Firepower 4100/9300의 논리적 디바이스

Firepower 4100/9300은 하나 이상의 논리적 디바이스를 설치할 수 있는 유연한 보안 플랫폼입니다. threat defense을 management center에 추가하기 전에 섀시 인터페이스를 구성하고 논리적 디바이스를 추가하고 Secure Firewall 섀시 관리자 또는 FXOS CLI를 사용하는 Firepower 4100/9300 섀시의 디바이스에 인터페이스를 할당해야 합니다. 이 장에서는 기본 인터페이스 구성 및 Secure Firewall 섀시 관리자를 사용하여 독립형 디바이스 또는 고가용성 논리적 디바이스를 추가하는 방법을 설명합니다. 클러스터형 논리적 디바이스를 추가하려면 Firepower 4100/9300 클러스터링의 내용을 참조하십시오. FXOS CLI를 사용하려면 FXOS CLI 구성 가이드를 참조하십시오. 고급 FXOS 절차 및 트러블슈팅에 대한 자세한 내용은 FXOS 구성 가이드를 참조하십시오.

- 인터페이스 정보, 1 페이지
- 논리적 디바이스 정보, 17 페이지
- 컨테이너 인스턴스용 라이선스, 26 페이지
- 논리적 디바이스의 요구 사항 및 사전 요구 사항, 26 페이지
- 논리적 디바이스 관련 지침 및 제한 사항, 30 페이지
- 인터페이스 구성, 33 페이지
- 논리적 디바이스 구성, 38 페이지
- 논리적 디바이스의 기록, 52 페이지

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 섀시 관리자를 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. 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(위협 방어-사용-management center 전용)에서 공유할 수 있습니다. 각 컨테이너 인스턴스는 이 인터페이스를 공유하는 다른 모든 인스턴스와 백플레인을 통해 통신할 수 있습니다. 공유 인터페이스는 배포할 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.개별 섀시 관리 인터페이스에 대한 내용은 섀시 관리 인터페이스, 1 페이지 항목을 참조하십시오.

참고 관리 인터페이스를 변경하면 논리적 디바이스가 재부팅됩니다. 예를 들어 e1/1에서 e1/2로 변경하면 논리적 디바이스가 재부팅되어 새 관리가 적용됩니다.

- 이벤트 처리-위협 방어-사용-management center 디바이스의 보조 관리 인터페이스로 사용합니다. 이 인터페이스를 사용하려면 위협 방어CLI에서 해당 IP 주소 및 기타 매개변수를 구성해야 합니

다. 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. device manager 및 CDO는 클러스터링을 지원하지 않습니다.

참고 이 장에서는 *FXOS* VLAN 하위 인터페이스에 대해서만 설명합니다. threat defense 애플리케이션 내에 하위 인터페이스를 추가할 수 있습니다. 자세한 내용은 FXOS 인터페이스와 애플리케이션 인터페이스 비교, 4 페이지를 참조하십시오.

독립형 및 클러스터 배포에서 threat defense 및 ASA 애플리케이션에 대한 인터페이스 유형 지원은 다음 표를 참조하십시오.

표 *1:* 인터페이스 유형 지원

애플리케이션		데이터	데이터: 하위 인터페이스	데이터 공유	데이터 공유: 하위 인터페이스	관리	이벤트	클러스터 (EtherChannel에만 해당)	클러스터: 하위 인터페이스
Threat Defense	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.

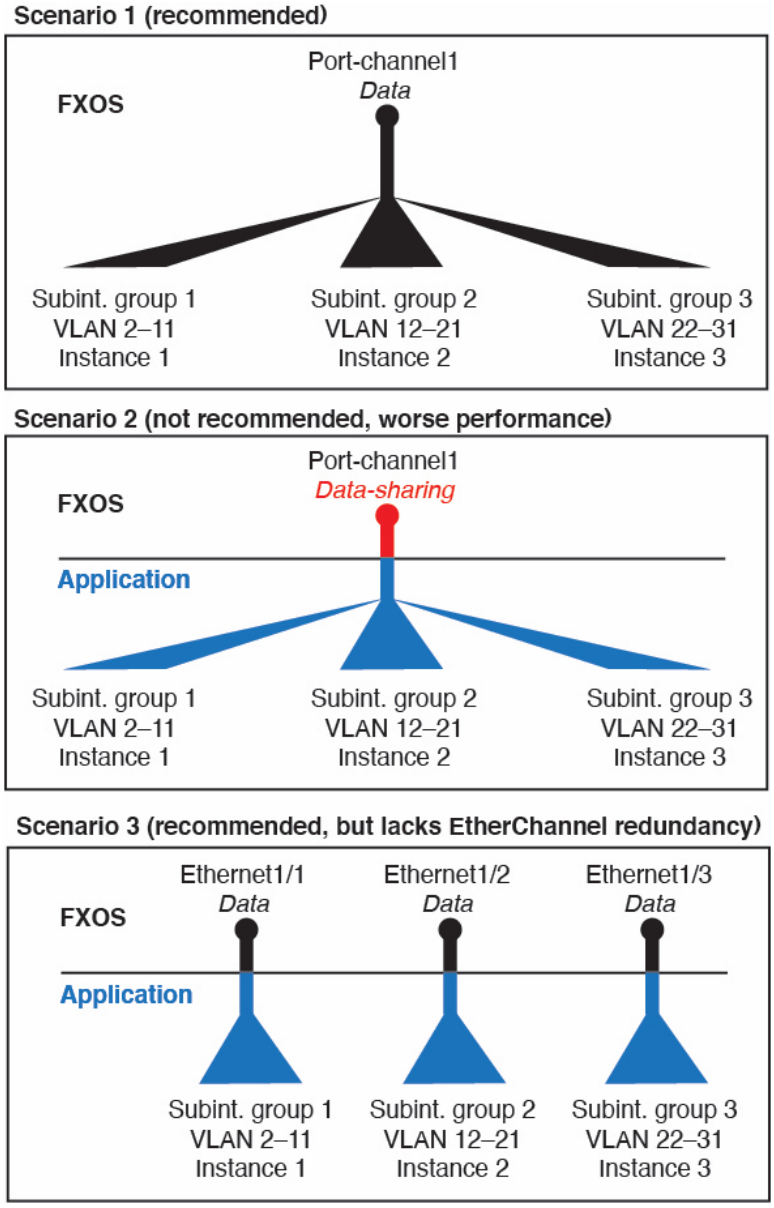
다음 섹션에서는 FXOS와 인터페이스에 대한 애플리케이션 간의 상호 작용에 대해 설명합니다.

VLAN 하위 인터페이스

논리적 디바이스의 경우에는 애플리케이션 내에서 VLAN 하위 인터페이스를 생성할 수 있습니다.

독립형 모드만의 컨테이너 인스턴스의 경우에는, 또한 FXOS에서 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.

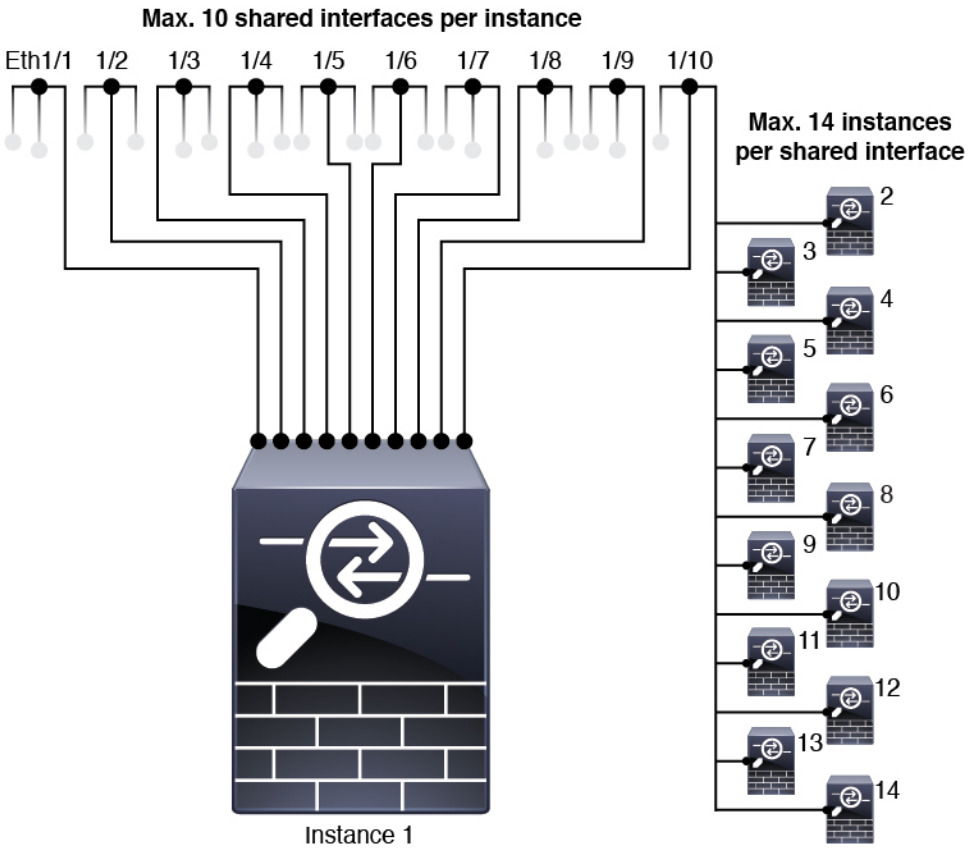
애플리케이션 내의 인터페이스 기본 상태는 인터페이스 유형에 따라 달라집니다. 예를 들어 물리적 인터페이스 또는 EtherChannel은 애플리케이션 내에서 기본적으로 비활성화되지만 하위 인터페이스는 기본적으로 활성화됩니다.

공유 인터페이스 확장성

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

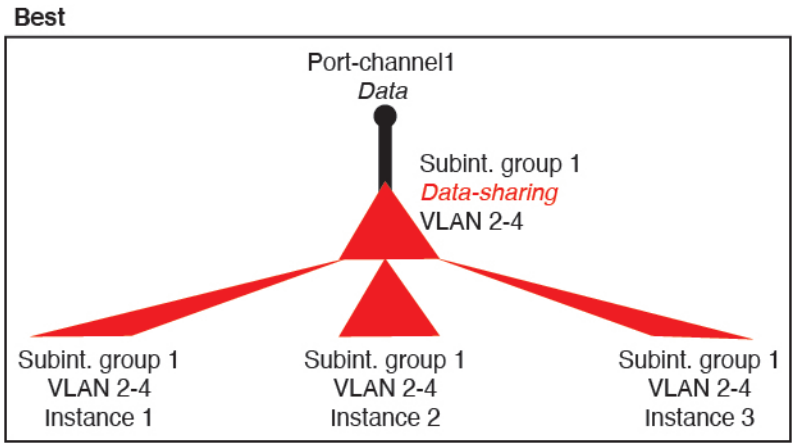
포워딩 테이블의 최적의 확장성을 위해 최대한 적은 수의 인터페이스를 공유합니다. 대신, 하나 이상의 물리적 인터페이스에서 최대 500개의 VLAN 하위 인터페이스를 생성하고 컨테이너 인스턴스 사이에 VLAN을 나눌 수 있습니다.

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.

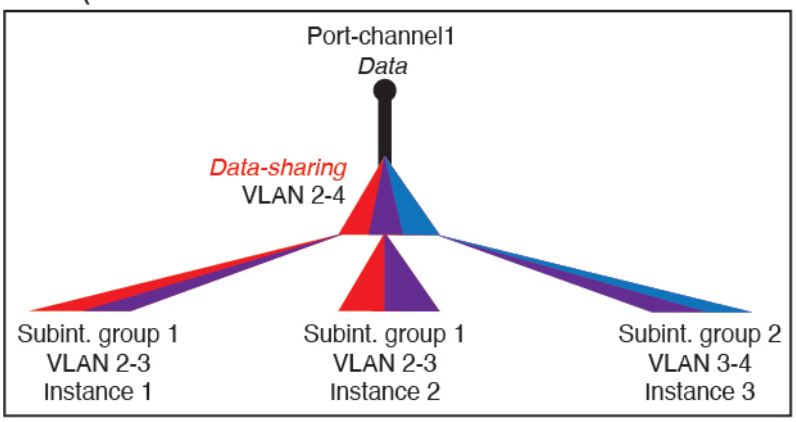
   예를 들어 대규모 EtherChannel 하나를 생성해 유사한 종류의 모든 인터페이스를 함께 번들로 묶은 다음 해당 EtherChannel의 하위 인터페이스를 공유합니다. 즉, Port-Channel2, Port-Channel3 및 Port-Channel4를 공유하는 대신 Port-Channel1.2, 3 및 4를 공유합니다. 단일 상위 인터페이스의 하위 인터페이스를 공유하면 상위 인터페이스 전체에서 하위 인터페이스를 공유하거나 물리적/EtherChannel 인터페이스를 공유할 때 VLAN 그룹 테이블이 전달 테이블보다 더 잘 확장됩니다.

   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.

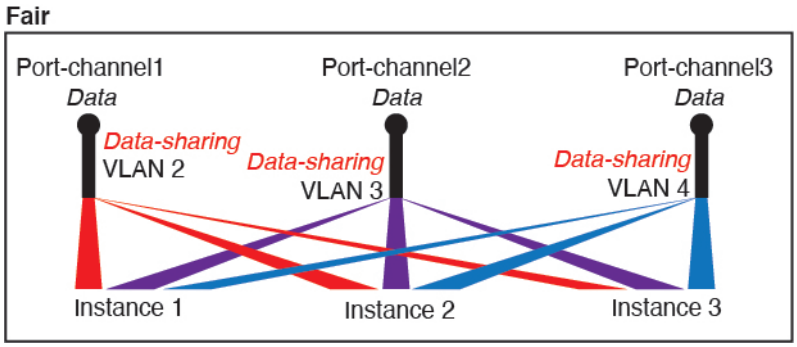
   그림 ***3:*** 좋음 **:** 하나의 상위에서 여러 하위 인터페이스 그룹 공유



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.

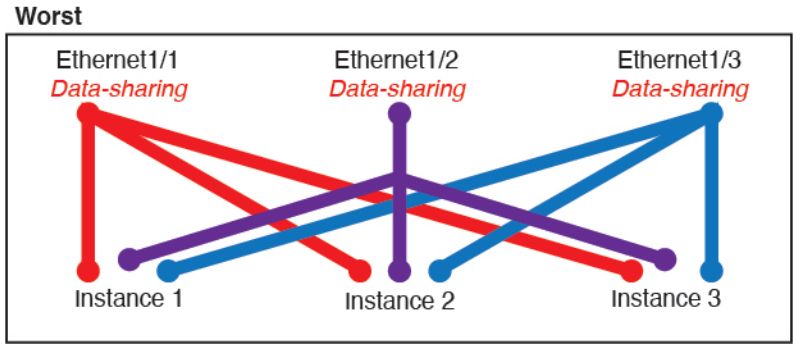
그림 *4:* 보통 *:* 개별 상위의 공유 하위 인터페이스



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 2: Firepower 9300(SM-44 3개)의 물리적/EtherChannel 인터페이스 및 인스턴스, 10 페이지
- 표 3: Firepower 9300(SM-44 3개)에 있는 상위 인터페이스 하나의 하위 인터페이스 및 인스턴스, 12 페이지
- 표 4: Firepower 9300(SM-44 1개)의 물리적/EtherChannel 인터페이스 및 인스턴스, 13 페이지
- 표 5: Firepower 9300(SM-44 1개)에 있는 상위 인터페이스 하나의 하위 인터페이스 및 인스턴스, 15 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 ***2: 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 *3: 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 ***4: 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.

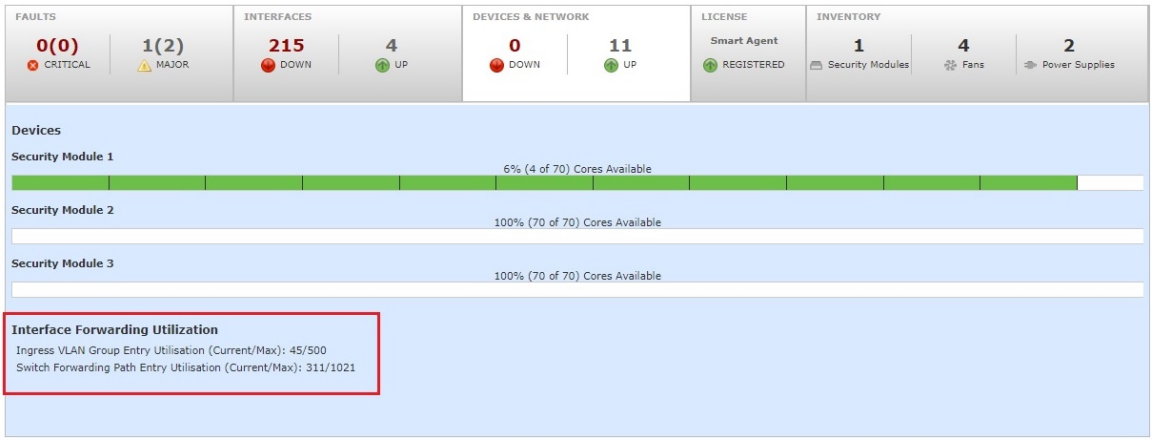
표 ***5: Firepower 9300(SM-44 1개)***에 있는 상위 인터페이스 하나의 하위 인터페이스 및 인스턴스

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 **Devices & Network(**디바이스 및 네트워크**)** > **Interface Forwarding Utilization(**인터페이스 포워딩 사용률**)** 영역을 확인하고 예를 들면 다음과 같습니다.



Threat Defense에 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.

위협 방어 애플리케이션에서 인라인 집합을 구성하고 링크 상태 전파를 활성화하면 위협 방어에서 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 발생

한 인터페이스가 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.

참고 하드웨어 바이패스를 활성화하고 동일한 인라인 세트에 대해 상태 전파를 연결하지 마십시오.

논리적 디바이스 정보

논리적 디바이스를 사용하면 애플리케이션 인스턴스 하나(ASA 또는 위협 방어)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 위협 방어)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. device manager에서는 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 management center를 사용하는 위협 방어에 대해서만 지원되며, ASA 또는 device manager를 사용하는 위협 방어에 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 위협 방어 기능이 지원됩니다. 다중 컨텍스트 모드에서는 리소스가 공유되므로 지정된 플랫폼에서 더 많은 컨텍스트가 지원됩니다. 위협 방어에서는 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, 7 페이지 및 컨테이너 인스턴스에 VLAN 하위 인터페이스 추가, 37 페이지를 참조하십시오.

참고 이 문서에서는 *FXOS* VLAN 하위 인터페이스에 대해서만 설명합니다. threat defense 애플리케이션 내에 하위 인터페이스를 추가할 수 있습니다. 자세한 내용은 FXOS 인터페이스와 애플리케이션 인터페이스 비교, 4 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 주소

가 없으면 인스턴스로 직접 라우팅할 수 없습니다. 또한 애플리케이션 내에서 각 인터페이스를 구성할 때 수동으로 MAC 주소를 설정할 수도 있습니다.

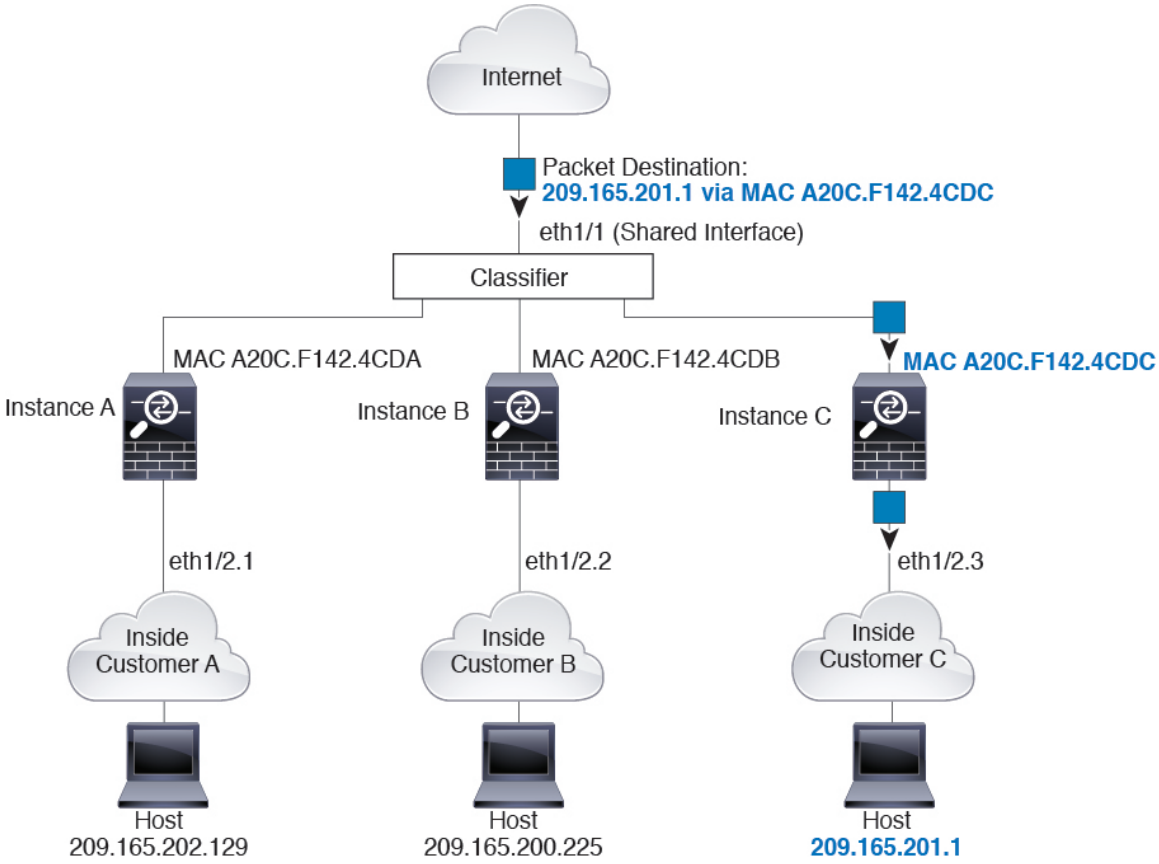
참고 대상 MAC 주소가 멀티캐스트 또는 브로드캐스트 MAC 주소인 경우 패킷이 복제되어 각 인스턴스에 배포됩니다.

분류의 예

MAC 주소를 사용하는 공유 인터페이스를 통한 패킷 분류

다음 그림은 외부 인터페이스를 공유하는 여러 인스턴스를 보여 줍니다. 분류자는 인스턴스 C에 패킷을 할당합니다. 라우터에서 패킷을 보내는 MAC 주소가 인스턴스 C에 포함되어 있기 때문입니다.

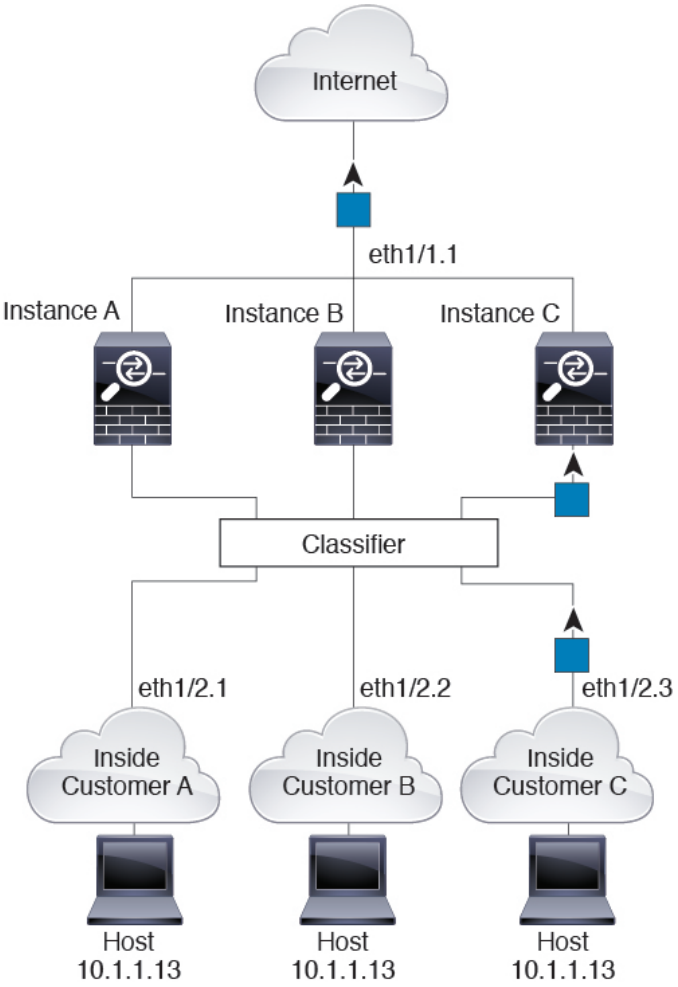
그림 *6: MAC* 주소를 사용하는 공유 인터페이스를 통한 패킷 분류



내부 네트워크로부터 수신하는 트래픽

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.

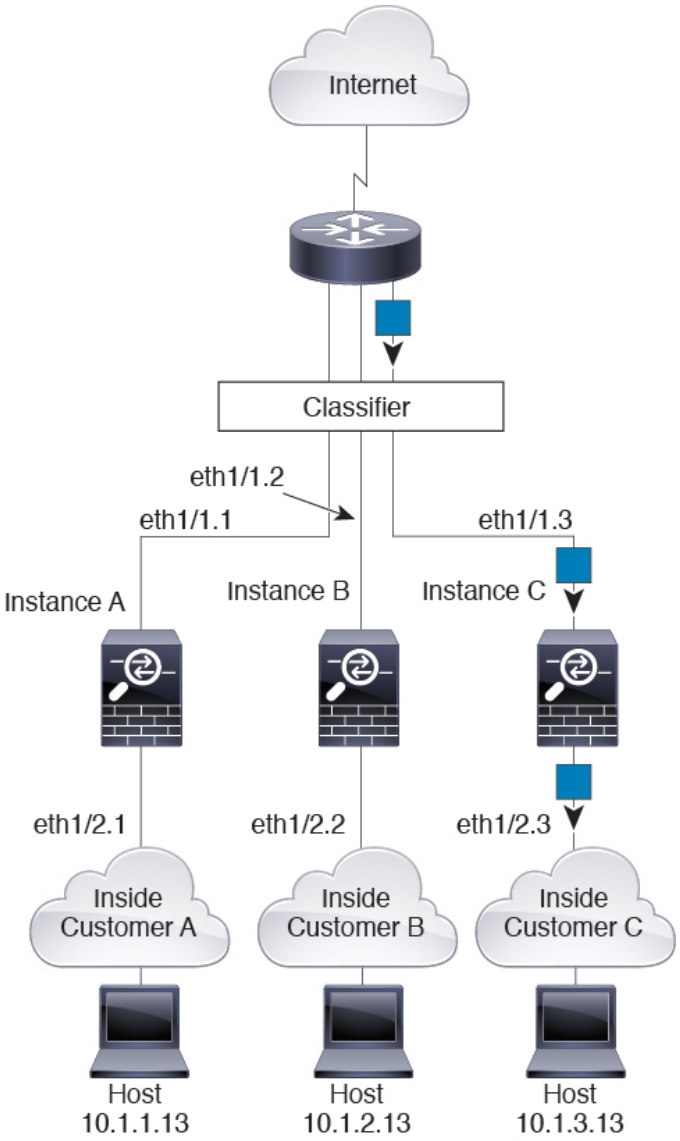
그림 *7:* 내부 네트워크로부터 수신하는 트래픽



투명한 방화벽 인스턴스

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.

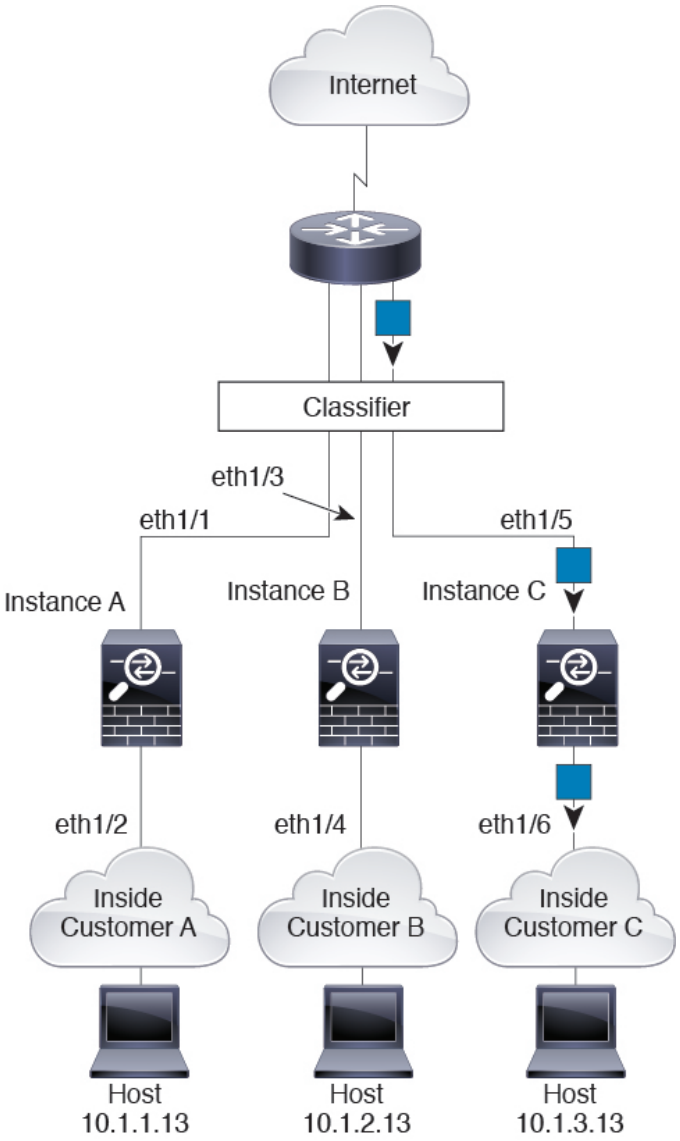
그림 *8:* 투명한 방화벽 인스턴스



인라인 세트

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 *9:* 인라인 세트

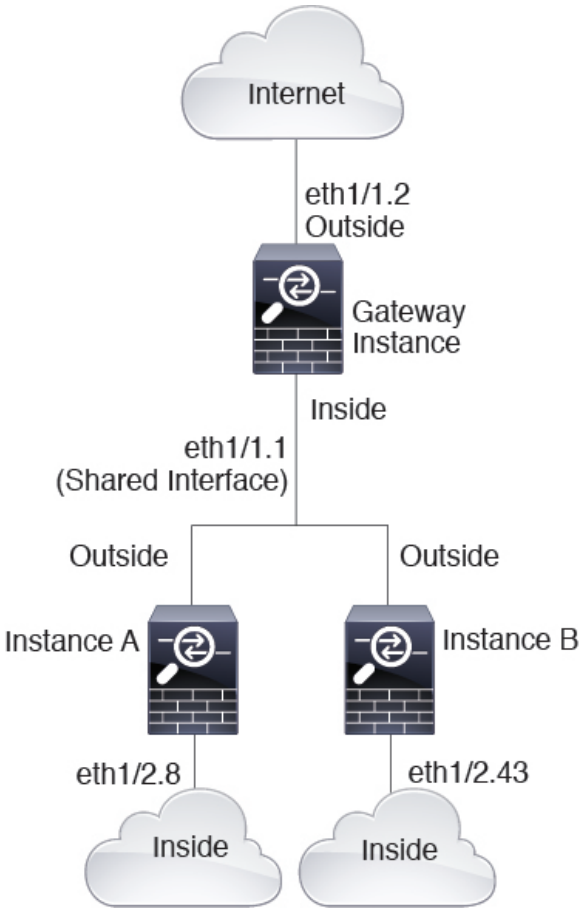


연속 컨테이너 인스턴스

다른 인스턴스 바로 앞에 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 ***10:*** 연속 인스턴스



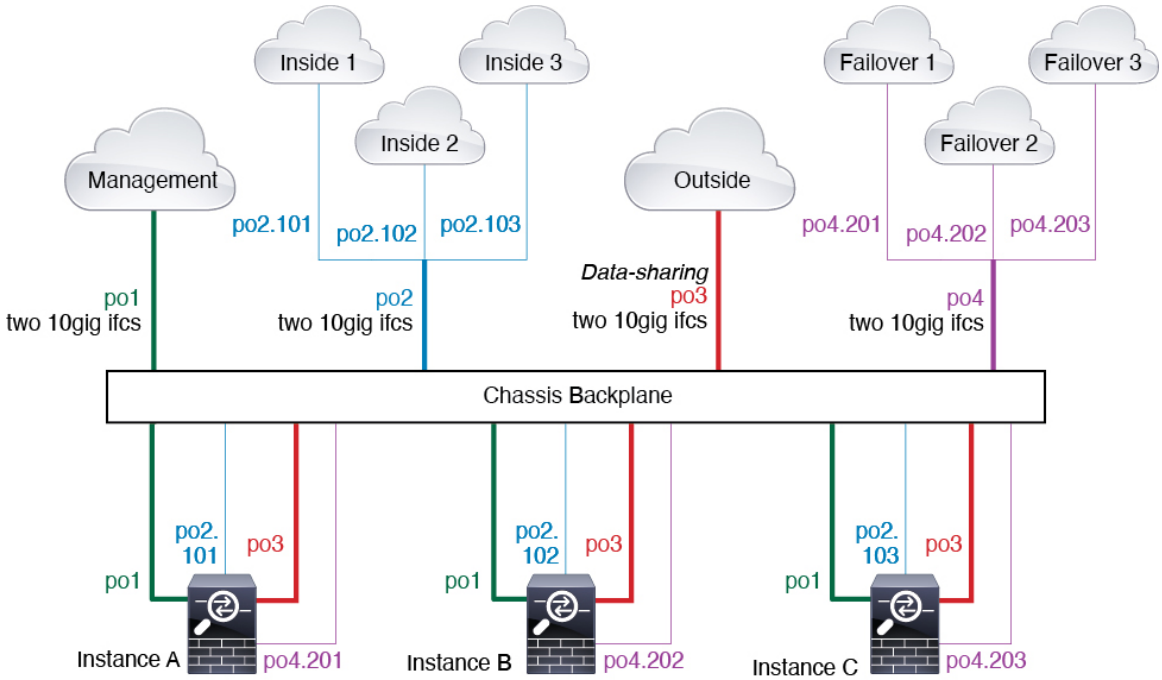
참고 고가용성에 연속 인스턴스(공유 인터페이스 사용)를 사용하지 마십시오. 페일오버가 수행되고 스탠바이 유닛이 다시 조인한 후에는 MAC 주소가 일시적으로 중복되어 중단이 발생할 수 있습니다. 대신 게이트웨이 인스턴스와 외부 스위치를 사용하는 내부 인스턴스에 고유한 인터페이스를 사용하여 인스턴스 간에 트래픽을 전달해야 합니다.

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

새시는 인스턴스 인터페이스용 MAC 주소를 자동으로 생성하며 각 인스턴스의 공유 인터페이스가 고유한 MAC 주소를 사용하도록 보장합니다.

인스턴스 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 인스턴스 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.

새시는 다음 형식을 사용하여 MAC 주소를 생성합니다.

A2*xx.yyzz.zzzz*

여기서 *xx.yy*는 사용자 정의 접두사 또는 시스템 정의 접두사이고 *zz.zzzz*는 새시에서 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, 29 페이지 섹션을 참조하십시오. 리소스 프로파일을 추가하려면 컨테이너 인스턴스에 대한 리소스 프로파일 추가, 39 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, 29 페이지 및 고가용성 쌍 추가, 47 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 사항의 내용을 참조하십시오.

컨테이너 인스턴스용 라이선스

모든 라이선스는 (Firepower 4100의) 보안 엔진/섀시 또는 (Firepower 9300의) 보안 모듈에 대해 소비되지만 컨테이너 라이선스에 대해서는 소비되지 않습니다. 자세한 내용은 다음을 참조하십시오.

- Essentials 라이선스는 보안 모듈/엔진당 하나씩 자동으로 할당됩니다.
- 기능 라이선스는 각 인스턴스에 대해 수동으로 할당되지만 사용자는 보안 모듈/엔진의 기능당 하나의 라이선스를 소비합니다. 예를 들어 3개의 보안 모듈이 있는 Firepower 9300에 대해서는 모듈당 하나의 URL 라이선스가 필요하므로 사용 중인 인스턴스 수와 관계없이 총 3개의 라이선스가 필요합니다.

대표적인 예는 다음과 같습니다.

표 ***6: Firepower 9300***의 컨테이너 인스턴스에 대한 샘플 라이선스 사용

Firepower 9300	인스턴스	라이선스
보안 모듈 1	인스턴스 1	Essentials, URL, 악성코드 방어
	인스턴스 2	Essentials, URL
	인스턴스 3	Essentials, URL
보안 모듈 2	인스턴스 4	Essentials, IPS
	인스턴스 5	Essentials, URL, 악성코드 방어, IPS
보안 모듈 3	인스턴스 6	Essentials, 악성코드 방어, IPS
	인스턴스 7	Essentials, IPS

표 ***7:*** 수총 라이선스 수

Essentials	**URL**	악성코드 방어	**IPS**
3	2	3	2

논리적 디바이스의 요구 사항 및 사전 요구 사항

요구 사항 및 사전 요구 사항에 대한 내용은 다음 섹션을 참조하십시오.

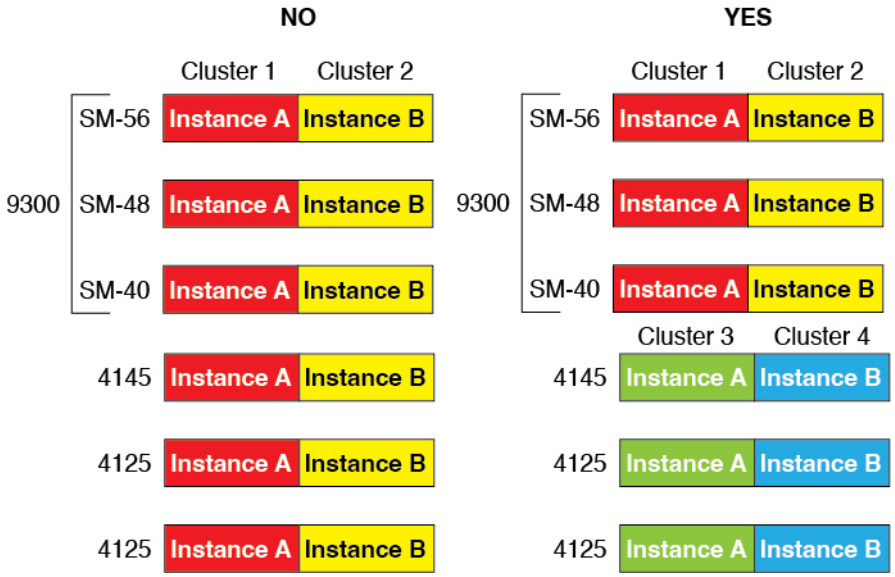
하드웨어 및 소프트웨어 조합에 대한 요구 사항 및 사전 요구 사항

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.
- 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 threat defense애플리케이션 유형 - 섀시의 개별 모듈에 서로 다른 애플리케이션 유형을 설치할 수 있습니다. 예를 들어, 모듈 1 및 모듈 2에는 ASA를 설치하고 모듈 3에는 threat defense를 설치할 수 있습니다.
- ASA 또는 threat defense 버전 - 애플리케이션 인스턴스 유형의 서로 다른 버전을 별도의 모듈에서 실행하거나 동일한 모듈에서 별도의 컨테이너 인스턴스로 실행할 수 있습니다. 예를 들어, 모듈 1에는 threat defense 6.3을, 모듈 2에는 threat defense 6.4를 설치하고, 모듈 3에는 threat defense 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 threat defense 애플리케이션 유형 - Firepower 4100에서는 하나의 애플리케이션 유형만 실행할 수 있습니다.
- threat defense 컨테이너 인스턴스 버전 - 동일한 모듈에서 별도의 컨테이너 인스턴스로 서로 다른 버전의 위협 방어를 실행할 수 있습니다.

컨테이너 인스턴스의 요구 사항 및 사전 요구 사항

지원되는 애플리케이션 유형

- management center을 사용한 threat defense

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 ***8:*** 모델당 최대 컨테이너 인스턴스 및 리소스

모델	최대 컨테이너 인스턴스 수	사용 가능한 **CPU** 코어	사용 가능한 **RAM**	사용 가능한 디스크 공간
Firepower 4112	3	22	78GB	308GB
Firepower 4115	7	46	162GB	308GB
Firepower 4125	10	62	162GB	644GB
Firepower 4140	7	70	222GB	311.8GB
Firepower 4145	14	86	344GB	608GB
Firepower 9300 SM-40 보안 모듈	13	78	334GB	1359GB
Firepower 9300 SM-48 보안 모듈	15	94	334GB	1341GB
Firepower 9300 SM-56 보안 모듈	18	110	334GB	1314GB

Management Center 필수조건

Firepower 4100 섀시 또는 Firepower 9300 모듈의 모든 인스턴스에서는 라이선싱 구현으로 인해 동일한 management center를 사용해야 합니다.

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.
- 컨테이너 인스턴스의 경우: 고가용성에 연속 인스턴스(공유 인터페이스 사용)를 사용하지 마십시오. 페일오버가 수행되고 스탠바이 유닛이 다시 조인한 후에는 MAC 주소가 일시적으로 중복되어 중단이 발생할 수 있습니다. 대신 게이트웨이 인스턴스와 외부 스위치를 사용하는 내부 인스턴스에 고유한 인터페이스를 사용하여 인스턴스 간에 트래픽을 전달해야 합니다.
- 기타 고가용성을 위한 시스템 요구 사항은 고가용성 시스템 요구 사항의 내용을 참조하십시오.

논리적 디바이스 관련 지침 및 제한 사항

지침 및 제한 사항은 다음 섹션을 참조하십시오.

인터페이스에 대한 지침 및 제한 사항

VLAN 하위 인터페이스

- 이 문서에서는 *FXOS* VLAN 하위 인터페이스에 대해서만 설명합니다. threat defense 애플리케이션 내에 하위 인터페이스를 추가할 수 있습니다. 자세한 내용은 FXOS 인터페이스와 애플리케이션 인터페이스 비교, 4 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.

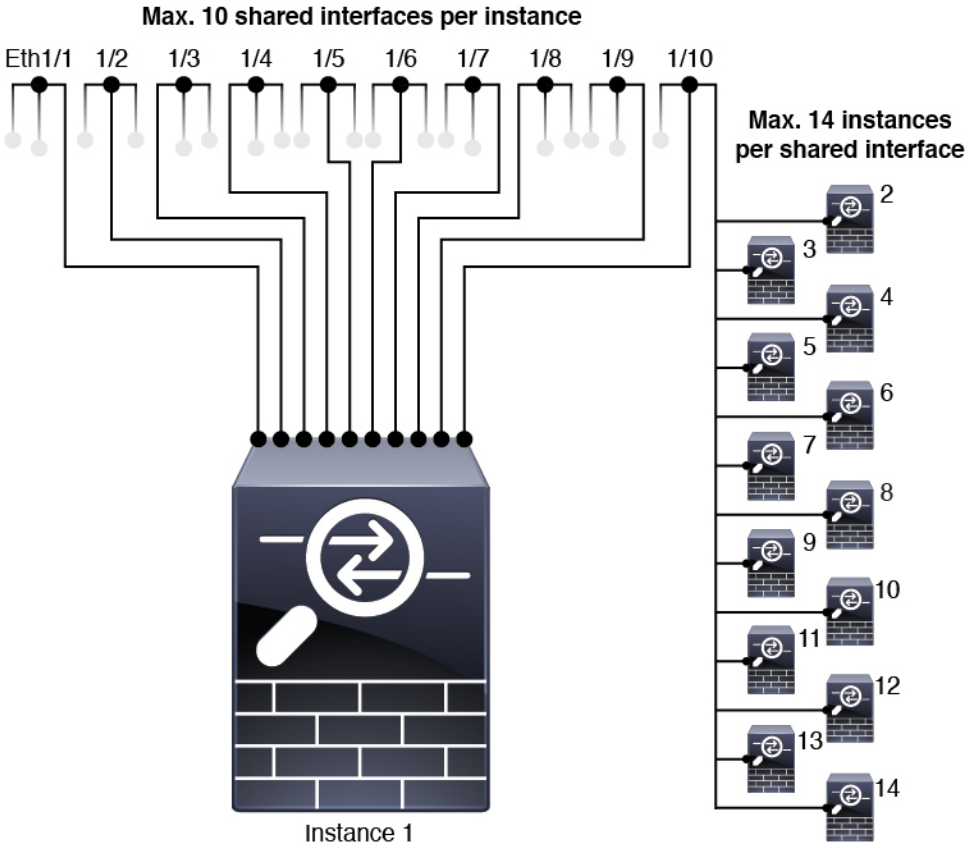
- 하위 인터페이스는 데이터 또는 데이터 공유 유형 인터페이스와 클러스터 유형 인터페이스에서 지원됩니다. 클러스터 인터페이스에 하위 인터페이스를 추가하면 네이티브 클러스터에서 해당 인터페이스를 사용할 수 없습니다
- 다중 인스턴스 클러스터링의 경우 FXOS 하위 인터페이스는 데이터 인터페이스에서 지원되지 않습니다. 그러나 하위 인터페이스는 클러스터 제어 링크에 대해 지원되므로 전용 EtherChannel 또는 EtherChannel의 하위 인터페이스를 클러스터 제어 링크에 사용할 수 있습니다. 애플리케이션 정의 하위 인터페이스는 데이터 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 위협 방어 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 위협 방어 인라인 집합 또는 패시브 인터페이스와 함께 사용할 수 없습니다.
- 데이터 공유 인터페이스는 페일오버 링크용으로 사용할 수 없습니다.

인라인 집합 **Threat Defense**

- 물리적 인터페이스(일반 포트와 breakout 포트 둘 다) 및 EtherChannel용으로 지원됩니다. 하위 인터페이스는 지원되지 않습니다.
- 링크 상태 전파가 지원됩니다.
- 하드웨어 바이패스를 활성화하고 동일한 인라인 세트에 대해 상태 전파를 연결하지 마십시오.

하드웨어 바이패스

- 위협 방어용으로 지원됩니다. ASA용 일반 인터페이스로 사용할 수 있습니다.
- 위협 방어에서는 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.
- 하드웨어 바이패스를 활성화하고 동일한 인라인 세트에 대해 상태 전파를 연결하지 마십시오.

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, 24 페이지의 내용을 참조하십시오.

일반 지침 및 제한 사항

방화벽 모드

위협 방어의 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 인스턴스

- 컨테이너 인스턴스와의 다중 인스턴스 기능은 management center를 사용하는 위협 방어에서만 사용 가능합니다.
- 위협 방어 컨테이너 인스턴스의 경우에는 단일 management center에서 보안 모듈/엔진의 모든 인스턴스를 관리해야 합니다.
- 위협 방어 컨테이너 인스턴스의 경우에는 다음 기능이 지원되지 않습니다.
  - Radware DefensePro 링크 데코레이터
  - Management Center UCAPL/CC 모드
  - 하드웨어로의 플로우 오프로드

인터페이스 구성

기본적으로 물리적 인터페이스는 비활성화되어 있습니다. 인터페이스 활성화, EtherChannels 추가, VLAN 하위 인터페이스 추가, 인터페이스 속성 수정 구성 작업을 수행할 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.

참고 QSFPH40G-CUxM의 경우, 자동 협상은 기본값으로 항상 활성화되어 있으며 비활성화할 수 없습니다.

시작하기 전에

- 이미 EtherChannel의 멤버인 인터페이스는 개별적으로 편집할 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.

인터페이스 유형 사용에 대한 자세한 내용은 인터페이스 유형, 2 페이지를 참고하십시오.

- 데이터

- 데이터 공유 - 컨테이너 인스턴스에만 해당됩니다.
- 관리
- **Firepower** - 위협 방어에만 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) 명시적으로 디바운스 시간**(ms)**을 구성합니다. 0~15000밀리초 사이의 값을 입력합니다.

단계 **9** **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.

- 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.

인터페이스 유형 사용에 대한 자세한 내용은 인터페이스 유형, 2 페이지를 참고하십시오.

- 데이터
- 데이터 공유 - 컨테이너 인스턴스에만 해당됩니다.
- 관리
- **Firepower** - 위협 방어에만 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

네트워크 구축에 따라 섀시에 VLAN 하위 인터페이스 250~500개를 추가할 수 있습니다. 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. threat defense 애플리케이션 내에 하위 인터페이스를 추가할 수 있습니다.FXOS 하위 인터페이스 및 애플리케이션 하위 인터페이스

를 사용하는 시기에 대한 자세한 내용은 FXOS 인터페이스와 애플리케이션 인터페이스 비교, 4 페이지의 내용을 참조하십시오.

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.

인터페이스 유형 사용에 대한 자세한 내용은 인터페이스 유형, 2 페이지를 참고하십시오.

- 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.

논리적 디바이스 구성

Firepower 4100/9300에서 독립형 논리적 디바이스 또는 고가용성 쌍을 추가합니다.

클러스터링에 대해서는 Firepower 4100/9300 클러스터링의 내용을 참조하십시오.

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, 29 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.

위협 방어 인스턴스를 management center에 추가한 후 리소스 프로파일 설정을 변경하는 경우 management center **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.

Management Center 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 threat defense)을 설치할 수 있습니다. 개별 모듈에서 애플리케이션 인스턴스 유형의 서로 다른 버전을 실행할 수도 있습니다.

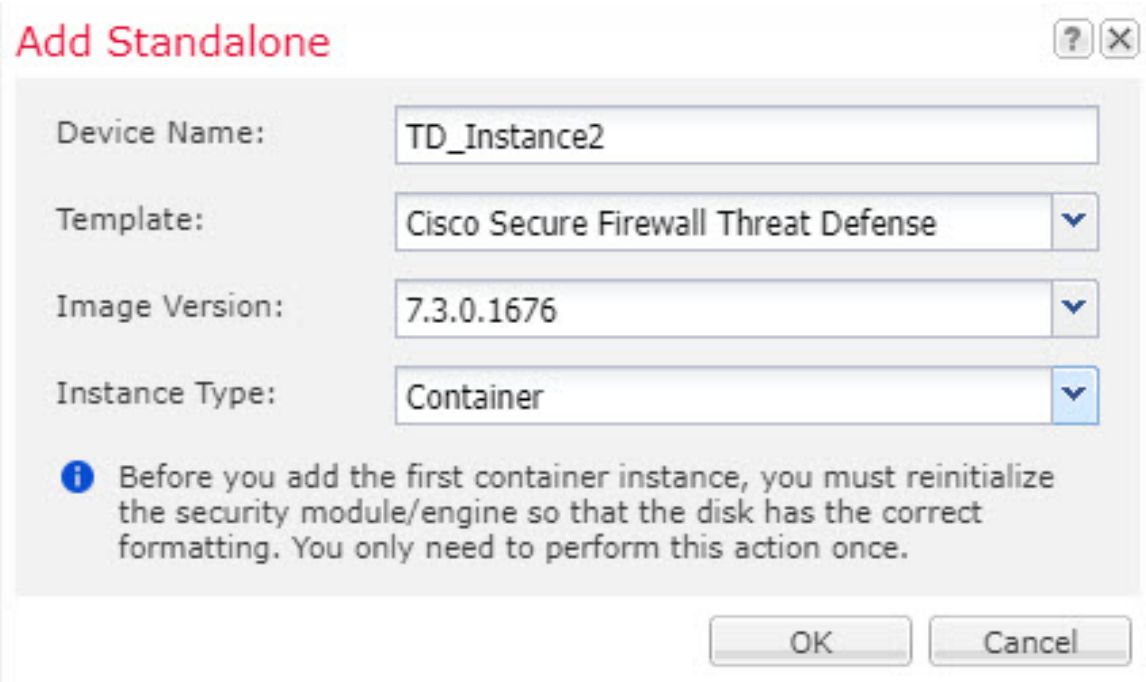
- 논리적 디바이스에 사용할 관리 인터페이스를 구성합니다. 관리 인터페이스는 필수 항목입니다. 이 관리 인터페이스는 섀시 관리용으로만 사용되는 섀시 관리 포트(**Interfaces(**인터페이스**)** 탭 상단에 **MGMT(**관리**)**로 표시됨)와는 다릅니다.
- 나중에 데이터 인터페이스에서 관리를 활성화할 수 있습니다. 데이터 관리를 활성화한 후 이를 사용하지 않으려는 경우에도 관리 인터페이스를 논리적 디바이스에 할당해야 합니다. 자세한 내용은 FTD 명령 참조의 **configure network management-data-interface** 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, 2 페이지를 참조하십시오.
- 컨테이너 인스턴스의 경우 기본 프로필을 사용하지 않으려면 컨테이너 인스턴스에 대한 리소스 프로파일 추가, 39 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.

- 다음 정보를 수집합니다.
  - 이 디바이스의 인터페이스 ID
  - 관리 인터페이스 IP 주소 및 네트워크 마스크
  - 게이트웨이 IP 주소
  - management center 선택한 IP 주소 및/또는 NAT ID
  - DNS 서버 IP 주소
  - 위협 방어 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.

참고 논리적 디바이스를 추가한 후에는 이 이름을 변경할 수 없습니다.

b) **Template(**템플릿**)**에서 **Cisco Firepower Threat Defense**를 선택합니다.

c) **Image Version(**이미지 버전**)**을 선택합니다.

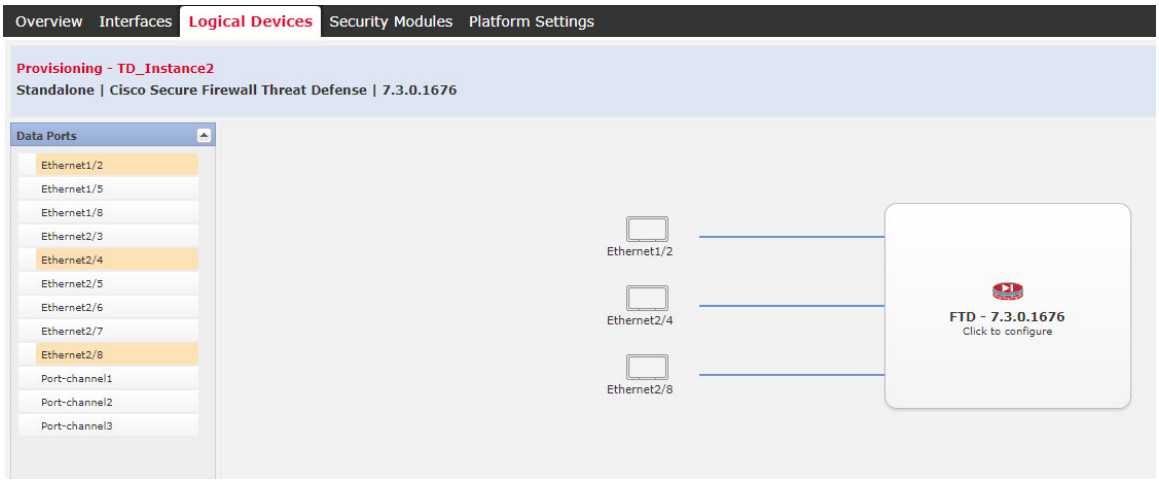
d) **Instance Type(**인스턴스 유형**)**을 **Container(**컨테이너**)** 또는 **Native(**기본**)** 중에서 선택합니다.

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 management center에서 이러한 인터페이스를 활성화하고 구성하게 됩니다.

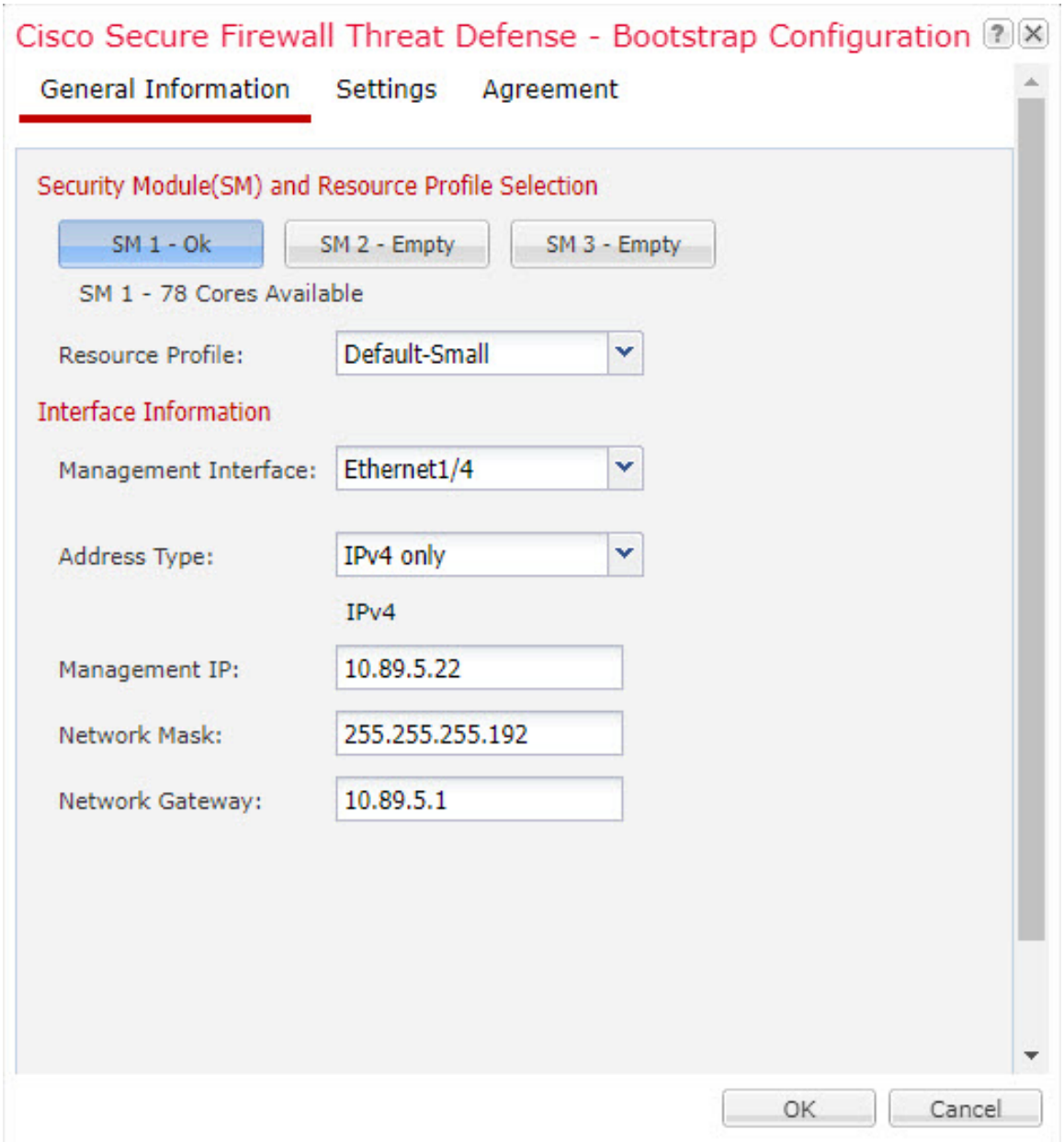
컨테이너 인스턴스에는 데이터 공유 인터페이스를 10개까지만 할당할 수 있습니다. 또한 각 데이터 공유 인터페이스는 최대 14개의 컨테이너 인스턴스에 할당할 수 있습니다. 데이터 공유 인터페이스는 공유 아이콘( )으로 표시됩니다.

하드웨어 바이패스 지원 포트가 아이콘( )과 함께 표시됩니다. 특정 인터페이스 모듈의 경우 인라인 집합 인터페이스에 대해서만 하드웨어 우회 기능을 활성화할 수 있습니다(management center 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 있습니다. 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 쌍 또는 클러스터에 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.

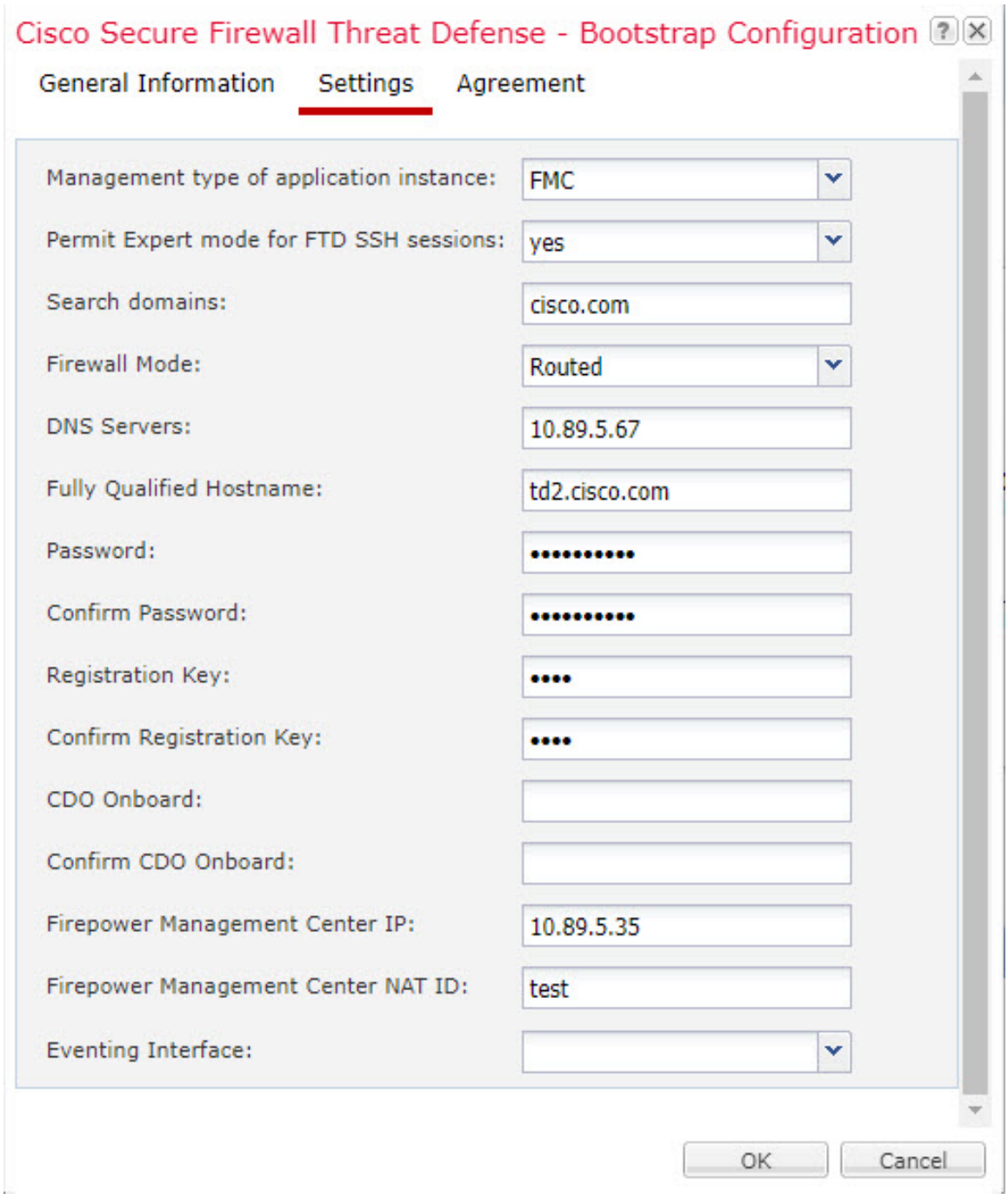
e) **Management IP(**관리 **IP)** 주소를 구성합니다.

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 device manager을 관리자로도 지원합니다. 논리적 디바이스를 구축한 후에는 관리자 유형을 변경할 수 없습니다.

b) management center 관리에 사용할 **Firepower Management Center IP**를 입력합니다. management center 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 위협 방어 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 위협 방어 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 위협 방어는 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, management center의 호스트 이름을 지정하는 경우, 위협 방어에서는 DNS를 사용합니다.

g) 위협 방어의 **Fully Qualified Hostname(**정규화된 호스트 이름**)**을 입력합니다.

h) 등록 시 management center와 디바이스 간에 공유할 **Registration Key(**등록 키**)**를 입력합니다.

이 키에 대해 1~37자의 텍스트 문자열을 선택할 수 있습니다. 위협 방어를 추가하는 경우 management center에 동일한 키를 입력합니다.

i) 위협 방어 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.

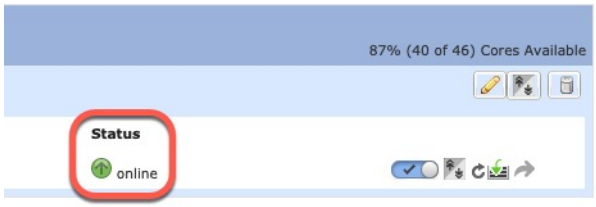
이 설정은 하드웨어에서 TLS 암호화 가속화를 활성화하고 특정 유형의 트래픽에 대한 성능을 개선합니다. 이 기능은 기본적으로 활성화되어 있습니다. 보안 모듈당 최대 16개의 인스턴스에 대해 TLS 암호화 가속화를 활성화할 수 있습니다. 이 기능은 네이티브 인스턴스에서 항상 사용할 수 있습니다. 이 인스턴스에 할당된 하드웨어 암호화 리소스의 백분율을 보려면 **show hw-crypto** 명령을 입력합니다.

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** 위협 방어를 매니지드 디바이스로 추가하고 보안 정책 구성을 시작하려면 management center 구성 가이드를 참조합니다.

고가용성 쌍 추가

Threat Defense 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, 29 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. 고가용성 섹션을 참조하십시오.

단계 **4** 고가용성을 활성화한 후에 인터페이스를 변경해야 하는 경우에는 먼저 스탠바이 유닛에서 변경을 수행한 다음 액티브 유닛에서 변경을 수행합니다.

Threat Defense 논리적 디바이스에서 인터페이스 변경

위협 방어 논리적 디바이스에서 인터페이스를 할당 또는 할당 해제하거나 관리 인터페이스를 교체할 수 있습니다. 그런 다음 management center에서 인터페이스 구성을 동기화할 수 있습니다.

새 인터페이스를 추가하거나 사용하지 않는 인터페이스를 삭제하는 경우 위협 방어 구성에 미치는 영향은 아주 적습니다. 그러나 보안 정책에 사용되는 인터페이스를 삭제하면 구성에 영향이 미칩니다. 액세스 규칙, NAT, SSL, ID 규칙, VPN, DHCP 서버 등 위협 방어 구성의 여러 위치에서 인터페이스를 직접 참조할 수 있습니다. 보안 영역을 참조하는 정책은 영향을 받지 않습니다. 논리적 디바이스에 영향을 주거나 management center에서 동기화할 필요 없이 할당된 EtherChannel의 멤버십을 수정할 수도 있습니다.

인터페이스를 삭제하면 해당 인터페이스와 연결된 모든 구성이 삭제됩니다.

시작하기 전에

- 인터페이스를 구성하고 실제 인터페이스 구성, 34 페이지 및 EtherChannel(포트 채널) 추가, 35 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. threat defense 디바이스가 리부팅되고(관리 인터페이스를 변경하면 리부팅됨) management center에서 구성을 동기화한 후에는 이제 할당 해제된 관리 인터페이스를 EtherChannel에 추가할 수도 있습니다.
- 클러스터링 또는 고가용성의 경우에는 management center에서 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.

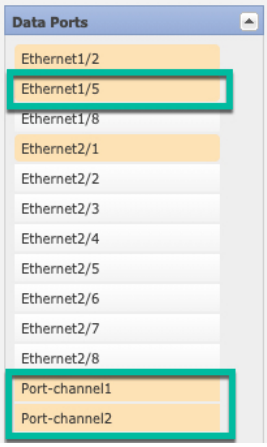
프로시저

단계 **1** 섀시 관리자에서 **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 management center에서 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** management center에서 인터페이스를 동기화합니다.

a) management center에 로그인합니다.
b) **Devices(**디바이스**)** > **Device Management(**디바이스 관리**)**를 선택하고 위협 방어 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.

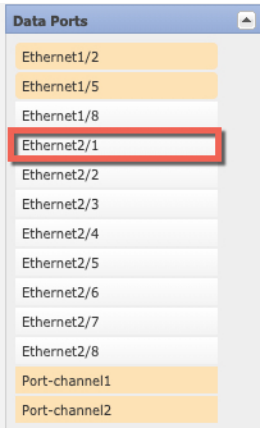
오류가 발생하는 경우 정책을 변경하고 유효성 검사를 다시 실행해야 합니다.

g) **Save(**저장**)**를 클릭합니다.

h) **Deploy(**구축**)** > **Deployments(**구축하기**)**를 클릭합니다.

i) 디바이스를 선택하고 **Deploy(**구축**)**를 클릭하여 할당된 디바이스에 정책을 구축합니다. 변경사항은 구축할 때까지 활성화되지 않습니다.

단계 **7** 섀시 관리자에서 **Data Ports(**데이터 포트**)** 영역에서 인터페이스를 선택 취소하여 데이터 인터페이스를 할당 해제합니다.



단계 **8** **Save(**저장**)**를 클릭합니다.

단계 **9** management center에서 인터페이스를 다시 동기화합니다.

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.

connect ftd *name*

인스턴스 이름을 확인하려면 이름 없이 명령을 입력합니다.

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.

- Threat Defense - **exit**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.

논리적 디바이스의 기록

기능	버전	세부 사항
threat defense 작동 링크 상태와 물리적 링크 상태 간 동기화	6.7	이제 섀시가 threat defense 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. threat defense 애플리케이션 인터페이스 관리 상태는 고려되지 않습니다. 예를 들어 threat defense에서 동기화하지 않으면 threat defense 애플리케이션이 완전히 온라인 상태가 되기 전에 데이터 인터페이스가 물리적으로 작동 상태가 되거나 threat defense 종료를 시작한 후 일정 기간 동안 작동 상태를 유지할 수 있습니다. 인라인 집합의 경우 threat defense에서 트래픽을 처리하기 전에 외부 라우터가 threat defense로 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 threat defense에는 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 management center를 사용하여 Threat Defense 구성 백업 및 복원	6.7	이제 threat defense 컨테이너 인스턴스에서 management center 백업/복원 도구를 사용할 수 있습니다. 신규/수정된 management center 화면: **System(**시스템**) > Tools(**도구**) > Backup/Restore(**백업/복원**) > Managed Device Backup(**매니지드 디바이스 백업**)** 신규/수정된 threat defense CLI 명령: **restore** 지원되는 플랫폼: Firepower 4100/9300 참고 FXOS 2.9가 필요합니다.

기능	버전	세부 사항
클러스터 유형 인터페이스의 VLAN 하위 인터페이스 지원(다중 인스턴스 전용)	6.6	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 인터페이스가 허용됩니다. 신규/수정된 Firepower Chassis Manager 화면: **Interfaces(**인터페이스**)** > **All Interfaces(**모든 인터페이스**)** > **Add New(**새로 추가**)** 드롭다운 메뉴 **> Subinterface(**하위 인터페이스**)** > **Type(**유형**)** 필드 신규/수정된 FXOS 명령: **set port-type** 클러스터 참고 FXOS 2.8.1이 필요합니다.
Threat Defense Firepower 4112	6.6	Firepower 4112가 도입되었습니다. 참고 FXOS 2.8.1이 필요합니다.
여러 컨테이너 인스턴스에 대한 TLS 암호화 가속	6.5	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 FXOS 2.7.1이 필요합니다.
Threat Defense Firepower 4115, 4125, 4145	6.4	Firepower 4115, 4125, 및 4145를 도입했습니다. 참고 FXOS 2.6.1.157이 필요합니다.
Firepower 9300 SM-40, SM-48 및 SM-56지원	6.4	다음 세 가지 보안 모듈을 도입했습니다: SM-40, SM-48, SM-56 참고 FXOS 2.6.1.157이 필요합니다.

기능	버전	세부 사항
동일한 Firepower 9300의 별도의 모듈에서 ASA 및 threat defense에 대한 지원	6.4	이제 동일한 Firepower 9300에서 ASA 및 threat defense 논리적 디바이스를 구축할 수 있습니다. 참고 FXOS 2.6.1.157이 필요합니다.
모듈/보안 엔진에서 하나의 threat defense 컨테이너 인스턴스에 대한 SSL 하드웨어 가속 지원	6.4	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. 신규/수정된 FXOS 명령: **config hwCrypto enable** 수정된 화면이 없습니다. 참고 FXOS 2.6.1.157이 필요합니다.

기능	버전	세부 사항
Firepower 4100/9300의 threat defense에 대한 다중 인스턴스 기능	6.3	이제 단일 보안 엔진/모듈에서 여러 논리적 디바이스를 각각 threat defense 컨테이너 인스턴스와 함께 구축할 수 있습니다. 이전에는 단일 기본 애플리케이션 인스턴스만 구축할 수 있었습니다. 물리적 인터페이스를 유연하게 사용할 수 있도록 FXOS에서 VLAN 하위 인터페이스를 생성하는 동시에 여러 인스턴스 간에 인터페이스를 공유할 수 있습니다. 리소스 관리를 사용하면 각 인스턴스에 대한 성능 기능을 맞춤화할 수 있습니다. 2개의 개별 섀시에서 컨테이너 인스턴스를 활용한 고가용성을 사용할 수 있습니다. 클러스터링은 지원되지 않습니다. 참고 다중 인스턴스 기능은 ASA 다중 컨텍스트 모드와 비슷하지만 구현은 서로 다릅니다. threat defense에서는 다중 상황 모드를 사용할 수 없습니다. 신규/수정된 management center 화면: • **Devices(**디바이스**)** > **Device Management(**디바이스 관리**)** > **Edit(**수정**)** 아이콘 **>** **Interfaces(**인터페이스**)** 탭 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 FXOS 명령: **connect ftd** *name*, **connect module telnet, createbootstrap-key PERMIT_EXPERT_MODE, create resource-profile, create subinterface, scope auto-macpool, set cpu-core-count, set deploy-type, set port-type data-sharing, set prefix, set resource-profile-name, setvlan, scope app-instance ftd** *name***, show cgroups container, show interface, show mac-address, show subinterface, show tech-support module app-instance, show version** 지원되는 플랫폼: Firepower 4100/9300

기능	버전	세부 사항
Firepower 4100/9300에 대한 클러스터 제어 링크의 맞춤화 가능한 IP 주소	6.3	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 Firepower Chassis Manager 화면: • **Logical Devices(**논리적 디바이스**)** > **Add Device(**디바이스 추가**)** > **Cluster Information(**클러스터 정보**)** > **CCL Subnet IP(CCL** 서브넷 **IP)** 필드 신규/수정된 FXOS 명령: **set cluster-control-link network** 지원되는 플랫폼: Firepower 4100/9300
On(켜기) 모드에서 데이터 EtherChannel 지원	6.3	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 Firepower Chassis Manager 화면: • **Interfaces(**인터페이스**)** > **All Interfaces(**모든 인터페이스**)** > **Edit Port Channel(**포트 채널 수정**)** > **Mode(**모드**)** 신규/수정된 FXOS 명령: **set port-channel-mode** 지원되는 플랫폼: Firepower 4100/9300
threat defense 인라인 집합에서 EtherChannel 지원	6.2	이제 threat defense 인라인 집합에서 EtherChannel을 사용할 수 있습니다. 지원되는 플랫폼: Firepower 4100/9300
threat defense 모듈 6개를 위한 섀시 간 클러스터링	6.2	이제 threat defense를 위한 섀시 간 클러스터링을 활성화할 수 있습니다. 최대 6개의 섀시에 최대 6개의 모듈을 포함할 수 있습니다. 신규/수정된 Firepower Chassis Manager 화면: • 논리적 디바이스 > 구성 지원되는 플랫폼: Firepower 4100/9300

기능	버전	세부 사항
지원되는 네트워크 모듈용 Firepower 4100/9300에 대한 하드웨어 우회 지원	6.1	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 화면: • **Devices(**디바이스**)** > **Device Management(**디바이스 관리**)** > **Interfaces(**인터페이스**)** > **Edit Physical Interface(**물리적 인터페이스 수정**)** 지원되는 플랫폼: Firepower 4100/9300
인라인 집합 링크 상태 전파 지원 threat defense	6.1	threat defense 애플리케이션에서 인라인 집합을 구성하고 링크 상태 전파를 활성화하면 threat defense에서 FXOS 섀시로 인라인 집합 멤버십을 전송합니다. 링크 상태 전파는 인라인 집합의 인터페이스 중 하나가 중단될 때 인라인 인터페이스 쌍에서 두 번째 인터페이스를 자동으로 불러옵니다. 신규/수정된 FXOS 명령: **show fault |grep link-down**, **show interface detail** 지원되는 플랫폼: Firepower 4100/9300
Firepower 9300의 threat defense에서 인트라 섀시 클러스터링(intra-chassis clustering) 지원	6.0.1	Firepower 9300은 threat defense 애플리케이션이 있는 인트라 섀시 클러스터링(intra-chassis clustering)을 지원합니다. 신규/수정된 Firepower Chassis Manager 화면: • 논리적 디바이스 > 설정 신규/수정된 FXOS 명령: **enter mgmt-bootstrap ftd, enter bootstrap-key FIREPOWER_MANAGER_IP, enter bootstrap-key FIREWALL_MODE, enter bootstrap-key-secret REGISTRATION_KEY, enter bootstrap-key-secret PASSWORD, enter bootstrap-key FQDN, enter bootstrap-key DNS_SERVERS, enter bootstrap-key SEARCH_DOMAINS, enter ipv4 firepower, enter ipv6 firepower, set value, set gateway, set ip, accept-license-agreement** 지원되는 플랫폼: Firepower 4100/9300