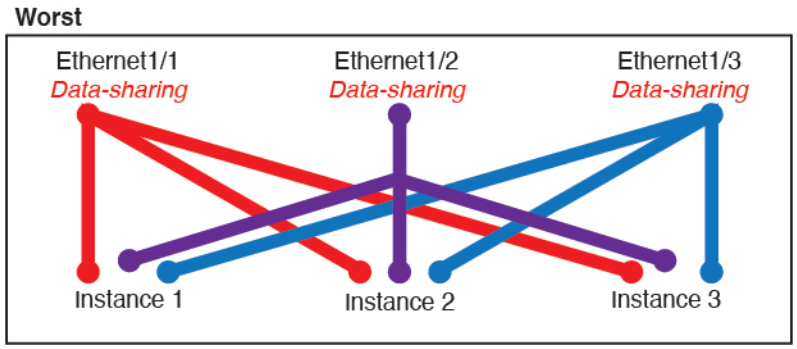
그림 *4:* 보통 *:* 개별 상위의 공유 하위 인터페이스



3. 최악 - 개별 상위 인터페이스(물리적 또는 EtherChannel)를 공유합니다.

   이 방법에서는 대부분의 전달 테이블 항목을 사용합니다.

   그림 *5:* 최악 *:* 공유 상위 인터페이스



공유 인터페이스 사용 예시

인터페이스 공유 및 확장성에 대한 예시는 다음 표를 참조하십시오. 아래 시나리오는 모든 인스턴스 간에 공유되는 관리를 위해 하나의 물리적/EtherChannel 인터페이스를 사용하거나 고가용성에 사용하기 위해 전용 하위 인터페이스와 함께 다른 물리적 인터페이스 또는 EtherChannel 인터페이스를 사용하는 것으로 가정합니다.

- 표 12: Firepower 9300(SM-44 3개)의 물리적/EtherChannel 인터페이스 및 인스턴스, 174 페이지
- 표 13: Firepower 9300(SM-44 3개)에 있는 상위 인터페이스 하나의 하위 인터페이스 및 인스턴스, 176 페이지
- 표 14: Firepower 9300(SM-44 1개)의 물리적/EtherChannel 인터페이스 및 인스턴스, 177 페이지
- 표 15: Firepower 9300(SM-44 1개)에 있는 상위 인터페이스 하나의 하위 인터페이스 및 인스턴스, 179 페이지

Firepower 9300(SM-44 3개)

다음 표의 내용은 물리적 인터페이스 또는 EtherChannel만 사용하는 9300의 SM-44 보안 모듈 3개에 적용됩니다. 하위 인터페이스가 없으면 최대 인터페이스 수가 제한됩니다. 또한 여러 물리적 인터페이스를 공유하는 경우 여러 하위 인터페이스를 공유할 때보다 더 많은 전달 테이블 리소스를 사용합니다.

각 SM-44 모듈은 인스턴스를 14개까지 지원할 수 있습니다. 제한을 초과하지 않도록 하기 위해 필요에 따라 모듈 간에 인스턴스가 분할됩니다.

표 ***12: Firepower 9300(SM-44 3개)***의 물리적***/EtherChannel*** 인터페이스 및 인스턴스

전용 인터페이스	공유 인터페이스	인스턴스 수	사용되는 전달 테이블의 퍼센트
32: • 8 • 8 • 8 • 8	**0**	**4:** • 인스턴스 1 • 인스턴스 2 • 인스턴스 3 • 인스턴스 4	16%
30: • 15 • 15	**0**	**2:** • 인스턴스 1 • 인스턴스 2	14%
14: • 14(각 1개)	**1**	**14:** • 인스턴스 1~인스턴스 14	46%
33: • 11(각 1개) • 11(각 1개) • 11(각 1개)	**3:** • 1 • 1 • 1	**33:** • 인스턴스 1~인스턴스 11 • 인스턴스 12~인스턴스 22 • 인스턴스 23~인스턴스 33	98%
33: • *11(각 1개)* • *11(각 1개)* • *12(각 1개)*	***3:*** • *1* • *1* • *1*	***34:*** • 인스턴스 *1*~인스턴스 *11* • 인스턴스 *12*~인스턴스 *22* • 인스턴스 *23*~인스턴스 *34*	*102%* 허용 안 됨
30: • 30(각 1개)	**1**	**6:** • 인스턴스 1~인스턴스 6	25%

전용 인터페이스	공유 인터페이스	인스턴스 수	사용되는 전달 테이블의 퍼센트
30: • 10(각 5개) • 10(각 5개) • 10(각 5개)	**3:** • 1 • 1 • 1	**6:** • 인스턴스 1~인스턴스 2 • 인스턴스 2~인스턴스 4 • 인스턴스 5~인스턴스 6	23%
30: • 30(각 6개)	**2**	**5:** • 인스턴스 1~인스턴스 5	28%
30: • 12(각 6개) • 18(각 6개)	**4:** • 2 • 2	**5:** • 인스턴스 1~인스턴스 2 • 인스턴스 2~인스턴스 5	26%
24: • 6 • 6 • 6 • 6	**7**	**4:** • 인스턴스 1 • 인스턴스 2 • 인스턴스 3 • 인스턴스 4	44%
24: • 12(각 6개) • 12(각 6개)	**14:** • 7 • 7	**4:** • 인스턴스 1~인스턴스 2 • 인스턴스 2~인스턴스 4	41%

다음 표의 내용은 단일 상위 물리적 인터페이스에서 하위 인터페이스를 사용하는 9300의 SM-44 보안 모듈 3개에 적용됩니다. 예를 들어 대형 EtherChannel 하나를 생성해 유사한 종류의 모든 인터페이스를 함께 번들로 포함한 다음 해당 EtherChannel의 하위 인터페이스를 공유합니다. 여러 물리적 인터페이스를 공유하는 경우 여러 하위 인터페이스를 공유할 때보다 더 많은 전달 테이블 리소스를 사용합니다.

각 SM-44 모듈은 인스턴스를 14개까지 지원할 수 있습니다. 제한을 초과하지 않도록 하기 위해 필요에 따라 모듈 간에 인스턴스가 분할됩니다.

표 ***13: Firepower 9300(SM-44 3***개***)***에 있는 상위 인터페이스 하나의 하위 인터페이스 및 인스턴스

전용 하위 인터페이스	공유 하위 인터페이스	인스턴스 수	사용되는 전달 테이블의 퍼센트
168: • 168(각 4개)	**0**	**42:** • 인스턴스 1~인스턴스 42	33%
224: • 224(각 16개)	**0**	**14:** • 인스턴스 1~인스턴스 14	27%
14: • 14(각 1개)	**1**	**14:** • 인스턴스 1~인스턴스 14	46%
33: • 11(각 1개) • 11(각 1개) • 11(각 1개)	**3:** • 1 • 1 • 1	**33:** • 인스턴스 1~인스턴스 11 • 인스턴스 12~인스턴스 22 • 인스턴스 23~인스턴스 33	98%
70: • 70(각 5개)	**1**	**14:** • 인스턴스 1~인스턴스 14	46%
165: • 55(각 5개) • 55(각 5개) • 55(각 5개)	**3:** • 1 • 1 • 1	**33:** • 인스턴스 1~인스턴스 11 • 인스턴스 12~인스턴스 22 • 인스턴스 23~인스턴스 33	98%
70: • 70(각 5개)	**2**	**14:** • 인스턴스 1~인스턴스 14	46%
165: • 55(각 5개) • 55(각 5개) • 55(각 5개)	**6:** • 2 • 2 • 2	**33:** • 인스턴스 1~인스턴스 11 • 인스턴스 12~인스턴스 22 • 인스턴스 23~인스턴스 33	98%
70: • 70(각 5개)	**10**	**14:** • 인스턴스 1~인스턴스 14	46%

전용 하위 인터페이스	공유 하위 인터페이스	인스턴스 수	사용되는 전달 테이블의 퍼센트
165: • *55(각 5개)* • *55(각 5개)* • *55(각 5개)*	***30:*** • *10* • *10* • *10*	***33:*** • 인스턴스 *1*~인스턴스 *11* • 인스턴스 *12*~인스턴스 *22* • 인스턴스 *23*~인스턴스 *33*	*102%* 허용 안 됨

Firepower 9300(SM-44 1개)

다음 표의 내용은 물리적 인터페이스 또는 EtherChannel만 사용하는 Firepower 9300(SM-44 1개)에 적용됩니다. 하위 인터페이스가 없으면 최대 인터페이스 수가 제한됩니다. 또한 여러 물리적 인터페이스를 공유하는 경우 여러 하위 인터페이스를 공유할 때보다 더 많은 전달 테이블 리소스를 사용합니다.

Firepower 9300(SM-44 1개)은 인스턴스를 14개까지 지원할 수 있습니다.

표 ***14: Firepower 9300(SM-44 1개)***의 물리적***/EtherChannel*** 인터페이스 및 인스턴스

전용 인터페이스	공유 인터페이스	인스턴스 수	사용되는 전달 테이블의 퍼센트
32: • 8 • 8 • 8 • 8	**0**	**4:** • 인스턴스 1 • 인스턴스 2 • 인스턴스 3 • 인스턴스 4	16%
30: • 15 • 15	**0**	**2:** • 인스턴스 1 • 인스턴스 2	14%
14: • 14(각 1개)	**1**	**14:** • 인스턴스 1~인스턴스 14	46%
14: • 7(각 1개) • 7(각 1개)	**2:** • **1** • **1**	**14:** • 인스턴스 1~인스턴스 7 • 인스턴스 8~인스턴스 14	37%

전용 인터페이스	공유 인터페이스	인스턴스 수	사용되는 전달 테이블의 퍼센트
32: • 8 • 8 • 8 • 8	**1**	**4:** • 인스턴스 1 • 인스턴스 2 • 인스턴스 3 • 인스턴스 4	21%
32: • 16(각 8개) • 16(각 8개)	**2**	**4:** • 인스턴스 1~인스턴스 2 • 인스턴스 3~인스턴스 4	20%
32: • 8 • 8 • 8 • 8	**2**	**4:** • 인스턴스 1 • 인스턴스 2 • 인스턴스 3 • 인스턴스 4	25%
32: • 16(각 8개) • 16(각 8개)	**4:** • 2 • 2	**4:** • 인스턴스 1~인스턴스 2 • 인스턴스 3~인스턴스 4	24%
24: • 8 • 8 • 8	**8**	**3:** • 인스턴스 1 • 인스턴스 2 • 인스턴스 3	37%
10: • 10(각 2개)	**10**	**5:** • 인스턴스 1~인스턴스 5	69%
10: • 6(각 2개) • 4(각 2개)	**20:** • 10 • 10	**5:** • 인스턴스 1~인스턴스 3 • 인스턴스 4~인스턴스 5	59%

전용 인터페이스	공유 인터페이스	인스턴스 수	사용되는 전달 테이블의 퍼센트
14: • *12(각 2개)*	***10***	***7:*** • 인스턴스 *1*~인스턴스 *7*	*109%* 허용 안 됨

다음 표의 내용은 단일 상위 물리적 인터페이스에서 하위 인터페이스를 사용하는 Firepower 9300(SM-44 1개)에 적용됩니다. 예를 들어 대형 EtherChannel 하나를 생성해 유사한 종류의 모든 인터페이스를 함께 번들로 포함한 다음 해당 EtherChannel의 하위 인터페이스를 공유합니다. 여러 물리적 인터페이스를 공유하는 경우 여러 하위 인터페이스를 공유할 때보다 더 많은 전달 테이블 리소스를 사용합니다.

Firepower 9300(SM-44 1개)은 인스턴스를 14개까지 지원할 수 있습니다.

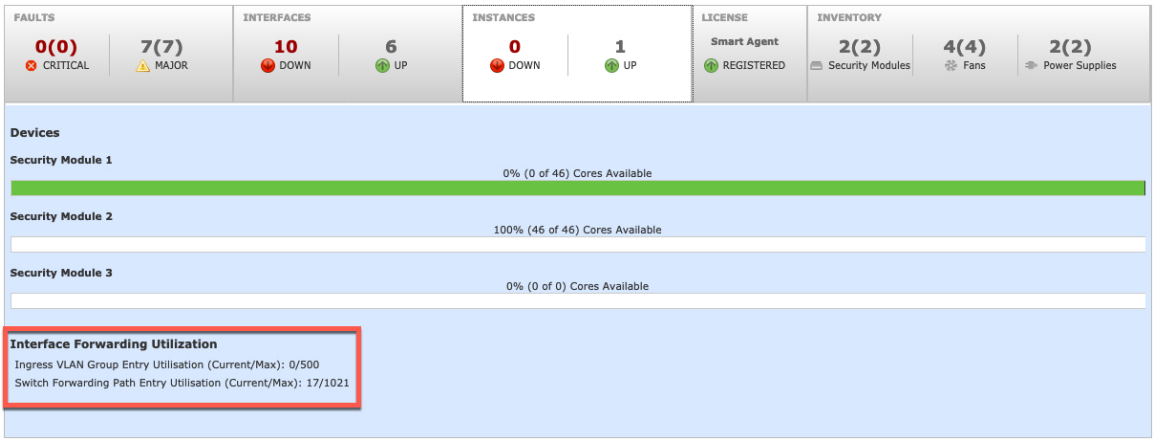
표 ***15: Firepower 9300(SM-44 1개)***에 있는 상위 인터페이스 하나의 하위 인터페이스 및 인스턴스

전용 하위 인터페이스	공유 하위 인터페이스	인스턴스 수	사용되는 전달 테이블의 퍼센트
112: • 112(각 8개)	**0**	**14:** • 인스턴스 1~인스턴스 14	17%
224: • 224(각 16개)	**0**	**14:** • 인스턴스 1~인스턴스 14	17%
14: • 14(각 1개)	**1**	**14:** • 인스턴스 1~인스턴스 14	46%
14: • 7(각 1개) • 7(각 1개)	**2:** • 1 • 1	**14:** • 인스턴스 1~인스턴스 7 • 인스턴스 8~인스턴스 14	37%
112: • 112(각 8개)	**1**	**14:** • 인스턴스 1~인스턴스 14	46%
112: • 56(각 8개) • 56(각 8개)	**2:** • 1 • 1	**14:** • 인스턴스 1~인스턴스 7 • 인스턴스 8~인스턴스 14	37%
112: • 112(각 8개)	**2**	**14:** • 인스턴스 1~인스턴스 14	46%

전용 하위 인터페이스	공유 하위 인터페이스	인스턴스 수	사용되는 전달 테이블의 퍼센트
112: • 56(각 8개) • 56(각 8개)	**4:** • 2 • 2	**14:** • 인스턴스 1~인스턴스 7 • 인스턴스 8~인스턴스 14	37%
140: • 140(각 10개)	**10**	**14:** • 인스턴스 1~인스턴스 14	46%
140: • 70(각 10개) • 70(각 10개)	**20:** • 10 • 10	**14:** • 인스턴스 1~인스턴스 7 • 인스턴스 8~인스턴스 14	37%

공유 인터페이스 리소스 보기

포워딩 테이블 및 VLAN 그룹 사용량을 보려면 **Instances(**인스턴스**)** > **Interface Forwarding Utilization(**인터페이스 포워딩 사용률**)** 영역을 확인하고 예를 들면 다음과 같습니다.



FTD에 대한 인라인 집합 링크 상태 전파

비활성 엔드포인트(bump in the wire)처럼 작동하는 인라인 집합은 두 인터페이스를 함께 슬롯에 포함해 기존 네트워크에 바인딩합니다. 이 기능을 사용하면 인접한 네트워크 디바이스의 구성 없이 네트워크 환경에 시스템을 설치할 수 있습니다. 인라인 인터페이스는 모든 트래픽을 조건 없이 수신하지만 이러한 인터페이스에서 수신한 모든 트래픽은 명시적으로 삭제되지 않는 한 인라인 집합으로부터 다시 전송됩니다.

FTD 애플리케이션에서 인라인 집합을 구성하고 링크 상태 전파를 활성화하면 FTD에서 FXOS 섀시로 인라인 집합 멤버십을 전송합니다. 링크 상태 전파는 인라인 집합의 인터페이스 중 하나가 중단될 때 인라인 인터페이스 쌍에서 두 번째 인터페이스를 자동으로 불러옵니다. 장애가 발생한 인터페이

스가 복원되면 두 번째 인터페이스도 자동으로 활성화됩니다. 다시 말해, 한 인터페이스의 링크 상태가 변경되면 섀시가 변경사항을 감지하고 다른 인터페이스의 링크 상태도 일치하도록 업데이트합니다. 섀시가 링크 상태 변경사항을 전파하려면 최대 4초가 걸립니다. 링크 상태 전파는 라우터가 장애 상태인 네트워크 디바이스를 우회해 트래픽을 자동으로 다시 라우팅하도록 구성된 탄력적인 네트워크 환경에서 특히 유용합니다.

인터페이스에 대한 지침 및 제한 사항

VLAN 하위 인터페이스

- 이 문서에서는 *FXOS* VLAN 하위 인터페이스에 대해서만 설명합니다. FTD 애플리케이션 내에 하위 인터페이스를 추가할 수 있습니다. 자세한 내용은 FXOS 인터페이스와 애플리케이션 인터페이스 비교, 166 페이지를 참조하십시오.
- 하위 인터페이스(및 상위 인터페이스)는 컨테이너 인스턴스에만 할당할 수 있습니다.

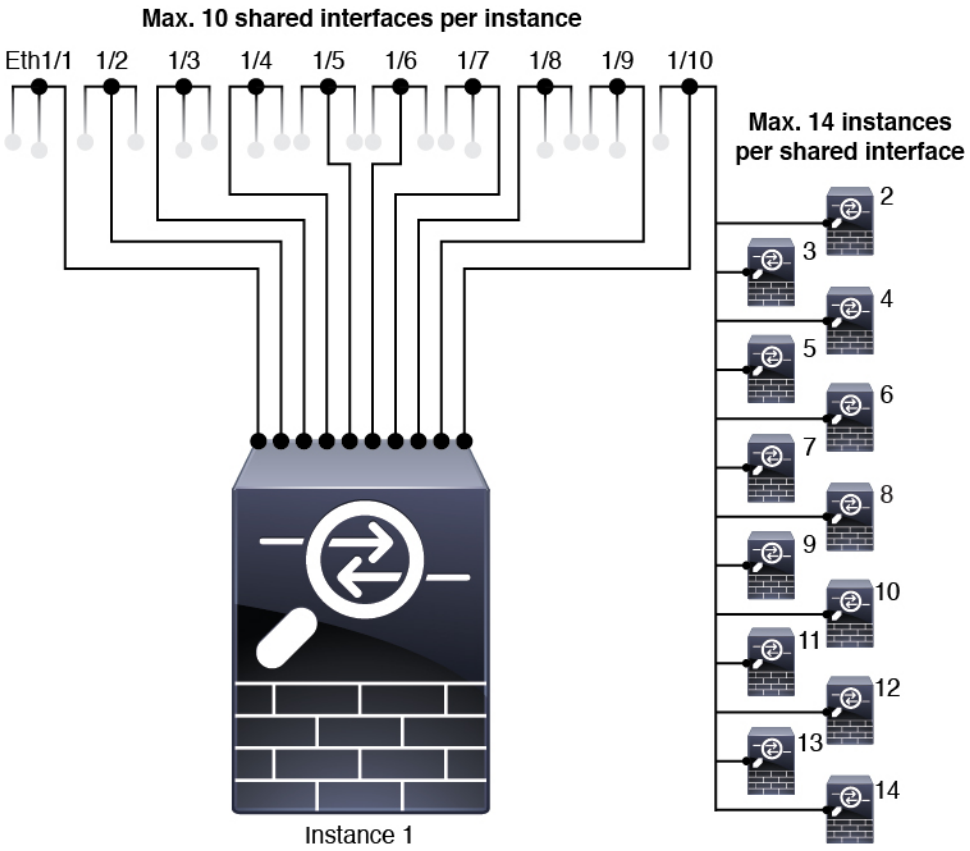
참고 컨테이너 인스턴스에 상위 인터페이스를 할당하는 경우에는 태그가 지정되지 않은(비 VLAN) 트래픽만 전달합니다. 태그가 지정되지 않은 트래픽을 전달하려는 경우가 아니라면 상위 인터페이스를 할당하지 마십시오. 클러스터 유형 인터페이스에는 상위 인터페이스를 사용할 수 없습니다.

- 하위 인터페이스는 데이터 또는 데이터 공유 유형 인터페이스와 클러스터 유형 인터페이스에서 지원됩니다. 클러스터 인터페이스에 하위 인터페이스를 추가하면 네이티브 클러스터에서 해당 인터페이스를 사용할 수 없습니다
- 다중 인스턴스 클러스터링의 경우 FXOS 하위 인터페이스는 데이터 인터페이스에서 지원되지 않습니다. 그러나 하위 인터페이스는 클러스터 제어 링크에 대해 지원되므로 전용 EtherChannel 또는 EtherChannel의 하위 인터페이스를 클러스터 제어 링크에 사용할 수 있습니다. 애플리케이션 정의 하위 인터페이스는 데이터 인터페이스에 대해 지원됩니다.
- VLAN ID는 최대 500개까지 생성할 수 있습니다.
- 논리적 디바이스 애플리케이션 내에서 다음과 같은 제한 사항을 참조하십시오. 인터페이스 할당을 계획할 때 이러한 제한 사항을 염두에 두십시오.
  - 하위 인터페이스를 FTD 인라인 집합용으로 또는 패시브 인터페이스로 사용할 수는 없습니다.
  - 페일오버 링크용으로 하위 인터페이스를 사용하는 경우에는 해당 상위 인터페이스의 모든 하위 인터페이스와 상위 인터페이스 자체가 페일오버 링크로 사용되도록 제한됩니다. 페일오버 링크로 사용할 수 없는 하위 인터페이스도 있고, 일반 데이터 인터페이스로 사용할 수 없는 하위 인터페이스도 있습니다.

데이터 공유 인터페이스

- 데이터 공유 인터페이스는 기본 인터페이스와 함께 사용할 수 없습니다.
- 공유 인터페이스당 최대 인스턴스 수는 14개입니다. 예를 들어 Instance1~Instance14에 Ethernet1/1을 할당할 수 있습니다.

  인스턴스당 최대 공유 인터페이스 수는 10개입니다. 예를 들어 Instance1에 Ethernet1/1.1~Ethernet1/1.10을 할당할 수 있습니다.



- 데이터 공유 인터페이스는 클러스터에서 사용할 수 없습니다.
- 논리적 디바이스 애플리케이션 내에서 다음과 같은 제한 사항을 참조하십시오. 인터페이스 할당을 계획할 때 이러한 제한 사항을 염두에 두십시오.
  - 데이터 공유 인터페이스는 투명 방화벽 모드 디바이스에서 사용할 수 없습니다.
  - 데이터 공유 인터페이스는 FTD 인라인 집합 또는 패시브 인터페이스와 함께 사용할 수 없습니다.
  - 데이터 공유 인터페이스는 페일오버 링크용으로 사용할 수 없습니다.

인라인 집합 **FTD**

- 물리적 인터페이스(일반 포트와 breakout 포트 둘 다) 및 EtherChannel용으로 지원됩니다. 하위 인터페이스는 지원되지 않습니다.
- 링크 상태 전파가 지원됩니다.

하드웨어 바이패스

- FTD용으로 지원됩니다. ASA용 일반 인터페이스로 사용할 수 있습니다.
- FTD에서는 인라인 집합을 사용하는 하드웨어 바이패스만 지원합니다.
- Breakout 포트에 대해 하드웨어 바이패스 지원 인터페이스를 구성할 수 없습니다.
- 하드웨어 바이패스 인터페이스를 EtherChannel에 포함해 하드웨어 바이패스용으로 사용할 수는 없으며 EtherChannel에서 일반 인터페이스로 사용할 수는 있습니다.
- 하드웨어 바이패스 은(는) 고가용성 모드에서 지원되지 않습니다.

기본 **MAC** 주소

기본 인스턴스의 경우**:**

기본 MAC 주소 할당은 인터페이스의 유형에 따라 다릅니다.

- 물리적 인터페이스 - 물리적 인터페이스는 버닝된 MAC 주소를 사용합니다.
- EtherChannel - EtherChannel의 경우 채널 그룹에 속한 모든 인터페이스가 동일한 MAC 주소를 공유합니다. 이 기능은 EtherChannel을 네트워크 애플리케이션 및 사용자에게 투명하게 만듭니다. 이들은 논리적 연결만 볼 수 있으며, 개별 링크에 대해서는 모르기 때문입니다. 포트 채널 인터페이스는 풀의 고유 MAC 주소를 사용하며 인터페이스 멤버십은 MAC 주소에 영향을 주지 않습니다.

컨테이너 인스턴스의 경우**:**

- 모든 인터페이스의 MAC 주소를 MAC 주소 풀에서 가져옵니다. 하위 인터페이스의 경우에는 MAC 주소를 수동으로 구성할 때 적절한 분류를 위해 동일한 상위 인터페이스의 모든 하위 인터페이스에 대해 고유한 MAC 주소를 사용해야 합니다. 컨테이너 인스턴스 인터페이스용 자동 MAC 주소, 209 페이지의 내용을 참조하십시오.

인터페이스 구성

기본적으로 물리적 인터페이스는 비활성화되어 있습니다. 인터페이스 활성화, EtherChannels 추가, VLAN 하위 인터페이스 추가, 인터페이스 속성 수정, breakout 포트 구성 작업을 수행할 수 있습니다.

참고 네트워크 모듈/EtherChannel을 제거하거나 EtherChannel에 인터페이스를 재할당하는 등 FXOS에서 인터페이스를 제거하면 ASA 구성에서 원래 명령이 유지되므로 필요한 조정을 수행할 수 있습니다. 구성에서 인터페이스를 제거하면 그 영향이 광범위하게 미칠 수 있습니다. ASA OS에서 이전 인터페이스 구성을 수동으로 제거할 수 있습니다.

인터페이스 활성화 또는 비활성화

각 인터페이스의 **Admin State(**관리 상태**)**를 활성화 또는 비활성화로 변경할 수 있습니다. 기본적으로 물리적 인터페이스는 비활성화되어 있습니다. VLAN 하위 인터페이스의 경우 관리 상태는 상위 인터페이스에서 상속됩니다.

프로시저

단계 **1** **Interfaces(**인터페이스**)**를 선택하여 Interfaces(인터페이스) 페이지를 엽니다.

Interfaces(인터페이스) 페이지는 페이지 상단에 현재 설치된 인터페이스를 시각적으로 표시하며 아래 표에서 설치된 인터페이스 목록을 제공합니다.

단계 **2** 인터페이스를 활성화하려면 비활성화된 슬라이더 비활성화됨( )를 클릭하여 활성화된 슬라이더 활성화됨( )로 변경합니다.

Yes(예**)**를 클릭하여 변경을 확인합니다. 해당 인터페이스의 시각적 표시가 회색에서 녹색으로 변경됩니다.

단계 **3** 인터페이스를 비활성화하려면 활성화된 슬라이더 활성화됨( )를 클릭하여 비활성화된 슬라이더 비활성화됨( )로 변경합니다.

Yes(예**)**를 클릭하여 변경을 확인합니다. 해당 인터페이스의 시각적 표시가 녹색에서 회색으로 변경됩니다.

실제 인터페이스 구성

인터페이스를 물리적으로 활성화 및 비활성화할 뿐만 아니라 인터페이스 속도 및 듀플렉스를 설정할 수 있습니다. 인터페이스를 사용하려면 FXOS에서 인터페이스를 물리적으로 활성화하고 애플리케이션에서 논리적으로 활성화해야 합니다.

시작하기 전에

- 이미 EtherChannel의 멤버인 인터페이스는 개별적으로 수정할 수 없습니다. EtherChannel에 인터페이스를 추가하기 전에 설정을 구성하십시오.

프로시저

단계 **1** **Interfaces(**인터페이스**)**를 선택하여 Interfaces(인터페이스) 페이지를 엽니다.

All Interfaces(모든 인터페이스**)** 페이지 상단에는 현재 설치되어 있는 인터페이스가 시각적으로 표시되며, 아래 표에는 설치되어 있는 인터페이스의 목록이 나와 있습니다.

단계 **2** 편집하려는 인터페이스 행에서 **Edit(**편집**)**를 클릭하여 **Edit Interface(**인터페이스 편집**)** 대화 상자를 엽니다.

단계 **3** 인터페이스를 활성화하려면 **Enable(**활성화**)** 확인란을 선택합니다. 인터페이스를 비활성화하려면 **Enable(**활성화**)** 확인란의 선택을 취소합니다.

단계 **4** 인터페이스 유형을 선택합니다.

- 데이터
- 데이터 공유 - 컨테이너 인스턴스에만 해당됩니다.
- 관리
- **Firepower** - FTD에만 해당됩니다.
- 클러스터 - 클러스터 유형은 선택하지 마십시오. 기본적으로 클러스터 제어 링크는 Port-channel 48에서 자동으로 생성됩니다.

단계 **5** (선택 사항) **Speed(**속도**)** 드롭다운 목록에서 인터페이스의 속도를 선택합니다.

단계 **6** (선택 사항) 인터페이스가 **Auto Negotiation(**자동 협상**)**을 지원하는 경우 **Yes(**예**)** 또는 **No(**아니요**)** 라디오 버튼을 클릭합니다.

단계 **7** (선택 사항) **Duplex(**듀플렉스**)** 드롭다운 목록에서 인터페이스의 듀플렉스를 선택합니다.

단계 **8** (선택 사항) 이전에 구성한 **Network Control Policy(**네트워크 제어 정책**)**를 선택합니다.

단계 **9** (선택 사항) 명시적으로 디바운스 시간**(ms)**을 구성합니다. 0~15000밀리초 사이의 값을 입력합니다.

단계 **10** **OK(**확인**)**를 클릭합니다.

EtherChannel(포트 채널) 추가

EtherChannel(포트 채널로 알려짐)은 동일한 미디어 유형 및 용량의 멤버 인터페이스를 최대 16개까지 포함할 수 있으며 동일한 속도 및 듀플렉스로 설정해야 합니다. 미디어 유형은 RJ-45 또는 SFP일 수 있으며 서로 다른 유형(구리 및 광섬유)의 SFP를 혼합할 수 있습니다. 더 큰 용량의 인터페이스에서는 속도를 낮게 설정하여 인터페이스 용량(예: 1GB 및 10GB 인터페이스)을 혼합하여 사용할 수 없습니다. LACP(Link Aggregation Control Protocol)에서는 두 네트워크 디바이스 간의 LACPDU(Link Aggregation Control Protocol Data Units)를 교환하여 인터페이스를 취합합니다.

EtherChannel의 각 물리적 데이터 또는 데이터 공유 인터페이스를 다음과 같이 구성할 수 있습니다.

- Active(활성화) — LACP 업데이트를 보내고 받습니다. 액티브 EtherChannel은 액티브 또는 패시브 EtherChannel과의 연결을 설정할 수 있습니다. LACP 트래픽 양을 최소화할 필요가 없는 한 액티브 모드를 사용해야 합니다.
- On(켜짐) — EtherChannel은 항상 켜져 있으며 LACP는 사용되지 않습니다. "on"으로 된 EtherChannel은 오로지 또 다른 "on" 상태의 EtherChannel과 연결을 설정할 수 있습니다.

참고 On에서 활성화, 또는 활성화에서 On으로 모드를 변경하는 경우 EtherChannel가 작동하는 데 최대 3분이 걸립니다.

비 데이터 인터페이스는 액티브 모드만 지원합니다.

LACP에서는 사용자의 작업 없이 EtherChannel에 링크를 자동으로 추가 및 삭제하는 작업을 조율합니다. 또한 구성 오류를 처리하고 멤버 인터페이스의 양끝이 모두 올바른 채널 그룹에 연결되어 있는지 확인합니다. "On" 모드에서는 인터페이스가 중단될 경우 채널 그룹의 스탠바이 인터페이스를 사용할 수 없으며, 연결 및 구성이 확인되지 않습니다.

Firepower 4100/9300 섀시에서 EtherChannel을 생성하면 물리적 링크가 가동 중이더라도 EtherChannel은 논리적 디바이스에 할당될 때까지 Active LACP(액티브 LACP) 모드인 경우 **Suspended(**일시 중단**)** 상태로, On LACP(LACP 켜짐) 모드인 경우 **Down(**중단**)** 상태로 유지됩니다. 다음의 상황에서는 EtherChannel의 **Suspended(**일시 중단**)** 상태가 해제됩니다.

참고 QSFPH40G-CUxM의 경우, 자동 협상은 기본값으로 항상 활성화되어 있으며 비활성화할 수 없습니다.

- EtherChannel이 독립형 논리적 디바이스에 대한 데이터 인터페이스 또는 관리 인터페이스로 추가됩니다.
- EtherChannel이 클러스터의 일부인 논리적 디바이스에 대한 관리 인터페이스 또는 클러스터 제어 링크로 추가됩니다.
- EtherChannel이 클러스터의 일부이며 유닛 하나 이상이 클러스터에 조인된 논리적 디바이스에 대한 데이터 인터페이스로 추가됩니다.

EtherChannel은 논리적 디바이스에 할당될 때까지 나타나지 않습니다. EtherChannel을 논리적 디바이스에서 제거하거나 논리적 디바이스가 삭제된 경우, EtherChannel은 **Suspended(**일시 중단**)** 또는 **Down(**중단**)** 상태로 전환됩니다.

프로시저

단계 **1** **Interfaces(**인터페이스**)**를 선택하여 Interfaces(인터페이스) 페이지를 엽니다.

All Interfaces(모든 인터페이스**)** 페이지 상단에는 현재 설치되어 있는 인터페이스가 시각적으로 표시되며, 아래 표에는 설치되어 있는 인터페이스의 목록이 나와 있습니다.

단계 **2** 인터페이스 테이블 위에 있는 **Add Port Channel(**포트 채널 추가**)**을 클릭하여 **Add Port Channel(**포트 채널 추가**)** 대화 상자를 엽니다.

단계 **3** **Port Channel ID(**포트 채널 **ID)** 필드에 포트 채널의 ID를 입력합니다. 유효한 값은 1~47입니다.

Port-channel 48은 클러스터된 논리적 디바이스를 구축할 때 클러스터 제어 링크로 예약됩니다. 클러스터 제어 링크에 포트 채널 48을 사용하지 않으려면 포트 채널 48을 삭제한 다음 다른 ID로 클러스터 유형 EtherChannel을 구성하면 됩니다.여러 클러스터 유형 EtherChannel과 다중 인스턴스 클러스터링에 사용할 VLAN 하위 인터페이스를 추가할 수 있습니다. 인트라 섀시 클러스터링(intra-chassis clustering)의 경우, 클러스터 EtherChannel에 인터페이스를 할당하지 마십시오.

단계 **4** 포트 채널을 활성화하려면 **Enable(**활성화**)** 확인란을 선택합니다. 포트 채널을 비활성화하려면 **Enable(**활성화**)** 확인란의 선택을 취소합니다.

단계 **5** 인터페이스 유형을 선택합니다.

- 데이터
- 데이터 공유 - 컨테이너 인스턴스에만 해당됩니다.
- 관리
- **Firepower** - FTD에만 해당됩니다.
- 클러스터

단계 **6** 드롭다운 목록에서 멤버 인터페이스의 필요한 **Admin Speed(**관리 속도**)**를 설정합니다.

지정된 속도가 아닌 멤버 인터페이스를 추가하면 포트 채널에 성공적으로 조인되지 않습니다.

단계 **7** 데이터 또는 데이터 공유 인터페이스의 경우 LACP 포트 채널 모드를 **Active(**액티브**)** 또는 **On(**켜짐**)** 중에서 선택합니다.

비 데이터 또는 비 데이터 공유 인터페이스의 경우 모드는 항상 액티브입니다.

단계 **8** 멤버 인터페이스에 대해 필요한 **Admin Duplex(**관리 듀플렉스**)**, **Full Duplex(**풀 듀플렉스**)** 또는 **Half Duplex(**하프 듀플렉스**)**를 설정합니다

지정된 듀플렉스로 설정된 멤버 인터페이스를 추가하면 포트 채널에 성공적으로 조인되지 않습니다.

단계 **9** 인터페이스를 포트 채널에 추가하려면 **Available Interface(**사용 가능한 인터페이스**)** 목록에서 인터페이스를 선택하고 **Add Interface(**인터페이스 추가**)**를 클릭하여 Member ID(멤버 ID) 목록으로 해당 인터페이스를 이동시킵니다.

미디어 유형과 용량이 동일한 멤버 인터페이스는 최대 16개까지 추가할 수 있습니다. 멤버 인터페이스는 동일한 속도 및 듀플렉스로 설정되어야 하며, 이 포트 채널에 대해 설정한 속도 및 듀플렉스와 일치해야 합니다. 미디어 유형은 RJ-45 또는 SFP일 수 있으며 서로 다른 유형(구리 및 광섬유)의 SFP를 혼합할 수 있습니다. 더 큰 용량의 인터페이스에서는 속도를 낮게 설정하여 인터페이스 용량(예: 1GB 및 10GB 인터페이스)을 혼합하여 사용할 수 없습니다.

팁 한 번에 여러 인터페이스를 추가할 수 있습니다. 여러 개별 인터페이스를 선택하려면 **Ctrl** 키를 누른 상태에서 필요한 인터페이스를 클릭합니다. 인터페이스 범위를 선택하려면 범위에서 첫 번째 인터페이스를 선택한 다음 **Shift** 키를 누른 상태에서 범위에 있는 마지막 인터페이스를 선택합니다.

단계 **10** 포트 채널에서 인터페이스를 제거하려면 Member ID(멤버 ID) 목록의 인터페이스 오른쪽에 있는 **Delete(**삭제**)** 버튼을 클릭합니다.

단계 **11** **OK(**확인**)**를 클릭합니다.

컨테이너 인스턴스에 **VLAN** 하위 인터페이스 추가

섀시에는 하위 인터페이스를 500 개까지 추가할 수 있습니다.

다중 인스턴스 클러스터링의 경우 클러스터 유형 인터페이스에 하위 인터페이스만 추가할 수 있습니다. 데이터 인터페이스의 하위 인터페이스는 지원되지 않습니다.

인터페이스당 VLAN ID는 고유해야 하며 컨테이너 인스턴스 내에서 VLAN ID는 모든 할당된 인터페이스에 대해 고유해야 합니다. VLAN ID가 다른 컨테이너 인스턴스에 할당되었다면 별도의 인터페이스에서 해당 VLAN ID를 재사용할 수 있습니다. 그러나 동일한 ID를 사용하더라도 계속해서 각 하위 인터페이스에는 이 제한이 적용됩니다.

이 문서에서는 *FXOS* VLAN 하위 인터페이스에 대해서만 설명합니다. FTD 애플리케이션 내에 하위 인터페이스를 추가할 수 있습니다.

프로시저

단계 **1** **Interfaces(**인터페이스**)**를 선택하여 **All Interfaces(**모든 인터페이스**)** 탭을 엽니다.

All Interfaces(모든 인터페이스**)** 탭은 페이지 상단에 현재 설치된 인터페이스를 시각적으로 표시하며 아래 표에서 설치된 인터페이스 목록을 제공합니다.

단계 **2** **Add New(**새로 추가**)** > **Subinterface(**하위 인터페이스**)**를 클릭하여 **Add Subinterface(**하위 인터페이스 추가**)** 대화 상자를 엽니다.

단계 **3** 인터페이스 유형을 선택합니다.

- 데이터
- 데이터 공유
- 클러스터 — 클러스터 인터페이스에 하위 인터페이스를 추가하면 네이티브 클러스터에서 해당 인터페이스를 사용할 수 없습니다.

데이터 및 데이터 공유 인터페이스: 유형은 상위 인터페이스 유형의 영향을 받지 않으므로 상위 인터페이스가 Data-sharing(데이터 공유) 유형이더라도 하위 인터페이스는 Data(데이터) 유형으로 설정할 수 있습니다.

단계 **4** 드롭다운 목록에서 상위 **Interface(**인터페이스**)**를 선택합니다.

논리적 디바이스에 현재 할당되어 있는 물리적 인터페이스에 하위 인터페이스를 추가할 수는 없습니다. 상위 인터페이스의 다른 하위 인터페이스가 할당되어 있는 경우 상위 인터페이스 자체가 할당되어 있지 않다면 새 하위 인터페이스를 추가할 수 있습니다.

단계 **5** 1~4294967295 사이의 **Subinterface ID(**하위 인터페이스 **ID)**를 입력합니다.

이 ID는 상위 인터페이스 ID에 *interface_id.subinterface_id*로 추가됩니다. 예를 들어 ID가 100인 Ethernet1/1에 하위 인터페이스를 추가하는 경우 하위 인터페이스 ID는 Ethernet1/1.100이 됩니다. 이 ID는 VLAN ID와는 다르지만 편의상 두 ID가 일치하도록 설정할 수 있습니다.

단계 **6** 1~4095 사이의 **VLAN ID**를 설정합니다.

단계 **7** **OK(**확인**)**를 클릭합니다.

상위 인터페이스를 확장하여 해당 인터페이스 아래의 모든 하위 인터페이스를 표시합니다.

분할 케이블 구성

다음 절차에서는 Firepower 4100/9300 섀시에서 사용할 분할 케이블을 구성하는 방법을 보여줍니다. 분할 케이블을 사용하여 40Gbps 포트 1개 대신 10Gbps 포트 4개를 제공할 수 있습니다.

시작하기 전에

Breakout 포트에 대해 하드웨어 바이패스 지원 인터페이스를 구성할 수 없습니다.

프로시저

단계 **1** **Interfaces(**인터페이스**)**를 선택하여 Interfaces(인터페이스) 페이지를 엽니다.

Interfaces(인터페이스) 페이지는 페이지 상단에 현재 설치된 인터페이스를 시각적으로 표시하며 아래 표에서 설치된 인터페이스 목록을 제공합니다.

Breakout 케이블을 지원할 수 있지만 현재 구성되어 있지 않은 인터페이스는 해당 인터페이스 행에 Breakout 포트 아이콘으로 표시되어 있습니다. 분할 케이블을 사용하도록 구성된 인터페이스의 경우, 개별 분할 인터페이스가 별도로 나열되어 있습니다(예: Ethernet 2/1/1, 2/1/2, 2/1/3 및 2/1/4).

단계 **2** 40Gbps 인터페이스 1개를 10Gbps 인터페이스 4개로 변환하려면 다음과 같이 합니다.

a) 변환할 인터페이스의 **Breakout Port(Breakout** 포트**)** 아이콘을 클릭합니다.

Breakout Port Creation(Breakout 포트 생성) 대화 상자가 열리면서 계속 진행할 것인지 확인을 요청하고 섀시가 재부팅된다고 경고합니다.

b) **Yes(**예**)**를 클릭하여 확인합니다.

Firepower 섀시가 재부팅되고 지정된 인터페이스가 10Gbps 인터페이스 4개로 변환됩니다.

단계 **3** 10Gbps 분할 인터페이스 4개를 40Gbps 인터페이스 1개로 다시 변환하려면 다음과 같이 합니다.

a) 분할 인터페이스 중 하나에 대해 **Delete(**삭제**)**를 클릭합니다.

확인 대화 상자가 열리면서 계속 진행할 것인지 확인을 요청하고 분할 인터페이스 4개가 모두 삭제되며 섀시가 재부팅된다고 경고합니다.

b) **Yes(**예**)**를 클릭하여 확인합니다.

Firepower 섀시가 재부팅되고 지정된 인터페이스가 40Gbps 인터페이스 1개로 변환됩니다.

모니터링 인터페이스

Firepower Chassis Manager의 Interfaces(인터페이스) 페이지에서 섀시에 설치된 인터페이스의 상태를 확인하고 인터페이스 속성을 편집하며 인터페이스를 활성화 또는 비활성화하고 포트 채널을 생성할 수 있습니다.

인터페이스 페이지는 다음의 두 가지 섹션으로 구성됩니다.

- 상위 섹션에서는 섀시에 설치된 인터페이스를 시각적으로 표시합니다. 인터페이스에 마우스 커서를 대면 해당 인터페이스에 대한 자세한 정보를 얻을 수 있습니다.

  인터페이스에는 현재 상태를 표시하는 다음과 같은 색상 코드가 지정됩니다.

  - 녹색 — 인터페이스가 설치 및 활성화된 상태입니다.
  - 어두운 회색 — 인터페이스가 설치되었지만 비활성화된 상태입니다.
  - 빨간색 — 인터페이스의 작동 상태에 문제가 있습니다.
  - 밝은 회색 — 인터페이스가 설치되지 않았습니다.

참고 포트 채널에서 포트 역할을 하는 인터페이스는 이 목록에 나타나지 않습니다.

- 하단 섹션에는 **All Interfaces(**모든 인터페이스**)** 및 하드웨어 바이패스 등 2개의 탭이 있습니다. **All Interfaces(**모든 인터페이스**)** 탭에서: 각 인터페이스에 대해 인터페이스를 활성화 또는 비활성화할 수 있습니다. 또한 **Edit(**편집**)**을 클릭하면 속도 및 인터페이스 유형 등 인터페이스 속성을 편집할 수 있습니다. 하드웨어 바이패스에 대한 자세한 내용은 하드웨어 바이패스 쌍, 169 페이지 섹션을 참조하십시오.

참고 Port-channel 48 클러스터 유형 인터페이스에 멤버 인터페이스가 포함되지 않은 경우 **Operation State(**운영 상태**)**는 **failed(**실패**)**로 표시됩니다. 인트라 섀시 클러스터링(intra-chassis clustering)의 경우 이 EtherChannel에는 멤버 인터페이스가 필요하지 않으므로 이 Operation State(운영 상태)를 무시할 수 있습니다.

인터페이스 트러블슈팅

오류: 스위치 전달 경로에는 제한 개수인 **1024**개를 초과하는 **1076**개의 항목이 있습니다. 인터페이스를 추가하는 경우, 논리적 디바이스에 할당되는 공유 인터페이스의 수를 줄이거나 인터페이스를 공유하는 논리적 디바이스의 수를 줄이거나 공유되지 않는 하위 인터페이스를 대신 사용하십시오. 하위 인터페이스를 삭제하는 경우, 나머지 구성이 더 이상 스위치 전달 경로 테이블 내부에 맞게 최적화되지 않기 때문에 이 메시지가 표시됩니다. 삭제 활용 사례에 대한 트러블슈팅 정보는 **FXOS** 구성 가이드를 참조하십시오. **'fabric-interconnect'** 범위 아래에서 **'show detail'**을 사용하여 현재 스위치 전달 경로 항목 개수를 확인하십시오.

하나의 논리적 디바이스에서 하나의 공유 하위 인터페이스를 삭제하려고 시도할 때 이 오류가 표시되는 경우, 이는 동일한 논리적 디바이스 그룹과 동일한 하위 인터페이스 집합을 사용하라는 공유 하위 인터페이스에 대한 지침을 새 구성에서 따르지 않기 때문입니다. 하나의 논리적 디바이스에서 하나의 공유 하위 인터페이스를 삭제하는 경우, VLAN 그룹이 더 많아지므로 전달 테이블을 덜 효율적으로 사용하게 됩니다. 이 상황을 해결하려면 동일한 논리적 디바이스 그룹에 동일한 하위 인터페이스 집합을 유지할 수 있도록 CLI를 사용하여 공유 하위 인터페이스를 동시에 추가하고 삭제해야 합니다.

자세한 내용은 다음과 같은 시나리오를 참조하십시오. 이러한 시나리오는 다음과 같은 인터페이스와 논리적 디바이스로 시작됩니다.

- 동일한 상위 인터페이스에 설정되어 있는 공유 하위 인터페이스: Port-Channel1.100(VLAN 100), Port-Channel1.200(VLAN 200), Port-Channel1.300(VLAN 300)
- 논리적 디바이스 그룹: LD1, LD2, LD3, LD4

시나리오 **1:** 하나의 논리적 디바이스에서 하나의 하위 인터페이스를 제거하되, 해당 하위 인터페이스가 다른 논리적 디바이스에 할당된 상태로 두기

하위 인터페이스를 제거하지 마십시오. 대신, 애플리케이션 구성에서 해당 하위 인터페이스를 비활성화하십시오. 하위 인터페이스를 제거해야 하는 경우, 일반적으로 공유 인터페이스의 수를 줄여야 전달 테이블에서 적합한 상태를 유지할 수 있습니다.

시나리오 **2:** 하나의 논리적 디바이스의 집합에 있는 모든 하위 인터페이스 제거

CLI에서 논리적 디바이스의 집합에 있는 모든 하위 인터페이스를 제거한 다음, 제거 작업이 동시에 이루어지도록 구성을 저장합니다.

1. 참조하려면 VLAN 그룹을 확인합니다. 다음 출력에서 그룹 1에는 3개의 공유 하위 인터페이스를 나타내는 VLAN 100, 200 및 300이 포함되어 있습니다.

```
firepower# connect fxos
[...]
firepower(fxos)# show ingress-vlan-groups
ID   Class ID  Status         INTF         Vlan Status
1    1         configured
                                          100  present
                                          200  present
                                          300  present
2048 512       configured
                                          0    present
```

```
2049 511       configured
                                          0    present
firepower(fxos)# exit
firepower#
```

2. 변경할 논리적 디바이스에 할당된 공유 하위 인터페이스를 확인합니다.

```
firepower# scope ssa
firepower /ssa # scope logical-device LD1
firepower /ssa/logical-device # show external-port-link

External-Port Link:
    Name                          Port or Port Channel Name Port Type          App Name
   Description
    ------------------------------ ------------------------ ------------------ ----------
 -----------
    Ethernet14_ftd                Ethernet1/4              Mgmt               ftd
    PC1.100_ftd                   Port-channel1.100        Data Sharing       ftd
    PC1.200_ftd                   Port-channel1.200        Data Sharing       ftd
    PC1.300_ftd                   Port-channel1.300        Data Sharing       ftd
```

3. 논리적 디바이스에서 하위 인터페이스를 제거한 다음, 구성을 저장합니다.

```
firepower /ssa/logical-device # delete external-port-link PC1.100_ftd
firepower /ssa/logical-device* # delete external-port-link PC1.200_ftd
firepower /ssa/logical-device* # delete external-port-link PC1.300_ftd
firepower /ssa/logical-device* # commit-buffer
firepower /ssa/logical-device #
```

중간에 구성을 커밋한 경우 VLAN 그룹이 2개가 되므로 스위치 전달 경로 오류가 생성되어 구성을 저장하지 못했을 수 있습니다.

시나리오 **3:** 그룹에 있는 모든 논리적 디바이스에서 하위 인터페이스 제거

CLI에서 그룹에 있는 모든 논리적 디바이스에서 하위 인터페이스를 제거한 다음, 제거 작업이 동시에 이루어지도록 구성을 저장합니다. 예를 들면 다음과 같습니다.

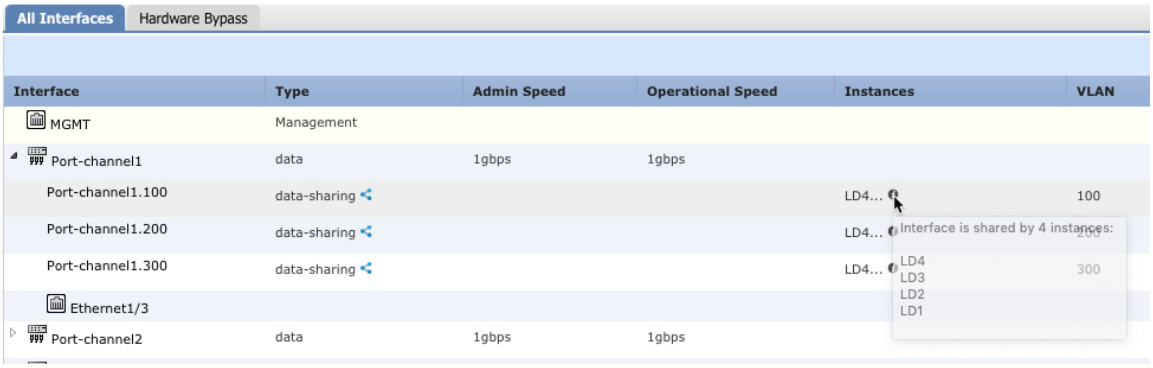
1. 참조하려면 VLAN 그룹을 확인합니다. 다음 출력에서 그룹 1에는 3개의 공유 하위 인터페이스를 나타내는 VLAN 100, 200 및 300이 포함되어 있습니다.

```
firepower# connect fxos
[...]
firepower(fxos)# show ingress-vlan-groups
ID   Class ID  Status         INTF         Vlan Status
1    1         configured
                                          100  present
                                          200  present
                                          300  present
2048 512       configured
                                          0    present
2049 511       configured
                                          0    present
```

2. 각 논리적 디바이스에 할당된 인터페이스를 확인하고 공통된 공유 하위 인터페이스를 참고합니다. 해당하는 하위 인터페이스가 동일한 상위 인터페이스에 있다면 하나의 VLAN 그룹에 속하며

show ingress-vlan-groups 목록과 일치해야 합니다. Firepower Chassis Manager에서 각 공유 하위 인터페이스로 마우스를 가져가 할당된 인스턴스를 확인할 수 있습니다.

그림 ***6:*** 공유 인터페이스당 인스턴스



CLI에서 할당된 인터페이스를 비롯한 모든 논리적 디바이스의 특성을 볼 수 있습니다.

```
firepower# scope ssa
firepower /ssa # show logical-device expand

Logical Device:
    Name: LD1
    Description:
    Slot ID: 1
    Mode: Standalone
    Oper State: Ok
    Template Name: ftd

    External-Port Link:
        Name: Ethernet14_ftd
        Port or Port Channel Name: Ethernet1/4
        Port Type: Mgmt
        App Name: ftd
        Description:

        Name: PC1.100_ftd
        Port or Port Channel Name: Port-channel1.100
        Port Type: Data Sharing
        App Name: ftd
        Description:

        Name: PC1.200_ftd
        Port or Port Channel Name: Port-channel1.200
        Port Type: Data Sharing
        App Name: ftd
        Description:

        System MAC address:
            Mac Address
            -----------
            A2:F0:B0:00:00:25

        Name: PC1.300_ftd
        Port or Port Channel Name: Port-channel1.300
        Port Type: Data Sharing
        App Name: ftd
        Description:
```

```
[...]

    Name: LD2
    Description:
    Slot ID: 1
    Mode: Standalone
    Oper State: Ok
    Template Name: ftd

    External-Port Link:
        Name: Ethernet14_ftd
        Port or Port Channel Name: Ethernet1/4
        Port Type: Mgmt
        App Name: ftd
        Description:

        Name: PC1.100_ftd
        Port or Port Channel Name: Port-channel1.100
        Port Type: Data Sharing
        App Name: ftd
        Description:

        Name: PC1.200_ftd
        Port or Port Channel Name: Port-channel1.200
        Port Type: Data Sharing
        App Name: ftd
        Description:

        System MAC address:
            Mac Address
            -----------
            A2:F0:B0:00:00:28

        Name: PC1.300_ftd
        Port or Port Channel Name: Port-channel1.300
        Port Type: Data Sharing
        App Name: ftd
        Description:

[...]

    Name: LD3
    Description:
    Slot ID: 1
    Mode: Standalone
    Oper State: Ok
    Template Name: ftd

    External-Port Link:
        Name: Ethernet14_ftd
        Port or Port Channel Name: Ethernet1/4
        Port Type: Mgmt
        App Name: ftd
        Description:

        Name: PC1.100_ftd
        Port or Port Channel Name: Port-channel1.100
        Port Type: Data Sharing
        App Name: ftd
        Description:

        Name: PC1.200_ftd
        Port or Port Channel Name: Port-channel1.200
```

```
        Port Type: Data Sharing
        App Name: ftd
        Description:

        System MAC address:
            Mac Address
            -----------
            A2:F0:B0:00:00:2B

        Name: PC1.300_ftd
        Port or Port Channel Name: Port-channel1.300
        Port Type: Data Sharing
        App Name: ftd
        Description:

[...]

    Name: LD4
    Description:
    Slot ID: 1
    Mode: Standalone
    Oper State: Ok
    Template Name: ftd

    External-Port Link:
        Name: Ethernet14_ftd
        Port or Port Channel Name: Ethernet1/4
        Port Type: Mgmt
        App Name: ftd
        Description:

        Name: PC1.100_ftd
        Port or Port Channel Name: Port-channel1.100
        Port Type: Data Sharing
        App Name: ftd
        Description:

        Name: PC1.200_ftd
        Port or Port Channel Name: Port-channel1.200
        Port Type: Data Sharing
        App Name: ftd
        Description:

        System MAC address:
            Mac Address
            -----------
            A2:F0:B0:00:00:2E

        Name: PC1.300_ftd
        Port or Port Channel Name: Port-channel1.300
        Port Type: Data Sharing
        App Name: ftd
        Description:

[...]
```

3. 각 논리적 디바이스에서 하위 인터페이스를 제거한 다음, 구성을 저장합니다.

```
firepower /ssa # scope logical device LD1
firepower /ssa/logical-device # delete external-port-link PC1.300_ftd
firepower /ssa/logical-device* # exit
firepower /ssa* # scope logical-device LD2
```

```
firepower /ssa/logical-device* # delete external-port-link PC1.300_ftd
firepower /ssa/logical-device* # exit
firepower /ssa* # scope logical-device LD3
firepower /ssa/logical-device* # delete external-port-link PC1.300_ftd
firepower /ssa/logical-device* # exit
firepower /ssa* # scope logical-device LD4
firepower /ssa/logical-device* # delete external-port-link PC1.300_ftd
firepower /ssa/logical-device* # commit-buffer
firepower /ssa/logical-device #
```

중간에 구성을 커밋한 경우, 2개의 VLAN 그룹이 생성되어 스위치 전달 경로 오류가 생성되어 구성을 저장하지 못했을 수 있습니다.

시나리오 **4:** 하나 이상의 논리적 디바이스에 하위 인터페이스 추가

CLI에서 그룹에 있는 모든 논리적 디바이스에 하위 인터페이스를 추가한 다음, 추가 작업이 동시에 이루어지도록 구성을 저장합니다.

1. 각 논리적 디바이스에 하위 인터페이스를 추가한 다음, 구성을 저장합니다.

```
firepower# scope ssa
firepower /ssa # scope logical-device LD1
firepower /ssa/logical-device # create external-port-link PC1.400_ftd Port-channel1.400
 ftd
firepower /ssa/logical-device/external-port-link* # exit
firepower /ssa/logical-device* # exit
firepower /ssa # scope logical-device LD2
firepower /ssa/logical-device # create external-port-link PC1.400_ftd Port-channel1.400
 ftd
firepower /ssa/logical-device/external-port-link* # exit
firepower /ssa/logical-device* # exit
firepower /ssa # scope logical-device LD3
firepower /ssa/logical-device # create external-port-link PC1.400_ftd Port-channel1.400
 ftd
firepower /ssa/logical-device/external-port-link* # exit
firepower /ssa/logical-device* # exit
firepower /ssa # scope logical-device LD4
firepower /ssa/logical-device # create external-port-link PC1.400_ftd Port-channel1.400
 ftd
firepower /ssa/logical-device/external-port-link* # commit-buffer
firepower /ssa/logical-device/external-port-link #
```

중간에 구성을 커밋한 경우, 2개의 VLAN 그룹이 생성되어 스위치 전달 경로 오류가 생성되어 구성을 저장하지 못했을 수 있습니다.

2. Port-channel1.400 VLAN ID가 VLAN 그룹 1에 추가된 것을 확인할 수 있습니다.

```
firepower /ssa/logical-device/external-port-link # connect fxos
[...]
firepower(fxos)# show ingress-vlan-groups
ID   Class ID  Status         INTF         Vlan Status
1    1         configured
                                          200  present
                                          100  present
                                          300  present
                                          400  present
2048 512       configured
                                          0    present
2049 511       configured
```

```
                                                    0     present
firepower(fxos)# exit
firepower /ssa/logical-device/external-port-link #
```

인터페이스 내역

기능 이름	플랫폼 릴리스	기능 정보
FTD 작동 링크 상태와 물리적 링크 상태 간 동기화	2.9.1	이제 새시가 FTD 작동 링크 상태를 데이터 인터페이스의 물리적 링크 상태와 동기화할 수 있습니다. 현재로서는, FXOS 관리 상태가 작동 중이고 물리적 링크 상태가 작동 중이면 인터페이스는 작동 상태가 됩니다. FTD 애플리케이션 인터페이스 관리 상태는 고려되지 않습니다. 예를 들어 FTD에서 동기화하지 않으면 FTD 애플리케이션이 완전히 온라인 상태가 되기 전에 데이터 인터페이스가 물리적으로 작동 상태가 되거나 FTD 종료를 시작한 후 일정 기간 동안 작동 상태를 유지할 수 있습니다. 인라인 집합의 경우 FTD에서 트래픽을 처리하기 전에 외부 라우터가 FTD로 트래픽 전송을 시작할 수 있으므로 이러한 상태 불일치로 인해 패킷이 삭제될 수 있습니다. 이 기능은 기본적으로 비활성화되어 있으며 FXOS에서 논리적 디바이스별로 활성화할 수 있습니다. 참고 이 기능은 클러스터링, 컨테이너 인스턴스 또는 Radware vDP 데코레이터가 포함된 FTD에는 지원되지 않습니다. ASA에서도 지원되지 않습니다. 신규/수정된 Firepower Chassis Manager 화면: **Logical Devices(**논리적 디바이스**) > Enable Link State(**링크 상태 활성화**)** 신규/수정된 FXOS 명령: **set link-state-sync enabled**, **show interface expand detail**
클러스터 유형 인터페이스의 VLAN 하위 인터페이스 지원(다중 인스턴스 전용)	2.8.1	다중 인스턴스 클러스터에 사용하기 위해 클러스터 유형 인터페이스에서 VLAN 하위 인터페이스를 생성할 수 있습니다. 각 클러스터에는 고유한 클러스터 제어 링크가 필요하므로 VLAN 하위 인터페이스는 이 요구 사항을 충족하는 간단한 방법을 제공합니다. 아니면 클러스터마다 전용 EtherChannel을 할당해도 됩니다. 이제 여러 클러스터 유형 인터페이스가 허용됩니다. 신규/수정된 화면: **Interfaces(**인터페이스**) > All Interfaces(**모든 인터페이스**) > Add New(**새로 추가**)** 드롭다운 메뉴 **> Subinterface(**하위 인터페이스**) > Type(**유형**)** 필드
우발 상황 없이 500개의 VLAN 지원	2.7.1	이전에는 디바이스에서 상위 인터페이스 수 및 기타 구축 결정 사항에 따라 250~500개의 VLAN을 지원했습니다. 이제 모든 경우에 500 VLAN을 사용할 수 있습니다.

기능 이름	플랫폼 릴리스	기능 정보
컨테이너 인스턴스에 사용할 VLAN 하위 인터페이스	2.4.1	물리적 인터페이스를 유연하게 사용할 수 있도록 FXOS에서 VLAN 하위 인터페이스를 생성하는 동시에 여러 인스턴스 간에 인터페이스를 공유할 수 있습니다. 참고 FTD 버전 6.3 이상이 필요합니다. 신규/수정된 화면: **Interfaces(**인터페이스**)** > **All Interfaces(**모든 인터페이스**)** > **Add New(**새로 추가**)** 드롭다운 메뉴 **> Subinterface(**하위 인터페이스**)** 신규/수정된 FMC 화면: **Devices(**디바이스**)** > **Device Management(**디바이스 관리**)** > **Edit(**수정**)** 아이콘 > **Interfaces(**인터페이스**)** 탭
컨테이너 인스턴스용 데이터 공유 인터페이스	2.4.1	물리적 인터페이스를 유연하게 사용할 수 있도록 여러 인스턴스 간에 인터페이스를 공유할 수 있습니다. 참고 FTD 버전 6.3 이상이 필요합니다. 신규/수정된 화면: **Interfaces(**인터페이스**)** > **All Interfaces(**모든 인터페이스**)** > **Type(**유형**)**
On(켜기) 모드에서 데이터 EtherChannel 지원	2.4.1	이제 데이터 및 데이터 공유 EtherChannel을 Active LACP(액티브 LACP) 모드 또는 On(켜기) 모드로 설정할 수 있습니다. 다른 유형의 Etherchannel은 Active(액티브) 모드만 지원합니다. 신규/수정된 화면: **Interfaces(**인터페이스**)** > **All Interfaces(**모든 인터페이스**)** > **Edit Port Channel(**포트 채널 수정**)** > **Mode(**모드**)**
FTD 인라인 집합에서 EtherChannel 지원	2.1.1	이제 FTD 인라인 집합에서 EtherChannel을 사용할 수 있습니다.
인라인 집합 링크 상태 전파 지원 FTD	2.0.1	FTD 애플리케이션에서 인라인 집합을 구성하고 링크 상태 전파를 활성화하면 FTD에서 FXOS 섀시로 인라인 집합 멤버십을 전송합니다. 링크 상태 전파는 인라인 집합의 인터페이스 중 하나가 중단될 때 인라인 인터페이스 쌍에서 두 번째 인터페이스를 자동으로 불러옵니다.
하드웨어 우회 네트워크 모듈 지원 FTD	2.0.1	Hardware Bypass는 정전 중에 트래픽이 인라인 인터페이스 쌍 사이에서 계속 흐르도록 합니다. 이 기능은 소프트웨어 또는 하드웨어 오류의 경우 네트워크 연결성을 유지 관리하는 데 사용될 수 있습니다. 신규/수정된 FMC 화면: **Devices(**디바이스**)** > **Device Management(**디바이스 관리**)** > **Interfaces(**인터페이스**)** > **Edit Physical Interface(**물리적 인터페이스 수정**)**

기능 이름	플랫폼 릴리스	기능 정보
Firepower 이벤트 유형 인터페이스 FTD	1.1.4	FTD에서 사용할 인터페이스의 유형을 Firepower 이벤트로 지정할 수 있습니다. 이 인터페이스는 FTD 디바이스의 보조 관리 인터페이스입니다. 이 인터페이스를 사용하려면 FTD CLI에서 해당 IP 주소 및 기타 매개변수를 구성해야 합니다. 예를 들면 관리 트래픽을 이벤트(예: 웹 이벤트)에서 분리할 수 있습니다. FMC 구성 가이드 시스템 구성 장의 "관리 인터페이스" 섹션을 참조하십시오. 신규/수정된 Firepower Chassis Manager 화면: **Interfaces(**인터페이스**)** > **All Interfaces(**모든 인터페이스**)** > **Type(**유형**)**

10 장

논리적 디바이스

논리적 디바이스 정보

논리적 디바이스를 사용하면 애플리케이션 인스턴스 하나(ASA 또는 FTD)와 선택적 데코레이터 애플리케이션(Radware DefensePro)을 실행하여 서비스 체인을 만들 수 있습니다 .

논리적 디바이스를 추가할 때는 애플리케이션 인스턴스 유형 및 버전 정의, 인터페이스 할당, 애플리케이션 구성으로 푸시되는 부트스트랩 설정 작업도 수행합니다.

참고 Firepower 9300의 경우 섀시 내의 개별 모듈에 서로 다른 애플리케이션 유형(ASA 및 FTD)을 설치할 수 있습니다. 개별 모듈에서 애플리케이션 인스턴스 유형의 서로 다른 버전을 실행할 수도 있습니다.

독립형 논리적 디바이스와 클러스터형 논리적 디바이스

다음의 논리적 디바이스 유형을 추가할 수 있습니다.

- 독립형 — 독립형 유닛으로 또는 고가용성 쌍의 유닛으로 작동하는 독립형 논리적 디바이스입니다.
- 클러스터 — 클러스터형 논리적 디바이스에서는 여러 유닛을 함께 그룹화할 수 있으므로 처리량 증대 및 여러 디바이스의 이중화라는 목표를 달성하는 동시에 단일 디바이스(관리, 네트워크에 통합)의 모든 편의성을 제공합니다. Firepower 9300과 같은 다중 모듈 디바이스는 인트라 섀시 클러스터링(intra-chassis clustering)을 지원합니다. Firepower 9300의 경우 세 개 모듈 모두가 네이티브와 컨테이너 인스턴스 모두에 대해 클러스터에 참여해야 합니다. FDM에서는 클러스터링을 지원하지 않습니다.

논리적 디바이스 애플리케이션 인스턴스: 컨테이너 및 기본

다음 구축 유형으로 애플리케이션 인스턴스가 실행됩니다.

- 기본 인스턴스 — 기본 인스턴스는 보안 모듈/엔진의 모든 리소스(CPU, RAM 및 디스크 공간)를 사용합니다. 따라서 하나의 기본 인스턴스만 설치할 수 있습니다.
- 컨테이너 인스턴스 — 컨테이너 인스턴스는 보안 모듈/엔진의 리소스 하위 집합을 사용합니다. 따라서 여러 개의 컨테이너 인스턴스를 설치할 수 있습니다. 다중 인스턴스 기능은 FMC를 사용하는 FTD에 대해서만 지원되며, ASA 또는 FDM를 사용하는 FTD에 대해서는 지원되지 않습니다.

참고 다중 인스턴스 기능은 ASA 다중 컨텍스트 모드와 비슷하지만 구현은 서로 다릅니다. 다중 컨텍스트 모드에서는 단일 애플리케이션 인스턴스를 분할하는 반면 다중 인스턴스 기능 사용 시에는 독립적인 컨테이너 인스턴스를 사용할 수 있습니다. 컨테이너 인스턴스에서는 하드 리소스 분리, 별도의 구성 관리/다시 로드/소프트웨어 업데이트가 허용되며 전체 FTD 기능이 지원됩니다. 다중 컨텍스트 모드에서는 리소스가 공유되므로 지정된 플랫폼에서 더 많은 컨텍스트가 지원됩니다. FTD에서는 다중 상황 모드를 사용할 수 없습니다.

Firepower 9300의 경우 일부 모듈에서는 기본 인스턴스를, 다른 모듈에서는 컨테이너 인스턴스를 사용할 수 있습니다.

컨테이너 인스턴스 인터페이스

컨테이너 인스턴스에 대해 물리적 인터페이스를 유연하게 사용할 수 있도록 FXOS에서 VLAN 하위 인터페이스를 생성하는 동시에 여러 인스턴스 간에 인터페이스(VLAN 또는 물리적)를 공유할 수 있습니다. 기본 인스턴스는 VLAN 하위 인터페이스 또는 공유 인터페이스를 사용할 수 없습니다. 멀티 인스턴스 클러스터는 VLAN 하위 인터페이스 또는 공유된 인터페이스를 사용할 수 없습니다. 클러스터 EtherChannel의 하위 인터페이스를 사용할 수 있는 클러스터 제어 링크는 예외입니다. 공유 인터페이스 확장성, 170 페이지 및 컨테이너 인스턴스에 VLAN 하위 인터페이스 추가, 188 페이지를 참조하십시오.

참고 이 문서에서는 *FXOS* VLAN 하위 인터페이스에 대해서만 설명합니다. FTD 애플리케이션 내에 하위 인터페이스를 추가할 수 있습니다. 자세한 내용은 FXOS 인터페이스와 애플리케이션 인터페이스 비교, 166 페이지를 참조하십시오.

섀시가 패킷을 분류하는 방법

섀시에 들어오는 각 패킷은 분류되어야 합니다. 그러면 섀시에서 어떤 인스턴스에 패킷을 보낼지 판단할 수 있습니다.

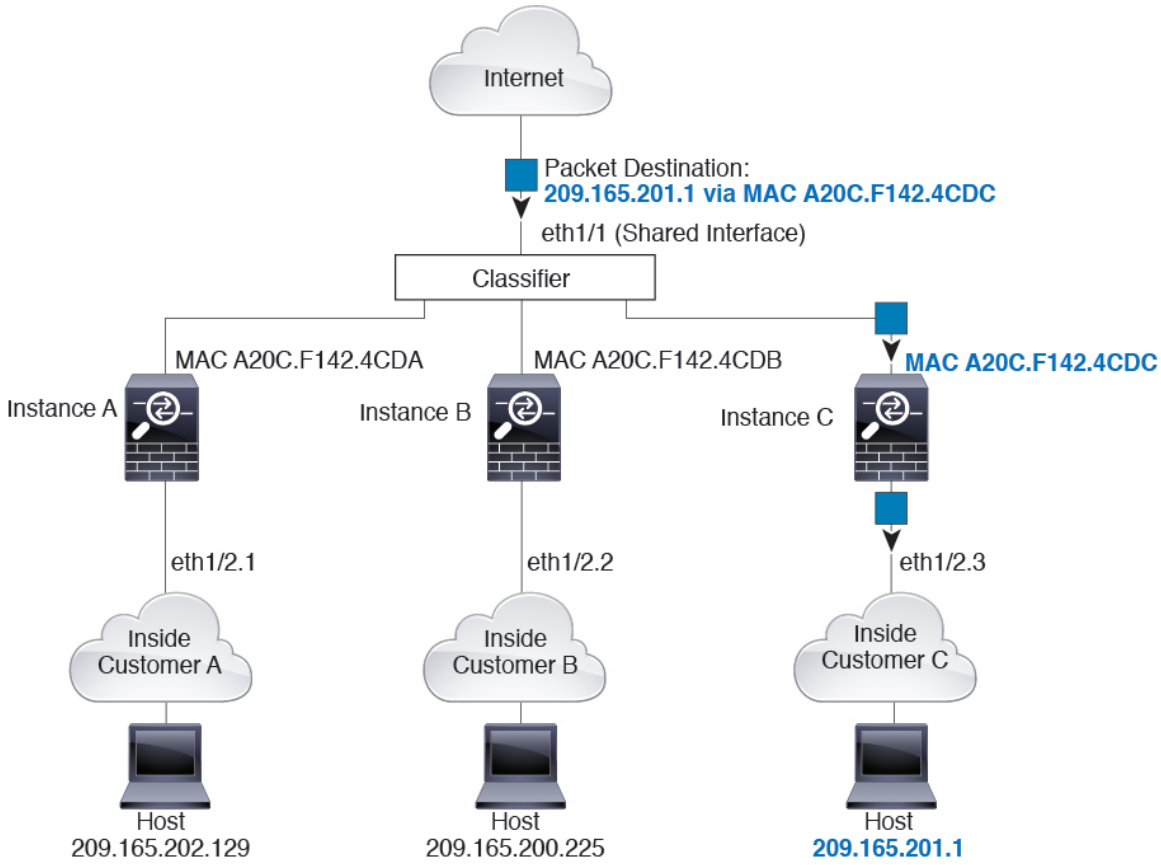
- 고유 인터페이스 - 단 하나의 인스턴스가 인그레스 인터페이스와 연결된 경우 섀시는 해당 패킷을 해당 인스턴스로 분류합니다. 투명 모드 또는 라우티드 모드의 브리지 그룹 멤버 인터페이스, 인라인 집합 또는 패시브 인터페이스의 경우에는 항상 이 방법을 사용하여 패킷을 분류합니다.
- 고유 MAC 주소 - 섀시가 공유 인터페이스를 포함한 모든 인터페이스에 대해 고유한 MAC 주소를 자동으로 생성합니다. 여러 인스턴스가 인터페이스 하나를 공유하는 경우 분류자는 각 인스턴스의 인터페이스에 할당된 고유 MAC 주소를 사용합니다. 업스트림 라우터는 고유 MAC 주소가 없으면 인스턴스로 직접 라우팅할 수 없습니다. 또한 애플리케이션 내에서 각 인터페이스를 구성할 때 수동으로 MAC 주소를 설정할 수도 있습니다.

참고 대상 MAC 주소가 멀티캐스트 또는 브로드캐스트 MAC 주소인 경우 패킷이 복제되어 각 인스턴스에 배포됩니다.

분류의 예

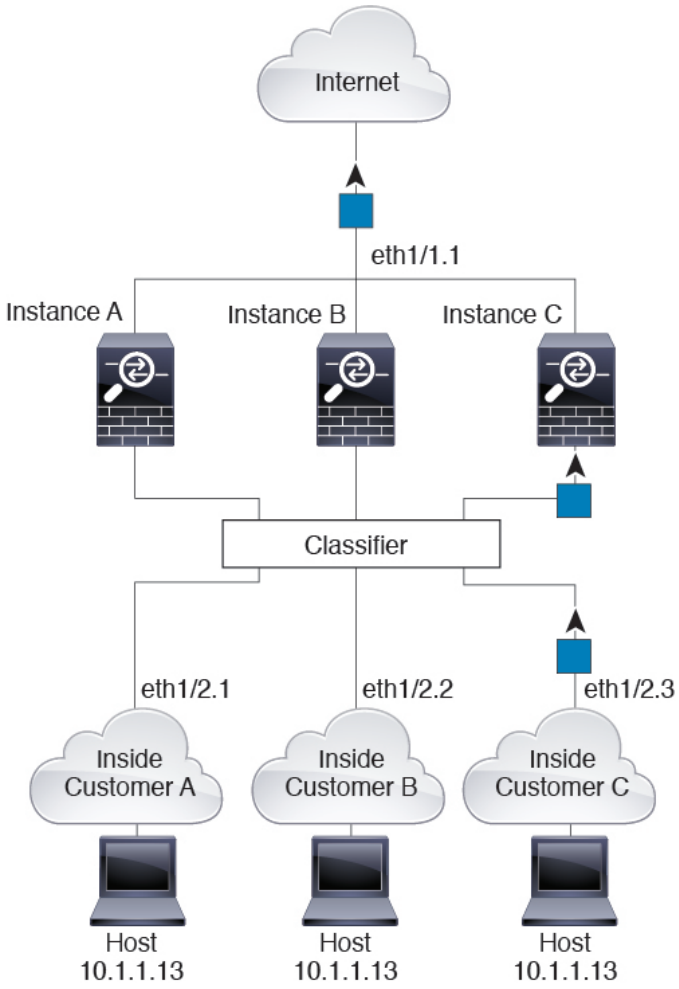
다음 그림은 외부 인터페이스를 공유하는 여러 인스턴스를 보여 줍니다. 분류자는 인스턴스 C에 패킷을 할당합니다. 라우터에서 패킷을 보내는 MAC 주소가 인스턴스 C에 포함되어 있기 때문입니다.

그림 *7: MAC* 주소를 사용하는 공유 인터페이스를 통한 패킷 분류



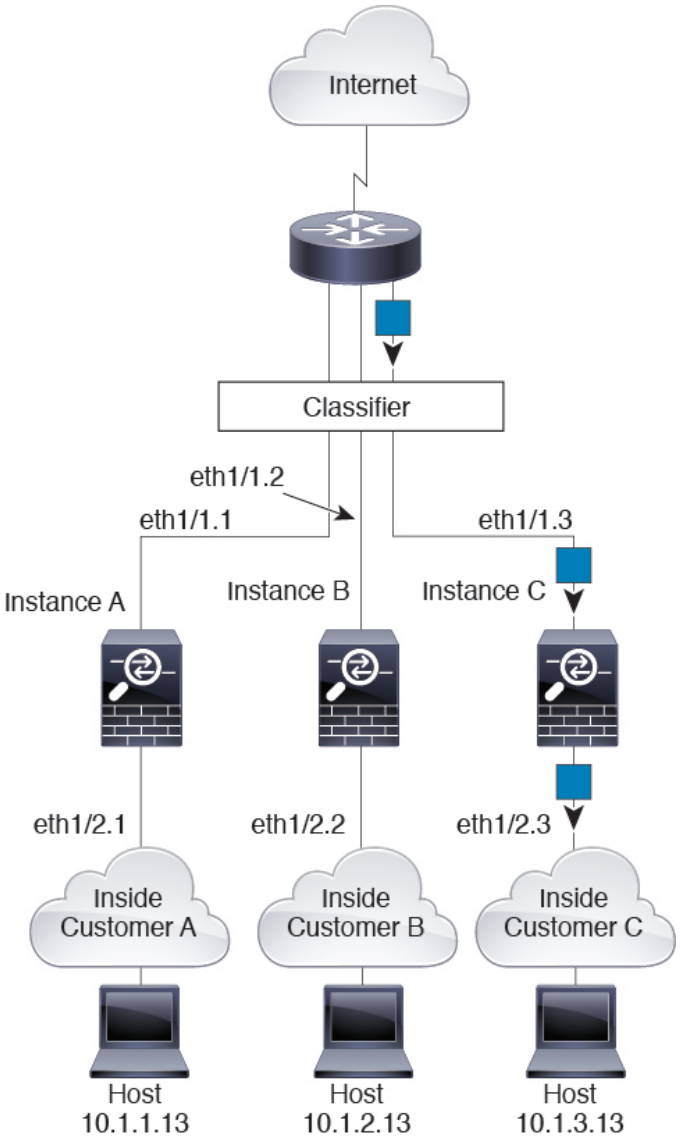
내부 네트워크에서 보낸 것을 비롯하여 모든 신규 수신 트래픽은 분류되어야 합니다. 다음 그림에는 인터넷에 액세스하는 네트워크 내의 인스턴스 C에 있는 호스트가 나와 있습니다. 분류자는 인스턴스 C에 패킷을 할당합니다. 인그레스 인터페이스가 인스턴스 C에 할당된 이더넷 1/2.3이기 때문입니다.

그림 *8:* 내부 네트워크로부터 수신하는 트래픽



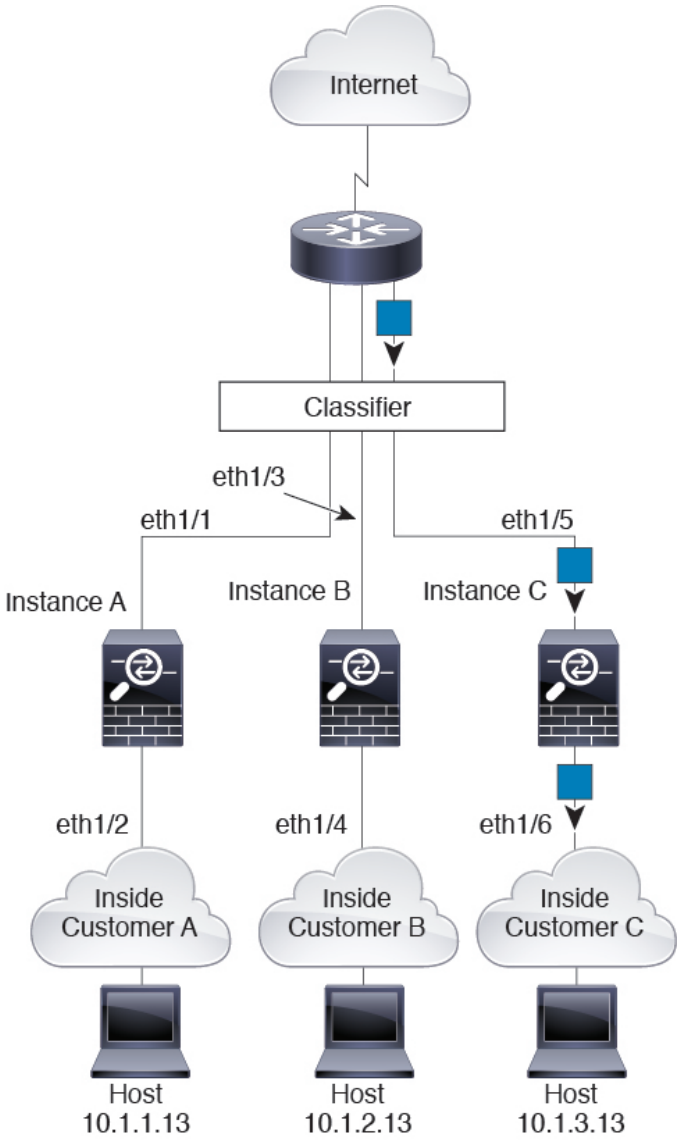
투명 방화벽의 경우 고유한 인터페이스를 사용해야 합니다. 다음 그림에는 인터넷의 네트워크 내 인스턴스 C에 있는 호스트로 전송되는 패킷이 나와 있습니다. 분류자는 인스턴스 C에 패킷을 할당합니다. 인그레스 인터페이스가 인스턴스 C에 할당된 이더넷 1/2.3이기 때문입니다.

그림 *9:* 투명한 방화벽 인스턴스



인라인 집합의 경우에는 고유 인터페이스를 사용해야 하며, 해당 인터페이스는 물리적 인터페이스 또는 EtherChannel이어야 합니다. 다음 그림에는 인터넷의 네트워크 내 인스턴스 C에 있는 호스트로 전송되는 패킷이 나와 있습니다. 분류자는 인스턴스 C에 패킷을 할당합니다. 인그레스 인터페이스가 인스턴스 C에 할당된 이더넷 1/5이기 때문입니다.

그림 ***10:*** 인라인 집합 ***FTD***

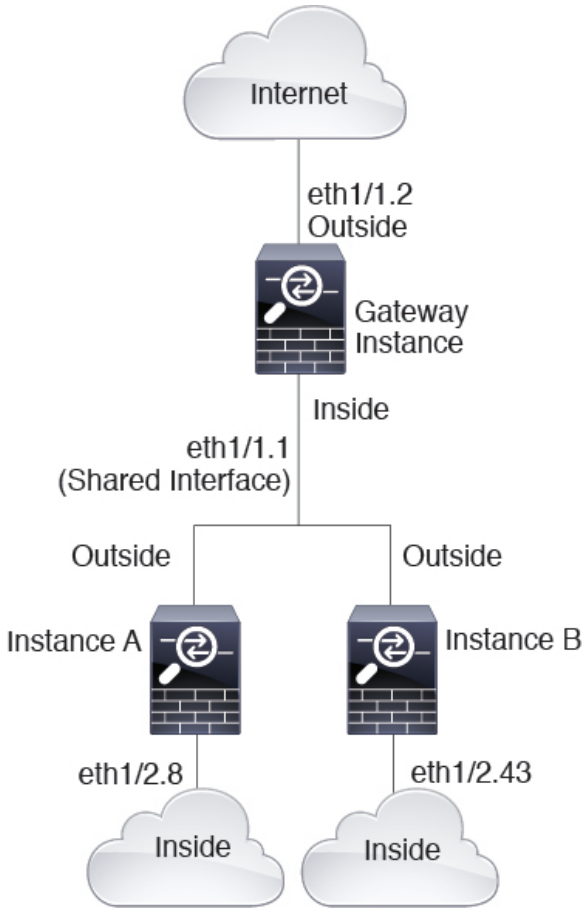


연속 컨테이너 인스턴스

다른 인스턴스 바로 앞에 컨테이너 인스턴스를 배치하는 것을 연속 컨테이너 인스턴스라고 합니다. 하나의 인스턴스의 외부 인터페이스는 다른 인스턴스의 내부 인터페이스와 동일한 인터페이스입니다. 최상위 인스턴스에서 공유 파라미터를 구성함으로써 일부 인스턴스의 구성을 간소화하고 싶다면 인스턴스 캐스케이딩이 유용할 수 있습니다.

다음 그림에는 게이트웨이 뒤에 인스턴스가 2개 있는 게이트웨이 인스턴스가 나와 있습니다.

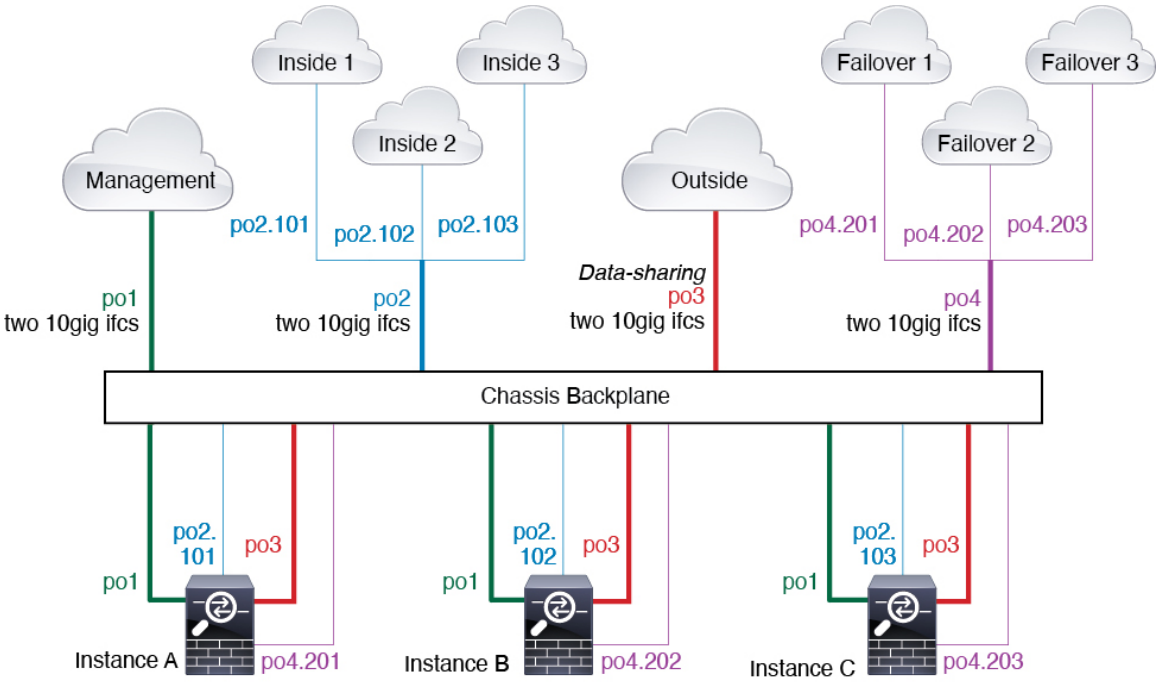
그림 *11:* 연속 컨테이너 인스턴스



일반적인 다중 인스턴스 구축

다음 예에는 라우팅된 방화벽 모드의 컨테이너 인스턴스 3개가 포함되어 있습니다. 이러한 컨테이너 인스턴스는 다음 인터페이스를 포함합니다.

- Management(관리) — 모든 인스턴스가 Port-Channel1 인터페이스(관리 유형)를 사용합니다. 이 EtherChannel에는 10기가비트 이더넷 인터페이스 2개가 포함됩니다. 각 애플리케이션 내에서 인터페이스는 동일한 관리 네트워크의 고유 IP 주소를 사용합니다.
- Inside(내부) — 각 인스턴스가 Port-Channel2(데이터 유형)의 하위 인터페이스를 사용합니다. 이 EtherChannel에는 10기가비트 이더넷 인터페이스 2개가 포함됩니다. 각 하위 인터페이스는 별도 네트워크에 있습니다.
- Outside(외부) — 모든 인스턴스가 Port-Channel3 인터페이스(데이터 공유 유형)를 사용합니다. 이 EtherChannel에는 10기가비트 이더넷 인터페이스 2개가 포함됩니다. 각 애플리케이션 내에서 인터페이스는 동일한 외부 네트워크의 고유 IP 주소를 사용합니다.
- Failover(페일오버) — 각 인스턴스가 Port-Channel4(데이터 유형)의 하위 인터페이스를 사용합니다. 이 EtherChannel에는 10기가비트 이더넷 인터페이스 2개가 포함됩니다. 각 하위 인터페이스는 별도 네트워크에 있습니다.



컨테이너 인스턴스 인터페이스용 자동 **MAC** 주소

FXOS 섀시는 컨테이너 인스턴스 인터페이스용 MAC 주소를 자동으로 생성하며 각 인스턴스의 공유 인터페이스가 고유한 MAC 주소를 사용하도록 보장합니다.

애플리케이션 내의 공유 인터페이스에 직접 MAC 주소를 할당하는 경우 직접 할당한 MAC 주소가 사용됩니다. 나중에 수동 MAC 주소를 삭제할 경우 자동 생성 주소가 사용됩니다. 드물지만, 생성된 MAC 주소가 네트워크의 다른 사설 MAC 주소와 충돌할 경우 애플리케이션 내에서 인터페이스의 MAC 주소를 직접 설정하는 것이 좋습니다.

자동 생성 주소는 A2로 시작하기 때문에, 주소가 겹칠 위험이 있으므로 수동 MAC 주소를 A2로 시작해서는 안 됩니다.

FXOS 섀시는 다음 형식을 사용하여 MAC 주소를 생성합니다.

A2*xx.yyzz.zzzz*

여기서 *xx.yy*는 사용자 정의 접두사 또는 시스템 정의 접두사이고 *zz.zzzz*는 섀시에서 생성되는 내부 카운터입니다. 시스템 정의 접두사는 IDPROM에 프로그래밍되는 번인된 MAC 주소 풀의 첫 번째 MAC 주소의 하위 2바이트와 일치합니다. MAC 주소 풀을 확인하려면 **connect fxos**, **show module**을 차례로 사용합니다. 예를 들어 모듈 1에 대해 표시되는 MAC 주소 범위가 b0aa.772f.f0b0~b0aa.772f.f0bf이면 시스템 접두사는 f0b0입니다.

사용자 정의 접두사는 16진수로 변환되는 정수입니다. 사용자 정의 접두사가 어떻게 사용되는지 예를 들어 설명하자면, 접두사를 77로 설정하는 경우 섀시에서는 77을 16진수 값 004D(*yyxx*)로 변환합니다. 접두사가 MAC 주소에서 쓰일 때는 섀시 기본 형식에 부합하도록 역전됩니다(*xxyy*).

A2**4D.00***zz.zzzz*

접두사가 1009 (03F1)일 때 MAC 주소는 다음과 같습니다.

A2**F1.03***zz.zzzz*

컨테이너 인스턴스 리소스 관리

컨테이너 인스턴스당 리소스 사용량을 지정하려면 FXOS에서 리소스 프로파일을 하나 이상 생성합니다. 논리적 디바이스/애플리케이션 인스턴스를 구축할 때 사용할 리소스 프로필을 지정합니다. 리소스 프로파일은 CPU 코어 수를 설정합니다. RAM은 코어 수에 따라 동적으로 할당되며 디스크 공간은 인스턴스당 40GB로 설정됩니다. 모델당 사용 가능한 리소스를 확인하려면 컨테이너 인스턴스의 요구 사항 및 사전 요구 사항, 218 페이지 섹션을 참조하십시오. 리소스 프로파일을 추가하려면 컨테이너 인스턴스에 대한 리소스 프로파일 추가, 160 페이지 섹션을 참조하십시오.

다중 인스턴스 기능의 성능 확장 요인

플랫폼의 최대 처리량(연결, VPN 세션 및 TLS 프록시 세션)은 네이티브 인스턴스의 메모리 및 CPU 사용에 대해 계산됩니다. 이 값은 **show resource usage**에 표시됩니다. 다중 인스턴스를 사용하는 경우 처리량은 인스턴스에 할당하는 CPU 코어의 비율을 기준으로 계산해야 합니다. 예를 들어, 코어가 50%인 컨테이너 인스턴스를 사용하는 경우, 처음에는 처리량의 50%를 계산해야 합니다. 또한, 컨테이너 인스턴스에 사용 가능한 처리량은 기본 인스턴스로 줄여야 합니다.

인스턴스의 처리량 계산에 대한 자세한 지침은 https://www.cisco.com/c/en/us/products/collateral/security/firewalls/white-paper-c11-744750.html의 내용을 참조하십시오.

컨테이너 인스턴스 및 고가용성

2개의 개별 섀시에서 컨테이너 인스턴스를 사용하여 고가용성을 사용할 수 있습니다. 예를 들어 각각 인스턴스가 10개인 섀시가 2개 있으면 고가용성 쌍 10개를 생성할 수 있습니다. FXOS에서 고가용성이 구성되지 않았으면 애플리케이션 관리자에서 각 고가용성 쌍을 구성합니다.

자세한 요구 사항은 고가용성 요구 사항 및 사전 요건, 217 페이지 및 고가용성 쌍 추가, 238 페이지의 내용을 참조하십시오.

컨테이너 인스턴스 및 클러스터링

보안 모듈/엔진당 하나의 컨테이너 인스턴스를 사용하여 컨테이너 인스턴스 클러스터를 생성할 수 있습니다. 자세한 요구 사항은 클러스터링의 요구 사항 및 사전 요구 사항, 213 페이지의 내용을 참조하십시오.

논리적 디바이스의 요구 사항 및 사전 요구 사항

요구 사항 및 사전 요구 사항에 대한 내용은 다음 섹션을 참조하십시오.

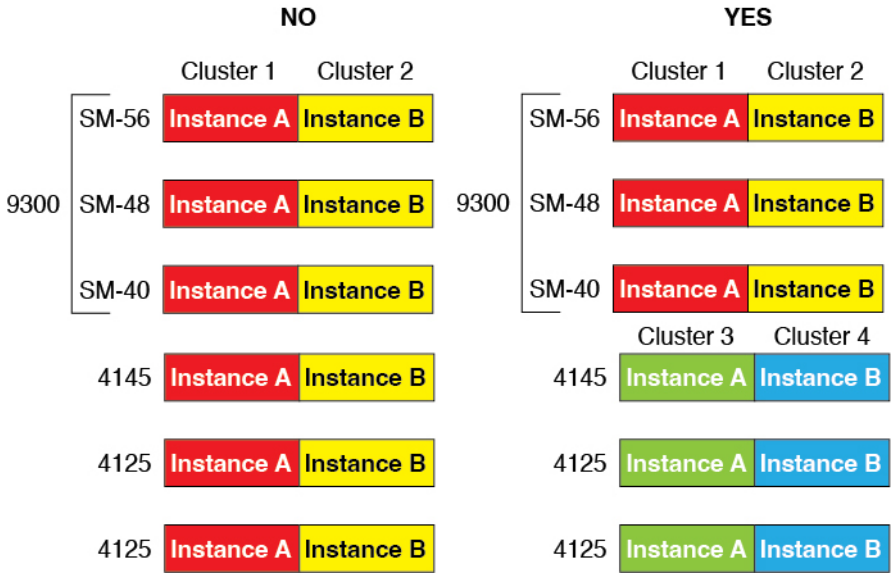
하드웨어 및 소프트웨어 조합에 대한 요구 사항 및 사전 요구 사항

Firepower 4100/9300에서는 여러 모델, 보안 모듈, 애플리케이션 유형, 고가용성 및 확장성 기능을 지원합니다. 허용되는 조합에 대한 다음과 같은 요건을 참조하십시오.

Firepower 9300 요건

Firepower 9300에는 3개의 보안 모듈 슬롯 및 여러 유형의 보안 모듈이 포함되어 있습니다. 다음 요건을 참조하십시오.

- 보안 모듈 유형 - Firepower 9300에 다양한 유형의 모듈을 설치할 수 있습니다. 예를 들어, SM-48을 모듈 1로, SM-40을 모듈 2로, SM-56를 모듈 3으로 설치할 수 있습니다.
- 기본 및 컨테이너 인스턴스 - 보안 모듈에 컨테이너 인스턴스를 설치하는 경우 해당 모듈에서는 다른 컨테이너 인스턴스만 지원할 수 있습니다. 기본 인스턴스에서는 모듈의 모든 리소스를 사용하므로 모듈에는 하나의 기본 인스턴스만 설치할 수 있습니다. 일부 모듈에서는 기본 인스턴스를, 다른 모듈에서는 컨테이너 인스턴스를 사용할 수 있습니다. 예를 들어 모듈 1 및 모듈 2에는 기본 인스턴스를 설치할 수 있지만, 모듈 3에는 컨테이너 인스턴스를 설치할 수 있습니다.
- 네이티브 인스턴스 클러스터링 - 클러스터의 모든 보안 모듈이 인트라 섀시(intra-chassis)든, 섀시 간(inter-chassis)이든 상관없이 동일한 유형이어야 합니다. 빈 슬롯을 포함하여 섀시에 있는 모든 모듈은 클러스터에 속해야 하지만 각 섀시에 설치된 보안 모듈의 수는 다를 수 있습니다. 예를 들어, 섀시 1에는 2개의 SM-40을 설치하고 섀시 2에는 3개의 SM-40을 설치할 수 있습니다. 동일한 섀시에 1개의 SM-48 및 2개의 SM-40을 설치하는 경우에는 클러스터링을 사용할 수 없습니다.
- 컨테이너 인스턴스 클러스터링 - 다양한 모델 유형에서 인스턴스를 사용하여 클러스터를 생성할 수 있습니다. 예를 들어 Firepower 9300 SM-56, SM-48, SM-40에서 인스턴스를 사용해 하나의 클러스터를 생성할 수 있습니다. 그러나 동일한 클러스터에서 Firepower 9300과 Firepower 4100을 혼합할 수는 없습니다.



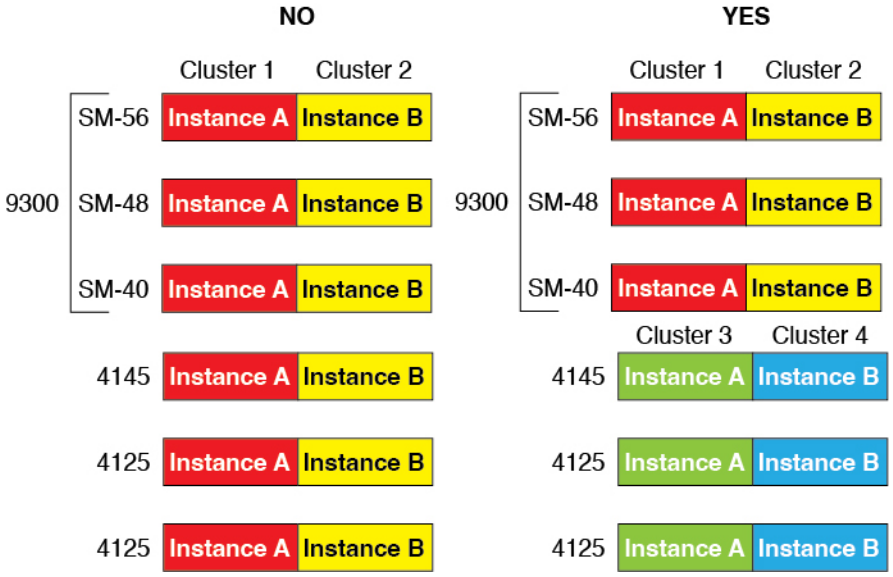
- 고가용성 - 고가용성은 Firepower 9300에서 동일한 유형의 모듈 간에만 지원됩니다. 그러나 두 섀시에는 혼합 모듈을 포함할 수 있습니다. 각 섀시에 SM-56, SM-48 및 SM-40이 있는 경우를 예로 들 수 있습니다. SM-40 모듈 간, SM-48 모듈 간, SM-56 모듈 간에 고가용성 쌍을 생성할 수 있습니다.

- ASA 및 FTD애플리케이션 유형 - 섀시의 개별 모듈에 서로 다른 애플리케이션 유형을 설치할 수 있습니다. 예를 들어, 모듈 1 및 모듈 2에는 ASA를 설치하고 모듈 3에는 FTD를 설치할 수 있습니다.
- ASA 또는 FTD 버전 - 애플리케이션 인스턴스 유형의 서로 다른 버전을 별도의 모듈에서 실행하거나 동일한 모듈에서 별도의 컨테이너 인스턴스로 실행할 수 있습니다. 예를 들어, 모듈 1에는 FTD 6.3을, 모듈 2에는 FTD 6.4를 설치하고, 모듈 3에는 FTD 6.5를 설치할 수 있습니다.

Firepower 4100 요건

Firepower 4100은 여러 모델로 제공됩니다. 다음 요건을 참조하십시오.

- 기본 및 컨테이너 인스턴스 - Firepower 4100에 컨테이너 인스턴스를 설치하는 경우 해당 디바이스에서는 다른 컨테이너 인스턴스만 지원할 수 있습니다. 기본 인스턴스에서는 디바이스의 모든 리소스를 사용하므로 디바이스에는 하나의 기본 인스턴스만 설치할 수 있습니다.
- 네이티브 인스턴스 클러스터링 - 클러스터의 모든 섀시는 동일한 모델이어야 합니다.
- 컨테이너 인스턴스 클러스터링 - 다양한 모델 유형에서 인스턴스를 사용하여 클러스터를 생성할 수 있습니다. 예를 들어 Firepower 4145 및 4125에서 인스턴스를 사용해 하나의 클러스터를 생성할 수 있습니다. 그러나 동일한 클러스터에서 Firepower 9300과 Firepower 4100을 혼합할 수는 없습니다.



- 고가용성 - 고가용성은 동일한 유형의 모듈 간에만 지원됩니다.
- ASA 및 FTD 애플리케이션 유형 - Firepower 4100에서는 하나의 애플리케이션 유형만 실행할 수 있습니다.
- FTD 컨테이너 인스턴스 버전 - 동일한 모듈에서 별도의 컨테이너 인스턴스로 서로 다른 버전의 FTD를 실행할 수 있습니다.

클러스터링의 요구 사항 및 사전 요구 사항

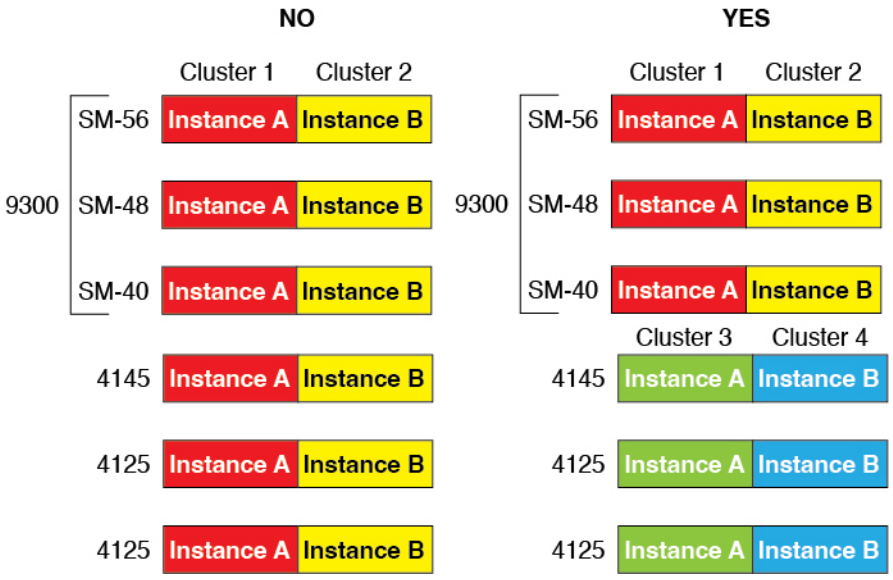
클러스터 모델 지원

- Firepower 9300의 ASA - 최대 16개 모듈 예를 들어 섀시 16개에 모듈 1개, 섀시 8개에 모듈 2개, 또는 모듈을 16개까지 제공하는 어떤 조합도 사용할 수 있습니다. 섀시의 모든 모듈은 클러스터에 속해야 합니다. 섀시 내, 섀시 간 및 사이트 간 클러스터링에 지원됨.
- ASA의 Firepower 4100 Series - 최대 16개 섀시. 섀시 간 및 사이트 간 클러스터링에 지원됨.
- FMC를 사용한 Firepower 9300의 FTD- 1 섀시에 최대 6개 모듈 예를 들어 섀시 3개에 모듈 2개, 섀시 2개에 모듈 3개, 또는 모듈을 6개까지 제공하는 어떤 조합도 사용할 수 있습니다. 섀시의 모든 모듈은 클러스터에 속해야 합니다. 섀시 내 및 섀시 간 클러스터링에 지원됨.
- FMC를 사용한 Firepower 4100 Series의 FTD - 최대 6개 섀시 섀시 간 클러스터링에 지원됨.
- Radware DefensePro- ASA와의 섀시 내 클러스터링에 지원됨.
- Radware DefensePro - FTD와의 섀시 내 클러스터링에 지원됨. 다중 인스턴스 클러스터링을 지원하지 않습니다.

클러스터링 하드웨어 및 소프트웨어 요구 사항

클러스터의 모든 섀시:

- 네이티브 인스턴스 클러스터링 - Firepower 4100의 경우 모든 섀시가 동일한 모델이어야 합니다. Firepower 9300의 경우: 모든 보안 모듈이 동일한 유형이어야 합니다. 예를 들어 클러스터링을 사용하는 경우 Firepower 9300의 모든 모듈은 SM-40이어야 합니다. 빈 슬롯을 포함하여 섀시에 있는 모든 모듈은 클러스터에 속해야 하지만 각 섀시에 설치된 보안 모듈의 수는 다를 수 있습니다.
- 컨테이너 인스턴스 클러스터링 - 각 클러스터 인스턴스에 동일한 보안 모듈 또는 섀시 모델을 사용하는 것이 좋습니다. 그러나 Firepower 9300 보안 모듈 유형이나 Firepower 4100 모델에서는 필요한 경우 동일한 클러스터에서 다른 컨테이너 인스턴스를 혼용해 사용할 수 있습니다. 동일한 클러스터에서 Firepower 9300 및 4100 인스턴스를 혼용할 수 없습니다. 예를 들어 Firepower 9300 SM-56, SM-48, SM-40에서 인스턴스를 사용해 하나의 클러스터를 생성할 수 있습니다. 또는 Firepower 4145 및 4125에서 클러스터를 생성할 수 있습니다.



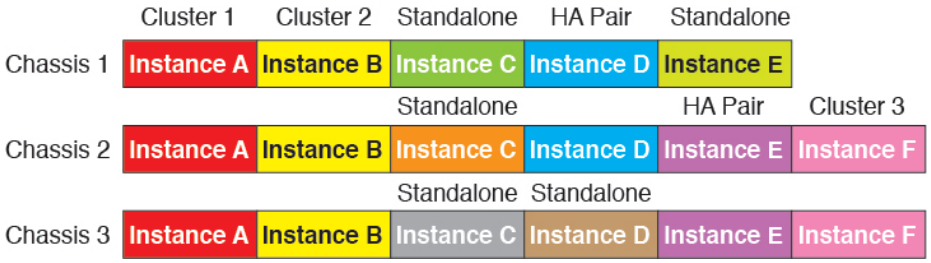
- 이미지 업그레이드 시 동일한 FXOS 소프트웨어 예외를 실행해야 합니다.
- 클러스터에 할당하는 인터페이스에 대한 것과 동일한 인터페이스 구성을 포함해야 합니다(예: EtherChannel, 활성 인터페이스, 속도 및 이중 등). 동일한 인터페이스 ID에 대해 용량이 일치하고 동일한 Spanned EtherChannel에서 성공적인 인터넷 번들링이 가능한 한 섀시에서 서로 다른 네트워크 모듈 유형을 사용할 수 있습니다. 모든 데이터 인터페이스는 섀시 간 클러스터링에서 EtherChannel이어야 합니다. 인터페이스 모듈을 추가 또는 제거하거나 EtherChannel을 구성하는 등의 방법을 통해 클러스터링을 활성화한 후 FXOS에서 인터페이스를 변경하는 경우에는 각 섀시에서 데이터 노드부터 시작하여 마지막으로 제어 노드까지 같은 변경을 수행합니다.
- 동일한 NTP 서버를 사용해야 합니다. FTD의 경우 FMC는 동일한 NTP 서버를 사용해야 합니다. 시간을 수동으로 설정해서는 안 됩니다.
- ASA: 각 FXOS 섀시를 License Authority 또는 Satellite Server에 등록해야 합니다. 데이터 노드에 대한 추가 비용은 없습니다. 영구 라이선스를 예약하려면 각 섀시용으로 별도의 라이선스를 구매해야 합니다. FTD의 경우 모든 라이선싱이 FMC에서 처리됩니다.

멀티 인스턴스 클러스터링 요구 사항

- 모든 내부 보안 모듈/엔진 클러스터링 안 함 - 지정된 클러스터에 대해 보안 모듈/엔진당 단일 컨테이너 인스턴스만 사용할 수 있습니다. 동일한 모듈에서 실행 중인 경우에는 두 컨테이너 인스턴스를 동일한 클러스터에 추가할 수 없습니다.

- 클러스터 및 독립형 인스턴스를 혼용 - 보안 모듈/엔진의 모든 컨테이너 인스턴스가 하나의 클러스터에 속할 필요가 없습니다. 일부 인스턴스는 독립형이나 고가용성 노드로 사용할 수 있습니

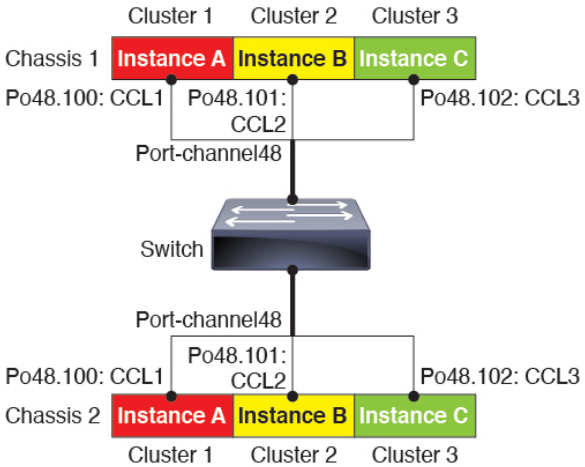
다. 동일한 보안 모듈/엔진에서 별도의 인스턴스를 사용해 여러 클러스터를 생성할 수도 있습니다.



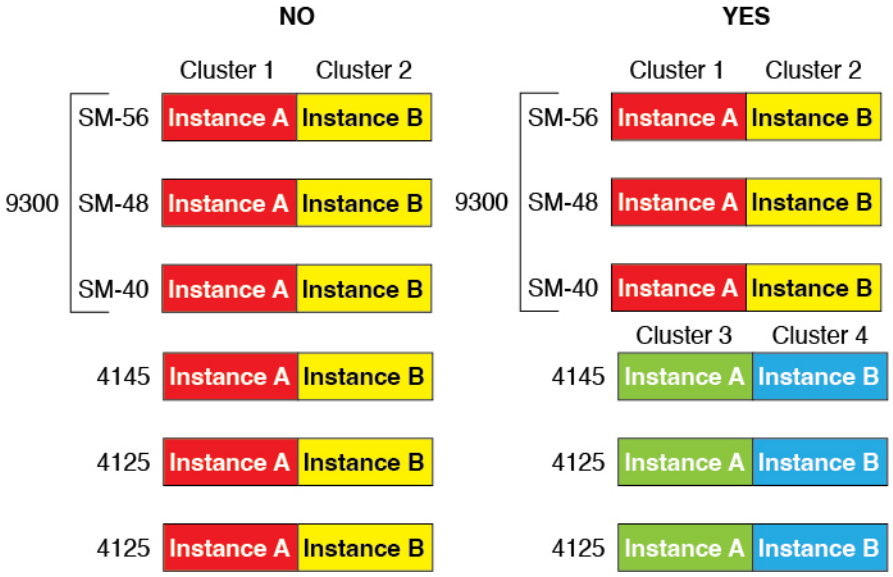
- Firepower 9300의 모든 모듈 세 가지는 해당 클러스터에 속해야 합니다 - Firepower 9300의 경우 클러스터에는 모든 3개의 모듈에서 단일 컨테이너 인스턴스가 필요합니다. 모듈 1 및 2의 인스턴스를 사용하여 클러스터를 생성한 다음 모듈 3 또는 예제에서 네이티브 인스턴스를 사용할 수 없습니다.

- 리소스 프로파일 일치 - 클러스터의 각 노드가 동일한 리소스 프로파일 특성을 사용하는 것이 좋습니다. 그러나 클러스터 노드를 다른 리소스 프로파일로 변경하거나 다른 모델을 사용하는 경우 일치하지 않는 리소스가 허용됩니다.

- 전용 클러스터 제어 링크 - 섀시 간 클러스터링의 경우 각 클러스터에 전용 클러스터 제어 링크가 필요합니다. 예를 들어 각 클러스터는 동일한 클러스터 유형 EtherChannel에서 별도의 하위 인터페이스를 사용하거나 별도의 EtherChannel을 사용할 수 있습니다.



- 공유 인터페이스 없음 - 클러스터링에서 공유 유형 인터페이스를 지원하지 않습니다. 그러나 동일한 관리 및 이벤트 인터페이스는 여러 클러스터에서 사용할 수 있습니다.
- 하위 인터페이스 없음 - 다중 인스턴스 클러스터는 FXOS 정의 VLAN 하위 인터페이스를 사용할 수 없습니다. 클러스터 EtherChannel의 하위 인터페이스를 사용할 수 있는 클러스터 제어 링크는 예외입니다.
- 섀시 모델 혼합 - 각 클러스터 인스턴스에 동일한 보안 모듈 또는 섀시 모델을 사용하는 것이 좋습니다. 그러나 Firepower 9300 보안 모듈 유형이나 Firepower 4100 모델에서는 필요한 경우 동일한 클러스터에서 다른 컨테이너 인스턴스를 혼용해 사용할 수 있습니다. 동일한 클러스터에서 Firepower 9300 및 4100 인스턴스를 혼용할 수 없습니다. 예를 들어 Firepower 9300 SM-56, SM-48, SM-40에서 인스턴스를 사용해 하나의 클러스터를 생성할 수 있습니다. 또는 Firepower 4145 및 4125에서 클러스터를 생성할 수 있습니다.



- 최대 6개 노드 - 하나의 클러스터에서 최대 6개의 컨테이너 인스턴스를 사용할 수 있습니다.

섀시 간 클러스터링을 위한 스위치 요구 사항

- Firepower 4100/9300 섀시에서 클러스터링을 구성하기 전에 스위치 구성을 완료하고 섀시의 모든 EtherChannel을 스위치에 성공적으로 연결하십시오.
- 지원되는 스위치 특성은 Cisco FXOS 호환성을 참고하십시오.

사이트 간 클러스터링을 위한 **Data Center Interconnect** 크기 조정

클러스터 제어 링크 트래픽을 처리하기 위한 DCI(data center interconnect) 대역폭을 다음 계산과 같이 예약해야 합니다.

$$\frac{\text{\# of cluster members per site}}{2} \times \text{cluster control link size per member}$$

각 사이트의 멤버 수가 다를 경우, 더 큰 숫자를 계산에 사용합니다. DCI의 최소 대역폭은 한 멤버에 대한 클러스터 제어 링크의 크기보다 작으면 안 됩니다.

예를 들면 다음과 같습니다.

- 2개 사이트에 멤버가 4개인 경우:
  - 총 클러스터 멤버 4개
  - 각 사이트당 멤버 2개
  - 멤버당 5Gbps 클러스터 제어 링크

  예약된 DCI 대역폭 = 5Gbps(2/2 x 5Gbps)
- 3개 사이트에 멤버가 6개인 경우 크기가 다음과 같이 증가함:
  - 총 클러스터 멤버 6개
  - 사이트 1에 멤버 3개, 사이트 2에 멤버 2개, 사이트 3에 멤버 1개
  - 멤버당 10Gbps 클러스터 제어 링크

  예약된 DCI 대역폭 = 15Gbps(3/2 x 10Gbps)
- 2개 사이트에 멤버가 2개인 경우:
  - 총 클러스터 멤버 2개
  - 사이트당 멤버 1개
  - 멤버당 10Gbps 클러스터 제어 링크

  예약된 DCI 대역폭 = 10Gbps(1/2 x 10Gbps = 5Gbps). 그러나 최소 대역폭은 클러스터 제어 링크의 크기(10Gbps)보다 작으면 안 됩니다.

고가용성 요구 사항 및 사전 요건

- 고가용성 페일오버 설정에는 2개의 유닛이 필요합니다.
  - 별도의 섀시에 있어야 합니다. Firepower 9300용 섀시 내 고가용성은 지원되지 않습니다.
  - 같은 모델이어야 합니다.
  - 고가용성 논리 디바이스에는 동일한 인터페이스가 할당되어야 합니다.
  - 인터페이스 개수와 유형이 같아야 합니다. 고가용성을 활성화하기 전에 모든 인터페이스는 FXOS와 동일하게 사전 설정되어야 합니다.
- 고가용성은 Firepower 9300에서 동일한 유형의 모듈 간에만 지원되지만, 두 섀시는 혼합된 모듈을 포함할 수 있습니다. 각 섀시에 SM-56, SM-48 및 SM-40이 있는 경우를 예로 들 수 있습니다. SM-56 모듈 간, SM-48 모듈 간, SM-40 모듈 간에 고가용성 쌍을 생성할 수 있습니다.

- 컨테이너 인스턴스의 각 유닛은 동일한 리소스 프로파일 속성을 사용해야 합니다.
- 기타 고가용성을 위한 시스템 요구 사항은 고가용성을 위한 애플리케이션 구성 가이드 장의 내용을 참조하십시오.

컨테이너 인스턴스의 요구 사항 및 사전 요구 사항

지원되는 애플리케이션 유형

- FMC을 사용한 FTD

모델당 최대 컨테이너 인스턴스 및 리소스

각 컨테이너 인스턴스에 대해 인스턴스에 할당할 CPU 코어의 수를 지정할 수 있습니다. 코어 수에 따라 RAM은 동적으로 할당되며 디스크 공간은 인스턴스당 40GB로 설정됩니다.

표 ***16:*** 모델당 최대 컨테이너 인스턴스 및 리소스

모델	최대 컨테이너 인스턴스 수	사용 가능한 **CPU** 코어	사용 가능한 **RAM**	사용 가능한 디스크 공간
Firepower 4110	3	22	53GB	125.6GB
Firepower 4112	3	22	78GB	308GB
Firepower 4115	7	46	162GB	308GB
Firepower 4120	3	46	101GB	125.6GB
Firepower 4125	10	62	162GB	644GB
Firepower 4140	7	70	222GB	311.8GB
Firepower 4145	14	86	344GB	608GB
Firepower 4150	7	86	222GB	311.8GB
Firepower 9300 SM-24 보안 모듈	7	46	226GB	656.4GB
Firepower 9300 SM-36 보안 모듈	11	70	222GB	640.4GB
Firepower 9300 SM-40 보안 모듈	13	78	334GB	1359GB
Firepower 9300 SM-44 보안 모듈	14	86	218GB	628.4GB
Firepower 9300 SM-48 보안 모듈	15	94	334GB	1341GB
Firepower 9300 SM-56 보안 모듈	18	110	334GB	1314GB

FMC 필수조건

Firepower 4100 섀시 또는 Firepower 9300 모듈의 모든 인스턴스에서는 라이선싱 구현으로 인해 동일한 FMC를 사용해야 합니다.

논리적 디바이스 관련 지침 및 제한 사항

지침 및 제한 사항은 다음 섹션을 참조하십시오.

일반 지침 및 제한 사항

방화벽 모드

FTD 및 ASA의 부트스트랩 구성에서 방화벽 모드를 라우팅 또는 투명으로 설정할 수 있습니다.

고가용성

- 애플리케이션 구성 내에서 고가용성을 구성합니다.
- 모든 데이터 인터페이스를 페일오버 및 상태 링크로 사용할 수 있습니다. 데이터 공유 인터페이스가 지원되지 않습니다.

다중 인스턴스및컨텍스트 모드

- 다중 상황 모드는 ASA에서만 지원됩니다.
- 구축 후에 ASA에서 다중 컨텍스트 모드를 활성화합니다.
- 컨테이너 인스턴스와의 다중 인스턴스 기능은 FMC를 사용하는 FTD에서만 사용 가능합니다.
- FTD 컨테이너 인스턴스의 경우에는 단일 FMC에서 보안 모듈/엔진의 모든 인스턴스를 관리해야 합니다.
- 의 TLS 암호화 가속에서 최대 16 개의 컨테이너 인스턴스를 활성화할 수 있습니다.
- FTD 컨테이너 인스턴스의 경우에는 다음 기능이 지원되지 않습니다.
  - Radware DefensePro 링크 데코레이터
  - FMC UCAPL/CC 모드
  - 하드웨어로의 플로우 오프로드

클러스터링 지침 및 제한 사항

새시 간 클러스터링을 위한 스위치

- 연결된 스위치가 클러스터 데이터 인터페이스 및 클러스터 제어 링크 인터페이스 모두의 MTU와 일치해야 합니다. 클러스터 제어 링크 인터페이스 MTU를 데이터 인터페이스 MTU보다 100바이트 이상 높게 설정해야 하므로 스위치를 연결하는 클러스터 제어 링크를 적절하게 설정해야 합니다. 클러스터 제어 링크 트래픽에는 데이터 패킷 전달이 포함되므로 클러스터 제어 링크는 데이터 패킷의 전체 크기와 클러스터 트래픽 오버헤드를 모두 수용해야 합니다.
- Cisco IOS XR 시스템의 경우 기본이 아닌 MTU를 설정하려면 IOS 인터페이스 MTU를 클러스터 디바이스 MTU보다 14바이트 높게 설정합니다. 그렇지 않으면, **mtu-ignore** 옵션을 사용하지 않는 경우 OSPF 인접 피어링 시도에 실패할 수 있습니다. 클러스터 디바이스 MTU는 IOS XR *IPv4* MTU와 일치해야 합니다. Cisco Catalyst 및 Cisco Nexus 스위치에는 이 조정이 필요하지 않습니다.
- 클러스터 제어 링크 인터페이스용 스위치의 경우, 클러스터 유닛에 연결된 스위치 포트에서 Spanning Tree PortFast를 사용하도록 선택하여 새 유닛에 대한 참가 프로세스 속도를 높일 수 있습니다.
- 스위치에서는 **source-dest-ip**또는 **source-dest-ip-port** EtherChannel 로드 밸런싱 알고리즘 중 하나를 사용하는 것이 좋습니다(Cisco Nexus OS 및 Cisco IOS **port-channel load-balance** 명령 참조). 로드 밸런싱 알고리즘에서는 **vlan** 키워드를 사용하지 마십시오. 이렇게 할 경우 클러스터의 디바이스에 트래픽이 균일하지 않게 분산될 수 있습니다.
- 스위치에서 EtherChannel의 로드 밸런싱 알고리즘을 변경할 경우, 스위치의 EtherChannel 인터페이스에서 트래픽 전달이 일시적으로 중단되며 Spanning Tree Protocol이 재시작됩니다. 트래픽에서 흐름을 다시 시작하기 전까지 지연이 발생하게 됩니다.
- 일부 스위치에서는 LACP를 통한 동적 포트 우선순위를 지원하지 않습니다(활성 및 스탠바이 링크). 동적 포트 우선순위를 비활성화하여 Spanned EtherChannel과의 호환성을 향상할 수 있습니다.
- 클러스터 제어 링크 경로의 스위치에서는 L4 체크섬을 확인하지 않습니다. 클러스터 제어 링크를 통해 리디렉션된 트래픽에는 올바른 L4 체크섬이 없습니다. L4 체크섬을 확인하는 스위치의 경우 트래픽이 감소하는 결과를 초래할 수 있습니다.
- 포트 채널 번들링 다운타임은 구성된 keepalive 기간을 초과하면 안 됩니다.
- Supervisor 2T EtherChannel에서 기본 해시 분산 알고리즘은 적응형입니다. VSS 설계에서 비대칭 트래픽을 방지하려면 클러스터 디바이스에 연결된 포트 채널의 해시 알고리즘을 다음과 같이 변경하여 수정합니다.

  router(config) # **port-channel** *id* **hash-distribution fixed**

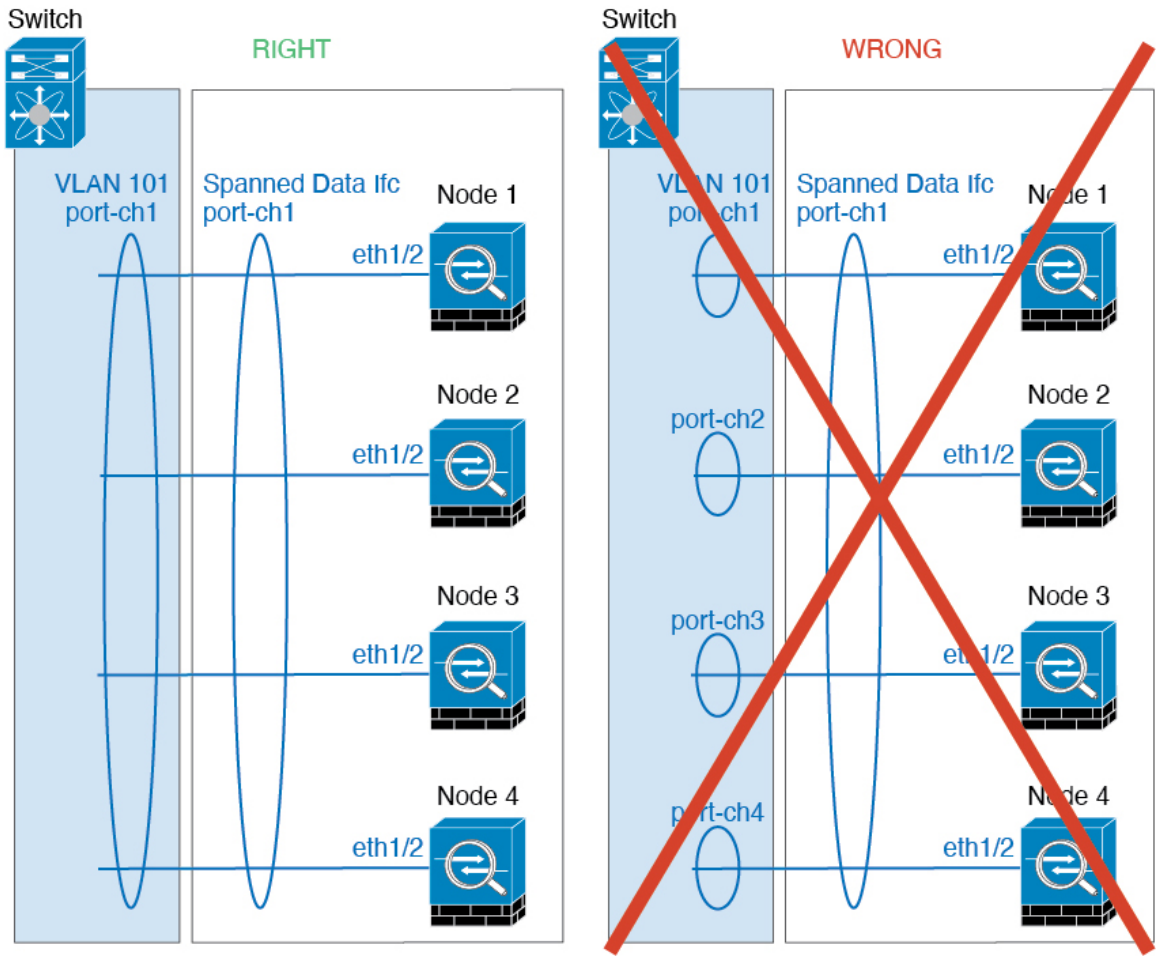
  VSS 피어 링크의 적응형 알고리즘을 활용할 때가 있을 수 있으므로 알고리즘을 전역으로 변경하지 마십시오.

- Firepower 4100/9300 클러스터는 LACP 단계적 통합을 지원합니다. 따라서 연결된 Cisco Nexus 스위치에서 LACP 단계적 통합을 활성화된 상태로 둘 수 있습니다.

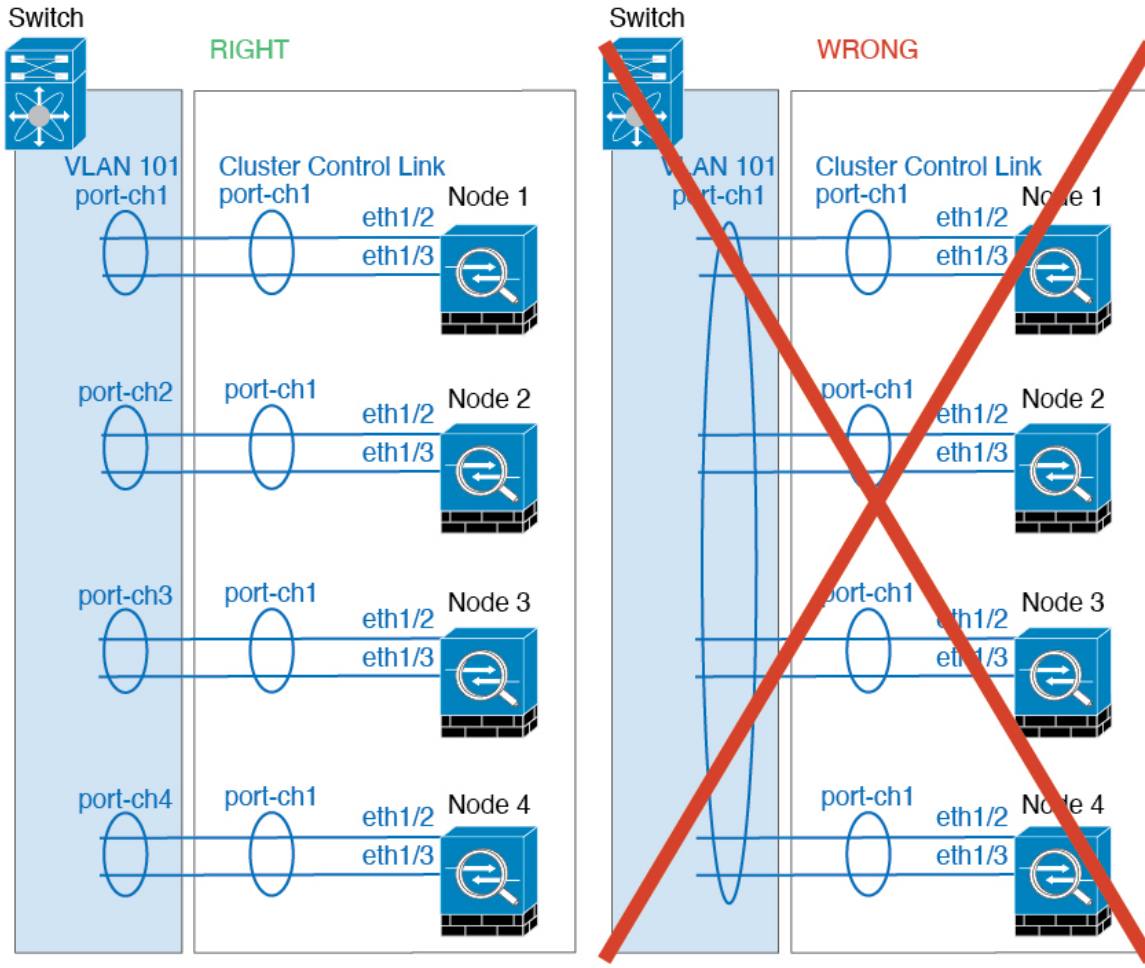
- 스위치에서 Spanned EtherChannel의 번들링 속도가 저하될 경우, 스위치의 개별 인터페이스에 대한 LACP 속도를 빠르게 설정할 수 있습니다. FXOS EtherChannel에서는 기본적으로 LACP 속도가 fast(고속)로 설정됩니다. Nexus Series와 같은 일부 스위치는 ISSU(In-Service Software Upgrade) 수행 시 고속 LACP를 지원하지 않으므로 클러스터링에서는 ISSU를 사용하지 않는 것이 좋습니다.

섀시 간 클러스터링을 위한 **EtherChannel**

- 15.1(1)S2 이전 Catalyst 3750-X Cisco IOS 소프트웨어 버전에서는 클러스터 유닛에서 EtherChannel과 스위치 스택 간 연결을 지원하지 않았습니다. 기본 스위치 설정으로 클러스터 유닛 EtherChannel이 교차 스택에 연결되어 있는 상태에서 제어 유닛 스위치의 전원이 꺼질 경우, 나머지 스위치에 연결된 EtherChannel은 가동되지 않습니다. 호환성을 개선하려면 **stack-mac persistent timer** 명령을 다시 로드 시간을 고려하여 충분히 큰 값으로 설정합니다(예: 8분 또는 무한인 경우 0). 또는 15.1(1)S2 같은 더 안정적인 스위치 소프트웨어 버전으로 업그레이드할 수 있습니다.
- Spanned EtherChannel 구성과 디바이스-로컬 EtherChannel 구성 — Spanned EtherChannel과 디바이스-로컬 EtherChannel에서 각각 알맞게 스위치를 구성해야 합니다.
  - Spanned EtherChannel — 클러스터의 모든 멤버 전체를 포괄하는 클러스터 유닛 스팬 EtherChannels의 경우, 인터페이스가 스위치의 단일 EtherChannel에 통합됩니다. 각 인터페이스가 스위치의 동일한 채널 그룹에 있는지 확인하십시오.



- 디바이스-로컬 EtherChannel - 클러스터 제어 링크에 대해 구성된 모든 EtherChannel을 비롯한 클러스터 유닛 디바이스-로컬 EtherChannel의 경우 스위치에서 별도의 EtherChannel을 구성해야 합니다. 여러 클러스터 유닛 EtherChannel을 스위치에서 하나의 EtherChannel에 통합하지 마십시오.



사이트 간 클러스터링

사이트 간 클러스터링에 대한 다음 지침을 참조하십시오.

- 클러스터 제어 링크 레이턴시는 RTT(왕복 시간)가 20ms 이하여야 합니다.
- 클러스터 제어 링크는 오류가 나거나 폐기된 패킷이 없는 안정적인 상태여야 합니다. 예를 들어, 전용 링크를 사용해야 합니다.
- 연결 리밸런싱을 구성하지 마십시오. 이렇게 할 경우 다른 사이트의 클러스터 멤버에 연결이 리밸런싱됩니다.
- 는 DCCI(Data Center Interconnect)에서 사용되는 경우에도 전용 링크이므로 클러스터 제어 링크에서 전달된 데이터 트래픽을 암호화하지 않습니다. OTV(Overlay Transport Virtualization)를 사용하거나 로컬 제어 도메인 외부에서 클러스터 제어 링크를 확장하는 경우 OTV를 통한 802.1AE MacSec과 같은 보더 라우터에서 암호화를 구성할 수 있습니다.
- 클러스터를 구현할 경우 들어오는 연결에 대한 여러 사이트에 있는 멤버가 구분되지 않습니다. 따라서 하나의 특정한 연결의 연결 역할은 사이트 전체를 포괄하게 될 수 있습니다. 이는 정상적

인 동작입니다. 그러나 관리자 지역화를 활성화하는 경우 항상 연결 소유자와 동일한 사이트에서 로컬 관리자 역할이 선택됩니다(사이트 ID에 따라). 원래 소유자가 실패하면 로컬 관리자는 동일한 사이트에서 새 소유자를 선택합니다.(참고: 트래픽이 사이트 간에 비동기 상태이고 원래 소유자가 실패한 후 원격 사이트로부터 계속 트래픽이 발생하면, 원격 사이트의 노드가 재호스팅 기간 내에 데이터 패킷을 수신하는 경우 새로운 소유자가 될 수 있습니다.)

- 관리자 지역화의 경우 NAT 또는 PAT 트래픽, SCTP에서 검사된 트래픽, 단편화 소유자 쿼리 등의 트래픽 유형은 지역화를 지원하지 않습니다.
- 투명 모드에서, 클러스터가 내부 및 외부 라우터(north-south 삽입이라고도 함) 쌍 사이에 위치하면 내부 라우터 모두에서 MAC 주소를 공유해야 하며 외부 라우터 모두에서도 MAC 주소를 공유해야 합니다. 사이트 1의 클러스터 멤버가 사이트 2의 멤버에 연결을 전달할 경우, 목적지 MAC 주소가 유지됩니다. MAC 주소가 사이트 1의 라우터와 동일할 경우 패킷은 사이트 2의 라우터에만 도달합니다.
- 투명 모드에서 클러스터가 내부 네트워크(East-West 삽입이라고 함) 사이에서 방화벽을 위해 각 사이트에서 데이터 네트워크 및 게이트웨이 라우터 사이에 위치하면 각 게이트웨이 라우터는 HSRP와 같은 첫 번째 홉 이중화 프로토콜(FHRP)을 사용하여 각 사이트에서 동일한 가상 IP 및 MAC 주소 대상을 제공해야 합니다. 데이터 VLAN은 OTV(오버레이 전송 가상화) 또는 유사한 기능을 사용하는 사이트 전체로 확장됩니다. DCI를 통해 다른 사이트로 전송 중인 로컬 게이트웨이 라우터에 예약된 트래픽을 방지하려면 필터를 생성해야 합니다. 게이트웨이 라우터가 1개의 사이트에 연결할 수 없게 되면, 모든 필터를 제거해야 트래픽이 성공적으로 다른 사이트의 게이트웨이에 연결할 수 있습니다.
- 투명 모드의 경우, 클러스터가 HSRP 라우터에 연결된 경우 라우터 HSRP MAC 주소를 . 인접 라우터가 HSRP를 사용하는 경우, HSRP IP 주소로 향하는 트래픽은 HSRP MAC 주소로 전송되지만, 반환 트래픽은 HSRP 쌍에 있는 특정 라우터 인터페이스의 MAC 주소에서 제공됩니다. 따라서 MAC 주소 테이블은 일반적으로 HSRP IP 주소에 대한 ARP 테이블 항목이 만료되고 가 ARP 요청을 보내고 응답을 수신하는 경우에만 업데이트됩니다. 의 ARP 테이블 항목은 기본적으로 14,400초 후에 만료되지만 MAC 주소 테이블 항목은 기본적으로 300초 후에 만료되므로 MAC 주소 테이블 만료 트래픽 삭제를 방지하려면 고정 MAC 주소 항목이 필요합니다.
- Spanned EtherChannel을 사용하는 라우팅 모드의 경우 사이트별 MAC 주소를 구성하십시오. OTV 또는 유사한 것을 사용하여 사이트 전체로 데이터 VLAN을 확장하십시오. 전역 MAC 주소로 향하는 트래픽이 DCI를 통해 다른 사이트에 가지 않도록 필터를 생성해야 합니다. 어떤 사이트에서 클러스터가 연결할 수 없게 되면 트래픽이 다른 사이트의 클러스터 노드에 성공적으로 도달할 수 있도록 모든 필터를 제거해야 합니다. 사이트 간 클러스터가 확장 세그먼트의 FHR(First Hop Router)로 작동하는 경우에는 동적 라우팅이 지원되지 않습니다.

추가 지침

- 기존 클러스터에 유닛을 추가하거나 유닛을 다시 로드할 경우, 일시적이고 제한적으로 패킷/연결이 감소하며 이는 정상적인 동작입니다. 경우에 따라 감소된 패킷으로 인해 연결이 끊어질 수 있습니다. 예를 들어, FTP 연결의 FIN/ACK 패킷이 감소할 경우 FTP 클라이언트가 끊어집니다. 이 경우 FTP 연결을 다시 설정해야 합니다.
- Spanned EtherChannel 인터페이스에 연결된 Windows 2003 서버를 사용할 경우 syslog 서버 포트가 중지되면 서버에서 ICMP 오류 메시지를 제한하지 않아 대량의 ICMP 메시지가 클러스터에

다시 전송됩니다. 이러한 메시지로 인해 클러스터의 일부 유닛에서 CPU 점유율이 높아져 성능에 영향을 미칠 수 있습니다. 이러한 문제를 방지하려면 ICMP 오류 메시지를 제한하는 것이 좋습니다.

- 이중화를 위해 EtherChannel을 VSS, vPC, StackWise 또는 StackWise Virtual에 연결하는 것이 좋습니다.
- 섀시 내에서 일부 보안 모듈을 클러스터하여 독립형 모드에서 다른 보안 모듈을 실행할 수 없습니다. 클러스터에 모든 보안 모듈을 포함해야 합니다.
- 암호 해독된 TLS/SSL 연결의 경우, 암호 해독 상태가 동기화되지 않습니다. 연결 소유자 장애가 발생하는 경우, 암호 해독된 연결이 재설정됩니다. 새 유닛에 새 연결을 설정해야 합니다. 암호 해독되지 않은 연결(암호 해독 안 함 규칙과 일치하는 연결)은 영향을 받지 않으며, 올바르게 복제됩니다.

기본값

- 클러스터 상태 검사 기능은 기본적으로 활성화되어 있으며 3초간의 대기 시간이 있습니다. 인터페이스 상태 모니터링은 모든 인터페이스에서 기본적으로 활성화됩니다.
- 실패한 클러스터 제어 링크에 대한 클러스터 자동 다시 참가 기능은 5분마다 무제한으로 시도하도록 설정됩니다.
- 실패한 데이터 인터페이스에 대한 클러스터 자동 다시 참가 기능은 간격이 2로 늘어 5분마다 3번 시도하도록 설정됩니다.
- 5초 연결 복제 지연은 HTTP 트래픽에 대해 기본적으로 활성화되어 있습니다.

독립형 논리적 디바이스 추가

단독으로 또는 고가용성 유닛으로 독립형 논리적 디바이스를 사용할 수 있습니다. 고가용성 사용량에 대한 자세한 내용은 고가용성 쌍 추가, 238 페이지 섹션을 참조하십시오.

독립형 **ASA** 추가

독립형 논리적 디바이스는 단독으로 작동하거나 고가용성 쌍으로 작동합니다. 보안 모듈이 여러 개인 Firepower 9300에서는 클러스터 또는 독립형 디바이스를 구축할 수 있습니다. 클러스터는 모든 모듈을 사용해야 하므로 모듈이 2개인 클러스터와 단일 독립형 디바이스를 혼용하는 방식은 사용할 수 없습니다.

Firepower 4100/9300 섀시에서 라우팅된 방화벽 모드 또는 투명 방화벽 모드 ASA를 구축할 수 있습니다.

다중 컨텍스트 모드의 경우 먼저 논리적 디바이스를 구축한 다음 ASA 애플리케이션에서 다중 컨텍스트 모드를 활성화해야 합니다.

시작하기 전에

- Cisco.com에서 논리적 디바이스에 사용할 애플리케이션 이미지를 다운로드한 다음 해당 이미지를 Firepower 4100/9300 섀시.

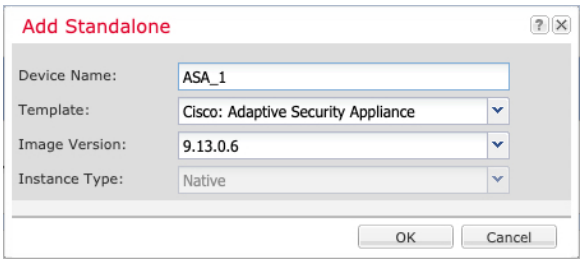
참고 Firepower 9300의 경우 섀시 내의 개별 모듈에 서로 다른 애플리케이션 유형(ASA 및 FTD)을 설치할 수 있습니다. 개별 모듈에서 애플리케이션 인스턴스 유형의 서로 다른 버전을 실행할 수도 있습니다.

- 논리적 디바이스에 사용할 관리 인터페이스를 구성합니다. 관리 인터페이스는 필수 항목입니다. 이 관리 인터페이스는 섀시 관리용으로만 사용되는 섀시 관리 포트(**Interfaces(**인터페이스**)** 탭 상단에 **MGMT(**관리**)**로 표시됨)와는 다릅니다.
- 다음 정보를 수집합니다.
  - 이 디바이스의 인터페이스 ID
  - 관리 인터페이스 IP 주소 및 네트워크 마스크
  - 게이트웨이 IP 주소

프로시저

단계 **1** **Logical Devices(**논리적 디바이스**)**를 선택합니다.

단계 **2** **Add(**추가**)** > **Standalone(**독립형**)**를 클릭하고 다음 파라미터를 설정합니다.



a) **Device Name(**디바이스 이름**)**을 입력합니다.

이 이름은 섀시 수퍼바이저가 관리 설정을 구성하고 인터페이스를 할당하는 데 사용됩니다. 이는 애플리케이션 구성에 사용되는 디바이스 이름이 아닙니다.

b) **Template(**템플릿**)**은 **Cisco: Adaptive Security Appliance**를 선택합니다.

c) **Image Version(**이미지 버전**)**을 선택합니다.

d) **OK(**확인**)**를 클릭합니다.

Provisioning - *device name*(프로비저닝 - 디바이스 이름) 창이 표시됩니다.

단계 **3** **Data Ports(**데이터 포트**)** 영역을 확장하고 디바이스에 할당할 각 포트를 클릭합니다.

이전에 **Interfaces(**인터페이스**)** 페이지에서 활성화한 데이터 인터페이스만 할당할 수 있습니다. 나중에 IP 주소 설정을 비롯하여 ASA에서 이러한 인터페이스를 활성화하고 구성하게 됩니다.

단계 **4** 화면 중앙의 디바이스 아이콘을 클릭합니다.

초기 부트스트랩 설정을 구성할 수 있는 대화 상자가 표시됩니다. 이러한 설정은 초기 구축 전용 또는 재해 복구용입니다. 일반 작업 시에는 애플리케이션 CLI 구성에서 대부분의 값을 나중에 변경할 수 있습니다.

단계 **5** **General Information(**일반 정보**)** 페이지에서 다음 작업을 수행합니다.

a) (Firepower 9300의 경우) **Security Module Selection(**보안 모듈 선택**)** 아래에서 이 논리적 디바이스에 사용할 보안 모듈을 클릭합니다.

b) **Management Interface(**관리 인터페이스**)**를 선택합니다.

이 인터페이스는 논리적 디바이스를 관리하는 데 사용됩니다. 이 인터페이스는 섀시 관리 포트와 별개입니다.

c) 관리 인터페이스 **Address Type(**주소 유형**)**을 **IPv4 only(IPv4** 전용**)**, **IPv6 only(IPv6** 전용**)** 또는 **IPv4 and IPv6(IPv4** 및 **IPv6)** 중에서 선택합니다.

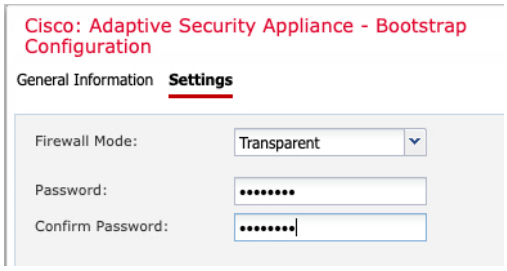
d) **Management IP(**관리 **IP)** 주소를 구성합니다.

이 인터페이스의 고유 IP 주소를 설정합니다.

e) **Network Mask(**네트워크 마스크**)** 또는 **Prefix Length(**접두사 길이**)**를 입력합니다.

f) **Network Gateway(**네트워크 게이트웨이**)** 주소를 입력합니다.

단계 **6** **Settings(**설정**)** 탭을 클릭합니다.



단계 **7** **Firewall Mode(**방화벽 모드**)**를 **Routed(**라우팅**)** 또는 **Transparent(**투명**)** 중에서 선택합니다.

라우팅 모드에서 ASA는 네트워크의 라우터 홉으로 간주됩니다. 라우팅할 각 인터페이스가 다른 서브넷에 있습니다. 이와 반대로 투명 방화벽은 "비활성 엔드포인트(bump in the wire)" 또는 "은폐형 방화벽(stealth firewall)" 같은 역할을 수행하는 레이어 2 방화벽이며, 연결된 디바이스에 대한 라우터 홉으로 표시되지 않습니다.

방화벽 모드는 초기 구축 시에만 설정됩니다. 부트스트랩 설정을 다시 적용하는 경우에는 이 설정이 사용되지 않습니다.

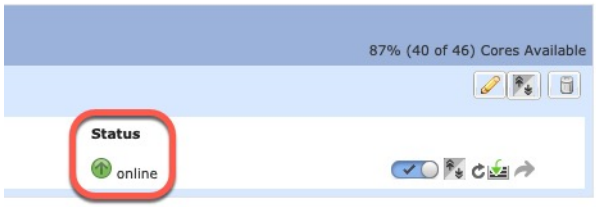
단계 **8** 관리자 및 비밀번호 활성화에 대해 **Password(**비밀번호**)**를 입력하고 확인합니다.

비밀번호를 복구할 때는 사전 구성된 ASA 관리 사용자/비밀번호 및 비밀번호 활성화를 사용하면 유용합니다. FXOS 액세스 권한이 있는데 관리 사용자 비밀번호/비밀번호 활성화를 잊어버린 경우 이를 재설정할 수 있습니다.

단계 **9** **OK(**확인**)**를 클릭하여 구성 대화 상자를 닫습니다.

단계 **10** **Save(**저장**)**를 클릭합니다.

섀시에서 지정된 소프트웨어 버전을 다운로드하고 부트스트랩 구성 및 관리 인터페이스 설정을 애플리케이션 인스턴스에 푸시하여 논리적 디바이스를 구축합니다. **Logical Devices(**논리적 디바이스**)** 페이지에서 새 논리적 디바이스의 상태를 확인합니다. 논리적 디바이스의 **Status(**상태**)**가 **online(**온라인**)**으로 표시되면 애플리케이션 내에서 보안 정책 구성을 시작할 수 있습니다.



단계 **11** 보안 정책 구성을 시작하려면 ASA 구성 가이드를 참조하십시오.

FMC에 대한 독립형 FTD 추가

독립형 논리적 디바이스는 단독으로 작동하거나 고가용성 쌍으로 작동합니다. 보안 모듈이 여러 개인 Firepower 9300에서는 클러스터 또는 독립형 디바이스를 구축할 수 있습니다. 클러스터는 모든 모듈을 사용해야 하므로 모듈이 2개인 클러스터와 단일 독립형 디바이스를 혼용하는 방식은 사용할 수 없습니다.

일부 모듈에서는 기본 인스턴스를, 다른 모듈에서는 컨테이너 인스턴스를 사용할 수 있습니다.

시작하기 전에

- Cisco.com에서 논리적 디바이스에 사용할 애플리케이션 이미지를 다운로드한 다음 해당 이미지를 Firepower 4100/9300 섀시.

참고 Firepower 9300의 경우 섀시 내의 개별 모듈에 서로 다른 애플리케이션 유형(ASA 및 FTD)을 설치할 수 있습니다. 개별 모듈에서 애플리케이션 인스턴스 유형의 서로 다른 버전을 실행할 수도 있습니다.

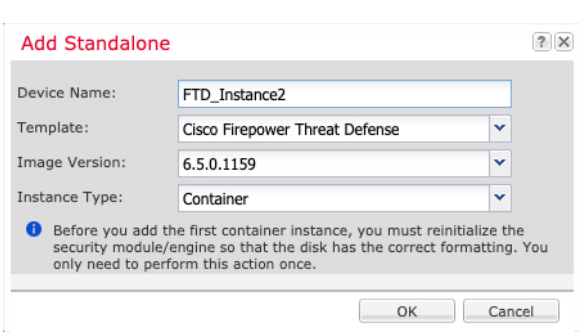
- 논리적 디바이스에 사용할 관리 인터페이스를 구성합니다. 관리 인터페이스는 필수 항목입니다. 이 관리 인터페이스는 섀시 관리용으로만 사용되는 섀시 관리 포트(**Interfaces(**인터페이스**)** 탭 상단에 **MGMT(**관리**)**로 표시됨)와는 다릅니다.

- 나중에 데이터 인터페이스에서 관리를 활성화할 수 있습니다. 데이터 관리를 활성화한 후 이를 사용하지 않으려는 경우에도 관리 인터페이스를 논리적 디바이스에 할당해야 합니다. 자세한 내용은 FTD 명령 참조의 **configure network management-data-interface** 명령을 참조하십시오.
- 최소 하나 이상의 데이터 유형 인터페이스도 구성해야 합니다. 또는 Firepower 이벤트 처리 인터페이스를 생성하여 모든 이벤트 트래픽을 전달할 수 있습니다(예: 웹 이벤트). 자세한 내용은 인터페이스 유형, 164 페이지를 참조하십시오.
- 컨테이너 인스턴스의 경우 기본 프로필을 사용하지 않으려면 컨테이너 인스턴스에 대한 리소스 프로파일 추가, 160 페이지에 따라 리소스 프로필을 추가합니다.
- 컨테이너 인스턴스의 경우 컨테이너 인스턴스를 처음으로 설치하기 전에 디스크가 올바른 형식을 갖도록 보안 모듈/엔진을 다시 초기화해야 합니다. **Security Modules(**보안 모듈**)** 또는 **Security Engine(**보안 엔진**)**을 선택하고 **Reinitialize(**초기화**)** 아이콘을 클릭합니다. 기존 논리적 디바이스가 삭제된 후에 새 디바이스로 재설치되며 로컬 애플리케이션 구성은 손실됩니다. 기본 인스턴스를 컨테이너 인스턴스로 교체할 때는 어떤 경우든 기본 인스턴스를 삭제해야 합니다. 기본 인스턴스를 컨테이너 인스턴스로 자동 마이그레이션할 수는 없습니다. 자세한 내용은 보안 모듈/엔진 확인 다시 초기화, 300 페이지를 참조하십시오.
- 다음 정보를 수집합니다.
  - 이 디바이스의 인터페이스 ID
  - 관리 인터페이스 IP 주소 및 네트워크 마스크
  - 게이트웨이 IP 주소
  - FMC 선택한 IP 주소 및/또는 NAT ID
  - DNS 서버 IP 주소
  - FTD 호스트 이름 및 도메인 이름

프로시저

단계 **1** **Logical Devices(**논리적 디바이스**)**를 선택합니다.

단계 **2** **Add(**추가**)** > **Standalone(**독립형**)**를 클릭하고 다음 파라미터를 설정합니다.



a) **Device Name(**디바이스 이름**)**을 입력합니다.

이 이름은 섀시 수퍼바이저가 관리 설정을 구성하고 인터페이스를 할당하는 데 사용됩니다. 이는 애플리케이션 구성에 사용되는 디바이스 이름이 아닙니다.

b) **Template(**템플릿**)**에서 **Cisco Firepower Threat Defense**를 선택합니다.

c) **Image Version(**이미지 버전**)**을 선택합니다.

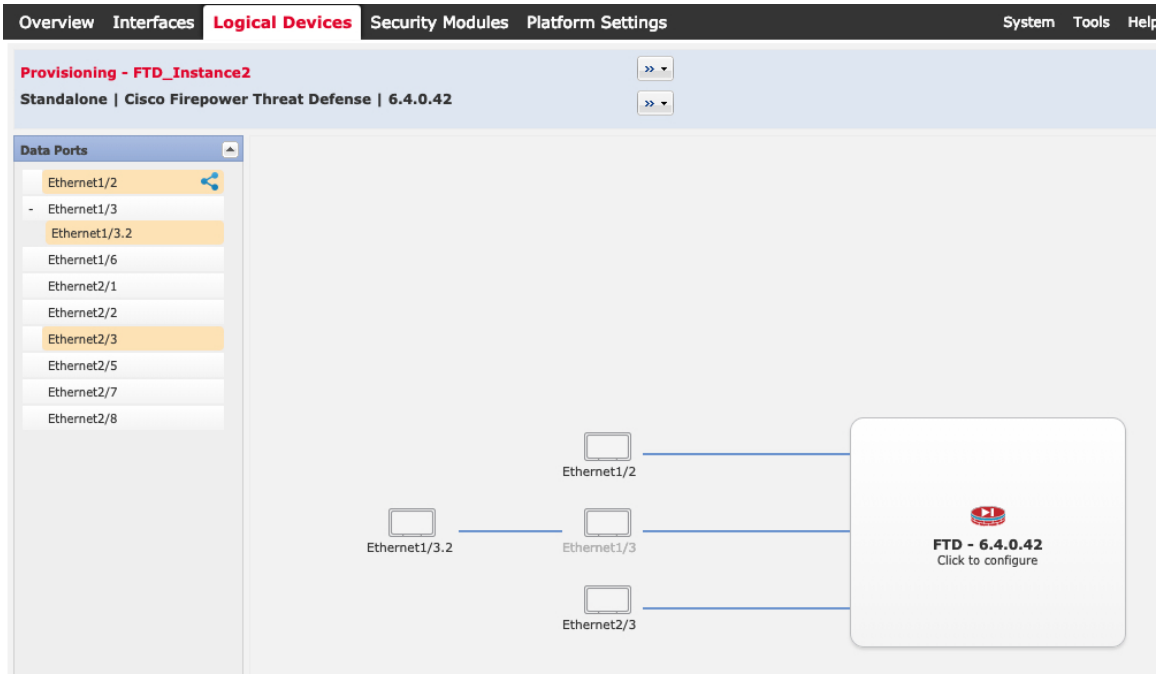
d) **Instance Type(**인스턴스 유형**)**을 **Container(**컨테이너**)** 또는 **Native(**기본**)** 중에서 선택합니다.

기본 인스턴스에서는 보안 모듈/엔진의 모든 리소스(CPU, RAM 및 디스크 공간)를 사용합니다. 따라서 하나의 기본 인스턴스만 설치할 수 있습니다. 컨테이너 인스턴스에서는 보안 모듈/엔진의 리소스 하위 집합을 사용합니다. 따라서 여러 개의 컨테이너 인스턴스를 설치할 수 있습니다.

e) **OK(**확인**)**를 클릭합니다.

Provisioning - *device name*(프로비저닝 - 디바이스 이름) 창이 표시됩니다.

단계 **3** **Data Ports(**데이터 포트**)** 영역을 확장하고 디바이스에 할당할 각 인터페이스를 클릭합니다.



이전에 **Interfaces(**인터페이스**)** 페이지에서 활성화한 데이터 및 데이터 공유 인터페이스만 할당할 수 있습니다. 나중에 IP 주소 설정을 비롯하여 FMC에서 이러한 인터페이스를 활성화하고 구성하게 됩니다.

컨테이너 인스턴스에는 데이터 공유 인터페이스를 10개까지만 할당할 수 있습니다. 또한 각 데이터 공유 인터페이스는 최대 14개의 컨테이너 인스턴스에 할당할 수 있습니다. 데이터 공유 인터페이스는 공유 아이콘( )으로 표시됩니다.

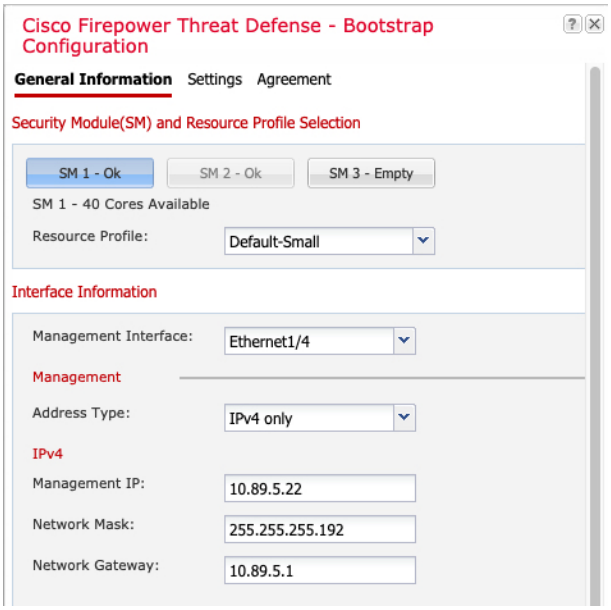
하드웨어 바이패스 지원 포트가 아이콘( )과 함께 표시됩니다. 특정 인터페이스 모듈의 경우 인라인 집합 인터페이스에 대해서만 하드웨어 우회 기능을 활성화할 수 있습니다(FMC 구성 가이드 참조). Hardware Bypass는 정전 중에 트래픽이 인라인 인터페이스 쌍 사이에서 계속 흐르도록 합니다. 이 기능은 소프트웨어 또는 하드웨어 오류의 경우 네트워크 연결성을 유지 관리하는 데 사용될 수 있

습니다. 하드웨어 바이패스 쌍에서 두 인터페이스를 할당하지 않는 경우 그러한 할당이 의도적인지를 확인하는 경고 메시지가 표시됩니다. 하드웨어 바이패스 기능을 사용할 필요가 없으므로 원하는 경우 단일 인터페이스를 할당할 수 있습니다.

단계 **4** 화면 중앙의 디바이스 아이콘을 클릭합니다.

초기 부트스트랩 설정을 구성할 수 있는 대화 상자가 표시됩니다. 이러한 설정은 초기 구축 전용 또는 재해 복구용입니다. 일반 작업 시에는 애플리케이션 CLI 구성에서 대부분의 값을 나중에 변경할 수 있습니다.

단계 **5** **General Information(**일반 정보**)** 페이지에서 다음 작업을 수행합니다.



a) (Firepower 9300의 경우) **Security Module Selection(**보안 모듈 선택**)** 아래에서 이 논리적 디바이스에 사용할 보안 모듈을 클릭합니다.

b) 컨테이너 인스턴스에 대해 **Resource Profile(**리소스 프로필**)**을 지정합니다.

나중에 다른 리소스 프로파일을 할당하는 경우 인스턴스가 다시 로드됩니다. 다시 로드는 5분 정도 걸릴 수 있습니다. 설정된 고가용성 쌍에 대해 크기가 다른 리소스 프로파일을 할당하는 경우에는 최대한 빠른 시간 내에 모든 멤버를 같은 크기로 설정해야 합니다.

c) **Management Interface(**관리 인터페이스**)**를 선택합니다.

이 인터페이스는 논리적 디바이스를 관리하는 데 사용됩니다. 이 인터페이스는 섀시 관리 포트와 별개입니다.

d) 관리 인터페이스 **Address Type(**주소 유형**)**을 **IPv4 only(IPv4** 전용**)**, **IPv6 only(IPv6** 전용**)** 또는 **IPv4 and IPv6(IPv4** 및 **IPv6)** 중에서 선택합니다.

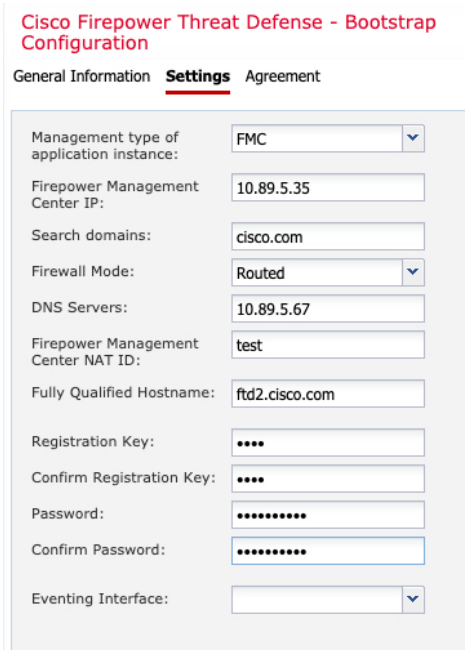
e) **Management IP(**관리 **IP)** 주소를 구성합니다.

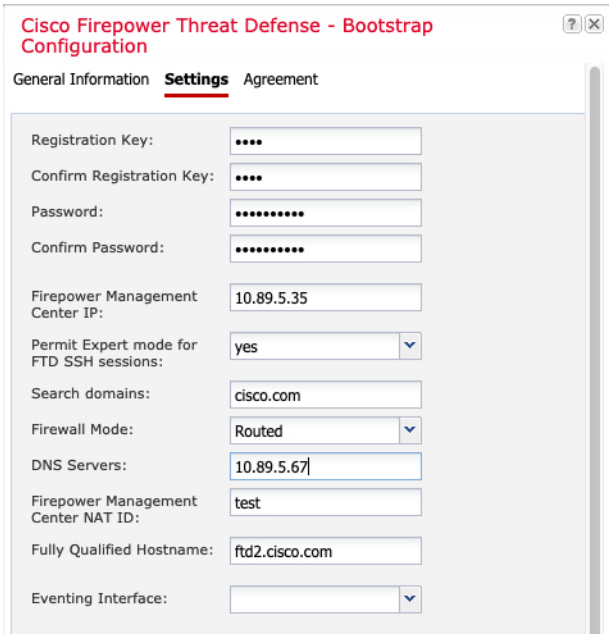
이 인터페이스의 고유 IP 주소를 설정합니다.

f) **Network Mask(**네트워크 마스크**)** 또는 **Prefix Length(**접두사 길이**)**를 입력합니다.

g) **Network Gateway(**네트워크 게이트웨이**)** 주소를 입력합니다.

단계 6 **Settings(**설정**)** 탭에서 다음 작업을 수행합니다.





a) 네이티브 인스턴스의 경우, **Management type of application instance(**애플리케이션 인스턴스의 관리 유형**)** 드롭다운 목록에서 **FMC**를 선택합니다.

   네이티브 인스턴스에서는 FDM을 관리자로도 지원합니다. 논리적 디바이스를 구축한 후에는 관리자 유형을 변경할 수 없습니다.

b) FMC 관리에 사용할 **Firepower Management Center IP**를 입력합니다. FMC IP 주소를 알 수 없는 경우, 이 필드를 비워두고 **Firepower Management Center NAT ID** 필드에 암호를 입력합니다.

c) 컨테이너 인스턴스의 경우, **Permit Export mode from FTD SSH sessions(FTD SSH** 세션에서 전문가 모드 허용**)**에 대해 **Yes(**예**)** 또는 **No(**아니요**)**를 선택합니다. 전문가 모드에서는 고급 트러블슈팅을 위한 FTD 셸 액세스 기능이 제공됩니다.

이 옵션에 대해 **Yes(**예**)**를 선택하는 경우 SSH 세션에서 컨테이너 인스턴스에 직접 액세스할 수 있는 사용자가 전문가 모드를 시작할 수 있습니다. **No(**아니요**)**를 선택하는 경우에는 FXOS CLI에서 컨테이너 인스턴스에 액세스할 수 있는 사용자만 전문가 모드를 시작할 수 있습니다. 각 인스턴스를 더욱 명확하게 격리할 수 있도록 **No(**아니요**)**를 선택하는 것이 좋습니다.

문서에 설명되어 있는 절차에 따라 Expert 모드가 필요하다고 알려주는 경우 또는 Cisco Technical Assistance Center에서 사용하도록 요청하는 경우에만 Expert 모드를 사용합니다. 이 모드를 설정하려면 FTD CLI에서 **expert** 명령을 사용합니다.

d) **Search Domains(**검색 도메인**)**를 쉼표로 구분된 목록으로 입력합니다.

e) **Firewall Mode(**방화벽 모드**)**를 **Transparent(**투명**)** 또는 **Routed(**라우팅**)** 중에서 선택합니다.

라우팅 모드에서 FTD는 네트워크의 라우터 홉으로 간주됩니다. 라우팅할 각 인터페이스가 다른 서브넷에 있습니다. 이와 반대로 투명 방화벽은 "비활성 엔드포인트(bump in the wire)" 또는 "은폐형 방화벽(stealth firewall)" 같은 역할을 수행하는 레이어 2 방화벽이며, 연결된 디바이스에 대한 라우터 홉으로 표시되지 않습니다.

방화벽 모드는 초기 구축 시에만 설정됩니다. 부트스트랩 설정을 다시 적용하는 경우에는 이 설정이 사용되지 않습니다.

f) **DNS Servers(DNS** 서버**)**를 쉼표로 구분된 목록으로 입력합니다.

예를 들어, FMC의 호스트 이름을 지정하는 경우, FTD에서는 DNS를 사용합니다.

g) FTD의 **Fully Qualified Hostname(**정규화된 호스트 이름**)**을 입력합니다.

h) 등록 시 FMC와 디바이스 간에 공유할 **Registration Key(**등록 키**)**를 입력합니다.

이 키에 대해 1~37자의 텍스트 문자열을 선택할 수 있습니다. FTD를 추가하는 경우 FMC에 동일한 키를 입력합니다.

i) FTD 관리 사용자가 CLI에 액세스할 때 사용할 **Password(**비밀번호**)**를 입력합니다.

j) 이벤트를 전송할 **Eventing Interface(**이벤트 인터페이스**)**를 선택합니다. 인터페이스가 지정되지 않은 경우, 관리 인터페이스가 사용됩니다.

이 인터페이스는 Firepower 이벤트 처리 인터페이스로 정의해야 합니다.

k) 컨테이너 인스턴스의 경우 **Hardware Crypto(**하드웨어 암호화**)**를 **Enabled(**활성화됨**)** 또는 **Disabled(**비활성화됨**)**로 설정합니다.

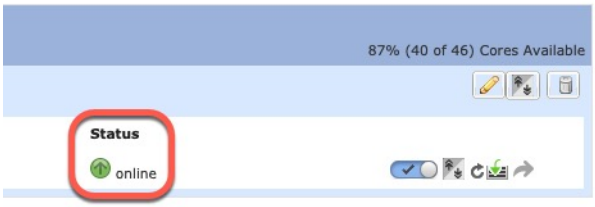
이 설정은 하드웨어에서 TLS 암호화 가속화를 활성화하고 특정 유형의 트래픽에 대한 성능을 개선합니다. 이 기능은 기본적으로 활성화되어 있습니다. 보안 모듈당 최대 16개의 인스턴스에 대해 TLS 암호화 가속화를 활성화할 수 있습니다. 이 기능은 네이티브 인스턴스에서 항상 사용할 수 있습니다. 이 인스턴스에 할당된 하드웨어 암호화 리소스의 백분율을 보려면 **show hw-crypto** 명령을 입력합니다.

단계 **7** **Agreement(**계약**)** 탭에서 EULA(End User License Agreement)를 읽고 내용에 동의해야 합니다.

단계 **8** **OK(**확인**)**를 클릭하여 구성 대화 상자를 닫습니다.

단계 **9** **Save(**저장**)**를 클릭합니다.

섀시에서 지정된 소프트웨어 버전을 다운로드하고 부트스트랩 구성 및 관리 인터페이스 설정을 애플리케이션 인스턴스에 푸시하여 논리적 디바이스를 구축합니다. **Logical Devices(**논리적 디바이스**)** 페이지에서 새 논리적 디바이스의 상태를 확인합니다. 논리적 디바이스의 **Status(**상태**)**가 **online(**온라인**)**으로 표시되면 애플리케이션 내에서 보안 정책 구성을 시작할 수 있습니다.



단계 **10** FTD를 매니지드 디바이스로 추가하고 보안 정책 구성을 시작하려면 FMC 구성 가이드를 참조합니다.

FDM에 대한 독립형 FTD 추가

네이티브 인스턴스로 FDM을 사용할 수 있습니다. 컨테이너 인스턴스는 지원되지 않습니다. 독립형 논리적 디바이스는 단독으로 작동하거나 고가용성 쌍으로 작동합니다.

시작하기 전에

- Cisco.com에서 논리적 디바이스에 사용할 애플리케이션 이미지를 다운로드한 다음 해당 이미지를 Firepower 4100/9300 섀시.

참고 Firepower 9300의 경우 섀시 내의 개별 모듈에 서로 다른 애플리케이션 유형(ASA 및 FTD)을 설치할 수 있습니다. 개별 모듈에서 애플리케이션 인스턴스 유형의 서로 다른 버전을 실행할 수도 있습니다.

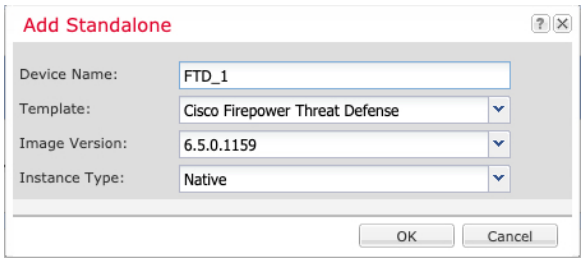
- 논리적 디바이스에 사용할 관리 인터페이스를 구성합니다. 관리 인터페이스는 필수 항목입니다. 이 관리 인터페이스는 섀시 관리용으로만 사용되는 섀시 관리 포트**(Interfaces(**인터페이스**)** 탭 상단에 **MGMT(**관리**)**로 표시됨)와는 다릅니다.
- 최소 하나 이상의 데이터 유형 인터페이스도 구성해야 합니다.
- 다음 정보를 수집합니다.
  - 이 디바이스의 인터페이스 ID
  - 관리 인터페이스 IP 주소 및 네트워크 마스크

- 게이트웨이 IP 주소
- DNS 서버 IP 주소
- FTD 호스트 이름 및 도메인 이름

프로시저

단계 1 **Logical Devices(**논리적 디바이스**)**를 선택합니다.

단계 2 **Add(**추가**)** > **Standalone(**독립형**)**를 클릭하고 다음 파라미터를 설정합니다.



a) **Device Name(**디바이스 이름**)**을 입력합니다.

이 이름은 섀시 수퍼바이저가 관리 설정을 구성하고 인터페이스를 할당하는 데 사용됩니다. 이는 애플리케이션 구성에 사용되는 디바이스 이름이 아닙니다.

b) **Template(**템플릿**)**에서 **Cisco Firepower Threat Defense**를 선택합니다.

c) **Image Version(**이미지 버전**)**을 선택합니다.

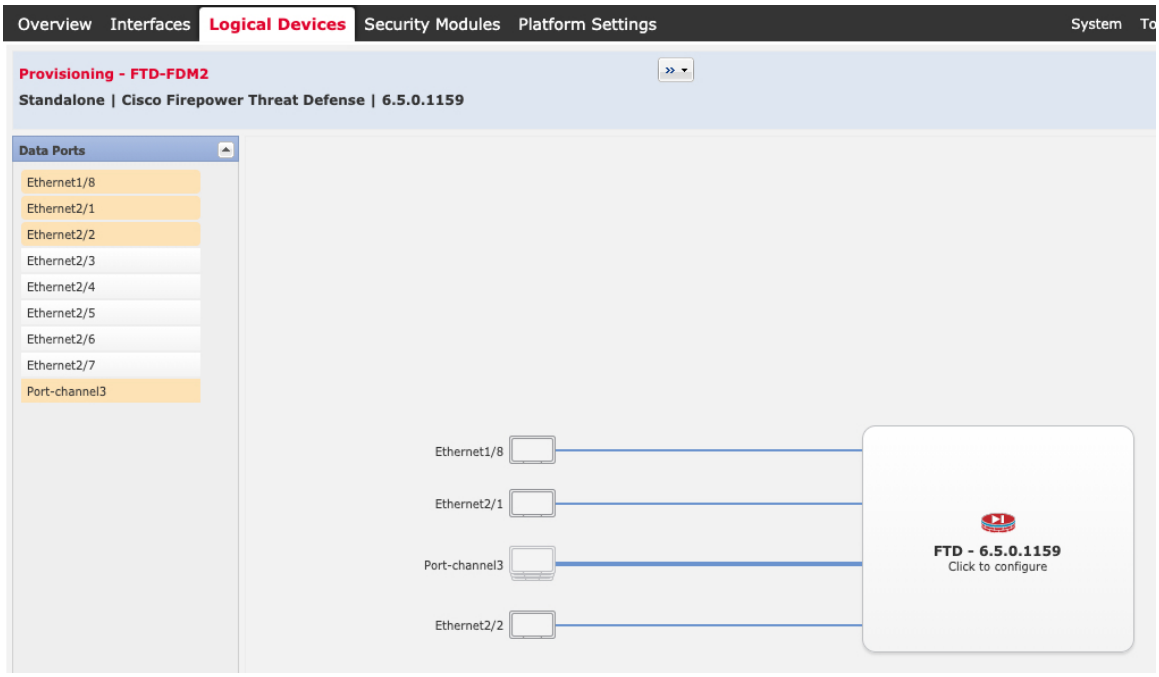
d) **Instance Type(**인스턴스 유형**)**은 **Native(**네이티브**)**를 선택합니다.

컨테이너 인스턴스는 FDM에서 지원되지 않습니다.

e) **OK(**확인**)**를 클릭합니다.

Provisioning - *device name*(프로비저닝 - 디바이스 이름) 창이 표시됩니다.

단계 3 **Data Ports(**데이터 포트**)** 영역을 확장하고 디바이스에 할당할 각 인터페이스를 클릭합니다.

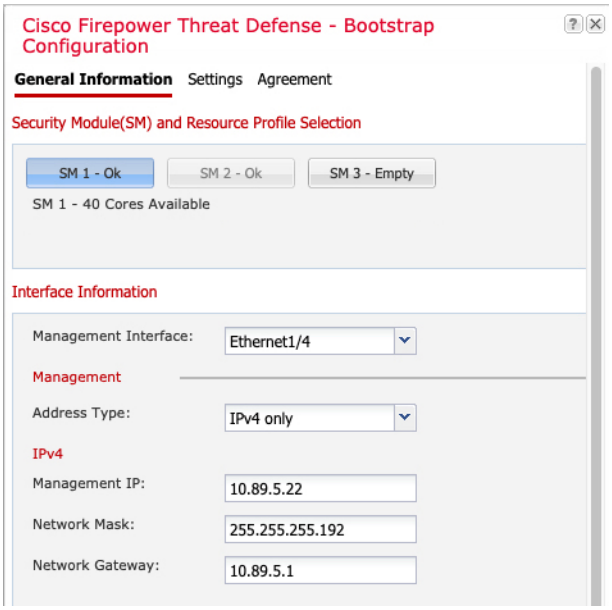


이전에 **Interfaces(**인터페이스**)** 페이지에서 활성화한 데이터 인터페이스만 할당할 수 있습니다. 나중에 IP 주소 설정을 비롯하여 FDM에서 이러한 인터페이스를 활성화하고 구성하게 됩니다.

단계 **4** 화면 중앙의 디바이스 아이콘을 클릭합니다.

초기 부트스트랩 설정을 구성할 수 있는 대화 상자가 표시됩니다. 이러한 설정은 초기 구축 전용 또는 재해 복구용입니다. 일반 작업 시에는 애플리케이션 CLI 구성에서 대부분의 값을 나중에 변경할 수 있습니다.

단계 **5** **General Information(**일반 정보**)** 페이지에서 다음 작업을 수행합니다.



a) (Firepower 9300의 경우) **Security Module Selection(**보안 모듈 선택**)** 아래에서 이 논리적 디바이스에 사용할 보안 모듈을 클릭합니다.

b) **Management Interface(**관리 인터페이스**)**를 선택합니다.

이 인터페이스는 논리적 디바이스를 관리하는 데 사용됩니다. 이 인터페이스는 섀시 관리 포트와 별개입니다.

c) 관리 인터페이스 **Address Type(**주소 유형**)**을 **IPv4 only(IPv4** 전용**)**, **IPv6 only(IPv6** 전용**)** 또는 **IPv4 and IPv6(IPv4** 및 **IPv6)** 중에서 선택합니다.

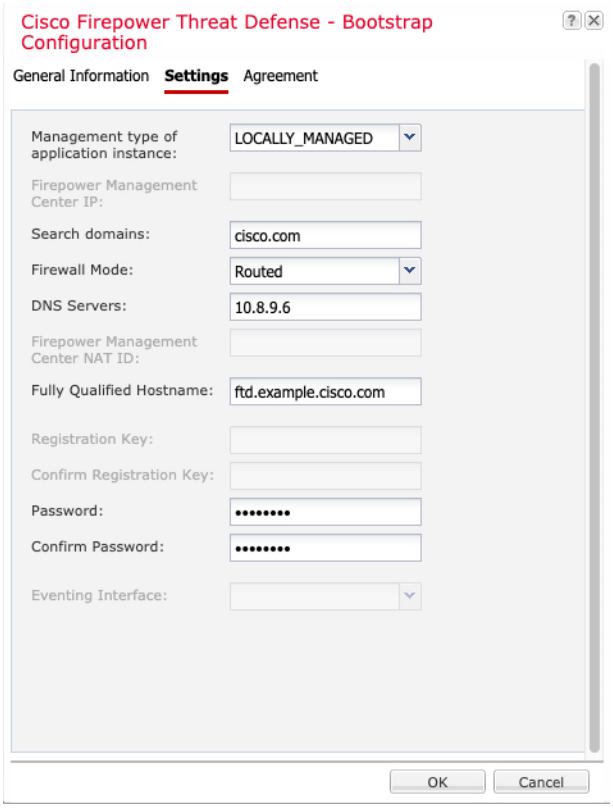
d) **Management IP(**관리 **IP)** 주소를 구성합니다.

이 인터페이스의 고유 IP 주소를 설정합니다.

e) **Network Mask(**네트워크 마스크**)** 또는 **Prefix Length(**접두사 길이**)**를 입력합니다.

f) **Network Gateway(**네트워크 게이트웨이**)** 주소를 입력합니다.

단계 **6** **Settings(**설정**)** 탭에서 다음 작업을 수행합니다.



a) **Management type of application instance(**애플리케이션 인스턴스의 관리 유형**)** 드롭다운 목록에서 **LOCALLY_MANAGED**를 선택합니다.

네이티브 인스턴스에서는 Firepower Management Center을 관리자로도 지원합니다. 논리적 디바이스를 구축한 후 관리자를 변경하면 구성이 지워지고 디바이스가 다시 초기화됩니다.

b) **Search Domains(**검색 도메인**)**를 쉼표로 구분된 목록으로 입력합니다.

c) **Firewall Mode(**방화벽 모드**)**에서는 **Routed(**라우팅**)** 모드만 지원합니다.

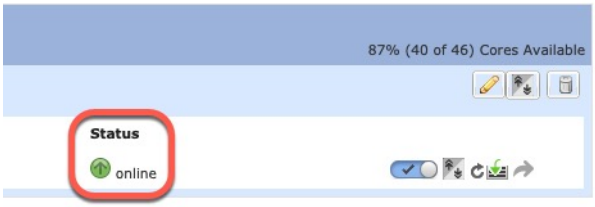
d) **DNS Servers(DNS** 서버**)**를 쉼표로 구분된 목록으로 입력합니다.
e) FTD의 **Fully Qualified Hostname(**정규화된 호스트 이름**)**을 입력합니다.
f) FTD 관리 사용자가 CLI에 액세스할 때 사용할 **Password(**비밀번호**)**를 입력합니다.

단계 **7** **Agreement(**계약**)** 탭에서 EULA(End User License Agreement)를 읽고 내용에 동의해야 합니다.

단계 **8** **OK(**확인**)**를 클릭하여 구성 대화 상자를 닫습니다.

단계 **9** **Save(**저장**)**를 클릭합니다.

섀시에서 지정된 소프트웨어 버전을 다운로드하고 부트스트랩 구성 및 관리 인터페이스 설정을 애플리케이션 인스턴스에 푸시하여 논리적 디바이스를 구축합니다. **Logical Devices(**논리적 디바이스**)** 페이지에서 새 논리적 디바이스의 상태를 확인합니다. 논리적 디바이스의 **Status(**상태**)**가 **online(**온라인**)**으로 표시되면 애플리케이션 내에서 보안 정책 구성을 시작할 수 있습니다.



단계 **10** 보안 정책 구성을 시작하려면 FDM 구성 가이드를 참조하십시오.

고가용성 쌍 추가

FTD 또는 ASA 고가용성(장애 조치라고도 함)은 FXOS가 아닌 애플리케이션 내에 구성됩니다. 그러나 고가용성을 사용할 수 있도록 섀시를 준비하려는 경우 다음 단계를 참조하십시오.

시작하기 전에

고가용성 요구 사항 및 사전 요건, 217 페이지의 내용을 참조하십시오.

프로시저

단계 **1** 각 논리적 디바이스에 동일한 인터페이스를 할당합니다.

단계 **2** 페일오버 및 상태 링크용으로 데이터 인터페이스 1~2개를 할당합니다.

이러한 인터페이스는 두 섀시 간의 고가용성 트래픽을 교환합니다. 페일오버 및 상태 링크를 함께 사용하려면 10GB 데이터 인터페이스를 사용하는 것이 좋습니다. 사용 가능한 인터페이스가 있다면 페일오버 및 상태 링크를 각각 별도로 사용할 수 있습니다. 상태 링크에는 최대 대역폭이 필요합니다.

관리 유형 인터페이스는 페일오버 또는 상태 링크용으로 사용할 수 없습니다. 페일오버 인터페이스와 같은 네트워크 세그먼트에 다른 디바이스가 없는 상태로 섀시 간에 스위치를 사용하는 것이 좋습니다.

컨테이너 인스턴스의 경우 데이터 공유 인터페이스는 페일오버 링크용으로 지원되지 않습니다. 상위 인터페이스 또는 EtherChannel에서 하위 인터페이스를 생성한 다음 각 인스턴스에 대해 페일오버 링크로 사용할 하위 인터페이스를 할당하는 것이 좋습니다. 동일한 상위 인터페이스에 있는 모든 하위 인터페이스를 페일오버 링크로 사용해야 합니다. 하위 인터페이스 하나를 페일오버 링크로 사용하고 다른 하위 인터페이스(또는 상위 인터페이스)를 일반 데이터 인터페이스로 사용할 수는 없습니다.

단계 **3** 논리적 디바이스에서 고가용성을 활성화합니다.

단계 **4** 고가용성을 활성화한 후에 인터페이스를 변경해야 하는 경우에는 먼저 스탠바이 유닛에서 변경을 수행한 다음 액티브 유닛에서 변경을 수행합니다.

참고 ASA의 경우 네트워크 모듈/EtherChannel을 제거하거나 EtherChannel에 인터페이스를 재할당하는 등 FXOS에서 인터페이스를 제거하면 ASA 구성에서 원래 명령이 유지되므로 필요한 조정을 수행할 수 있습니다. 구성에서 인터페이스를 제거하는 경우에는 구성 전반에 걸쳐 영향을 줄 수 있습니다. ASA OS에서 이전 인터페이스 구성을 수동으로 제거할 수 있습니다.

클러스터 추가

클러스터링을 사용하면 여러 개의 디바이스를 하나의 논리적 디바이스로 그룹화할 수 있습니다. 클러스터는 처리량 증대 및 여러 디바이스의 이중화라는 목표를 달성하는 동시에 단일 디바이스(관리, 네트워크에 통합)의 모든 편의성을 제공합니다. 여러 모듈을 포함하는 Firepower 9300은 단일 섀시의 모든 모듈을 하나의 클러스터로 그룹화하는 인트라 섀시 클러스터링(intra-chassis clustering)을 지원합니다. 여러 섀시가 그룹화되는 섀시 간 클러스터링을 사용할 수도 있습니다. Firepower 4100 Series 같은 단일 모듈 디바이스에는 섀시 간 클러스터링이 유일한 옵션입니다.

Firepower 4100/9300 섀시 클러스터링 정보

Firepower 4100/9300 섀시에서 클러스터를 구축할 때는 다음 작업이 수행됩니다.

- 네이티브 인스턴스 클러스터링의 경우: 유닛 간 통신에 사용되는 클러스터 제어 링크(기본값: port-channel 48)를 생성합니다.

  다중 인스턴스 클러스터링의 경우에는 하나 이상의 클러스터 유형 Etherchannel에서 하위 인터페이스를 사전 구성해야 합니다. 각 인스턴스에는 자체 클러스터 제어 링크가 필요 합니다.

  섀시 내 클러스터링(Firepower 9300 전용)의 경우, 이 링크는 클러스터 통신에 Firepower 9300 백플레인을 활용합니다.

  섀시 간 클러스터링의 경우, 섀시 간의 통신을 위해 물리적 인터페이스를 이 EtherChannel에 수동으로 할당해야 합니다.

- 애플리케이션 내부에 클러스터 부트스트랩 구성을 생성합니다.

  클러스터를 구축할 때, 섀시 수퍼바이저는 클러스터 이름, 클러스터 제어 링크 인터페이스 및 기타 클러스터 설정을 포함하는 각 유닛에 최소한의 부트스트랩 구성을 푸시합니다. 클러스터링 환경을 사용자 정의하려는 경우, 사용자가 일부 부트스트랩 구성을 애플리케이션 내부에 구성할 수 있습니다.

- 데이터 인터페이스를 *Spanned* 인터페이스로 클러스터에 할당합니다.

  섀시 내 클러스터링의 경우, 스팬 인터페이스는 섀시 간 클러스터링과 마찬가지로 EtherChannel에 국한되지 않습니다. Firepower 9300 수퍼바이저는 EtherChannel 기술을 내부에 사용하여 트래픽을 공유 인터페이스의 다중 모듈에 로드 밸런싱하므로 모든 데이터 인터페이스 유형이 Spanned(스팬) 모드에서 작동합니다. 섀시 간 클러스터링의 경우, 모든 데이터 인터페이스에 Spanned EtherChannel을 사용해야 합니다.

참고 개별 인터페이스는 관리 인터페이스를 제외하고 지원되지 않습니다.

- 관리 인터페이스를 클러스터의 모든 유닛에 할당합니다.

기본 유닛 및 보조 유닛 역할

클러스터의 멤버 중 하나는 기본 유닛입니다. 기본 유닛은 자동으로 결정됩니다. 기타 모든 멤버는 보조 유닛입니다.

기본 유닛에서만 모든 구성을 수행해야 하며 이후에 구성은 보조 유닛에 복제됩니다.

클러스터 제어 링크

네이티브 인스턴스 클러스터링의 경우: 클러스터 제어 링크는 Port-channel 48 인터페이스를 사용하여 자동으로 생성됩니다.

다중 인스턴스 클러스터링의 경우에는 하나 이상의 클러스터 유형 Etherchannel에서 하위 인터페이스를 사전 구성해야 합니다. 각 인스턴스에는 자체 클러스터 제어 링크가 필요 합니다.

섀시 내 클러스터링의 경우, 이 인터페이스에는 멤버 인터페이스가 없습니다. 이 클러스터 유형 EtherChannel은 인트라 섀시 클러스터링(intra-chassis clustering)을 위한 클러스터 통신에 Firepower 9300 백플레인을 활용합니다. 섀시 간 클러스터링의 경우에는 EtherChannel에 인터페이스를 하나 이상 추가해야 합니다.

2-멤버 섀시 간 클러스터의 경우 클러스터 제어 링크를 한 섀시에서 다른 섀시로 직접 연결하지 마십시오. 인터페이스에 직접 연결할 경우, 유닛 하나에 오류가 발생하면 클러스터 제어 링크에도 오류가 발생하므로 나머지 정상 유닛에도 오류가 발생합니다. 스위치를 통해 클러스터 제어 링크를 연결할 경우 클러스터 제어 링크는 가동 상태를 유지하여 정상 유닛을 지원합니다.

클러스터 제어 링크 트래픽에는 제어 및 데이터 트래픽이 모두 포함됩니다.

섀시 간 클러스터링을 위한 클러스터 제어 링크 크기 조정

가능한 경우, 각 섀시의 예상 처리량에 맞게 클러스터 제어 링크의 크기를 조정하여 클러스터 제어 링크가 최악의 시나리오를 처리할 수 있게 해야 합니다.

클러스터 제어 링크 트래픽은 주로 상태 업데이트 및 전달된 패킷으로 구성되어 있습니다. 클러스터 제어 링크의 트래픽 양은 언제든지 달라질 수 있습니다. 전달된 트래픽의 양은 로드 밸런싱 효율성 또는 중앙 집중식 기능에 많은 트래픽이 있는지에 따라 좌우됩니다. 예를 들면 다음과 같습니다.

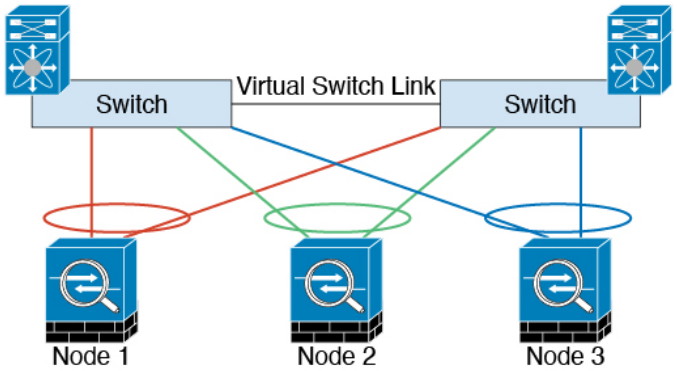
- NAT의 경우 연결의 로드 밸런싱이 저하되며, 모든 반환 트래픽을 올바른 유닛으로 다시 밸런싱해야 합니다.
- 멤버가 변경된 경우, 클러스터에서는 다량의 연결을 다시 밸런싱해야 하므로 일시적으로 많은 양의 클러스터 제어 링크 대역폭을 사용합니다.

대역폭이 높은 클러스터 제어 링크를 사용하면 멤버가 변경될 경우 클러스터를 더 빠르게 통합할 수 있고 처리량 병목 현상을 방지할 수 있습니다.

참고 클러스터에 비대칭(다시 밸런싱된) 트래픽이 많은 경우 클러스터 제어 링크 크기를 늘려야 합니다.

섀시 간 클러스터링을 위한 클러스터 제어 링크 이중화

다음 다이어그램은 EtherChannel을 VSS(Virtual Switching System), vPC(Virtual Port Channel), StackWise 또는 StackWise 가상 환경에서 클러스터 제어 링크로 사용하는 방법을 보여줍니다. EtherChannel의 모든 링크가 활성화되어 있습니다. 스위치가 중복 시스템의 일부인 경우 동일한 EtherChannel 내의 방화벽 인터페이스를 중복 시스템의 개별 스위치에 연결할 수 있습니다. 이러한 별도의 스위치는 단일 스위치 역할을 하므로, 스위치 인터페이스는 동일한 EtherChannel 포트 채널 인터페이스의 멤버입니다. 이러한 EtherChannel은 디바이스 로컬이 아닌 스팬 EtherChannel입니다.



섀시 간 클러스터링을 위한 클러스터 제어 링크 안정성

클러스터 제어 링크 기능을 보장하려면 유닛 간의 RTT(round-trip time)가 20ms 이하여야 합니다. 이러한 최대 레이턴시는 서로 다른 지리적 사이트에 설치된 클러스터 멤버와의 호환성을 개선하는 역할을 합니다. 레이턴시를 확인하려면 유닛 간의 클러스터 제어 링크에서 Ping을 수행합니다.

클러스터 제어 링크는 오류가 나거나 폐기된 패킷이 없는 안정적인 상태여야 합니다. 예를 들어, 사이트 간 구축의 경우 전용 링크를 사용해야 합니다.

클러스터 제어 링크 네트워크

Firepower 4100/9300 섀시에서는 섀시 ID 및 슬롯 ID 127.2.*chassis_id*.*slot_id*를 기준으로 하여 각 유닛에 대해 클러스터 제어 링크 인터페이스 IP 주소를 자동 생성합니다. 일반적으로 같은 EtherChannel의 다른 VLAN 하위 인터페이스를 사용하는 다중 인스턴스 클러스터의 경우 VLAN 분리로 인해 서로 다른 클러스터에 같은 IP 주소를 사용할 수 있습니다. 클러스터를 구축할 때 이 IP 주소를 맞춤 설정할 수 있습니다. 클러스터 제어 링크 네트워크는 유닛 간에 라우터를 포함할 수 없으며 레이어 2 스위칭만 허용됩니다. 사이트 간 트래픽의 경우에는 OTV(Overlay Transport Virtualization)를 사용하는 것이 좋습니다.

관리 네트워크

모든 유닛을 단일한 관리 네트워크에 연결할 것을 권장합니다. 이 네트워크는 클러스터 제어 링크와 분리되어 있습니다.

관리 인터페이스

클러스터에 관리 유형 인터페이스를 할당해야 합니다. 이 인터페이스는 Spanned 인터페이스와는 다른 특수 개별 인터페이스입니다. 관리 인터페이스를 사용하면 각 유닛에 직접 연결할 수 있습니다.

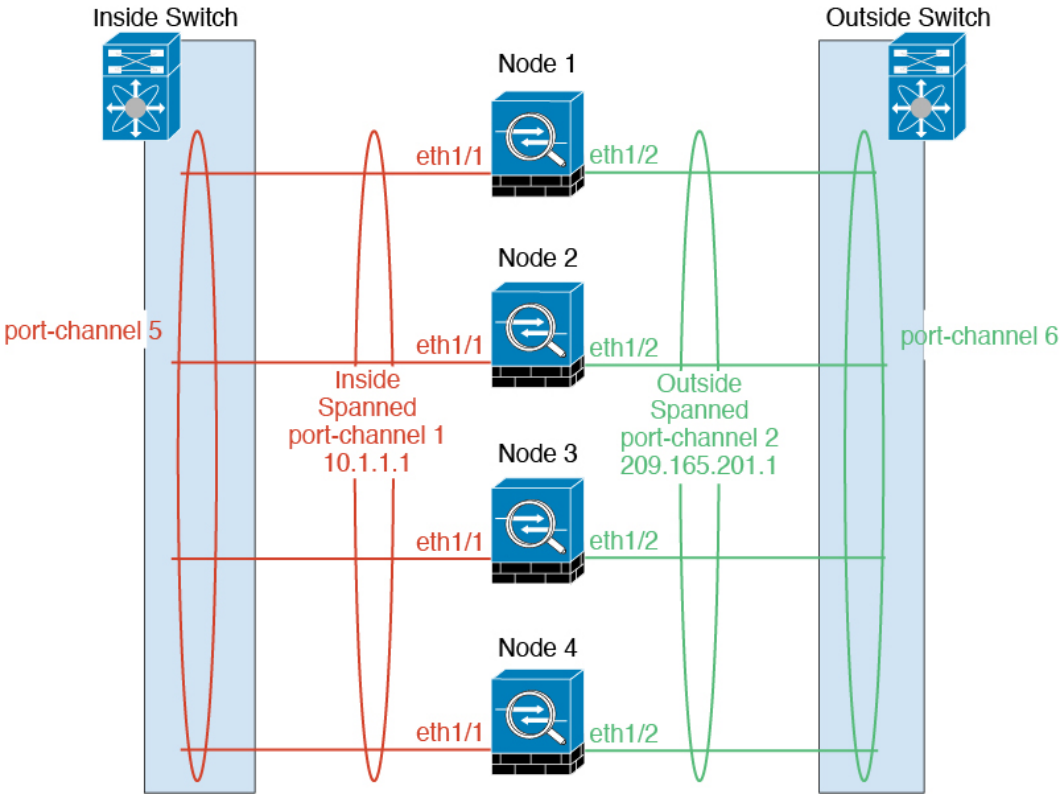
ASA의 경우, 기본 클러스터 IP 주소는 현재 기본 유닛에 항상 속해 있는 클러스터를 위한 고정 주소입니다. 또한 주소의 범위를 구성하여 현재 기본 유닛을 비롯한 각 유닛에서 해당 범위의 로컬 주소를 사용할 수 있도록 해야 합니다. 기본 클러스터 IP 주소에서는 주소에 대한 일관된 관리 액세스를 제공합니다. 기본 유닛이 변경될 경우 기본 클러스터 IP 주소는 새 기본 유닛으로 이동되므로 클러스터는 지속적으로 원활하게 관리됩니다. 로컬 IP 주소는 라우팅에 사용되며 트러블슈팅에도 도움이 됩니다. 예를 들어, 현재 기본 유닛에 항상 연결되어 있는 기본 클러스터 IP 주소에 연결하여 클러스터를 관리할 수 있습니다. 로컬 IP 주소에 연결하여 개별 멤버를 관리할 수 있습니다. TFTP 또는 syslog 같은 아웃바운드 관리 트래픽의 경우 기본 유닛을 비롯한 각 유닛에서 로컬 IP 주소를 사용하여 서버에 연결합니다.

FTD의 경우, 동일한 네트워크의 각 유닛에 관리 IP 주소를 할당합니다. 각 유닛을 FMC에 추가할 때 이 IP 주소를 사용합니다.

스팬 **EtherChannels**

섀시당 하나 이상의 인터페이스를 클러스터 내의 모든 섀시를 포괄하는 EtherChannel로 그룹화할 수 있습니다. EtherChannel에서는 채널에서 사용 가능한 모든 활성 인터페이스 전반의 트래픽을 취합합니다. 스팬 EtherChannel은 라우팅 및 투명 방화벽 모드에서 모두 구성할 수 있습니다. 라우팅 모드의 경우 EtherChannel은 단일 IP 주소를 통해 라우팅된 인터페이스로 구성됩니다. 투명 모드의 경우 브리지 그룹 멤버 인터페이스가 아닌 BVI에 IP 주소가 할당됩니다. EtherChannel은 기본적인 작동 시 로드 밸런싱을 함께 제공합니다.

다중 인스턴스 클러스터의 경우 각 클러스터에 전용 데이터 EtherChannel이 필요하며 공유 인터페이스 또는 VLAN 하위 인터페이스를 사용할 수 없습니다.



사이트 간 클러스터링

사이트 간 설치 시 다음 권장 지침을 준수하면 클러스터링을 활용할 수 있습니다.

각 클러스터 섀시를 별도의 사이트 ID에 속하도록 구성할 수 있습니다.

사이트 ID는 사이트별 MAC 주소 및 IP 주소와 작동합니다. 클러스터에서 이그레스되는 패킷은 사이트별 MAC 주소 및 IP 주소를 사용하는 반면, 클러스터가 수신한 패킷은 전역 MAC 주소 및 IP 주소를 사용합니다. 이 기능은 스위치가 서로 다른 두 포트의 두 사이트로부터 동일한 전역 MAC 주소를 학습하지 못하게 하는 한편, MAC 플래핑(flapping)을 일으킵니다. 대신 스위치는 사이트 MAC 주소만 학습합니다. 사이트별 MAC 주소 및 IP 주소는 Spanned EtherChannel만을 사용하는 라우팅 모드에서 지원됩니다.

사이트 ID는 LISP 검사를 사용한 플로우 모빌리티 활성화, 데이터 센터의 사이트 간 클러스터링에 대해 왕복 시간 레이턴시를 줄이고 성능을 개선하기 위한 관리자 지역화, 그리고 트래픽 플로우의 백업 소유자가 항상 소유자와 다른 사이트에 있는 연결에 대한 사이트 이중화에도 사용됩니다.

사이트 간 클러스터링에 대한 자세한 내용은 다음 섹션을 참조하십시오.

- 데이터 센터 인터커넥트 크기 조정 -클러스터링의 요구 사항 및 사전 요구 사항, 213 페이지
- 사이트 간 지침 -클러스터링 지침 및 제한 사항, 220 페이지
- 사이트 간 예시 -사이트 간 클러스터링 예시, 287 페이지

ASA 클러스터 추가

단일 Firepower 9300 섀시를 섀시 내 클러스터로 추가하거나 섀시 간 클러스터링용으로 여러 섀시를 추가할 수 있습니다. 섀시 간 클러스터링의 경우, 각 섀시를 개별적으로 구성해야 합니다. 섀시 하나에 클러스터를 추가한 다음 쉽게 구축하기 위해 첫 번째 섀시의 부트스트랩 구성을 다음 섀시에 복사합니다.

ASA 클러스터 생성

이미지 버전의 범위를 설정합니다.

Firepower 4100/9300 섀시 수퍼바이저에서 클러스터를 손쉽게 구축할 수 있습니다. 모든 초기 구성은 유닛마다 자동으로 생성됩니다.

섀시 간 클러스터링의 경우 각 섀시를 개별적으로 구성해야 합니다. 섀시 하나에 클러스터를 구축한 다음 쉽게 구축하기 위해 첫 번째 섀시의 부트스트랩 컨피그레이션을 다음 섀시에 복사합니다.

모듈을 설치하지 않은 경우에도 Firepower 9300 섀시의 3개 모듈 슬롯 모두 또는 컨테이너 인스턴스, 각 슬롯의 컨테이너 인스턴스에 대해 클러스터링을 활성화해야 합니다. 3개 모듈을 모두 구성하지 않은 경우 클러스터가 나타나지 않습니다.

다중 컨텍스트 모드의 경우 먼저 논리적 디바이스를 구축한 다음 ASA 애플리케이션에서 다중 컨텍스트 모드를 활성화해야 합니다.

시작하기 전에

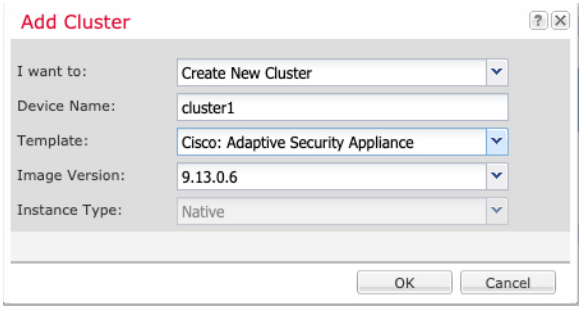
- Cisco.com에서 논리적 디바이스에 사용할 애플리케이션 이미지를 다운로드한 다음, 해당 이미지를 Firepower 4100/9300 섀시에 업로드합니다.
- 다음 정보를 수집합니다.
  - 관리 인터페이스 ID, IP 주소, 네트워크 마스크
  - 게이트웨이 IP 주소

프로시저

단계 **1** 인터페이스를 구성합니다.

단계 **2** **Logical Devices(**논리적 디바이스**)**를 선택합니다.

단계 **3** **Add(**추가**)** > **Cluster(**클러스터**)**를 클릭하고 다음 파라미터를 설정합니다.



a) **I want to:(**수행할 작업**:)** > **Create New Cluster(**새 클러스터 생성**)**를 선택합니다.
b) **Device Name(**디바이스 이름**)**을 입력합니다.

   이 이름은 내부적으로 섀시 수퍼바이저가 관리 설정을 구성하고 인터페이스를 할당하는 데 사용됩니다. 이는 애플리케이션 구성에 사용되는 디바이스 이름이 아닙니다.

c) **Template(**템플릿**)**은 **Cisco Adaptive Security Appliance**를 선택합니다.
d) **Image Version(**이미지 버전**)**을 선택합니다.
e) **Instance Type(**인스턴스 유형**)**의 경우, **Native(**네이티브**)** 유형만 지원됩니다.
f) **OK(**확인**)**를 클릭합니다.

   Provisioning - *device name*(프로비저닝 - 디바이스 이름) 창이 표시됩니다.

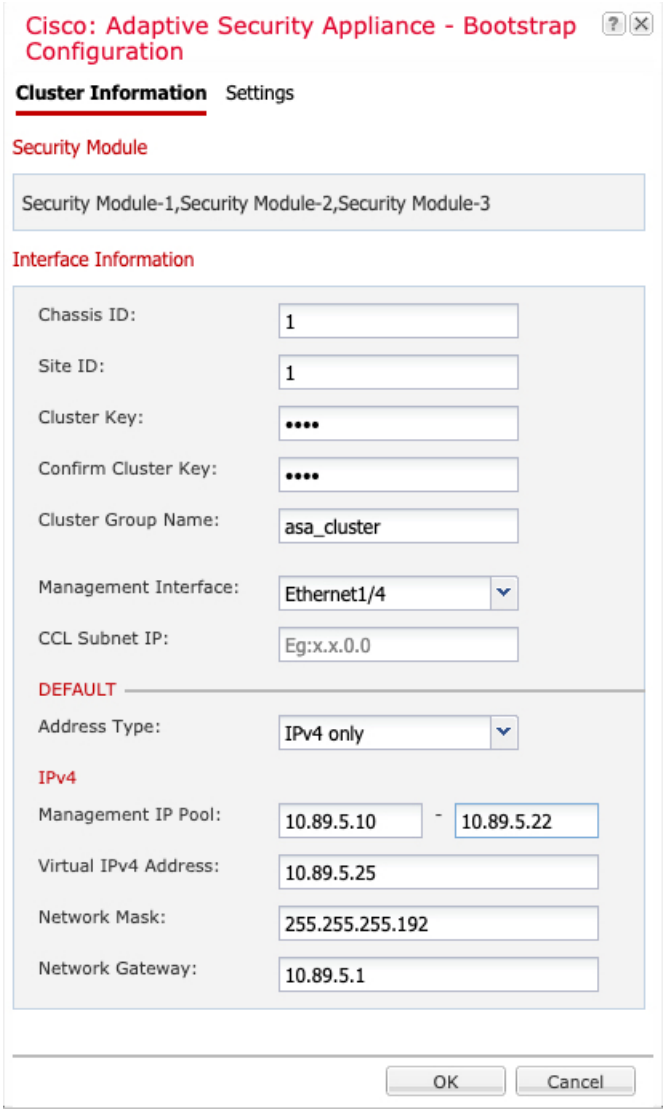
단계 **4** 이 클러스터에 할당할 인터페이스를 선택합니다.

유효한 모든 인터페이스가 기본적으로 할당되어 있습니다. 여러 클러스터 유형의 인터페이스를 지정했다면 하나를 제외하고 모두 선택 해제합니다.

단계 **5** 화면 중앙의 디바이스 아이콘을 클릭합니다.

초기 부트스트랩 설정을 구성할 수 있는 대화 상자가 표시됩니다. 이러한 설정은 초기 구축 전용 또는 재해 복구용입니다. 일반 작업 시에는 애플리케이션 CLI 구성에서 대부분의 값을 나중에 변경할 수 있습니다.

단계 **6** **Cluster Information(**클러스터 정보**)** 페이지에서 다음 작업을 수행합니다.



a) 섀시 간 클러스터링의 경우, **Chassis ID(**섀시 **ID)** 필드에 섀시 ID를 입력합니다. 클러스터의 각 섀시는 고유 ID를 사용해야 합니다.

   이 필드는 클러스터 제어 링크 Port-Channel 48에 멤버 인터페이스를 추가한 경우에만 나타납니다.

b) 사이트 간 클러스터링의 경우 이 섀시에 대해 **Site ID(**사이트 **ID)** 필드에 1~8의 사이트 ID를 입력합니다.

c) **Cluster Key(**클러스터 키**)** 필드에서 클러스터 제어 링크의 제어 트래픽에 대한 인증 키를 구성합니다.

   공유 비밀은 1자 ~ 63자로 된 ASCII 문자열입니다. 공유 비밀은 키를 생성하는 데 사용됩니다. 이 옵션은 연결 상태 업데이트 및 전달된 패킷을 비롯한 데이터 경로 트래픽에 영향을 미치지 않으며, 항상 일반 텍스트로 전송됩니다.

d) **Cluster Group Name(**클러스터 그룹 이름**)**(논리적 디바이스 구성의 클러스터 그룹 이름)을 설정합니다.

이름은 1자 ~ 38자로 된 ASCII 문자열이어야 합니다.

e) **Management Interface(**관리 인터페이스**)**를 선택합니다.

이 인터페이스는 논리적 디바이스를 관리하는 데 사용됩니다. 이 인터페이스는 섀시 관리 포트와 별개입니다.

f) (선택 사항) **CCL Subnet IP(CCL** 서브넷 **IP)**를 *a.b***.0.0**으로 설정합니다.

기본적으로 클러스터 제어 링크는 127.2.0.0/16 네트워크를 사용합니다. 그러나 일부 네트워킹 구축에서는 127.2.0.0/16 트래픽 통과를 허용하지 않습니다. 이 경우 루프백(127.0.0.0/8) 및 멀티캐스트(224.0.0.0/4) 및 내부 (169.254.0.0/16) 주소를 제외한 모든 /16 네트워크 주소를 클러스터용 고유 네트워크에 지정합니다. 값을 0.0.0.0으로 설정하는 경우 기본 네트워크가 사용됩니다.

섀시에서는 섀시 ID 및 슬롯 ID *a.b.chassis_id.slot_id*를 기준으로 하여 각 유닛에 대해 클러스터 제어 링크 인터페이스 IP 주소를 자동 생성합니다.

g) 관리 인터페이스의 **Address Type(**주소 유형**)**을 선택합니다.

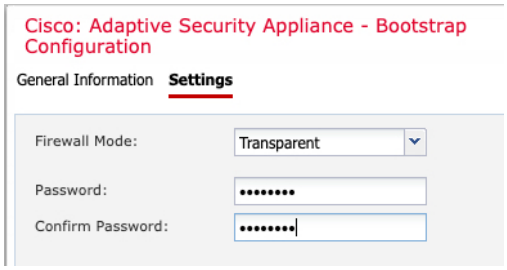
이 정보는 ASA 구성에서 관리 인터페이스를 구성하는 데 사용됩니다. 다음 정보를 설정합니다.

- **Management IP Pool(**관리 **IP** 풀**)** - 시작 및 종료 주소를 하이픈으로 구분하여 입력해 로컬 IP 주소의 풀을 구성합니다. 이 중 하나는 인터페이스의 각 클러스터 유닛에 할당됩니다.

  최소한 클러스터에 있는 유닛 수에 상응하는 개수의 주소를 포함해야 합니다. Firepower 9300에서는 모든 모듈 슬롯을 채우지 않은 경우에도 섀시당 3개 주소를 포함해야 합니다. 클러스터를 확장하려는 경우, 추가 주소를 포함하십시오. 현재 제어 유닛에 속하는 가상 IP 주소(기본 클러스터 IP 주소)는 이러한 풀에 속하지 않습니다. 따라서 동일한 네트워크에서 기본 클러스터 IP 주소에 대한 IP 주소를 예약해 두어야 합니다. IPv4 및/또는 IPv6 주소를 사용할 수 있습니다.

- **Network Mask(**네트워크 마스크**)** 또는 **Prefix Length(**접두사 길이**)**
- 네트워크 게이트웨이
- **Virtual IP address(**가상 **IP** 주소**)** — 현재 제어 유닛의 관리 IP 주소를 설정합니다. 이 IP 주소는 같은 네트워크의 클러스터 풀 주소로 있어야 하지만 풀의 일부는 아닙니다.

단계 **7** **Settings(**설정**)** 페이지에서 다음 작업을 완료합니다.



a) **Firewall Mode(**방화벽 모드**)** 드롭다운 목록에서 **Transparent(**투명**)** 또는 **Routed(**라우팅됨**)**를 선택합니다.

라우팅 모드에서 FTD는 네트워크의 라우터 홉으로 간주됩니다. 라우팅할 각 인터페이스가 다른 서브넷에 있습니다. 이와 반대로 투명 방화벽은 "비활성 엔드포인트(bump in the wire)" 또는 "은폐형 방화벽(stealth firewall)" 같은 역할을 수행하는 레이어 2 방화벽이며, 연결된 디바이스에 대한 라우터 홉으로 표시되지 않습니다.

방화벽 모드는 초기 구축 시에만 설정됩니다. 부트스트랩 설정을 다시 적용하는 경우에는 이 설정이 사용되지 않습니다.

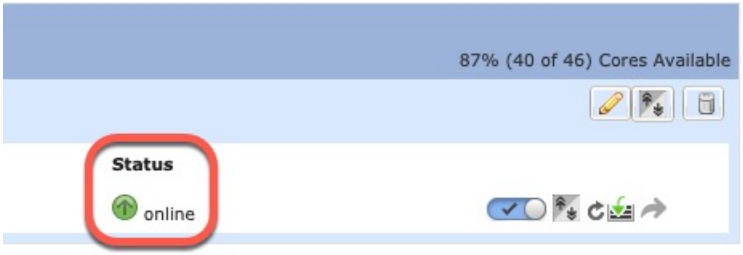
b) 관리자 및 비밀번호 활성화에 대해 **Password(**비밀번호**)**를 입력하고 확인합니다.

비밀번호를 복구할 때는 사전 구성된 ASA 관리자가 있으면 유용합니다. FXOS 액세스 권한이 있다면 관리자 비밀번호를 잊어버린 경우 재설정할 수 있습니다.

단계 **8** **OK(**확인**)**를 클릭하여 구성 대화 상자를 닫습니다.

단계 **9** **Save(**저장**)**를 클릭합니다.

섀시에서 지정된 소프트웨어 버전을 다운로드하고 부트스트랩 구성 및 관리 인터페이스 설정을 애플리케이션 인스턴스에 푸시하여 논리적 디바이스를 구축합니다. **Logical Devices(**논리적 디바이스**)** 페이지에서 새 논리적 디바이스의 상태를 확인합니다. 논리적 디바이스의 상태가 **online(**온라인**)**으로 표시되면 나머지 클러스터 섀시를 추가할 수도 있고, 섀시 내 클러스터링의 경우 애플리케이션 내에서 클러스터 구성을 시작할 수도 있습니다. 프로세스의 일부로 "보안 모듈이 응답하지 않음" 상태가 표시될 수 있습니다. 이 상태는 정상이며 일시적입니다.



단계 **10** 섀시 간 클러스터링의 경우, 다음 섀시를 클러스터에 추가합니다.

a) Firepower Chassis Manager의 첫 번째 섀시에서 오른쪽 상단에 있는 **Show Configuration(**구성 표시**)** 아이콘을 클릭하여 표시된 클러스터 구성을 복사합니다.

b) 다음 섀시에 있는 Firepower Chassis Manager에 연결하고 이 절차에 따라 논리적 디바이스를 추가합니다.

c) **I want to:(**수행할 작업**:) > Join an Existing Cluster(**기존 클러스터에 조인**)**를 선택합니다.

d) **OK(**확인**)**를 클릭합니다.

e) **Copy Cluster Details(**클러스터 세부사항 복사**)** 상자에서 첫 번째 섀시의 클러스터 구성에 붙여 넣고 **OK(**확인**)**를 클릭합니다.

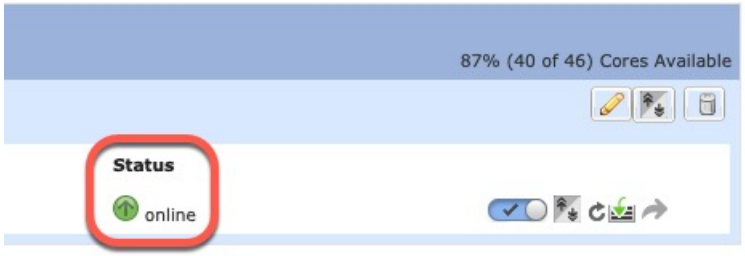
f) 화면 중앙의 디바이스 아이콘을 클릭합니다. 클러스터 정보는 대부분 미리 채워지지만 다음 설정은 변경해야 합니다.

- **Chassis ID(**섀시 **ID)** - 고유한 섀시 ID를 입력합니다.
- **Site ID(**사이트 **ID)** - 올바른 사이트 ID를 입력합니다.
- **Cluster Key(**클러스터 키**)** - (미리 채워지지 않음) 동일한 클러스터 키를 입력합니다.

OK(확인**)**를 클릭합니다.

g) **Save(**저장**)**를 클릭합니다.

섀시에서 지정된 소프트웨어 버전을 다운로드하고 부트스트랩 구성 및 관리 인터페이스 설정을 애플리케이션 인스턴스에 푸시하여 논리적 디바이스를 구축합니다. **Logical Devices(**논리적 디바이스**)** 페이지에서 각 클러스터 멤버의 새 논리적 디바이스의 상태를 확인합니다. 각 클러스터 멤버의 논리적 디바이스의 **Status(**상태**)**가 **online(**온라인**)**으로 표시되면 애플리케이션 내에서 클러스터 구성을 시작할 수 있습니다. 프로세스의 일부로 "보안 모듈이 응답하지 않음" 상태가 표시될 수 있습니다. 이 상태는 정상이며 일시적입니다.



단계 **11** 제어 유닛 ASA에 연결하여 클러스터링 컨피그레이션을 맞춤화합니다.

클러스터 멤버 더 추가

ASA 클러스터 멤버를 추가하거나 교체합니다.

참고 이 절차는 섀시 추가 또는 교체 시에만 적용됩니다. 클러스터링이 이미 활성화된 Firepower 9300에 모듈을 추가하거나 교체하는 경우에는 모듈이 자동으로 추가됩니다.

시작하기 전에

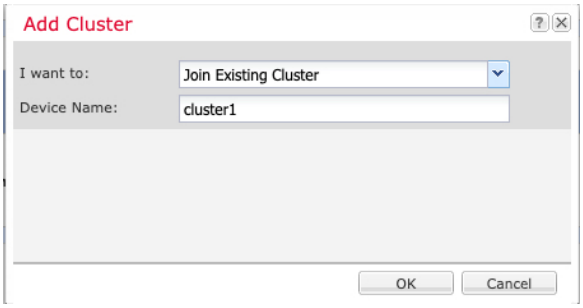
- 기존 클러스터에서 이 새 멤버의 관리 IP 주소 풀에 충분한 IP 주소가 있는지 확인하십시오. IP 주소가 충분하지 않은 경우, 이 새 멤버를 추가하기 전에 각 섀시에서 기존 클러스터 부트스트랩 구성을 수정해야 합니다. 이러한 변경으로 인해 논리적 디바이스가 재시작됩니다.
- 인터페이스 구성은 새 섀시에서 동일해야 합니다. FXOS 섀시 구성 내보내기와 가져오기를 통해 이 프로세스를 더 쉽게 수행할 수 있습니다.
- 다중 컨텍스트 모드의 경우 첫 번째 클러스터 멤버의 ASA 애플리케이션에서 다중 컨텍스트 모드를 활성화합니다. 그러면 추가 클러스터 멤버가 다중 컨텍스트 모드 구성을 자동으로 상속합니다.

프로시저

단계 **1** 기존 클러스터 Firepower Chassis Manager에서 **Logical Devices(**논리적 디바이스**)**를 선택하여 **Logical Devices(**논리적 디바이스**)** 페이지를 엽니다.

단계 **2** 오른쪽 상단의 구성 표시 아이콘( )를 클릭하여 표시되는 클러스터 구성을 복사합니다.

단계 **3** 새 섀시에서 Firepower Chassis Manager에 연결한 다음 **Add(**추가**)** > **Cluster(**클러스터**)**를 클릭합니다.



단계 **4** **I want to:(**수행할 작업**:)** > **Join an Existing Cluster(**기존 클러스터에 조인**)**를 선택합니다.

단계 **5** **Device Name(**디바이스 이름**)**에 논리적 디바이스의 이름을 입력합니다.

단계 **6** **OK(**확인**)**를 클릭합니다.

단계 **7** **Copy Cluster Details(**클러스터 세부사항 복사**)** 상자에서 첫 번째 섀시의 클러스터 구성에 붙여 넣고 **OK(**확인**)**를 클릭합니다.

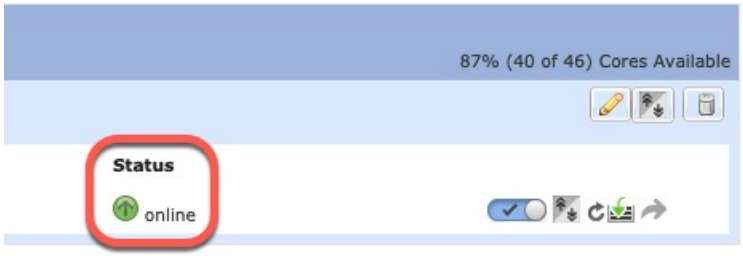
단계 **8** 화면 중앙의 디바이스 아이콘을 클릭합니다. 클러스터 정보는 대부분 미리 채워지지만 다음 설정은 변경해야 합니다.

- **Chassis ID(**섀시 **ID)** - 고유한 섀시 ID를 입력합니다.
- **Site ID(**사이트 **ID)** - 올바른 사이트 ID를 입력합니다.
- **Cluster Key(**클러스터 키**)** - (미리 채워지지 않음) 동일한 클러스터 키를 입력합니다.

OK(확인**)**를 클릭합니다.

단계 **9** **Save(**저장**)**를 클릭합니다.

섀시에서 지정된 소프트웨어 버전을 다운로드하고 부트스트랩 구성 및 관리 인터페이스 설정을 애플리케이션 인스턴스에 푸시하여 논리적 디바이스를 구축합니다. **Logical Devices(**논리적 디바이스**)** 페이지에서 각 클러스터 멤버의 새 논리적 디바이스의 상태를 확인합니다. 각 클러스터 멤버의 논리적 디바이스의 **Status(**상태**)**가 **online(**온라인**)**으로 표시되면 애플리케이션 내에서 클러스터 구성을 시작할 수 있습니다. 프로세스의 일부로 "보안 모듈이 응답하지 않음" 상태가 표시될 수 있습니다. 이 상태는 정상이며 일시적입니다.



FTD 클러스터 추가

네이티브 모드에서 단일 Firepower 9300 섀시를 섀시 내 클러스터로 추가하거나 섀시 간 클러스터링용으로 여러 섀시를 추가할 수 있습니다.

다중 인스턴스 모드에서 단일 Firepower 9300 섀시에 하나 이상의 클러스터를 섀시 내 클러스터로 추가하거나(각 모듈에 인스턴스를 포함해야 함) 섀시 간 클러스터링용으로 여러 섀시에 하나 이상의 클러스터를 추가할 수 있습니다.

섀시 간 클러스터링의 경우 각 섀시를 개별적으로 구성해야 합니다. 섀시 하나에 클러스터를 추가한 다음 쉽게 구축하기 위해 첫 번째 섀시의 부트스트랩 구성을 다음 섀시에 복사합니다.

FTD 클러스터 생성

Firepower 4100/9300 섀시 수퍼바이저에서 클러스터를 손쉽게 구축할 수 있습니다. 모든 초기 구성은 유닛마다 자동으로 생성됩니다.

섀시 간 클러스터링의 경우 각 섀시를 개별적으로 구성해야 합니다. 섀시 하나에 클러스터를 구축한 다음 쉽게 구축하기 위해 첫 번째 섀시의 부트스트랩 컨피그레이션을 다음 섀시에 복사합니다.

모듈을 설치하지 않은 경우에도 Firepower 9300 섀시의 3개 모듈 슬롯 모두 또는 컨테이너 인스턴스, 각 슬롯의 컨테이너 인스턴스에 대해 클러스터링을 활성화해야 합니다. 3개 모듈을 모두 구성하지 않은 경우 클러스터가 나타나지 않습니다.

시작하기 전에

- Cisco.com에서 논리적 디바이스에 사용할 애플리케이션 이미지를 다운로드한 다음, 해당 이미지를 Firepower 4100/9300 섀시에 업로드합니다.
- 컨테이너 인스턴스의 경우 기본 프로필을 사용하지 않으려면 컨테이너 인스턴스에 대한 리소스 프로파일 추가, 160 페이지에 따라 리소스 프로필을 추가합니다.
- 컨테이너 인스턴스의 경우 컨테이너 인스턴스를 처음으로 설치하기 전에 디스크가 올바른 형식을 갖도록 보안 모듈/엔진을 다시 초기화해야 합니다. **Security Modules(**보안 모듈**)** 또는 **Security Engine(**보안 엔진**)**을 선택하고 다시 초기화 아이콘(⟲)을 클릭합니다. 기존 논리적 디바이스가 삭제된 후에 새 디바이스로 재설치되며 로컬 애플리케이션 구성은 손실됩니다. 기본 인스턴스를 컨테이너 인스턴스로 교체할 때는 어떤 경우든 기본 인스턴스를 삭제해야 합니다. 기본 인스턴스를 컨테이너 인스턴스로 자동 마이그레이션할 수는 없습니다. 자세한 내용은 보안 모듈/엔진 확인 다시 초기화, 300 페이지를 참조하십시오.
- 다음 정보를 수집합니다.
  - 관리 인터페이스 ID, IP 주소, 네트워크 마스크
  - 게이트웨이 IP 주소
  - FMC 선택한 IP 주소 및/또는 NAT ID
  - DNS 서버 IP 주소
  - FTD 호스트 이름 및 도메인 이름

프로시저

단계 **1** 인터페이스를 구성합니다.

단계 **2** **Logical Devices(**논리적 디바이스**)**를 선택합니다.

단계 **3** **Add(**추가**)** > **Cluster(**클러스터**)**를 클릭하고 다음 파라미터를 설정합니다.

그림 ***12:*** 네이티브 클러스터

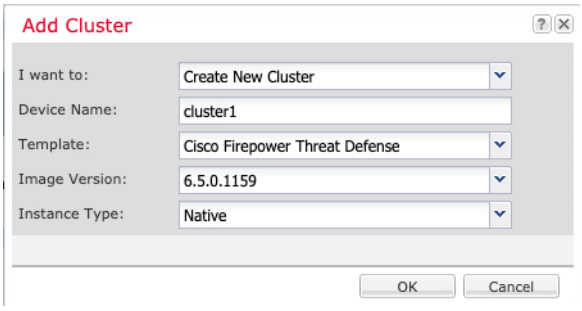
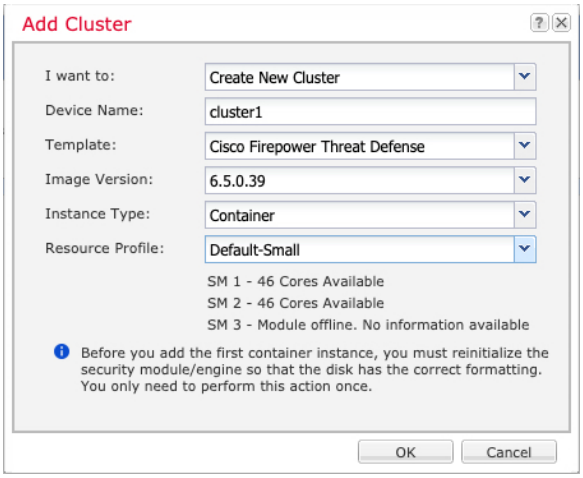
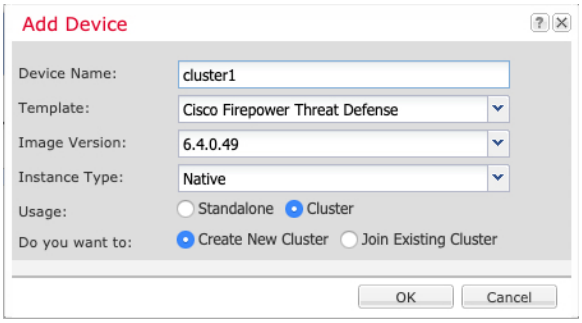


그림 *13:* 다중 인스턴스 클러스터





a) **I want to:(**수행할 작업**:)** > **Create New Cluster(**새 클러스터 생성**)**를 선택합니다.

b) **Device Name(**디바이스 이름**)**을 입력합니다.

이 이름은 내부적으로 섀시 수퍼바이저가 관리 설정을 구성하고 인터페이스를 할당하는 데 사용됩니다. 이는 애플리케이션 구성에 사용되는 디바이스 이름이 아닙니다.

c) **Template(**템플릿**)**에서 **Cisco Firepower Threat Defense**를 선택합니다.

d) **Image Version(**이미지 버전**)**을 선택합니다.

e) **Instance Type(**인스턴스 유형**)**의 경우 **Native(**네이티브**)** 또는 **Container(**컨테이너**)**를 선택합니다.

네이티브 인스턴스는 보안 모듈/엔진의 모든 리소스(CPU, RAM 및 디스크 공간)를 사용합니다. 따라서 하나의 기본 인스턴스만 설치할 수 있습니다. 컨테이너 인스턴스는 보안 모듈/엔진의 리소스 하위 집합을 사용합니다. 따라서 여러 개의 컨테이너 인스턴스를 설치할 수 있습니다.

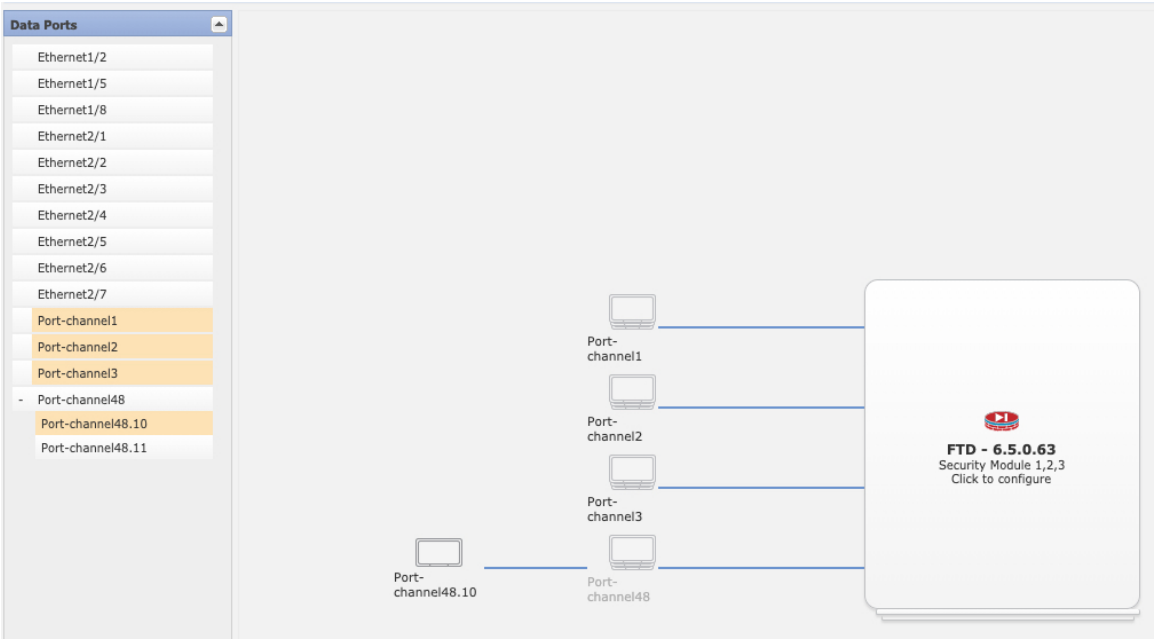
f) (컨테이너 인스턴스만 해당) **Resource Type(**리소스 유형**)**의 드롭다운 목록에서 리소스 프로파일 중 하나를 선택합니다.

Firepower 9300의 경우 이 프로파일은 보안 모듈의 각 인스턴스에 적용됩니다. 이 절차에서 나중에 보안 모듈별로 서로 다른 프로파일을 설정할 수 있습니다. 예를 들어 다른 보안 모듈 유형을 사용하면서 더 성능이 낮은 모델에서 더 많은 CPU를 사용할 수도 있습니다. 클러스터를 생성하기 전에 올바른 프로파일을 선택하는 것이 좋습니다. 새 프로파일을 생성해야 하는 경우 클러스터 생성을 취소하고 컨테이너 인스턴스에 대한 리소스 프로파일 추가, 160 페이지을 사용해 하나를 추가합니다.

g) **OK(**확인**)**를 클릭합니다.

Provisioning - *device name*(프로비저닝 - 디바이스 이름) 창이 표시됩니다.

단계 **4** 이 클러스터에 할당할 인터페이스를 선택합니다.



네이티브 모드 클러스터링의 경우: 유효한 모든 인터페이스가 기본적으로 할당되어 있습니다. 여러 클러스터 유형의 인터페이스를 지정했다면 하나를 제외하고 모두 선택 해제합니다.

다중 인스턴스 클러스터링의 경우: 클러스터에 할당할 각 데이터 인터페이스를 선택하고 클러스터 유형 포트 채널 또는 포트 채널 하위 인터페이스도 선택합니다.

단계 **5** 화면 중앙의 디바이스 아이콘을 클릭합니다.

초기 부트스트랩 설정을 구성할 수 있는 대화 상자가 표시됩니다. 이러한 설정은 초기 구축 전용 또는 재해 복구용입니다. 일반 작업 시에는 애플리케이션 CLI 구성에서 대부분의 값을 나중에 변경할 수 있습니다.

단계 **6** **Cluster Information(**클러스터 정보**)** 페이지에서 다음 작업을 수행합니다.

그림 *14:* 네이티브 클러스터

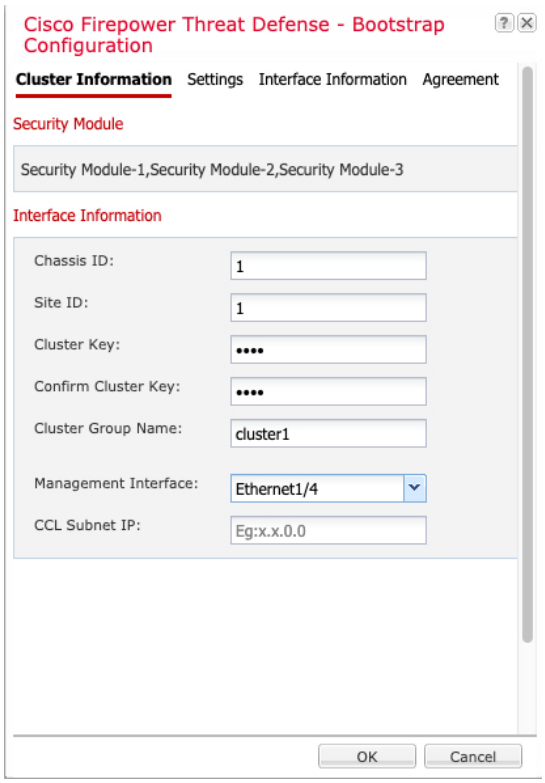
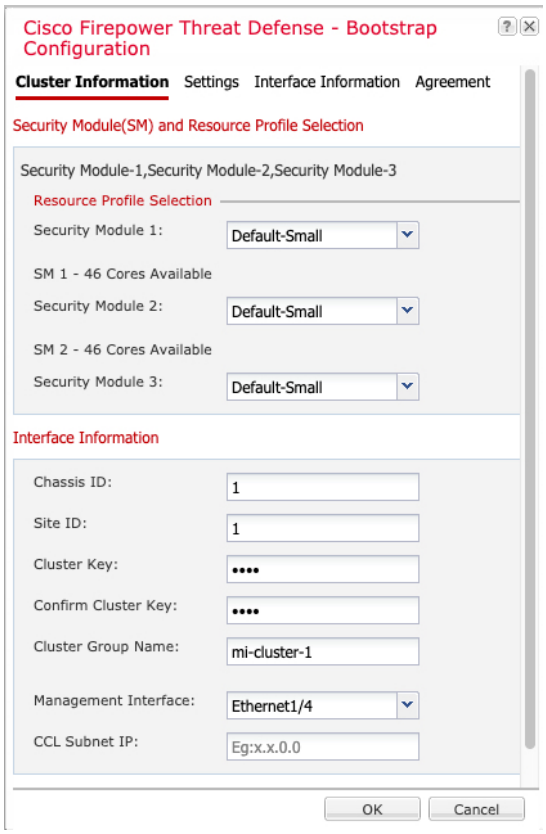


그림 *15:* 다중 인스턴스 클러스터



a) (Firepower 9300 컨테이너 인스턴스만 해당) 보안 모듈**(SM)** 및 리소스 프로파일 선택 영역에서 모듈별로 다른 리소스 프로파일을 설정할 수 있습니다. 예를 들어 다른 보안 모듈 유형을 사용하면서 더 성능이 낮은 모델에서 더 많은 CPU를 사용할 수도 있습니다.

b) 섀시 간 클러스터링의 경우, **Chassis ID(**섀시 **ID)** 필드에 섀시 ID를 입력합니다. 클러스터의 각 섀시는 고유 ID를 사용해야 합니다.

이 필드는 클러스터 제어 링크 Port-Channel 48에 멤버 인터페이스를 추가한 경우에만 나타납니다.

c) 사이트 간 클러스터링의 경우 이 섀시에 대해 **Site ID(**사이트 **ID)** 필드에 1~8의 사이트 ID를 입력합니다. FlexConfig 기능. 디렉터 현지화, 사이트 이중화, 클러스터 플로우 이동성 같은 이중화 및 안정성을 개선하기 위한 추가적인 사이트 간 클러스터 맞춤화는 FMC FlexConfig 기능을 사용하는 경우에만 구성 가능합니다.

d) **Cluster Key(**클러스터 키**)** 필드에서 클러스터 제어 링크의 제어 트래픽에 대한 인증 키를 구성합니다.

공유 비밀은 1자 ~ 63자로 된 ASCII 문자열입니다. 공유 비밀은 키를 생성하는 데 사용됩니다. 이 옵션은 연결 상태 업데이트 및 전달된 패킷을 비롯한 데이터 경로 트래픽에 영향을 미치지 않으며, 항상 일반 텍스트로 전송됩니다.

e) **Cluster Group Name(**클러스터 그룹 이름**)**(논리적 디바이스 구성의 클러스터 그룹 이름)을 설정합니다.

이름은 1자 ~ 38자로 된 ASCII 문자열이어야 합니다.

f) **Management Interface(**관리 인터페이스**)**를 선택합니다.

이 인터페이스는 논리적 디바이스를 관리하는 데 사용됩니다. 이 인터페이스는 섀시 관리 포트와 별개입니다.

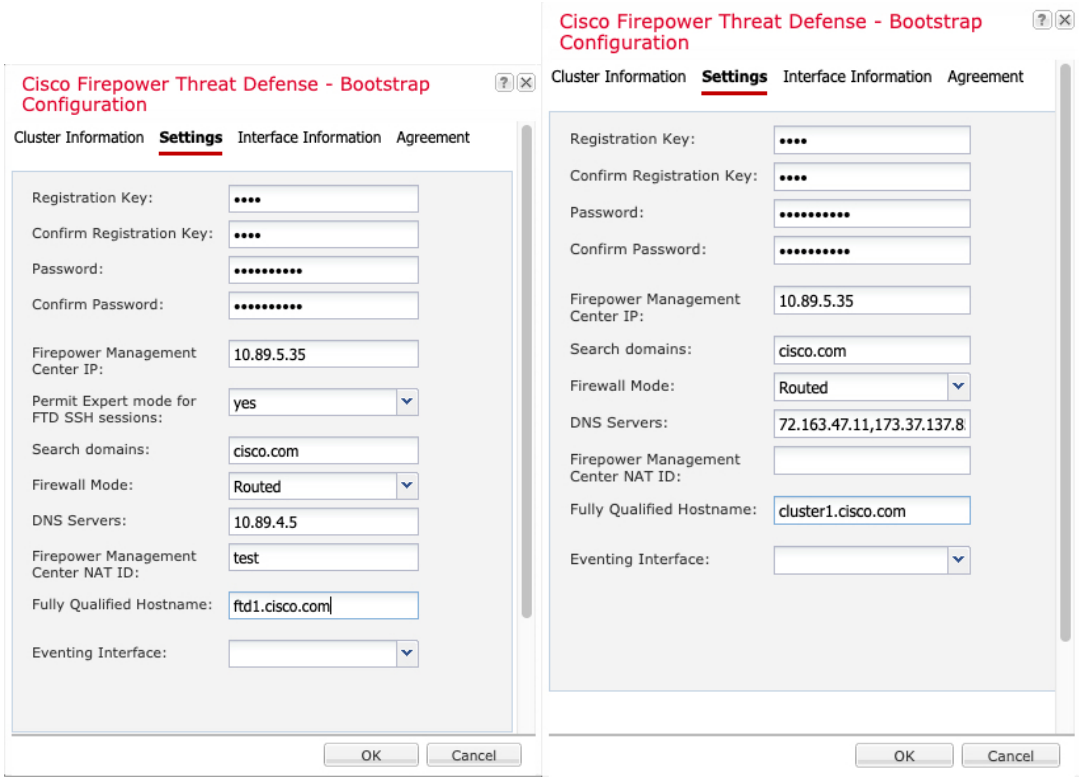
하드웨어 바이패스 지원 인터페이스를 Management(관리) 인터페이스로 할당할 경우 그러한 할당이 의도적인지를 확인하는 경고 메시지가 표시됩니다.

g) (선택 사항) **CCL Subnet IP(CCL** 서브넷 **IP)**를 ***a.b*.0.0**으로 설정합니다.

기본적으로 클러스터 제어 링크는 127.2.0.0/16 네트워크를 사용합니다. 그러나 일부 네트워킹 구축에서는 127.2.0.0/16 트래픽 통과를 허용하지 않습니다. 이 경우 루프백(127.0.0.0/8) 및 멀티캐스트(224.0.0.0/4) 및 내부 (169.254.0.0/16) 주소를 제외한 모든 /16 네트워크 주소를 클러스터용 고유 네트워크에 지정합니다. 값을 0.0.0.0으로 설정하는 경우 기본 네트워크가 사용됩니다.

섀시에서는 섀시 ID 및 슬롯 ID *a.b.chassis_id.slot_id*를 기준으로 하여 각 유닛에 대해 클러스터 제어 링크 인터페이스 IP 주소를 자동 생성합니다.

단계 **7** **Settings(**설정**)** 페이지에서 다음 작업을 완료합니다.



a) **Registration Key(**등록 키**)**필드에 등록하는 동안 FMC와 클러스터 멤버 간에 공유할 키를 입력합니다.

이 키에 대해 1~37자의 텍스트 문자열을 선택할 수 있습니다. FTD를 추가하는 경우 FMC에 동일한 키를 입력합니다.

b) FTD 관리 사용자가 CLI에 액세스할 때 사용할 **Password(**비밀번호**)**를 입력합니다.

c) **Firepower Management Center IP** 필드에 FMC를 관리하기 위한 IP 주소를 입력합니다. FMC IP 주소를 알 수 없는 경우, 이 필드를 비워두고 **Firepower Management Center NAT ID** 필드에 암호를 입력합니다.

d) (선택 사항) 컨테이너 인스턴스의 경우, **Permit Export mode from FTD SSH sessions(FTD SSH** 세션에서 전문가 모드 허용**)**에 대해 **Yes(**예**)** 또는 **No(**아니요**)**를 선택합니다. 전문가 모드에서는 고급 트러블슈팅을 위한 FTD 셸 액세스 기능이 제공됩니다.

이 옵션에 대해 **Yes(**예**)**를 선택하는 경우 SSH 세션에서 컨테이너 인스턴스에 직접 액세스할 수 있는 사용자가 전문가 모드를 시작할 수 있습니다. **No(**아니요**)**를 선택하는 경우에는 FXOS CLI에서 컨테이너 인스턴스에 액세스할 수 있는 사용자만 전문가 모드를 시작할 수 있습니다. 각 인스턴스를 더욱 명확하게 격리할 수 있도록 **No(**아니요**)**를 선택하는 것이 좋습니다.

문서에 설명되어 있는 절차에 따라 Expert 모드가 필요하다고 알려주는 경우 또는 Cisco Technical Assistance Center에서 사용하도록 요청하는 경우에만 Expert 모드를 사용합니다. 이 모드를 설정하려면 FTD CLI에서 **expert** 명령을 사용합니다.

e) (선택 사항) **Search Domains(**검색 도메인**)** 필드에 관리 네트워크의 쉼표로 구분된 검색 도메인 목록을 입력합니다.

f) (선택 사항) **Firewall Mode(**방화벽 모드**)** 드롭다운 목록에서 **Transparent(**투명**)** 또는 **Routed(**라우팅됨**)**를 선택합니다.

라우팅 모드에서 FTD는 네트워크의 라우터 홉으로 간주됩니다. 라우팅할 각 인터페이스가 다른 서브넷에 있습니다. 이와 반대로 투명 방화벽은 "비활성 엔드포인트(bump in the wire)" 또는 "은폐형 방화벽(stealth firewall)" 같은 역할을 수행하는 레이어 2 방화벽이며, 연결된 디바이스에 대한 라우터 홉으로 표시되지 않습니다.

방화벽 모드는 초기 구축 시에만 설정됩니다. 부트스트랩 설정을 다시 적용하는 경우에는 이 설정이 사용되지 않습니다.

g) (선택 사항) **DNS Servers(DNS** 서버**)** 필드에 쉼표로 구분된 DNS 서버 목록을 입력합니다.

예를 들어, FMC의 호스트 이름을 지정하는 경우, FTD에서는 DNS를 사용합니다.

h) (선택 사항) 새 디바이스로 클러스터를 추가할 때 FMC에도 입력할 암호를 **Firepower Management Center NAT ID** 필드에 입력합니다.

일반적으로 라우팅 목적과 인증 두 가지 목적에 IP 주소(등록 키와 함께)가 모두 필요합니다. FMC는 디바이스 IP 주소를 지정하고 디바이스는 FMC IP 주소를 지정합니다. 그러나 라우팅을 위한 최소 요구 사항인 IP 주소 중 하나만 알고 있는 경우, 초기 통신에 대한 신뢰를 설정하고 올바른 등록 키를 조회하려면 연결의 양쪽에서 고유 NAT ID도 지정해야 합니다. 1~37자의 임의의 텍스트 문자열을 NAT ID로 지정할 수 있습니다. FMC 및 디바이스는 등록 키 및 NAT ID(IP 주소 대신)를 사용하여 초기 등록을 인증하고 권한을 부여합니다.

i) (선택 사항) **Fully Qualified Hostname(**정규화된 호스트 이름**)** 필드에 FTD 디바이스의 정규화된 이름을 입력합니다.

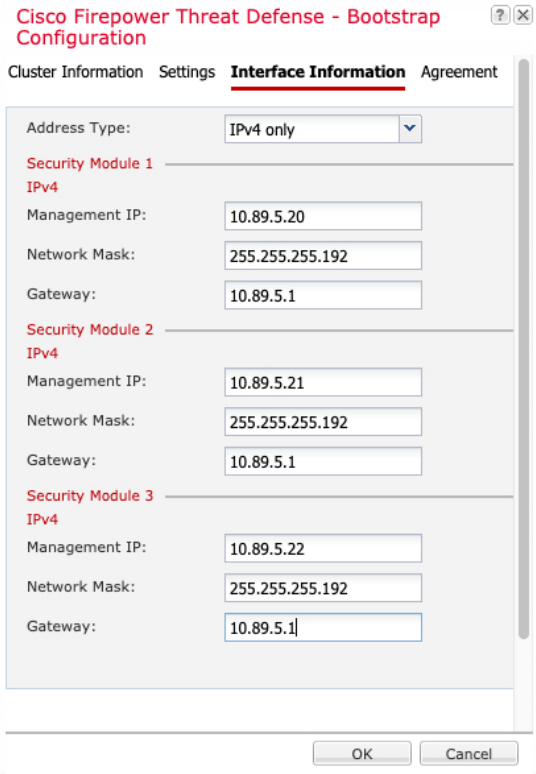
유효한 문자는 a부터 z까지의 문자, 0과 9 사이의 숫자, 점(.) 및 하이픈(-)입니다. 최대 문자 수는 253자입니다.

j) (선택 사항) **Eventing Interface(Eventing** 인터페이스**)** 드롭다운 목록에서 이벤트가 전송되어야 할 인터페이스를 선택합니다. 인터페이스가 지정되지 않은 경우, 관리 인터페이스가 사용됩니다.

이벤트에 사용할 별도의 인터페이스를 지정하려면 인터페이스를 *Firepower* 이벤트 처리 인터페이스로 구성해야 합니다. 하드웨어 바이패스 지원 인터페이스를 Eventing(이벤트) 인터페이스로 할당하는 경우 그러한 할당이 의도적인지를 확인하는 경고 메시지가 표시됩니다.

단계 **8** **Interface Information(**인터페이스 정보**)** 페이지에서 클러스터의 각 보안 모듈의 관리 IP 주소를 구성합니다. **Address Type(**주소 유형**)** 드롭다운 목록에서 주소 유형을 선택한 다음 각 보안 모듈에 대해 다음 작업을 수행합니다.

참고 모듈을 설치하지 않은 경우에도 섀시의 3개 모듈 슬롯 모두에 대해 IP 주소를 설정해야 합니다. 3개 모듈을 모두 구성하지 않은 경우 클러스터가 나타나지 않습니다.



a) **Management IP(**관리 **IP)** 필드에서 IP 주소를 구성합니다.

각 모듈에 대해 동일한 네트워크에서 고유한 IP 주소를 지정합니다.

b) **Network Mask(**네트워크 마스크**)** 또는 **Prefix Length(**접두사 길이**)**를 입력합니다.

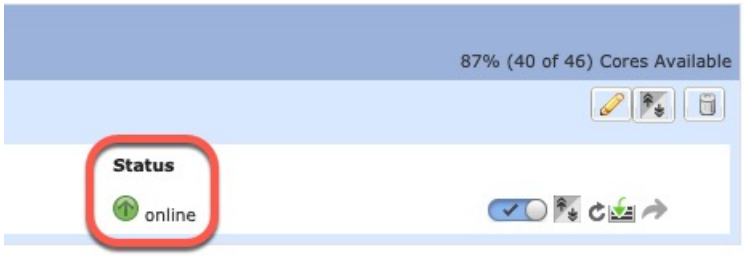
c) **Network Gateway(**네트워크 게이트웨이**)** 주소를 입력합니다.

단계 **9** **Agreement(**계약**)** 탭에서 EULA(End User License Agreement)를 읽고 내용에 동의해야 합니다.

단계 **10** **OK(**확인**)**를 클릭하여 구성 대화 상자를 닫습니다.

단계 **11** **Save(**저장**)**를 클릭합니다.

섀시에서 지정된 소프트웨어 버전을 다운로드하고 부트스트랩 구성 및 관리 인터페이스 설정을 애플리케이션 인스턴스에 푸시하여 논리적 디바이스를 구축합니다. **Logical Devices(**논리적 디바이스**)** 페이지에서 새 논리적 디바이스의 상태를 확인합니다. 논리적 디바이스의 상태가 **online(**온라인**)**으로 표시되면 나머지 클러스터 섀시를 추가할 수도 있고, 섀시 내 클러스터링의 경우 애플리케이션 내에서 클러스터 구성을 시작할 수도 있습니다. 프로세스의 일부로 "보안 모듈이 응답하지 않음" 상태가 표시될 수 있습니다. 이 상태는 정상이며 일시적입니다.



단계 **12** 섀시 간 클러스터링의 경우, 다음 섀시를 클러스터에 추가합니다.

a) Firepower Chassis Manager의 첫 번째 섀시에서 오른쪽 상단에 있는 **Show Configuration(**구성 표시**)** 아이콘을 클릭하여 표시된 클러스터 구성을 복사합니다.

b) 다음 섀시에 있는 Firepower Chassis Manager에 연결하고 이 절차에 따라 논리적 디바이스를 추가합니다.

c) **I want to:(**수행할 작업**:) > Join an Existing Cluster(**기존 클러스터에 조인**)**를 선택합니다.

d) **OK(**확인**)**를 클릭합니다.

e) **Copy Cluster Details(**클러스터 세부사항 복사**)** 상자에서 첫 번째 섀시의 클러스터 구성에 붙여 넣고 **OK(**확인**)**를 클릭합니다.

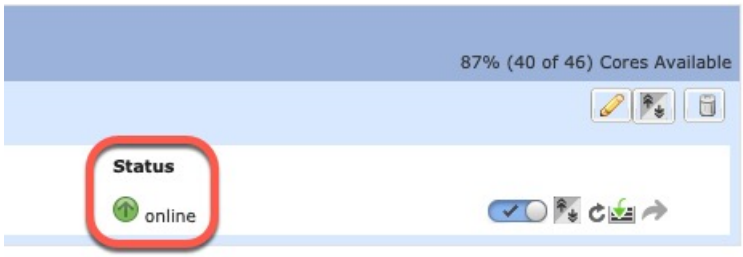
f) 화면 중앙의 디바이스 아이콘을 클릭합니다. 클러스터 정보는 대부분 미리 채워지지만 다음 설정은 변경해야 합니다.

- **Chassis ID(**섀시 **ID) -** 고유한 섀시 ID를 입력합니다.
- **Site ID(**사이트 **ID) -** 사이트 간 클러스터링의 경우 이 섀시에 대해 1~8 사이의 사이트 ID를 입력합니다. 디렉터 현지화, 사이트 이중화, 클러스터 플로우 이동성 같은 이중화 및 안정성을 개선하기 위한 추가적인 사이트 간 클러스터 맞춤화는 FMC FlexConfig 기능을 사용하는 경우에만 구성 가능합니다.
- **Cluster Key(**클러스터 키**) -** (미리 채워지지 않음) 동일한 클러스터 키를 입력합니다.
- **Management IP(**관리 **IP) -** 각 모듈의 관리 주소를 다른 클러스터 멤버와 동일한 네트워크에 있는 고유 IP 주소로 변경합니다.

OK(확인**)**를 클릭합니다.

g) **Save(**저장**)**를 클릭합니다.

섀시에서 지정된 소프트웨어 버전을 다운로드하고 부트스트랩 구성 및 관리 인터페이스 설정을 애플리케이션 인스턴스에 푸시하여 논리적 디바이스를 구축합니다. **Logical Devices(**논리적 디바이스**)** 페이지에서 각 클러스터 멤버의 새 논리적 디바이스의 상태를 확인합니다. 각 클러스터 멤버의 논리적 디바이스의 **Status(**상태**)**가 **online(**온라인**)**으로 표시되면 애플리케이션 내에서 클러스터 구성을 시작할 수 있습니다. 프로세스의 일부로 "보안 모듈이 응답하지 않음" 상태가 표시될 수 있습니다. 이 상태는 정상이며 일시적입니다.



단계 **13** 관리 IP 주소를 사용하여 제어 유닛을 FMC에 추가합니다.

FMC에 추가하기 전에 모든 클러스터 유닛이 FXOS에서 성공적으로 형성된 클러스터에 있어야 합니다.

그러면 FMC에서 데이터 유닛을 자동으로 탐지합니다.

클러스터 노드 추가

기존 클러스터에서 FTD 클러스터 노드를 추가하거나 교체합니다.FXOS에서 새 클러스터 노드를 추가할 때 FMC에서는 노드를 자동으로 추가합니다.

참고 이 절차의 FXOS 단계는 새 섀시 추가 시에만 적용됩니다. 클러스터링이 이미 활성화된 Firepower 9300에 새 모듈을 추가하는 경우에는 모듈이 자동으로 추가됩니다.

시작하기 전에

- 교체 시 기존 클러스터 노드를 FMC에서 삭제해야 합니다. 새 노드로 교체할 경우, 해당 유닛은 FMC에서 새 디바이스로 간주됩니다.

- 인터페이스 구성은 새 섀시에서 동일해야 합니다. FXOS 섀시 구성 내보내기와 가져오기를 통해 이 프로세스를 더 쉽게 수행할 수 있습니다.

프로시저

단계 **1** 이전에 FMC를 사용하여 FTD 이미지를 업그레이드한 경우 클러스터의 각 섀시에서 다음 단계를 수행합니다.

FMC에서 업그레이드할 때 FXOS 구성의 시작 버전이 업데이트되지 않았으며 독립형 패키지가 섀시에 설치되지 않았습니다. 새 노드가 올바른 이미지 버전을 사용하여 클러스터에 참여할 수 있도록 이러한 항목을 모두 수동으로 설정해야 합니다.

참고 패치 릴리스만 적용한 경우 이 단계를 건너뛸 수 있습니다. Cisco는 패치용 독립형 패키지를 제공하지 않습니다.

a) **System(**시스템**)** > **Updates(**업데이트**)** 페이지를 사용하여 섀시에 실행 중인 FTD 이미지를 설치합니다.

b) **Logical Devices(**논리적 디바이스**)**를 클릭하고 버전 설정 아이콘( )를 클릭합니다. 여러 모듈이 있는 Firepower 9300의 경우 각 모듈의 버전을 설정합니다.

Startup Version(시작 버전**)**에는 구축에 사용한 원래 패키지가 표시됩니다. **Current Version(**현재 버전**)**에는 업그레이드한 버전이 표시됩니다.

c) **New Version(**새 버전**)** 드롭다운 메뉴에서 업로드한 버전을 선택합니다. 이 버전은 표시된 현재 버전과 일치해야 하며, 새 버전과 일치하도록 시작 버전을 설정합니다.

d) 새 섀시에 새 이미지 패키지가 설치되어 있는지 확인합니다.

단계 **2** 기존 클러스터 섀시 Firepower Chassis Manager에서 **Logical Devices(**논리적 디바이스**)**를 클릭합니다.

단계 **3** 오른쪽 상단에 있는 설정 표시 아이콘을 클릭하여 표시된 클러스터 설정을 복사합니다.

단계 **4** 새 섀시에서 Firepower Chassis Manager에 연결한 다음 **Add(**추가**)** > **Cluster(**클러스터**)**를 클릭합니다.

단계 **5** **Device Name(**디바이스 이름**)**에 논리적 디바이스의 이름을 입력합니다.

단계 **6** **OK(**확인**)**를 클릭합니다.

단계 **7** **Copy Cluster Details(**클러스터 세부사항 복사**)** 상자에서 첫 번째 섀시의 클러스터 구성에 붙여 넣고 **OK(**확인**)**를 클릭합니다.

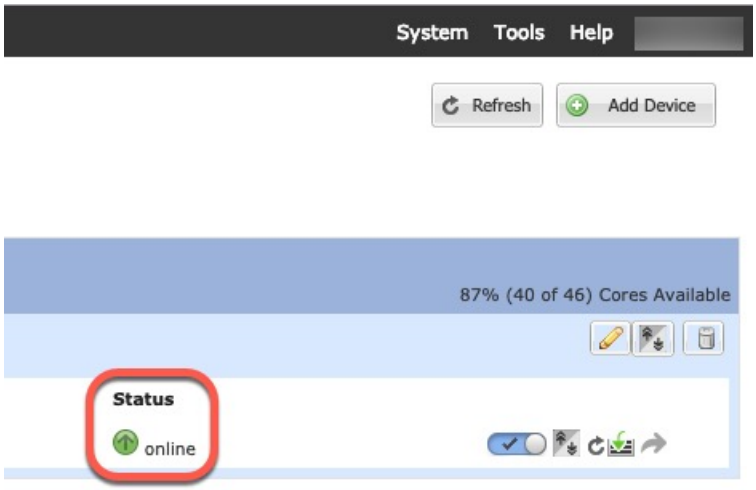
단계 **8** 화면 중앙의 디바이스 아이콘을 클릭합니다. 클러스터 정보는 대부분 미리 채워지지만 다음 설정은 변경해야 합니다.

- **Chassis ID(**섀시 **ID)** - 고유한 섀시 ID를 입력합니다.
- **Site ID(**사이트 **ID)** - 사이트 간 클러스터링의 경우 이 섀시에 대해 1~8 사이의 사이트 ID를 입력합니다. FMC FlexConfig 기능을 통해서만 이 기능을 구성할 수 있습니다.
- **Cluster Key(**클러스터 키**)** - (미리 채워지지 않음) 동일한 클러스터 키를 입력합니다.
- **Management IP(**관리 **IP)** - 각 모듈의 관리 주소를 다른 클러스터 멤버와 동일한 네트워크에 있는 고유 IP 주소로 변경합니다.

OK(확인**)**를 클릭합니다.

단계 **9** **Save(**저장**)**를 클릭합니다.

섀시에서 지정된 소프트웨어 버전을 다운로드하고 부트스트랩 구성 및 관리 인터페이스 설정을 애플리케이션 인스턴스에 푸시하여 논리적 디바이스를 구축합니다. **Logical Devices(**논리적 디바이스**)** 페이지에서 각 클러스터 멤버의 새 논리적 디바이스의 상태를 확인합니다. 각 클러스터 멤버의 논리적 디바이스의 **Status(**상태**)**가 **online(**온라인**)**으로 표시되면 애플리케이션 내에서 클러스터 구성을 시작할 수 있습니다. 프로세스의 일부로 "보안 모듈이 응답하지 않음" 상태가 표시될 수 있습니다. 이 상태는 정상이며 일시적입니다.



Radware DefensePro 구성

Cisco Firepower 4100/9300 섀시에서는 단일 블레이드에 있는 여러 서비스(예: 방화벽 및 서드파티 DDoS 애플리케이션)를 지원할 수 있습니다. 이 애플리케이션 및 서비스는 서비스 체인을 구성하기 위해 함께 연결될 수 있습니다.

Radware DefensePro 정보

현재 지원되는 서비스 체이닝 구성에서 서드파티 Radware DefensePro 가상 플랫폼을 설치하여 ASA 방화벽 또는 FTD 앞에서 실행할 수 있습니다. Radware DefensePro는 Firepower 4100/9300 섀시에서 DDoS(Distributed Denial-of-Service) 탐지 및 완화 기능을 제공하는 KVM 기반 가상 플랫폼입니다. 서비스 체이닝이 Firepower 4100/9300 섀시에서 활성화된 경우, 네트워크의 트래픽은 기본 ASA 또는 FTD 방화벽에 도달하기 전에 먼저 DefensePro 가상 플랫폼을 통과해야 합니다.

참고

- Radware DefensePro 가상 플랫폼은 *Radware vDP*(가상 DefensePro) 또는 간단하게 *vDP*라고도 합니다.
- Radware DefensePro 가상 플랫폼은 경우에 따라 링크 데코레이터라고도 합니다.

Radware DefensePro에 대한 사전 요구 사항

Firepower 4100/9300 섀시에 Radware DefensePro를 구축하기 전에 **etc/UTC** 표준 시간대로 NTP 서버를 사용하도록 Firepower 4100/9300 섀시를 구성해야 합니다. Firepower 4100/9300 섀시에서 날짜 및 시간을 설정하는 방법에 대한 자세한 내용은 날짜 및 시간 설정, 111 페이지을 참조하십시오.

서비스 체이닝 관련 지침

모델

- ASA - Radware DefensePro(vDP) 플랫폼은 다음 모델에서 ASA와 함께 지원됩니다.
  - Firepower 9300
  - Firepower 4110
  - Firepower 4115
  - Firepower 4120
  - Firepower 4125
  - Firepower 4140
  - Firepower 4145
  - Firepower 4150
- FTD- Radware DefensePro 플랫폼은 다음 모델에서 FTD와 함께 지원됩니다.
  - Firepower 9300
  - Firepower 4110 - 논리적 디바이스와 동시에 데코레이터를 구축해야 합니다. 논리적 디바이스가 이미 디바이스에 구성된 이후에는 데코레이터를 설치할 수 없습니다.
  - Firepower 4112
  - Firepower 4115
  - Firepower 4120 - 논리적 디바이스와 동시에 데코레이터를 구축해야 합니다. 논리적 디바이스가 이미 디바이스에 구성된 이후에는 데코레이터를 설치할 수 없습니다.
  - Firepower 4125

- Firepower 4140
- Firepower 4145
- Firepower 4150

추가 지침

- 서비스 체이닝은 섀시 간 클러스터 구성에서 지원되지 않습니다. 그러나 섀시 간 클러스터 시나리오의 독립형 구성에서는 Radware DefensePro(vDP) 애플리케이션을 구축할 수 있습니다.

독립형 논리적 디바이스에 **Radware DefensePro** 구성

다음 절차는 독립형 ASA 또는 FTD 논리적 디바이스의 앞에 있는 단일 서비스 체인에 Radware DefensePro를 설치하는 방법을 보여줍니다.

참고 이 절차가 끝날 때 vDP 애플리케이션을 설정하고 변경 사항을 커밋하면 논리적 디바이스(ASA 또는 FTD)가 재부팅됩니다.

Firepower 4120 또는 4140 보안 어플라이언스에서 ASA 앞에 Radware vDP를 설치하는 경우 FXOS CLI를 사용하여 데코레이터를 구축해야 합니다. Firepower 4100 디바이스에서 ASA 앞의 서비스 체인에 Radware DefensePro를 설치하고 구성하는 방법에 대한 전체 CLI 지침은 FXOS CLI 환경 설정 가이드를 참조하십시오.

시작하기 전에

- Cisco.com에서 vDP 이미지를 다운로드(Cisco.com에서 이미지 다운로드, 62 페이지 참조)한 다음 해당 이미지를 Firepower 4100/9300 섀시에 업로드합니다(Security Appliance에 이미지 업로드, 62 페이지 참조).
- 섀시 내 클러스터에서 독립형 구성으로 Radware DefensePro 애플리케이션을 구축할 수 있습니다. 섀시 내 클러스터링에 대해서는 인트라 섀시(Intra-Chassis) 클러스터에 Radware DefensePro 구성, 266 페이지 섹션을 참조하십시오.

프로시저

단계 **1** vDP용으로 별도의 관리 인터페이스를 사용하려는 경우 인터페이스를 활성화한 다음 실제 인터페이스 구성, 184 페이지에 따라 mgmt 유형으로 설정합니다. 그렇지 않은 경우에는 애플리케이션 관리 인터페이스를 공유할 수 있습니다.

단계 **2** **Logical Devices(**논리적 디바이스**)**를 선택하여 Logical Devices(논리적 디바이스) 페이지를 엽니다.

Logical Devices(논리적 디바이스) 페이지는 섀시에 구성되어 있는 논리적 디바이스 목록을 보여줍니다. 논리적 디바이스가 구성되어 있지 않은 경우, 이를 알리는 메시지가 표시됩니다.

단계 3 독립형 ASA 또는 FTD 논리적 디바이스를 생성합니다(독립형 ASA 추가, 225 페이지 또는 FMC에 대한 독립형 FTD 추가, 228 페이지 참조).

단계 4 **Decorators(**데코레이터**)** 영역에서 vDP를 선택합니다. Radware: Virtual DefensePro - Configuration(Radware: Virtual DefensePro - 구성) 창이 나타납니다. **General Information(**일반 정보**)** 탭에서 다음 필드를 구성합니다.

단계 5 둘 이상의 vDP 버전을 Firepower 4100/9300 섀시에 업로드한 경우, 사용할 버전을 **Version(**버전**)** 드롭다운에서 선택합니다.

단계 6 리소스를 구성할 수 있는 Radware DefensePro 애플리케이션이 있는 경우 **Resource Profile(**리소스 프로필**)** 드롭다운 아래에 지원되는 리소스 프로필 목록이 나타납니다. 디바이스에 할당할 리소스 프로필을 선택합니다. 리소스 프로필을 선택하지 않으면 기본 설정이 사용됩니다.

단계 7 **Management Interface(**관리 인터페이스**)** 드롭다운에서 이 절차의 1단계에서 생성한 관리 인터페이스를 선택합니다.

단계 8 **Address Type(**주소 유형**)**을 IPv4 전용, IPv6 전용 또는 IPv4 및 IPv6 중에서 선택합니다.

단계 9 이전 단계의 **Address Type(**주소 유형**)** 선택을 기준으로 다음 필드를 구성합니다.

a) **Management IP(**관리 **IP)** 필드에서 로컬 IP 주소를 구성합니다.

b) IPv4 전용: **Network Mask(**네트워크 마스크**)**를 입력합니다.

IPv6 전용: **Prefix Length(**접두사 길이**)**를 입력합니다.

c) **Network Gateway(**네트워크 게이트웨이**)** 주소를 입력합니다.

단계 10 디바이스에 할당할 각 데이터 포트 옆에 있는 체크 박스를 클릭합니다.

단계 11 **OK(**확인**)**를 클릭합니다.

단계 12 **Save(**저장**)**를 클릭합니다.

FXOS에서 지정된 소프트웨어 버전을 다운로드하고 부트스트랩 구성 및 관리 인터페이스 설정을 지정된 보안 모듈에 입력하여 논리적 디바이스를 구축합니다.

다음에 수행할 작업

DefensePro 애플리케이션의 비밀번호를 설정합니다. 비밀번호를 설정할 때까지 애플리케이션은 온라인 상태가 되지 않습니다. 자세한 내용은 cisco.com에서 Radware DefensePro DDoS 완화 사용 설명서를 참조하십시오.

인트라 섀시(Intra-Chassis) 클러스터에 Radware DefensePro 구성

다음 절차는 Radware DefensePro 이미지를 설치하고 이 이미지를 ASA 또는 FTD 내장 섀시 클러스터 앞에 있는 서비스 체인에 구성하는 방법을 보여줍니다.

참고 서비스 체이닝은 섀시 간 클러스터 구성에서 지원되지 않습니다. 그러나 섀시 간 클러스터 시나리오의 독립형 구성에서는 Radware DefensePro 애플리케이션을 구축할 수 있습니다.

시작하기 전에

- Cisco.com에서 vDP 이미지를 다운로드(Cisco.com에서 이미지 다운로드, 62 페이지 참조)한 다음 해당 이미지를 Firepower 4100/9300 섀시에 업로드합니다(Security Appliance에 이미지 업로드, 62 페이지 참조).

프로시저

단계 1 vDP용으로 별도의 관리 인터페이스를 사용하려는 경우 인터페이스를 활성화한 다음 실제 인터페이스 구성, 184 페이지에 따라 mgmt 유형으로 설정합니다. 그렇지 않은 경우에는 애플리케이션 관리 인터페이스를 공유할 수 있습니다.

단계 2 ASA 또는 FTD 내장 섀시 클러스터를 구성합니다(ASA 클러스터 생성, 244 페이지 또는 FTD 클러스터 생성, 251 페이지 참조).

인트라 섀시 클러스터를 구성하는 마지막 절차에서 **Save**(저장)를 클릭하기 전에 먼저 다음 단계를 수행하여 vDP 데코레이터를 클러스터에 추가해야 합니다.

단계 3 **Decorators**(데코레이터) 영역에서 vDP를 선택합니다. **Radware: Virtual DefensePro - Configuration**(**Radware: Virtual DefensePro** - 구성) 대화 상자가 나타납니다. **General Information**(일반 정보) 탭에서 다음 필드를 구성합니다.

단계 4 둘 이상의 vDP 버전을 Firepower 4100/9300 섀시에 업로드한 경우, 사용할 vDP 버전을 **Version**(버전) 드롭다운에서 선택합니다.

단계 5 리소스를 구성할 수 있는 Radware DefensePro 애플리케이션이 있는 경우 Resource Profile(리소스 프로파일) 드롭다운 아래에 지원되는 리소스 프로파일 목록이 나타납니다. 디바이스에 할당할 리소스 프로필을 선택합니다. 리소스 프로필을 선택하지 않으면 기본 설정이 사용됩니다.

단계 6 **Management Interface**(관리 인터페이스) 드롭다운에서 관리 인터페이스를 선택합니다.

단계 7 vDP 데코레이터에 할당할 각 데이터 포트 옆에 있는 확인란을 클릭합니다.

단계 8 **Interface Information**(인터페이스 정보) 탭을 클릭합니다.

단계 9 사용할 **Address Type**(주소 유형)을 IPv4 전용, IPv6 전용 또는 IPv4 및 IPv6 중에서 선택합니다.

단계 10 각 보안 모듈에 대해 다음 필드를 구성합니다. 표시되는 필드는 이전 단계의 **Address Type**(주소 유형) 선택에 따라 결정됩니다.

a) **Management IP**(관리 **IP**) 필드에서 로컬 IP 주소를 구성합니다.
b) IPv4 전용: **Network Mask**(네트워크 마스크)를 입력합니다.

IPv6 전용: **Prefix Length**(접두사 길이)를 입력합니다.

c) **Network Gateway**(네트워크 게이트웨이) 주소를 입력합니다.

단계 11 **OK**(확인)를 클릭합니다.

단계 12 **Save**(저장)를 클릭합니다.

FXOS에서 지정된 소프트웨어 버전을 다운로드하고 부트스트랩 구성 및 관리 인터페이스 설정을 지정된 보안 모듈에 입력하여 논리적 디바이스를 구축합니다.

단계 13 **Logical Devices**(논리적 디바이스)를 선택하여 Logical Devices(논리적 디바이스) 페이지를 엽니다.

단계 **14** 구성된 논리적 디바이스 목록에서 vDP 항목으로 스크롤합니다. **Management IP(**관리 **IP)** 열에 나열된 해당 속성을 확인합니다.

- **CLUSTER-ROLE** 요소가 DefensePro 인스턴스에 대해 *unknown(*알 수 없음*)*으로 표시되는 경우, DefensePro 애플리케이션을 시작하고 제어 유닛 IP 주소를 구성하여 vDP 클러스터 생성을 완료합니다.
- **CLUSTER-ROLE** 요소가 DefensePro 인스턴스에 대해 *primary* 또는 *secondary* 로 표시되는 경우, 애플리케이션이 온라인 상태로 클러스터에서 형성됩니다.

다음에 수행할 작업

DefensePro 애플리케이션의 비밀번호를 설정합니다. 비밀번호를 설정할 때까지 애플리케이션은 온라인 상태가 되지 않습니다. 자세한 내용은 cisco.com에서 Radware DefensePro DDoS 완화 사용 설명서를 참조하십시오.

UDP/TCP 포트 열기 및 vDP 웹 서비스 활성화

Radware APSolute Vision Manager 인터페이스는 다양한 UDP/TCP 포트를 사용하여 Radware vDP 애플리케이션과 통신합니다. vDP 애플리케이션이 APSolute Vision Manager와 통신하려면 이러한 포트에 액세스 가능하며 방화벽으로 인해 차단되지 않는지 확인해야 합니다. 열려는 특정 포트에 대한 자세한 내용은 APSolute Vision 사용 설명서의 다음 표를 참조하십시오.

- **APSolute Vision Server-WBM** 통신 및 운영 체제에 대한 포트
- **Radware** 디바이스를 사용하는 **APSolute Vision Server**의 통신 포트

Radware APSolute Vision에서 FXOS 섀시에 구축된 가상 DefensePro 애플리케이션을 관리하려면 FXOS CLI를 사용하여 vDP 웹 서비스를 활성화해야 합니다.

프로시저

단계 **1** FXOS CLI에서 vDP 애플리케이션 인스턴스에 연결합니다.

connect module 슬롯 **console**

connect vdp

단계 **2** vDP 웹 서비스를 활성화합니다.

manage secure-web status set enable

단계 **3** vDP 애플리케이션 콘솔을 종료하고 FXOS 모듈 CLI로 돌아갑니다.

Ctrl]

TLS 암호화 가속화 구성

다음 주제에서는 TLS 암호화 가속을 설명하고, 활성화하는 방법 및 FMC를 사용하여 상태를 확인하는 방법에 대해 설명합니다.

다음 표에서는 FTD 및 FXOS 버전을 필수 TSL 암호화와 매핑합니다.

참고 FXOS 2.6.1이 FXOS 2.7.x 이상으로 업그레이드된 경우, FTD 6.4는 TLS 암호화와 호환되지 않으므로 6.4는 암호화를 자동으로 활성화하지 않습니다.

FTD	FXOS	Crypto
6.4	2.6	하나의 컨테이너 인스턴스만 지원(1단계)
6.4	2.7 이상	해당 없음
6.5 이상	2.7 이상	16 컨테이너 인스턴스 지원 (2단계)

정보 TLS 암호화 가속

Firepower 4100/9300은 전송 레이어 보안(TLS) 암호화 가속을 지원합니다. 이는 하드웨어에서 전송 레이어 보안(TLS)/보안 소켓 레이어(SSL)(TLS/SSL) 암호화 및 복호화를 수행하여 다음을 수행하는 속도를 크게 향상시킵니다.

- TLS/SSL 암호화 및 복호화
- TLS/SSL 및 IPsec을 포함한 VPN

네이티브 인스턴스에서는 TLS 암호화 가속화를 비활성화할 수 없습니다. 보안 엔진/모듈당 ~16 FTD 컨테이너 인스턴스개의 TLS 암호화 가속를 활성화할 수도 있습니다.

TLS 암호화 가속화 가이드라인 및 제한사항

FTD이 TLS 암호화 가속를 활성화한 경우 다음 사항에 유의하십시오.

검사 엔진 오류

검사 엔진이 연결을 유지하도록 구성되고 검사 엔진이 예기치 않게 실패하는 경우 엔진이 재시작될 때까지 TLS/SSL트래픽이 중단됩니다.

이 동작은 FTD 명령 **configure snort preserve-connection {enable | disable}** 명령이 제어합니다.

HTTP 전용 성능

트래픽을 암호 해독하지 않는 FTD 컨테이너 인스턴스에서 TLS 암호화 가속를 사용하면 성능에 영향을 줄 수 있습니다. TLS/SSL 트래픽을 암호 해독하는 FTD 컨테이너 인스턴스에 한해 TLS 암호화 가속를 활성화하는 것을 권장합니다.

FIPS(Federal Information Processing Standards)

TLS 암호화 가속 및 FIPS(Federal Information Processing Standard)가 모두 활성화되는 경우, 다음 옵션과의 연결은 실패합니다.

- 크기가 2,048 바이트보다 작은 RSA 키
- RC4(Rivest Cpher 4)
- 단일 데이터 암호화 표준(단일 DES)
- MD5(Merkle–Damgard 5)
- SSL v3

보안 인증 컴플라이언스 모드에서 작동하도록 FMC 및 FTD를 구성하는 경우 FIPS가 활성화됩니다. 해당 모드에서 작동 중 연결을 허용하려면, FTD 컨테이너 인스턴스에서 TLS 암호화 가속를 비활성화하거나 웹 브라우저가 더 안전한 옵션을 허용하도록 구성합니다.

자세한 내용:

- 공통 평가 기준

HA(High Availability, 고가용성)및 클러스터링

HA(High Availability, 고가용성) 또는 클러스터링된 FTD가 있을 경우, 각 FTD에서 개별적으로 TLS 암호화 가속를 활성화해야 합니다. HA 쌍 cluster에서 한 디바이스의 TLS 암호화 가속 구성은 다른 디바이스와 공유되지 않습니다.

TLS 하트비트

일부 애플리케이션은 *TLS* 하트비트를 TLS(Transport Layer Security) 및 DTLS(Datagram Transport Layer Security) 프로토콜로 확장합니다. 이 프로토콜은 RFC6520에서 정의합니다. TLS 하트비트는 연결 상태를 확인하는 방법을 제공합니다. 즉 클라이언트 또는 서버가 특정 바이트의 데이터를 전송하고 상대방의 에코 응답을 요청합니다. 성공한 경우, 암호화된 데이터가 전송됩니다.

TLS 암호화 가속된 FMC에 의해 관리되는 FTD가 TLS 하트비트 확장을 사용하는 패킷을 발견하면, FTD는 SSL 정책의 **Undecryptable Actions(**암호 해독 불가 작업**)**의 **Decryption Errors(**암호 해독 오류**)**에 대한 FMC 설정에서 지정된 작업을 수행합니다.

- Block(차단)
- Block with Reset(차단 후 재설정)

애플리케이션에서 TLS 하트비트를 사용하는지 확인하려면, *Firepower Management Center*구성 가이드에서 TLS/SSL 규칙 트러블슈팅에 대한 장을 참조하십시오.

TLS 암호화 가속가 FTD 컨테이너 인스턴스에서 비활성화되어 있을 경우, FMC의 NAP(Network Analysis Policy)에서 **Max Heartbeat Length(**최대 하트비트 길이**)**를 구성하여 TLS 하트비트를 처리하는 방법을 결정할 수 있습니다.

TLS 하트비트에 대한 자세한 내용은 *Firepower Management Center* 구성 가이드의 TLS/SSL 규칙 트러블슈팅 장을 참조하십시오.

TLS/SSL 초과 서브스크립션

*TLS/SSL oversubscription(*오버서브스크립션*)*은 FTD가 TLS/SSL 래픽으로 오버로드된 상태입니다. 모든 FTD에서 TLS/SSLoversubscription이 발생할 수 있지만 TLS 암호화 가속를 지원하는 FTD만 이를 처리하는 구성 방법을 제공합니다.

TLS 암호화 가속이 활성화된 FMC에 의해 관리되는 FTD가 oversubscription되는 경우, FTD이 수신한 모든 패킷은 SSL 정책 **Undecryptable Actions(**암호 해독 불가 작업**)**의 **Handshake Errors(**핸드셰이크 오류**)**설정에 따라 수행됩니다.

- 기본 작업 상속
- Do not decrypt(암호 해독 안 함)
- Block(차단)
- Block with Reset(차단 후 재설정)

SSL 정책 **Undecryptable Actions(**암호 해독 불가 작업**)**의 **Handsake Errors(**핸드셰이크 오류**)**에 대한 설정이 **Do Not decrypt(**암호 해독 안 함**)**이며 관련 액세스 제어 정책이 트래픽을 검사하도록 구성하는 경우, 검사가 이루어지며 암호 해독은 진행되지 않습니다.

초과 서브스크립션이 많이 발생하는 경우, 다음 방법을 사용합니다.

- TLS/SSL 처리 용량이 더 많은 FTD로 업그레이드하십시오.
- SSL 정책을 변경하여 암호 해독 우선 순위가 높지 않은 트래픽의 **Do Not Decrypt(**암호 해독 안 함**)** 규칙을 추가합니다.

TLS 초과 서브스크립션에 대한 자세한 내용은 *Firepower Management Center* 구성 가이드의 TLS/SSL 규칙 트러블슈팅 장을 참조하십시오.

패시브 및 인라인 탭 세트는 지원되지 않음

TLS 암호화 가속가 활성화되어 있으면 패시브 또는 인라인 탭 모드 세트에서 TLS/SSL 트래픽 암호를 해독할 수 없습니다.

컨테이너 인스턴스에 대해 **TLS** 암호화 가속화 활성화

FMC에 대한 독립형 FTD 추가, 228 페이지에 설명된 대로 논리적 인스턴스를 구축할 때 TLS 암호화 가속이 자동으로 활성화됩니다.

TLS 암호화 가속은 모든 네이티브 인스턴스에서 활성화되며 비활성화할 수 없습니다.

TLS 암호화 가속 상태 보기

이 주제에서는 TLS 암호화 가속 활성화 여부를 확인하는 방법을 설명합니다.

FMC에서 다음 작업을 수행하십시오.

프로시저

단계 **1** FMC에 로그인합니다.

단계 **2** **Devices(**디바이스**)** > **Device Management(**디바이스 관리**)**를 클릭합니다.

단계 **3** 수정( )을 클릭하여 매니지드 디바이스를 편집합니다.

단계 **4** **Device(**디바이스**)** 페이지를 클릭합니다. TLS 암호화 가속 상태가 General(일반) 섹션에 표시됩니다.

FTD 링크 상태 동기화를 활성화합니다.

이제 섀시가 FTD 작동 링크 상태를 데이터 인터페이스의 물리적 링크 상태와 동기화할 수 있습니다. 현재로서는, FXOS 관리 상태가 작동 중이고 물리적 링크 상태가 작동 중이면 인터페이스는 작동 상태가 됩니다. FTD 애플리케이션 인터페이스 관리 상태는 고려되지 않습니다. 예를 들어 FTD에서 동기화하지 않으면 FTD 애플리케이션이 완전히 온라인 상태가 되기 전에 데이터 인터페이스가 물리적으로 작동 상태가 되거나 FTD 종료를 시작한 후 일정 기간 동안 작동 상태를 유지할 수 있습니다. 인라인 집합의 경우 FTD에서 트래픽을 처리하기 전에 외부 라우터가 FTD로 트래픽 전송을 시작할 수 있으므로 이러한 상태 불일치로 인해 패킷이 삭제될 수 있습니다.

이 기능은 기본적으로 비활성화되어 있으며 FXOS에서 논리적 디바이스별로 활성화할 수 있습니다. 이 기능은 관리 또는 클러스터와 같은 비 데이터 인터페이스에는 영향을 주지 않습니다.

FTD 링크 상태 동기화를 활성화하면, FXOS에 있는 인터페이스의 **Service State(**서비스 상태**)**가 FTD에 있는 이 인터페이스의 관리 상태와 동기화됩니다. 예를 들어 FTD에서 인터페이스를 종료하는 경우 Service State(서비스 상태)가 Disabled(비활성화됨)로 표시됩니다. FTD 애플리케이션을 종료하면 모든 인터페이스가 Disabled(비활성화됨)로 표시됩니다. 하드웨어 우회 인터페이스의 경우 FTD에서 인터페이스를 관리적으로 종료하면 Service State(서비스 상태)가 Disabled(비활성화됨)로 설정됩니다. 하지만 FTD 애플리케이션을 종료하거나 다른 섀시 레벨 종료(전원 끄기 포함)를 수행하면 인터페이스 쌍이 Disabled(활성화됨) 상태로 유지됩니다.

FTD 링크 상태 동기화를 비활성화하면 Service State(서비스 상태)는 항상 Enabled(활성화됨)로 표시됩니다.

참고 이 기능은 클러스터링, 컨테이너 인스턴스 또는 Radware vDP 데코레이터가 포함된 FTD에는 지원되지 않습니다. ASA에서도 지원되지 않습니다.

프로시저

단계 **1** **Logical Devices(**논리적 디바이스**)**를 선택하고, FTD 논리적 디바이스에 대해 드롭다운 목록에서 **Enable Link State(**링크 상태 활성화**)**를 선택합니다.

이 기능을 비활성화하려면 **Disable Link State(**링크 상태 비활성화**)**를 선택합니다.

단계 **2** 현재 인터페이스 상태와 마지막 중단 이유를 봅니다.

show interface expand detail

예제**:**

```
Firepower # scope eth-uplink
Firepower /eth-uplink # scope fabric a
Firepower /eth-uplink/fabric # show interface expand detail
Interface:
    Port Name: Ethernet1/2
    User Label:
    Port Type: Data
    Admin State: Enabled
    Oper State: Up
    State Reason:
    flow control policy: default
    Auto negotiation: Yes
    Admin Speed: 1 Gbps
    Oper Speed: 1 Gbps
    Admin Duplex: Full Duplex
    Oper Duplex: Full Duplex
    Ethernet Link Profile name: default
    Oper Ethernet Link Profile name: fabric/lan/eth-link-prof-default
    Udld Oper State: Admin Disabled
    Inline Pair Admin State: Enabled
    Inline Pair Peer Port Name:
    Service State: Enabled
    Last Service State Down Reason: None
    Allowed Vlan: All
    Network Control Policy: default
    Current Task:
<...>
```

논리적 디바이스 관리

논리적 디바이스를 삭제하고, ASA를 투명 모드로 변환하고, 인터페이스 구성을 변경하고, 기존 논리적 디바이스에서 기타 작업을 수행할 수 있습니다.

애플리케이션 콘솔에 연결

다음 절차를 수행하여 애플리케이션의 콘솔에 연결합니다.

프로시저

단계 **1** 콘솔 연결 또는 텔넷 연결을 사용하여 모듈 CLI에 연결합니다.

connect module *slot_number* {**console** | **telnet**}

여러 보안 모듈을 지원하지 않는 디바이스의 보안 엔진에 연결하려면 항상 **1**을 *slot_number*로 사용합니다.

텔넷 연결 사용 시에는 동시에 여러 세션을 모듈에 연결할 수 있으며 연결 속도가 더 빠르다는 이점이 있습니다.

예제:

```
Firepower# connect module 1 console
Telnet escape character is '~'.
Trying 127.5.1.1...
Connected to 127.5.1.1.
Escape character is '~'.

CISCO Serial Over LAN:
Close Network Connection to Exit

Firepower-module1>
```

단계 **2** 애플리케이션 콘솔에 연결합니다. 디바이스에 적절한 명령을 입력합니다.

connect asa *name*

connect ftd *name*

connect vdp *name*

인스턴스 이름을 확인하려면 이름 없이 명령을 입력합니다.

예제:

```
Firepower-module1> connect asa asa1
Connecting to asa(asa1) console... hit Ctrl + A + D  to return to bootCLI
[...]
asa>
```

예제:

```
Firepower-module1> connect ftd ftd1
Connecting to ftd(ftd-native) console... enter exit to return to bootCLI
[...]
>
```

단계 **3** 애플리케이션 콘솔을 FXOS 모듈 CLI로 종료합니다.

- ASA - **Ctrl-a, d**를 입력합니다.
- FTD - **exit**를 입력합니다.
- vDP - **Ctrl-], .**를 입력합니다.

단계 **4** FXOS CLI의 Supervisor(관리자) 수준으로 돌아갑니다.

콘솔을 종료합니다**.**

a) **~**를 입력합니다.

텔넷 애플리케이션을 종료합니다.

b) 텔넷 애플리케이션을 종료하려면 다음을 입력합니다.

telnet>**quit**

텔넷 세션을 종료합니다**.**

a) **Ctrl-], .**를 입력합니다.

예시

다음 예시에서는 보안 모듈 1에 있는 ASA에 연결한 다음 FXOS CLI의 수퍼바이저 레벨로 다시 종료합니다.

```
Firepower# connect module 1 console
Telnet escape character is '~'.
Trying 127.5.1.1...
Connected to 127.5.1.1.
Escape character is '~'.

CISCO Serial Over LAN:
Close Network Connection to Exit

Firepower-module1>connect asa asa1
asa> ~
telnet> quit
Connection closed.
Firepower#
```

논리적 디바이스 삭제

프로시저

단계 **1** **Logical Devices(**논리적 디바이스**)**를 선택하여 Logical Devices(논리적 디바이스) 페이지를 엽니다.

Logical Devices(논리적 디바이스) 페이지는 섀시에 구성되어 있는 논리적 디바이스 목록을 보여줍니다. 논리적 디바이스가 구성되어 있지 않은 경우, 대신 이를 알리는 메시지가 표시됩니다.

단계 **2** 삭제할 논리적 디바이스에 대해 **Delete(**삭제**)**를 클릭합니다.

단계 **3** **Yes(**예**)**를 클릭하여 논리적 디바이스를 삭제할 것임을 확인합니다.

단계 **4** **Yes(**예**)**를 클릭하여 애플리케이션 구성을 삭제할 것임을 확인합니다.

클러스터 유닛 제거

다음 섹션에서는 클러스터에서 유닛을 일시적으로 또는 영구적으로 제거하는 방법을 설명합니다.

임시 제거

하드웨어나 네트워크 장애 등의 이유 때문에 클러스터 유닛이 클러스터에서 자동으로 제거됩니다. 이 제거는 조건을 수정할 때까지 임시로 적용되며, 클러스터에 다시 참여할 수 있습니다. 클러스터링을 수동으로 비활성화할 수도 있습니다.

디바이스가 현재 클러스터에 있는지 확인하려면, 애플리케이션에서 **show cluster info** 명령을 사용해 Firepower Chassis Manager **Logical Devices(**논리적 디바이스**)** 페이지:

FMC을(를) 사용하는 FTD의 경우에는 FMC 디바이스 목록에 디바이스를 남겨 두어야 클러스터링 재활성화 후 전체 기능을 다시 사용할 수 있습니다.

- 애플리케이션에서 클러스터링 비활성화 - 애플리케이션 CLI를 사용하여 클러스터링을 비활성화할 수 있습니다. **cluster remove unit** *name* 명령을 입력해 로그인한 유닛 외의 모든 유닛을 제거합니다. 부트스트랩 설정과 제어 유닛에서 동기화한 마지막 설정도 그대로 유지되므로 나중에 설정이 유실되는 일 없이 유닛을 다시 추가할 수 있습니다. 이 명령을 데이터 유닛에 입력해서 제어 유닛을 제거하면 새로운 제어 유닛이 선택됩니다.

  디바이스가 비활성화되면 모든 데이터 인터페이스가 종료되며, 관리 인터페이스에서만 트래픽을 주고받을 수 있습니다. 트래픽 흐름을 재개하려면 클러스터링을 다시 활성화합니다. 관리 인터페이스에서는 부트스트랩 구성에서 유닛으로 전송된 IP 주소를 사용하여 가동 상태를 유지합니다. 그러나 다시 로드해도 유닛이 클러스터에서 여전히 비활성 상태인 경우에는 관리 인터페이스가 비활성화됩니다.

  클러스터링을 다시 활성화하려면 ASA에 **cluster group** *name*을 입력하고 **enable**을(를) 입력합니다.클러스터링을 다시 활성화하려면 FTD에 **cluster enable**을(를) 입력합니다.

- 애플리케이션 인스턴스 비활성화 - **Logical Devices(**논리적 디바이스**)** 페이지의 Firepower Chassis Manager에서 슬라이더 활성화됨()을(를) 클릭합니다. 나중에 슬라이더 비활성화됨 ()을(를) 사용하여 다시 활성화할 수 있습니다.
- 보안 모듈/엔진 종료 - **Security Module/Engine(**보안 모듈/엔진**)** 페이지의 Firepower Chassis Manager에서 전원 끄기 아이콘을 클릭합니다.
- 섀시 종료 - **Overview(**개요**)** 페이지의 Firepower Chassis Manager에서 종료 아이콘을 클릭합니다.

영구 제거

다음 방법을 사용하면 클러스터 멤버를 영구적으로 제거할 수 있습니다.

FMC을(를) 사용하는 FTD의 경우, 섀시에서 클러스터링을 비활성화하면 유닛을 FMC 디바이스 목록에서 제거해야 합니다.

- 논리적 디바이스 삭제 - **Logical Devices(**논리적 디바이스**)** 페이지의 Firepower Chassis Manager에서 삭제()을(를) 클릭합니다. 이제 독립형 논리적 디바이스, 새 클러스터를 구축하거나 동일한 클러스터에 새 논리적 디바이스를 추가할 수 있습니다.
- 서비스에서 섀시 또는 보안 모듈 제거- 서비스에서 디바이스를 제거하면, 교체 하드웨어를 클러스터의 새 멤버로 추가할 수 있습니다.

논리적 디바이스와 연결되지 않은 애플리케이션 인스턴스 삭제

논리적 디바이스를 삭제하면 논리적 디바이스의 애플리케이션 구성도 삭제할 것인지 묻는 프롬프트가 표시됩니다. 애플리케이션 구성을 삭제하지 않는 경우, 해당 애플리케이션 인스턴스를 삭제할 때까지 다른 애플리케이션을 사용하여 논리적 디바이스를 생성할 수 없습니다. 논리적 디바이스와 더 이상 연결되지 않은 애플리케이션 인스턴스를 보안 모듈/엔진에서 삭제하려면 다음 절차를 사용할 수 있습니다.

프로시저

단계 **1** **Logical Devices(**논리적 디바이스**)**를 선택하여 Logical Devices(논리적 디바이스) 페이지를 엽니다.

Logical Devices(논리적 디바이스) 페이지는 섀시에 구성되어 있는 논리적 디바이스 목록을 보여줍니다. 논리적 디바이스가 구성되어 있지 않은 경우, 대신 이를 알리는 메시지가 표시됩니다. 논리적 디바이스 목록 아래에서 논리적 디바이스와 연결되지 않은 애플리케이션 인스턴스 목록을 확인할 수 있습니다.

단계 **2** 삭제할 애플리케이션 인스턴스에 대해 **Delete(**삭제**)**를 클릭합니다.

단계 **3** **Yes(**예**)**를 클릭하여 애플리케이션 인스턴스를 삭제할 것임을 확인합니다.

FTD 논리적 디바이스에서 인터페이스 변경

FTD 논리적 디바이스에서 인터페이스를 할당 또는 할당 해제하거나 관리 인터페이스를 교체할 수 있습니다. 그런 다음 FMC 또는 FDM에서 인터페이스 구성을 동기화할 수 있습니다.

새 인터페이스를 추가하거나 사용하지 않는 인터페이스를 삭제하는 경우 FTD 구성에 미치는 영향은 아주 적습니다. 그러나 보안 정책에 사용되는 인터페이스를 삭제하면 구성에 영향이 미칩니다. 액세스 규칙, NAT, SSL, ID 규칙, VPN, DHCP 서버 등 FTD 구성의 여러 위치에서 인터페이스를 직접 참조할 수 있습니다. 보안 영역을 참조하는 정책은 영향을 받지 않습니다. 논리적 디바이스에 영향을 주거나 FMC 또는 FDM에서 동기화할 필요 없이 할당된 EtherChannel의 멤버십을 수정할 수도 있습니다.

FMC의 경우: 인터페이스를 삭제하면 해당 인터페이스와 연결된 모든 구성이 삭제됩니다.

FDM의 경우: 기존 인터페이스를 삭제하기 전에 한 인터페이스에서 다른 인터페이스로 구성을 마이그레이션할 수 있습니다.

시작하기 전에

- 인터페이스를 구성하고 실제 인터페이스 구성, 184 페이지 및 EtherChannel(포트 채널) 추가, 185 페이지에 따라 EtherChannel을 추가합니다.
- 모든 인터페이스가 기본적으로 클러스터에 할당된 경우와 같이 이미 할당된 인터페이스를 EtherChannel에 추가하려는 경우에는 먼저 논리적 디바이스에서 인터페이스 할당을 해제한 다음 EtherChannel에 인터페이스를 추가해야 합니다. 새 EtherChannel의 경우 이렇게 한 후에 디바이스에 EtherChannel을 할당할 수 있습니다.
- 관리 또는 이벤트 인터페이스를 관리 EtherChannel로 교체하려는 경우에는 미할당 데이터 멤버 인터페이스가 하나 이상 포함된 EtherChannel을 생성한 다음 현재 관리 인터페이스를 EtherChannel로 교체해야 합니다. FTD 디바이스가 리부팅되고(관리 인터페이스를 변경하면 리부팅됨) FMC 또는 FDM에서 구성을 동기화한 후에는 이제 할당 해제된 관리 인터페이스를 EtherChannel에 추가할 수도 있습니다.
- 클러스터링 또는 고가용성의 경우에는 FMC 또는 FDM에서 구성을 동기화하기 전에 모든 유닛에서 인터페이스를 추가하거나 제거해야 합니다. 인터페이스는 먼저 데이터/스탠바이 유닛에서 변경한 후에 제어/액티브 유닛에서 변경하는 것이 좋습니다. 새 인터페이스는 관리를 위해 다운된 상태로 추가되므로 인터페이스 모니터링에는 영향을 주지 않습니다.

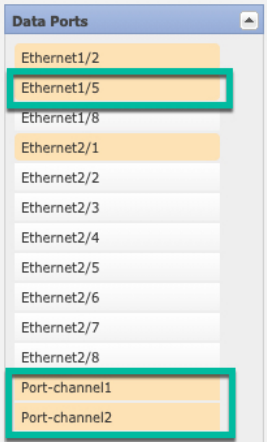
프로시저

단계 **1** Firepower Chassis Manager에서 **Logical Devices(**논리적 디바이스**)**를 선택합니다.

단계 **2** 오른쪽 상단의 **Edit(**수정**)** 아이콘을 클릭하여 논리적 디바이스를 수정합니다.

단계 **3** **Data Ports(**데이터 포트**)** 영역에서 인터페이스를 선택하여 새 데이터 인터페이스를 할당합니다.

아직 인터페이스를 삭제하지 마십시오.



단계 **4** 관리 또는 이벤트 처리 인터페이스를 교체합니다.

이러한 인터페이스 유형의 경우 변경 사항을 저장하고 나면 디바이스가 리부팅됩니다.

a) 페이지 중앙의 디바이스 아이콘을 클릭합니다.
b) **General(**일반**)** 또는 **Cluster Information(**클러스터 정보**)** 탭의 드롭다운 목록에서 새 **Management Interface(**관리 인터페이스**)**를 선택합니다.
c) **Settings(**설정**)** 탭의 드롭다운 목록에서 새 **Eventing Interface(**이벤트 인터페이스**)**를 선택합니다.
d) **OK(**확인**)**를 클릭합니다.

관리 인터페이스의 IP 주소를 변경하는 경우에는 FMC에서 디바이스의 IP 주소도 변경해야 합니다. 이렇게 하려면 **Device(**디바이스**)** > **Device Management(**디바이스 관리**)** > **Device/Cluster(**디바이스/클러스터**)**로 이동합니다. **Management(**관리**)** 영역에서 부트스트랩 구성 주소와 일치하도록 IP 주소를 설정합니다.

단계 **5** **Save(**저장**)**를 클릭합니다.

단계 **6** FMC에서 인터페이스를 동기화합니다.

a) FMC에 로그인합니다.
b) **Devices(**디바이스**)** > **Device Management(**디바이스 관리**)**를 선택하고 FTD 디바이스에 대한 수정 ( )를 클릭합니다. 기본적으로는 **Interfaces(**인터페이스**)** 페이지가 선택됩니다.
c) **Interfaces(**인터페이스**)** 페이지 왼쪽 상단의 **Sync Device(**디바이스 동기화**)** 버튼을 클릭합니다.
d) 변경 사항이 탐지되면 **Interfaces(**인터페이스**)** 페이지에 인터페이스 구성이 변경되었음을 나타내는 빨간색 배너가 표시됩니다. 인터페이스 변경 사항을 보려면 클릭하여 더 보기 링크를 클릭합니다.
e) 인터페이스를 삭제하려는 경우, 기존 인터페이스에서 새 인터페이스로 모든 인터페이스 구성을 수동으로 전송합니다.

   아직 인터페이스를 삭제하지 않았으므로 기존 구성을 참조할 수 있습니다. 이전 인터페이스를 삭제하고 검증을 다시 실행한 후에 구성을 추가로 수정할 수 있습니다. 검증을 수행하면 이전 인터페이스가 아직 사용되고 있는 모든 위치가 표시됩니다.

f) 인터페이스 변경 이후에도 정책이 계속 작동할 수 있도록 변경 사항 유효성 확인을 클릭합니다.

   오류가 발생하는 경우 정책을 변경하고 유효성 검사를 다시 실행해야 합니다.

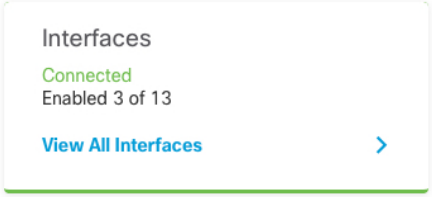
g) **Save(**저장**)**를 클릭합니다.

h) 디바이스를 선택하고 **Deploy(**구축**)**를 클릭하여 할당된 디바이스에 정책을 구축합니다. 변경사항은 구축할 때까지 활성화되지 않습니다.

단계 **7** FDM에서 인터페이스를 동기화하고 마이그레이션합니다.

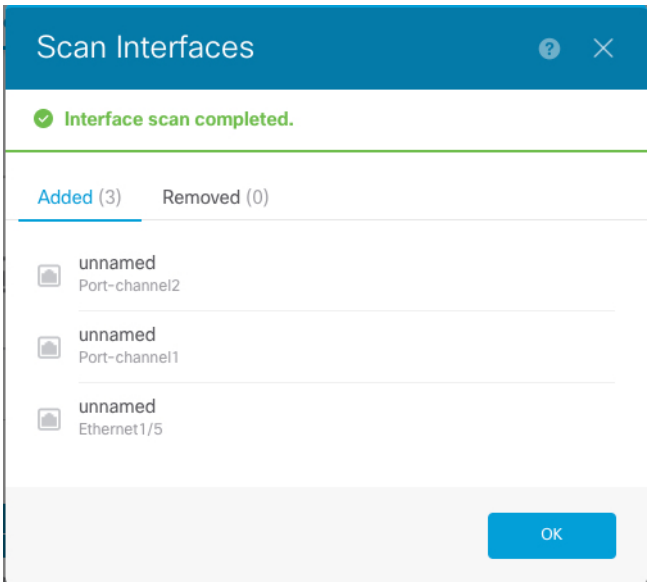
a) FDM에 로그인합니다.

b) **Device(**디바이스**)**를 클릭한 다음, **Interfaces(**인터페이스**)** 요약에서 **View All Interfaces(**모든 인터페이스 보기**)** 링크를 클릭합니다.



c) **Scan Interfaces icon(**인터페이스 스캔 아이콘**)**을 클릭합니다.

d) 인터페이스가 스캔될 때까지 기다린 다음, **OK(**확인**)**를 클릭합니다.



e) 이름, IP 주소 등을 사용하여 새 인터페이스를 구성합니다.

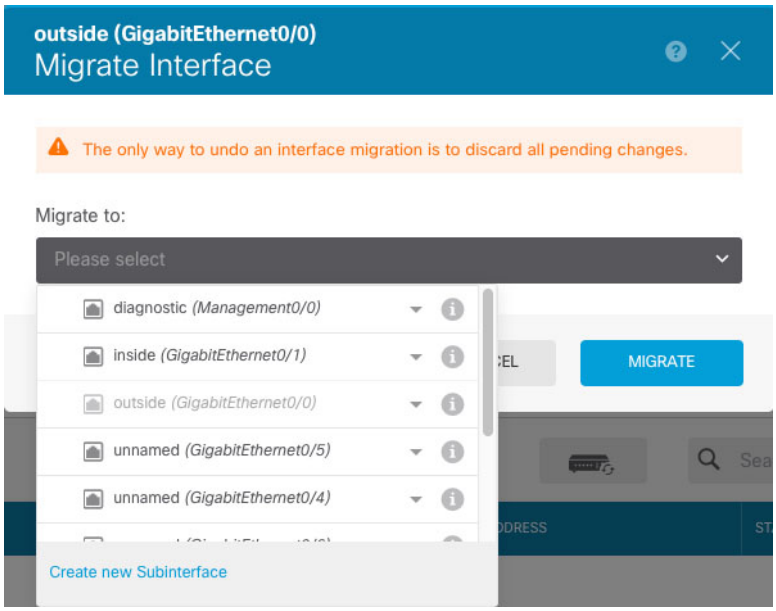
제거할 인터페이스의 기존 IP 주소 및 이름을 사용하려는 경우에는 새 인터페이스에서 해당 설정을 사용할 수 있도록 기존 인터페이스를 더미 이름 및 IP 주소로 다시 구성해야 합니다.

f) 기존 인터페이스를 새 인터페이스로 교체하려면 기존 인터페이스의 Replace(교체) 아이콘을 클릭합니다.

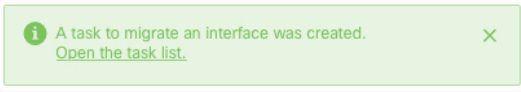
바꾸기 아이콘

이 프로세스에서는 인터페이스를 참조하는 모든 구성 설정에서 기존 인터페이스가 새 인터페이스로 교체됩니다.

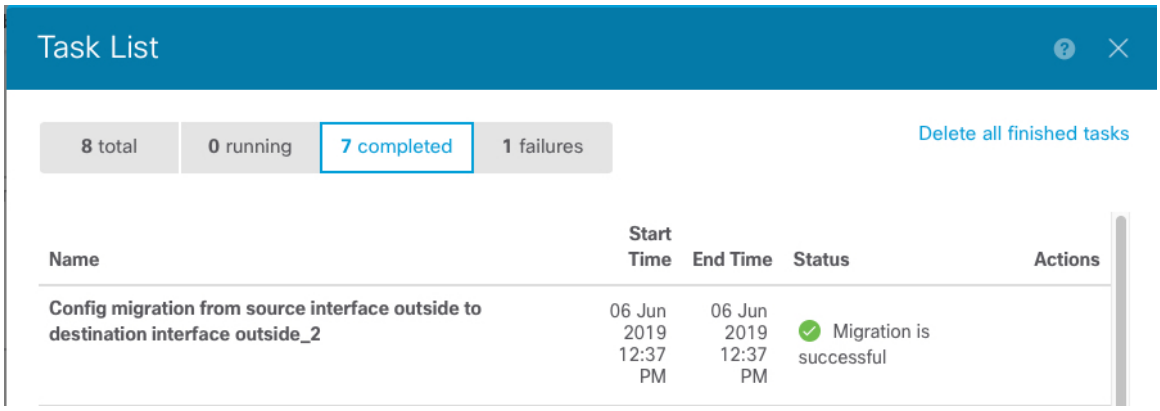
g) **Replacement Interface(**교체 인터페이스**)** 드롭다운 목록에서 새 인터페이스를 선택합니다.



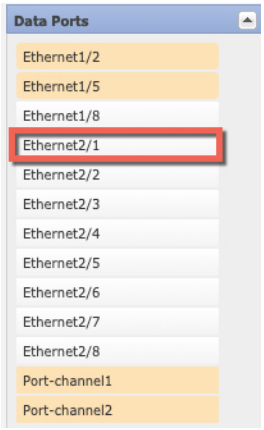
h) **Interfaces(**인터페이스**)** 페이지에 메시지가 나타납니다. 메시지에서 링크를 클릭합니다.



i) **Task List(**작업 목록**)**를 확인하여 마이그레이션에 성공했는지 확인합니다.



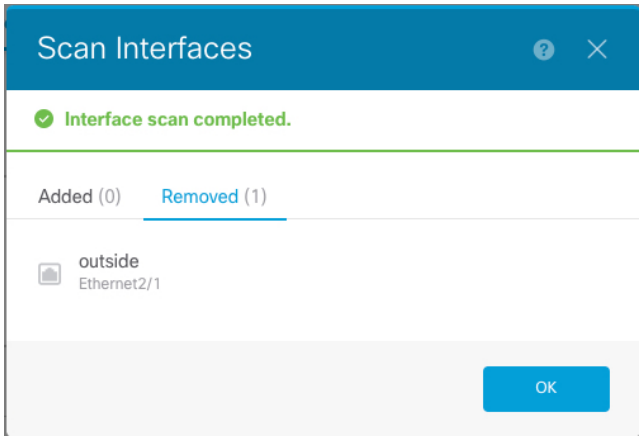
단계 **8** Firepower Chassis Manager에서 **Data Ports(**데이터 포트**)** 영역에서 인터페이스를 선택 취소하여 데이터 인터페이스를 할당 해제합니다.



단계 **9** **Save**(저장)를 클릭합니다.

단계 **10** FMC 또는 FDM에서 인터페이스를 다시 동기화합니다.

그림 ***16: FDM*** 스캔 인터페이스



ASA 논리적 디바이스에서 인터페이스 변경

ASA 논리적 디바이스에서 관리 인터페이스를 할당, 할당 해제 또는 교체할 수 있습니다. ASDM은 새 인터페이스를 자동으로 검색합니다.

새 인터페이스를 추가하거나 사용하지 않는 인터페이스를 삭제하는 경우 ASA 구성에 미치는 영향은 아주 적습니다. 그러나 FXOS에서 할당된 인터페이스를 제거하고(예: 네트워크 모듈을 제거하거나, EtherChannel을 제거하거나, 할당된 인터페이스를 EtherChannel에 재할당하는 경우), 해당 인터페이스가 보안 정책에서 사용되는 경우, 제거하면 ASA 구성에 영향을 미칩니다. 이 경우 ASA 구성은 원래 명령을 유지하므로 필요한 조정을 수행할 수 있습니다. ASA OS에서 이전 인터페이스 구성을 수동으로 제거할 수 있습니다.

참고 논리적 디바이스에 영향을 주지 않고 할당된 EtherChannel의 멤버십을 수정할 수 있습니다.

시작하기 전에

- 실제 인터페이스 구성, 184 페이지 및 EtherChannel(포트 채널) 추가, 185 페이지에 따라 인터페이스를 구성하고 EtherChannel을 추가합니다.
- 모든 인터페이스가 기본적으로 클러스터에 할당된 경우와 같이 이미 할당된 인터페이스를 EtherChannel에 추가하려는 경우에는 먼저 논리적 디바이스에서 인터페이스 할당을 해제한 다음 EtherChannel에 인터페이스를 추가해야 합니다. 새 EtherChannel의 경우 이렇게 한 후에 디바이스에 EtherChannel을 할당할 수 있습니다.
- 관리 인터페이스를 관리 EtherChannel로 교체하려는 경우에는 미할당 데이터 멤버 인터페이스가 하나 이상 포함된 EtherChannel을 생성한 다음 현재 관리 인터페이스를 EtherChannel로 교체해야 합니다. ASA가 다시 로드되고 나면(관리 인터페이스를 변경하면 ASA가 다시 로드됨) 이제 미할당 상태가 된 관리 인터페이스를 EtherChannel에 추가할 수도 있습니다.
- 클러스터링 또는 페일오버의 경우 모든 유닛에서 인터페이스를 추가하거나 제거해야 합니다. 인터페이스는 먼저 데이터/스탠바이 유닛에서 변경한 후에 제어/액티브 유닛에서 변경하는 것이 좋습니다. 새 인터페이스는 관리를 위해 다운된 상태로 추가되므로 인터페이스 모니터링에는 영향을 주지 않습니다.

프로시저

단계 **1** Firepower Chassis Manager에서 **Logical Devices(**논리적 디바이스**)**를 선택합니다.

단계 **2** 오른쪽 상단의 **Edit(**수정**)** 아이콘을 클릭하여 논리적 디바이스를 수정합니다.

단계 **3** **Data Ports(**데이터 포트**)** 영역에서 인터페이스를 선택 취소하여 데이터 인터페이스를 할당 해제합니다.

단계 **4** **Data Ports(**데이터 포트**)** 영역에서 인터페이스를 선택하여 새 데이터 인터페이스를 할당합니다.

단계 **5** 관리 인터페이스를 교체합니다.

이 인터페이스 유형의 경우 변경 사항을 저장하고 나면 디바이스가 다시 로드됩니다.

a) 페이지 중앙의 디바이스 아이콘을 클릭합니다.

b) **General/Cluster Information(**일반/클러스터 정보**)** 탭의 드롭다운 목록에서 새 **Management Interface(**관리 인터페이스**)**를 선택합니다.

c) **OK(**확인**)**를 클릭합니다.

단계 **6** **Save(**저장**)**를 클릭합니다.

논리적 디바이스의 부트스트랩 설정 수정 또는 복구

논리적 디바이스의 부트스트랩 설정을 수정할 수 있습니다. 그런 다음 이러한 새 설정을 사용하여 애플리케이션 인스턴스를 즉시 재시작하거나 변경 사항을 저장하고 나중에 새 설정을 사용하여 애플리케이션 인스턴스를 재시작할 수 있습니다.

프로시저

단계 **1** Firepower Chassis Manager에서 **Logical Devices(**논리적 디바이스**)**를 선택합니다.

단계 **2** 오른쪽 상단의 **Edit(**수정**)** 아이콘을 클릭하여 논리적 디바이스를 수정합니다.

단계 **3** 페이지 중앙의 디바이스 아이콘을 클릭합니다.

단계 **4** 필요에 따라 논리적 디바이스 설정을 수정합니다.

단계 **5** **OK(**확인**)**를 클릭합니다.

단계 **6** 변경 사항을 저장하고 애플리케이션 인스턴스를 즉시 재시작하려면 **Restart Now(**지금 재시작**)**를 클릭합니다. 애플리케이션 인스턴스를 재시작하지 않고 변경 사항을 저장하려면 **Restart Later(**나중에 재시작**)**를 클릭합니다.

참고 **Restart Later(**나중에 재시작**)**를 선택한 경우 준비가 되면 Logical Devices(논리적 디바이스) 페이지에서 **Restart Instance(**인스턴스 재시작**)**를 클릭하여 애플리케이션 인스턴스를 재시작할 수 있습니다.

논리적 디바이스 페이지

Firepower Chassis Manager의 **Logical Devices(**논리적 디바이스**)** 페이지를 사용하여 논리적 디바이스를 생성, 수정 및 삭제합니다. **Logical Devices(**논리적 디바이스**)** 페이지에는 각 Firepower 4100/9300 섀시 보안 모듈/엔진에 설치된 논리적 디바이스에 대한 정보 영역이 포함되어 있습니다.

각 논리적 디바이스 영역의 헤더에서는 다음 정보를 제공합니다.

- 논리적 디바이스의 고유한 이름.
- 논리적 디바이스 모드(독립형 또는 클러스터형).
- **Status(**상태**)** - 논리적 디바이스의 상태를 표시합니다.
  - ok - 논리적 디바이스 구성이 완료되었습니다.
  - incomplete-configuration - 논리적 디바이스 구성이 완료되지 않았습니다.

각 논리적 디바이스 영역에서는 다음 정보를 제공합니다.

- **Application(**애플리케이션**)** - 보안 모듈에서 실행 중인 애플리케이션을 표시합니다.
- **Version(**버전**)** - 보안 모듈에서 실행 중인 애플리케이션의 소프트웨어 버전 번호를 표시합니다.

참고 FTD 논리적 디바이스에 대한 업데이트는 FMC를 사용하여 완료되며, Firepower Chassis Manager의 **Logical Devices(**논리적 디바이스**)** > **Edit(**수정**)** 및 **System(**시스템**)** > **Updates(**업데이트**)** 페이지에 반영되지 않습니다. 이러한 페이지에 표시되는 버전은 FTD 논리적 디바이스를 만드는 데 사용된 소프트웨어 버전(CSP 이미지)을 나타냅니다.

- **Resource Profile(**리소스 프로파일**)** - 논리적 디바이스/애플리케이션 인스턴스에 할당된 리소스 프로파일을 표시합니다.
- **Management IP(**관리 **IP)** - 논리적 디바이스 관리 IP로 할당된 로컬 IP 주소를 표시합니다.
- **Gateway(**게이트웨이**)** - 애플리케이션 인스턴스에 할당된 네트워크 게이트웨이 주소를 표시합니다.
- **Management Port(**관리 포트**)** - 애플리케이션 인스턴스에 할당된 관리 포트를 표시합니다.
- **Status(**상태**)** - 애플리케이션 인스턴스의 상태를 표시합니다.
  - Online(온라인) - 애플리케이션이 실행되어 작동 중입니다.
  - Offline(오프라인) - 애플리케이션이 중지되어 작동하지 않습니다.
  - Installing(설치 중) - 애플리케이션 설치가 진행 중입니다.
  - Not Installed(설치되지 않음) - 애플리케이션이 설치되지 않았습니다.
  - Install Failed(설치 실패) - 애플리케이션 설치가 실패했습니다.
  - Starting(시작 중) - 애플리케이션이 시작 중입니다.
  - Start Failed(시작 실패) - 애플리케이션 시작에 실패했습니다.
  - Started(시작됨) - 애플리케이션이 성공적으로 시작되었고, 앱 에이전트 하트비트를 대기 중입니다.
  - Stopping(중지 중) - 애플리케이션이 중지 중입니다.
  - Stop Failed(중지 실패) - 애플리케이션을 오프라인으로 전환하지 못했습니다.
  - Not Responding(응답 없음) - 애플리케이션이 응답하지 않습니다.
  - Updating(업데이트 중) - 애플리케이션 소프트웨어 업데이트가 진행 중입니다.
  - Update Failed(업데이트 실패) - 애플리케이션 소프트웨어 업데이트에서 장애가 발생했습니다.
  - Update Succeeded(업데이트 성공) - 애플리케이션 소프트웨어 업데이트가 성공했습니다.
  - Unsupported(지원되지 않음) - 설치된 애플리케이션이 지원되지 않습니다.

보안 모듈이 없거나 결함이 있는 상태이면 Status(상태) 필드에 해당 정보가 표시됩니다. 정보 아이콘에 마우스를 올려 결함에 대한 추가 정보를 확인할 수 있습니다. 보안 모듈 결함에 대한 자세한 내용은 FXOS 보안 모듈/보안 엔진 정보, 297 페이지 섹션을 참조하십시오.

- **Expanded Information Area(**확장된 정보 영역**)** - 현재 실행 중인 애플리케이션 인스턴스에 대한 추가 속성을 표시합니다.

참고 애플리케이션 인스턴스를 즉시 재시작하지 않고 애플리케이션의 부트스트랩 설정을 수정하는 경우, Attributes(속성) 필드는 현재 실행 중인 애플리케이션에 대한 정보를 표시하며 애플리케이션이 재시작될 때까지 수행된 변경 사항을 반영하지 않습니다.

  - **Ports(**포트**)** - 애플리케이션 인스턴스에 할당된 포트를 표시합니다.
  - **Cluster Operation Status(**클러스터 작동 상태**)** - 애플리케이션 인스턴스에 할당된 관리 URL을 표시합니다.
  - **Management IP/Firepower Management IP(**관리 **IP/Firepower** 관리 **IP)** - 애플리케이션 인스턴스에 할당된 관리 IP 주소를 표시합니다.
  - **Cluster Role(**클러스터 역할**)** - 애플리케이션 인스턴스, 제어 또는 데이터에 대한 클러스터 역할을 표시합니다.
  - **Cluster IP(**클러스터 **IP)** - 애플리케이션 인스턴스에 할당된 네트워크 IP 주소를 표시합니다.
  - **HA Role(HA** 역할**)** - 애플리케이션 인스턴스의 고가용성 역할(액티브 또는 스탠바이)을 표시합니다.
  - **Management URL(**관리 **URL)** - 애플리케이션 인스턴스에 할당된 관리 애플리케이션의 URL을 표시합니다.
  - **UUID** - 애플리케이션 인스턴스의 UUID(Universally Unique Identifier)를 표시합니다.

Firepower Chassis Manager의 **Logical Devices(**논리적 디바이스**)** 페이지에서 논리적 디바이스에 대해 다음 기능을 수행할 수 있습니다.

- **Refresh(**새로 고침**)** - Logical Devices(논리적 디바이스) 페이지의 정보를 새로 고칩니다.
- **Add Device(**디바이스 추가**)** - 논리적 디바이스를 생성할 수 있습니다.
- **Edit(**수정**)** - 기존 논리적 디바이스를 수정할 수 있습니다.
- **Set Version(**버전 설정**)**- 논리적 디바이스의 소프트웨어를 업그레이드하거나 다운그레이드할 수 있습니다.
- **Delete(**삭제**)** - 논리적 디바이스를 삭제합니다.
- **Show Configuration(**구성 표시**)** - 논리적 디바이스나 클러스터에 대한 구성 정보가 JSON 형식으로 표시되는 대화 상자를 엽니다. 구성 정보를 복사하여 클러스터의 일부분인 추가 디바이스를 생성할 때 사용할 수 있습니다.

- **Enable/Disable(**활성화/비활성화**)** - 애플리케이션 인스턴스를 활성화하거나 비활성화합니다.
- **Upgrade/Downgrade(**업그레이드/다운그레이드**)** - 애플리케이션 인스턴스를 업그레이드하거나 다운그레이드할 수 있습니다.
- **Restart Instance(**인스턴스 재시작**)** - 애플리케이션 인스턴스를 재시작할 수 있습니다. 디바이스 부트스트랩 정보를 수정했는데 애플리케이션 인스턴스는 아직 재시작하지 않은 경우 Restart Instance(인스턴스 재시작)를 클릭하여 기존 관리 부트스트랩 정보를 지우고 새 부트스트랩 정보를 사용하여 애플리케이션 인스턴스를 재시작할 수 있습니다.
- **Reinstall Instance(**인스턴스 재설치**)** - 애플리케이션 인스턴스를 재설치할 수 있습니다.
- **Go To Device Manager(**디바이스 매니저로 이동**)** - 애플리케이션 인스턴스에 대해 정의된 FMC 또는 ASDM으로 이동하는 링크를 제공합니다.
- **Enable/Disable Link State(**링크 상태 활성화/비활성화**)** - FTD 링크 상태 동기화를 활성화하거나 비활성화합니다. 자세한 내용은 FTD 링크 상태 동기화를 활성화합니다., 272 페이지를 참고하십시오.

사이트 간 클러스터링 예시

다음 예에는 지원되는 클러스터 구축에 대한 내용이 나와 있습니다.

사이트별 **MAC** 주소가 있는 **Spanned EtherChannel** 라우팅 모드의 예

다음 예에서는 게이트웨이 라우터와 각 사이트의 내부 네트워크 사이에 위치한(이스트-웨스트 삽입) 2개 데이터 센터 각각에 2개의 클러스터 멤버를 보여줍니다. 클러스터 멤버는 DCI를 통해 클러스터 제어 링크로 연결됩니다. 각 사이트의 클러스터 멤버는 내부 및 외부 네트워크용 Spanned EtherChannel을 사용하여 로컬 스위치에 연결됩니다. 각 EtherChannel은 클러스터의 모든 섀시를 포괄합니다.

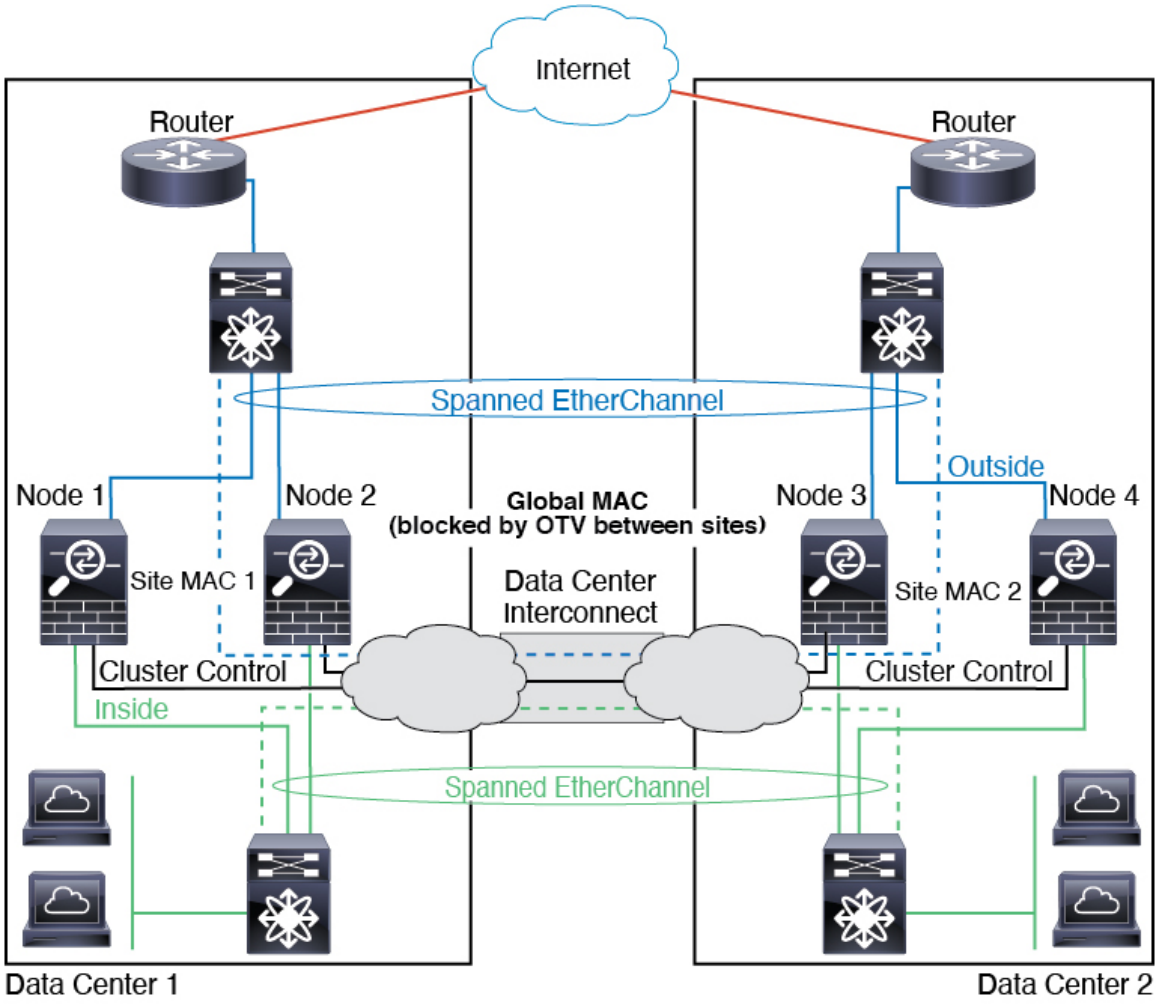
OTV(Overlay Transport Virtualization) 또는 이와 유사한 방법으로 데이터 VLAN이 사이트 전반에 확장됩니다. 트래픽이 클러스터로 향할 때 DCI를 통과하여 반대쪽 사이트에 가지 않도록 전역 MAC 주소를 차단하는 필터를 추가해야 합니다. 어떤 사이트의 클러스터 노드가 연결할 수 없게 되면 트래픽이 다른 사이트의 클러스터 노드에 성공적으로 도달할 수 있도록 모든 필터를 제거해야 합니다. VACL을 사용하여 전역 MAC 주소를 필터링해야 합니다. ARP 검사를 비활성화해야 합니다.

클러스터는 내부 네트워크의 게이트웨이 역할을 합니다. 모든 클러스터 노드에서 공유되는 전역 가상 MAC은 패킷 수신에만 사용됩니다. 발신 패킷은 각 DC 클러스터의 사이트별 MAC 주소를 사용합니다. 이 기능은 스위치가 서로 다른 두 포트의 두 사이트로부터 동일한 전역 MAC 주소를 학습하지 못하게 하는 한편, MAC 플래핑(flapping)을 일으킵니다. 대신 스위치는 사이트 MAC 주소만 학습합니다.

이 시나리오에서:

- 클러스터에서 전송한 모든 이그레스(egress) 패킷은 사이트 MAC 주소를 사용하며 데이터 센터에서 지역화됩니다.

- 클러스터에 대한 모든 인그레스 패킷은 전역 MAC 주소를 사용하여 전송되므로, 양 사이트의 어느 노드에서나 수신할 수 있습니다. OTV의 필터는 데이터 센터 내에서 트래픽을 지역화합니다.



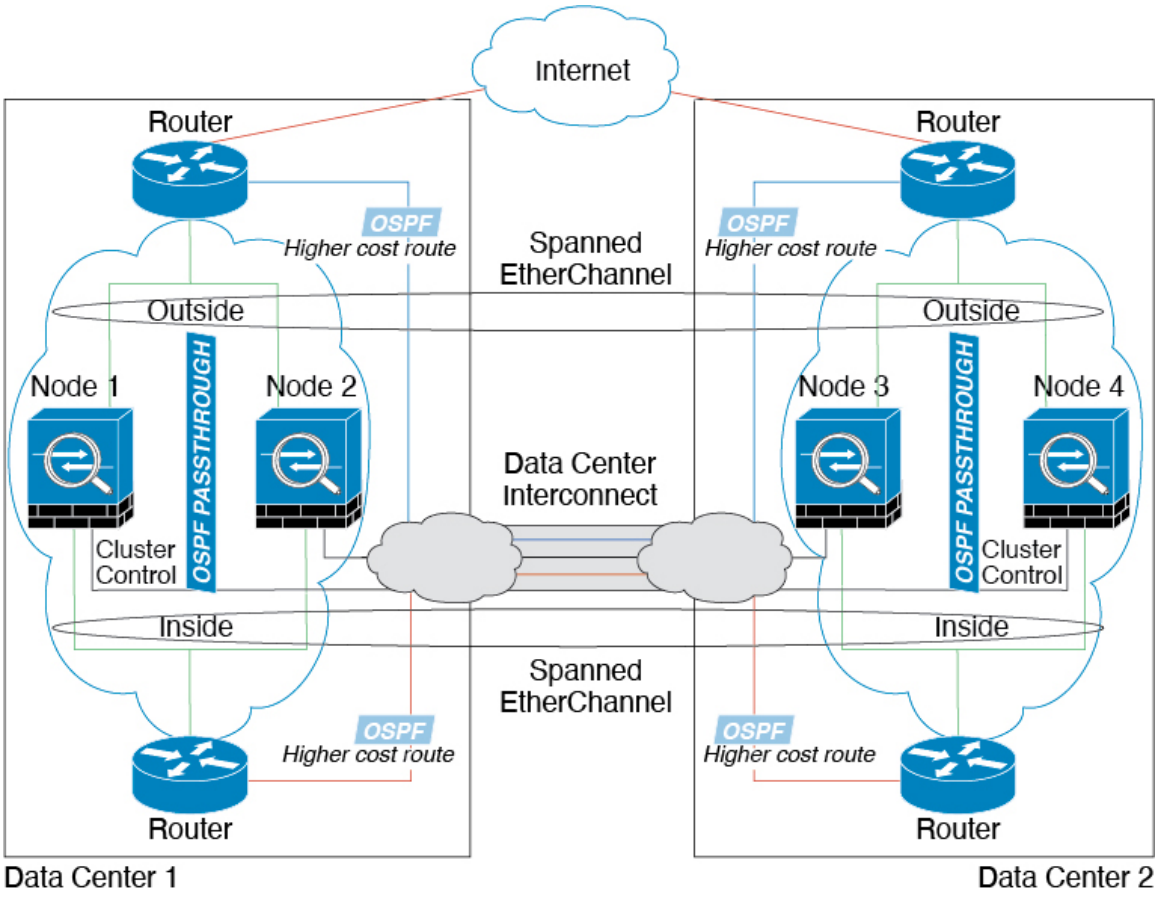
Spanned EtherChannel 투명 모드 노스-사우스 사이트 간의 예

다음 예에서는 내부 라우터와 외부 라우터의 사이에 위치한(노스-사우스 삽입) 2개 데이터 센터 각각에 2개의 클러스터 멤버가 있습니다. 클러스터 멤버는 DCI를 통해 클러스터 제어 링크로 연결됩니다. 각 사이트의 클러스터 멤버는 내부 및 외부용 스팬 EtherChannel을 사용하여 로컬 스위치에 연결됩니다. 각 EtherChannel은 클러스터의 모든 섀시를 포괄합니다.

각 데이터 센터의 내부 및 외부 라우터에서는 투명 ASA를 통과하는 OSPF를 사용합니다. MAC과 달리 라우터 IP는 모든 라우터마다 고유합니다. DCI를 통해 비용이 높은 경로를 할당하면 특정 사이트의 모든 클러스터 멤버가 가동 중지되지 않는 한 각 데이터 센터 내에서 트래픽이 유지됩니다. ASA를 통과하는 비용이 낮은 경로의 경우, 클러스터의 각 사이트에 있는 같은 브리지 그룹을 거쳐 비대칭 연결을 유지해야 합니다. 어느 한 사이트의 모든 클러스터 멤버에 오류가 발생할 경우, 각 라우터의 트래픽은 DCI를 통해 다른 사이트의 클러스터 멤버로 이동합니다.

각 사이트의 스위치 구현 과정에는 다음 사항이 포함될 수 있습니다.

- 사이트 간 VSS, vPC, StackWise 또는 StackWise Virtual - 이 시나리오에서는 데이터 센터 1에 스위치 하나를 설치하고 데이터 센터 2에 다른 스위치를 설치합니다. 한 가지 옵션은 각 데이터 센터의 클러스터 노드가 로컬 스위치에만 연결하는 반면 중복 스위치 트래픽은 DCI를 통과하는 것입니다. 이 경우 연결의 대부분은 각 데이터 센터에 로컬로 저장됩니다. DCI에서 추가 트래픽을 처리할 수 있는 경우, 선택에 따라 각 노드를 DCI 전반의 스위치에 연결할 수 있습니다. 이 경우 트래픽이 데이터 센터 전반에 분산되므로 DCI의 성능이 매우 뛰어나야 합니다.
- 각 사이트의 로컬 VSS, vPC, StackWise 또는 StackWise Virtual - 더 나은 스위치 이중화를 위해 각 사이트에 2개의 개별 이중화 스위치 쌍을 설치할 수 있습니다. 이 경우 여전히 클러스터 노드의 Spanned EtherChannel은 두 로컬 스위치에만 연결된 데이터 센터 1 섀시 및 이러한 로컬 스위치에 연결된 데이터 센터 2 섀시로 이루어져 있지만, 사실상 Spanned EtherChannel은 "분리"되어 있습니다. 각 로컬 중복 스위치 시스템은 스팬 EtherChannel을 사이트 로컬 EtherChannel로 간주합니다.

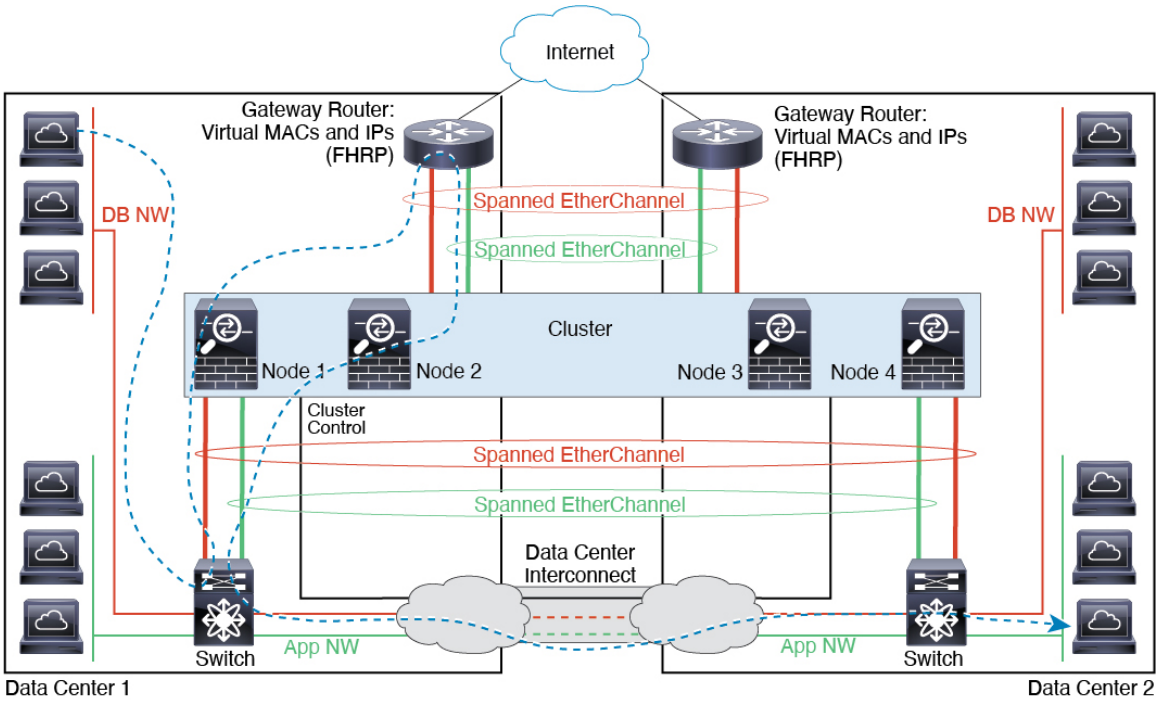


Spanned EtherChannel 투명 모드 이스트-웨스트 사이트 간의 예

다음 예에서는 게이트웨이 라우터와 각 사이트의 두 내부 네트워크, 즉 애플리케이션 네트워크 및 DB 네트워크의 사이에 위치한(이스트-웨스트 삽입) 2개 데이터 센터 각각에 2개의 클러스터 멤버가 있

습니다. 클러스터 멤버는 DCI를 통해 클러스터 제어 링크로 연결됩니다. 각 사이트의 클러스터 멤버는 내부 및 외부에 있는 애플리케이션 및 DB 네트워크에 대한 스팬 EtherChannel을 사용하여 로컬 스위치에 연결됩니다. 각 EtherChannel은 클러스터의 모든 섀시를 포괄합니다.

각 사이트의 게이트웨이 라우터는 HSRP와 같은 FHRP를 사용하여 각 사이트에 동일한 목적지 가상 MAC 및 IP 주소를 제공합니다. 의도치 않은 MAC 주소 플래핑(flapping)을 피하는 좋은 방법은. 이러한 항목이 없으면, 사이트 1의 게이트웨이가 사이트 2의 게이트웨이와 통신할 경우 해당 트래픽이 ASA를 통과해 내부 인터페이스에서 사이트 2에 도달하려고 시도하여 문제를 일으킬 수 있습니다. OTV(Overlay Transport Virtualization) 또는 이와 유사한 방법으로 데이터 VLAN이 사이트 전반에 확장됩니다. 트래픽이 게이트웨이 라우터로 예정된 경우 트래픽에서 다른 사이트에 DCI를 전달하는 것을 방지하려면 필터를 추가해야 합니다. 한 개의 사이트에서 게이트웨이 라우터에 연결할 수 없는 경우, 필터를 제거해야 트래픽이 다른 사이트의 게이트웨이 라우터에 전송될 수 있습니다.



논리적 디바이스의 기록

기능 이름	플랫폼 릴리스	기능 정보
FTD 작동 링크 상태와 물리적 링크 상태 간 동기화	2.9.1	이제 섀시가 FTD 작동 링크 상태를 데이터 인터페이스의 물리적 링크 상태와 동기화할 수 있습니다. 현재로서는, FXOS 관리 상태가 작동 중이고 물리적 링크 상태가 작동 중이면 인터페이스는 작동 상태가 됩니다. FTD 애플리케이션 인터페이스 관리 상태는 고려되지 않습니다. 예를 들어 FTD에서 동기화하지 않으면 FTD 애플리케이션이 완전히 온라인 상태가 되기 전에 데이터 인터페이스가 물리적으로 작동 상태가 되거나 FTD 종료를 시작한 후 일정 기간 동안 작동 상태를 유지할 수 있습니다. 인라인 집합의 경우 FTD에서 트래픽을 처리하기 전에 외부 라우터가 FTD로 트래픽 전송을 시작할 수 있으므로 이러한 상태 불일치로 인해 패킷이 삭제될 수 있습니다. 이 기능은 기본적으로 비활성화되어 있으며 FXOS에서 논리적 디바이스별로 활성화할 수 있습니다. 참고 이 기능은 클러스터링, 컨테이너 인스턴스 또는 Radware vDP 데코레이터가 포함된 FTD에는 지원되지 않습니다. ASA에서도 지원되지 않습니다. 신규/수정된 Firepower Chassis Manager 화면: **Logical Devices(**논리적 디바이스**) > Enable Link State(**링크 상태 활성화**)** 신규/수정된 FXOS 명령: **set link-state-sync enabled**, **show interface expand detail**
컨테이너 인스턴스에 FMC를 사용하여 FTD 구성 백업 및 복원	2.9.1	이제 FTD 컨테이너 인스턴스에서 FMC 백업/복원 도구를 사용할 수 있습니다. 신규/수정된 FMC 화면: **System(**시스템**) > Tools(**도구**) > Backup/Restore(**백업/복원**) > Managed Device Backup(**매니지드 디바이스 백업**)** 신규/수정된 FTD CLI 명령: **restore** 지원되는 플랫폼: Firepower 4100/9300 참고 Firepower 6.7 필요

기능 이름	플랫폼 릴리스	기능 정보
다중 인스턴스 클러스터링	2.8.1	이제 컨테이너 인스턴스로 클러스터를 생성할 수 있습니다. Firepower 9300에서 클러스터의 각 모듈에 하나의 컨테이너 인스턴스를 포함해야 합니다. 보안 엔진/모듈마다 하나 이상의 컨테이너 인스턴스를 추가할 수 없습니다. 각 클러스터 인스턴스에 동일한 보안 모듈 또는 섀시 모델을 사용하는 것이 좋습니다. 그러나 Firepower 9300 보안 모듈 유형이나 Firepower 4100 모델에서는 필요한 경우 동일한 클러스터에서 다른 컨테이너 인스턴스를 혼용해 사용할 수 있습니다. 동일한 클러스터에서 Firepower 9300 및 4100 인스턴스를 혼용할 수 없습니다. 신규/수정된 화면: • 논리적 디바이스 > 클러스터 추가 • **Interfaces(**인터페이스**)** > **All Interfaces(**모든 인터페이스**)** > **Add New(**새로 추가**)** 드롭다운 메뉴 **> Subinterface(**하위 인터페이스**) > Type(**유형**)** 필드 참고 Firepower 6.6 이상이 필요합니다.
FDM로 FTD을 지원	2.7.1	이제 기본 FTD 인스턴스를 구축하고 FDM 관리를 지정할 수 있습니다. 컨테이너 인스턴스는 지원되지 않습니다. 신규/수정된 Firepower Chassis Manager 화면: **Logical Devices(**논리 디바이스**)** > **Add Device(**디바이스 추가**)** > **Settings(**설정**)** > 애플리케이션 인스턴스의 관리 유형 참고 FTD 6.5 이상이 필요합니다.
여러 컨테이너 인스턴스에 대한 TLS 암호화 가속	2.7.1	TLS 암호화 가속은 이제 Firepower 4100/9300 섀시의 여러 컨테이너 인스턴스(최대 16 개)에서 지원됩니다. 이전에는 모듈/보안 엔진 당 하나의 컨테이너 인스턴스에 대해서만 TLS 암호화 가속을 활성화 할 수 있었습니다. 새 인스턴스에는 기본적으로 이 기능이 활성화되어 있습니다. 그러나 업그레이드는 기존 인스턴스에서 가속화를 활성화하지 않습니다. 대신 **enter hw-crypto** 및 **set admin-state enabled** FXOS 명령을 사용합니다. 신규/수정된 Firepower Chassis Manager 화면: **Logical Devices(**논리 디바이스**)** > **Add Device(**디바이스 추가**)** > **Settings(**설정**)** > **Hardware Crypto(**하드웨어 암호화**)** 드롭 다운 메뉴 참고 FTD 6.5 이상이 필요합니다.
Firepower 4115, 4125 및 4145 test	2.6.1	Firepower 4115, 4125, 및 4145를 도입했습니다. 참고 ASA 9.12(1)이 필요합니다. Firepower 6.4.0에는 FXOS 2.6.1.157이 필요합니다. 수정된 화면이 없습니다.

기능 이름	플랫폼 릴리스	기능 정보
Firepower 9300 SM-40, SM-48 및 SM-56지원	2.6.1	다음 세 가지 보안 모듈을 도입했습니다: SM-40, SM-48, SM-56 참고 SM-40 및 SM-48에는 ASA 9.12(1)이 필요합니다. SM-56에는 ASA 9.12(2) 및 FXOS 2.6.1.157이 필요합니다. 모든 모듈에는 FTD 6.4 및 FXOS 2.6.1.157이 필요합니다. 수정된 화면이 없습니다.
동일한 Firepower 9300의 별도의 모듈에서 ASA 및 FTD에 대한 지원	2.6.1	이제 동일한 Firepower 9300에서 ASA 및 FTD 논리적 디바이스를 구축할 수 있습니다. 참고 ASA 9.12(1)이 필요합니다. Firepower 6.4.0에는 FXOS 2.6.1.157이 필요합니다. 수정된 화면이 없습니다.
FTD 부트스트랩 구성의 경우, 이제 Firepower Chassis Manager에서 FMC의 NAT ID를 설정할 수 있습니다.	2.6.1	이제 Firepower Chassis Manager에서 FMC NAT ID를 설정할 수 있습니다. 이전에는 FXOS CLI 또는 FTD CLI 내에서만 NAT ID를 설정할 수 있었습니다. 일반적으로 라우팅 목적과 인증 두 가지 목적에 IP 주소(등록 키와 함께)가 모두 필요합니다. FMC는 디바이스 IP 주소를 지정하고 디바이스는 FMC IP 주소를 지정합니다. 그러나 라우팅을 위한 최소 요구 사항인 IP 주소 중 하나만 알고 있는 경우, 초기 통신에 대한 신뢰를 설정하고 올바른 등록 키를 조회하려면 연결의 양쪽에서 고유 NAT ID도 지정해야 합니다. FMC 및 디바이스는 등록 키 및 NAT ID(IP 주소 대신)를 사용하여 초기 등록을 인증하고 권한을 부여합니다. 신규/수정된 화면: **Logical Devices(**논리적 디바이스**)** > **Add Device(**디바이스 추가**)** > **Settings(**설정**)** > **Firepower Management Center NAT ID** 필드
모듈/보안 엔진에서 하나의 FTD 컨테이너 인스턴스에 대한 SSL 하드웨어 가속 지원	2.6.1	이제 모듈/보안 엔진에서 하나의 컨테이너 인스턴스에 대해 SSL 하드웨어 가속을 활성화할 수 있습니다. SSL 하드웨어 가속은 다른 컨테이너 인스턴스에 대해서는 비활성화되어 있지만 기본 인스턴스에 대해서는 활성화되어 있습니다. 자세한 내용은 FMC 구성 가이드를 참조하십시오. 신규/수정된 명령: **config hwCrypto enable**, **show hwCrypto** 수정된 화면이 없습니다.

기능 이름	플랫폼 릴리스	기능 정보
FTD을 위한 다중 인스턴스 기능	2.4.1	이제 단일 보안 엔진/모듈에서 여러 논리적 디바이스를 각각 FTD 컨테이너 인스턴스와 함께 구축할 수 있습니다. 이전에는 단일 기본 애플리케이션 인스턴스만 구축할 수 있었습니다. 기본 인스턴스도 여전히 지원됩니다. Firepower 9300의 경우 일부 모듈에서는 기본 인스턴스를, 다른 모듈에서는 컨테이너 인스턴스를 사용할 수 있습니다. 물리적 인터페이스를 유연하게 사용할 수 있도록 FXOS에서 VLAN 하위 인터페이스를 생성하는 동시에 여러 인스턴스 간에 인터페이스를 공유할 수 있습니다. 컨테이너 인스턴스를 구축할 때 할당된 CPU 코어 수를 지정해야 합니다. RAM이 코어 수에 따라 동적으로 할당되며, 디스크 공간이 인스턴스당 40GB로 설정됩니다. 이 리소스 관리를 사용하면 각 인스턴스에 대한 성능 기능을 맞춤화할 수 있습니다. 개별 섀시 2개의 컨테이너 인스턴스를 사용하여 고가용성을 사용할 수 있습니다. 예를 들어 각각 인스턴스가 10개인 섀시가 2개 있으면 고가용성 쌍 10개를 생성할 수 있습니다. 클러스터링은 지원되지 않습니다. 참고 다중 인스턴스 기능은 ASA 다중 컨텍스트 모드와 비슷하지만 구현은 서로 다릅니다. 다중 컨텍스트 모드에서는 단일 애플리케이션 인스턴스를 분할하는 반면 다중 인스턴스 기능 사용 시에는 독립적인 컨테이너 인스턴스를 사용할 수 있습니다. 컨테이너 인스턴스에서는 하드 리소스 분리, 별도의 구성 관리/다시 로드/소프트웨어 업데이트가 허용되며 전체 FTD 기능이 지원됩니다. 다중 컨텍스트 모드에서는 리소스가 공유되므로 지정된 플랫폼에서 더 많은 컨텍스트가 지원됩니다. FTD에서는 다중 상황 모드를 사용할 수 없습니다. 참고 FTD 버전 6.3 이상이 필요합니다. 신규/수정된 Firepower Chassis Manager 화면: **Overview(**개요**)** > **Devices(**디바이스**)** **Interfaces(**인터페이스**)** > **All Interfaces(**모든 인터페이스**)** > **Add New(**새로 추가**)** 드롭다운 메뉴 > **Subinterface(**하위 인터페이스**)** **Interfaces(**인터페이스**)** > **All Interfaces(**모든 인터페이스**)** > **Type(**유형**)** **Logical Devices(**논리적 디바이스**)** > **Add Device(**디바이스 추가**)** **Platform Settings(**플랫폼 설정**)** > **MAC Pool(MAC** 풀**)** **Platform Settings(**플랫폼 설정**)** > **Resource Profiles(**리소스 프로파일**)** 신규/수정된 FMC 화면: **Devices(**디바이스**)** > **Device Management(**디바이스 관리**)** > **Edit(**수정**)** 아이콘 > **Interfaces(**인터페이스**)** 탭

기능 이름	플랫폼 릴리스	기능 정보
ASA 논리적 디바이스에 대한 투명 모드 구축 지원	2.4.1	이제 ASA를 구축할 때 투명 또는 라우팅된 모드를 지정할 수 있습니다. 신규/수정된 Firepower Chassis Manager 화면: **Logical Devices(**논리적 디바이스**)** > **Add Device(**디바이스 추가**)** > **Settings(**설정**)** 신규/수정된 옵션: **Firewall Mode(**방화벽 모드**)** 드롭다운 목록
클러스터 제어 링크 사용자 정의 가능한 IP 주소	2.4.1	기본적으로 클러스터 제어 링크는 127.2.0.0/16 네트워크를 사용합니다. 이제 FXOS에서 클러스터를 구축하는 경우 네트워크를 설정할 수 있습니다. 섀시에서는 섀시 ID 및 슬롯 ID 127.2.*chassis_id*.*slot_id*를 기준으로 하여 각 유닛에 대해 클러스터 제어 링크 인터페이스 IP 주소를 자동 생성합니다. 그러나 일부 네트워킹 구축에서는 127.2.0.0/16 트래픽 통과를 허용하지 않습니다. 따라서 이제 FXOS에서 루프백(127.0.0.0/8) 및 멀티캐스트(224.0.0.0/4) 주소를 제외하고 클러스터 제어 링크의 맞춤형 /16 서브넷을 설정할 수 있습니다. 신규/수정된 화면: **Logical Devices(**논리적 디바이스**)** > **Add Device(**디바이스 추가**)** > **Cluster Information(**클러스터 정보**)** > **CCL Subnet IP(CCL** 서브넷 **IP)** 필드
FTD 부트스트랩 구성의 경우 이제 FXOS CLI에서 FMC의 NAT ID를 설정할 수 있습니다.	2.4.1	이제 FXOS CLI에서 FMC NAT ID를 설정할 수 있습니다. 이전에는 FTD CLI 내에서만 NAT ID를 설정할 수 있었습니다. 일반적으로 라우팅 목적과 인증 두 가지 목적에 IP 주소(등록 키와 함께)가 모두 필요합니다. FMC는 디바이스 IP 주소를 지정하고 디바이스는 FMC IP 주소를 지정합니다. 그러나 라우팅을 위한 최소 요구 사항인 IP 주소 중 하나만 알고 있는 경우, 초기 통신에 대한 신뢰를 설정하고 올바른 등록 키를 조회하려면 연결의 양쪽에서 고유 NAT ID도 지정해야 합니다. FMC 및 디바이스는 등록 키 및 NAT ID(IP 주소 대신)를 사용하여 초기 등록을 인증하고 권한을 부여합니다. 신규/수정된 명령: **enter bootstrap-key NAT_ID**
ASA에 대한 사이트 간 클러스터링 개선	2.1.1	이제 ASA 클러스터를 구축할 때 각 Firepower 4100/9300 섀시에 대한 사이트 ID를 구성할 수 있습니다. 전에는 ASA 애플리케이션 내에서 사이트 ID를 구성해야 했습니다. 이 기능 덕분에 초기 구축이 수월해졌습니다. 더 이상 ASA 구성 내에서 사이트 ID를 설정할 수 없습니다. 또한 사이트 간 클러스터링과의 호환성을 최대한 활용하려면 안정성과 성능이 개선된 ASA 9.7(1) 및 FXOS 2.1.1로 업그레이드하는 것이 좋습니다. 수정된 화면: **Logical Devices(**논리적 디바이스**)** > **Configuration(**구성**)**
Firepower 9300의 6개 FTD 모듈에 대한 섀시 간 클러스터링	2.1.1	이제 Firepower 9300에서 FTD을 위해 섀시 간 클러스터링을 활성화할 수 있습니다. 최대 6개의 모듈을 포함할 수 있습니다. 예를 들어 섀시 6개에 모듈 1개, 섀시 3개에 모듈 2개, 또는 모듈을 6개까지 제공하는 어떤 조합도 사용할 수 있습니다. 수정된 화면: **Logical Devices(**논리적 디바이스**)** > **Configuration(**구성**)**

기능 이름	플랫폼 릴리스	기능 정보
Firepower 4100에서 FTD 클러스터링 지원	2.1.1	FTD 클러스터에서 최대 6개의 섀시를 클러스터링할 수 있습니다.
ASA 클러스터에서 16 Firepower 4100 섀시에 대한 지원	2.0.1	ASA 클러스터에서 최대 16개의 섀시를 클러스터링할 수 있습니다.
Firepower 4100에서 ASA 클러스터링에 대한 지원	1.1.4	ASA 클러스터에서 최대 6개의 섀시를 클러스터링할 수 있습니다.
Firepower 9300의 FTD에서 인트라 섀시 클러스터링(intra-chassis clustering) 지원	1.1.4	Firepower 9300은 FTD 애플리케이션이 있는 인트라 섀시 클러스터링(intra-chassis clustering)을 지원합니다. 수정된 화면: **Logical Devices(**논리적 디바이스**)** > **Configuration(**구성**)**
Firepower 9300에서 ASA 모듈 16개를 위한 인트라 섀시 클러스터링(intra-chassis clustering)	1.1.3	현재 ASA를 위한 섀시 간 클러스터링을 활성화할 수 있습니다. 최대 16개의 모듈을 포함할 수 있습니다. 예를 들어 섀시 16개에 모듈 1개, 섀시 8개에 모듈 2개, 또는 모듈을 16개까지 제공하는 어떤 조합도 사용할 수 있습니다. 수정된 화면: **Logical Devices(**논리적 디바이스**)** > **Configuration(**구성**)**
Firepower 9300에서 ASA를 위한 인트라 섀시 클러스터링(intra-chassis clustering)	1.1.1	Firepower 9300 섀시 내부에서 모든 ASA 보안 모듈을 클러스터링할 수 있습니다. 추가된 화면: **Logical Devices(**논리적 디바이스**)** > **Configuration(**구성**)**

11 장

보안 모듈/엔진 관리

FXOS 보안 모듈/보안 엔진 정보

Firepower Chassis Manager의 Security Modules/Security Engine(보안 모듈/보안 엔진 정보) 페이지에서 보안 모듈/엔진의 상태를 보고 보안 모듈/엔진에서 다양한 기능을 수행할 수 있습니다.

Security Modules/Security Engine(보안 모듈/보안 엔진) 페이지에서는 다음 정보를 제공합니다.

- Hardware State(하드웨어 상태) - 보안 모듈/엔진 하드웨어의 상태를 보여줍니다.
  - Up(가동) - 보안 모듈/엔진의 전원이 성공적으로 켜졌으며 보안 모듈/엔진에 연결된 논리적 디바이스가 없는 경우에도 에 하드웨어 결함이 표시되지 않습니다.
  - Booting Up(부팅 중) - 보안 모듈/엔진의 전원을 켜는 중입니다.
  - Restarting(재시작) - 보안 모듈/엔진이 재시작되는 중입니다.
  - Down(중단) - 보안 모듈/엔진의 전원이 켜지지 않았거나, 하드웨어 장애 때문에 보안 모듈/엔진을 성공적으로 시작할 수 없습니다.
  - Mismatch(불일치) - 보안 모듈이 해제되었거나 슬롯에 새 보안 모듈이 설치되었습니다. Acknowledge(확인) 기능을 사용하여 보안 모듈을 작동 상태로 되돌립니다.
  - Empty(비어 있음) - 보안 모듈이 해당 슬롯에 설치되어 있지 않습니다.
- Service State(서비스 상태) - 보안 모듈/엔진에서 소프트웨어의 상태를 보여줍니다.

- Not-available(사용 불가) - 보안 모듈이 섀시 슬롯에서 제거되었습니다. 보안 모듈을 정상적인 작동 상태로 전환하려면 다시 설치합니다.
- Online(온라인) - 보안 모듈/엔진이 설치되었고 정상 작동 모드에 있습니다.
- Not Responding(응답 없음) - 보안 모듈/엔진이 응답하지 않습니다.
- Token Mismatch(토큰 불일치) - 전에 구성된 것이 아닌 보안 모듈이 섀시 슬롯에 설치되었음을 나타냅니다. 또한 소프트웨어 설치 오류로 인해 발생할 수도 있습니다. 보안 모듈을 작동 상태로 전환하려면 Reinitialize(다시 초기화) 기능을 사용합니다.
- Fault(장애) - 보안 모듈/엔진이 장애 상태에 있습니다. 결함 상태를 일으킬 수 있는 것에 대해 자세히 알아보려면 시스템 결함 목록을 검토하십시오. 결함의 정보 아이콘에 마우스를 올려 추가 정보를 확인할 수도 있습니다.

보안 모듈 결함

- Failsafe Mode(페일세이프 모드) - 보안 모듈이 페일세이프 모드입니다. 이 모드에서는 애플리케이션 시작이 차단됩니다. 트러블슈팅하거나 페일세이프 모드를 비활성화하려면 보안 모듈에 연결합니다. 앱 인스턴스를 삭제할 수도 있습니다.
- HDD Error(HDD 오류) - 보안 모듈 디스크 드라이브에 오류가 있습니다. 디스크 드라이브가 있는지 확인하고 결함이 해결되지 않으면 결함이 있는 디스크 드라이브를 교체합니다.
- Filesystem Error(파일 시스템 오류) - 보안 모듈의 디스크 파티션이 호환되지 않습니다. 보안 모듈을 리부팅하면 결함이 복구될 수도 있습니다. 결함이 지속되면 외부 디바이스에 데이터를 백업한 후 슬롯을 다시 초기화하십시오.
- Format Failure(포맷 장애) - 보안 모듈 디스크 드라이브에서 자동 포맷 시 장애가 발생했습니다. 보안 모듈을 다시 초기화하여 다시 포맷하십시오.

- Power(전원) - 보안 모듈/엔진의 전원 상태를 보여줍니다.
  - On(켜짐) - 보안 모듈/엔진의 전원 상태를 전환하려면 전원 끄기/켜기 기능을 사용합니다.
  - Off(꺼짐) - 보안 모듈/엔진의 전원 상태를 전환하려면 전원 끄기/켜기 기능을 사용합니다.
- Application(애플리케이션) - 보안 모듈/엔진에 설치된 논리적 디바이스 유형을 보여줍니다.

Firepower Chassis Manager의 Security Modules/Security Engine(보안 모듈/보안 엔진) 페이지에서 보안 모듈/엔진에 대해 다음 기능을 수행할 수 있습니다.

- Decommission(해제, 보안 모듈만 해당) - 보안 모듈을 해제하면 보안 모듈이 유지 관리 모드로 전환됩니다. 또한 특정 오류상태를 수정하려면 보안 모듈을 해제한 후 승인할 수도 있습니다. 보안 모듈 해제, 299 페이지을 참조하십시오.
- Acknowledge(확인) - 새로 설치된 보안 모듈을 온라인 상태로 전환합니다. 보안 모듈/엔진 승인, 299 페이지을 참조하십시오.
- Power Cycle(전력 사이클) 보안 모듈/엔진을 재시작합니다. 보안 모듈/엔진 전원 껐다 켜기, 300 페이지을 참조하십시오.

- Reinitialize(다시 초기화) - 보안 모듈/엔진 하드 디스크를 다시 포맷하여 모든 구축된 애플리케이션과 구성을 보안 모듈/엔진에서 제거한 다음 시스템을 다시 시작합니다. 다시 초기화를 완료한 후, 보안 모듈/엔진에 대해 논리적 디바이스가 구성되어 있으면 FXOS에서는 애플리케이션 소프트웨어를 다시 설치하고, 논리적 디바이스를 재구축하고, 애플리케이션을 자동으로 시작합니다. 보안 모듈/엔진 확인 다시 초기화, 300 페이지을 참조하십시오.

경고! 보안 모듈/엔진의 모든 애플리케이션 데이터는 다시 초기화하는 동안 삭제됩니다. 보안 모듈/엔진을 다시 초기화하기 전에 모든 애플리케이션 데이터를 백업하십시오.

- 전원 끄기/켜기 - 보안 모듈/엔진의 전원 상태를 전환합니다. 보안 모듈/엔진 전원 껐다 켜기, 300 페이지의 내용을 참조하십시오.

보안 모듈 해제

보안 모듈을 해제하면, 보안 모듈 객체가 구성에서 삭제되고 보안 모듈은 관리되지 않는 상태가 됩니다. 보안 모듈에서 실행되는 모든 논리적 디바이스 또는 소프트웨어는 비활성 상태가 됩니다.

보안 모듈의 사용을 일시적으로 중단하려는 경우 보안 모듈을 해제할 수 있습니다.

프로시저

단계 **1** **Security Modules(**보안 모듈**)**를 선택하여 Security Modules(보안 모듈) 페이지를 엽니다.

단계 **2** 보안 모듈을 해제하려면 해당 보안 모듈에 대해 **Decommission(**디커미션**)**을 클릭합니다.

단계 **3** **Yes(**예**)**를 클릭하여 지정된 보안 모듈의 해제 또는 재위임을 확인합니다.

보안 모듈/엔진 승인

새 보안 모듈을 섀시에 설치하거나 기존 모듈을 PID(제품 ID)가 다른 모듈로 교체하는 경우 보안 모듈 사용을 시작하려면 해당 모듈을 승인해야 합니다.

보안 모듈의 상태가 "mismatch(불일치)" 또는 "token mismatch(토큰 불일치)"로 표시되는 경우 슬롯에 설치된 보안 모듈에 이전에 슬롯에 설치되었던 것과 일치하지 않는 데이터가 있는 것입니다. 보안 모듈에 기존의 데이터가 있고 이것을 새 슬롯에서 사용하려는 경우(다시 말하면, 보안 모듈을 실수로 잘못된 슬롯에 설치한 것이 아닌 경우), 여기에 논리적 디바이스를 구축하려면 먼저 보안 모듈을 다시 초기화해야 합니다.

프로시저

단계 **1** **Security Modules/Security Engine(**보안 모듈/보안 엔진**)**을 선택하여 Security Modules/Security Engine(보안 모듈/보안 엔진) 페이지를 엽니다.

단계 **2** 확인할 보안 모듈/엔진에 대해 **Acknowledge(**확인**)**를 클릭합니다.

단계 **3** **Yes(**예**)**를 클릭하여 지정된 보안 모듈/엔진을 확인합니다.

보안 모듈/엔진 전원 껐다 켜기

다음 단계에 따라 보안 모듈/엔진의 전원을 껐다가 켭니다.

프로시저

단계 **1** **Security Modules/Security Engine(**보안 모듈/보안 엔진**)**을 선택하여 Security Modules/Security Engine(보안 모듈/보안 엔진) 페이지를 엽니다.

단계 **2** 재부팅하려는 보안 모듈/엔진에 대해 **Power Cycle(**전원 주기**)**를 클릭합니다.

단계 **3** 다음 중 하나를 수행합니다.

- 지정된 보안 모듈/엔진의 전원을 껐다가 켜기 전에 시스템이 보안 모듈/엔진에서 실행 중인 애플리케이션을 셧다운하도록 최대 5분간 기다리려면 **Safe Power Cycle(**안전하게 전원 껐다 켜기**)**을 클릭합니다.
- 시스템이 지정된 보안 모듈/엔진을 즉시 껐다가 켜려면 **Power Cycle Immediately(**즉시 전원 껐다 켜기**)**를 클릭합니다.

보안 모듈/엔진 확인 다시 초기화

보안 모듈/엔진을 다시 초기화하면 보안 모듈/엔진 하드 디스크가 포맷되고 설치된 모든 애플리케이션 인스턴스, 구성 및 데이터가 제거됩니다. 다시 초기화를 완료한 후, 보안 모듈/엔진에 대해 논리적 디바이스가 구성되어 있으면 FXOS에서는 애플리케이션 소프트웨어를 다시 설치하고, 논리적 디바이스를 재구축하고, 애플리케이션을 자동으로 시작합니다.

⚠

주의 보안 모듈/엔진의 모든 애플리케이션 데이터는 다시 초기화하는 동안 삭제됩니다. 보안 모듈/엔진을 다시 초기화하기 전에 모든 애플리케이션 데이터를 백업하십시오.

프로시저

단계 **1** **Security Modules/Security Engine(**보안 모듈/보안 엔진**)**을 선택하여 Security Modules/Security Engine(보안 모듈/보안 엔진) 페이지를 엽니다.

단계 **2** 다시 초기화할 보안 모듈/엔진에 대해 **Reinitialize(**다시 초기화**)**를 클릭합니다.

단계 **3** **Yes(**예**)**를 클릭하여 지정된 보안 모듈/엔진의 다시 초기화를 확인합니다.

보안 모듈/엔진이 다시 시작되고 보안 모듈의 모든 데이터가 삭제됩니다. 이 작업은 몇 분 정도 걸릴 수 있습니다.

네트워크 모듈 승인

섀시에 새 네트워크 모듈을 설치되거나 기존 모듈을 제품 ID(PID)가 다른 모듈로 교체하면 사용을 시작하기 전에 네트워크 모듈을 승인해야 합니다.

프로시저

단계 **1** `scope fabric-interconnect` 모드를 입력합니다.

```
scope fabric-interconnect
```

단계 **2** 새 모듈을 설치하거나 네트워크 모듈을 유형이 다른(즉, PID가 다른) 다른 모듈로 교체한 후 **acknowledge(**승인**)** 명령을 입력하십시오.

```
acknowledge
```

예제**:**

```
FPR1 /fabric-interconnect # acknowledge
  fault  Fault
  slot   Card Config Slot Id  <======
```

단계 **3** 삽입된 슬롯을 승인할 **acknowledge slot(**승인 슬롯**)** 을 입력합니다.

```
acknowledge slot
```

예제**:**

```
FPR1 /fabric-interconnect # acknowledg slot 2
  0-4294967295  Slot Id
```

단계 **4** 구성을 커밋합니다.

```
commit-buffer
```

네트워크 모듈 오프라인 또는 온라인 설정

CLI 명령을 사용하여 네트워크 모듈을 오프라인으로 설정하거나 다시 온라인으로 설정하려면 다음 단계를 수행합니다. 이는 모듈 OIR(온라인 삽입 및 제거) 수행 시 예로 사용된 단계입니다.

참고

- 네트워크 모듈을 제거하고 교체하는 경우 장치에 적절한 설치 가이드의 "유지 보수 및 업그레이드" 장의 지침을 따르십시오. https://www.cisco.com/c/en/us/support/security/firepower-ngfw/products-installation-guides-list.html를 참조하십시오.
- 8 포트 1G Copper FTW 네트워크 모듈(FPR-NM-8X1G-F FTW)에서 네트워크 모듈 온라인 삽입 및 제거(OIR)를 수행하는 경우, 이 절차를 사용하여 카드를 온라인 상태로 전환할 때까지 네트워크 모듈 LED가 꺼진 상태로 유지됩니다. 먼저 LED가 황색으로 깜박인 다음 네트워크 모듈이 검색되고 애플리케이션이 온라인 상태가 되면 녹색으로 변경됩니다.

참고 FTW 네트워크 모듈을 제거하고 슬롯을 승인하면 네트워크 모듈 포트가 FTD 논리적 디바이스에서 삭제됩니다. 이 경우 네트워크 모듈을 다시 삽입하기 전에 FMC를 사용하여 하드웨어 우회 인라인 집합 구성을 삭제해야 합니다. 네트워크 모듈을 다시 삽입한 후에는 다음을 수행해야 합니다.

- Firepower Chassis Manager 또는 FXOS CLI(Command Line Interface)를 사용하여 네트워크 모듈 포트를 온라인 관리 상태로 구성합니다.
- 네트워크 모듈 포트를 FTD 논리적 디바이스에 추가하고 FMC를 사용하여 포트를 재구성합니다.

슬롯을 승인하지 않고 네트워크 모듈을 제거하면, 인라인 집합 구성이 유지되고 포트가 FMC에 down(다운)으로 표시됩니다. 네트워크 모듈을 다시 삽입하면, 이전 구성이 복원됩니다.

인라인 집합의 하드웨어 우회에 대한 자세한 내용은 하드웨어 바이패스 쌍, 169 페이지을 참조하십시오.

프로시저

단계 **1** 모듈을 오프라인으로 설정하려면 다음 명령을 사용하여 `/fabric-interconnect` 모드를 시작한 다음 `/card` 모드를 시작합니다.

```
scope fabric-interconnect a
scope card ID
```

단계 **2** **show detail** 명령을 사용하면 현재 상태를 비롯하여 이 카드에 대한 정보를 볼 수 있습니다.

단계 **3** 모듈을 오프라인으로 설정하려면 다음을 입력합니다.

```
set adminstate offline
```

단계 **4** 구성 변경 사항을 저장하려면 **commit-buffer** 명령을 입력합니다.

모듈이 오프라인 상태인지 확인하려면 **show detail** 명령을 다시 사용할 수 있습니다.

단계 **5** 네트워크 모듈을 다시 온라인 상태로 설정하려면 다음을 입력합니다.

```
set adminstate online
commit-buffer
```

예

```
FP9300-A# scope fabric-interconnect a
FP9300-A /fabric-interconnect # scope card 2
FP9300-A /fabric-interconnect/card # show detail

Fabric Card:
    Id: 2
    Description: Firepower 4x40G QSFP NM
    Number of Ports: 16
    State: Online
    Vendor: Cisco Systems, Inc.
    Model: FPR-NM-4X40G
    HW Revision: 0
    Serial (SN): JAD191601DE
    Perf: N/A
    Admin State: Online
    Power State: Online
    Presence: Equipped
    Thermal Status: N/A
    Voltage Status: N/A
FP9300-A /fabric-interconnect/card # set adminstate offline
FP9300-A /fabric-interconnect/card* # commit-buffer
FP9300-A /fabric-interconnect/card # show detail

Fabric Card:
    Id: 2
    Description: Firepower 4x40G QSFP NM
    Number of Ports: 16
    State: Offline
    Vendor: Cisco Systems, Inc.
    Model: FPR-NM-4X40G
    HW Revision: 0
    Serial (SN): JAD191601DE
    Perf: N/A
    Admin State: Offline
    Power State: Off
    Presence: Equipped
    Thermal Status: N/A
    Voltage Status: N/A
FP9300-A /fabric-interconnect/card #
```

블레이드 상태 모니터링

Failsafe는 블레이드에서 예기치 않은 애플리케이션 재시작이 지정된 횟수만큼 탐지되면 보안 모듈 또는 엔진에서 무한 부트 루프 상태를 방지하기 위해 작동하며, 이로 인해 이중화 HA 또는 클러스터 구축에서 추가 부작용이 발생할 수 있습니다.

블레이드 플랫폼은 주기적으로 상태 확인을 수행하여 MIO에 보고합니다. 블레이드가 실패 상태인 경우 결함 및 오류 메시지가 표시됩니다.

결함 및 오류 메시지

블레이드에 문제가 있는 경우 플랫폼의 Overview(개요) 페이지에서 결함 및 오류 메시지를 볼 수 있습니다.

- Overview(개요) 페이지 - 보안 모듈에 작동 상태가 Fault(결함)인 결함 기호가 표시됩니다.
- Security Module(보안 모듈) 페이지 - 블레이드의 Service State(서비스 상태)가 Fault(결함)로 표시됩니다. 마우스를 올려놓으면 'i' 아이콘에 오류 메시지가 표시됩니다.
- Logical Devices(논리적 디바이스) 페이지 - 논리적 디바이스를 사용할 수 있고 보안 모듈에 결함이 있는 경우 마우스를 올려놓으면 "i" 아이콘에 오류 메시지가 표시됩니다.

참고 FXOS CLI에서 failsafe 설정을 구성하고 관리할 수 있습니다.

12 장

구성 가져오기/내보내기

구성 가져오기/내보내기 정보

구성 내보내기 기능을 사용하여 Firepower 4100/9300 섀시의 논리적 디바이스 및 플랫폼 구성 설정을 포함하는 XML 파일을 원격 서버 또는 로컬 컴퓨터로 내보낼 수 있습니다. 나중에 해당 구성 파일을 가져와서 구성 설정을 Firepower 4100/9300 섀시에 빠르게 적용하여, 알려진 정상적인 구성으로 돌아가거나 시스템 장애로부터 복구할 수 있습니다.

지침 및 제한 사항

- FXOS 2.6.1부터 이제 암호화 키를 구성할 수 있습니다. 암호화 키를 설정해야 구성을 내보낼 수 있습니다. 해당 구성을 가져올 때는 시스템에 동일한 암호화 키가 설정되어 있어야 합니다. 내보내기 수행 시 사용한 암호화 키와 다르게 암호화 키를 수정하는 경우 가져오기 작업이 실패합니다. 내보낸 각 설정에 대한 암호화 키를 기억해 두어야 합니다.
- 구성 파일의 내용을 수정하지 마십시오. 구성 파일을 수정하면 해당 파일을 사용한 구성 가져오기가 실패할 수 있습니다.
- 애플리케이션 관련 구성 설정은 구성 파일에 포함되지 않습니다. 애플리케이션 관련 설정 및 구성을 관리하려면 애플리케이션에서 제공하는 구성 백업 도구를 사용해야 합니다.
- Firepower 4100/9300 섀시에서 설정을 가져오면 Firepower 4100/9300 섀시에 있는 모든 기존의 설정(논리적 디바이스 포함)이 삭제되고 가져오기 파일에 포함된 설정으로 완전히 교체됩니다.
- RMA 시나리오를 제외하고 설정을 내보낸 곳과 동일한 Firepower 4100/9300 섀시로 설정 파일만 가져오는 것이 좋습니다.

- 구성을 가져오는 Firepower 4100/9300 섀시의 플랫폼 소프트웨어 버전은 내보낼 때와 동일한 버전이어야 합니다. 버전이 다르면 가져오기 작업의 성공이 보장되지 않습니다. Firepower 4100/9300 섀시를 업그레이드 또는 다운그레이드할 때마다 백업 설정을 내보내는 것이 좋습니다.
- 구성을 가져오는 Firepower 4100/9300 섀시에는 내보냈을 때와 동일한 슬롯에 동일한 네트워크 모듈이 설치되어 있어야 합니다.
- 구성을 가져오는 Firepower 4100/9300 섀시에는, 가져오는 내보내기 파일에 정의된 논리적 디바이스에 대해 올바른 소프트웨어 애플리케이션 이미지가 설치되어 있어야 합니다.
- 애플리케이션에 EULA(End-User License Agreement)가 있는 논리적 디바이스가 가져오는 설정 파일에 포함되어 있으면, 설정을 가져오기 전에 Firepower 4100/9300 섀시에서 해당 애플리케이션의 EULA에 동의해야 합니다. 아니면 작업이 실패합니다.
- 기존 백업 파일을 덮어쓰지 않으려면, 백업 작업 시 파일 이름을 변경하거나 기존 파일을 다른 위치에 복사합니다.

참고 FXOS 가져오기/내보내기는 FXOS 구성만 백업하므로 논리적 앱을 별도로 백업해야 합니다. FXOS 구성 가져오기로 인해 논리적 디바이스가 재부팅되고 디바이스가 공장 기본 구성으로 재구축됩니다.

구성 가져오기/내보내기를 위한 암호화 키 설정

구성을 내보낼 때 FXOS에서는 비밀번호 및 키와 같은 민감한 데이터를 암호화합니다.

FXOS 2.6.1부터 이제 암호화 키를 구성할 수 있습니다. 암호화 키를 설정해야 구성을 내보낼 수 있습니다. 해당 구성을 가져올 때는 시스템에 동일한 암호화 키가 설정되어 있어야 합니다. 내보내기 수행 시 사용한 암호화 키와 다르게 암호화 키를 수정하는 경우 가져오기 작업이 실패합니다. 내보낸 각 구성에 사용한 암호화 키를 기억해 두어야 합니다.

암호화 키는 Export(내보내기) 페이지 또는 Import(가져오기) 페이지에서 설정할 수 있습니다. 단, 이를 설정하면 내보내기와 가져오기에 모두 동일한 키가 사용됩니다.

FXOS 2.6.1 이전 릴리스에서 내보낸 구성을 FXOS 2.6.1 이상 버전으로 가져오는 경우, 시스템에서는 암호화 키를 확인하지 않고 가져오기를 허용합니다.

참고 구성을 가져오는 플랫폼 소프트웨어 버전은 내보낼 때와 동일한 버전이 아니며 가져오기 작업이 성공한다는 보장은 없습니다. Firepower 4100/9300 섀시를 업그레이드 또는 다운그레이드할 때마다 백업 구성을 내보내는 것이 좋습니다.

새 시작 버전이 업그레이드된 버전의 소프트웨어 릴리스와 일치하도록 FTD논리적 어플라이언스가 새 소프트웨어로 업그레이드될 때마다 'Set Version(버전 설정)' 옵션을 사용하고 백업 구성을 내보냅니다.

프로시저

단계 1 **System(**시스템**)** > **Configuration(**구성**)** > **Export(**내보내기**)**를 선택합니다.

단계 2 **Encryption(**암호화**)** 아래의 **Key(**키**)** 필드에서 민감한 데이터 암호화/암호 해독에 사용할 키를 입력합니다. 암호화 키의 길이는 4~40자여야 합니다.

단계 3 **Save Key(**키 저장**)**를 클릭합니다.

암호화 키가 설정되어 있으며, 이는 구성을 내보내고 가져올 때 민감한 데이터 암호화/암호 해독에 사용됩니다. 암호화 키가 설정되었음을 나타내기 위해 *Key(*키*)* 필드 옆에 **Set:Yes(**설정**:** 예**)**가 표시됩니다.

FXOS 구성 파일 내보내기

Firepower 4100/9300 섀시의 논리적 디바이스 및 플랫폼 구성 설정을 포함하는 XML 파일을 원격 서버 또는 로컬 컴퓨터로 내보내려면 구성 내보내기 기능을 사용합니다.

시작하기 전에

구성 가져오기/내보내기 정보을 검토합니다.

프로시저

단계 1 에서 **System(**시스템**)** > **Configuration(**설정**)** > **Export(**내보내기**)**를 선택합니다.

단계 2 구성 파일을 로컬 컴퓨터로 내보내려면 **Export Locally(**로컬로 내보내기**)**를 클릭합니다.
구성 파일이 생성되고, 브라우저에 따라 기본 다운로드 위치로 파일이 자동으로 다운로드되거나 파일을 저장하라는 프롬프트가 표시될 수 있습니다.

단계 3 구성 파일을 미리 구성된 원격 서버로 내보내려면 사용할 원격 구성에 대해 **Export(**내보내기**)**를 클릭합니다.
구성 파일이 생성되고 지정된 위치로 내보내기가 수행됩니다.

단계 4 구성 파일을 새로운 원격 서버로 내보내려면:

a) On-Demand Export(온디맨드 내보내기) 아래에서 **Add On-Demand Configuration(**온디맨드 구성 추가**)**을 클릭합니다.

b) 원격 서버와의 통신에서 사용할 프로토콜을 선택합니다. FTP, TFTP, SCP, SFTP 중 하나일 수 있습니다.

c) 백업 파일을 저장할 위치의 IP 주소 또는 호스트 이름을 입력합니다. 이는 Firepower 4100/9300 섀시가 네트워크를 통해 액세스할 수 있는 서버, 스토리지 어레이, 로컬 드라이브, 기타 읽기/쓰기 미디어일 수 있습니다.

IP 주소가 아니라 호스트 이름을 사용하는 경우 DNS 서버를 구성해야 합니다.

d) 기본값 이외의 포트를 사용하려는 경우 **Port(**포트**)** 필드에 포트 번호를 입력합니다.

e) 시스템이 원격 서버에 로그인할 때 사용할 사용자 이름을 입력합니다. 프로토콜이 TFTP일 경우 이 필드는 적용되지 않습니다.
f) 원격 서버 사용자 이름의 비밀번호를 입력합니다. 프로토콜이 TFTP일 경우 이 필드는 적용되지 않습니다.
g) **Location(**위치**)** 필드에 구성 파일을 내보낼 전체 경로(파일 이름 포함)를 입력합니다.
h) **OK(**확인**)**를 클릭합니다.
On-Demand Export(온디맨드 내보내기) 테이블에 원격 구성이 추가됩니다.
i) 사용할 원격 구성에 대해 **Export(**내보내기**)**를 클릭합니다.
구성 파일이 생성되고 지정된 위치로 내보내기가 수행됩니다.

자동 구성 내보내기 예약

Firepower 4100/9300 섀시의 논리적 디바이스 및 플랫폼 구성 설정을 포함하는 XML 파일을 원격 서버 또는 로컬 컴퓨터로 자동으로 내보내려면 예약된 내보내기 기능을 사용합니다. 내보내기를 매일, 매주 또는 2주마다 실행하도록 예약할 수 있습니다. 구성 내보내기는 예약된 내보내기 기능이 활성화된 시기를 기반으로 예약에 따라 실행됩니다. 예를 들어 매주 수요일 오후 10시에 내보내기를 예약한 경우 시스템은 수요일마다 오후 10시에 새로운 내보내기를 트리거합니다.

구성 내보내기 기능 사용에 대한 중요한 정보는 구성 가져오기/내보내기 정보을 참조하십시오.

프로시저

단계 **1** **System(**시스템**)** > **Configuration(**구성**)** > **Export(**내보내기**)**를 선택합니다.

단계 **2** **Schedule Export(**내보내기 예약**)**를 클릭합니다.
Configure Scheduled Export(예약된 내보내기 구성**)** 대화 상자가 표시됩니다.

단계 **3** 원격 서버와의 통신에서 사용할 프로토콜을 선택합니다. FTP, TFTP, SCP, SFTP 중 하나일 수 있습니다.

단계 **4** 예약된 내보내기를 활성화하려면 **Enable(**활성화**)** 확인란을 선택합니다.

참고 나중에 이 확인란을 사용하여 예약 내보내기를 활성화 또는 비활성화할 수 있습니다. 그러나 예약된 내보내기를 활성화 또는 비활성화할 때 비밀번호를 다시 지정해야 합니다.

단계 **5** 백업 파일을 저장할 위치의 IP 주소 또는 호스트 이름을 입력합니다. 이는 Firepower 4100/9300 섀시가 네트워크를 통해 액세스할 수 있는 서버, 스토리지 어레이, 로컬 드라이브, 기타 읽기/쓰기 미디어일 수 있습니다.

IP 주소가 아니라 호스트 이름을 사용하는 경우 DNS 서버를 구성해야 합니다.

단계 **6** 기본값 이외의 포트를 사용하려는 경우 **Port(**포트**)** 필드에 포트 번호를 입력합니다.

단계 **7** 시스템이 원격 서버에 로그인할 때 사용할 사용자 이름을 입력합니다. 프로토콜이 TFTP일 경우 이 필드는 적용되지 않습니다.

단계 **8** 원격 서버 사용자 이름의 비밀번호를 입력합니다. 프로토콜이 TFTP일 경우 이 필드는 적용되지 않습니다.

단계 **9** **Location(**위치**)** 필드에 구성 파일을 내보낼 전체 경로(파일 이름 포함)를 입력합니다. 파일 이름을 생략할 경우 내보내기 절차에서 파일에 이름을 할당합니다.

단계 **10** 구성 자동 내보내기를 수행할 일정을 선택합니다. Daily(매일), Weekly(매주) 또는 BiWeekly(격주) 중 하나일 수 있습니다.

단계 **11** **OK(**확인**)**를 클릭합니다.
예약된 내보내기가 생성됩니다. 예약된 내보내기를 활성화하면, 선택한 일정에 따라 시스템이 지정된 위치로 구성 파일을 자동으로 내보냅니다.

구성 내보내기 미리 알림 설정

특정 일수에 구성 내보내기가 실행되지 않은 경우 시스템에서 오류를 생성하도록 하려면 Export Reminder(내보내기 알림) 기능을 사용합니다.

기본적으로 내보내기 알림은 30일 간격으로 활성화됩니다.

참고 알림 빈도가 예약된 내보내기 정책의 일 수(매일, 매주 또는 격주)보다 작은 경우 내보내기 알림 오류 메시지("구성 백업이 오래되었을 수 있습니다")를 받게 됩니다. 예를 들어 내보내기 일정이 매주이고 알림 빈도가 5일인 경우, 해당 시간에 구성을 내보내지 않으면 이 오류 메시지가 5일마다 발생합니다.

프로시저

단계 **1** **System(**시스템**)** > **Configuration(**구성**)** > **Export(**내보내기**)**를 선택합니다.

단계 **2** 구성 내보내기 미리 알림을 활성화하려면 **Reminder to trigger an export(**내보내기 트리거 미리 알림**)** 아래에서 확인란을 선택합니다.

단계 **3** 미리 알림 오류를 생성하기 전에 시스템이 구성 내보내기 사이에 대기해야 할 일수(1~365)를 입력합니다.

단계 **4** **Save Reminder(**미리 알림 저장**)**를 클릭합니다.

구성 파일 가져오기

Firepower 4100/9300 섀시에서 전에 내보낸 구성 설정을 적용하려면 구성 가져오기 기능을 사용할 수 있습니다. 이 기능을 사용하면 알려진 양호한 구성으로 돌아가거나 시스템 장애로부터 복구할 수 있습니다.

시작하기 전에

구성 가져오기/내보내기 정보을 검토합니다.

프로시저

단계 **1** 에서 **System(**시스템**)** > **Tools(**도구**)** > **Import(**가져오기/내보내기**)**를 선택합니다.

단계 **2** 로컬 구성 파일로부터 가져오려면:

a) **Choose File(**파일 선택**)**을 클릭하고 가져올 구성 파일을 찾아 선택합니다.

b) **Import(**가져오기**)**를 클릭합니다.
확인 대화 상자가 열리면서 계속 진행할 것인지를 물어보고 섀시를 재시작해야 한다고 경고합니다.

c) **Yes(**예**)**를 클릭하여 지정된 구성 파일을 가져올 것임을 확인합니다.
기존의 구성이 삭제되고, 가져오기 파일에 지정된 구성이 Firepower 4100/9300 섀시에 적용됩니다. 가져오는 동안 Breakout 포트 구성이 변경되는 경우 Firepower 4100/9300 섀시를 다시 시작해야 합니다.

단계 **3** 전에 구성된 원격 서버로부터 구성 파일을 가져오려면:

a) Remote Import(원격 가져오기) 테이블에서, 사용할 원격 구성에 대해 **Import(**가져오기**)**를 클릭합니다.
확인 대화 상자가 열리면서 계속 진행할 것인지를 물어보고 섀시를 재시작해야 한다고 경고합니다.

b) **Yes(**예**)**를 클릭하여 지정된 구성 파일을 가져올 것임을 확인합니다.
기존의 구성이 삭제되고, 가져오기 파일에 지정된 구성이 Firepower 4100/9300 섀시에 적용됩니다. 가져오는 동안 Breakout 포트 구성이 변경되는 경우 Firepower 4100/9300 섀시를 다시 시작해야 합니다.

단계 **4** 새로운 원격 서버에 있는 구성 파일로부터 가져오려면:

a) Remote Import(원격 가져오기) 아래에서 **Add Remote Configuration(**원격 구성 추가**)**을 클릭합니다.

b) 원격 서버와의 통신에서 사용할 프로토콜을 선택합니다. FTP, TFTP, SCP, SFTP 중 하나일 수 있습니다.

c) 기본값 이외의 포트를 사용하려는 경우 **Port(**포트**)** 필드에 포트 번호를 입력합니다.

d) 백업 파일을 저장할 위치의 IP 주소 또는 호스트 이름을 입력합니다. 이는 Firepower 4100/9300 섀시가 네트워크를 통해 액세스할 수 있는 서버, 스토리지 어레이, 로컬 드라이브, 기타 읽기/쓰기 미디어일 수 있습니다.

IP 주소가 아니라 호스트 이름을 사용하는 경우 DNS 서버를 구성해야 합니다.

e) 시스템이 원격 서버에 로그인할 때 사용할 사용자 이름을 입력합니다. 프로토콜이 TFTP일 경우 이 필드는 적용되지 않습니다.

f) 원격 서버 사용자 이름의 비밀번호를 입력합니다. 프로토콜이 TFTP일 경우 이 필드는 적용되지 않습니다.

g) **File Path(**파일 경로**)** 필드에 설정 파일의 전체 경로(파일 이름 포함)를 입력합니다.

h) **Save(**저장**)**를 클릭합니다.

Remote Import(원격 가져오기) 테이블에 원격 구성이 추가됩니다.

i) 사용할 원격 구성에 대해 **Import(**가져오기**)**를 클릭합니다.
확인 대화 상자가 열리면서 계속 진행할 것인지를 물어보고 섀시를 재시작해야 한다고 경고합니다.

j) **Yes(**예**)**를 클릭하여 지정된 구성 파일을 가져올 것임을 확인합니다.
기존의 구성이 삭제되고, 가져오기 파일에 지정된 구성이 Firepower 4100/9300 섀시에 적용됩니다. 가져오는 동안 Breakout 포트 구성이 변경되는 경우 Firepower 4100/9300 섀시를 다시 시작해야 합니다.

13 장

문제 해결

패킷 캡처

패킷 캡처는 연결 및 구성 문제를 디버깅하고 Firepower 4100/9300 섀시를 통과하는 트래픽 흐름을 파악하기 위해 사용할 수 있는 매우 유용한 자산입니다. 패킷 캡처 도구를 사용하면 Firepower 4100/9300 섀시의 특정 인터페이스를 통과하는 트래픽을 로깅할 수 있습니다.

여러 패킷 캡처 세션을 생성할 수 있으며, 각 세션은 여러 인터페이스의 트래픽을 캡처할 수 있습니다. 패킷 캡처 세션에 포함된 각 인터페이스에 대해 별도의 패킷 캡처(PCAP) 파일이 생성됩니다.

백플레인 포트 매핑

Firepower 4100/9300 섀시는 내부 백플레인 포트에 다음 매핑을 사용합니다.

보안 모듈	포트 매핑	설명
보안 모듈 1/검색 엔진	Ethernet1/9	Internal-Data0/0
보안 모듈 1/검색 엔진	Ethernet1/10	Internal-Data1/0

보안 모듈	포트 매핑	설명
보안 모듈 2	Ethernet1/11	Internal-Data0/0
보안 모듈 2	Ethernet1/12	Internal-Data1/0
보안 모듈 3	Ethernet1/13	Internal-Data0/0
보안 모듈 3	Ethernet1/14	Internal-Data1/0

패킷 캡처 관련 지침 및 제한 사항

패킷 캡처 도구의 제한 사항은 다음과 같습니다.

- 최대 100Mbps까지만 캡처할 수 있습니다.
- 패킷 캡처 세션을 실행하기 위해 사용할 저장 공간이 충분하지 않을 경우에도 패킷 캡처 세션을 만들 수 있습니다. 패킷 캡처 세션을 시작하기 전에 저장 공간이 충분한지 확인해야 합니다.
- 싱글 와이드 4x100Gbps 또는 2x100Gbps 네트워크 모듈(각각 부품 번호 FPR-NM-4X100G 및 FPR-NM-2X100G)의 패킷 캡처 세션에서, 모듈 `adminstate`가 `off`로 설정된 경우 캡처 세션은 "Oper State Reason(상태 이유): Unknown Error(알 수 없는 오류)."로 자동으로 비활성화됩니다. 모듈 `adminstate`를 `on`으로 다시 설정한 후 캡처 세션을 다시 시작해야 합니다.

  다른 모든 네트워크 모듈을 사용하는 경우 패킷 캡처 세션은 모듈 `adminstate` 상태가 변경되는 동안 계속됩니다.
- 여러 활성 패킷 캡처 세션은 지원되지 않습니다.
- 내부 스위치의 인그레스 단계에서만 캡처합니다.
- 내부 스위치에서 이해할 수 없는 패킷(Security Group Tag 및 Network Service Header 패킷)에는 필터가 효과적이지 않습니다.
- 상위 인터페이스 하나 이상에 하위 인터페이스가 여러 개 있더라도 세션당 하위 인터페이스 하나에 대해서만 패킷을 캡처할 수 있습니다.
- EtherChannel 전체나 EtherChannel의 하위 인터페이스에 대해 패킷을 캡처할 수는 없습니다. 그러나 논리적 디바이스에 할당된 EtherChannel의 경우에는 EtherChannel의 각 멤버 인터페이스에서 패킷을 캡처할 수 있습니다. 하위 인터페이스는 할당하고 상위 인터페이스는 할당하지 않는 경우에는 멤버 인터페이스에서 패킷을 캡처할 수 없습니다.
- 캡처 세션이 활성 상태인 동안에는 PCAP 파일을 복사하거나 내보낼 수 없습니다.
- 패킷 캡처 세션을 삭제하면 해당 세션과 연결된 모든 패킷 캡처 파일도 삭제됩니다.

패킷 캡처 세션 생성 또는 수정

프로시저

단계 **1** **Tools(**도구**)** > **Packet Capture(**패킷 캡처**)**를 선택합니다.

Capture Session(캡처 세션**)** 탭에 현재 구성된 패킷 캡처 세션 목록이 표시됩니다. 현재 구성된 패킷 캡처 세션이 없는 경우 그러한 내용의 메시지가 대신 표시됩니다.

단계 **2** 다음 중 하나를 수행합니다.

- 패킷 캡처 세션을 만들려면 **Capture Session(**캡처 세션**)** 버튼을 클릭합니다.
- 기존 패킷 캡처 세션을 수정하려면 해당 세션의 **Edit(**수정**)** 버튼을 클릭합니다.

창의 왼쪽에서 특정 애플리케이션 인스턴스를 선택하면 해당 인스턴스가 표시됩니다. 이 표시는 패킷을 캡처할 인터페이스를 선택하는 데 사용됩니다. 창 오른쪽에는 패킷 캡처 세션을 정의하기 위한 필드가 있습니다.

단계 **3** 드롭다운 메뉴에서 인스턴스를 선택합니다.

단계 **4** 트래픽을 캡처할 인터페이스를 클릭합니다. 선택한 인터페이스에는 확인 표시가 나타납니다.

단계 **5** 하위 인터페이스의 경우 상위 인터페이스 왼쪽의 아이콘을 클릭하여 **Subinterface selection(**하위 인터페이스 선택**)** 열에서 하위 인터페이스를 확인합니다. 해당 열에서 하위 인터페이스 하나를 클릭합니다. 상위 인터페이스 하나 이상에 하위 인터페이스가 여러 개 있더라도 캡처 세션당 하위 인터페이스 하나에 대해서만 패킷을 캡처할 수 있습니다.

하위 인터페이스가 여러 개인 경우 아이콘에는 **Subinterfaces(n)(**하위 인터페이스**(n))** 레이블이 표시됩니다. 단일 하위 인터페이스에는 하위 인터페이스 ID가 레이블로 표시됩니다. 상위 인터페이스도 인스턴스에 할당되는 경우 상위 인터페이스나 하위 인터페이스 중 하나를 선택할 수 있으며 둘 다 선택할 수는 없습니다. 할당되지 않은 상위 인터페이스는 흐리게 표시됩니다. EtherChannel에 대한 하위 인터페이스는 지원되지 않습니다.

단계 **6** 백플레인 포트를 통해 나가는 논리적 디바이스에서 트래픽을 캡처하려면:

a) 애플리케이션 인스턴스를 나타내는 상자를 클릭합니다.

Configure Packet Capture Session(패킷 캡처 세션 구성**)** 창의 오른쪽에서 **Capture On(**캡처**)**, **Application Port(**애플리케이션 포트**)** 및 **Application Capture Direction(**애플리케이션 캡처 방향**)** 필드를 사용할 수 있습니다.

b) 트래픽을 캡처할 백플레인 포트를 선택하거나, **Capture On** 드롭다운 목록에서 **All Backplane Ports(**모든 백플레인 포트**)**를 선택합니다.

단계 **7** **Session Name(**세션 이름**)** 필드에 패킷 캡처 세션에 대한 이름을 입력합니다.

단계 **8** **Buffer Size(**버퍼 크기**)** 목록에서 미리 정의된 값 중 하나를 선택하거나 **Custom in MB(MB**의 커스텀**)**를 선택하고 원하는 버퍼 크기를 입력하여 이 패킷 캡처 세션을 사용하기 위한 버퍼 크기를 지정합니다. 1~2048MB 범위에서 버퍼 크기를 지정해야 합니다.

단계 **9** 캡처할 패킷의 길이를 **Snap Length(**스냅 길이**)** 필드에 지정합니다. 유효한 값은 64~9006바이트입니다. 기본 스냅 길이는 1518바이트입니다.

단계 **10** 이 패킷 캡처 세션이 실행될 때 기존 PCAP 파일을 덮어쓸지, 아니면 데이터를 PCAP 파일에 첨부할지를 지정합니다.

단계 **11** 애플리케이션 인스턴스와 특정 인터페이스 간 트래픽을 캡처하려면 다음을 수행합니다.

a) 논리적 디바이스를 나타내는 확인란을 클릭합니다.

b) **Capture On(**캡처 대상**)** 드롭다운 목록에서 애플리케이션 유형(예: **asa**)을 선택합니다.

c) 수신 또는 전송 트래픽을 캡처할 **Application Port(**애플리케이션 포트**)**를 선택합니다.

d) 논리적 디바이스로부터 지정된 인터페이스로 이동하는 트래픽만 캡처하려면 **Application Capture Direction(**애플리케이션 캡처 방향**)** 옆에 있는 **Egress Packets(**이그레스 패킷**)** 옵션을 클릭합니다.

참고 **Egress Packets(**이그레스 패킷**)**을 선택하면 선택한 백플레인 포트에서만 트래픽이 캡처됩니다. 물리적 포트를 선택한 경우에도 트래픽이 캡처되지 않습니다.

e) 지정된 인터페이스에서 들어오고 나가는 트래픽을 캡처하려면 **Application Capture Direction(**애플리케이션 캡처 방향**)** 옆에 있는 **All Packets(**모든 패킷**)** 옵션을 클릭합니다.

단계 **12** 캡처되는 트래픽을 필터링하려면:

a) **Capture Filter(**캡처 필터**)** 필드에서 **Apply Filter(**필터 적용**)** 옵션을 클릭합니다.

필터를 구성하기 위한 필드 집합이 표시됩니다.

b) 필터를 만들어야 하는 경우 **Create Filter(**필터 생성**)**를 클릭합니다.

Create Packet Filter(패킷 필터 생성**)** 대화 상자가 나타납니다. 자세한 내용은 패킷 캡처에 대한 필터 구성, 317 페이지를 참고하십시오.

c) **Apply(**적용**)** 드롭다운 목록에서 사용할 필터를 선택합니다.

d) **To(**대상**)** 드롭다운 목록에서 필터를 적용할 인터페이스를 선택합니다.

e) 추가 필터를 적용하려면 **Apply Another Filter(**다른 필터 적용**)**를 클릭하고 위의 단계를 반복하여 추가 필터를 적용합니다.

단계 **13** 다음 중 하나를 수행합니다.

- 이 패킷 캡처 세션을 저장하고 지금 실행하려면 **Save and Run(**저장 및 실행**)** 버튼을 클릭합니다. 이 옵션은 현재 실행 중인 다른 패킷 캡처 세션이 없는 경우에만 사용할 수 있습니다.
- 나중에 실행할 수 있도록 이 패킷 캡처 세션을 저장하려면 **Save(**저장**)** 버튼을 클릭합니다.

생성된 다른 세션과 함께 해당 세션이 **Capture Session(**캡처 세션**)** 탭에 나열됩니다. **Save and Run(**저장 및 실행**)**을 선택한 경우 패킷 캡처 세션이 패킷을 캡처합니다. 세션에서 PCAP 파일을 다운로드하려면 먼저 캡처를 중지해야 합니다.

패킷 캡처에 대한 필터 구성

패킷 캡처 세션에 포함된 트래픽을 제한할 필터를 만들 수 있습니다. 패킷 캡처 세션을 생성하는 동안 특정 필터를 사용해야 하는 인터페이스를 선택할 수 있습니다.

참고 현재 실행 중인 패킷 캡처 세션에 적용되는 필터를 수정하거나 삭제하는 경우, 해당 세션을 비활성화한 후 다시 활성화해야 변경 내용이 적용됩니다.

프로시저

단계 **1** **Tools(**도구**)** > **Packet Capture(**패킷 캡처**)**를 선택합니다.

Capture Session(캡처 세션**)** 탭에 현재 구성된 패킷 캡처 세션 목록이 표시됩니다. 현재 구성된 패킷 캡처 세션이 없는 경우 그러한 내용의 메시지가 대신 표시됩니다.

단계 **2** 다음 중 하나를 수행합니다.

- 필터를 생성하려면 **Add Filter(**필터 추가**)** 버튼을 클릭합니다.
- 기존 필터를 수정하려면 해당 필터의 **Edit(**수정**)** 버튼을 클릭합니다.

Create or Edit Packet Filter(패킷 필터 생성 또는 수정**)** 대화 상자가 나타납니다.

단계 **3** **Filter Name(**필터 이름**)** 필드에 패킷 캡처 필터에 대한 이름을 입력합니다.

단계 **4** 특정 프로토콜을 필터링하려면 **Protocol(**프로토콜**)** 목록에서 선택하거나, **Custom(**커스텀**)**을 선택한 다음 원하는 프로토콜을 입력합니다. 커스텀 프로토콜은 10진수 형식의 IANA 정의 프로토콜이어야 합니다(0-255).

단계 **5** 특정 EtherType을 필터링하려면 **EtherType** 목록에서 선택하거나, **Custom(**커스텀**)**을 선택한 다음 원하는 EtherType을 입력합니다. 커스텀 EhterType은 10진수 형식의 IANA 정의 EtherType이어야 합니다(예: IPv4 = 2048, IPv6 = 34525, ARP = 2054, SGT = 35081).

단계 **6** Inner VLAN(포트로 들어가는 동안의 VLAN ID) 또는 Outer VLAN(Firepower 4100/9300 섀시에 의해 추가된 VLAN ID)을 기반으로 트래픽을 필터링하려면 지정된 필드에 VLAN ID를 입력합니다.

단계 **7** 특정 소스 또는 목적지의 트래픽을 필터링하려면 지정된 소스 또는 목적지 필드에 IP 주소와 포트를 입력하거나 MAC 주소를 입력합니다.

참고 IPv4 또는 IPv6 주소를 사용하여 필터링할 수 있지만, 동일한 패킷 캡처 세션에서 두 주소를 모두 필터링할 수는 없습니다.

단계 **8** 필터를 저장하려면 **Save(**저장**)**를 클릭합니다.

생성된 다른 필터와 함께 해당 필터가 **Filter List(**필터 목록**)** 탭에 나열됩니다.

패킷 캡처 세션 시작 및 중지

프로시저

단계 **1** **Tools(**도구**)** > **Packet Capture(**패킷 캡처**)**를 선택합니다.

Capture Session(캡처 세션**)** 탭에 현재 구성된 패킷 캡처 세션 목록이 표시됩니다. 현재 구성된 패킷 캡처 세션이 없는 경우 그러한 내용의 메시지가 대신 표시됩니다.

단계 **2** 패킷 캡처 세션을 시작하려면 해당 세션에 대해 **Enable Session(**세션 활성화**)** 버튼을 클릭한 다음 **Yes(**예**)**를 클릭하여 확인합니다.

참고 다른 세션이 실행 중인 동안에는 패킷 캡처 세션을 시작할 수 없습니다.

세션에 포함된 인터페이스에 대한 PCAP 파일이 트래픽 수집을 시작합니다. 세션 데이터를 덮어쓰도록 세션을 구성한 경우 기존 PCAP 데이터가 지워집니다. 아닌 경우 데이터가 기존 파일(있는 경우)에 추가됩니다.

패킷 캡처 세션이 실행 중인 동안에는 트래픽이 캡처될 때 개별 PCAP 파일의 크기가 증가합니다. 버퍼 크기 제한에 도달하면 시스템이 패킷 삭제를 시작하고 Drop Count(삭제 수) 필드가 증가합니다.

단계 **3** 패킷 캡처 세션을 중지하려면 해당 세션에 대해 **Disable Session(**세션 비활성화**)** 버튼을 클릭한 다음 **Yes(**예**)**를 클릭하여 확인합니다.

세션이 비활성화된 후 PCAP 파일을 다운로드할 수 있습니다(패킷 캡처 파일 다운로드, 318 페이지 참조).

패킷 캡처 파일 다운로드

네트워크 패킷 분석기를 사용하여 분석할 수 있도록 세션에서 로컬 컴퓨터로 PCAP(Packet Capture) 파일을 다운로드할 수 있습니다.

프로시저

단계 **1** **Tools(**도구**)** > **Packet Capture(**패킷 캡처**)**를 선택합니다.

Capture Session(캡처 세션**)** 탭에 현재 구성된 패킷 캡처 세션 목록이 표시됩니다. 현재 구성된 패킷 캡처 세션이 없는 경우 그러한 내용의 메시지가 대신 표시됩니다.

단계 **2** 패킷 캡처 세션에서 특정 인터페이스에 대한 PCAP 파일을 다운로드하려면 인터페이스에 해당하는 **Download(**다운로드**)** 버튼을 클릭합니다.

참고 패킷 캡처 세션이 실행 중인 동안에는 PCAP 파일을 다운로드할 수 없습니다.

브라우저에 따라, 지정된 PCAP 파일이 기본 다운로드 위치에 자동으로 다운로드되거나 파일을 저장하라는 프롬프트가 표시됩니다.

패킷 캡처 세션 삭제

현재 실행하고 있지 않은 개별 패킷 캡처 세션을 삭제하거나, 모든 비활성 패킷 캡처 세션을 삭제할 수 있습니다.

프로시저

단계 **1** **Tools(**도구**)** > **Packet Capture(**패킷 캡처**)**를 선택합니다.

Capture Session(캡처 세션**)** 탭에 현재 구성된 패킷 캡처 세션 목록이 표시됩니다. 현재 구성된 패킷 캡처 세션이 없는 경우 그러한 내용의 메시지가 대신 표시됩니다.

단계 **2** 특정 패킷 캡처 세션을 삭제하려면 세션에 해당하는 **Delete(**삭제**)** 버튼을 클릭합니다.

단계 **3** 모든 비활성 패킷 캡처 세션을 삭제하려면 패킷 캡처 세션 목록 위에 있는 **Delete All Sessions(**모든 세션 삭제**)** 버튼을 클릭합니다.

네트워크 연결성 테스트

시작하기 전에

호스트 이름 또는 IPv4 주소가 있는 네트워크에서 다른 디바이스를 ping하여 기본 네트워크 연결을 테스트하려면 **ping** 명령을 사용합니다. 호스트 이름 또는 IPv6 주소가 있는 네트워크에서 다른 디바이스를 ping하려면 **ping6** 명령을 사용합니다.

호스트 이름 또는 IPv4 주소를 사용하는 네트워크에서 다른 디바이스에 대한 경로를 추적하려면 **traceroute** 명령을 사용합니다. 호스트 이름 또는 IPv6 주소를 사용하는 네트워크에서 다른 디바이스에 대한 경로를 추적하려면 **traceroute6** 명령을 사용합니다.

- **ping** 및 **ping6** 명령은 `local-mgmt` 모드에서 사용할 수 있습니다.
- **ping** 명령은 `module` 모드에서도 사용할 수 있습니다.
- **traceroute** 및 **traceroute6** 명령은 `local-mgmt` 모드에서 사용할 수 있습니다.
- **traceroute** 명령은 `module` 모드에서도 사용할 수 있습니다.

프로시저

단계 **1** 다음 명령 중 하나를 입력하여 `local-mgmt` 또는 `module` 모드에 연결합니다.

- **connect local-mgmt**
- **connect module** *module-ID* {**console** | **telnet**}

예제:

```
FP9300-A# connect local-mgmt
FP9300-A(local-mgmt)#
```

단계 **2** 호스트 이름 또는 IPv4 주소를 사용하는 네트워크에서 다른 디바이스를 ping하여 기본 네트워크 연결을 테스트합니다.

ping {*hostname* | *IPv4_address*} [**count** *number_packets*] | [**deadline** *seconds*] | [**interval** *seconds*] | [**packet-size** *bytes*]

예제:

이 예에서는 네트워크에 있는 다른 디바이스를 12번 ping하여 연결하는 방법을 보여 줍니다.

```
FP9300-A(local-mgmt)# ping 198.51.100.10 count 12
PING 198.51.100.10 (198.51.100.10) from 203.0.113.5 eth0: 56(84) bytes of data.
64 bytes from 198.51.100.10: icmp_seq=1 ttl=61 time=0.264 ms
64 bytes from 198.51.100.10: icmp_seq=2 ttl=61 time=0.219 ms
64 bytes from 198.51.100.10: icmp_seq=3 ttl=61 time=0.234 ms
64 bytes from 198.51.100.10: icmp_seq=4 ttl=61 time=0.205 ms
64 bytes from 198.51.100.10: icmp_seq=5 ttl=61 time=0.216 ms
64 bytes from 198.51.100.10: icmp_seq=6 ttl=61 time=0.251 ms
64 bytes from 198.51.100.10: icmp_seq=7 ttl=61 time=0.223 ms
64 bytes from 198.51.100.10: icmp_seq=8 ttl=61 time=0.221 ms
64 bytes from 198.51.100.10: icmp_seq=9 ttl=61 time=0.227 ms
64 bytes from 198.51.100.10: icmp_seq=10 ttl=61 time=0.224 ms
64 bytes from 198.51.100.10: icmp_seq=11 ttl=61 time=0.261 ms
64 bytes from 198.51.100.10: icmp_seq=12 ttl=61 time=0.261 ms

--- 198.51.100.10 ping statistics ---
12 packets transmitted, 12 received, 0% packet loss, time 11104ms
rtt min/avg/max/mdev = 51.005/51.062/51.164/0.064 ms

FP9300-A(local-mgmt)#
```

단계 **3** 호스트 이름 또는 IPv4 주소를 사용하는 네트워크에서 다른 디바이스에 대한 경로를 추적하려면 다음을 수행합니다.

traceroute {*hostname* | *IPv4_address*}

예제:

```
FP9300-A(local-mgmt)# traceroute 198.51.100.10
traceroute to 198.51.100.10 (198.51.100.10), 30 hops max, 40 byte packets
 1  198.51.100.57 (198.51.100.57)  0.640 ms  0.737 ms  0.686 ms
 2  net1-gw1-13.cisco.com (198.51.100.101)  2.050 ms  2.038 ms  2.028 ms
 3  net1-sec-gw2.cisco.com (198.51.100.201)  0.540 ms  0.591 ms  0.577 ms
 4  net1-fp9300-19.cisco.com (198.51.100.108)  0.336 ms  0.267 ms  0.289 ms
```

```
FP9300-A(local-mgmt)#
```

단계 **4** (선택 사항) **exit**를 입력하여 `local-mgmt` 모드를 종료하고 최상위 레벨 모드로 돌아갑니다.

관리 인터페이스 상태 트러블슈팅

초기화 및 구성 중에 관리 인터페이스가 작동하지 않는 것으로 의심되는 경우(예: 섀시 관리자에 액세스할 수 없음) `local-mgmt` 셸의 **show mgmt-port** 명령을 사용하여 관리 인터페이스의 상태를 확인합니다.

참고 `fxos` 셸에서는 현재 잘못된 정보를 표시하므로 **show interface brief**명령을 사용하지 마십시오.

프로시저

단계 **1** 다음 명령 중 하나를 입력하여 `local-mgmt` 모드에 연결합니다.

- **connect local-mgmt**

예제:

```
firepower# connect local-mgmt
firepower(local-mgmt)#
```

단계 **2** **show mgmt-port** 명령을 사용하여 관리 인터페이스의 상태를 확인합니다.

예제:

```
firepower(local-mgmt)# show mgmt-port
eth0      Link encap:Ethernet  HWaddr b0:aa:77:2f:f0:a9
          inet addr:10.89.5.14  Bcast:10.89.5.63  Mask:255.255.255.192
          inet6 addr: fe80::b2aa:77ff:fe2f:f0a9/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:3210912 errors:0 dropped:0 overruns:0 frame:0
          TX packets:705434 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1648941394 (1.5 GiB)  TX bytes:138386379 (131.9 MiB)

firepower(local-mgmt)#
```

show mgmt-ip-debug 명령을 사용할 수도 있습니다. 그러나 인터페이스 구성 정보의 광범위 한 목록을 생성 합니다.

포트 채널 상태 확인

다음 단계를 수행하여 현재 정의된 포트 채널의 상태를 확인할 수 있습니다.

프로시저

단계 **1** 다음 명령을 입력하여 `/eth-uplink/fabric` 모드를 시작합니다.

- **scope eth-uplink**
- **scope fabric** {**a** | **b**}

예제:

```
FP9300-A# scope eth-uplink
FP9300-A /eth-uplink # scope fabric a
FP9300-A /eth-uplink/fabric #
```

단계 **2** **show port-channel** 명령을 입력하여 각각의 관리 상태 및 작동 상태와 함께 현재 포트 채널 목록을 표시합니다.

예제:

```
FP9300-A /eth-uplink/fabric # show port-channel

Port Channel:
    Port Channel Id Name             Port Type          Admin
 State Oper State       State Reason
    --------------- ---------------- ------------------ -----
------ ---------------- ------------
    10              Port-channel10   Data               Enabl
ed     Failed           No operational members
    11              Port-channel11   Data               Enabl
ed     Failed           No operational members
    12              Port-channel12   Data               Disab
led    Admin Down       Administratively down
    48              Port-channel48   Cluster            Enabl
ed     Up

FP9300-A /eth-uplink/fabric #
```

단계 **3** 다음 명령을 입력하여 `/port-channel` 모드를 시작하고 개별 포트 채널 및 포트 정보를 표시합니다.

- **scope port-channel** *ID*

예제:

```
FP9300-A /eth-uplink/fabric/port-channel # top
FP9300-A# connect fxos
Cisco Firepower Extensible Operating System (FX-OS) Software
TAC support: http://www.cisco.com/tac
Copyright (c) 2002-2017, Cisco Systems, Inc. All rights reserved.

The copyrights to certain works contained in this software are
owned by other third parties and used and distributed under
```

```
license.

    <--- remaining lines removed for brevity --->

FP9300-A(fxos)#
```

단계 **4** **show** 명령을 입력하여 지정된 포트 채널에 대한 상태 정보를 표시합니다.

예제:

```
FP9300-A /eth-uplink/fabric/port-channel # show

Port Channel:
    Port Channel Id Name             Port Type          Admin
 State Oper State       State Reason
    ---------------- ---------------- ------------------ -----
------ ---------------- ------------
    10              Port-channel10   Data               Enabl
ed     Failed           No operational members

FP9300-A /eth-uplink/fabric/port-channel #
```

단계 **5** **show member-port** 명령을 입력하여 포트 채널의 멤버 포트에 대한 상태 정보를 표시합니다.

예제:

```
FP9300-A /eth-uplink/fabric/port-channel # show member-port

Member Port:
    Port Name       Membership         Oper State       State Reas
on
    --------------- ------------------ ---------------- ----------
--
    Ethernet2/3     Suspended          Failed           Suspended
    Ethernet2/4     Suspended          Failed           Suspended

FP9300-A /eth-uplink/fabric/port-channel #
```

포트 채널은 논리적 디바이스에 할당될 때까지 나타나지 않습니다. 포트 채널을 논리적 디바이스에서 제거하거나 논리적 디바이스가 삭제된 경우, 포트 채널은 일시 중단 상태로 전환됩니다.

단계 **6** 추가 포트 채널 및 LACP 정보를 보려면 `/eth-uplink/fabric/port-channel` 모드를 종료하고 다음 명령을 입력하여 `fxos` 모드를 시작합니다.

- **top**
- **connect fxos**

예제:

단계 **7** **show port-channel summary** 명령을 입력하여 현재 포트 채널에 대한 요약 정보를 표시합니다.

예제:

```
FP9300-A(fxos)# show port-channel summary
Flags:  D - Down        P - Up in port-channel (members)
        I - Individual  H - Hot-standby (LACP only)
        s - Suspended   r - Module-removed
        S - Switched    R - Routed
        U - Up (port-channel)
```

```
        M - Not in use. Min-links not met
--------------------------------------------------------------
------------------
Group Port-       Type     Protocol  Member Ports
      Channel
--------------------------------------------------------------
------------------
10    Po10(SD)    Eth      LACP      Eth2/3(s)    Eth2/4(s)

11    Po11(SD)    Eth      LACP      Eth2/1(s)    Eth2/2(s)

12    Po12(SD)    Eth      LACP      Eth1/4(D)    Eth1/5(D)

48    Po48(SU)    Eth      LACP      Eth1/1(P)    Eth1/2(P)
```

추가 **show port-channel** 및 **show lacp** 명령은 `fxos` 모드에서 사용할 수 있습니다. 이러한 명령은 다양한 포트 채널 및 용량, 트래픽, 카운터, 사용량 등의 LACP 정보를 표시하는 데 사용할 수 있습니다.

다음에 수행할 작업

포트 채널 생성 관련 정보는 EtherChannel(포트 채널) 추가, 185 페이지의 내용을 참조하십시오.

소프트웨어 장애에서 복구

시작하기 전에

시스템의 성공적인 부팅을 방해하는 소프트웨어 장애가 발생하면 다음 절차에 따라 소프트웨어의 새 버전을 부팅할 수 있습니다. 이 프로세스를 완료하려면 킥스타트 이미지를 TFTP 부팅하고, 새 시스템과 관리자 이미지를 다운로드하고, 새 이미지를 사용하여 부팅해야 합니다.

Cisco.com의 다음 위치에서 특정 FXOS 버전에 대한 복구 이미지를 가져올 수 있습니다.

- Firepower 9300—https://software.cisco.com/portal/pub/download/portal/select.html?&mdfid=286287252&flowid=77282&softwareid=286287263
- Firepower 4100 Series—https://software.cisco.com/portal/pub/download/portal/select.html?&mdfid=286305187&flowid=79423&softwareid=286287263

복구 이미지에는 세 개의 별도 파일이 포함되어 있습니다. 예를 들어 다음은 FXOS 2.1.1.64의 현재 복구 이미지입니다.

```
Recovery image (kickstart) for FX-OS 2.1.1.64.
fxos-k9-kickstart.5.0.3.N2.4.11.63.SPA

Recovery image (manager) for FX-OS 2.1.1.64.
fxos-k9-manager.4.1.1.63.SPA

Recovery image (system) for FX-OS 2.1.1.64.
fxos-k9-system.5.0.3.N2.4.11.63.SPA
```

프로시저

단계 **1** ROMMON에 액세스합니다.

a) 콘솔 포트에 연결합니다.

b) 시스템을 재부팅합니다.

시스템이 로딩을 시작하며, 로딩 프로세스 중에 카운트다운 타이머가 표시됩니다.

c) 카운트다운 중에 **Escape** 키를 눌러 ROMMON 모드로 들어갑니다.

예제:

```
Cisco System ROMMON, version 1.0.09, RELEASE SOFTWARE
Copyright (c) 1994-2015 by Cisco Systems, Inc.
Compiled Sun 01/01/1999 23:59:59:59.99 by user

Current image running: Boot ROM0
Last reset cause: LocalSoft
DIMM Slot 0 : Present
DIMM Slot 1 : Present
No USB drive !!

Platform FPR9K-SUP with 16384 Mbytes of main memory
MAC Address aa:aa:aa:aa:aa:aa

find the string ! boot bootflash:/installables/switch/fxos-k9-kickstart.5.0.3.N2.0.00.00.SPA

  bootflash:/installables/switch/fxos-k9-system.5.0.3.N2.0.00.00.SPA

Use BREAK, ESC or CTRL+L to interrupt boot.
use SPACE to begin boot immediately.
Boot interrupted.

rommon 1  >
```

단계 **2** 킥스타트 이미지를 TFTP 부팅합니다.

a) 관리 IP 주소, 관리 넷마스크, 게이트웨이 IP 주소가 올바르게 설정되었는지 확인합니다. **set** 명령을 사용하여 해당 값을 볼 수 있습니다. **ping** 명령을 사용하여 TFTP 서버에 대한 연결을 테스트할 수 있습니다.

```
rommon 1 > set
    ADDRESS=
    NETMASK=
    GATEWAY=
    SERVER=
    IMAGE=
    PS1="ROMMON ! > "
rommon > address <ip-address>
rommon > netmask <network-mask>
rommon > gateway <default-gateway>
```

b) 킥스타트 이미지를 Firepower 4100/9300 섀시에서 액세스 가능한 TFTP 디렉터리에 복사합니다.

참고 킥스타트 이미지 버전 번호는 번들 버전 번호와 일치하지 않습니다. Cisco.com 소프트웨어 다운로드 페이지에서 FXOS 버전과 킥스타트 이미지 간 매핑을 보여주는 정보를 찾을 수 있습니다.

c) boot 명령을 사용하여 ROMMON에서 이미지를 부팅합니다.

```
boot tftp://<IP address>/<path to image>
```

참고 Firepower 4100/9300 섀시의 전면 패널에 있는 USB 슬롯에 삽입한 USB 미디어 디바이스를 사용하여 ROMMON에서 킥스타트를 부팅할 수도 있습니다. 시스템이 실행 중일 때 USB 디바이스를 삽입하는 경우 시스템을 리부팅해야 USB 디바이스가 인식됩니다.

이미지가 수신 중임을 나타내는 일련의 # 표시가 나타난 다음 킥스타트 이미지가 로드됩니다.

예제:

```
rommon 1 > set
    ADDRESS=
    NETMASK=
    GATEWAY=
    SERVER=
    IMAGE=
    PS1="ROMMON ! > "

rommon 2 > address 10.0.0.2
rommon 3 > netmask 255.255.255.0
rommon 4 > gateway 10.0.0.1
rommon 5 > ping 10.0.0.2
..!!!!!!!!!!
Success rate is 100 percent (10/10)
rommon 6 > ping 192.168.1.2
..!!!!!!!!!!
Success rate is 100 percent (10/10)

rommon 7 > boot tftp://192.168.1.2/fxos-k9-kickstart.5.0.3.N2.1.11.1.SPA
             ADDRESS: 10.0.0.2
             NETMASK: 255.255.255.0
             GATEWAY: 10.0.0.1
              SERVER: 192.168.1.2
               IMAGE: fxos-k9-kickstart.5.0.3.N2.1.11.1.SPA

        TFTP_MACADDR: aa:aa:aa:aa:aa:aa
.....................................................................

Receiving fxos-k9-kickstart.5.0.3.N2.1.11.1.SPA from 192.168.1.2

############################################################################
############################################################################
############################################################################

File reception completed.
```

단계 **3** Firepower 4100/9300 섀시에 방금 로드한 킥스타트 이미지와 일치하는 복구 시스템 및 관리자 이미지를 다운로드합니다.

a) 복구 시스템 및 관리자 이미지를 다운로드하려면 관리 IP 주소 및 게이트웨이를 설정해야 합니다. USB를 통해 이러한 이미지를 다운로드할 수 없습니다.

```
switch(boot)# config terminal
switch(boot)(config)# interface mgmt 0
switch(boot)(config-if)# ip address <ip address> <netmask>
switch(boot)(config-if)# no shutdown
switch(boot)(config-if)# exit
switch(boot)(config)# ip default-gateway <gateway>
```

```
switch(boot)(config)# exit
```

b) 원격 서버에서 bootflash로 복구 시스템 및 관리자 이미지를 복사합니다.

switch(boot)# **copy** *URL* **bootflash:**

다음 구문 중 하나를 사용하여 가져올 파일의 URL을 지정합니다.

- **ftp**://*username@hostname*/*path*/*image_name*
- **scp**://*username@hostname*/*path*/*image_name*
- **sftp**://*username@hostname*/*path*/*image_name*
- **tftp**://*hostname*/*path*/*image_name*

예제:

```
switch(boot)# copy
  scp://<username>@192.168.1.2/recovery_images/fxos-k9-system.5.0.3.N2.4.11.69.SPA
  bootflash:

switch(boot)# copy
  scp://<username>@192.168.1.2/recovery_images/fxos-k9-manager.4.1.1.69.SPA
  bootflash:
```

c) 이미지를 성공적으로 Firepower 4100/9300 섀시에 복사한 후 nuova-sim-mgmt-nsg.0.1.0.001.bin에서 관리자 이미지로 symlink를 만듭니다. 이 링크는 로드할 관리자 이미지를 로드 메커니즘에 알려줍니다. 어떤 이미지를 로드하려고 하는지와 상관없이 symlink 이름은 항상 nuova-sim-mgmt-nsg.0.1.0.001.bin이어야 합니다.

```
switch(boot)# copy bootflash:<manager-image>
  bootflash:nuova-sim-mgmt-nsg.0.1.0.001.bin
```

예제:

```
switch(boot)# config terminal
Enter configuration commands, one per line.  End with CNTL/Z.

switch(boot)(config)# interface mgmt 0
switch(boot)(config-if)# ip address 10.0.0.2 255.255.255.0
switch(boot)(config-if)# no shutdown
switch(boot)(config-if)# exit
switch(boot)(config)# ip default-gateway 10.0.0.1
switch(boot)(config)# exit
switch(boot)# copy
  tftp://192.168.1.2/recovery_images/fxos-k9-system.5.0.3.N2.4.11.69.SPA
  bootflash:
Trying to connect to tftp server......
Connection to server Established. Copying Started.....
/
TFTP get operation was successful
Copy complete, now saving to disk (please wait)...

switch(boot)# copy
  tftp://192.168.1.2/recovery_images/fxos-k9-manager.4.1.1.69.SPA
  bootflash:
Trying to connect to tftp server......
```

```
Connection to server Established. Copying Started.....
/
TFTP get operation was successful
Copy complete, now saving to disk (please wait)...

switch(boot)# copy bootflash:fxos-k9-manager.4.1.1.69.SPA
  bootflash:nuova-sim-mgmt-nsg.0.1.0.001.bin

Copy complete, now saving to disk (please wait)...

switch(boot)#
```

단계 **4** 방금 다운로드한 시스템 이미지를 로드합니다.

```
switch(boot)# load bootflash:<system-image>
```

예제:

```
switch(boot)# load bootflash:fxos-k9-system.5.0.3.N2.4.11.69.SPA
Uncompressing system image: bootflash:/fxos-k9-system.5.0.3.N2.4.11.69.SPA


Manager image digital signature verification successful
...
System is coming up ... Please wait ...


Cisco FPR Series Security Appliance
FP9300-A login:
```

단계 **5** 시스템이 이전 이미지를 로드하려고 시도하지 못하게 하려면, 복구 이미지를 로드한 후 다음 명령을 입력합니다.

참고 이 단계는 복구 이미지를 로드한 직후 수행해야 합니다.

```
FP9300-A# scope org
FP9300-A /org # scope fw-platform-pack default
FP9300-A /org/fw-platform-pack # set platform-bundle-version ""
Warning: Set platform version to empty will result software/firmware incompatibility issue.
FP9300-A /org/fw-platform-pack* # commit-buffer
```

단계 **6** Firepower 4100/9300 섀시에서 사용할 플랫폼 번들 이미지를 다운로드 및 설치합니다. 자세한 내용은 이미지 관리, 61 페이지를 참고하십시오.

예제:

```
FP9300-A# scope firmware
FP9300-A /firmware # show download-task

Download task:
    File Name Protocol Server          Port       Userid          State
    --------- -------- --------------- ---------- --------------- -----
    fxos-k9.2.1.1.73.SPA
              Tftp     192.168.1.2           0                  Downloaded
FP9300-A /firmware # show package fxos-k9.2.1.1.73.SPA detail
Firmware Package fxos-k9.2.1.1.73.SPA:
    Version: 2.1(1.73)
    Type: Platform Bundle
    State: Active
```

```
Time Stamp: 2012-01-01T07:40:28.000
Build Date: 2017-02-28 13:51:08 UTC
FP9300-A /firmware #
```

손상된 파일 시스템에서 복구

시작하기 전에

Supervisor의 온보드 플래시가 손상되고 시스템을 더 이상 성공적으로 시작할 수 없는 경우 다음 절차를 사용하여 시스템을 복구할 수 있습니다. 이 프로세스를 완료하려면 킥스타트 이미지를 TFTP 부팅하고, 플래시를 재포맷하고, 새 시스템과 관리자 이미지를 다운로드하고, 새 이미지를 사용하여 부팅해야 합니다.

참고 이 절차에는 시스템 플래시 재포맷이 포함됩니다. 그 결과, 시스템이 복구된 후 완전히 다시 구성해야 합니다.

Cisco.com의 다음 위치에서 특정 FXOS 버전에 대한 복구 이미지를 가져올 수 있습니다.

- Firepower 9300—https://software.cisco.com/portal/pub/download/portal/select.html?&mdfid=286287252&flowid=77282&softwareid=286287263
- Firepower 4100 Series—https://software.cisco.com/portal/pub/download/portal/select.html?&mdfid=286305187&flowid=79423&softwareid=286287263

복구 이미지에는 세 개의 별도 파일이 포함되어 있습니다. 예를 들어 다음은 FXOS 2.1.1.64의 복구 이미지입니다.

```
Recovery image (kickstart) for FX-OS 2.1.1.64.
fxos-k9-kickstart.5.0.3.N2.4.11.63.SPA

Recovery image (manager) for FX-OS 2.1.1.64.
fxos-k9-manager.4.1.1.63.SPA

Recovery image (system) for FX-OS 2.1.1.64.
fxos-k9-system.5.0.3.N2.4.11.63.SPA
```

프로시저

단계 **1** ROMMON에 액세스합니다.

a) 콘솔 포트에 연결합니다.

b) 시스템을 재부팅합니다.

시스템이 로딩을 시작하며, 로딩 프로세스 중에 카운트다운 타이머가 표시됩니다.

c) 카운트다운 중에 **Escape** 키를 눌러 ROMMON 모드로 들어갑니다.

예제:

```
Cisco System ROMMON, version 1.0.09, RELEASE SOFTWARE
Copyright (c) 1994-2015 by Cisco Systems, Inc.
Compiled Sun 01/01/1999 23:59:59:59.99 by user

Current image running: Boot ROM0
Last reset cause: LocalSoft
DIMM Slot 0 : Present
DIMM Slot 1 : Present
No USB drive !!

Platform FPR9K-SUP with 16384 Mbytes of main memory
MAC Address aa:aa:aa:aa:aa:aa

find the string ! boot bootflash:/installables/switch/fxos-k9-kickstart.5.0.3.N2.0.00.00.SPA

  bootflash:/installables/switch/fxos-k9-system.5.0.3.N2.0.00.00.SPA

Use BREAK, ESC or CTRL+L to interrupt boot.
use SPACE to begin boot immediately.
Boot interrupted.

rommon 1  >
```

단계 **2** 킥스타트 이미지를 TFTP 부팅합니다.

a) 관리 IP 주소, 관리 넷마스크, 게이트웨이 IP 주소가 올바르게 설정되었는지 확인합니다. **set** 명령을 사용하여 해당 값을 볼 수 있습니다. **ping** 명령을 사용하여 TFTP 서버에 대한 연결을 테스트할 수 있습니다.

```
rommon 1 > set
    ADDRESS=
    NETMASK=
    GATEWAY=
    SERVER=
    IMAGE=
    PS1="ROMMON ! > "
rommon > address <ip-address>
rommon > netmask <network-mask>
rommon > gateway <default-gateway>
```

b) 킥스타트 이미지를 Firepower 4100/9300 섀시에서 액세스 가능한 TFTP 디렉터리에 복사합니다.

참고 킥스타트 이미지 버전 번호는 번들 버전 번호와 일치하지 않습니다. Cisco.com 소프트웨어 다운로드 페이지에서 FXOS 버전과 킥스타트 이미지 간 매핑을 보여주는 정보를 찾을 수 있습니다.

c) boot 명령을 사용하여 ROMMON에서 이미지를 부팅합니다.

```
boot tftp://<IP address>/<path to image>
```

참고 Firepower 4100/9300 섀시의 전면 패널에 있는 USB 슬롯에 삽입한 USB 미디어 디바이스를 사용하여 ROMMON에서 킥스타트를 부팅할 수도 있습니다. 시스템이 실행 중일 때 USB 디바이스를 삽입하는 경우 시스템을 리부팅해야 USB 디바이스가 인식됩니다.

이미지가 수신 중임을 나타내는 일련의 # 표시가 나타난 다음 킥스타트 이미지가 로드됩니다.

예제:

```
rommon 1 > set
    ADDRESS=
    NETMASK=
    GATEWAY=
    SERVER=
    IMAGE=
    PS1="ROMMON ! > "

rommon 2 > address 10.0.0.2
rommon 3 > netmask 255.255.255.0
rommon 4 > gateway 10.0.0.1
rommon 5 > ping 10.0.0.2
..!!!!!!!!!!
Success rate is 100 percent (10/10)
rommon 6 > ping 192.168.1.2
..!!!!!!!!!!
Success rate is 100 percent (10/10)

rommon 7 > boot tftp://192.168.1.2/fxos-k9-kickstart.5.0.3.N2.1.11.1.SPA
             ADDRESS: 10.0.0.2
             NETMASK: 255.255.255.0
             GATEWAY: 10.0.0.1
              SERVER: 192.168.1.2
               IMAGE: fxos-k9-kickstart.5.0.3.N2.1.11.1.SPA

        TFTP_MACADDR: aa:aa:aa:aa:aa:aa
.....................................................................

Receiving fxos-k9-kickstart.5.0.3.N2.1.11.1.SPA from 192.168.1.2

############################################################################
############################################################################
############################################################################

File reception completed.
```

단계 **3** 킥스타트 이미지가 로드된 후 **init system** 명령을 사용하여 플래시를 재포맷합니다.

init system 명령은 시스템에 다운로드된 모든 소프트웨어 이미지 및 시스템의 모든 구성을 포함하여 플래시의 콘텐츠를 지웁니다. 이 명령을 완료하는 데 약 20~30분 정도 소요됩니다.

예제:

```
switch(boot)# init system

This command is going to erase your startup-config, licenses as well as the contents of
your bootflash:.

Do you want to continue? (y/n)  [n] y

Detected 32GB flash...
Initializing the system
mke2fs 1.35 (28-Feb-2004)
Checking for bad blocks (read-only test):        done
Initializing startup-config and licenses
mke2fs 1.35 (28-Feb-2004)
Checking for bad blocks (read-only test):        done
mke2fs 1.35 (28-Feb-2004)
Checking for bad blocks (read-only test):        done
mke2fs 1.35 (28-Feb-2004)
Checking for bad blocks (read-only test):        done
```

```
Formatting bootflash:
mke2fs 1.35 (28-Feb-2004)
Checking for bad blocks (read-only test):        done
Formatting SAM partition:
mke2fs 1.35 (28-Feb-2004)
Checking for bad blocks (read-only test):        done
Formatting Workspace partition:
mke2fs 1.35 (28-Feb-2004)
Checking for bad blocks (read-only test):        done
Formatting Sysdebug  partition:
mke2fs 1.35 (28-Feb-2004)
Checking for bad blocks (read-only test):        done
```

단계 **4** 복구 이미지를 Firepower 4100/9300 섀시에 다운로드합니다.

a) 복구 이미지를 다운로드하려면 관리 IP 주소 및 게이트웨이를 설정해야 합니다. USB를 통해 이러한 이미지를 다운로드할 수 없습니다.

```
switch(boot)# config terminal
switch(boot)(config)# interface mgmt 0
switch(boot)(config-if)# ip address <ip address> <netmask>
switch(boot)(config-if)# no shutdown
switch(boot)(config-if)# exit
switch(boot)(config)# ip default-gateway <gateway>
switch(boot)(config)# exit
```

b) 원격 서버에서 bootflash로 복구 이미지 세 개를 모두 복사합니다.

switch(boot)# **copy** *URL* **bootflash:**

다음 구문 중 하나를 사용하여 가져올 파일의 URL을 지정합니다.

- **ftp**://*username@hostname*/*path*/*image_name*
- **scp**://*username@hostname*/*path*/*image_name*
- **sftp**://*username@hostname*/*path*/*image_name*
- **tftp**://*hostname*/*path*/*image_name*

예제**:**

```
switch(boot)# copy
  scp://<username>@192.168.1.2/recovery_images/fxos-k9-kickstart.5.0.3.N2.4.11.69.SPA
  bootflash:

switch(boot)# copy
  scp://<username>@192.168.1.2/recovery_images/fxos-k9-system.5.0.3.N2.4.11.69.SPA
  bootflash:

switch(boot)# copy
  scp://<username>@192.168.1.2/recovery_images/fxos-k9-manager.4.1.1.69.SPA
  bootflash:
```

c) 이미지를 성공적으로 Firepower 4100/9300 섀시에 복사한 후 nuova-sim-mgmt-nsg.0.1.0.001.bin에서 관리자 이미지로 symlink를 만듭니다. 이 링크는 로드할 관리자 이미지를 로드 메커니즘에 알려줍니다. 어떤 이미지를 로드하려고 하는지와 상관없이 symlink 이름은 항상 nuova-sim-mgmt-nsg.0.1.0.001.bin이어야 합니다.

```
switch(boot)# copy bootflash:<manager-image>
  bootflash:nuova-sim-mgmt-nsg.0.1.0.001.bin
```

예제:

```
switch(boot)# config terminal
Enter configuration commands, one per line.  End with CNTL/Z.

switch(boot)(config)# interface mgmt 0
switch(boot)(config-if)# ip address 10.0.0.2 255.255.255.0
switch(boot)(config-if)# no shutdown
switch(boot)(config-if)# exit
switch(boot)(config)# ip default-gateway 10.0.0.1
switch(boot)(config)# exit
switch(boot)# copy
  tftp://192.168.1.2/recovery_images/fxos-k9-kickstart.5.0.3.N2.4.11.69.SPA
  bootflash:
Trying to connect to tftp server......
Connection to server Established. Copying Started.....
/
TFTP get operation was successful
Copy complete, now saving to disk (please wait)...

switch(boot)# copy
  tftp://192.168.1.2/recovery_images/fxos-k9-system.5.0.3.N2.4.11.69.SPA
  bootflash:
Trying to connect to tftp server......
Connection to server Established. Copying Started.....
/
TFTP get operation was successful
Copy complete, now saving to disk (please wait)...

switch(boot)# copy
  tftp://192.168.1.2/recovery_images/fxos-k9-manager.4.1.1.69.SPA
  bootflash:
Trying to connect to tftp server......
Connection to server Established. Copying Started.....
/
TFTP get operation was successful
Copy complete, now saving to disk (please wait)...

switch(boot)# copy bootflash:fxos-k9-manager.4.1.1.69.SPA
  bootflash:nuova-sim-mgmt-nsg.0.1.0.001.bin

Copy complete, now saving to disk (please wait)...

switch(boot)#
```

단계 **5** 스위치를 로드합니다.

```
switch(boot)# reload
```

예제:

```
switch(boot)# reload
This command will reboot this supervisor module. (y/n) ? y
[ 1866.310313] Restarting system.

!!  Rommon image verified successfully  !!
```

```
Cisco System ROMMON, Version 1.0.11, RELEASE SOFTWARE
Copyright (c) 1994-2016  by Cisco Systems, Inc.
Compiled Wed 11/23/2016 11:23:23.47 by builder
Current image running: Boot ROM1
Last reset cause: ResetRequest
DIMM Slot 0 : Present
DIMM Slot 1 : Present
No USB drive !!
BIOS has been locked !!

Platform FPR9K-SUP with 16384 Mbytes of main memory
MAC Address: bb:aa:77:aa:aa:bb

autoboot: Can not find autoboot file 'menu.lst.local'
          Or can not find correct boot string !!
rommon 1 >
```

단계 **6** 킥스타트 및 시스템 이미지에서 부팅합니다.

```
rommon 1 > boot <kickstart-image> <system-image>
```

참고 시스템 이미지가 로드되는 동안 라이선스 관리자 실패 메시지가 표시됩니다. 이러한 메시지는 안전하게 무시할 수 있습니다.

예제:

```
rommon 1 > dir
Directory of: bootflash:\

  01/01/12  12:33a <DIR>          4,096  .
  01/01/12  12:33a <DIR>          4,096  ..
  01/01/12  12:16a <DIR>         16,384  lost+found
  01/01/12  12:27a           34,333,696  fxos-k9-kickstart.5.0.3.N2.4.11.69.SPA
  01/01/12  12:29a          330,646,465  fxos-k9-manager.4.1.1.69.SPA
  01/01/12  12:31a          250,643,172  fxos-k9-system.5.0.3.N2.4.11.69.SPA
  01/01/12  12:34a          330,646,465  nuova-sim-mgmt-nsg.0.1.0.001.bin
          4 File(s) 946,269,798 bytes
          3 Dir(s)

rommon 2 > boot fxos-k9-kickstart.5.0.3.N2.4.11.69.SPA fxos-k9-system.5.0.3.N2.4.11.69.SPA
  !!   Kickstart Image verified successfully   !!

Linux version: 2.6.27.47 (security@cisco.com) #1 SMP Thu Nov 17 18:22:00 PST 2016
[    0.000000] Fastboot Memory at 0c100000 of size 201326592
Usage: init 0123456SsQqAaBbCcUu

INIT: version 2.86 booting

POST INIT Starts at Sun Jan  1 00:27:32 UTC 2012
S10mount-ramfs.supnuovaca Mounting /isan 3000m
Mounted /isan
Creating /callhome..
Mounting /callhome..
Creating /callhome done.
Callhome spool file system init done.
Platform is BS or QP MIO: 30
FPGA Version 0x00010500 FPGA Min Version 0x00000600
Checking all filesystems..r.r..r done.
Warning: switch is starting up with default configuration
Checking NVRAM block device ... done
.
FIPS power-on self-test passed
```

```
Unpack CMC Application software
Loading system software
Uncompressing system image: bootflash:/fxos-k9-system.5.0.3.N2.4.11.69.SPA


Manager image digital signature verification successful

...

System is coming up ... Please wait ...
nohup: appending output to `nohup.out'

           ---- Basic System Configuration Dialog ----

  This setup utility will guide you through the basic configuration of
  the system. Only minimal configuration including IP connectivity to
  the Fabric interconnect and its clustering mode is performed through these steps.

  Type Ctrl-C at any time to abort configuration and reboot system.
  To back track or make modifications to already entered values,
  complete input till end of section and answer no when prompted
  to apply configuration.


  You have chosen to setup a new Security Appliance. Continue? (y/n):
```

단계 **7** 이미지가 로드되면 초기 구성 설정을 입력하라는 프롬프트가 표시됩니다. 자세한 내용은 콘솔 포트를 사용한 초기 구성, 8 페이지를 참고하십시오.

단계 **8** Firepower 4100/9300 섀시에서 사용할 플랫폼 번들 이미지를 다운로드합니다. 자세한 내용은 이미지 관리, 61 페이지를 참조하십시오.

예제:

```
FP9300-A# scope firmware
FP9300-A /firmware # show download-task

Download task:
    File Name Protocol Server          Port       Userid          State
    --------- -------- --------------- ---------- --------------- -----
    fxos-k9.2.1.1.73.SPA
              Tftp     192.168.1.2            0                 Downloaded
FP9300-A /firmware # show package fxos-k9.2.1.1.73.SPA detail
Firmware Package fxos-k9.2.1.1.73.SPA:
    Version: 2.1(1.73)
    Type: Platform Bundle
    State: Active
Time Stamp: 2012-01-01T07:40:28.000
Build Date: 2017-02-28 13:51:08 UTC
FP9300-A /firmware #
```

단계 **9** 이전 단계에서 다운로드한 플랫폼 번들 이미지를 설치합니다.

참고 설치 프로세스는 일반적으로 15~20분 정도 소요됩니다.

a) 자동 설치 모드를 입력합니다.

Firepower-chassis /firmware # **scope auto-install**

b) FXOS 플랫폼 번들을 설치합니다.

Firepower-chassis /firmware/auto-install # **install platform platform-vers** *version_number*

*version_number*는 설치 중인 FXOS 플랫폼 번들의 버전 번호입니다(예: 2.1(1.73)).

c) 시스템은 설치할 소프트웨어 패키지를 먼저 확인합니다. 시스템은 현재 설치된 애플리케이션과 지정된 FXOS 플랫폼 소프트웨어 패키지 간에 비호환성이 있는지 알려줍니다. 또한 기존 세션이 종료되고 시스템이 업그레이드의 일부로 재부팅되어야 한다고 경고합니다.

yes를 입력하여 검증을 계속할 것인지 확인합니다.

d) **yes**를 입력하여 설치를 계속할 것인지 확인하거나 **no**를 입력하여 설치를 취소합니다.

FXOS에서는 번들의 압축을 풀고 구성 요소를 업그레이드하거나 다시 로드합니다.

e) 업그레이드 프로세스를 모니터링하려면 다음과 같이 합니다.

- **scope firmware**을 입력합니다.
- **scope auto-install**을 입력합니다.
- **show fsm status expand**을 입력합니다.

예:

```
TB10 /firmware/auto-install # show fsm status expand
    FSM Status:
        Affected Object: sys/fw-system/fsm
        Current FSM: Deploy
        Status: In Progress
        Completion Time:
        Progress (%): 98

        FSM Stage:
        Order  Stage Name                                Status       Try
        ------ ----------------------------------------- ------------ ---
        1      DeployWaitForDeploy                       Success      0
        2      DeployResolveDistributableNames           Skip         0
        3      DeployResolveDistributable                Skip         0
        4      DeployResolveImages                       Skip         0
        5      DeployValidatePlatformPack                Success      1
        6      DeployDebundlePort                        Success      0
        7      DeployPollDebundlePort                    Success      1
        8      DeployActivateUCSM                        Success      0
        9      DeployPollActivateOfUCSM                  Success      0
        10     DeployActivateMgmtExt                     Skip         0
        11     DeployPollActivateOfMgmtExt               Skip         0
        12     DeployUpdateIOM                           Skip         0
        13     DeployPollUpdateOfIOM                     Skip         0
        14     DeployActivateIOM                         Skip         0
        15     DeployPollActivateOfIOM                   Skip         0
        16     DeployActivateRemoteFI                    Skip         0
        17     DeployPollActivateOfRemoteFI              Skip         0
        18     DeployWaitForUserAck                      Skip         0
        19     DeployActivateLocalFI                     Success      0
        20     DeployPollActivateOfLocalFI               In Progress  1
```

참고 단계의 상태가 "In Progress(진행 중)"에서 "Skip(건너뛰기)" 또는 "Success(성공)"로 변경될 때까지 다음 단계로 진행하지 마십시오.

단계 **10** 시스템 복구에 사용한 이미지에 맞는 플랫폼 번들 이미지가 설치되어 있는 경우, 나중에 시스템을 로드할 때 사용할 수 있도록 수동으로 킥스타트 및 시스템 이미지를 활성화해야 합니다. 사용된 복구 이미지와 동일한 이미지가 있는 플랫폼 번들을 설치하는 경우 자동 활성화가 적용되지 않습니다.

a) 패브릭 인터커넥트 a의 범위를 설정합니다.

```
FP9300-A# scope fabric-interconnect a
```

b) 실행 중인 커널 버전 및 실행 중인 시스템 버전을 보려면 **show version** 명령을 사용합니다. 이러한 문자열을 사용하여 이미지를 활성화합니다.

```
FP9300-A /fabric-interconnect # show version
```

참고 Startup-Kern-Vers 및 Startup-Sys-Vers가 이미 설정되어 있고 Running-Kern-Vers 및 Running-Sys-Vers와 일치하는 경우, 이미지를 활성화할 필요가 없으며 11단계를 진행할 수 있습니다.

c) 다음 명령을 입력하여 이미지를 활성화합니다.

```
FP9300-A /fabric-interconnect # activate firmware
  kernel-version <running_kernel_version> system-version <running_system_version>
commit-buffer
```

참고 서버 상태가 "Disk Failed(디스크 실패)"로 변경될 수 있습니다. 이 메시지에 대해 걱정할 필요가 없으며 이 절차를 계속 진행할 수 있습니다.

d) 시작 버전이 올바르게 설정되었는지 확인하고 이미지의 활성화 상태를 모니터링하려면 **show version** 명령을 사용합니다.

중요 상태가 "Activating(활성)"에서 "Ready(준비)"로 변경될 때까지 다음 단계로 진행하지 마십시오.

```
FP9300-A /fabric-interconnect # show version
```

예제:

```
FP9300-A /firmware # top
FP9300-A# scope fabric-interconnect a
FP9300-A /fabric-interconnect # show version
Fabric Interconnect A:
    Running-Kern-Vers: 5.0(3)N2(4.11.69)
    Running-Sys-Vers: 5.0(3)N2(4.11.69)
    Package-Vers: 2.1(1.73)
    Startup-Kern-Vers:
    Startup-Sys-Vers:
    Act-Kern-Status: Ready
    Act-Sys-Status: Ready
    Bootloader-Vers:

FP9300-A /fabric-interconnect # activate firmware kernel-version
  5.0(3)N2(4.11.69) system-version 5.0(3)N2(4.11.69)
Warning: When committed this command will reset the end-point
FP9300-A /fabric-interconnect* # commit-buffer
FP9300-A /fabric-interconnect # show version
Fabric Interconnect A:
```

```
    Running-Kern-Vers: 5.0(3)N2(4.11.69)
    Running-Sys-Vers: 5.0(3)N2(4.11.69)
    Package-Vers: 2.1(1.73)
    Startup-Kern-Vers: 5.0(3)N2(4.11.69)
    Startup-Sys-Vers: 5.0(3)N2(4.11.69)
    Act-Kern-Status: Activating
    Act-Sys-Status: Activating
    Bootloader-Vers:

FP9300-A /fabric-interconnect # show version
Fabric Interconnect A:
    Running-Kern-Vers: 5.0(3)N2(4.11.69)
    Running-Sys-Vers: 5.0(3)N2(4.11.69)
    Package-Vers: 2.1(1.73)
    Startup-Kern-Vers: 5.0(3)N2(4.11.69)
    Startup-Sys-Vers: 5.0(3)N2(4.11.69)
    Act-Kern-Status: Ready
    Act-Sys-Status: Ready
    Bootloader-Vers:
```

단계 **11** 시스템을 재부팅합니다.

예제:

```
FP9300-A /fabric-interconnect # top
FP9300-A# scope chassis 1
FP9300-A /chassis # reboot no-prompt
Starting chassis reboot. Monitor progress with the command "show fsm status"
FP9300-A /chassis #
```

시스템은 각 보안 모듈/엔진의 전원을 끈 다음 마지막으로 Firepower 4100/9300 섀시의 전원을 끄고 재시작합니다. 이 프로세스는 약 5~10분 정도 걸립니다.

단계 **12** 시스템 상태를 모니터링합니다. 서버 상태가 "Discovery(검색)"에서 "Config(구성)"로 바뀐 다음 마지막으로 "Ok"로 바뀝니다.

예제:

```
FP9300-A# show server status
Server  Slot Status                      Overall Status        Discovery
------- -------------------------------- --------------------- ---------
1/1     Equipped                         Discovery             In Progress
1/2     Equipped                         Discovery             In Progress
1/3     Empty

FP9300-A# show server status
Server  Slot Status                      Overall Status        Discovery
------- -------------------------------- --------------------- ---------
1/1     Equipped                         Config                Complete
1/2     Equipped                         Config                Complete
1/3     Empty

FP9300-A# show server status
Server  Slot Status                      Overall Status        Discovery
------- -------------------------------- --------------------- ---------
1/1     Equipped                         Ok                    Complete
1/2     Equipped                         Ok                    Complete
1/3     Empty
```

Overall Status(전체 상태)가 "Ok"이면 시스템이 복구된 것입니다. 여전히 보안 어플라이언스를 재구성하고(라이선스 구성 포함) 논리적 디바이스를 다시 생성해야 합니다. 자세한 내용:

- Firepower 9300 빠른 시작 가이드—http://www.cisco.com/go/firepower9300-quick
- Firepower 9300 환경 설정 가이드—http://www.cisco.com/go/firepower9300-config
- Firepower 4100 Series 빠른 시작 가이드—http://www.cisco.com/go/firepower4100-quick
- Firepower 4100 Series 환경 설정 가이드—http://www.cisco.com/go/firepower4100-config

관리자 암호를 알 수 없는 경우 공장 기본 구성 복원

이 절차에서는 Firepower 4100/9300 섀시 시스템을 관리자 비밀번호를 포함한 기본 구성 설정으로 되돌립니다. 관리자 비밀번호를 알 수 없는 경우 이 절차를 사용하여 디바이스에서 구성을 재설정합니다. 이 절차를 수행하면 설치된 모든 논리적 디바이스도 지워집니다.

참고 이 절차를 수행하려면 Firepower 4100/9300 섀시에 대한 콘솔 액세스가 필요합니다.

프로시저

단계 **1** 제공된 콘솔 케이블을 사용하여 컴퓨터를 콘솔 포트에 연결하고, 9600 baud, 8 data bit, 패리티 없음, 1 stop bit, flow control 없음으로 설정된 터미널 에뮬레이터를 사용하여 콘솔에 연결합니다. 콘솔 케이블에 대한 자세한 내용은 Cisco Firepower 9300 하드웨어 설치 가이드를 참조하십시오.

단계 **2** 디바이스 전원을 켭니다. 다음 프롬프트가 표시되면 ESC를 눌러 부팅을 중지합니다.

예제:

```
!!  Rommon image verified successfully  !!

Cisco System ROMMON, Version 1.0.09, RELEASE SOFTWARE
Copyright (c) 1994-2015  by Cisco Systems, Inc.

Current image running: Boot ROM0
Last reset cause: ResetRequest
DIMM Slot 0 : Present
DIMM Slot 1 : Present
No USB drive !!
BIOS has been locked !!

Platform FPR9K-SUP with 16384 Mbytes of main memory
MAC Address: 00:00:00:00:00:00

find the string ! boot bootflash:/installables/switch/fxos-k9-kickstart.5.0.3.N2.3.14.69.SPA
 bootflash:/installables/switch/fxos-k9-system.5.0.3.N2.3.14.69.SPA

Use BREAK, ESC or CTRL+L to interrupt boot.
Use SPACE to begin boot immediately.
```

```
Boot interrupted.
rommon 1 >
```

단계 **3** 킥스타트 및 시스템 이미지 이름을 기록합니다.

예제:

```
bootflash:/installables/switch/fxos-k9-kickstart.5.0.3.N2.3.14.69.SPA
bootflash:/installables/switch/fxos-k9-system.5.0.3.N2.3.14.69.SPA
```

단계 **4** 킥스타트 이미지를 로드합니다.

rommon 1 > **boot** *kickstart_image*

예제:

```
rommon 1 > boot bootflash:/installables/switch/fxos-k9-kickstart.5.0.3.N2.3.14.69.SPA
 !! Kickstart Image verified successfully !!

Linux version: 2.6.27.47 (security@cisco.com) #1 SMP Tue Nov 24 12:10:28 PST 2015
[ 0.000000] Fastboot Memory at 0c100000 of size 201326592
Usage: init 0123456SsQqAaBbCcUu
INIT: POST INIT Starts at Wed Jun 1 13:46:33 UTC 2016
can't create lock file /var/lock/mtab~302: No such file or directory (use -n flag to override)
S10mount-ramfs.supnuovaca Mounting /isan 3000m
Mounted /isan
TAC support: http://www.cisco.com/tac
Copyright (c) 2002-2015, Cisco Systems, Inc. All rights reserved.
The copyrights to certain works contained in this software are
owned by other third parties and used and distributed under
license. Certain components of this software are licensed under
the GNU General Public License (GPL) version 2.0 or the GNU
Lesser General Public License (LGPL) Version 2.1. A copy of each
such license is available at
http://www.opensource.org/licenses/gpl-2.0.php and
http://www.opensource.org/licenses/lgpl-2.1.php
switch(boot)#
```

단계 **5** config terminal 모드를 시작합니다.

switch(boot) # **config terminal**

예제:

```
switch(boot)#
switch(boot)# config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
```

단계 **6** 비밀번호를 재설정하고 변경 사항을 확인합니다.

switch(boot) (config) # **admin-password erase**

참고 이 단계에서는 모든 구성을 지우고 시스템을 기본 구성 설정으로 되돌립니다.

예제:

```
switch(boot)(config)# admin-password erase
Your password and configuration will be erased!
Do you want to continue? (y/n)  [n] y
```

단계 **7** config terminal 모드를 종료합니다.

switch(boot) (config) # **exit**

단계 **8** 이 절차의 3단계에서 기록한 시스템 이미지를 로드하고 초기 구성, 8 페이지 작업 플로우를 사용하여 시스템을 처음부터 구성합니다.

switch(boot) # **load** *system_image*

예제**:**

```
switch(boot)# load bootflash:/installables/switch/fxos-k9-system.5.0.3.N2.3.14.69.SPA

Uncompressing system image: bootflash:/installables/switch/fxos-k9-system.5.0.3.N2.3.14.69.SPA
```

트러블슈팅 로그 파일 생성

로그 파일을 생성하여 트러블슈팅에 도움을 주거나, 요청을 받은 경우 Cisco TAC에 전송할 수 있습니다.

프로시저

단계 **1** **Tools(**도구**)** > **Troubleshooting Logs(**트러블슈팅 로그**)**를 선택합니다.

단계 **2** 드롭다운 목록에서 생성하려는 로그 파일 유형을 선택합니다.

- 섀시 - 수퍼바이저 및 서비스 매니저를 비롯하여 섀시 문제 및 소프트웨어 문제를 트러블슈팅하는 데 사용할 로그 파일을 생성합니다.
- 모듈 <#> - 보안 모듈/엔진 문제를 트러블슈팅하는 데 사용할 로그 파일을 생성합니다.

단계 **3** **Generate Log(**로그 생성**)**를 클릭합니다.

단계 **4** **Yes(**예**)**를 클릭하여 로그 파일을 생성할 것을 확인합니다.

로그 파일이 생성됩니다. 이 프로세스는 시간이 걸릴 수 있습니다. 로그 파일을 생성하는 동안 노란색 상태 메시지가 표시됩니다. 로그 파일 생성을 취소하려면 상태 메시지에서 **Abort Job(**작업 중단**)**을 클릭하면 됩니다. 로그 파일이 생성되고 나면, 상태 메시지가 녹색으로 변경되어 작업이 성공적으로 완료되었음을 나타냅니다.

단계 **5** 생성된 로그 파일을 다운로드하려면 **Download Files(**파일 다운로드**)** 목록에서 로그 파일을 찾은 다음, **Download(**다운로드**)**를 클릭합니다. 로그 파일은 techsupport 폴더 아래에 저장됩니다.

참고 새로 생성된 파일이 **Download Files(**파일 다운로드**)** 목록에 표시되도록 **Refresh(**새로 고침**)**를 클릭해야 할 수 있습니다.

단계 **6** 생성된 로그 파일을 삭제하려면 **Download Files(**파일 다운로드**)** 목록에서 로그 파일을 찾은 다음, **Delete(**삭제**)**를 클릭합니다.

모듈 코어 덤프 활성화

모듈에서 코어 덤프를 활성화하면 시스템 충돌 시 트러블슈팅에 도움이 되거나, 요청 시 Cisco TAC로 전송할 수 있습니다.

프로시저

단계 **1** 원하는 모듈에 연결합니다. 예:

Firepower# connect module 1 console

단계 **2** (선택 사항) 다음 명령을 입력하여 현재 코어 덤프 상태를 확인합니다.

Firepower-module1> show coredump detail

명령 출력에는 코어 덤프 압축의 활성화 여부를 포함하여 현재 코어 덤프 상태 정보가 표시 됩니다.

예제:

```
Firepower-module1>show coredump detail
Configured status: ENABLED.
ASA Coredump: ENABLED.
Bootup status: ENABLED.
Compress during crash: DISABLED.
```

참고 이 명령은 어플라이언스에서 ASA 논리적 디바이스를 실행하는 경우에만 사용할 수 있으며, 어플라이언스에서 FTD 논리적 디바이스를 실행할 때는 사용할 수 없습니다.

단계 **3** **config coredump** 명령을 사용하여 코어 덤프를 활성화 또는 비활성화하고, 충돌 중에 코어 덤프 압축을 활성화 또는 비활성화합니다.

- **config coredump enable** 을 사용하여 충돌 중에 코어 덤프 생성을 활성합니다.
- **config coredump disable** 을 사용하여 충돌 중에 코어 덤프 생성을 비활성화합니다.
- **config coredump compress enable** 을 사용하여 코어 덤프의 압축을 활성합니다.
- **config coredump compress disable** 을 사용하여 코어 덤프 압축을 비활성화합니다.

예제:

```
Firepower-module1>config coredump enable
Coredump enabled successfully.
ASA coredump enabled, do 'config coredump disableAsa' to disable
Firepower-module1>config coredump compress enable
WARNING: Enabling compression delays system reboot for several minutes after a system
failure.  Are you sure? (y/n):
y
Firepower-module1>
```

참고 코어 덤프 파일은 디스크 공간을 사용하며, 공간이 부족하고 압축이 활성화되지 않은 경우 코어 덤프가 활성화된 경우에도 코어 덤프 파일이 저장되지 않을 수 있습니다.

Firepower 4100/9300 섀시의 일련 번호 찾기

Firepower 4100/9300 섀시및 일련 번호에 대한 세부 정보를 찾을 수 있습니다. Firepower 4100/9300 섀시의 일련 번호는 논리적 디바이스의 일련 번호와 다릅니다.

프로시저

단계 **1** **Overview** > **Inventory** > **All**을 선택합니다.

이 표에서는 섀시에 설치된 구성 요소를 나열하고 해당 구성 요소에 대한 관련 세부 정보를 제공합니다.

단계 **2** **Serial** 열에서 섀시 일련번호를 찾으십시오.

RAID 가상 드라이브 재구성

RAID(Redundant Array of Independent Disks)는 고성능과 내결함성(fault tolerance)을 제공하는 여러 독립적인 실제 드라이브로 구성된 어레이 또는 그룹입니다. 드라이브 그룹은 실제 드라이브의 그룹입니다. 이러한 드라이브는 가상 드라이브로 알려진 파티션으로 관리됩니다.

RAID 드라이브 그룹은 단일 드라이브 스토리지 시스템보다 데이터 스토리지 안정성과 내결함성을 더욱 개선합니다. 나머지 드라이브에서 누락된 데이터를 재구성하여 드라이브 장애로 인한 데이터 손실을 방지할 수 있습니다. RAID는 I/O 성능을 개선하고 스토리지 하위 시스템의 안정성을 향상합니다.

RAID 드라이브 중 하나에 장애가 발생했거나 오프라인인 경우 RAID 가상 드라이브는 성능이 저하된 상태로 간주됩니다. 이 절차를 사용하여 RAID 가상 드라이브가 성능이 저하된 상태인지 확인하고, 일시적으로 로컬 디스크 구성 보호 정책을 no로 설정하여 필요한 경우 재구축합니다.

참고 로컬 디스크 구성 보호 정책을 no로 설정하면 디스크의 모든 데이터가 삭제됩니다.

프로시저

단계 **1** RAID 상태를 확인합니다.

1. 섀시 모드로 들어갑니다.

scope chassis

2. 서버 모드를 활성화합니다.

   scope server 1

3. RAID 컨트롤러를 입력합니다.

   scope raid-controller 1 sas

4. 가상 드라이브를 봅니다.

   show virtual-drive

   RAID 가상 드라이브의 성능이 저하된 경우에는, 작동성이 **Degraded**로 표시됩니다. 예를 들면 다음과 같습니다.

```
Virtual Drive:
    ID: 0
    Block Size: 512
    Blocks: 3123046400
    Size (MB): 1524925
    Operability: Degraded
Presence: Equipped
```

단계 **2** RAID 드라이브를 재구축하려면 로컬 디스크 구성 정책 보호를 no로 설정합니다. 참고 - 이 단계를 완료하면 디스크의 모든 데이터가 삭제됩니다.

1. 조직 범위를 입력합니다.

   scope org

2. 로컬 디스크 구성 정책 범위를 입력합니다.

   scope local-disk-config-policy ssp-default

3. 보호를 no로 설정합니다.

   set protect no

4. 구성을 커밋합니다.

   commit-buffer

단계 **3** RAID 드라이브가 재구축될 때까지 기다립니다. RAID 재구축 상태를 확인합니다.

scope chassis 1

show server

RAID 드라이브가 성공적으로 재구축되면, 슬롯의 전체 상태가 **Ok**로 표시됩니다. 예를 들면 다음과 같습니다.

예제**:**

```
Server:
    Slot    Overall Status        Service Profile
    ------- --------------------- ---------------
          1 Ok                    ssp-sprof-1
```

단계 **4** RAID 드라이브가 성공적으로 재구축되면, 로컬 디스크 구성 정책 보호를 다시 yes로 설정합니다.

1. 조직 범위를 입력합니다.

   scope org

2. 로컬 디스크 컨피그레이션 정책 범위를 입력합니다.

   scope local-disk-config-policy ssp-default

3. 보호를 no로 설정합니다.

   set protect yes

4. 구성을 커밋합니다.

   commit-buffer

SSD 문제 식별

다음 절차를 사용하여 정보를 수집하고 디바이스에 설치된 SSD에서 발생할 수 있는 문제를 파악합니다. SSD 문제의 한 가지 예시 증상은 DME(Data Management Engine) 프로세스가 시작되지 않는 것입니다.

참고 새 SSD를 삽입하면 블레이드 BIOS가 탐지된 후 기본 정보(유형, 모델, SN 등)만 인테리어 아래에 채워집니다. SSP-OS 업그레이드가 완료되어야 인벤토리 아래에 로컬 디스크 데이터가 채워집니다. SSP OS 업그레이드가 여전히 "Updating state(업데이트 중 상태)"인 경우 인벤토리에 로컬 디스크에 대한 항목이 표시되지 않으며 SSD 연결과 관련된 오류 메시지도 표시되지 않습니다.

아래 로깅 파일의 출력에서 SSD에 문제가 있는 경우 TAC에 문의합니다(https://www.cisco.com/c/en/us/buy/product-returns-replacements-rma.html 참조).

프로시저

단계 **1** FXOS 명령 셸에 연결합니다.

connect fxos

단계 **2** nvram 로깅 파일을 표시합니다.

show logging nvram

오류 출력 예:

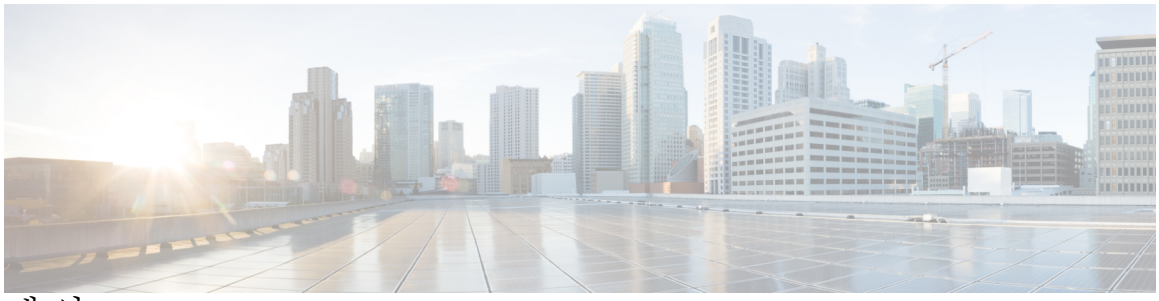
```
2020 Oct 22 13:03:26 MDCNGIPSAPL02 %$ VDC-1 %$ Oct 22 13:03:25 %KERN-2-SYSTEM_MSG:
[28175880.598580] EXT3-fs error (device sda4): ext3_get_inode_loc: unable to read inode
block - inode=14, block=6
```

단계 **3** 로깅 파일을 표시합니다.

show logging logfile

오류 출력 예:

```
   2020 Oct 21 21:11:25  (none) kernel: [28118744.718445] EXT3-fs error (device sda4):
ext3_get_inode_loc: unable to read inode block - inode=14, block=6
```



색 인

G

H

I

J

K

L

M

N

U